Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Laboratorio de Lenguajes Formales y de Programación

Manual Técnico

Practica 1: Películas

Catedrático: Mario Josué Solis Solórzano

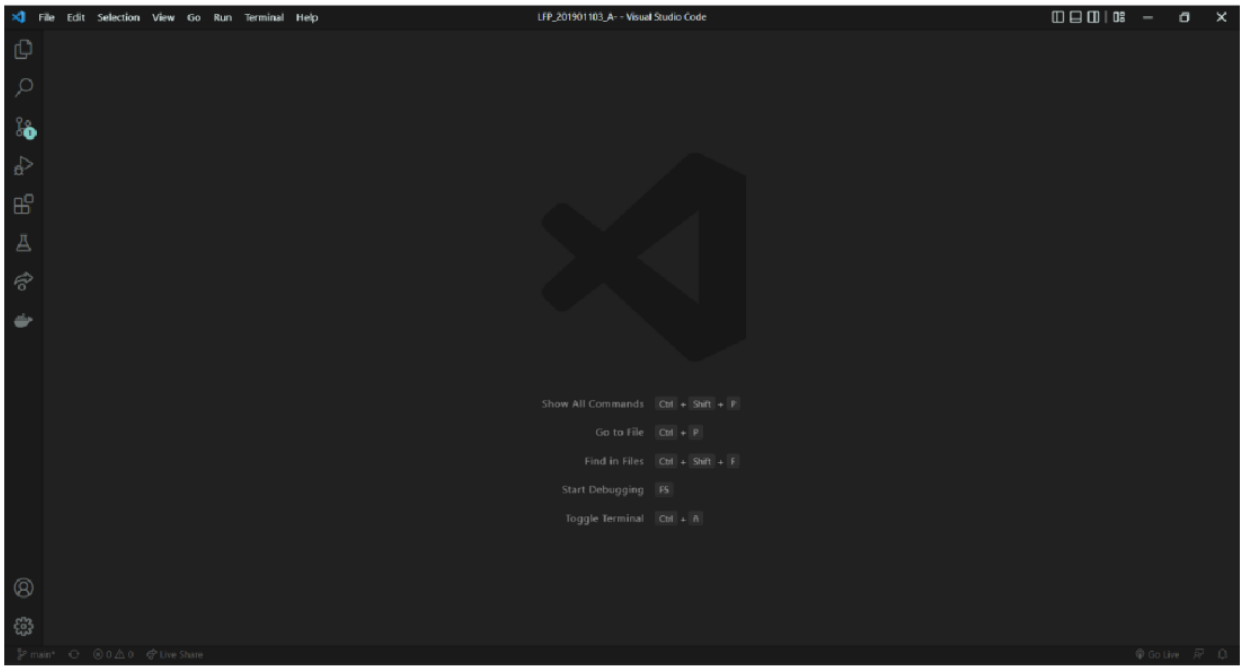
Sección: A-

Nombre: Josué Daniel Rojché García

Carné: 201901103

Requerimientos del Sistema

En la realización del software el Editor de texto que se utilizó fue Visual Studio Code.



El programa fue realizado con la versión de Python 3.11.1 para sistemas operativos de 64 bits.

El programa fue realizado con la versión de Python 3.11.1 para sistemas operativos de 64 bits.

Películas

Para que el programa funcione correctamente se realizaron diversas validaciones y manejo de errores para que no finalice indebidamente durante la ejecución de esta. Por lo cual explicaremos lo que realiza el código. Además utilizamos el paradigma Orientado a Objetos (POO) para el manejo de los datos.

Primero se importan las librerías necesarias, las cuales se utilizarán más adelante, por ejemplo: la librería graphviz, la cual servirá para la opción de graficar.

```
1  import os
2  from pelicula import Pelicula
3  import msvcrt
4  import graphviz
5
```

Menú Principal

Para mostrar el menú principal se crea el main, el cual será el que se ejecute inicialmente, y comenzará por mostrar la información del desarrollador para lo cual se utiliza print().

```
253  #   ----------------------------------------metodo main
254  if __name__ == '__main__':
255      print("\n*****************************************")
256      print("       Información del desarrollador")
257      print("Lenguajes Formales y de Programación \n"+"Sección: A-\n" + "Carné: 201901103\n"+"Josué Daniel Rojché García")
258      print("*****************************************\n")
```

Después se imprimirá el mensaje que indica que se debe presionar una tecla para continuar, por lo cual es necesario utilizar la librería msvcrt y llamar al método getch().

```
261      print("-----Presione una tecla para continuar-----")
262      #presionar = str(input())
263      msvcrt.getch()
```

Para que el menú siga apareciendo después de haber completado una opción se crea un bucle while valuado en True, para que así permanezca ejecutándose hasta que el usuario seleccione la opción de salir.

