



# MANUAL TECNICO

Gestión de inventario

## [ENCABEZADO]

El presente manual describe las funcionalidades, métodos, herramientas y validaciones que se utilizaron para la realización de la aplicación.

## [PRINCIPIO, TÉCNICA O PARADIGMA APLICADO DE PROGRAMACIÓN]

Programación orientada a objetos

- Polimorfismo

- Modularidad

Paradigma Imperativo

- Programación modular

## [CONVENCIONES DE NOMENCLATURA]

[DECLARACION DE VARIABLES, METODOS O FUNCIONES]

`es_variable=""`

`es_variable=0`

`def es_metodo():`

`print("hola mundo")`

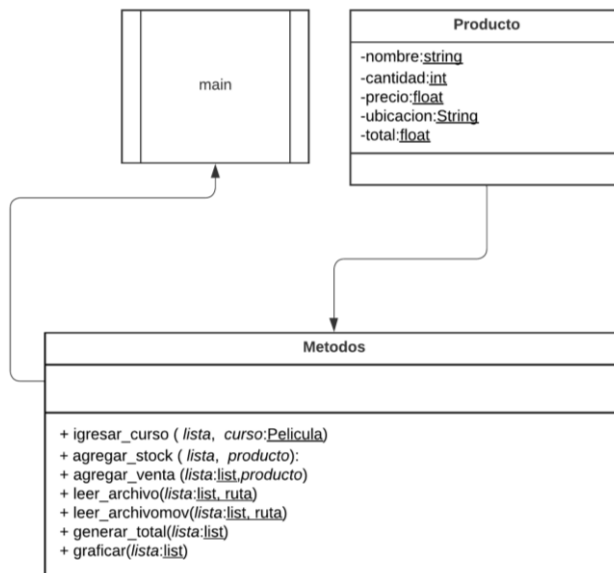
```
def es_funcion():
    es_variable="hola mundo"

return es_variable
```

## [DIAGRAMA DE CLASES]

Clase UML Gestion de inventario

luis lemus | August 22, 2023



## [MÉTODOS PRINCIPALES]

- Leer archivo

Descripción: A partir de una ruta de un archivo inv y mov con uso de la función `.split` se separan por (;) y comas para guardar los datos del archivo

```

1  def leer_archivo(lista:list,ruta):
2      cantidad=0
3      pelicula=""
4      precio=""
5      ubicacion=""
6
7      with open(ruta, encoding='utf-8') as f :
8
9          for x in f:
10             listalocal=[]
11             listalocal=x.split(";")
12             i=1
13             pelicula=listalocal[0]
14             cadena1=pelicula.strip()
15             nombre=cadena1.replace("crear_producto ", "")
16             cantidad=int(listalocal[1])
17             precio=listalocal[2]
18             ubicacion=listalocal[3].strip()
19             pro=Producto(nombre,cantidad,precio,ubicacion)
20             Metodos.igresar_curso(lista,pro)

```

## Agregar venta

Descripción: A partir de una lista de productos se evalúa si el producto existe en la ubicación y si lo que se va vender sea <= de lo que se tiene en bodega

```

1  def agregar_venta ( lista, curso:Producto):
2      ubicaicon=curso.ubicacion
3      existe = False
4      menor=False
5
6      if not lista:
7          print("no hay productos")
8
9      else:
10         for x in lista:
11             nux=x.ubicacion
12             #print(x.ubicacion,curso.ubicacion,curso.nombre,"*****//")
13
14             if nux==ubicaicon and x.nombre==curso.nombre and curso.cantidad > x.cantidad:
15                 menor=True
16                 break
17             if nux==ubicaicon and x.nombre==curso.nombre and curso.cantidad<=x.cantidad:
18                 existe = True
19                 x.cantidad-=curso.cantidad
20                 break
21
22         if existe == False:
23             print("-----ERROR---- NO existe el producto ",curso.nombre," en esa ubicacion")
24         if menor==True:
25             print("-----ERROR---- No hay existencias del producto ",curso.nombre," en esta ubicacion")

