



PRACTICO – PROYECTO FINAL

“TARANTINO”

INTEGRANTES

Alexander Yanarico Quispe

José Andres Salame Lijeron

Jeremy Said Pardo Camacho

Erwin David Pardo Hernandez

Pablo Jesús Fernández Jaime

Cristhian David Puña Salto

Hugo Clementelli Castedo

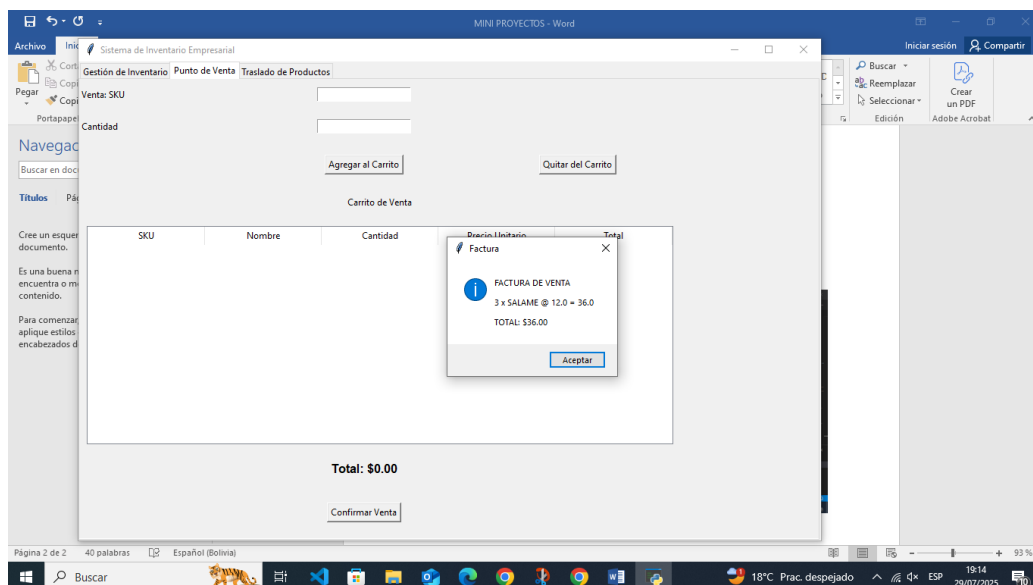
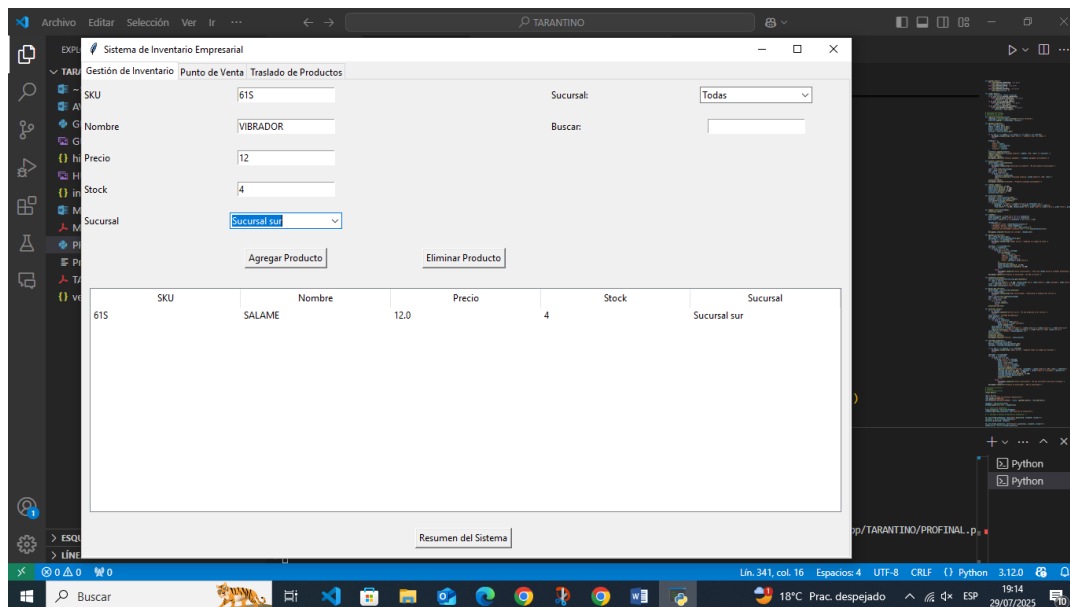
DOCENTE

