UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS
DEPARTAMENTO DE MATEMÁTICA
SEGUNDO SEMESTRE 2023

LFP B-

PRACTICA #1

Nombre: Javier Ricardo Yllescas Barrios

Carné: 201906795

Fecha: 22/08/2023

Descripcion del Proyecto

Es un programa en Python que permita gestionar un inventario y registrar y gestionar movimientos en el inventario por medio de instrucciones en archivos de texto con extenciones especificas.

Objetivos

• Cargar un archivo con extension .inv para el registro del inventario

• Cargar un archivo con extension .mov para el registro de los movimientos como agregar o vender productos

• Realizar un informe.txt con la informacion del inventario.

Indice

Estructura del Codigo

Librerias (import os)

Variables Globales

(Son validadores, contadores, listas y diccionarios para el funcionamiento de las funciones)

Funciones

```
22 > def mensajebienvenida(): ···
29
30 > def mensajeadvertenciaerrores(): ···
42
43 > def menu(): ···
80
81
82 > def abrirarchivo(extensionvalida): ···
155
156   #////////////////////////////////////////////////
157
158 > def ordenarinventario(): ···
207
208   #////////////////////////////////////////////////
209
210 > def ordenarmovimientos(): ···
292
293   #////////////////////////////////////////////////
294 > def crearinformeinventariomensaje(): ···
345
346   #////////////////////////////////////////////////
347 > def creararchivoinventario(texto): ···
363
364   #////////////////////////////////////////////////
365 > def opcion1(): ···
400
401   #////////////////////////////////////////////////
402 > def opcion2(): ···
431
432   #////////////////////////////////////////////////
433 > def opcion3(): ···
469
470   #////////////////////////////////////////////////
471 > def opcion4(): ···
473   #////////////////////////////////////////////////
```