Dentro del bucle while se utiliza try except para capturar el error que el usuario pueda cometer al ingresar un valor alfanumérico en vez de solo el numero que corresponda a alguna de las opciones del menú. Además, cuando ocurra el error se mostrará un mensaje y la ejecución no será interrumpida.

Dentro del try se imprimen con print las opciones del menú, y con input podemos pedirle al usuario que ingrese la opción, esta opción será convertida a un valor entero, ya que son solo números los que debe ingresar el usuario, también se valida que el usuario ingrese una opción que no esté entre las opciones validas del menú por lo cual se muestra un mensaje indicando que debe ingresar una opción correcta.

Dentro de cada if ó elif se ingresa el método al cual será accedido para cada acción a realizar dependiendo de la opción seleccionada por el usuario, además se valida con otro if anidado que cada vez que el usuario intente ingresar a una opción sin haber utilizado la opción de cargar archivo previamente, se genere un error que indique que se debe cargar el mismo antes de seleccionar cualquier otra opción.

```
265     while True:
266         try:
267             print("\n------- Menu Principal --------")
268             print("1. Cargar archivo de entrada"+"\n2. Gestionar peliculas"+"\n3. Filtrado"+"\n4. Gráfica"+"\n5. Salir")
269             # recibe la opcion ingresada y la guarda como entero
270             opcion = int(input("Ingrese el número de la opción: "))
271             print()
272             if opcion == 1:
273                 cargarArchivo()
274             elif opcion == 2:
275                 if len(objetoPelicula) != 0:
276                     menuGestionar()
277                 else:
278                     print("Por favor carge el archivo de peliculas primero")
279             elif opcion == 3:
280                 if len(objetoPelicula) != 0:
281                     menuFiltrado()
282                 else:
283                     print("Por favor carge el archivo de peliculas primero")
284             elif opcion == 4:
285                 if len(objetoPelicula) != 0:
286                     graficar()
287                 else:
288                     print("Por favor carge el archivo de peliculas primero")
289             elif opcion == 5:
290                 print("Gracias por utilizar el programa \n")
291                 break
292             else:
293                 print("Ingrese una opcion correcta\n")
294         except ValueError:
295             print("\nPor favor ingrese solo numeros")
```

Método cargar archivo

Dentro de este método creamos una lista donde se almacenará cada línea del archivo a leer, luego utilizamos la validación de try except para capturar cualquier error que pueda ocurrir al leer el archivo, si ocurre un error se mostrará un mensaje correspondiente al mismo.

Dentro del try imprimimos un mensaje indicando que el usuario debe ingresar la ruta del archivo a leer utilizando input(), la cual se almacenara en la variable ruta y será enviada al método open, donde también se indicara que será un archivo de solo lectura con el parámetro “r” y el parámetro encoding, el cual indicará que leerá cualquier carácter especial. Todo lo que lea con el método open será almacenado en la variable.

Luego se utilizará validación if para indicar que si el archivo es igual a nulo es porque no se ha ingresado un archivo, de lo contrario se crea otra variable donde se almacenara todo el contenido del archivo, después se crea un bucle for para leer cada línea que ha almacenado en la variable texto, luego que se realiza la validación se agrega cada línea a la lista creada anteriormente y al finalizar el bucle se cerrará el archivo que se leyó, y para finalizar se muestra un mensaje que el archivo se ha leído correctamente.

```python
#   ----------------------------------------metodos del menu principal
def cargarArchivo():
    listaLinea = []
    try:
        print("\nIngrese la ruta del archivo:")
        ruta = str(input())
        leerArchivo = open(ruta, "r", encoding="utf8")
        if leerArchivo is None: # Valida si no se seleccionó un archivo
            print("No ha ingresado un archivo")
        else:
            texto = leerArchivo.readlines() # si el archivo existe lee todo el contenido del archivo
            for linea in texto:
                if linea.strip() == "":  # si una linea está vacia la omite
                    pass
                else: # guarda cada linea que no esté vacía
                    listaLinea.append(linea.strip("\n"))
            leerArchivo.close()
            print("\nArchivo leído correctamente")
    except IOError:
        print("\nPor favor ingrese una ruta de archivo valida")
```

