

UNIVERSIDAD AMERICANA
Facultad: Ingeniería y Arquitectura
Carrera: Ingeniería en Sistemas de Información
Clase: Introducción a la Programación



**PROGRAMA DE INVENTARIO
COSMÉTICOS CASTILLO**

Integrantes:

- Hendrix Devadip Aguirre Zamora
- Shylon Emeil Andrews Hunter
- Mia Kristeen Flores Silva

Docente:

- Prof. Silvia Gigdalia Ticay López

04 de Julio del 2025

ÍNDICE

I.	INTRODUCCIÓN	1
II.	OBJETIVOS	2
2.1.	OBJETIVO GENERAL.....	2
2.2.	OBJETIVOS ESPECÍFICOS.....	2
2.2.1.	Crear funciones que permitan el ingreso de nuevos productos en el inventario.....	2
2.2.2.	Implementar funciones de modificación que faciliten la actualización de la información de los productos.	2
2.2.3.	Desarrollar un módulo que permita eliminar productos del inventario de manera segura y permanente, evitando errores humanos.....	2
2.2.4.	Crear un sistema de visualización que muestre de forma clara y ordenada todos los productos existentes en el inventario.....	2
2.2.5.	Incluir un sistema de ingreso mediante usuario y contraseña, para proteger el acceso al sistema.....	2
III.	ANÁLISIS DEL PROBLEMA	3
3.1.	REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES	3
3.1.1.	Requerimientos funcionales.....	3
3.1.2.	Requerimientos no funcionales	4
IV.	DISEÑO DE ALGORITMO.....	4
V.	CODIFICACIÓN, EJECUCIÓN, VERIFICACIÓN Y DEPURACIÓN	4
VI.	DIAGRAMA DEL PROGRAMA.....	7
VII.	CONCLUSIONES	8
VIII.	RECOMENDACIONES	8

I. INTRODUCCIÓN

La gestión de inventario representa un aspecto fundamental en el funcionamiento de cualquier actividad comercial, especialmente en los pequeños negocios, donde el control adecuado de los productos disponibles incide directamente en la rentabilidad y eficiencia del emprendimiento. En el caso particular de una venta de cosméticos, el manejo organizado de los artículos es crucial, dado que se trata de productos con variedad de presentaciones, marcas, precios y fechas de vencimiento. Una administración manual puede dar lugar a errores frecuentes, como el registro incorrecto de entradas y salidas, la acumulación de productos vencidos, el desconocimiento del stock real o la pérdida de ventas por falta de existencias.

Frente a esta necesidad, se desarrolló un programa de inventario para la venta de “Cosméticos Castillo”, mediante la utilización de pseudocódigo (PSeint) y una programación estructurada (Python), el cual, está enfocado en cubrir las demandas de una pequeña venta de cosméticos. Este sistema tiene como propósito principal facilitar el registro, actualización y monitoreo de los productos, proporcionando al usuario una herramienta simple pero eficiente para mantener el control del negocio. Entre sus funciones básicas se incluyen el ingreso de nuevos artículos, la modificación de información existente, la eliminación de productos y la visualización general del inventario.

Además, el programa busca brindar mayor agilidad al proceso de gestión, reducir el margen de error humano, y generar información útil para la toma de decisiones o cuáles necesitan ser reabastecidos. Al ser un recurso adaptable y de fácil uso, se convierte en una solución práctica y accesible para emprendedores que no cuentan con grandes recursos tecnológicos, pero que desean profesionalizar la administración de su negocio.

Este documento presenta el diseño, desarrollo y funcionalidades principales del programa, así como su relevancia dentro del entorno comercial de una pequeña venta de cosméticos, evidenciando los beneficios concretos que puede aportar en la optimización de los procesos internos.

II. OBJETIVOS

2.1. OBJETIVO GENERAL

Desarrollar un programa de inventario que facilite el registro de productos del comercio de cosméticos.

2.2. OBJETIVOS ESPECÍFICOS

- 2.2.1.** Crear funciones que permitan el ingreso de nuevos productos en el inventario.
- 2.2.2.** Implementar funciones de modificación que faciliten la actualización de la información de los productos.
- 2.2.3.** Desarrollar un módulo que permita eliminar productos del inventario de manera segura y permanente, evitando errores humanos.
- 2.2.4.** Crear un sistema de visualización que muestre de forma clara y ordenada todos los productos existentes en el inventario.
- 2.2.5.** Incluir un sistema de ingreso mediante usuario y contraseña, para proteger el acceso al sistema.

III. ANÁLISIS DEL PROBLEMA

El sistema fue diseñado para la gestión manual del inventario en “Cosméticos Castillo”, el cual, presenta múltiples desafíos, como errores en el registro de productos, desconocimiento del inventario real y pérdidas por falta de existencias. Estos problemas afectan la eficiencia del negocio y dificultan la toma de decisiones informadas. Por ello, se requiere de un programa que automatice y facilite el control del inventario de manera organizada, precisa y en tiempo real.

