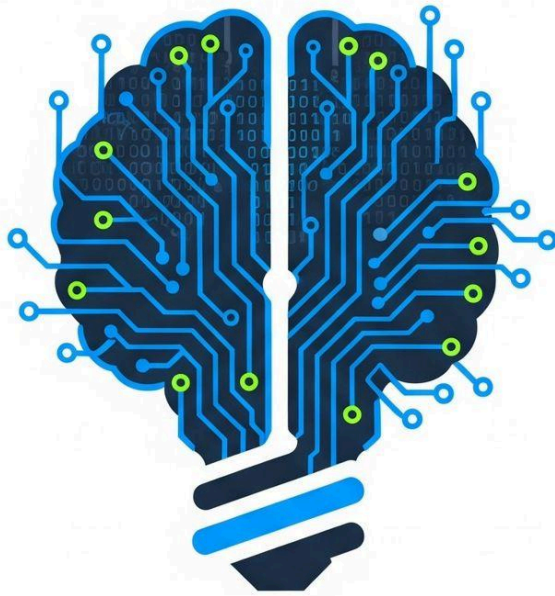


Sistema de inventario para un supermercado

“Los Programistas”



LOS PROGRAMISTAS

Integrantes:

Mindy Mariana Acosta Herrera / Senior

Emilio Asael Robles Sanchez / Mid

Cristina Pérez Torres / Mid

Víctor Gonzáles Andazola / Junior

Marvin Jesus Chavez González / Junior

ÍNDICE

Contexto y problema.....	3
Reglas del negocio.....	4
Requerimientos funcionales.....	5
Gestión de productos.....	5
I/O.....	6
Diagrama de flujo:.....	7
Plan de trabajo:.....	8
Asignaciones de roles:.....	9
Esqueleto del código (pseudocódigo):.....	10
Evidencias de funcionamiento.....	13
Manual de usuario.....	20
Conclusiones.....	22

Contexto y problema

En la tienda “Doña Chula” tienen un problema, todo el inventario lo llevan a mano. Cuales productos entran, cuántos son y hasta el precio de cada uno de ellos son llevados de manera escrita, lo cual, lleva demasiado tiempo dejando un mayor margen de error a la hora de registrar todos estos datos.

El equipo de “Los programistas” harán un código para poder realizar un sistema de inventario y registro de cada uno de los productos, para poder así, mejorar la eficiencia al momento de hacer inventarios.

El código funcionará de la siguiente manera: El código recibirá el nombre del producto, la cantidad de producto que hay y su precio de venta individual, luego al final se hará un resumen en orden de los datos ingresados para poder así mejorar la eficiencia del registro de productos y precios, agregando también el presupuesto inicial del día, total de ingresos del día y el presupuesto total después de haber sumado el ingreso del día, para dejar en claro cuanto es lo que se va generando de presupuesto para, próximamente, re abastecer la tienda.

Reglas del negocio

- 1.El acceso al sistema está restringido mediante una **contraseña de seguridad** (“OXO”). El usuario tiene un máximo de **tres intentos**; si los supera, el programa se cierra automáticamente para proteger la información.
- 2.Una vez dentro del sistema, se muestra un **menú principal** con ocho opciones: mostrar inventario, agregar producto, actualizar producto, eliminar producto, buscar producto, calcular presupuesto total, registrar ventas o salir del programa.
- 3.Cada operación realizada debe ser **validada**. Por ejemplo, al agregar un producto se revisa que el nombre o el código no estén duplicados; al vender, se valida que la cantidad solicitada no supere el stock disponible.
- 4.Si el **presupuesto resulta insuficiente** para cubrir las operaciones o los productos, el sistema despliega un mensaje de advertencia para que el usuario pueda tomar decisiones informadas.
- 5.Al finalizar cada proceso, el sistema genera un **resumen automático** que incluye inventario actualizado, ganancias acumuladas o detalle de las ventas realizada