Creamos la clase Persona y su respectivo método constructor que permitirá en ingreso de los parámetros necesarios para la creación del objeto, también se agregan los métodos setters para almacenar los datos, y los métodos getters que servirán para obtener los valores almacenados.

pelicula.py U ×

Practica1 > pelicula.py > Pelicula > getYear

```python
class Pelicula:
    def __init__(self, nombre,actores, year, genero):
        self.nombrePelicula = nombre #string
        self.actores = actores # lista de nombres string
        self.year = year # numero
        self.generoPelicula = genero # string

    def setNombrePelicula(self, nombrePelicula):
        self.nombrePelicula = nombrePelicula

    def getNombrePelicula(self):
        return self.nombrePelicula

    def setYear(self, year):
        self.year = year

    def getYear(self):
        return self.year

    def setGeneroPelicula(self,genero):
        self.generoPelicula = genero

    def getGeneroPelicula(self):
        return self.generoPelicula
```

Después se crea un bucle for para recorrer la lista donde se almacenó cada línea del archivo leído, y se obtiene en una variable temporal cada una de las líneas, después se utiliza el método Split para guardar los datos en una lista temporal, los datos están separados por punto y coma “;”, luego almacenamos cada valor en otras variables temporales de acuerdo a la posición en la que se almacenaron, luego volvemos a utilizar split para separar en una lista a los actores que se encuentran separados por comas “,”, estos datos se envían al objeto Película y se agregan a la lista de películas que está almacenada globalmente.

```
139     for i in listaLinea: #recorremos la lista de las lineas leídas en el archivo
140         #separar los campos por los punto y comas
141         textoTemporal = str(i)
142         temp = textoTemporal.split(";")
143         nombre_pelicula = str(temp[0])
144         actores_pelicula = str(temp[1])
145         year_pelicula = int(temp[2])
146         genero_pelicula = str(temp[3])
147         #separar los nombres de los actores en una lista de actores
148         actoresLista = actores_pelicula.split(",")
149         #enviar los datos al objeto
150         peli = Pelicula(nombre_pelicula.strip(),actoresLista,year_pelicula,genero_pelicula.strip())
151         objetoPelicula.append(peli)
```

Lista de objetos de películas creada globalmente para poder acceder desde cualquier método.

```
5
6   objetoPelicula = []
```

Se crea un contador que servirá para recorrer la lista de objetos de películas, creamos un bucle while que permitirá recorrer con el contador hasta que llegue al final de la lista de objetos de películas, creamos dentro del bucle una nueva variable de lista que servirá para guardar las posiciones de los datos que se encuentran repetidos en la lista, luego con un bucle for recorremos la lista de objetos de películas y validamos si el nombre de la película actual es igual a la película siguiente en la lista, si es igual agregamos a la lista que almacena las posiciones de los datos repetidos el numero correspondiente, de no repetirse el dato simplemente no hacemos nada, al finalizar el recorrido del bucle for continuamos con la creación de una variable que almacena la longitud de la lista de posiciones de los datos repetidos. Luego, creamos un if para validar que la longitud deba ser mayor o igual a 2 datos, si es correcto creamos una variable contador que servirá para recorrer con un bucle while las primeras posiciones que se repiten de los datos, y así proceder a eliminarlas de la lista y dejar almacenado solamente el ultimo dato que se repetía en la lista. Después solamente se muestra un mensaje indicando que los archivos repetidos fueron eliminados correctamente.

```python

    #Buscar los datos que se repiten y eliminarlos
    count = 0
    while count < len(objetoPelicula):
        numeroPosiciones = [] #lista que almacena las posiciones de los datos repetidos
        l = objetoPelicula[count]
        eliminarPeliculaRepetida = l.getNombrePelicula()
        iteratorEliminar =0
        for r in objetoPelicula:
            eliminarPeliculaRepetidaAuxiliar= r.getNombrePelicula()
            if eliminarPeliculaRepetida ==eliminarPeliculaRepetidaAuxiliar:
                numeroPosiciones.append(iteratorEliminar)
            else:
                pass
            iteratorEliminar +=1
        longiturIteraciones = len(numeroPosiciones)
        if longiturIteraciones >=2:
            count1 =0
            while count1 <= (longiturIteraciones-2):
                objetoPelicula.pop(int(numeroPosiciones[count1])-count1)
                count1 +=1
            count -=1
            print("\nLos datos repetidos fueron eliminados correctamente")
        else:
            pass
        count +=1
```

