PROYECTO SISTEMA APLICANDO CONCEPTOS DE PROGRAMACIÓN ESTRUCTURADA

Mateo Jeremy Calvopiña Huerlo

Ing. Tecnología de la información y comunicación

2do ciclo.

Programación Estructurada y Funcional

Nombre del proyecto: **TechMedium**

Proyecto dirigido para:

Este programa está dirigido a personas y pequeños negocios que necesitan un control básico y práctico de productos sin usar sistemas complejos. Es ideal para tiendas pequeñas, técnicos independientes, talleres, ferreterías o cualquier negocio que maneje inventario y ventas simples. También es adecuado para estudiantes o proyectos educativos, ya que permite aprender de forma clara cómo funcionan una base de datos, un sistema de ventas y una interfaz gráfica. El usuario final no necesita conocimientos técnicos ni de programación, solo interactúa con botones sencillos para vender productos, consultar el stock y agregar o modificar artículos. El programa funciona de manera local, es fácil de usar y está pensado para el control diario de inventario de forma rápida y confiable.

Problema a solucionar:

El problema a solucionar es la falta de un control simple, claro y confiable del inventario en pequeños negocios o proyectos básicos. Muchas personas llevan el stock en cuadernos, notas sueltas o en la memoria, lo que provoca errores frecuentes como vender productos que ya no existen, no saber cuántas unidades quedan disponibles o perder información importante de los artículos. Además, sin un sistema organizado es difícil consultar rápidamente los productos, actualizar el stock después de una venta o mantener los datos guardados de forma segura. Este programa soluciona ese problema ofreciendo una manera sencilla de registrar productos, controlar ventas, consultar el inventario en una tabla clara y mantener toda la información almacenada en una base de datos local sin complicaciones técnicas para el usuario.

CODIGO:

- BLOQUE 1– IMPORTAR

```
import tkinter as tk
from tkinter import messagebox, simpledialog, ttk
import sqlite3
```

Para que sirve cada importación:

tkinter → crear ventanas, botones, tablas.

messagebox → mostrar mensajes (error, aviso, OK).

simpledialog → pedir datos al usuario (código, cantidad, etc.).

sqlite3 → guardar datos en la base de datos.

ttk → usar la tabla (Treeview).

- BLOQUE 2 – CONEXIÓN A LA BASE DE DATOS

```
def conectar_db():
    return sqlite3.connect("productos.db")
```

Cada vez que se consulta, guarda o elimina, se usa esto.

- BLOQUE 3 – CREAR TABLA

```
def crear_tabla():
    conn = conectar_db()
    cursor = conn.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS productos (
            codigo TEXT PRIMARY KEY,
            nombre TEXT NOT NULL,
            stock INTEGER NOT NULL
        )
    """)
    conn.commit()
    conn.close()
```

El “crear_tabla”, crea una tabla tipo Excel que va a permite guarda muchos productos, donde van a estar ordenados los códigos, nombres y el stock de cada producto por filas y columnas. Y se ejecuta una sola vez al iniciar el programa.

La parte de:

```
CREATE TABLE IF NOT EXISTS productos (
    codigo TEXT PRIMARY KEY,
    nombre TEXT NOT NULL,
    stock INTEGER NOT NULL
)
```

Aquí se crea la tabla llamada “productos” si no llega a existir. Es una protección.

- BLOQUE 4 – OBTENER PRODUCTO

```
def obtener_producto(codigo):
    conn = conectar_db()
    cursor = conn.cursor()
    cursor.execute(
        "SELECT nombre, stock FROM productos WHERE codigo = ?",
        (codigo,)
    )
    producto = cursor.fetchone()
    conn.close()
    return producto

```

Lee los datos de la base de datos. Busca un producto en el archivo **productos.db** (la base de datos)

- BLOQUE 5 – OBTENER TODOS LOS PRODUCTOS

```
def obtener_todos_los_productos():
    conn = conectar_db()
    cursor = conn.cursor()
    cursor.execute("SELECT codigo, nombre, stock FROM productos")
    productos = cursor.fetchall()
    conn.close()
    return productos
```

Devuelve todos los productos de la opción 2

- BLOQUE 6 – GUARDAR / ACTUALIZAR PRODUCTO

```
def guardar_producto(codigo, nombre, stock):
    conn = conectar_db()
    cursor = conn.cursor()
    cursor.execute("""
        INSERT OR REPLACE INTO productos (codigo, nombre, stock)
        VALUES (?, ?, ?)
    """, (codigo, nombre, stock))
    conn.commit()
    conn.close()
```

Permite guardar un producto nuevo o actualiza uno existente (mismo código).