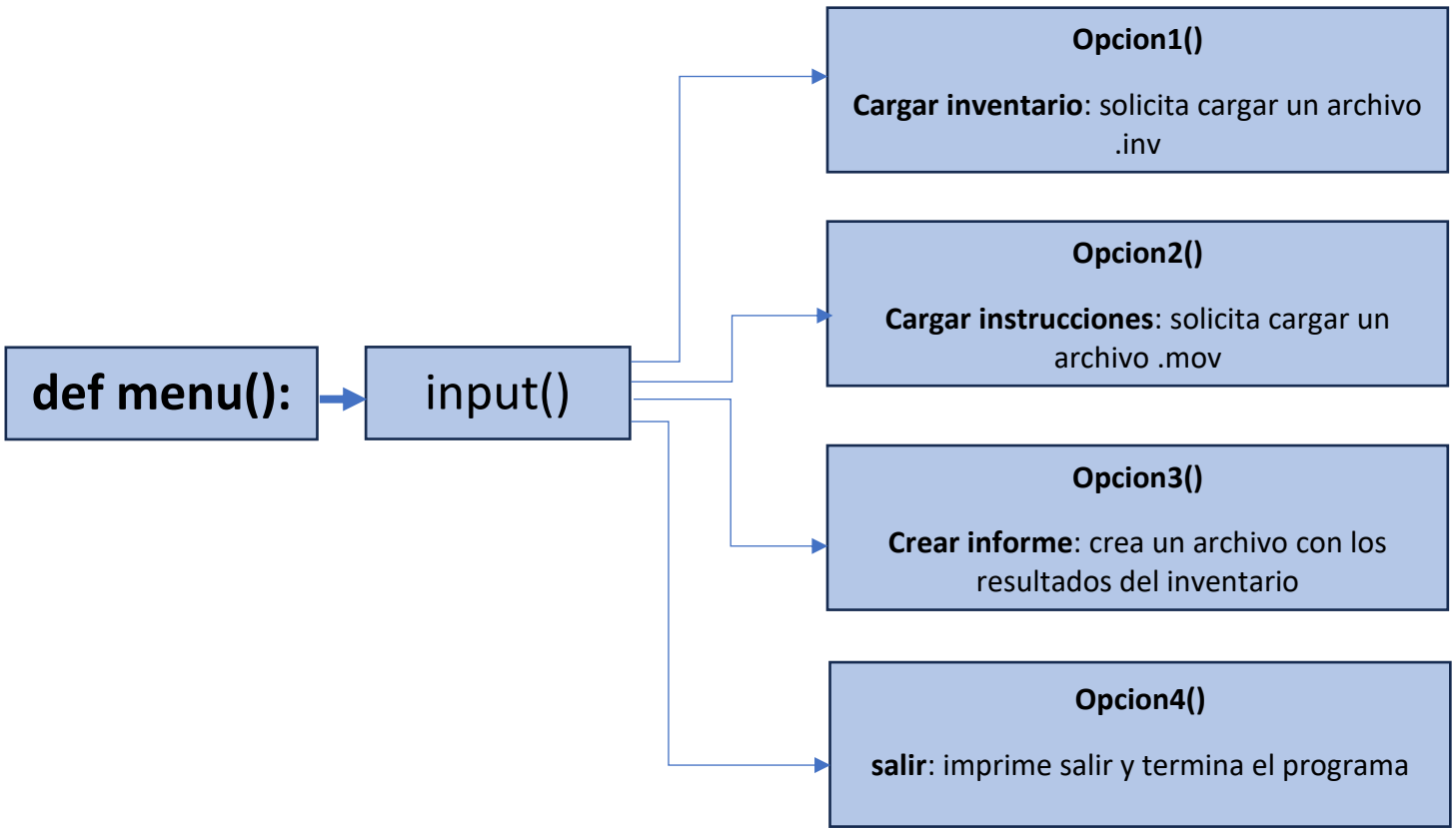
Inicio del programa

(Llama a las funciones)

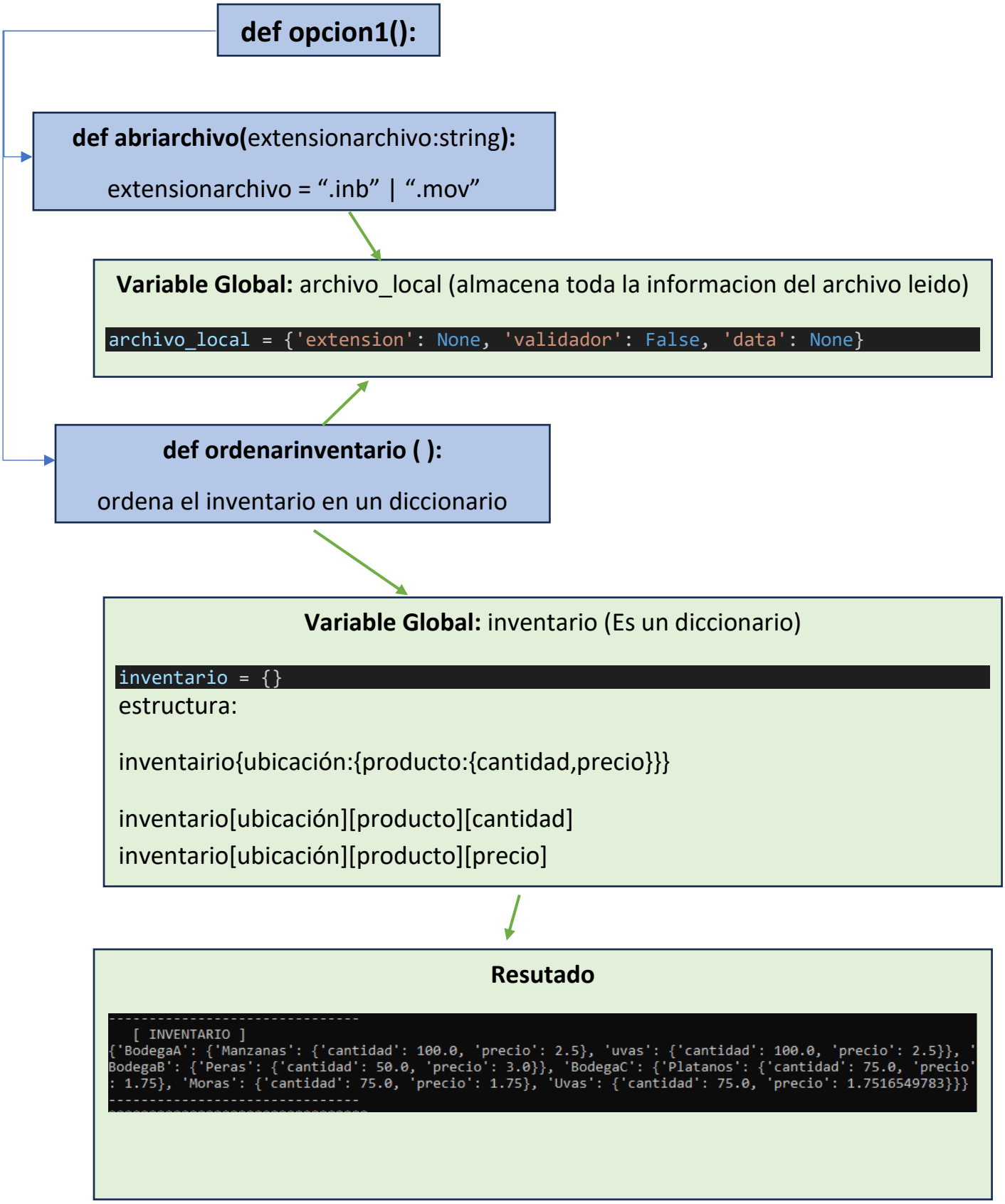
```
############################################################
print(mensajebienvenida())

#Adevertencia Errores
print(mensajeadvertenciaerrores())
opcionerrores = input()
if opcionerrores == 's' or opcionerrores == 'S' or opcionerrores == ' s' or opcionerrores ==' S':
    errores['general'] = True
    print('• Detector de errores activado')

menu()
############################################################
```