El alcance de esta solución se limita a la gestión de órdenes básicas, incluyendo funciones como: ingreso mediante usuario y contraseña, visualización de menú, agregar productos al inventario, actualizar datos de productos existentes, eliminar productos del inventario, visualizar inventario y generación de inventario en archivo txt. No permite integración con plataformas digitales o sistemas en redes. Pero, a pesar de no ser capaz de lograr esos aspectos, se presenta al comercio como un prototipo funcional que permite demostrar de manera efectiva cómo los conocimientos que adquirimos durante las clases pueden convertirse en herramientas de alta utilidad en la vida diaria.

3.1. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

3.1.1. Requerimientos funcionales

Los requerimientos funcionales son las acciones que el sistema debe ser capaz de realizar para cumplir con su objetivo. En el caso del programa de inventario para Cosméticos Castillo, se incluyen:

- Permitir el registro de nuevos productos al inventario, ingresando nombre, cantidad y precio unitario
- Habilitar la actualización de la información de los productos ya existentes en el inventario
- Posibilitar la eliminación de productos del inventario de forma permanente
- Mostrar el inventario completo de forma ordenada y legible
- Gestionar el acceso mediante usuario y contraseña, para así garantizar un control de seguridad básico
- Guardar los datos en un archivo de texto (.txt) para que la información se mantenga en distintas sesiones del programa
- Cargar automáticamente los datos guardados al iniciar el programa, tanto del inventario como de los usuarios

3.1.2. Requerimientos no funcionales

Los requerimientos no funcionales definen las características generales que debe tener el sistema para garantizar su eficiencia, facilidad de uso y estabilidad. Para este proyecto se establecen los siguientes requerimientos:

- **Facilidad de uso:** la interfaz debe ser clara y sencilla, mediante menús de texto comprensibles para personas sin conocimientos técnicos avanzados
- **Ejecución en entorno local:** el programa debe funcionar sin necesidad de conexión a internet
- **Compatibilidad:** debe ejecutarse correctamente en sistemas operativos comunes, especialmente en Windows
- **Persistencia de datos:** La información debe guardarse de manera segura en archivos externos (.txt), permitiendo su recuperación incluso después de cerrar el programa
- **Estructura modular:** el código debe estar organizado en funciones y módulos específicos (registro, actualización, eliminación y visualización), lo cual facilita el mantenimiento y futuras ampliaciones
- **Seguridad básica:** el acceso al sistema debe requerir autenticación, mediante usuario y contraseña
- **Funcionamiento eficiente:** el sistema debe ejecutar las operaciones CRUD de forma rápida, sin errores y sin atrasos para el usuario

IV. DISEÑO DE ALGORITMO

Primeramente, se elaboró un diseño en pseudocódigo con PSeint, que es un entorno de desarrollo integrado (IDE) para aprender a programar, especialmente diseñado para principiantes. El cual, fue diseñado mediante un orden lógico que diera solución a nuestra problemática, que es el correcto inventario de la venta de cosméticos.

V. CODIFICACIÓN, EJECUCIÓN, VERIFICACIÓN Y DEPURACIÓN

El diseño de algoritmo de PSeint fue implementado en el lenguaje de programación de alto nivel Python, mediante el uso de Visual Studio Code, que contiene todas las funciones necesarias para que el programa funcione correctamente y, además, es conocido por su flexibilidad y personalización gracias a un amplio ecosistema de extensiones que lo hacen ser compatible con numerosos lenguajes de programación y entornos de desarrollo, como en nuestro caso, Python.

La codificación en el lenguaje de programación Python, fue mucho menos complicada, ya que, ya contábamos con un orden de solución gracias al algoritmo elaborado en PSeint. Solamente se necesitaron hacer algunos ajustes que no se podían lograr en PSeint.

Fragmentos del código:

```
1 #Inventario de venta de cosméticos Castillo
2 import os
3 import pwinput
4
```

Se importan librerías necesarias para la verificación de existencia de un archivo y para ocultar la contraseña ingresada por el usuario mediante un caracter debidamente establecido.

```
def agregarProducto():
    nombre=input("Ingrese el nombre del producto: ").upper() # upper para cambiar a mayúsculas todas las letras
    for producto in inventario:
        if producto[0].strip().upper() == nombre.strip().upper():
            print ("El producto ya existe en el inventario!")
            return

    cantidad=int(input("Ingrese la cantidad disponible: "))
    precio=float(input("Ingrese el precio por unidad en córdobas: "))

    producto=[nombre,str(cantidad),str(precio)] # str para cambiar el valor de entero a cadena
    inventario.append(producto)

    with open("inventario.txt","a",encoding="utf-8") as archivo:
        archivo.write(",".join(producto) + "\n")

    print ("Producto agregado con éxito al inventario!")

#Se define función para mostrar el inventario
def mostrarInventario():
    print ("\n Inventario actual:")
    if len(inventario)==0:
        print ("No hay productos para mostrar "
              "en el inventario!\n")
    else:
        for i in range(len(inventario)):
            producto=inventario[i]
            print (f"{i+1}. {producto[0]} - cantidad: {producto[1]} - precio: C${producto[2]}")
        print ()
```