Se usa cuando, se vende, se agrega, se modifica

- BLOQUE 7 – MOSTRAR TABLA DE PRODUCTOS

```
def mostrar_tabla_productos():
    productos = obtener_todos_los_productos()

    ventana = tk.Toplevel(root)
    ventana.title("Consulta de productos")

    tabla = ttk.Treeview(
        ventana,
        columns=("codigo", "nombre", "stock"),
        show="headings"
    )

    tabla.heading("codigo", text="Código")
    tabla.heading("nombre", text="Nombre")
```

```
    tabla.heading("stock", text="Stock")

    tabla.column("codigo", width=120)
    tabla.column("nombre", width=200)
    tabla.column("stock", width=80)

    for codigo, nombre, stock in productos:
        tabla.insert("", tk.END, values=(codigo, nombre, stock))

    tabla.pack(expand=True, fill="both")

```

Esto tambien pertenece a la opcion 2, específicamente la parte de como muestra la tabla mostrando el código, nombre y el stock de los productos que se hayan ingresado. También de como cada carácter va en una celda.

- BLOQUE 8 – ELIMINAR PRODUCTO

```
def eliminar_producto(codigo):
    conn = conectar_db()
    cursor = conn.cursor()
    cursor.execute("DELETE FROM productos WHERE codigo = ?", (codigo,))
    conn.commit()
    conn.close()

```

Listo para ejecutarlo, borra un producto por código.

- BLOQUE 9 – INICIO DEL PROGRAMA

```
crear_tabla()
print ("Bienvenido al programa PARA NIÑOS INTELIGENTES")
```

Asegura que la tabla exista y garantiza que la base de datos esté lista.

- BLOQUE 10 – VENTANA PRINCIPAL

```
root = tk.Tk()
root.title("Bienvenido al programa")
root.geometry("420x300")

```

Esta es la interfaz principal del programa

- BLOQUE 11 – VENDER PRODUCTO

```
def vender_producto():
…
```

Pide código, verifica si existe, revisa stock, pide cantidad, resta stock y guarda cambios, además, evita vender más de lo disponible.

- BLOQUE 12 – CONSULTAR PRODUCTOS

```
def consultar_producto():
    productos = obtener_todos_los_productos()

    ventana = tk.Toplevel(root)
    ventana.title("Consulta de productos")
    ventana.geometry("450x300")
```

Muestra todos los productos en tabla

- BLOQUE 13 – AGREGAR / MODIFICAR PRODUCTO

```
def agregar_modificar():
…
```

1. Pide código
2. Si existe → permite modificar
3. Si no existe → permite agregar
4. Valida que el stock sea válido

- BLOQUE 14 – MENÚ CON BOTONES

```
tk.Label(
    root,
    text="TECHMEDIUM\n----MENÚ----\n",
    font=("Arial", 12)
).pack(pady=10)

tk.Button(root, text="1. Vender producto", width=30,
command=vender_producto).pack(pady=5)
tk.Button(root, text="2. Consultar productos", width=30,
command=consultar_producto).pack(pady=5)
tk.Button(root, text="3. Agregar o modificar código/producto", width=30,
command=agregar_modificar).pack(pady=5)
tk.Button(root, text="4. Salir del programa", width=30,
command=root.quit).pack(pady=10)

root.mainloop()

```

- Botón 1 → vender
- Botón 2 → consultar (tabla)
- Botón 3 → agregar/modificar
- Botón 4 → salir

Es el menú principal del programa.