Requerimientos funcionales

Gestión de productos

1. **Agregar productos:** El sistema debe registrar el nombre, código, stock y precio unitario. Además, valida que no existan duplicados en nombre o código.
2. **Mostrar inventario:** Se debe listar todo el inventario con formato tabular, mostrando de forma clara código, nombre, stock y precio unitario.
3. **Actualizar producto:** Permite modificar el stock existente de un producto. Se considera exitoso si el cambio queda guardado en el archivo.
4. **Eliminar producto:** El usuario puede quitar un producto del inventario, siempre que exista previamente.
5. **Buscar producto:** Debe localizar productos por nombre o código y mostrar la información correspondiente en menos de 2 segundos.

Gestión de inventario y ventas

6. **Presupuesto total:** Calcula el valor del inventario ($\text{stock} \times \text{precio unitario}$) y muestra además las ganancias acumuladas de ventas anteriores.
7. **Registrar venta:** El sistema permite realizar ventas, restar las cantidades vendidas del stock, calcular subtotal, total y actualizar las ganancias. También guarda un historial en un archivo externo.

Acceso y seguridad

8. **Contraseña de acceso:** El sistema requiere ingresar la contraseña “OXO” para habilitar el menú principal, con máximo 3 intentos permitidos.

Persistencia de datos

9. **Inventario persistente:** Todos los productos se guardan en un archivo `.txt` para asegurar que los datos permanezcan entre ejecuciones del programa.
10. **Historial de ventas:** Cada venta realizada queda registrada con detalle (producto, cantidad y subtotal), permitiendo auditar las operaciones en cualquier momento.

I/O

Entradas esperadas:

- **Nombre del producto:** string, sin espacios vacíos y máximo 30 caracteres.
- **Código del producto:** string único e irrepetible.
- **Stock:** número entero mayor a cero.
- **Precio unitario:** número decimal positivo.
- **Opciones de menú:** número entero del 1 al 8.
- **Cantidad de venta:** número entero mayor a cero y menor o igual al stock disponible.

Validaciones implementadas:

