

# MANUAL TECNICO

## PRACTICA 1

JOSE RICARDO MENOCAL KONG

202000886 LAB LFP B

# Manual técnico

## Informe sobre el programa

Manual dirigido únicamente y especialmente al técnico u programador que haga un estudio detallado del programa, fue desarrollado en el lenguaje Python versión 3.10.4., es un software que contiene distintas funciones que nos permite la lectura, gestión y almacenamiento de datos; toda la información referente será extraída de un archivo “.lfp”.

## Requerimientos del sistema

- Sistema operativo windos 7, windos 8, windos 10, windos 11 .
- Procesador pentium hasta procesadores de gama alta.
- 1Gb de ram
- Espacio en disco duro de 256mb
- Editor de texto utilizado visual Studio Code.

## Estructura del programa (codigo)

Este programa fue desarrollado con el paradigma de programación orientada a objetos, por lo que tenemos 4 archivos.

### 1. Class.py

- Class Peliculas():

Contiene la estructura inicial y las funciones creadas, asociadas con el archivo.py.

```
class Peliculas:
    #CONSTRUCTOR
    def __init__(self,num, nombre, actores, año, genero):
        self.num = num
        self.nombre = nombre
        self.actores = actores
        self.año = año
        self.genero = genero
```

- Funciones:

1. **mostrarinfo(self)**: esta función Muestra toda la cargada al programa.

```
#FUNCION DE MOSTRAR TODAS LAS PELICULAS
def mostrarinfo(self):
    print("ID:",self.num,
          "\nPELICULA: ",
          self.nombre,
          "\nACTORES: ",
          self.actores,
          "\nAÑO ESTRENO: "
          , self.año,
          "\nGENERO: ",
          self.genero)
```

2. **mostrarActorPorNumero(self,num)** :esta función muestra el actor por el Id de cada película.

```
#FUNCION DE MOSTRAR PELICULA POR NUMERO
def mostrarActorPorNumero(self,num):
    if num == self.num:
        print("ID:",self.num,
              "\nPELICULA: ",
              self.nombre,
              "\nACTORES: ",
              self.actores,
              "\nAÑO ESTRENO: "
              , self.año,
              "\nGENERO: ",
              self.genero)
```

3. **mostrarSoloPeliculas(self)**: esta función muestra únicamente el ID y el nombre de la película.

```
#FUNCION DE MOSTRAR SOLO PELICULAS CON SU ID
def mostrarSoloPelicula(self):
    print("ID:",self.num,"PELICULA: ",self.nombre)
```

4. **mostrarPorActor(self,dato):** esta función muestra todos los datos de las películas asociadas al actor.

```
#FUNCION DE FILTRAR PELICULAS POR ACTORES
def mostrarPorActor(self,dato):
    #se crea if para comparar si el dato es en self.actores
    if dato in self.actores:
        #Devuelve los datos de la pelicula que contiene self.actores
        print("\nID:",self.num,
              "\nPELICULA: ",
              self.nombre,
              "\nACTORES: ",
              self.actores,
              "\nAÑO ESTRENO: "
              , self.año,
              "\nGENERO: ",
              self.genero)
```

5. **mostrarPorAño(self,dato):** esta función muestra todo los datos de las películas asociadas al año.

```
#FUNCION DE FILTRAR PELICULAS POR AÑO
def mostrarPorAño(self,dato):
    #Devuelve los datos de la pelicula que contiene self.año
    if dato in self.año:
        print("\nID:",self.num,
              "\nPELICULA: ",
              self.nombre,
              "\nACTORES: ",
              self.actores,
              "\nAÑO ESTRENO: "
              , self.año,
              "\nGENERO: ",
              self.genero)
```

6. **mostrarPorGenero(self,dato):** esta función muestra todo los datos de las películas asociadas al género.

```
#FUNCION DE FILTRAR PELICULAS POR GENERO
def mostrarPorGenero(self,dato):
    #Devuelve los datos de la pelicula que contiene self.genero
    if dato in self.genero:
        print("\nID:",self.num,
              "\nPELICULA: ",
              self.nombre,
              "\nACTORES: ",
              self.actores,
              "\nAÑO ESTRENO: "
              , self.año,
              "\nGENERO:",
              self.genero,)
```

## 2. Carga\_archivo.py

Este archivo contiene la carga del archivo de texto que contiene los datos de nuestro programa.

