

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Lenguajes Formales y de Programación

Sección: B+

Ing. David Morales

Auxiliar: Diego Obín

MANUAL DE TECNICO

Nombre: Andrés Alejandro Agosto Méndez

Carnet: 202113580

Manual técnico

Archivo Main: Se utilizó el paradigma procedimental para hacer la ventana inicial, y al insertar la tecla que se pide se inicia el menú principal, el menú está en un while para que se repita luego de terminar alguna opción

```
carga = Cargar("", "", 0, "")

Peliculas = [] # arreglo contendrá todas las películas

print("\nLenguajes Formales y de Programación")
print("Andrés Alejandro Agosto Méndez 202113580 sección B+")
print("-----Menú principal-----")

Entrada = int(input("pulse 1 para ir al menú principal: "))

if Entrada == 1:
    aviso = False
    while aviso == False:
        aviso = False
        a=1
        print("\nMenú principal")
        print("1) Cargar archivo de entrada")
        print("2) Gestionar películas")
        print("3) Filtrado")
        print("4) Gráfica")
        print("5) Salir")
```

Objeto Cargar: Se creó un objeto con 4 atributos que corresponden a los datos que debe tener el archivo .lfp.

```
class Cargar:
    def __init__(self, nombre, actores, año, género):
        self.nombre = nombre
        self.actores = actores
        self.año = año
        self.género = género
```

Método LeerArchivo(): Esta función pide un arreglo definido, luego con la variable ruta pide la dirección del archivo para abrir el archivo y leerlo, luego con el método readlines convierte el archivo leído en una lista que contiene cada línea del archivo, luego con un for recorre la lista de líneas y con el método Split crea una lista para cada línea que contine los atributos del objeto.

```
def LeerArchivo(self, arreglo):
    #ruta = "/Users/gmg/Desktop/usac 2023/practica 1 LFP/prueba.lfp"
    ruta = input("escriba la ruta exacta del archivo: ")
    archivo = open(ruta, 'r') #abrimos el archivo y leemos con r de read
    Listalíneas = archivo.readlines() #leemos línea por línea, y crea un arreglo de líneas del archivo
    for i in Listalíneas: #recorremos las líneas
        contador = 1
        i=i.split(";") # separamos por punto y coma para hacer el arreglo por línea

        for j in i:
            if contador == 1:
                self.nombre = j
            elif contador == 2:
                j = [x.strip() for x in j.split(",")] # con sprit eliminamos espacios en blanco de las esquinas y se las pasamos a cada elemento de la
                self.actores = [x.strip() for x in j] #aquí se lo asignamos a cada variable la lista sin espacios
            elif contador ==3:
                self.año=j
            elif contador ==4:
                self.género= j
            contador +=1

        Existe = False
        for i in arreglo:
            if i.nombre == self.nombre: #en la segunda iteración de Listalíneas verifica
                Existe = True #si algún elemento es igual del arreglo, si lo es entonces no permite agregarlo a la lista principal
                break

        if Existe == False:
            lica = Cargar(self.nombre,self.actores,self.año,self.género)
            arreglo.append(lica) #agregamos al arreglo
        archivo.close()
```

Luego hay un for anidado para guardar los arreglos en cada atributo, con un contador, pero en el atributo actores se crea otra lista con los actores divididos, y con el método strip para cada elemento de la lista le quitamos los espacios vacíos de los lados, ya luego verificamos que no se repitan los nombres de las películas recorriendo el arreglo principal, y se verificó que no sean iguales antes de agregar el objeto con todos los atributos al arreglo principal, por último se cierra el archivo.

Métodos impresores: los primeros tres métodos solo llaman a los atributos, pero el imprimirActores1; pide el arreglo principal y crea un nuevo arreglo, luego recorrerá el arreglo principal y el arreglo de actores que se creó y así se irá agregando todos los actores al nuevo arreglo sin importar que se repitan, y luego se crea un arreglo self.menuDefinitivo y se almacenó el arreglo con los actores repetidos y con el método list se puede recorrer y se eliminan los repetidos con el método set, luego se recorre el arreglo self.menuDefinitivo e imprime los actores sin repetir.

```
def imprimirDatos(self):
    print("\nNombre: ", self.nombre, "\nActores: ", self.actores, "\nAño: ", self.año, "\nGe

def imprimirPelículas(self, a):
    if a == True:
        print(self.nombre+":")
    elif a == False:
        print(self.nombre)

def imprimirActores(self):
    print(self.actores)

def imprimirActores1(self, arr, aa):
    nuevo = []
    for i in range(len(arr)):
        for j in range(len(arr[i].actores)):
            nuevo.append(arr[i].actores[j])
    self.menuDefinitivo = list(set(nuevo))

    if aa == True:
        for k in range(len(self.menuDefinitivo)):
            print(str(k+1)+")"+self.menuDefinitivo[k])
```