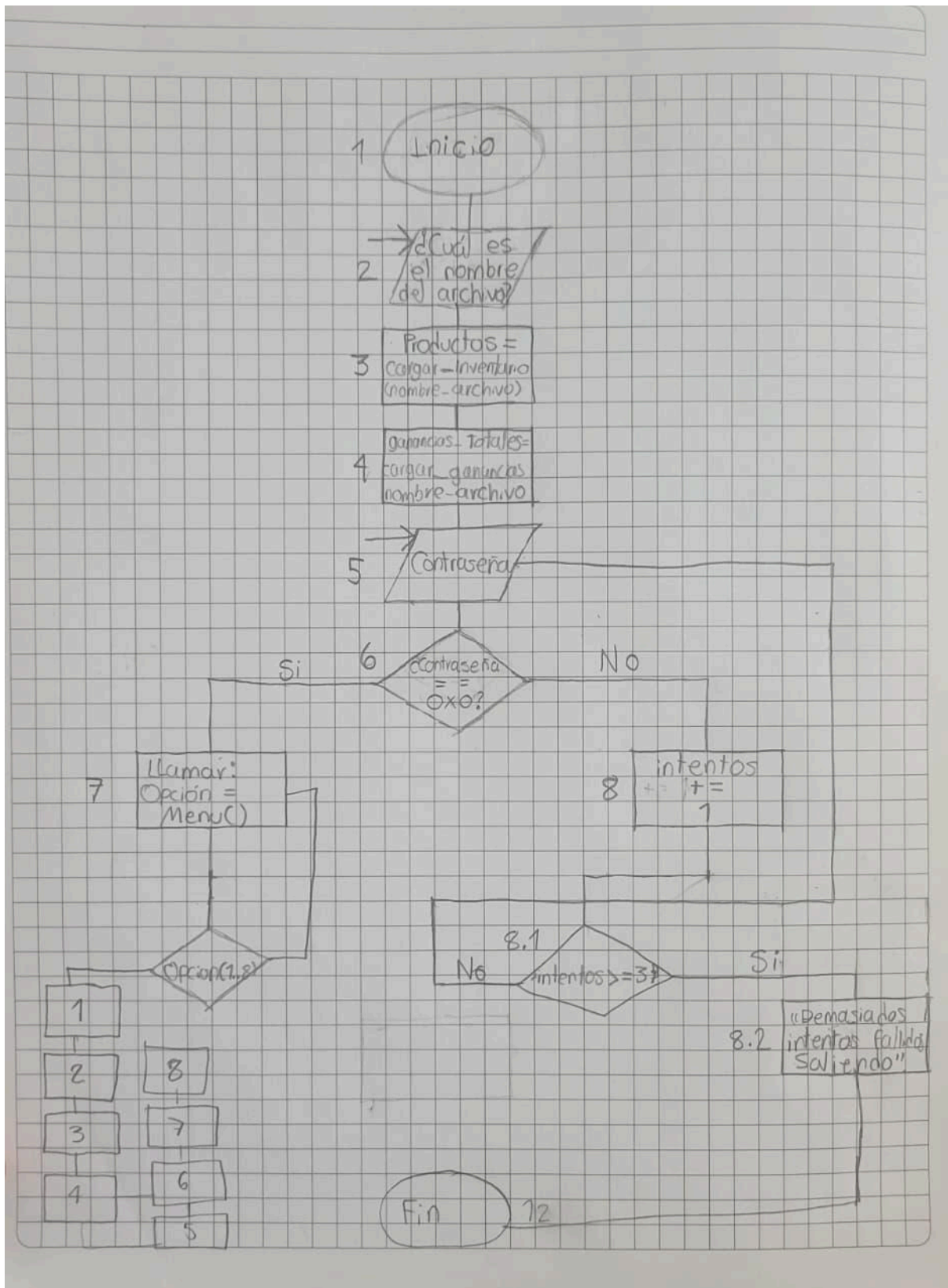
- No permitir productos con nombre o código duplicado.
- No aceptar valores negativos ni caracteres no numéricos en stock o precio.
- Al vender, no se puede registrar una cantidad mayor a la disponible en el inventario.
- Se requiere contraseña correcta para acceder, de lo contrario se bloquea el sistema tras 3 intentos fallidos.

Salidas generadas:

- Mensajes de confirmación o error según la operación realizada.
- Listado completo de inventario en tabla con producto, código, stock y precio.
- Reportes de ventas con cantidades, subtotales y total de la operación.
- Presupuesto general, que incluye valor total del inventario y ganancias acumuladas.

○

Diagrama de flujo:



Plan de trabajo:

Fases	Actividad	Responsable	sem 1	sem 2	sem 3	sem 4	sem 5	sem 6	sem 7
FASE 1	Desarrollar PDF	todos							
	Desarrollar pseudocodigo	Asael							
	Contexto y problema	mindy y marvin							
	Reglas del negocio	cristina, asael y mindy							
	Requerimientos funcionales	marvin							
	I/O	asael							
	Diagrama de flujo	cristina							
	Plan de trabajo	mindy y victor							
	Asignacion de roles	victor							
FASE 2	Actividad	Responsable	sem 1	sem 2	sem 3	sem 4	sem 5	sem 6	sem 7
	Esqueleto(Modificacion)	Cristina							
	PDF fase II (Modificacion)	Mindy, Marvin, Victor							
	PPT Fase II	Mindy y Victor							
	Desarrollo codigo principal	Cristina, Asael y Mindy							
	Busqueda de errores	Mindy, Asael							
	Apoyo en desarrollo de codigo	Marvin y victor							
	Revision final de codigo	Cristina, Asael y Mindy							
	Mejoras a planteamiento del proyecto	Cristina, Asael y Mindy							