Estructura de las Funciones

Estructura de la funcion Menu



Estructura de la funcion opcion1()



Estructura de la funcion opcion2()

def opcion2():

def abriarchivo(extensionarchivo:string**):**

extensionarchivo = “.inb” | “.mov”

Variable Global: archivo_local (almacena toda la informacion del archivo leido)

```
archivo_local = {'extension': None, 'validador': False, 'data': None}
```

def ordenarmovimientos ( ):

ordena y modifica al inventario

Variable Global: inventario (Es un diccionario)

```
inventario = {}
```

Actualiza

Agregegar o vender stock:

```
inventario[str(ubicacion)][str(producto)]['cantidad'] = nuevacantidad
```

Resutado

```
------------------------------------------------
• Se agrego el producto  Manzanas  en la ubicacion  BodegaA  |  100.0  +  50  ->  150.0  ## Linea:  1
• Se agrego el producto  uvas  en la ubicacion  BodegaA  |  100.0  +  50  ->  150.0  ## Linea:  2
  ERROR no se encontro la ubicacion: Bodegah ## Linea: 3
• Venta de  10  Manzanas  de  BodegaA . | Existencias:  150.0  -  10.0  ->  140.0  ## Linea:  4
  Error: NO hay suficientes Manzanas en BodegaA | Existencias:140.0 - 200.0 -> -60.0 ## Linea: 5
  Error: NO existe el producto Arandanos en la ubicacion BodegaB ## Linea: 6
  Error: NO se encontro la ubicacion: BodegaZ ## Linea: 7
• Venta de  50  Manzanas  de  BodegaA . | Existencias:  140.0  -  50.0  ->  90.0  ## Linea:  8
• Venta de  125  uvas  de  BodegaA . | Existencias:  150.0  -  125.0  ->  25.0  ## Linea:  9
• Venta de  25  uvas  de  BodegaA . | Existencias:  25.0  -  25.0  ->  0.0  ## Linea:  10
  Error: NO hay suficientes uvas en BodegaA | Existencias:0.0 - 25.0 -> -25.0 ## Linea: 11
Validador movimientos -> True
============================================================
```

