

JUAN CARLOS GONZALEZ VALDEZ  
202110180 - LFP - B+

# Manual tecnico

"Gestor de peliculas"



# Indice del manual

## TEMAS DEL MANUAL

Guia del programa

Creacion del menu

Funciones del programa

Funcion de cada boton

Datos personales



```
resp_iter = self.  
statuses = {}  
for data in  
    status = Statu  
        status_id=  
    )  
statuses[statu  
  
return statuses
```

# Guia del programa

## OBJETIVO DEL PROGRAMA

Realizar la implementacion del programa para cumplir con los requerimientos del usuario: gestion de carga de archivos, admnistracion de datos, filtrar y graficar como funciones base.



# Creacion del menu

PARA LA CREACION DE ESTE MENU SE IMPLMENTO COMO BASE VARIOS PRINT PARA QUE LO MUESTRE EL SISTEMA Y LA FUNCION DEF LA CUAL NOS PERMITE DEFINIR UN OBJETO

EN BASE A ESTO SE AGREGO UNA VARIABLE QUE AL SELECCIONAR CADA OPCION ESTE MISMO LO REDIRECCION A LA OPCION AGREGADA DEL MENU PRINCIPAL (CREAR UN TIPO BOTON), SE IMPLEMENTO UN WHILE PARA QUE ESTE SOLO ACEPTE DATOS QUE EXISTAN Y AL INGRESAR LOS DATOS SE REGRESE AL MISMO MENU.

SE LE AGREGO UN ELSE PARA QUE AL SELECCIONAR UN NUMERO NO VALIDO LO MARQUE Y REGRESE AL MENU JUNTO A UN INPUT PARA QUE EL POGRAMA SE INICIE UNICAMENTE CUANDO UNO LO PRESIONE.  
(A CADA OPCION DEL MENU SE LE AGREGO SU BOTON Y CONFIGURACION DE CLASE PARA QUE FUNCIONE)

```
#Menu principal
def print_menu():
    print("Menu de opciones:")
    print("1. Cargar archivo de entrada")
    print("2. Gestionar películas")
    print("3. Filtrado")
    print("4. Gráfica")
    print("5. Salir")

def main():
    print("JUAN CARLOS GONZALEZ VALDEZ - 202110180")
    print("LFP (Lenguajes Formales de programacion) SECCION B+")
    print("")
    print_menu()
    print("")

movies = []
#Bucle del menu
while True:
    choice = int(input("Elija una opcion (1-5): "))

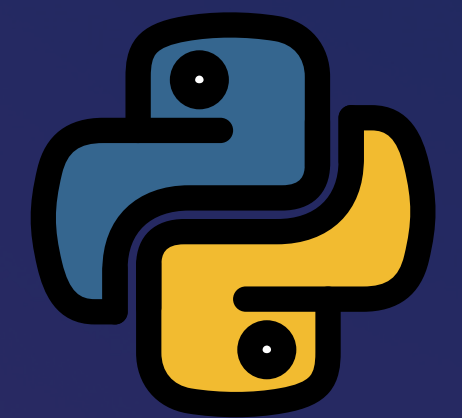
    if choice == 1:
        file_name = input("Ingrese el nombre del archivo a cargar: ")
        movies = load_movies(file_name)
        print(f"Se cargaron {len(movies)} películas desde el archivo {file_name}")
        print("")
        print_menu()
        print("")

    elif choice == 2:
        option2(movies)
        print("")
        print_menu()
        print("")

    elif choice == 3:
        option3(movies)
        print("")
        print_menu()
        print("")

    elif choice == 4:
        option4(movies)
        print("")
        print_menu()
        print("")

    elif choice == 5:
        print("")
        print("Saliendo del programa...")
        print("Gracias por usar el programa, vuelva pronto :3")
        print("")
        break
```