Función imprimirLica(): pide el arreglo principal y una variable de entrada, recorremos el arreglo principal y el arreglo de actores, luego se verifica que el arreglo self.menuDefinitivo con el índice de entrada sea igual con los recorridos del arreglo principal y el arreglo actores, y así imprimir el nombre del actor y el nombre de las películas que estuvo.

Función imprimirAños(): pide el arreglo principal y crea un nuevo arreglo, luego recorre el arreglo principal para agregar todos los años al nuevo arreglo, luego se crea el arreglo self.menuDefinitivo1; usa el método list para iterarlo y así eliminar los años repetidos, por último recorre el arreglo self.menuDefinitivo1 e imprime los años sin repetir.

```
def imprimirLica(self, arr2, entrada):
    a=1
    for i in range(len(arr2)):
        for j in range(len(arr2[i].actores)):
            if self.menuDefinitivo[entrada] == arr2[i].actores[j]:
                if a==1:
                    print("\npelículas en las que a participó "+arr2[i].actores[j]+":")
                    print(arr2[i].nombre)
                a+=1

def imprimirAños(self, arr1):
    nuevo1 = []
    for i in range(len(arr1)):
        nuevo1.append(arr1[i].año)

    self.menuDefinitivo1 = list(set(nuevo1))

    for j in range(len(self.menuDefinitivo1)):
        print(str(j+1)+")"+str(self.menuDefinitivo1[j]))
```

Función `printNombreYGenero()`: este pide el arreglo principal y una variable de entrada, primero recorre el arreglo principal y verifica que `self.menuDefinitivo1` con los años no repetidos con índice de la variable entrada sea igual al arreglo principal con el atributo del año, luego imprime el año y el género en ese momento.

Función `imprimirGeneros()`: pide el arreglo principal, luego define un nuevo arreglo, recorre el arreglo principal para llenar el arreglo nuevo con los géneros repetidos, luego crea un arreglo `self.menuDefinitivo2`, y agrega el arreglo nuevo e itera para eliminar los géneros repetidos, luego recorre el arreglo `self.menuDefinitivo2` para imprimir los géneros sin repetir.

Función `printLicasXgenero()`: pide el arreglo principal y una variable de entrada, recorre el arreglo principal y verifica que `self.menuDefinitivo2` con los géneros no repetidos sea igual al arreglo principal en el género en ese momento, imprime el nombre del arreglo principal en el momento que se verifica.

```
def printNombreYGenero(self, arr3, entrar):
    b=1
    for i in range(len(arr3)):
        if self.menuDefinitivo1[entrar] == arr3[i].año:
            if b==1:
                print("\nnombre y género de las películas del año "+arr3[i].año+":")
                print("\nNombres: "+arr3[i].nombre+"\nGénero: "+arr3[i].género+"\n")
            b+=1

def imprimirGeneros(self, arr4):
    nuevo2 = []
    for i in range(len(arr4)):
        nuevo2.append(arr4[i].género)

    self.menuDefinitivo2 = list(set(nuevo2))

    for n in range(len(self.menuDefinitivo2)):
        print(str(n+1)+")"+str(self.menuDefinitivo2[n]),end="")

def printLicasXgenero(self, arr5, entra):
    c=1
    for i in range(len(arr5)):
        if self.menuDefinitivo2[entra] == arr5[i].género:
            if c==1:
                print("\nnombres de las películas con el género de "+arr5[i].género+":")
                print("\n"+arr5[i].nombre+"\n")
            c+=1
```