Estructura de la funcion opcion3()

def opcion3():

def crearinformeinventariomensaje ( ):

crea un mensaje con la estructura

Variable local: textoinventario la funcion retorna un texto y lo almacena

```
textoinventario = crearinformeinventariomensaje()
```

creararchivoinventario(textoinventario**)**

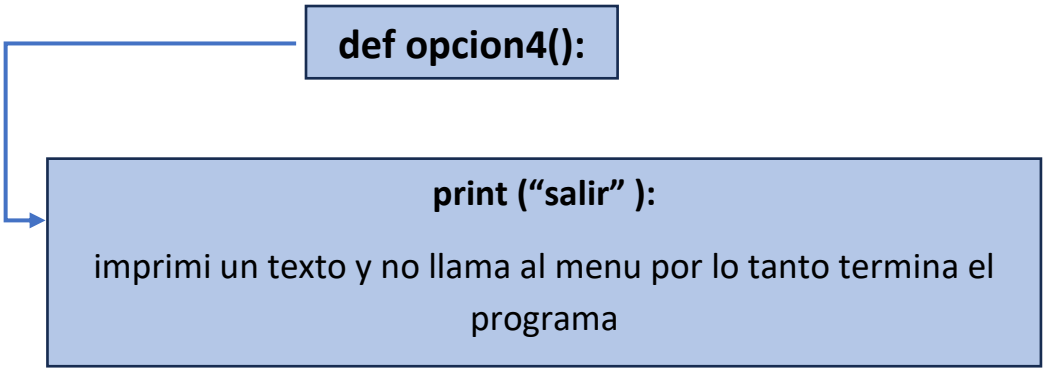
crea un archivo con el textoinventario

Resutado

Un archivo de texto con lo siguiente:

```
================================================================
# Crer Informe de invantario.

********************************************************************************
Informe de Inventario:

 ____________________________________________________________________
|   Producto   | Cantidad | Precio und. | Valor Total |   Ubicacion   |
 --------------------------------------------------------------------
|Manzanas      |      90.0| Q.......2.50| Q......92.50|   BodegaA     |
|uvas          |       0.0| Q.......2.50| Q.......2.50|   BodegaA     |
|Peras         |      50.0| Q.......3.00| Q......53.00|   BodegaB     |
|Platanos      |      75.0| Q.......1.75| Q......76.75|   BodegaC     |
|Moras         |      75.0| Q.......1.75| Q......76.75|   BodegaC     |
|Uvas          |      75.0| Q.......1.75| Q......76.75|   BodegaC     |
 --------------------------------------------------------------------

********************************************************************************
• Creando archivo inventario_201906795.txt
• Escritura del contenido correctamente
************************************************************
* Archivo inventario_201906795.txt creado correctamente *
************************************************************
================================================================
```

Estructura de la funcion opcion4()



Codigo

Variables Globales

```python

import os

#Variables globales
archivo_local = {'extension': None, 'validador': False, 'data': None}
inventario = {}
banderas = {'archivoinventario': False, 'movimientos': False}
errores = {'archivo-inv':[], 'archivo-mov':[],'errores-inv':False, 'errores-mov':False, 'errores-invtxt': True, 'general': False}

contadores = {'automatizacion': 0}
```

Archivo_local:
- Tipo: Diccionario
- Almacena la extencion si se proceso la informacion y la informacion del archivo leido puede ser .inv o .mov

inventario:
- Tipo: Diccionario
- Almacena la informacion del archivo .inv con un estructura:
- estructura

inventairio{ubicación:{producto:{cantidad,precio}}}

inventario[ubicación][producto][cantidad]
inventario[ubicación][producto][precio]

banderas:
- Tipo: Diccionario
- Almacena los validadores booleanes que sirve de indicador si se termino la tarea o ocurrio un error en el proceso por lo tanto no se validadarian (True)

errores:
- Tipo: Diccionario
- Almacena los los validadadores si hubo un error en los procesos
- errores[‘general’] = True habilita la pusa por cada error encontrado

contadores:
- Tipo: Diccionario

inicio

```
1  ######################################################################
2
3  print(mensajebienvenida())
4
5  #----------------------------------------------------------------------
6  #Adevertencia Errores
7
8  print(mensajeadvertenciaerrores())
9  opcionerrores = input()
10 if opcionerrores == 's' or opcionerrores == 'S' or opcionerrores == ' s' or opcionerr
   ores ==' S':
11    errores['general'] = True
12    print('• Detector de errores activado')
13 #----------------------------------------------------------------------
14
15 menu()
16
17
18
19 ######################################################################
```