Menú Gestionar

Se crea un bucle while valuado con True para que no finalice la ejecución del menú hasta que el usuario ingrese la opción de regresar al menú principal.

Utilizamos el try except para capturar el error que pueda cometer el usuario de ingresar incorrectamente una letra o palabra, ya que solo se aceptan números que correspondan a cada opción, además si el usuario ingresa correctamente un numero pero no se encuentra dentro de las opciones correspondientes, se muestra un mensaje indicando que se debe ingresar una opción correcta. Para ingresar una opción se crea una variable que almacene lo ingresado con el método input, luego se valida cada opción por medio del if y elif, por lo cual cuando sea la opción que el usuario quiera realizar se accederá al método que corresponda.

```python
    def menuGestionar():
        while True:
            try:
                print("\n------- Menu Gestionar Peliculas --------")
                print("1. Mostrar peliculas"+"\n2. Mostrar actores"+"\n3. Regresar")
                # recibe la opcion ingresada y la guarda como entero
                opcion1 = int(input("Ingrese una opcion: "))
                print()
                if opcion1 == 1:
                    mostrarPeliculas()
                elif opcion1 == 2:
                    mostrarActores()
                elif opcion1 == 3:
                    break
                else:
                    print("Ingrese una opcion correcta")
            except ValueError:
                print("\nPor favor ingrese solo numeros")
```

Método mostrar películas

Se imprime con print la estructura que tendrá la tabla que visualizara el usuario, después se crea un contador que enumerará cada película que se muestre en pantalla, con el for recorremos la lista de las películas para así obtener cada uno de los datos a mostrar con el método print, luego se aumenta en una unidad el contador.

```
# ----------------------------------------metodos del menu gestionar peliculas
def mostrarPeliculas():
    #Recorrer los listados
    print("********** Lista de Peliculas**********\n")
    print("No.  | Año     | Genero    --> Pelicula")
    print("-------------------------------------------------")
    no_numero = 1
    for j in objetoPelicula:
        print(str(no_numero) + "    | " + str(j.getYear())+"    | " + j.getGeneroPelicula()+"    --> " + j.getNombrePelicula())
        no_numero +=1

```

Método mostrar actores

Inicia mostrando la estructura del listado de películas con el print, luego creamos un try para capturar el error de ingresar cualquier carácter que no sea un número, luego realizamos una variable contador que nos servirá para la enumeración de las películas, por lo cual se mostrará en pantalla del usuario el numero y el nombre de la película con el método print, luego pedimos al usuario que ingrese el numero de la opción correspondiente a la pelicula y esta se ingresa con un input y se almacena en una variable como un numero entero, después se realiza la validación de un if indicando que la opción debe ser mayor que cero y menor que la cantidad de películas que se observan en la lista de películas, si la condición se cumple entonces ingresa al if y muestra el nombre de la película que el usuario ha elegido utilizando el método print, y dentro del if creamos un for para recorrer la lista de actores que está en la película elegida y se muestra en pantalla a esos actores con el método print.

```
def mostrarActores():
    print("********** Lista de Peliculas**********\n")
    print("No.  | Pelicula")
    print("-------------------------------------")
    try:
        no_numero1 = 1
        for k in objetoPelicula:
            print(str(no_numero1)+"    | " + k.getNombrePelicula())
            no_numero1 +=1
        print("\nElija el numero de la pelicula de la cual quiere ver los actores:")
        elejirMostrar = int(input())
        #mostrando los actores
        if (elejirMostrar > 0) and (elejirMostrar < (len(objetoPelicula)+1)):
            print("Pelicula elegida: " + objetoPelicula[elejirMostrar-1].getNombrePelicula())
            print("\n********** Lista de Actores**********")
            print("-------------------------------------")
            for ob in objetoPelicula[elejirMostrar-1].actores:
                print("\t · "+ob.strip())
        else:
            print("Por favor no ingrese un numero mayor o menor del que observa en la lista de Peliculas")
    except ValueError:
        print("\nPor favor ingrese solo numeros")