Ing. Jimmy Nataniel Requena Llorentty

ASIGNATURA

Programación II

```
Archivo  Editar  Selección  Ver  Ir  ...  TARANTINO
GUARDATXT.py x
1 # sistema_inventario.py
2 import tkinter as tk
3 from tkinter import ttk, messagebox
4 import json
5 import os
6 from datetime import datetime
7
8 # Archivos para persistencia
9 ARCHIVO_INVENTARIO = 'inventario.json'
10 ARCHIVO_VENTAS = 'ventas.json'
11 ARCHIVO_HISTORIAL = 'historial.json'
12
13 inventario = []
14 ventas = []
15 historial = []
16 sucursales = ["Todas", "Sucursal norte", "Sucursal sur", "Sucursal urubo", "Sucursal villa", "Sucursal centro"]
17 carrito = []
18 traslados = []
19
20 # =====
21 # FUNCIONES DE ARCHIVOS
22 # =====
23 def guardar_datos():
24     with open(ARCHIVO_INVENTARIO, 'w') as f:
25         json.dump(inventario, f)
26     with open(ARCHIVO_VENTAS, 'w') as f:
27         json.dump(ventas, f)
28     with open(ARCHIVO_HISTORIAL, 'w') as f:
29         json.dump(historial, f)
30
31 def cargar_datos():
32     global inventario, ventas, historial
33     with open(ARCHIVO_INVENTARIO, 'r') as f:
34         inventario = json.load(f)
35     with open(ARCHIVO_VENTAS, 'r') as f:
36         ventas = json.load(f)
37     with open(ARCHIVO_HISTORIAL, 'r') as f:
38         historial = json.load(f)
39
40 # =====
41 # FUNCIONES DE GUI
42 # =====
43 def agregar_producto():
44     sku = entry_sku.get()
45     nombre = entry_nombre.get()
46     precio = entry_precio.get()
47     stock = entry_stock.get()
48     sucursal = entry_sucursal.get()
49
50     if sku and nombre and precio and stock and sucursal:
51         producto = {
52             'SKU': sku,
53             'Nombre': nombre,
54             'Precio': precio,
55             'Stock': stock,
56             'Sucursal': sucursal
57         }
58         inventario.append(producto)
59         guardar_datos()
60         messagebox.showinfo("Éxito", "Producto agregado correctamente.")
61     else:
62         messagebox.showwarning("Error", "Todos los campos son obligatorios.")
63
64 def eliminar_producto():
65     sku = entry_sku.get()
66     if sku:
67         producto = next((p for p in inventario if p['SKU'] == sku), None)
68         if producto:
69             inventario.remove(producto)
70             guardar_datos()
71             messagebox.showinfo("Éxito", "Producto eliminado correctamente.")
72         else:
73             messagebox.showwarning("Error", "Producto no encontrado.")
74     else:
75         messagebox.showwarning("Error", "Se requiere un SKU para eliminar.")
76
77 def buscar_producto():
78     buscar = entry_buscar.get()
79     if buscar:
80         productos = [p for p in inventario if buscar in p['Nombre']]
81         if productos:
82             mostrar_tabla(productos)
83         else:
84             messagebox.showinfo("Info", "No se encontraron productos con ese nombre.")
85     else:
86         messagebox.showwarning("Error", "Se requiere un nombre para buscar.")
87
88 def mostrar_tabla(productos):
89     table.delete(*table.get_children())
90     for producto in productos:
91         table.insert('', 'end', (producto['SKU'], producto['Nombre'], producto['Precio'], producto['Stock'], producto['Sucursal']))
92
93 def agregar_al_carrito():
94     sku = entry_sku.get()
95     cantidad = entry_cantidad.get()
96     if sku and cantidad:
97         producto = next((p for p in inventario if p['SKU'] == sku), None)
98         if producto and int(cantidad) > 0:
99             carrito.append((producto, int(cantidad)))
100             guardar_datos()
101             messagebox.showinfo("Éxito", "Producto agregado al carrito.")
102         else:
103             messagebox.showwarning("Error", "Producto no encontrado o cantidad inválida.")
104     else:
105         messagebox.showwarning("Error", "Se requieren SKU y cantidad.")
106
107 def quitar_del_carrito():
108     sku = entry_sku.get()
109     if sku:
110         producto = next((p for p in carrito if p[0]['SKU'] == sku), None)
111         if producto:
112             carrito.remove(producto)
113             guardar_datos()
114             messagebox.showinfo("Éxito", "Producto quitado del carrito.")
115         else:
116             messagebox.showwarning("Error", "Producto no encontrado en el carrito.")
117     else:
118         messagebox.showwarning("Error", "Se requiere un SKU para quitar.")
119
120 def confirmar_venta():
121     if carrito:
122         generar_factura()
123     else:
124         messagebox.showwarning("Error", "El carrito está vacío.")
125
126 def generar_factura():
127     factura = {}
128     for producto, cantidad in carrito:
129         factura[producto['SKU']] = cantidad
130     total = sum(p['Precio'] * c for p, c in carrito)
131     factura['TOTAL'] = total
132     messagebox.showinfo("Factura de Venta", f"Factura de venta generada. Total: {total}")
133     carrito = []
134     guardar_datos()
135
136 def __init__(self):
137     self.root = tk.Tk()
138     self.root.title("Sistema de Inventario Empresarial")
139     self.root.geometry("1000x700")
140
141     # Menú de navegación
142     menu_bar = tk.Menu(self.root)
143     menu_bar.add_command(label="Inicio", command=self.mostrar_inicio)
144     menu_bar.add_command(label="Gestión de Inventario", command=self.mostrar_gestion_inventario)
145     menu_bar.add_command(label="Punto de Venta", command=self.mostrar_punto_venta)
146     menu_bar.add_command(label="Traslado de Productos", command=self.mostrar_traslado)
147     menu_bar.add_command(label="Resumen del Sistema", command=self.mostrar_resumen)
148     self.root.config(menu=menu_bar)
149
150     # Panel de gestión de inventario
151     self.mostrar_gestion_inventario()
152
153     # Cargar datos al iniciar
154     cargar_datos()
155
156     # Ejecutar la aplicación
157     self.root.mainloop()
158
159 if __name__ == '__main__':
160     app = SistemaInventario()
161     app.run()
```



DESCRIPCIÓN DEL CODIGO

Este sistema de inventario empresarial es una aplicación de escritorio construida con `Tkinter` para la gestión de productos en diferentes sucursales. Permite:

- Agregar, eliminar y listar productos
- Realizar ventas con generación de factura
- Trasladar productos entre sucursales
- Filtrar y buscar productos
- Guardar y cargar datos persistentes (JSON)
- Ver resúmenes e historial de acciones.

LINK DEL CODIGO

<https://github.com/joseandres2003/TARANTINO/blob/main/PROYECTOS/inventario.py>