

# INGENIERÍA EN TECNOLOGÍAS DE INFORMACIÓN

# PROGRAMACIÓN ORIENTADA A OBJETOS

# **TEMA:**

SISTEMA DE GESTIÓN DE INVENTARIOS MEJORADO

# **ESTUDIANTE:**

NORIEGA BALDEON EDWIN FABIÁN

## **DOCENTE**:

ING. NOGALES GUERRERO SANTIAGO ISRAEL

## **NIVEL:**

SEGUNDO NIVEL\_PARALELO "A"

2025-2025

**PUYO-ECUADOR** 



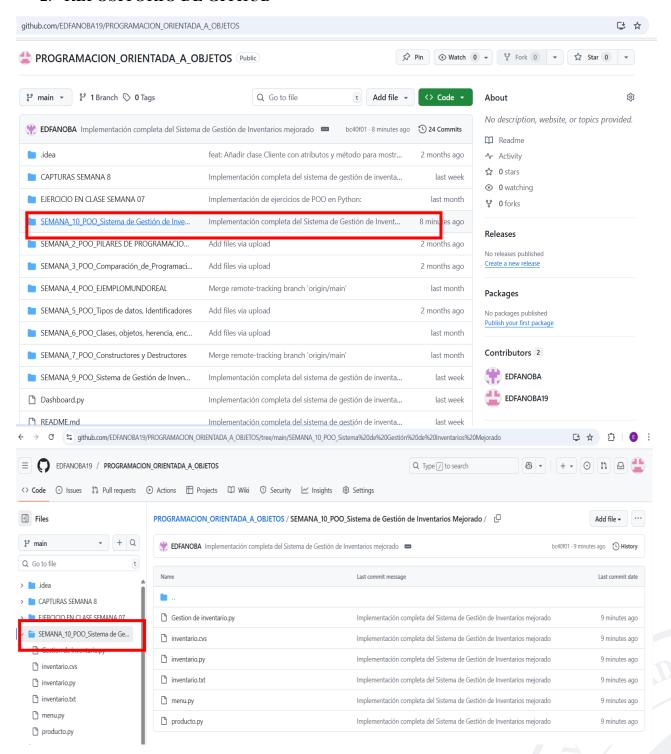
Mr. 2. 1/2 vía Puyo a Tena (Paso Lateral)

**#UEAesExcelencia** 



# https://github.com/EDFANOBA19/PROGRAMACION\_ORIENTADA\_A\_OBJETOS.git

#### 2. REPOSITORIO DE GITHUB



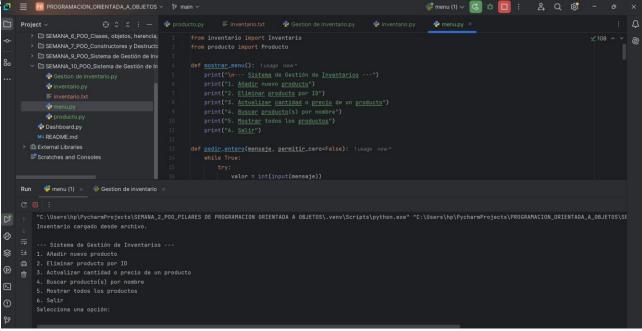
2.1 CÓDIGO



🔇 Km. 2. 1/2 vía Puyo a Tena (Paso Lateral)

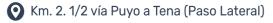
**#UEAesExcelencia** 





#### 2.1.1 Almacenamiento de Inventarios en Archivos:

Modificar la clase *Inventario* para que, al añadir, actualizar, o eliminar productos, estas modificaciones se reflejen en un archivo de texto (por ejemplo, *inventario.txt*).





