

Universidad de San Carlos de Guatemala
Facultad de ingeniería
Ingeniería en Ciencias y Sistemas
Lenguajes Formales y de Programación



MANUAL TECNICO

Leonel Antonio González García 201709088

Guatemala 22 de agosto de 2023

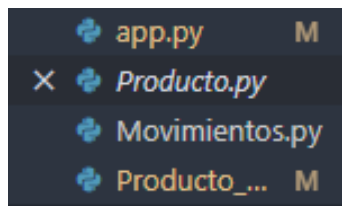
MANUAL TECNICO

Siempre al inicio del programa encontraremos las librerías que nos ayudaran con la creación de las funciones que se necesitan.

La librería tkinter nos sirve para el uso del explorador de archivos.

```
from tkinter import Tk
from tkinter import filedialog
from Producto_Data import Producto_Data
```

Para el desarrollo óptimo de este programa se recurrió al uso de clases.



Clase app.py

En esta clase están los métodos del menú y la carga de archivos, es nuestra clase principal en la que se corre el programa.

```
def CargarInstrucciones():
    root.attributes("-topmost", True)
    #Abre Ventana para Buscar el archivo
    archivo = filedialog.askopenfilename()

    if archivo:
        print("Archivo seleccionado:", archivo)
        print("")
        archivo_texto = open(archivo, "r+", encoding = "utf8")
        array_chars = []
        for line in archivo_texto.readlines():
            for char in line.replace(" ", ";").split(";"):
                array_chars.append(char)
        print(array_chars)

        global Data
        Data = array_chars
        #print(Data)
        print("")
        print("Carga Exitosa")
        print("")

    else:
        print("")
        print("No se seleccionó ningún archivo.")
```

Método Abrir_Txt().

En este método se escribe la información sobre un archivo txt nuevo que sirve para crear los inventarios.

```
def Abrir_Txt():
    with open("Resultado.txt","w", encoding= "utf8") as archivo_txt:
        archivo_txt.write("Informe de Inventario: \n"
                           "Producto      " " Cantidad      "
                           "Precio Unitario      " "Valor Total      " "Ubicación \n")
        for item in Data:
            archivo_txt.write(str(item))
    print("Archivo Creado Correctamente")
```

Clase Producto.

Esta clase sirve para definir el producto y sus atributos en el método constructor.

```
class Producto:

    def __init__(self, instruccion, nombre, cantidad, precio_unitario, ubicacion):
        self.instruccion = instruccion
        self.nombre = nombre
        self.cantidad = cantidad
        self.precio_unitario = precio_unitario
        self.ubicacion = ubicacion
```

Clase Movimientos.

Aquí se crea el método constructor que contiene los atributos necesarios para el objeto.

```
class Movimientos:

    def __init__(self, instruccion, nombre, cantidad, ubicacion):
        self.instruccion = instruccion
        self.nombre = nombre
        self.cantidad = cantidad
        self.ubicacion = ubicacion
```

Clase Producto_Data.

En esta clase se crean los métodos que se utilizarán para el funcionamiento del programa y la manipulación de la información de los archivos.

```
from Producto import Producto

class Producto_Data:

    def __init__(self):
        self.productos = []
        self.contador_productos = 0

    def Nuevo_Producto(self, instruccion, nombre, cantidad, precio_unitario, ubicacion):
        for producto in self.productos:
            if producto.nombre == nombre:
                print("Error, nombre existente")
                return False
        nuevo_producto = Producto(instruccion, nombre, cantidad, precio_unitario, ubicacion)
        self.productos.append(nuevo_producto)
        self.contador_productos += 1
        print(nuevo_producto)
        print("Se agregó un nuevo Producto")
        return True

    def Ver_Productos(self):
        print("")
        print("*****")
        print("")
        for producto in self.productos:
            print("Instrucción ", producto.instruccion, "Nombre:", producto.nombre, "Cantidad:", producto.cantidad,
                  "precio_unitario:", producto.precio_unitario, "Ubicación:" , producto.ubicacion)
        print("*****")
```

Fin del Programa