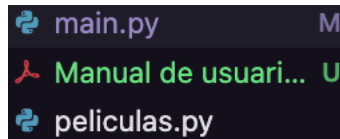


## Manual Técnico Sebastian Solares

Para la realización de este programa se utilizaron 2 archivos un archivo main.py que es donde va almacenado toda la programación del programa y un archivo ppelículas.py donde se almacenará toda la información agregada del archivo de entrada.



Se crearon varias funciones para realizar cada parte del proyecto como por ejemplo cargar archivo de entrada, filtrar películas, graficar etcétera.

```
> def cargarArchivos(): ...  
> def addPelículas(): ...  
  
> def grafico(): You, 2 days  
  
> def filtradoPelículas(): ...
```

Cargar archivo: en esta función se utilizó tekniker para obtener la ruta del archivo leerla y almacenar la informacion correspondiente.

```

def cargarArchivos():
    global df
    global ruta
    global leido

    sg.theme('LightTeal')

    layout = [[sg.Text('Filename')],
               [sg.Input(), sg.FileBrowse(file_types=(( "*.lfp"),)),
                sg.OK("ok")]]

    window = sg.Window('Cargar Archivo LFP', layout,)

    while True:
        event, values = window.read()
        if event in (sg.WIN_CLOSED, 'Cancesdal'):
            break
        elif event == "ok" :
            general =[]
            while True:
                try:
                    print("RUTA DEL ARCHIVO:" + values[0])

                    sg.popup('Archivo cargado correctamente')

                    ruta = values[0]

                    break

                except:
                    sg.popup('El archivo no se cargo correctamente')
                    break

            window.close()

```

AddPelículas: En este apartado se hicieron las validaciones necesarias para saber si una película ya está repetida o no y de lo contrario si no está repetida almacenarla en las listas correspondientes.

También está el apartado de gestionar películas en el cual se creó un menú donde podemos seleccionar la opción a desear, como mostrar películas, actores etc.

```

def addPelículas():
    exit = False
    pelirepetida = False

    try:
        objeto = open(ruta, "r+")

        info = objeto.read()

        infoxlinea = info.splitlines()

        for x in infoxlinea:
            line = x.split(";")

            name = line[0]
            actor = line[1]
            year = line[2]
            gender = line[3]

            peliagregada = Películas(name, actor, year, gender)
            actorAgg = actor

            for i in range(len(películas)):
                if películas[i].nombre == name:
                    print("Ya existe una película con ese nombre")
                    pelirepetida = True

            if pelirepetida == False:

```

```

while exit== False:

    #menu De gestion de peliculas
    print('-----')
    print('-----GESTION DE PELICULAS-----')
    print("--1. Mostrar peliculas--")
    print("--2. Mostrar Actores--")
    print("--3. Regresar--")
    print('-----')

    opcion2 = int(input("Ingrese una opcion: "))

    if opcion2 == 1:
        exit = False

        print('-----hola estas en el apartado de mostrar peliculas-----')
        for x in range(len(peliculas)):
            Peliculas.imprimirPeliculas(peliculas[x])
    elif opcion2 == 2:
        exit = False

        print('-----hola estas en el apartado de mostrar actores-----')
        for i in range(len(peliculas)):
            count= i+1
            print(count,peliculas[i].nombre)

        opcion3 = int(input("Ingrese una opcion: "))
        print('Actores: ',peliculas[opcion3-1].actores)

    elif opcion2 == 3:
        exit= True

```

Gráfico: en este apartado realice todo lo relacionado con graphviz para crear los nodos, las uniones, crear el archivo dot y el pdf

```

def crear_nodo(pelicula, anio, genero):
    global iteracion
    iteracion += 1
    return f'''\nnodo{iteracion} [label=<
<table border="0" cellborder="1" cellspacing="0">
<tr><td bgcolor="#0091ea" port="p1" colspan="2">{pelicula}</td></tr>
<tr><td> {anio}</td><td> {genero}</td></tr>
</table>>];\n\n'''

def crear_actor(actor):
    return f'\t"{actor}"\n' # Eso lo pueden omitir xd

def crear_relacion(nodo,actor):
    return f'''\tnodo{nodo}:p1 -> "{actor}";\n'''

def agregar_estilo():

    data += 'node [shape=box, style=filled, fillcolor="#00c853"]\n'

# Aqui creamos los nodos de peliculas
for pelicula in peli.keys():
    anio = peli[pelicula]['Anio']
    genero = peli[pelicula]['Genero']
    nodo =crear_nodo(pelicula, anio, genero)
    data += nodo

```

Filtrado de películas: para esta parte se creo un menú en donde el usuario pueda navegar entre las diferentes opciones como filtrar películas por año, actor o genero

```
def filtradoPelículas():
    exit = False
    while exit == False:

        #menu De FILTRACION DE PELICULAS
        print('-----')
        print('-----FILTRACION DE PELICULAS-----')
        print("--1. Filtrado Por Actor--")
        print("--2. Filtrado Por año--")
        print("--3. Filtrado Por Genero--")
        print("--4. Regresar--")
        print('-----')

        opcion2 = int(input("Ingrese una opcion: "))

        if opcion2 == 1:
            encontrado = False
            encontrado2 = False
            count=0
            print('-----hola estas en el apartado de filtrado por actor-----')
            opc = str(input("Ingrese el nombre del actor: "))

            for i in range(len(peliculas)):

                count= i+1

                if peliculas[i].actores== opc:

                    encontrado = True
                    print('actor encontrado')

                else:
                    encontrado = False
                    print('actor no encontrado')

            if encontrado == True:
                print('Película: ',peliculas[count-1].nombre)
```

Menu principal: Para realizar el menú principal se utilizo un while y un input para que al elegir la opción despliegue el apartado a desear

```
def MenuPrincipal():

    #menu en consola
    print('-----BIENVENIDO-----')
    print("--1. Cargar Archivo de Entrada--")
    print("--2. Gestionar Peliculas--")
    print("--3. Filtrado de Peliculas--")
    print("--4. Graficar Peliculas--")
    print("--5. Salir--")

while True:
    MenuPrincipal()
    opcion = int(input("Ingrese una opcion: "))
    if opcion == 1:
        cargarArchivos()
    elif opcion == 2:
        addPeliculas()

    elif opcion == 3:
        filtradoPeliculas()

    elif opcion == 4:
        grafico()

    elif opcion == 5:
        print("Gracias por usar el programa")
```