py

LFP - B+ - Juan Carlos Gonzalez

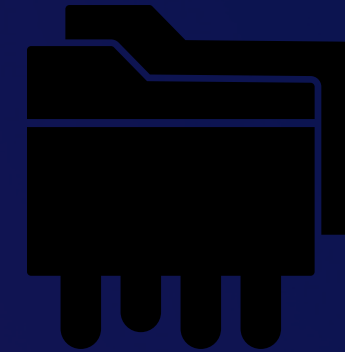
# Funciones del programa

ESTAS SON LAS  
IMPLEMNTACIONES DEL  
PROGRAMA

CARGAR ARCHIVOS DE ENTRADA



GESTIONAR PELICULAS

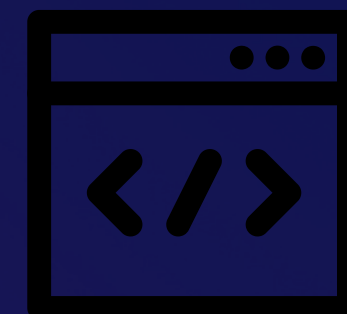


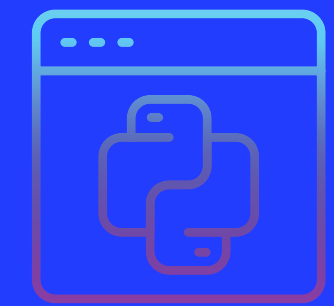
FILTRADO



GRAFICA

SALIR





# Funcion de la carga de archivo

## AGREGADO

Lo primero que se realizo fue un objeto con una clase para poder empezar a definir como moveriamos todo y seria la funcion de la carga, esta clase fue load\_movies, y creamos una sub clase de la misma llamada movies donde estaria definida la forma de lectura y la ruta el mismo y el funcionamiento el cual separaria el formato de lectura.

Luego de haber creado la funcion de lectura implemtamos el boton 1, que seria donde se cargarian los archivos del mismo programa, gracias a la variable y la ruta que se agregue, para este caso implmente que se argue la ruta manual para asi no dejar una precarga en el codigo y que ese mismo se ejecute.

Al este funcionar muestra la cantidad de archivos que pudo agregar el programa.

```
def load_movies(file_name):
    global loaded_movies # usar la variable global
    movies = []
    with open(file_name, 'r') as file:
        for line in file:
            line = line.strip()
            if line:
                name, actor, year, genre = line.split(';')
                movie = {
                    'nombre': name,
                    'actor': actor,
                    'anio': int(year),
                    'genero': genre
                }
                movies.append(movie)
    loaded_movies.extend(movies) # agregar las peliculas a la lista global
    return movies
```

```
def option1():
    file_name = input("Ingrese la ruta del archivo a cargar: ")
    movies = load_movies(file_name)
    print(f"Se cargaron {len(movies)} películas desde el archivo {file_name}.")
    input("Presione Enter para regresar al menú principal.") # esperar la entr
```



# Gestionar peliculas

## MOSTRAR PELICULA

Todo esto se trabajo en retorno a la variable movies que es donde se almacena nuestro dato, contatenamos todo en la variable movie para que estos sean los datos generales, al implementar este solo hacemos que llame al primer dato almacenado sacandolo de la variable movies mandando a que llame nombre.

## MOSTRAR ACTORES

En este caso es el mismo metodo con la unica diferencia que aqui lo llamamos con un set ya que son varios articulos y lo definimos de la misma manera con la varuable actor.



En este apartado el requerimiento es sobre mostrar peliculas y mostrar actores.

```
69 def option2(movies):
70     print("")
71     print("Gestion de películas")
72     print("")
73     print("1.Mostrar películas ")
74     print("2.Mostrar actores")
75     print("3.Regresar al menu")
76     print("")
77     choice = int(input("Elija una opción (1-3): "))
78     print("")
79     if choice == 1:
80         print("Lista de películas:")
81         for movie in movies:
82             print(movie['nombre'])
83     elif choice == 2:
84         actors = set()
85         for movie in movies:
86             actors.update(movie['actor'])
87         print("Lista de actores:")
88         for actor in actors:
89             print(actor)
90     elif choice == 3:
91         print("Regresando al menú principal...")
92         print("")
93         print("")
94         print("Se encuentra en el menu principal")
95         return
96     else:
97         print("Opción inválida. Por favor, elija una opción válida (1-3).")
98
```

# Funcion de filtrado

Esta misma estaba dividida por 3 categorías, actor, año y género para ser filtrados

Filtrado por actor:

Para la creación de el filtrado por actor se creo una variable llamada `filtered_movies` en la que concatenaba todos los datos para realizar un filtrado y al final se lee el ultimo dato el cual es actor esto hace que a la hora de escribir el dato buscado y este en el sistema se agregue un `len` para concatenar mas de uno y asi muestre todo los datos similares de la misma variable.