# **UNIVERSIDAD**

FSTATAL AMAZÓNICA

```
producto = self.buscar_producto_por_id(id_producto)
   if producto:
       self.productos.remove(producto)
           print("Producto eliminado y guardado exitosamente.")
           print("Error al guardar cambios en archivo.")
def actualizar_producto(self, id_producto, cantidad=None, precio=None): 1usage new*
   producto = self.buscar_producto_por_id(id_producto)
   if producto:
           producto.set_cantidad(cantidad)
       if precio is not None:
       if self.guardar_en_archivo():
           print("Producto actualizado y guardado exitosamente.")
def buscar_producto_por_id(self, id_producto): 5 usages new*
       if p.get_id() == id_producto:
     return None
def buscar_productos_por_nombre(self, nombre): 1usage new*
     nombre = nombre.lower()
     return [p for p in self.productos if nombre in p.get_nombre().lower()]
 def mostrar_productos(self): 1 usage new *
     return self.productos
 def cargar_desde_archivo(self): 1usage new*
     self.productos.clear()
         for linea in f:
             linea = linea.strip()
             if not linea:
             partes = linea.split('|')
             if len(partes) != 4:
                 print(f"Linea ignorada por formato incorrecto en archivo: {linea}")
             id_producto = partes[0].strip()
             nombre = partes[1].strip()
                 cantidad = int(partes[2].strip())
                 precio = float(partes[3].strip())
                 print(f"Línea ignorada por datos inválidos en cantidad o precio: {linea}")
             self.productos.append(Producto(id_producto, nombre, cantidad, precio))
```



www.uea.edu.ec



Mr. 2. 1/2 vía Puyo a Tena (Paso Lateral)

**#UEAesExcelencia** 

```
UNIVERSIDAD

ESTATAL AMAZÓNICA

def guardar_en_archivo(self): 3 usages
```

```
def guardar_en_archivo(self): 3 usages new*

try:

with open(self.archivo, 'w', encoding='utf-8') as f:

for p in self.productos:

f.write(f"{p.get_id()} | {p.get_nombre()} | {p.get_cantidad()} | {p.get_precio()}\n")

return True

except PermissionError:

print(f"No tienes permisos para escribir en el archivo '{self.archivo}'.")

return False

except Exception as e:

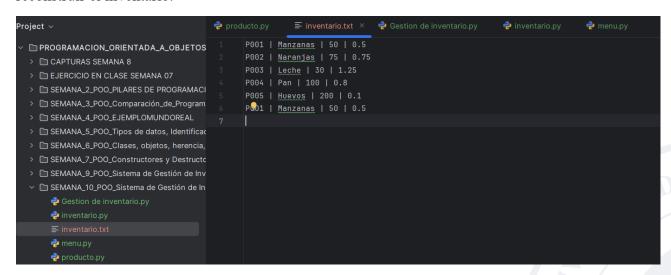
print(f"Error al guardar en archivo: {e}")

return False
```

#### inventario.txt

## 2.1.2. Recuperación de Inventarios desde Archivos:

Al iniciar el programa, cargar automáticamente los productos existentes en *inventario.txt* para reconstruir el inventario.



# 2.1.3. Manejo de Excepciones:







Implementar manejo de excepciones para capturar y tratar adecuadamente posibles errores durante la manipulación de archivos, como FileNotFoundError y PermissionError.

```
    ≡ inventario.txt

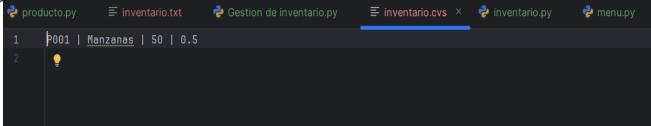
producto.py
                                    🕏 Gestion de inventario.py 🗵
                                                               inventario.py
                                                                                  nenu.py
      class Inventario: 1usage new*
               self.archivo = archivo
               self.productos = []
               self.cargar_desde_archivo()
           def cargar_desde_archivo(self): 1usage new*
                           # Aquí deberías parsear cada línea para crear objetos producto, simplifico
                           self.productos.append(linea.strip())
                  print(f"Archivo '{self.archivo}' cargado exitosamente.")
                   print(f"Sin permiso para leer el archivo '{self.archivo}'.")
           def agregar_producto(self, producto): 1usage new*
               self.productos.append(producto)
                   with open(self.archivo, 'a', encoding='utf-8') as f:
                       f.write(producto + '\n')
                   print(f"Sin permiso para escribir en el archivo '{self.archivo}'.")
```

Asegurar que el programa maneje casos en los que el archivo de inventario no exista, creándolo si es necesario.