- **Importaciones:** se importa la clase películas del archivo Class.

```
#importaciones
from Class import Peliculas
```

- **Funcion leerArchivoEntrada(Lista):** Esta función abre el archivo y lee su contenido para luego almacenar los datos en una lista.(Lea los comentarios de cada línea).

```
#FUNCION LEER ARCHIVO
def leerArchivoEntrada(lista):
    #ABRIR ARCHIVO
    #Le solicita al usuario la ruta para leer el archivo
    ruta = input("Ingrese la ruta ('C:\...') para leer el archivo .
    #La variable archivo es igual a la ruta del archivo a abrir
    archivo = open(ruta, 'r')
    #LEER ARCHIVO LINERA POR LINEA
    lineas = archivo.readlines()
    #CERRAR ARCHIVO
    archivo.close()
    #Declaramos Datonum y lo inicializamos
    Datonum=0
    #Creamos un ciclo for para recorrer las lineas del archivo
    for i in lineas:
        #declaramos que las iteraciones estaran separadas por ";"
        i = i.split(";")
        #Declaramos la variable count y la inicializamos
        count = 1
        #Declaramos la variable Datonum +=1 para que ...
        #... se incremente en uno
        Datonum +=1
        #Inicializamos las variables con una dato valido
        DatoNombre = None
        DatoActores= None
        DatoAño = None
        DatoGenero = None
        #creamos un ciclo for para recorrer las columnas de las lineas
        for j in i:
            #creamos if y elif para asignarle las...
            #...columnas a las variables de los datos
            if count == 1:
                DatoNombre = j
            elif count == 2:
                j = j.split(",")
                DatoActores = j
            elif count == 3:
                DatoAño = j
            elif count == 4:
                DatoGenero = j
            #incrementa el contador
            count += 1
        #Declaramos la variable y decimos que sus datos ...
        #...seran los datos de nuestro constructor
        peli = Peliculas(Datonum,DatoNombre, DatoActores, DatoAño, DatoGenero)
        #Agregamos los datos a la lista
        lista.append(peli)
```

### 3. Menu.py

Este archivo contiene el menú interactivo del programa, donde el usuario puede interactuar con el programa, también se encuentra un recuadro de datos del autor.

- **Importaciones:** se importa la función leerArchivosEntrada de carga\_archivo, también se importan librerías de graphviz y la de os.

```
from carga_archivo import leerArchivoEntrada
import graphviz
import os
---
```

- **Variables Globales:** En este programa solo se declararon dos variables globales una para almacenar los datos dentro de una lista y el otro el id inicial que es igual a 0.

```
#Inicializamos el arreglo y el id
global ArregloPeliculas
ArregloPeliculas = []
id=0
```

- **Cuadro de datos de autor:**

```
# Recuadro de Datos de autor
print(
    "\n|          Lenguajes Formales y de Programación          |",
    "\n|          Sección: A+                                     |",
    "\n|          202000886 - José Ricardo Menocal Kong          |",
    "\n|          |"
)
#Termina recuadro
```

- **Menu:** contiene el parte del código donde mandamos a llamar a las funciones dependiendo sea al caso, ((los comentarios de cada línea).A continuación lo se describe por opciones:

```
#SE CREA UN WHILE PARA RECORRER EL MENU UNA Y OTRA VEZ

while op != 0:

    #OPCIONES MENU

    print("-----MENU PRINCIPAL-----")
    print("1 CARGAR ARCHIVO DE ENTRADA")
    print("2 GESTIONAR PELICULAS")
    print("3 FILTRADO")
    print("4 GRAFICA")
    print("5 SALIR")
    print("-----")
    #Ingreso de la opcion por el usuario
    op = int(input("Elija una opcion de 1 a 5: "))
```

- 1. Cargar archivo: Esta opción llama a la funcion

```
# CARGA DE ARCHIVOS

if op == 1:
    #Se invoca la funcion de leerArchivoEntrada()
    print("***** CARGA DE ARCHIVOS *****\n")
    leerArchivoEntrada(ArregloPelículas)
    print("CARGA EXITOSAMENTE\n")
    print("***** ***** ** ***** *****")
```

