

UNIVERSIDAD AMERICANA

*Clase: Introducción a la Programación*

---



## VENTA DE COSMÉTICOS CASTILLO

### **Integrantes:**

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

### **Docente:**

- ❖ Prof. Silvia Gigdalia Ticay López

*del 2025*

## **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 en tiempo real, 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 discontinuados 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, cómo saber qué productos se venden más, cuáles necesitan ser reabastecidos o cuándo realizar ofertas para artículos próximos a vencer. 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.** Implementar funciones de modificación que faciliten la actualización de la información de los productos.

**2.2.2.** Desarrollar un módulo que permita eliminar productos del inventario de manera segura y permanente, evitando errores humanos.

**2.2.3.** Crear un sistema de visualización que muestre de forma clara y ordenada todos los productos existentes en el inventario

**2.2.4.** 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 "Venta de 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.

## Entradas requeridas

El sistema necesita ingresar los siguientes datos para funcionar correctamente:

- **Nombre del producto** (Cadena)
- **Precio unitario** (Real)
- **Cantidad en existencia** (Entero)

La cantidad de entradas dependerá del número de productos que se deseen registrar en el inventario.

## Salida deseada

El sistema debe generar y mostrar información útil en diferentes formatos, como:

- **Lista general del inventario actual** (con todos los productos registrados)
- **Consulta específica de productos por nombre**

Las salidas serán principalmente de tipo cadena y enteros, presentadas en pantalla y/o exportables en formato simple como lo es el txt.

## Método para producir la salida deseada

Para obtener estas salidas, el sistema utilizará:

- **Funciones de búsqueda**
- **Operaciones CRUD (create, read, update and delete) sobre los datos**
- **Almacenamiento estructurado (por ejemplo, archivos txt)**

Cada módulo del programa se enfocará en una funcionalidad específica, permitiendo una organización lógica y eficiente del proceso.

## Requisitos y restricciones

- El sistema debe ser **fácil de usar** para personas sin conocimientos técnicos avanzados.
- Debe funcionar en un entorno **local** (sin necesidad de conexión a internet).
- Preferiblemente, debe ser compatible con sistemas operativos comunes como **Windows**.
- Los datos deben **guardarse de forma segura y persistente**.

- El programa debe permitir **actualizaciones fáciles** de los productos existentes.
- Debe incluir una **interfaz gráfica sencilla** o menús de texto claros.

Para abordar el problema de forma más eficiente, se divide en módulos principales:

1. **Módulo de registro de productos**
2. **Módulo de actualización de inventario**
3. **Módulo de eliminación de productos**
4. **Módulo de visualización**

Esta división facilita el diseño, desarrollo y posterior mantenimiento del sistema, asegurando que cada parte cumpla con su función específica sin generar conflictos.

#### **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.

## 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. ANEXOS

```
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: "))

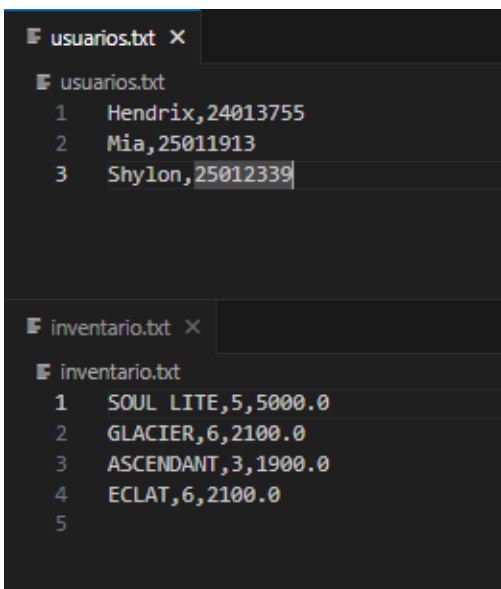
    producto=[nombre,str(cantidad),str(precio)] # str para cambiar el valor de entero a cadena
    inventario.append(producto)

    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
1 Hendrix,24013755
2 Mia,25011913
3 Shylon,25012339

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.