

## MANUAL TECNICO PRACTICA 1

### Menú principal

Un menú de opciones simple para realizar las funciones requeridas.

```
import carga_archivos

def menu():
    print("- - - - -")
    print("Practica 1 Lenguajes Formales y de Programacion")
    print("- - - - -")
    while True:
        print()
        print("Elige una opcion por favor:")
        print()
        print("1. Cargar inventario inicial")
        print("2. Cargar instrucciones de movimientos")
        print("3. Crear informe de inventario")
        print("4. Salir")
        print()

        opcion = input("Ingresa una opcion: ")
        print()

        if opcion == "1":
            print("Cargar Inventario Inicial")
            carga_archivos.carga_inventario()
            print()
        elif opcion == "2":
            print("Cargar instrucciones de movimientos")
            carga_archivos.carga_movimientos()
            print()
        elif opcion == "3":
            print("Crear informe de inventario")
            carga_archivos.reporte_inventario()
            print()
        elif opcion == "4":
            print("4. Salir")
            break
        else:
            print("Por favor elige una opcion del menu.")

if __name__ == "__main__":
    menu()
```

### Carga de archivos

Para la carga de archivos se utilizó la librería tkinter únicamente para mostrar un cuadro de dialogo que nos permitiera elegir la ubicación de los archivos.

```
import tkinter as tk
from tkinter import filedialog
```

### Carga de inventario inicial:

Se inicializa la ventana de dialogo de la librería mencionada y se valida que se elija un formato de archivo valido '.inv' en este caso, luego se procede a abrir el archivo para leer línea por línea, cada línea se crea una lista comprehensiva en la cual se utiliza el método "strip" por convención de lectura de archivos para eliminar cualquier espacio en blanco no deseado luego se utiliza el método "rstrip" el cual elimina los caracteres de salto de línea con lo cual se procede a utilizar el método "Split" con el que se realiza una separación de la línea por medio del carácter ";" de acuerdo a la estructura del archivo que vamos a leer.

Con los datos divididos de la manera que se desea los guardamos en un diccionario para facilitar el manejo de la información dentro de este diccionario aun necesitamos dividir el primer dato que esta dividido por espacio en blanco entre dos palabras la instrucción y el nombre del producto siguiendo la estructura del archivo.

```
def carga_inventario():
    root = tk.Tk()
    root.withdraw()
    file_path = filedialog.askopenfilename()

    if not file_path.endswith('.inv'):
        print('No es un archivo valido')
    else:
        with open(file_path, 'r') as file:
            for line in file:
                datos = [dato.strip() for dato in line.rstrip().split(';')]
                producto = {
                    'nombre': datos[0].split()[1],
                    'cantidad': int(datos[1]),
                    'precio': datos[2],
                    'bodega': datos[3]
                }
                if datos[0].split()[0] == 'crear_producto':
                    productos.append(producto)
            print('Inventario cargado con exito!!!')
```

Últimamente validamos que el primer dato sea la instrucción “crear\_producto” con lo cual agregamos a la lista de productos que tenemos declarada globalmente para poder utilizar durante toda la ejecución del programa.

### Carga de movimientos:

Esencialmente funciona similar a la carga del inventario, lo que lo diferencia es la validación del tipo de instrucción se realiza si es “agregar\_stock” o “vender\_producto”, se ejecuta la función correspondiente para realizar dicha instrucción enviando los datos necesarios.

```
def carga_movimientos():
    root = tk.Tk()
    root.withdraw()
    file_path = filedialog.askopenfilename()

    if not file_path.endswith('.mov'):
        print('No es un archivo valido')
    else:
        with open(file_path, 'r') as file:
            for line in file:
                datos = [dato.strip() for dato in line.rstrip().split(';')]
                movimiento = {
                    'nombre': datos[0].split()[1],
                    'cantidad': int(datos[1]),
                    'bodega': datos[2]
                }
                if datos[0].split()[0] == 'agregar_stock':
                    agregar_stock(movimiento['nombre'], movimiento['cantidad'], movimiento['bodega'])
                elif datos[0].split()[0] == 'vender_producto':
                    vender_producto(movimiento['nombre'], movimiento['cantidad'], movimiento['bodega'])
            print('Movimientos realizados con exito!!!')
```

- **Agregar Stock:** Validamos que el producto exista en la lista de productos y procedemos a validar que de la misma manera exista en la ubicación.

```
def agregar_stock(nombre, cantidad, bodega):
    for producto in productos:
        if producto['nombre'] == nombre:
            if producto['bodega'] == bodega:
                producto['cantidad'] += cantidad
                return
            else:
                print(f'El producto {producto["nombre"]} no existe en bodega {producto["bodega"]}')
                return
```

- **Vender Producto:** Validamos que el producto exista en la lista de productos y procedemos a validar que de la misma manera exista en la ubicación y que exista suficiente stock para poder vender.

```
def vender_producto(nombre, cantidad, bodega):  
    for producto in productos:  
        if producto['nombre'] == nombre:  
            if producto['bodega'] == bodega:  
                if producto['cantidad'] < cantidad:  
                    print(f'No hay suficiente stock de {nombre} en {bodega}')  
                    return  
                else:  
                    producto['cantidad'] -= cantidad  
                    return
```

## Reporte de Inventario

Para el reporte utilizamos dos librerías que nos sirven como utilidades, “tabulate” para poder darle formato al reporte que generamos como texto y la librería “os” que nos permite abrir el archivo inmediatamente para poder visualizarlo.

```
from tabulate import tabulate  
import os
```

Se realiza una lista de manera comprehensiva por facilidad indicando los datos que va a contener los cuales los obtenemos al iterar en la lista global de productos.

```
def reporte_inventario():  
    tabla = [  
        [producto['nombre'], producto['cantidad'], producto['precio'], producto['cantidad'] * float(producto['precio']), producto['bodega']]  
        for producto in productos  
    ]  
  
    with open('reporte.txt', 'w') as file:  
        file.write(tabulate(tabla, headers=['Producto', 'Cantidad', 'Precio Unitario', 'Valor Total', 'Ubicacion']))  
  
    os.system('start reporte.txt')
```

Luego realizamos la escritura del archivo y le damos un nombre predefinido en el cual finalmente utilizamos “tabulate” indicando los encabezados y la lista que contiene los datos con lo cual procedemos a mostrar el reporte final.