def menu()

```python
#////////////////////////////////////////////////
def menu():
    mensaje =  '\n\n# Sistema de inventario:\n'

    if (banderas['archivoinventario'] == True):
        mensaje += '## [ Inventario cargado ]\n'
    if (banderas['movimientos'] == True):
        mensaje += '### [ Movimientos cargado ]\n'

    mensaje += '\n'
    mensaje += '1) Cargar inventario inicial\n'
    mensaje += '2) Cargar instrucciones movimientos\n'
    mensaje += '3) Crear informe de inventario\n'
    mensaje += '4) Salir\n'
    mensaje += '\n'
    print('======================================================================')
    print(mensaje)
    #Ingresar opcion
    opcion = input()
    #Validar opcion
    try:
        opcion = int(opcion)
    except:
        print("Ingrese un opcion numerica, porfavor vuelva a intentar.")

    print('======================================================================')
     #SWITCH
    if opcion == 1:
        opcion1()
    elif opcion == 2:
        opcion2()
    elif opcion == 3:
        opcion3()
    elif opcion == 4:
        opcion4()
```

def opcion1()

```python
#////////////////////////////////////////////
def opcion1():
    #Validador
    banderas['archivoinventario'] = False

    #Leer archivo
    print('# Cargar inventario inicial:')
    extensionarchivo = '.inv'
    abrirarchivo(extensionarchivo)

    #Evaluar
    if archivo_local['validador'] == True and archivo_local['extension'] == extensionarchivo:
        print('*******************************')
        print('* Archivo leido correctamente *')
        print('*******************************')
        #print(archivo_local['data'])

        ordenarinventario()

        print('*********************************')
        print('* Lista ordenada correctamente *')
        print('*********************************')

        #Validador
        banderas['archivoinventario'] = True

    else:
        print('Error Inventario NO fueron leidos.')
        input()

    #Menu
    menu()
```

def abrirarchivo()

```python
#////////////////////////////////////////////
def abrirarchivo(extensionvalida):
    archivo_local['validador']= False
    #Mensaje
    print('\nSeleccione un archivo'+ str(extensionvalida)+'\n')
    print('-----------------------------------------------------------------------')
    print('Nota: el archivo debe estar en la misma carpeta que el proyecto.')
    print('Nota 2: si desea salir presione el numero " 4 ".')
    print('-----------------------------------------------------------------------')

    ruta = input("Ingrese la ruta del archivo: ")

    while True:
        #print('ruta', ruta)
        #Salir
        if ruta == '4':
            break
        #Validar extension
        nombre, extension = os.path.splitext(ruta)
        print('extension',extension)
        if extension == str(extensionvalida):
            print("\nArchivo valido\n")
            break
        else:
            print("\nArchivo invalido\n")
            ruta = input("\nIngrese la ruta del archivo: ")


    #Abrir archivo
    listaDatos = []
    try:
        with open(ruta, "r") as archivo:
            for linea in archivo:
                #Array por salto de linea
                array = linea.split('\n')
                for i in range(len(array)):
                    if array[i] == '' or array[i] == ' ':
                        None
                    else:
                        #Array por ; punto y coma
                        newarray = array[i].split(';')
                        listaDatos.append(newarray)


        #Almacenar Archivo
        try:
            archivo_local['extension']= str(extensionvalida)
            archivo_local['validador']= True
            archivo_local['data']= listaDatos
        except:
            print("Error al almacenar variables")


    except:
        print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
        print('! Error: Al abrir el archivo. !')
        print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
        menu()
```