## 2.1.4. Modificaciones a la Interfaz de Usuario en la Consola:

Actualizar la interfaz de usuario para notificar al usuario sobre el éxito o fallo de operaciones de archivo (por ejemplo, notificar al usuario cuando un producto se añade exitosamente al archivo de inventario)

```
Run menu (1) × Gestion de inventario ×

Color :

"C:\Users\hp\PycharmProjects\SEMANA_2_P00_PILARES DE PROGRAMACION ORIENT
Archivo 'inventario.txt' cargado exitosamente.
Producto agregado y guardado correctamente.

Process finished with exit code 0
```

```
Run menu (1) × inventario ×

6. Salir
Selecciona una opción: 1

--- Añadir nuevo producto ---
Ingresa ID único: P007
Nombre del producto: CHUPETES
Cantidad: 100
Precio: $0.15
Producto agregado y guardado exitosamente.
```







```
producto.py
                  \equiv inventario.txt 🗡 🛮 禕 Gestion de
            | Manzanas | 50 | 0.5
       P001
       P002 | Naranjas | 75 | 0.75
       P003
            | Leche | 30 | 1.25
       P004 | Pan | 100 | 0.8
       P005 | Huevos | 200 | 0.1
       P001 | Manzanas | 50 | 0.5
       P001 | Manzanas | 50 | 0.5
       P 07 | CHUPETES | 100 | 0.15
 9
       P008 | GALLETAS | 500 | 0.5
       P009
              KCHITOS | 1000 | 1.0
```

```
Selecciona una opción: 2

--- Eliminar producto ---
Ingresa el ID del producto a eliminar: P001
Producto eliminado y guardado exitosamente.
```

```
Run menu (1) × inventario ×

Selecciona una opción: 3

--- Actualizar producto ---

ID del producto a actualizar: P001

Producto actual: ID: P001 | Nombre: Manzanas | Cantidad: 50 | Precio: $0.50

Deja en blanco el campo que no deseas actualizar.

Nueva cantidad: 250

Nuevo precio: $0.25

Producto actualizado y guardado exitosamente.
```







```
🥰 menu (1) 🗴 🏻 🏺 inventario 🗵
Run
    5. Mostrar todos los productos
    6. Salir
    Selecciona una opción: 4
<u>=</u>↓
    --- Buscar productos por nombre ---
    Ingresa parte o todo el nombre: MANZANAS
Se encontraron 2 producto(s):
偷
    ID: P001 | Nombre: Manzanas | Cantidad: 250 | Precio: $0.25
    ID: P001 | Nombre: Manzanas | Cantidad: 50 | Precio: $0.50
       🥰 menu (1) 💉 🛛 📦 inventario 🗵
Run
    Selecciona una opción: 5
    --- Mostrar todos los productos ---
    ID: P002 | Nombre: Naranjas | Cantidad: 75 | Precio: $0.75
<u>=</u>↓
    ID: P003 | Nombre: Leche | Cantidad: 30 | Precio: $1.25
    ID: P004 | Nombre: Pan | Cantidad: 100 | Precio: $0.80
合
    ID: P005 | Nombre: Huevos | Cantidad: 200 | Precio: $0.10
偷
    ID: P001 | Nombre: Manzanas | Cantidad: 250 | Precio: $0.25
    ID: P001 | Nombre: Manzanas | Cantidad: 50 | Precio: $0.50
    ID: P007 | Nombre: CHUPETES | Cantidad: 100 | Precio: $0.15
    ID: P008 | Nombre: GALLETAS | Cantidad: 500 | Precio: $0.50
    ID: P009 | Nombre: KCHITOS | Cantidad: 1000 | Precio: $1.00
                                                             2SIDAD ES
```