Asignaciones de roles:

Roles	Encargado/s	Responsabilidades
Senior	Mindy Mariana Acosta Herrera	Coordinacion del proyecto, programa principal logica principal, validacion final de codigo y documentos.
Mid	Cristina Perez Torres	Programa principal, apoyar a los junior, apoyo con el pdf
Mid	Enilio Asael Robles Sanchez	Programa principal, apoyar a los junior, esqueleto
Junior	Marvin Jesus Chavez Gonzalez	Apoyo pdf, partes del codigo sencillas, pruebas y apoyo en busqueda de errores
Junior	VictorGonzalez Andazola	Apoyo pdf, partes del codigo sencillas, pruebas y apoyo en busqueda de errores

Esqueleto del código (pseudocódigo):

```
#Variables a mostrar
Contraseña=''
intentos=0
Producto=0
Opcion = ''

#---FUNCIONES PARA EL INVENTARIO---

#Mostrar el menu
def Menu():
    print('\n--- MENÚ INVENTARIO ---')
    print('1. Mostrar inventario')
    print('2. Agregar producto')
    print('3. Actualizar producto')
    print('4. Eliminar producto')
    print('5. Buscar producto')
    print('6. Presupuesto total')
    print('7. Venta')
    print('8. Salir')
    try:
        Opcion = int(input("Seleccione una opción: "))
        return Opcion
    except ValueError:
        print("Opción inválida.")
        return None

#Aqui se va el codigo para el inventario
def cargar_inventario(nombre_archivo):
    print("Agregar el codigo para la parte ""cargar_inventario"" )

#Aqui va el codigo para guardar el inventario
def guardar_inventario(nombre_archivo, inventario):
    print("Aqui va el codigo de la parte ""guardar_inventario""")

#Aqui va el codigo para cargar las ganancias
def cargar_ganancias(nombre_archivo):
    print("Aqui va el codigo para la parte ""cargar_ganancias""")

#Aqui va el codigo para guardar las ganancias
def guardar_ganancias(nombre_archivo, ganancias):
    print("Agregar el codigo para la parte ""guardar_ganancias""")
```

```

#Aqui se mostrara el producto y su informacion
def layout(nombre, datos):
    print(f'Producto; {nombre:<15} Código: {datos['codigo']:<10} Stock:
{datos['stock']:<5} Precio: ${datos['precio']:<8.2f}')

#Aqui va el codigo para agregar algun producto nuevo
def AgregarProducto(diccionario):
    print("Agregar el codigo para la parte ""AgregarProducto""")

#Aqui va el codigo para mostrar el inventario
def MostrarInventario(diccionario):
    print("Agregar el codigo para la parte ""MostrarInventario""")

#Aqui va el codigo para actualizar el inventario
def ActualizarProducto(diccionario, dato=None, nuevo_stock=None):
    print("Agregar el codigo para la parte ""ActualizarProducto""")

#Aqui va el codigo para eliminar un producto
def EliminarProducto(diccionario):
    print("Agregar el codigo para la parte ""EliminarProducto""")

#Aqui va el codigo para buscar un producto
def BuscarProducto(diccionario,dato=None):
    print("Agregar el codigo para la parte ""BuscarProducto""")

#Aqui va el codigo para el calculo final de precios
def Preciototal(diccionario):
    print("Agregar el codigo para la parte ""Preciototal""")

#Aqui va el codigo para registrar las ventas
def venta(diccionario):
    print("Agregar el codigo para la parte ""venta""")

#---PROGRAMA PRINCIPAL---
Contraseña=''
intentos=0
Precio_total=0
Productos ={}
Opcion=0
while intentos < 3 and Contraseña != 'OXXO':
    Contraseña=input("Ingrese la contraseña: ")
    if Contraseña=='OXXO':

