

Universidad de San Carlos De Guatemala
Facultad de Ingeniería
Lenguajes Formales y de Programación “B-“

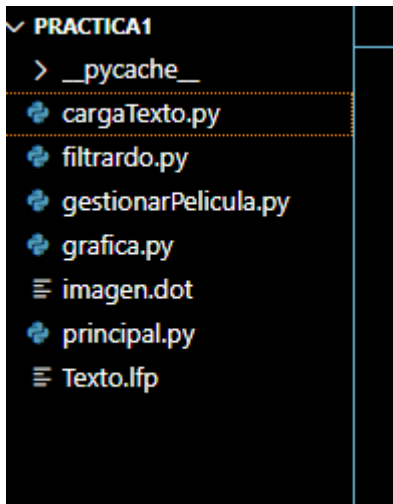
MANUAL TÉCNICO

Nombre: Billy David Must Ochoa

Carné: 202000353

Fecha: 24/02/2023

Para el desarrollo de esta práctica se utilizaron dos ficheros, el principal, contuvo toda la estructura para desarrollar el menú, el segundo fichero se utilizó para estructurar algunos métodos que posteriormente se iban a utilizar en la clase principal.



Los ficheros son cargaTexto.py y principal.py

Principal.py

```

while opcion!=5:
    print("-----")
    print("Menu principal")
    print("1)Carga")
    print("2)Gestion")
    print("3)Filtro")
    print("4)Grafico")
    print("5)Salir")
    print("-----")
    print("Ingrese una opcion para continuar")
    opcion=int(input())
    if opcion==1:
        #cargar archivos
        print("cargar archivo")
        cargarTexto(movieList)
        print("CARGO EXITOSAMENTE\n")
        pass
    elif opcion==2:
        while opGestion!=3:
            print("-----")
            print("Gestion de peliculas")
            print("1)Mostrar Películas")
            print("2)Mostrar Actores")
            print("3)Regresar")
            print("Ingrese una opcion para continuar")
            opGestion=int(input())
            if opGestion==1:
                for i in movieList:
                    i.showInfo()
                break

```

En la clase principal, se utilizó un while para que mostrara el menú y este siguiera apareciendo para mostrar el resto de opciones

En el if de la opcion==1 se utilizó el método para ingresar la ruta.

En el if de la opción==2, se mostró el menú con prints para que mostrara las películas y los actores, esto con métodos independientes para cada opción

```

elif opcion==3:
    #filtrar datos
    while opFiltro!=3:
        print("-----")
        print("1)Filtrado por actor")
        print("2)Filtrado por año")
        print("3)Filtrado por genero")
        print("4)Regresar")
        print("-----")
        print("ingrese una opcion para continuar")
        opFiltro=int(input)
        if opFiltro==1:
            pass
        if opFiltro==2:
            pass
        if opFiltro==3:
            pass
        if opFiltro==4:
            pass
        pass##pass de el while

    input()#input que espera una instruccion
    pass #pass de elif
elif opcion==4:
    grafica(movieList)

#graficar
input()

```

En el if de la opción==3 se utilizó para mostrar el filtrado de los actores, por nombre de película, por año o por genero

Y por ultimo, en el if==4 se agregó el método que grafica el texto.

cargaTexto.py

```

1 class movie:
2     def __init__(self,name, actor, year, genre):
3         self.name=name
4         self.actor=actor
5         self.year=year
6         self.genre=genre
7
8     def showInfo(self):
9         print("\nNOMBRE PELICULA: ",self.name, "\nAUTORES: ", self.actor,"\nAÑO ESTRENO: ", self.year,"\nGENERO: ", self.genre)
10
11     def showPeliculasEnum(self):
12         print("\nNOMBRE PELICULA: ",self.name)
13
14     def showActoresPeli(self):
15         print("AUTORES: ", self.actor)
16

```

En el `__init__` se inicializó con los datos que se trabajaron, como el nombre, actor, año y género.

El método `showInfo(self)` se utilizó para mostrar todos los datos, posteriormente, esta clase se utilizó en la gestión de películas.

El método `showPeliculasEnum` se utilizó para mostrar las películas de manera enumerada, para la gestión de películas.

El método `showActoresPeli`, se utilizó para mostrar a los actores que actuaban en su película correspondiente.

```

65 def cargarTexto(list):
66     ruta=input("Ingrese la ruta del .lfp a leer: ")
67     textoLfp=open(ruta, "r")
68     lines= textoLfp.readlines()
69
70     for i in lines:
71         i=i.split(";")
72         contador=1
73         tempo_name=None
74         tempo_actor=None
75         tempo_year=None
76         tempo_genre=None
77
78         for j in i:
79             if contador==1:
80                 tempo_name=j
81             elif contador==2:
82                 j=j.split(",")
83                 tempo_actor=j
84             elif contador==3:
85                 tempo_year=j
86             elif contador==4:
87                 tempo_genre=j
88             contador+=1
89
90         cine=movie(tempo_name, tempo_actor, tempo_year, tempo_genre)
91         list.append(cine)
92
93

```

Para `cargarTexto`, al principio se solicitó la ruta para extraer el archivo de texto correspondiente. El `for` recorrió todo el texto separando las que tenían `;` y luego se utilizó

un for anidado que asignara el nombre, autores, año y genero a una lista, para que se almacene y luego se muestre, se establece en la variable cine. Por último, el append añade lo que se realizó a una lista.

```
ruta=None
def grafica(list):
    ruta=input("Ingrese la ruta del .lfp a leer: ")
    textoLfp=open(ruta, "r")
    lines= textoLfp.readlines()
    for i in lines:
        i=i.replace('\n', ' ')
        i=i.split(";")
        x={"Pelicula": i[0], "Autores": i[1], "Año": i[2], "Genero": i[3]}
        list.append(x)

    print(list)
    textoLfp.close
    textoDOT=open("imagen.dot", "w")
    textoDOT.write("digraph{ \n")
    textoDOT.write('rankdir=LR \n')
    textoDOT.write('node[shape=record, fontname="Arial Black", fontsize=16] \n')
    for i in list:
        textoDOT.write(i["Pelicula"]+'->'+i["Autores"]+"\n")
    textoDOT.write(" } \n" )
    textoDOT.close
```

Para la gráfica, se solicitó que se ingresara nuevamente la ruta. Luego en el for, se reemplazó los saltos de línea por un espacio en blanco, luego con el Split, se separaron los textos, luego en el append se añade lo que se hizo anteriormente.

Luego se escribieron las instrucciones, tales como el write para graficar con la ayuda de graphviz.