```

Menú Filtrado

Se crea un bucle while valuado con True para que no se finalice hasta que el usuario ingrese a la opción de regresar, luego se crea un try except para capturar el error de ingresar cualquier carácter que no sea un número, luego se muestra en pantalla las opciones que el usuario podrá elegir usando el método print, después se le pide al usuario que ingrese el numero de la opción por lo cual se utiliza input y se convierte lo ingresado a un numero entero que se almacena en la variable que será validada en el if para verificar que opción ha ingresado el usuario y si lo ingresado coincide se ejecutará el método que corresponde a la opción, de lo contrario mostrará mensaje de opción incorrecta.

```
208  def menuFiltrado():
209      while True:
210          try:
211              print("\n------- Menu Filtrado --------")
212              print("1. Filtrado por actor"+"\n2. Filtrado por año"+"\n3. Filtrado por genero"+"\n4. Regresar")
213              # recibe la opcion ingresada y la guarda como entero
214              opcion2 = int(input("Ingrese una opcion: "))
215              print()
216              if opcion2 == 1:
217                  filtradoActor()
218              elif opcion2 == 2:
219                  filtradoYear()
220              elif opcion2 == 3:
221                  filtradoGenero()
222              elif opcion2 == 4:
223                  break
224              else:
225                  print("Ingrese una opcion correcta")
226          except ValueError:
227              print("\nPor favor ingrese solo numeros")
228
```

Método filtrar actor

Se realiza un print para mostrar en que opción se encuentra el usuario, luego se crea un try except para capturar el error de ingresar números en vez de palabras, ya que en esta opción se utiliza un input el cual maneja el tipo de dato string, por lo cual se utiliza un if para realizar dicha validación, para ello utilizamos el método isdigit que devuelve un valor booleano, indicando que si lo que el usuario ingreso un numero devuelva True y si se cumple la condicional muestra un mensaje con print, de lo contrario cuando sea False podrá mostrar el nombre del actor que se está buscando, y el listado de películas en las que participa, para ello se crea un contador y un bucle while que sirve para recorrer la lista de películas almacenadas, luego utilizamos un contador auxiliar con un while que servirá para recorrer a la lista de los actores que está almacenada dentro de la lista de películas, dentro de estos bucles se crea un if que sirve para comparar si el actor que ingreso el usuario es igual a alguno de los actores que se encuentra almacenado en la lista, para ver si en realidad es igual también se quitan los espacios al final o al inicio que pudiese haber cuando se cargó el archivo por lo que se utiliza el método strip, y para verificar mejor la validación se pasan todas las letras a minúsculas con el método lower,

y si la comparación resulta ser correcta, entonces se utiliza print para mostrar el nombre de la película en la que participa el actor.

```
60  # ----------------------------------------metodos del menu filtrado
61  def filtradoActor():
62      print("********** Filtrar por Actor**********\n")
63      try:
64          print("Buscar:")
65          palabra2 = input()
66          if palabra2.isdigit() == True:
67              print("Por favor ingrese solo valores alfabeticos, no ingrese numeros")
68          else:
69              print("\nActor: "+ palabra2)
70              print("\tLista de Peliculas")
71              print("-----------------------------------")
72              #se tiene el actor a buscar
73              #recorrer la lista de actores de la pelicula actual
74              contador = 0
75              while contador < len(objetoPelicula):
76                  contadoraux = 0
77                  while contadoraux < len(objetoPelicula[contador].actores):
78                      if objetoPelicula[contador].actores[contadoraux].strip().lower()== palabra2.strip().lower():
79                          print("\t · "+ objetoPelicula[contador].getNombrePelicula())
80                      contadoraux +=1
81                  contador+=1
82      except TypeError:
83          print("\nPor favor ingrese solo palabras")
```