```

```
while Opcion !=8:
    Opcion=Menu()
    if Opcion == 1:
        MostrarInventario(Productos)
    elif Opcion == 2:
        AgregarProducto(Productos)
    elif Opcion == 3:
        ActualizarProducto(Productos)
    elif Opcion == 4:
        EliminarProducto(Productos)
    elif Opcion == 5:
        BuscarProducto(Productos)
    elif Opcion == 6:
        Precio_total(Productos)
    elif Opcion == 7:
        venta(Productos)
    elif Opcion == 8:
        print('Saliendo...')
        print(Productos)
else:
    print("Contraseña incorrecta. No se pueden agregar o eliminar
productos.")
    intentos += 1
```

Evidencias de funcionamiento

```
sktop/Nuestro-proyecto/src/proyecto.py
¿Qué nombre tiene el archivo?: Inventario
Archivo no encontrado, se creará uno nuevo.
Ganancias acumuladas cargadas: $0.00
Ingrese la contraseña: OXO
```

```
--- MENÚ INVENTARIO ---
```

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

```
Seleccione una opción: 2
```

```
Ingrese el nombre del producto: LECHE
```

```
Ingrese el código: 12345
```

```
Ingrese stock: 12
```

```
Ingrese precio unitario: 34
```

```
Producto agregado.
```

```
--- MENÚ INVENTARIO ---
```

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

```
Seleccione una opción: 2
```

Seleccione una opción: 2
Ingrese el nombre del producto: CHOCOMILK
Ingrese el código: 54321
Ingrese stock: 20
Ingrese precio unitario: 10
Producto agregado.

--- MENÚ INVENTARIO ---

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

Seleccione una opción: 1

--- Inventario ---

Producto:	Código:	Stock:	Precio Unitario:
leche	12345	12	\$34.0
chocomilk	54321	20	\$10.0

--- MENÚ INVENTARIO ---

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

Seleccione una opción: 3

Seleccione una opción: 3
Ingrese producto a actualizar: leche
Ingrese nuevo stock: 14
Producto actualizado.

--- MENÚ INVENTARIO ---

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

Seleccione una opción: 1

--- Inventario ---

Producto:	Código:	Stock:	Precio Unitario:
leche	12345	14	\$34.0
chocomilk	54321	20	\$10.0

--- MENÚ INVENTARIO ---

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

Seleccione una opción: 5

Ingrese producto o código a buscar: 12345
Encontrado:

Ingrese producto o código a buscar: 12345

Encontrado:

Producto: leche	Código: 12345	Stock: 14	Precio: \$34.00
-----------------	---------------	-----------	-----------------

--- MENÚ INVENTARIO ---

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

Seleccione una opción: 6

--- PRESUPUESTO TOTAL ---

Valor del inventario (no es ganancia): \$676.00

Ganancias por ventas realizadas: \$0.00

TOTAL GENERAL (solo ganancias): \$0.00

--- MENÚ INVENTARIO ---

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

Seleccione una opción: 7

Ingrese producto o código a vender (o 'salir' para terminar): 12345

Ingrese cantidad de 'leche': 3

3 leche añadido al carrito. Subtotal: \$102.00

3 leche añadido al carrito. Subtotal: \$102.00

Ingrese producto o código a vender (o 'salir' para terminar): salir

--- RESUMEN DE VENTA ---

leche: 3 unidades → \$102.00

Total de la venta: \$102.00

Ganancias acumuladas actualizadas: \$102.00

--- INVENTARIO ACTUALIZADO ---

--- Inventario ---

Producto:	Código:	Stock:	Precio Unitario:
-----------	---------	--------	------------------

leche	12345	11	\$34.0
-------	-------	----	--------

chocomilk	54321	20	\$10.0
-----------	-------	----	--------

Venta finalizada.