Para el filtrado por año y género se realizo el mismo sistema solo cambiando que en año se agrego un contador y se igualo para poder contarse de manera correcta y el género se quedo de la misma manera solo cambiando su lectura.

```
#Menu del filtrado
def option3(movies):
    print("Filtrado")
    print("")
    print("1. Filtrar por actor")
    print("2. Filtrar por año")
    print("3. Filtrar por género")
    print("4. Regresar al menú principal")
    choice = int(input("Elija una opción (1-4): "))
    #Concatenacion de datos y separado
    #Nombre
    if choice == 1:
        actor = input("Ingrese el nombre del actor: ")
        filtered_movies = [movie for movie in movies if actor in movie['actor']]
        print(f"Se encontraron {len(filtered_movies)} películas con el actor {actor}:")
        for movie in filtered_movies:
            print(f"{movie['nombre']} ({movie['anio']}) - {movie['genero']}")

    #Año
    elif choice == 2:
        year = int(input("Ingrese el año: "))
        filtered_movies = [movie for movie in movies if movie['anio'] == year]
        print(f"Se encontraron {len(filtered_movies)} películas del año {year}:")
        for movie in filtered_movies:
            print(f"{movie['nombre']} ({movie['actor']}) - {movie['genero']}")

    #Genero
    elif choice == 3:
        genre = input("Ingrese el género: ")
        filtered_movies = [movie for movie in movies if genre in movie['genero']]
        print(f"Se encontraron {len(filtered_movies)} películas del género {genre}:")
        for movie in filtered_movies:
            print(f"{movie['nombre']} ({movie['actor']}, {movie['anio']})")

    #Regreso al menu
    elif choice == 4:
        print("Regresando al menu principal...")
        print("")
        print("Se encuentra en el menu principal")
        return

    else:
        print("Opcion invalida. Por favor, elija una opción valida (1-4).")
```



```
import graphviz

def generate_graph(movies):
    dot = graphviz.Digraph(comment='Películas')
    actors = {} # Almacenar actores ya en el grafico
    for movie in movies:
        actor = movie['actor']
        if actor not in actors:
            # si el actor no ha sido agregado al grafico, crear un nuevo nodo para el actor
            dot.node(actor, actor, shape='box')
            actors[actor] = [movie] # agregar actor al diccionario con la lista de peliculas
        else:
            # si el actor ya esta en el grafo, agregar la pelicula a la lista de peliculas del actor
            actors[actor].append(movie)

    for actor, movies in actors.items():
        # crear un subgrafo para cada actor y sus peliculas
        actor_node = graphviz.Digraph(name='actor_' + actor)
        for movie in movies:
            # agregar un nodo para cada pelicula con su informacion de anio y genero
            actor_node.node(movie['nombre'], label=movie['nombre'] + '\n' + str(movie['anio']) + '\n' + movie['genero'], shape='oval')
        # agregar un nodo para el actor
        actor_node.node(actor, label=actor, shape='box')
        # conectar los nodos de las peliculas correspondientes con el nodo del actor
        for movie in movies:
            actor_node.edge(actor, movie['nombre'], dir='back')
        # agregar el subgrafo de las peliculas al subgrafo del actor
        dot.subgraph(actor_node)

    dot.render('movies.gv', view=True)
```

# Grafica

## GRAFICAR

En este apartado se busca graficar concorde a los 2 tipos de nodos, nodo 1 que es pelicula, anio y genero y nodo 2 que serian los actores.

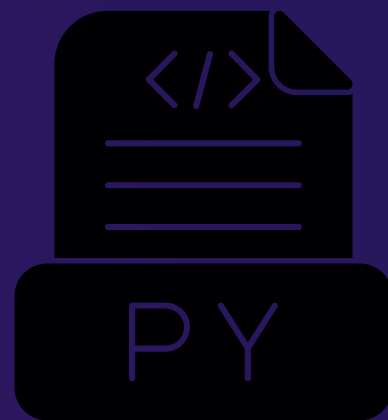
En esta se le hizo la agregacion de los dos para que amnos se guien de la otra, eso quiere decir una flecha a cada una, al agregar la funcion de subgrafos indepentes esto hace que se unan entre si y la funcion de dot.render hace que se exporte de manera inmediata por gv asi al no mas estar cargado los datos y presionar graficar la tira por defecto.



# Salir

## SALIR

Esta funcion funciona de manera casi unica ya que hace que el programa se detenga de manera inmediata, al ser implementado el break del ciclo while hace que el mismo no siga repitiendo las funciones y redireccionando a la variable main para retornar un menu.



```
#Funcion para salir
def option5():
    print("Gracias por usar el programa")
```

```
elif choice == 5:
    print("")
    print("Saliendo del programa...")
    print("Gracias por usar el programa, vuelva pronto :3")
    print("")
    break
```



# Este es un manual tecnico

EL MISMO PUEDE SER  
PROPENSO A PEQUENIOS  
CAMBIOS O MEJORAS DEL  
MIMSO EN BASE A LOS  
REQUISITOS DEL CLIENTE  
O CAMBIAS GRAFICOS.



NOMBRE

Juan Carlos Gonzalez Valdez

CLASE

Lenguajes formales de la  
programacion

SECCION

B+