def ordenarinventario()

```python
#//////////////////////////////////////////////
def ordenarinventario():
    print('ordenando...')
    listaDatos = archivo_local['data']
    #++++++++++++++++++++++++++++++++++++++++++
    #Agregar ubicaciones a inventario
    for i in range (0,len(listaDatos)):
        ubicacion = listaDatos[i][3]

        #Evaluar ubicacion
        if ubicacion in inventario:
            #print(str(ubicacion) +' ya se encuentra en el inventario')
            pass
        else:
            #print(' • Agregando nueva ubicacion ', ubicacion)
            inventario[str(ubicacion)] = {}

    #++++++++++++++++++++++++++++++++++++++++++
    #Agregar productos a inventario
    for i in range (0,len(listaDatos)):

        instruccionproducto = listaDatos[i][0]
        instruccion, producto = instruccionproducto.split()
        cantidad = listaDatos[i][1]
        precio = listaDatos[i][2]
        tempubicacion = listaDatos[i][3]

        #Validar instruccion
        if instruccion == 'crear_producto':
            #Validar si existe el producto
            if (producto in inventario[str(tempubicacion)]):
                print('▓▓ Error producto: ', producto, 'NO se agrego a ', tempubicacion, ' porque ya existe el producto', ' ##Linea: ',(i+1))
                if(errores['general'] == True):
                    input()
            else:
                print(' • Agregando produto ', producto, ' a ', tempubicacion)
                inventario[str(tempubicacion)][str(producto)] = {'cantidad':float(cantidad),'precio':float(precio)}
        else:
            print('▓▓ Error no se agrego el producto porque la instruccion es diferente. | '+ str(instruccion) +' != crear_producto '+ ' ##Linea '+str(i+1) )
            if(errores['general'] == True):
                input()


        #++++++++++++++++++++++++++++++++++++++++++
    print('------------------------------')
    print('   [ INVENTARIO ]')
    print(inventario)
    print('------------------------------')

```

def opcion2()

```python
#////////////////////////////////////////////
def opcion2():
    #Validar si hay inventario
    if banderas['archivoinventario'] == True:
        print('# Cargar inventario inicial:')
        extensionarchivo = '.mov'
        abrirarchivo(extensionarchivo)

        #Evaluar
        if archivo_local['validador'] == True and archivo_local['extension'] == extensionarchivo:
            print('*******************************')
            print('* Archivo leido correctamente *')
            print('*******************************')
            #print(archivo_local['data'])

            ordenarmovimientos()

            print('Validador movimientos ->',banderas['movimientos'])

            menu()
        else:
            print('Error: Inventario NO fueron leidos.')
            input()
    else:
      print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
      print('! Carge un inventario antes para continuar !')
      print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
      print('\n presione una tecla para continuar')
      input()
      menu()
```

def ordenarmovimiento ()

```python
#////////////////////////////////////////////
def ordenarmovimientos():
    #Validador
    banderas['movimientos'] = False
    print('ordenar movimientos')
    listaDatos = archivo_local['data']
    print('-----------------------------------------------')
    print(listaDatos)
    print('-----------------------------------------------')

    for i in range (0,len(listaDatos)):
        movimiento, producto = listaDatos[i][0].split()
        cantidad = listaDatos[i][1]
        ubicacion = listaDatos[i][2]

        #print(movimiento, producto,cantidad, ubicacion)

        #++++++++++++++++++++++++++++++++++++++++++
        # [ AGREGAR STOCK ]
        if (movimiento == 'agregar_stock'):
            #Producto Existe
            #Valida existencia ubicacion
            if (ubicacion in inventario):
                #Validar existencia producto
                if (producto in inventario[str(ubicacion)]):
                    #Actualizar inventario
                    antiguacantidad = inventario[str(ubicacion)][str(producto)]['cantidad']
                    nuevacantidad = float(antiguacantidad) + float(cantidad)
                    inventario[str(ubicacion)][str(producto)]['cantidad'] = nuevacantidad
                    print(' • Se agrego el producto ', producto,' en la ubicacion ', ubicacion, ' | ',antiguacantidad, ' + ', cantidad,' -> ',inventario[str(ubicacion)][str(producto)]['cantidad'], ' ## Linea: ', (i+1))
                    #Validador
                    banderas['movimientos'] = True
                else:
                    mensaje = '[] NO existe el producto ', producto,' en la ubicacion ', ubicacion, ' ## Linea: ',(i+1)
                    print(mensaje)
                    if (errores['general'] == True):
                        input()

            else:
                mensaje = '[] ERROR no se encontro la ubicacion: '+ str(ubicacion) + ' ## Linea: ' + str((i+1))
                print(mensaje)
                if (errores['general'] == True):
                    input()