MENÚ INVENTARIO

--- MENÚ INVENTARIO ---

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

Seleccione una opción: 1

--- Inventario ---

Producto:	Código:	Stock:	Precio Unitario:
leche	12345	11	\$34.0
chocomilk	54321	20	\$10.0

--- MENÚ INVENTARIO ---

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

Seleccione una opción: 4

Ingrese producto a eliminar: chocomilk

Producto eliminado

--- MENÚ INVENTARIO ---

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

Seleccione una opción: 1

--- Inventario ---

Producto:	Código:	Stock:	Precio Unitario:
leche	12345	11	\$34.0

--- MENÚ INVENTARIO ---

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

Seleccione una opción: 8

Saliendo...

Manual de usuario

Requisitos necesarios para su uso:

- Contar con Python 3 instalado en la computadora en la que se utilizara.
- Contar con el archivo del programa
- Tener a la mano la contraseña del programa: “OXO”

Como iniciar el programa:

1. Ejecuta el programa en la terminal o tu editor de python
2. Ingresa el nombre del archivo en el que se guardará el inventario (ej.tienda)
 - Esto creará 2 archivos:
 - tienda.txt —> Inventario de productos
 - tienda_ganancias.txt —> Historial de ventas y ganancias
3. El sistema pedirá la contraseña de acceso (OXO)
 - Tienes 3 intentos para ingresarla correctamente.

Menú Principal:

Después de ingresar la contraseña, aparecerá el siguiente menú:

--- MENÚ INVENTARIO ---

1. Mostrar inventario
2. Agregar producto
3. Actualizar producto
4. Eliminar producto
5. Buscar producto
6. Presupuesto total
7. Venta
8. Salir

Opciones del menú

1. Mostrar inventario

Muestra una lista tabular con todos los productos, su código, stock y precio unitario.

2. Agregar producto

Permite registrar un producto nuevo solicitando:

- Nombre (único)
- Código (único)
- Stock (número entero > 0)
- Precio unitario (decimal > 0)

3. Actualizar producto

Modifica el **stock** de un producto existente.

4. Eliminar producto

Elimina un producto del inventario si existe.

5. Buscar producto

Permite localizar un producto por **nombre** o **código**, mostrando sus datos.

6. Presupuesto total

Calcula y muestra:

- Valor total del inventario ($\text{stock} \times \text{precio}$).
- Ganancias acumuladas por ventas.

7. Venta

Permite vender productos solicitando:

- Nombre o código del producto.
- Cantidad a vender
- El sistema descuenta del inventario y actualiza el historial de ventas.

8. Salir

Guarda todos los cambios en los archivos y termina el programa.

Conclusiones

Mindy

Para mí este proyecto fue de gran apoyo para la práctica de temas vistos durante clase. Gracias al uso de funciones, estructuras de decisión y repetición, uso de archivos, etc, pudimos crear y trabajar un programa de un inventario de una tienda, enseñándome así que el uso de la programación puede tener un uso más aplicado de lo que podía pensar.

Asael

Como se puede ver con este programa se podrá hacer un inventario de forma sencilla de modo de que se crearán dos archivos de tipo .txt con los que se llevará el registro de los elementos que se encuentran en el inventario y el registro de ventas con un registro de las ventas.

Cristina

Para mí el proyecto permitió implementar un programa en Python que simplifica el control de inventarios y ventas. En base a que se pudo utilizar archivos de texto, funciones, etc. Gracias a esto también fue más fácil el organizar algunos datos y que el programa fuera más eficiente.

Marvin

Con este programa aprendí a crear un sistema de inventario que puede agregar, eliminar, actualizar y buscar productos, así como registrar ventas y ganancias. Me ayudó a practicar el uso de funciones y manejo de archivos, además de entender cómo organizar mejor un programa para que sea más fácil de usar y útil en la vida real.

Victor

Con el desarrollo de este programa pude aplicar lo aprendido en clase en un caso real para resolver una pequeña problemática de una tienda, pudiendo poner así en práctica los temas vistos en clase, lo cual, me apoyo para comprender mejor estos mismos temas al poderlos ver aplicados en un proyecto.