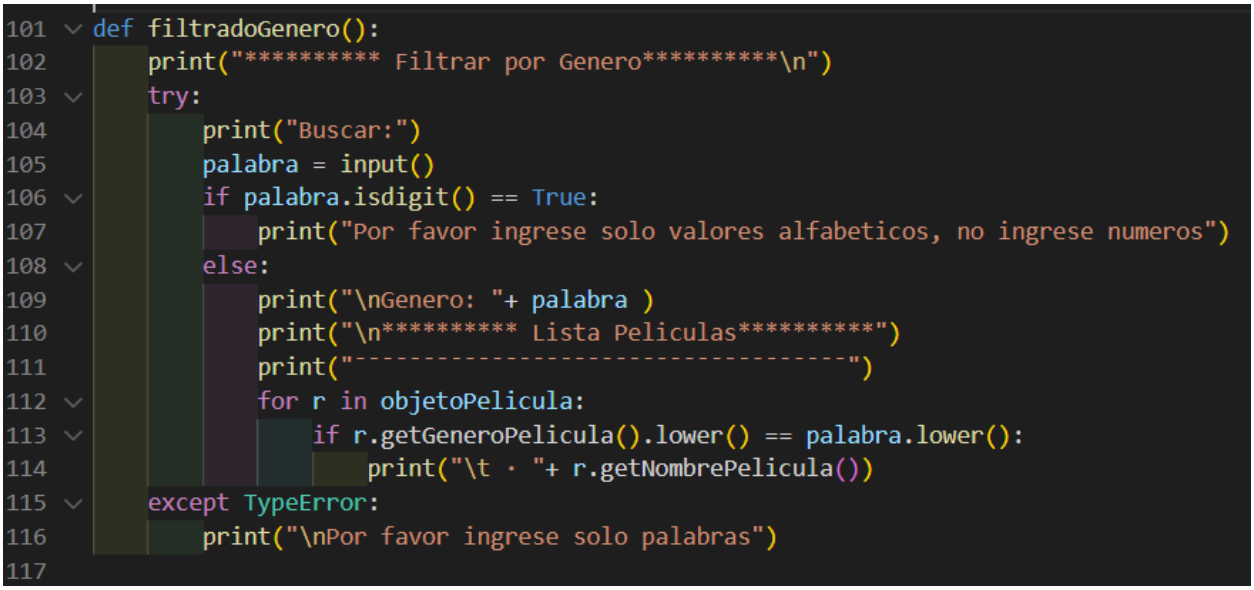
Método filtrar año

Imprime con print el nombre de la opción en la que se encuentra el usuario, luego se crea un try except para evitar el error de ingresar una letra en vez de algún número en el método input, luego se muestra cual fue el año ingresado por el usuario, y se crea un for para recorrer la lista de películas y dentro del mismo se realiza la condición if para verificar que el año ingresado por el usuario sea igual al año que se encuentra almacenado en la lista de películas, y si la validación se cumple se muestra el genero de la película y el nombre de la película utilizando print.

```
85  def filtradoYear():
86      print("********** Filtrar por Año**********\n")
87      try:
88          print("Buscar:")
89          palabra1 = int(input())
90          print("\nAño: "+ str(palabra1))
91          print("\n********** Lista Peliculas**********")
92          print("Genero    --> Pelicula")
93          print("-------------------------------------------------")
94
95          for r in objetoPelicula:
96              if r.getYear() == palabra1:
97                  print(r.getGeneroPelicula()+"    --> " + r.getNombrePelicula())
98      except ValueError:
99          print("\nPor favor ingrese solo numeros")
```