```

```
46
47        elif (movimiento == 'vender_producto'):
48        #++++++++++++++++++++++++++++++++++++++++++++
49        # [ VENDER PRODUCTO ]
50           #Validar existencia ubicacion (Bodega)
51           if (ubicacion in inventario):
52              #Validar existencia producto
53              if (producto in inventario[str(ubicacion)]):
54                 #Validar operacion valida
55                 ##operacioes
56                 cantidadproducto = inventario[str(ubicacion)][str(producto)]['cantida
   d']
57                 venta = float(cantidad)
58                 nuevacantidad = float(cantidadproducto) - venta
59                 #Validar operacion no den numero negativo (sin existencias)
60                 if (nuevacantidad >= 0):
61                    inventario[str(ubicacion)][str(producto)]['cantidad'] = nuevacantid
   ad
62                    print(' • Venta de ',cantidad,' ', producto, ' de ',ubicacion,'. |
   Existencias: ', cantidadproducto,' - ',venta,' -> ',nuevacantidad, ' ## Linea: ' ,str
   ((i+1)) )
63                    #Validador
64                    banderas['movimientos'] = True
65                 else:
66                    mensaje ='▓▓ Error: NO hay suficientes '+str(producto)+' en ' +str
   (ubicacion)+' | Existencias:' +str(cantidadproducto) +' - '+ str(venta)+ ' -> '+str(n
   uevacantidad) + ' ## Linea: ' + str((i+1))
67                    print(mensaje)
68                    if (errores['general'] == True):
69                       input()
70
71
72              else:
73                 mensaje = '▓▓ Error: NO existe el producto ' +  str(producto) + ' en l
   a ubicacion ' + str(ubicacion)  + ' ## Linea: ' + str((i+1))
74                 print(mensaje)
75                 if (errores['general'] == True):
76                    input()
77
78           else:
79              mensaje = '▓▓ Error: NO se encontro la ubicacion: '+ str(ubicacion) + ' #
   # Linea: ' +str(i+1)
80              print(mensaje)
81              if (errores['general'] == True):
82                    input()
```

def opcion3 ()

```python
#////////////////////////////////////////////
def opcion3():
    print('# Crer Informe de invantario.')

    if banderas['archivoinventario'] == True and banderas['movimientos'] == True:

        print()
        print('**********************************************************************
**********')

        textoinventario = crearinformeinventariomensaje()
        print(textoinventario)

        print('**********************************************************************
**********')

        creararchivoinventario(textoinventario)

        #Validador
        if (errores['errores-invtxt'] == False):
            print('*******************************************************')
            print('* Archivo inventario_201906795.txt creado correctamente *')
            print('*******************************************************')


    elif banderas['archivoinventario'] == False:
        print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
        print('! Carge un inventario antes para continuar !')
        print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
        if (errores['general'] == True):
            input()
    elif banderas['movimientos'] == False:
        print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
        print('! Carge un listado de movimientos antes para continuar !')
        print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
        if (errores['general'] == True):
            input()

    menu()
```

