

Proyecto Integrador - Módulo 1

Sistema de Gestión de Inventario (Punto de Venta)

Curso: Python de 0 a Interfaces Gráficas

1. Descripción del Proyecto

El objetivo de este proyecto final es desarrollar una aplicación de consola que permita administrar el inventario de una pequeña tienda. El sistema debe ser capaz de registrar productos, procesar ventas y, lo más importante, ****guardar la información en el disco duro**** para que no se pierda al cerrar el programa.

Este proyecto evalúa tu dominio sobre:

- **Variables y Tipos de Datos:** Manejo de precios (float) y cantidades (int).
- **Control de Flujo:** Uso de menús interactivos y validaciones.
- **POO:** Modelado de datos con Clases y Objetos.
- **Manejo de Errores:** Protección contra datos inválidos.
- **Archivos:** Persistencia de datos (Lectura/Escritura).

2. Requerimientos Técnicos

El sistema debe cumplir con las siguientes funcionalidades:

1. Clase Producto:

- Debe tener atributos: `id`, `nombre`, `precio`, `cantidad`.
- Debe tener un método `vender(cantidad)` que reste del stock solo si hay suficiente mercancía.
- Debe tener un método para convertirse a texto CSV para ser guardado.

2. Persistencia Automática:

- Al iniciar, el programa debe buscar el archivo `inventario.txt` y cargar los productos en memoria.
- Al salir, el programa debe guardar todos los cambios en el archivo.

3. Validación de Datos:

- No se pueden registrar precios negativos.
- Si el usuario ingresa letras en lugar de números para el precio o cantidad, el programa debe avisar del error y no fallar (`try-except`).

3. Estructura del Archivo de Texto

Usaremos un formato CSV simple (Valores Separados por Comas).

Ejemplo: A01,Coca Cola,15.50,100

4. Código Base (Tu Misión)

Completa el siguiente código rellenando los espacios marcados con `TODO`.

```
1 import os
2
3 # --- 1. CLASE MODELO ---
4 class Producto:
5     def __init__(self, id_prod, nombre, precio, cantidad):
6         self.id = id_prod
7         self.nombre = nombre
8         self.precio = precio
9         self.cantidad = cantidad
10
11     def vender(self, cantidad_venta):
12         # TODO: Verificar si hay suficiente stock (self.cantidad)
13         # TODO: Si hay, restar la cantidad y retornar True
14         # TODO: Si no hay, imprimir error y retornar False
15         pass
16
17     def a_texto_csv(self):
18         # TODO: Retornar string formateado: "id,nombre,precio,cantidad"
19         return f"{self.id},{self.nombre},{self.precio},{self.cantidad}"
20
21     def __str__(self):
22         return f"[{self.id}] {self.nombre} - ${self.precio} (Stock: {self.cantidad})"
23
24 # --- 2. CLASE GESTORA ---
25 class Tienda:
26     def __init__(self, archivo):
27         self.archivo = archivo
28         self.inventario = [] # Lista de objetos Producto
29         self.cargar_datos()
30
31     def cargar_datos(self):
32         if not os.path.exists(self.archivo): return
33
34         try:
35             with open(self.archivo, 'r', encoding='utf-8') as f:
36                 for linea in f:
37                     # TODO: Limpiar linea (.strip) y separar por comas
38                     # TODO: Convertir precio a float y cantidad a int
39                     # TODO: Crear objeto Producto y agregarlo a self.
40                     inventario
41                     pass
42                     print(f"--> Datos cargados: {len(self.inventario)} productos")
43         except Exception as e:
44             print(f"Error al cargar: {e}")
45
46     def guardar_datos(self):
47         # TODO: Abrir archivo en modo escritura ('w')
48         # TODO: Recorrer self.inventario y escribir cada producto.
49         a_texto_csv()
50         pass
```

```

50     def agregar_producto(self):
51         print("\n--- Nuevo Producto ---")
52         id_prod = input("ID: ")
53         nombre = input("Nombre: ")
54
55         # Validacion robusta
56         try:
57             precio = float(input("Precio: "))
58             cantidad = int(input("Cantidad: "))
59             if precio < 0 or cantidad < 0:
60                 print("Error: No se permiten negativos.")
61                 return
62
63             # TODO: Crear el objeto y agregarlo a la lista
64             print("Producto agregado.")
65         except ValueError:
66             print("Error: Debes ingresar numeros validos.")
67
68     def realizar_venta(self):
69         id_buscar = input("ID del producto a vender: ")
70         # TODO: Buscar el producto en la lista por su ID
71         # TODO: Si existe, pedir cantidad y llamar al metodo .vender()
72         pass
73
74     def mostrar_inventario(self):
75         print("\n--- Inventario ---")
76         for prod in self.inventario:
77             print(prod)
78
79 # --- 3. MAIN ---
80 def main():
81     tienda = Tienda("base_datos.txt")
82
83     while True:
84         print("\n1. Agregar | 2. Vender | 3. Ver | 4. Salir")
85         opc = input("Opcion: ")
86
87         if opc == "1":
88             tienda.agregar_producto()
89         elif opc == "2":
90             tienda.realizar_venta()
91         elif opc == "3":
92             tienda.mostrar_inventario()
93         elif opc == "4":
94             tienda.guardar_datos()
95             print("Datos guardados. Adios.")
96             break
97         else:
98             print("Opcion invalida.")
99
100 if __name__ == "__main__":
101     main()

```