Aplicaciones en el archivo main: Luego con condiciones `if` se mandana llamar todos los métodos vistos para imprimirlos en consola, y se acceden con variables de entrada según el

numero puesto en el menú principal, y con ciclos while se puede regresar a la parte que se necesita.

```
if Entrada1==1:
    carga.LeerArchivo(Peliculas)

    print("\nCargado Exitosamente")
elif Entrada1 == 2:
    aa= True
    while aa==True:
        a=1
        print("\n1) Mostrar películas")
        print("\n2) Mostrar actores")
        print("\n3) regresar")
        Entrada2= int(input("ingrese una opción: "))

        if Entrada2 == 1:
            print("\nInformación sobre las películas:")
            for i in Peliculas:
                i.imprimirDatos()
        elif Entrada2 == 2:
            for i in Peliculas: #impresión del menú
                print(str(a)+"",end=" ")
                i.imprimirPeliculas(False)
                a+=1
            print("presione 0 para regresar")

            entrada3= int(input("ingrese una opción: "))
            if entrada3 == 0:
                pass
            else:
                print ("Actores de la película ",end="")
                Peliculas[entrada3-1].imprimirPeliculas(True)

                Peliculas[entrada3-1].imprimirActores()
        elif Entrada2 == 3:
            aa = False
    elif Entrada1 == 3:
        h= True
        while h == True:
            print("\n1) Filtrado por actor")
            print("\n2) Filtrado por año")
            print("\n3) Filtrado por género")
            print("\n4) regresar")

            Entrada4= int(input("ingrese la opción: "))
            if Entrada4 == 1:
                carga.imprimirActores1(Peliculas,True)
                entrada5 = int(input("elijá un actor: "))
                carga.imprimirActores(Peliculas,entrada5-1)

            elif Entrada4 == 2:
                carga.imprimirAños(Peliculas)
                entrada6 = int(input("elijá un año: "))
                carga.printNombreYGenero(Peliculas,entrada6-1)
            elif Entrada4 == 3:
                carga.imprimirGeneros(Peliculas)
                entrada7 = int(input("elijá un género: "))
                carga.printPeliculasXGenero(Peliculas,entrada7-1)
            elif Entrada4 == 4:
                h= False
```

Gráfica de de graphviz: en la opción 4, primero crea una plantilla dot con la librería graphviz, luego se le denomina como nodo en texto plano, recorremos el arreglo principal para hacer los nodos que son el número de líneas del archivo, y en cada nodo rectangular se imprime el nombre, año y género en ese momento del arreglo, después recorremos el arreglo Carga.menuDefinitivo con los actores sin repetir y creamos un nodo para cada actor, por último se recorrió el arreglo Carga.menuDefinitivo, recorremos el arreglo principal y recorremos el arreglo actores en ese momento para verificar Carga.menuDefinitivo sea igual al actor en ese momento luego de todos los recorridos de los for anidados, para enlazar los nodos con las mismas condiciones, y se vea de manera gráfica que actores participaron en cierta película, para por último renderizar el gráfico en pdf.

Terminar programa: la opción 5, solo tiene un mensaje en consola como “Hasta la próxima . . .”, y termina el programa con el método exit.

```
elif Entrada1 == 4:

    dot = Digraph(comment='Tablas guías', format='pdf') # diagraph es toda la plantilla donde irán los nodos
    dot.attr('node', shape='plaintext') #establece la forma de los nodos como texto plano

    for i in range(len(Peliculas)):
        dot.node(f'tabla{i}',
f'''<table border="1" cellborder="1" cellspacing="0">
    <tr><td colspan="2" bgcolor="red">{Peliculas[i].nombre}</td></tr>
    <tr><td>{Peliculas[i].año}</td><td>{Peliculas[i].género}</td></tr>
</table>''')
        for i in range(len(carga.menuDefinitivo)):
            dot.node(f'tabla{i+100}', #inicializar el arreglo menú definitivo y dar un salto de línea
f'''<table border="1" cellborder="1" cellspacing="0">
    <tr><td colspan="2" bgcolor="yellow">{carga.menuDefinitivo[i]}</td></tr>
</table>''')

        for i in range(len(carga.menuDefinitivo)):
            for j in range(len(Peliculas)):
                for k in range(len(Peliculas[j].actores)):
                    if carga.menuDefinitivo[i] == Peliculas[j].actores[k]:
                        dot.edge(f'tabla{j}', f'tabla{i+100}', dir="back")
# Renderizamos todo
    dot.render('tablas', view=True)
    elif Entrada1 == 5:
        print("\nHasta la próxima . . .\n")
        exit()
```