```

## Filtro actor

Descripción: A partir de una lista de productos se evalúa si el producto existe en la bodega si es correcto se aumentara el inventario de ese producto.

```
1 def agregar_stock ( lista, curso:Producto):
2     ubicaicon=curso.ubicacion
3     existe = False
4
5     if not lista:
6         print("no hay productos")
7
8     else:
9         for x in lista:
10             nux=x.ubicacion
11             #print(x.ubicacion,curso.ubicacion,curso.nombre,"*****//")
12
13
14             if nux==ubicaicon and x.nombre==curso.nombre:
15                 existe = True
16                 x.cantidad+=curso.cantidad
17                 break
18
19     if existe == False:
20         print("-----ERROR----- NO existe el producto  ",curso.nombre,"  en esa ubicacion")
```

## Método para generar el resultado

Descripción: a partir de una lista de productos se muestra un reporte de los productos existentes en cada bodega

```
1 def graficar(lista:list):
2     Metodos.generar_total(lista)
3     cadenatotal=""
4     pie="..."
5     producto=Producto
6     textopro=""
7     cadena=""
8     encabezado="Informe de Inventario:\nProducto    Cantidad    Precio Unitario    Valor Total    Ubicación\n-----"
9     for producto in lista:
10         textopro=producto.nombre+"\t"
11         cadena=textopro+" "+str(producto.cantidad)+" "+str(producto.precio)+" "+str(producto.total)+" "+producto.ubicacion+"\n"
12     cadenatotal=encabezado+cadena+"\n"+pie
13
14     with open("resultado.txt", "w", encoding="utf-8") as f:
15         f.write(cadenatotal)
```

[REQUERIMIENTO O FUNCIONALIDAD ESPECÍFICA]

[DESCRIPCIÓN]

import os

from tkinter import filedialog

Se usó librerías de tkinter para abrir el explorador

Se usó os para la exportación del archivo.txt

[INTERFACES PRINCIPALES]

```
-----  
                        BIENVENIDO  
Lenguajes formales y de programacion  
Seccion:B+  
Nombre: Luis Humberto Lemus Perez      Carnet:201445840  
-----  
Presione 1 para continuar  ....  
█
```

[Mensaje de Bienvenida]

[VERIFICACIÓN DE VALIDACIONES]

[VALIDACION #1]

```
Cargando....  
-----Error El contenido del archivo no es valido o no has selecionado un archivo-----  
-----  
                        Selecciona una opcion:  
                        1) Cargar inventario inicial  
                        2) Cargar instrucciones de movimiento  
                        3) crear informe de inventario.  
                        4) Salir
```

[Error de búsqueda del archivo o un archivo incorrecto]

## [VALIDACION #2]

```
-----ERROR----- NO existe el producto   Plátanos   en esa ubicacion
-----ERROR----- NO existe el producto   Peras     en esa ubicacion
-----ERROR----- NO existe el producto   Naranjas   en esa ubicacion
-----ERROR----- NO existe el producto   Fresas     en esa ubicacion
-----ERROR----- No existen existencias del producto   Fresas     en esta ubicacion
-----ERROR----- NO existe el producto   Papayas    en esa ubicacion
-----ERROR----- NO existe el producto   Manzanas   en esa ubicacion
-----ERROR----- NO existe el producto   Mangos     en esa ubicacion
-----ERROR----- NO existe el producto   Melones    en esa ubicacion
-----ERROR----- NO existe el producto   Cerezas    en esa ubicacion
-----ERROR----- NO existe el producto   Duraznos   en esa ubicacion
-----ERROR----- NO existe el producto   Limones    en esa ubicacion
-----ERROR----- NO existe el producto   Sandías    en esa ubicacion
-----ERROR----- NO existe el producto   Moras      en esa ubicacion
-----ERROR----- NO existe el producto   Plátanos   en esa ubicacion
-----ERROR----- NO existe el producto   Naranjas   en esa ubicacion
-----ERROR----- NO existe el producto   Papayas    en esa ubicacion
-----ERROR----- NO existe el producto   Peras      en esa ubicacion
-----ERROR----- NO existe el producto   Melones    en esa ubicacion
```

[Error producto incorrecto o sin existencia en la bodega o fuera de rango]