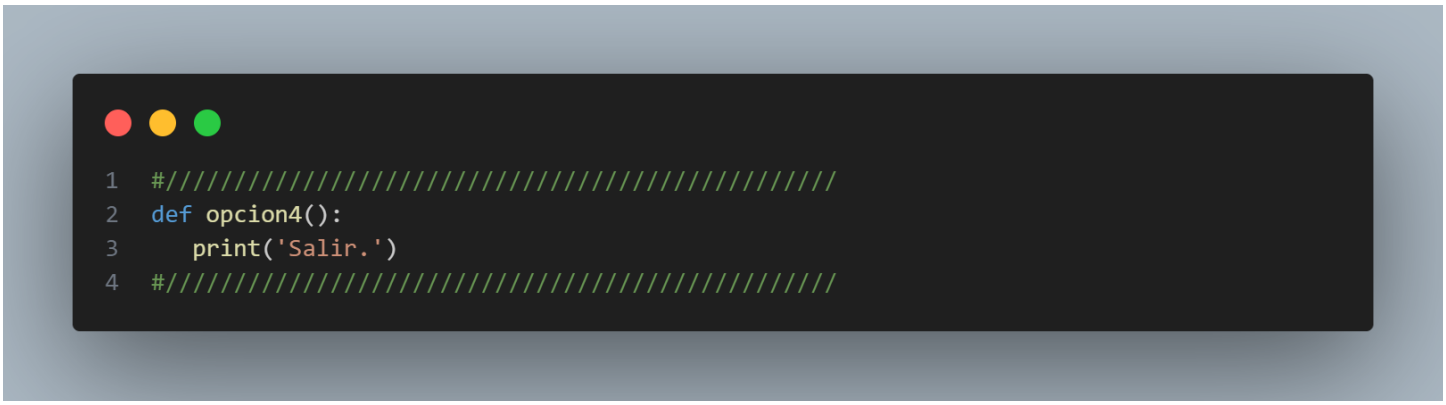
def crearinformeinventariomensaje ()

```python
#////////////////////////////////////////////////
def crearinformeinventariomensaje():
    mensajeinventario =  'Informe de Inventario: \n'
    mensajeinventario += '\n'
    mensajeinventario += ' ____________________________________________________________
________\n'
    mensajeinventario += '|   Producto   | Cantidad | Precio und. | Valor Total |   Ubi
cacion   | \n'
    mensajeinventario += ' ------------------------------------------------------------
--------\n'

    #Inventario
    #Obtener lista de ubicaciones
    listaubicaciones = list(inventario)
    #Obtener lista de productos
    listaproductos = []
    #Recorre ubicaicones
    for i in range (0,len(listaubicaciones)):
        ubicacion = listaubicaciones[i]
        #Recorre productos
        for producto in inventario[str(ubicacion)]:
            #Obtiene precio
            precio = inventario[str(ubicacion)][str(producto)]['precio']
            cantidad = inventario[str(ubicacion)][str(producto)]['cantidad']
            valortotal = float(precio) + float(cantidad)
            #Imprimir valores
            #Configuraciones Imprimir valores
            espaciosproducto = 13
            espacioscantidad = 10
            espaciospreciound = 13
            espaciosvalortotal = 13
            espaciosubicacion = 13
            #Formato de valores
            cantidad = f"{cantidad:.{1}f}"
            precio = f"{precio:.{2}f}"
            valortotal = f"{valortotal:.{2}f}"
            #Estructura con espacios para su tabulacion
            txtproducto = str(producto)[0:(espaciosproducto-1)].ljust(espaciosproducto," ")
            txtcantidad = str(cantidad)[0:(espaciosproducto-1)].rjust(espacioscantidad," ")
            txtpreciound = str(precio)[0:(espaciospreciound-1)].rjust((espaciospreciound-2),",")
            txtvalortotal = str(valortotal)[0:(espaciosvalortotal-1)].rjust((espaciosvalortotal-2),",")
            txtubicacion = str(ubicacion)[0:(espaciosubicacion-1)].center(espaciosubicacion+1)
            #Agregar nueva linea
            mensajeinventario += '|'+txtproducto+'|'+txtcantidad+'| Q'+txtpreciound+'| Q'+txtvalortotal+'|'+txtubicacion+'|\n'


    mensajeinventario += ' ------------------------------------------------------------
--------\n'
    return mensajeinventario
```

def creararchivoinventario ()

```python
#////////////////////////////////////////////////
def creararchivoinventario(texto):
    #Validador
    errores['errores-invtxt']=True
    print('• Creando archivo inventario_201906795.txt')
    try:
        #crear archivo
        archivo = open("inventario_201906795.txt", "w")
        archivo.write(texto)
        archivo.close()
        print('• Escritura del contenido correctamente')
        #Validador
        errores['errores-invtxt']=False
    except:
        print('Error al crear archivo inventario_201906795.txt')
        if(errores['general'] == True):
            input()

```

def opcion4()



```
#////////////////////////////////////////////////
def opcion4():
   print('Salir.')
#////////////////////////////////////////////////
```