Método filtrar genero

Imprime con print el nombre de la opción en la que el usuario se encuentra, luego con try except se evita que el programa finalice cuando el usuario ingresa un numero en lugar de palabras, para ello se valida con un if y con el método isdigit que devuelve True cuando el dato ingresado en input por el usuario sea igual a un número, por lo que con print muestra un mensaje en pantalla indicando el error, luego si la condición no es válida se devuelve False y por lo tanto muestra en pantalla el género ingresado por el usuario y se crea un for que recorre la lista de películas, dentro del for se crea un if que compara lo ingresado por el usuario y lo buscado por el for, luego se imprime con el print el nombre de la película que cumple con la condición.



```
101 def filtradoGenero():
102     print("********** Filtrar por Genero**********\n")
103     try:
104         print("Buscar:")
105         palabra = input()
106         if palabra.isdigit() == True:
107             print("Por favor ingrese solo valores alfabeticos, no ingrese numeros")
108         else:
109             print("\nGenero: "+ palabra )
110             print("\n********** Lista Peliculas**********")
111             print("-----------------------------------")
112             for r in objetoPelicula:
113                 if r.getGeneroPelicula().lower() == palabra.lower():
114                     print("\t · "+ r.getNombrePelicula())
115     except TypeError:
116         print("\nPor favor ingrese solo palabras")
117
```

Método graficar

Se utiliza la librería graphviz para realizar la grafica donde se mostrará la tabla de las películas con sus respectivos géneros y años, y cada una relacionada hacia los actores, por lo tanto se crea una variable grafo que iniciará indicando que utilizaremos un digraph con el nombre que tendrá el archivo .dot, luego a ese grafo le indicamos el atributo con attr que tendrá todo el documento del gráfico y se le indica que la separación entre las tablas y los actores será de 2 pulgadas, el tamaño será de 8.5, y se mostrará de izquierda a derecha. Luego con un bucle for recorremos los datos de la lista de películas, en la cual indicamos nuevamente la variable de grafo que escribe el .dot y le indicamos el atributo que tendrá la tabla con attr para que la tabla se vea cuadrada y no tenga la forma por defecto del grafo, se indica que se aplicará al node, luego creamos en el node la tabla con estructura de html y ingresamos el nombre con el que será conocido el grafo, por lo cual se indica el nombre de la película, después de la tabla indicamos nuevamente que el attr tendrá valores vacíos para quitar la configuración que tenía la tabla, ahora se crea un for que recorrerá la lista de actores que se encuentra en cada una de las películas almacenadas en la lista principal, por lo tanto también se crea otros atributos con attr para darle configuración al grafo que mostrará a los actores, por lo que se utiliza un rectangle, y color que distinguirá a los mismos y estos se aplicarán al node, luego al edge se le indica el nombre del nodo que corresponde a la película actual y tendrá la relación hacia los diferentes actores, según corresponda, y a estos se le agrega una forma especifica de flecha con “vee”, luego volvemos a indicar con attr que las configuraciones estén en blanco para que no afecte las demás iteraciones del bucle, al finalizar se indica con view que guarde todo en .dot y lo convierta a pdf y se abra automáticamente dentro de la carpeta donde se indica el directorio.

```
229 def graficar():
230     grafo = graphviz.Digraph('tabla',filename ='Peliculas y Actores.dot')
231     grafo.attr(rankdir = 'LR',size='8,5', ranksep="2", bgcolor = "lightgoldenrodyellow", margin = "0.1")
232     nombreN = 1
233     for pe in objetoPelicula:
234         grafo.attr('node', shape= 'plaintext')
235         grafo.node(str(nombreN),'''<
236             <TABLE BORDER="0" CELLBORDER="1" CELLSPACING="0" ALIGN="CENTER">
237                 <TR>
238                     <TD COLSPAN="2" BGCOLOR="YELLOW"><b>'''+ pe.getNombrePelicula()+'''</b></TD>
239                 </TR>
240                 <TR>
241                     <TD>'''+str(pe.getYear())+'''</TD>
242                     <TD PORT="f1">'''+pe.getGeneroPelicula()+'''</TD>
243                 </TR>
244             </TABLE>>''')
245         #grafo.attr('node',style='', color='')
246         for ac in pe.actores:
247             grafo.attr('node', shape= 'rectangle', style="filled", color="black",fillcolor="lightsalmon")
248             grafo.edge(str(nombreN)+':e', ac.strip(), arrowhead = 'vee')
249             grafo.attr('node',style='', color='')
250         nombreN +=1
251     #grafo.save(filename= "tabla.dot", directory="./Practica1") #,directory="Practica1\tabla.dot"
252     #os.system("dot.exe -Tpdf Practica1/tabla.dot -o Practica1/peliculas.pdf")
253     grafo.view(filename ="Peliculas y Actores.dot" ,directory="./Practica1")
```