Se definen funciones en un módulo para el funcionamiento del programa, por ejemplo, la agregación de productos al inventario y el mostrar inventario. Los cuales se llevan a cabo mediante procesos detallados y precisos para su correcto funcionamiento. Todos los procesos presentados en la imagen, fueron aprendidos en clases durante todas las sesiones.

```
usuarios.txt X
usuarios.txt
1 Hendrix,24013755
2 Mia,25011913
3 Shylon,25012339

inventario.txt X
inventario.txt
1 SOUL LITE,5,5000.0
2 GLACIER,6,2100.0
3 ASCENDANT,3,1900.0
4 ECLAT,6,2100.0
5
```

El inventario es debidamente guardado en archivo txt, el cual es leído cada vez que el usuario desea cambiar algo en el inventario. Además, se cargan los usuarios con sus contraseñas, para que sean leídos por el programa y que pueda verificar el correcto acceso.

```
PROGRAMAPRINCIPAL.PY > menu
1 #Inventario de venta de cosméticos Castillo
2
3 import Funciones # Se importan todas las funciones ya antes definidas
4
5 #Se define función para llamar al menú
6 def menu():
7     while True:
8         print ("---- Menú ----")
9         print ("1. Agregar producto")
10        print ("2. Actualizar producto")
11        print ("3. Eliminar producto")
12        print ("4. Mostrar inventario")
13        print ("5. Salir")
14
15        opción=input("Seleccione una opción (1/2/3/4/5): ")
16        if opción=="1":
17            Funciones.agregarProducto() # Se llama a la función agregarProductos del módulo Funciones
18        elif opción=="2":
19            Funciones.actualizarProducto() # Se llama la función actualizarProducto del módulo Funciones
20        elif opción=="3":
21            Funciones.eliminarProducto() # Se llama a la función eliminarProducto del módulo Funciones
22        elif opción=="4":
23            Funciones.mostrarInventario() # Se llama a la función mostrarInventario del módulo Funciones
24        elif opción=="5":
25            print ("Gracias por hacer uso del sistema. Hasta luego!")
26            break
27        else:
28            print ("Opción inválida. Intente de nuevo!")
29
30 # Si se cumple la función inicio se llama a las funciones cargarInventario y menu
31 if Funciones.inicio():
32     Funciones.cargarInventario()
33     menu()
```


Se crea un programa principal en el que se importa el módulo con todas las funciones definidas anteriormente, para luego llamarlas y que así se pueda desarrollar el programa de manera más estética y correcta.

VI. DIAGRAMA DEL PROGRAMA

Se identificó que el sistema debía resolver una necesidad concreta: llevar el control eficiente del inventario de una pequeña venta de cosméticos. Para ello, el programa debía permitir:



VII. CONCLUSIONES

La elaboración del presente proyecto permitió aplicar de manera práctica los conocimientos adquiridos durante la clase de Introducción a la Programación. A través del diseño y desarrollo del sistema de inventario para la venta de cosméticos, se logró integrar conceptos esenciales como estructuras de control, manejo de listas, funciones y almacenamiento de datos mediante archivos de texto.

El sistema desarrollado ofrece funcionalidades básicas de gestión de inventario, tales como el registro, modificación, eliminación y visualización de productos, así como un control de acceso mediante autenticación. Estas funciones contribuyen a optimizar el manejo de productos en pequeños negocios, permitiendo un mayor orden, eficiencia y facilidad en el proceso de control de existencias.

Asimismo, se comprobó la importancia de una planificación estructurada del código mediante módulos específicos, lo cual favoreció la organización lógica del sistema y facilitó su ejecución. El enfoque modular también representa una ventaja significativa para futuras mejoras o adaptaciones del programa.

En conclusión, el proyecto cumplió con los objetivos propuestos, demostrando que es posible desarrollar soluciones tecnológicas funcionales con herramientas básicas y principios elementales de programación estructurada. El sistema representa una alternativa sencilla y eficaz para la gestión de inventarios en pequeños emprendimientos.

VIII. RECOMENDACIONES

- **Agregar una interfaz más visual**
Aunque el menú que hicimos funciona bien, sería buena idea que más adelante se intente crear una versión con botones o ventanas, para que cualquier persona pueda usarlo sin tener que escribir nada.
- **Guardar los datos de forma más segura**
Nosotros usamos archivos `.txt` porque es lo que aprendimos en clase, pero más adelante se podría usar algo más avanzado como una base de datos, para guardar mejor la información si el negocio crece.
- **Hacer búsquedas más completas**
Actualmente, solo se puede buscar un producto por su nombre exacto. Se podría mejorar esto permitiendo buscar por palabras clave o mostrar sugerencias mientras se escribe.
- **Seguir practicando proyectos similares**
Este fue nuestro primer proyecto de programación y nos ayudó mucho a entender cómo se construye un sistema desde cero. Por eso, creemos que seguir haciendo ejercicios similares sería muy útil para reforzar todo lo que hemos aprendido.