- 2. Gestión: se divide en dos, mostrar película y mostrar actores.

```
#2. GESTION

elif op == 2:
    #Se muestra el menu de GESTION
    print("***** GESTIONAR PELICULAS *****\n")
    print("1. Mostrar peliculas: ",
          "\n2. Mostrar actores: ")
    #Se ingresa la opcion deseada
    op = int(input("Elija una opcion: "))
    print("***** ***** ***** *****\n")
```

```
#2.1 MOSTRAR PELICULAS

#se crea un if para que, si la opcion ingresada es igual a 1
if op ==1:
    #se recorre el arreglo a paso i
    for i in ArregloPelículas:
        #se invoca la funcion mostrar info()
        i.mostrarinfo()
    print("***** ***** ***** *****")
```

```
#2.2 MOSTRAR ACTORES DE LAS PELICULAS

#se crea un if para que, si la opcion ingresada es igual a 2
if op ==2:
    #se recorre el arreglo a paso i
    for i in ArregloPelículas:
        #se invoca la funcion mostrar mostrarSoloPelicula()
        i.mostrarSoloPelicula()
        print("***** ***** ***** *****")
    #se pide que ingrese el numero a visualizar de tipo INT
    id = int(input("Ingrese el numero de la pelicula: "))
    print("***** ***** ***** *****")
    #se recorre el arreglo a paso i
    for i in ArregloPelículas:
        #se invoca la funcion mostrar mostrarActorPorNumero()
        i.mostrarActorPorNumero(id)
    print("***** ***** ***** *****")
```

- 3. Filtro: se divide en 3, el filtrado por actor , por año y por género.

### #3. FILTRO

```
elif op == 3:
    print("*****FILTRO*****")
    print("1 Filtrado por actor")
    print("2 Filtrado por año")
    print("3 Filtrado por genero")
    #Ingreso de opcion tipo INT
    opc = int(input("Ingrese una opcion: "))
```

#### #3.1 FILTRO POR ACTOR

```
#se crea un if para que, si la opcion ingresada es igual a 1
if opc==1:
    print("***** FILTRADO POR ACTOR *****")
    #Ingreso del dato a buscar de tipo STRING
    act = str(input("Ingrese el nombre de un actor: "))
    print("***** ***** ***** *****")
    #se recorre el arreglo a paso i
    for i in ArregloPelículas:
        #se invoca la funcion mostrar mostrarPorActor(act)
        i.mostrarPorActor(act)
    print("***** ***** ***** *****")
```

#### #3.2 FILTRO POR AÑO

```
#se crea un if para que, si la opcion ingresada es igual a 2
elif opc==2:
    print("***** FILTRADO POR AÑO *****")
    #Ingreso del dato a buscar
    año = input("Ingrese el año: ")
    print("***** ***** ***** *****")
    #se recorre el arreglo a paso i
    for i in ArregloPelículas:
        #se invoca la funcion mostrar mostrarPorActor(act)
        i.mostrarPorAño(año)
    print("***** ***** ***** *****")
```



```
#3.3 FILTOR POR GENERO
#se crea un if para que, si la opcion ingresada es igual a 3
elif opc==3:
    print("***** FILTRADO POR GENERO *****")
    #Ingreso del dato a buscar de tipo STRING
    genero = str(input("Ingrese el genero: "))
    print("***** ***** ***** *****")
    #se recorre el arreglo a paso i
    for i in ArregloPeliculas:
        #se invoca la funcion mostrar mostrarPorGenero(genero)
        i.mostrarPorGenero(genero)
    print("***** ***** ***** *****")
```

- Gráfica: Muestra la gráfica.

```
#4. GRAFICO
elif op == 4:
    print("-----GRAFICA-----")
    s = graphviz.Digraph('structs', filename='structs_revisited.gv', node_attr={'shape': 'record'})
    for i in ArregloPeliculas:
        s.node('struct1', '{{<pele> {i.nombre} |{{i.Año}}|{i.genero}}}')
        s.node('struct2', '<f0> {i.actores}')
        s.edges([('struct1:pele', 'struct2:f0')])
    print(s.source)
    r=open("C:\LFP\LFP_202000886\src\Reportes\Report.dot","w")
    r.write(s.source)
    r.close()
    os.system("dot -Tpng Reportes/Report.dot -o Reportes/Report.png")
    webbrowser.open_new_tab("C:\LFP\LFP_202000886\src\Reportes\Report.png")
    print("***** ***** ***** *****")
    print("SE CREO LA GRAFICA EXITOSAMENTE")
    print("***** ***** ***** *****")
```

- Salir: Manda un mensaje y detiene la ejecución del programa.

```
#5. SALIDA
print("***** ***** ***** *****")
print("| GRACIAS POR USAR EL PROGRAMA |")
print("***** ***** ***** *****")
break
```

## Planificación

Tiempo en horas	36 horas invertidas
Tarea #1 general	Realizar el diseño del programa
Tarea #2 general	Crear la clase principal
Tarea #3 general	Carga y lectura de datos.
Tarea #4 general	Creación de Funciones

## Glosario

- **.split(";")**: función de Python para separar datos o caracteres.
- **Def nombre()**: Definición de una función en Python.
- **.append()**: Función de Python para agregar un dato.
- **Import graphviz**: importación de la librería de graphviz.