MANUAL TECNICO - PRACTICA 1

> Este programa se desarrollo con el lenguaje de programacion python en su version 3.10.6. haciendo uso de algunos modulos como csv, colorama y graphviz.

> Se utilizo la programacion orientada a objetos, para poder crear listas dinamicas (TDA) y tambien se utilizaron listas nativas de python para la manipulacion de informacion

> Este programa unicamente puede utilizarse mediante consola

MODOULO 1 - main

En este modulo se crearon e importaron los modulos y funciones necesarias para poder resolver la practica 1 de este curso. Lo primero que se encuentra en este modulo es la creacion de un objeto de tipo “ListaPeliculas” La funcion de este objeto sera gestionar y manipular todos los datos recolectados del archivo.

def main( )

En esta funcion se define un While el cual simulara lo que en otros programas llaman “do-while” esto para poder repetir el menu tantas veces como sea necesario.

Despues de esto se encuentra un manejo de excepciones en caso surga algun tipo de error y asi evitar que el programa se cierre

Se muestran las opciones del menu con prints con un poco de formato y luego se le pido mediante un input() al usuario que ingrese la opcion que desee realizar y mediante de if se compara ese valor y se ejecutan las instrucciones dentro.

```
while True:
        try:
            print(Fore.CYAN + '-----------------------------------------------------')
            print(Fore.CYAN + '|              BIENVEDIO AL PROGRAMA               |')
            print(Fore.CYAN + '|                                                  |')
            print(Fore.CYAN + '| Lenguajes Formales y de Programacion - Seccion A-|')
            print(Fore.CYAN + '| Jhonatan Alexander Aguilar Reyes     - 202106003 |')
            print(Fore.CYAN + '|                                                  |')
            print(Fore.CYAN + '-----------------------------------------------------')
            input(Fore.CYAN + '\nPresione ENTER para contuniar')

            while True:
                print(Fore.CYAN +"-----------------------------------------------------")
                print(Fore.CYAN +"|                    MENU PRINCIPAL               |")
                print(Fore.CYAN +"|                                                 |")
                print(Fore.CYAN +"|    [1]. Cargar archivo de entrada               |")
                print(Fore.CYAN +"|    [2]. Gestionar peliculas                     |")
                print(Fore.CYAN +"|    [3]. Filtrado                                |")
                print(Fore.CYAN +"|    [4]. Grafica                                 |")
                print(Fore.CYAN +"|    [5]. Salir                                   |")
                print(Fore.CYAN +"|                                                 |")
                print(Fore.CYAN +"-----------------------------------------------------")
                menu_principal_str = input(Fore.CYAN +"Ingrese un valor para seleccionar una opcion: ")
                menu_principal = int(menu_principal_str)    #Parseando opcion

                if menu_principal == 1:   #Opcion 1 del menu principal
                    CargaDeArchivo()    #funcion para cargar archivo

                if menu_principal == 2:     #Opcion 2 del menu principal
                    GestionadorPeliculas() #funcion para mostrar sub menu Gestionador

                if menu_principal == 3:     #Opcion tres del menu Principal
                    Filtros()               #funcion para mostrar sub menu Filtro
```

```
            if menu_principal == 4:     #Opcion 4 del menu principapal
                print(Fore.LIGHTMAGENTA_EX+"Procesando relaciones...")
                mi_peliculas.GraficaRelacion()

            if menu_principal == 5:        #if que simula el do-while
                print(Fore.RED+"Hasta la proxima :)")
                break
    except Exception as err:
        print(Fore.RED+"\n\tSe acaba de producir un error :( " + str(err))
        print(Fore.RED+"\tAsegurate de leer el MANUAL DE USUARIO")
    else:
        break
```

En caso el usuario ingrese la opcion numero 1 el programa llama a la funcion…

def CargaDeArchivo( )

En esta funcion se le pide al usuario que ingrese una direccion absoluta para poder ubicar el archivo, esta direccion se guarde en una varible que luego se le pasa a la funcion open() y se le indica que solo pueda leerlo.

Despues de eso se define un contador para contar la cantidad de filas que contiene el documento, esto mediante un for que recorre el archivo.

Seguido de esto se crea otro for pero el objetivo de este es obtener los datos del archivo y almacenarlos en variables temporales para luego pasarlas como parametro a la lista que se creo al inicio, este for dara iteraciones igual que el contador.

En caso de ocurrir algun tipo de error en la lectura del archivo o en los datos del archivo se mostraran los mensajes en el bloque “except” y en caso todo salga bien se ejecutara el bloque “else”.

```
try:
    print(Fore.LIGHTMAGENTA_EX+"\n\tUSTED SE ENCUENTRA EN LA OPOCION 1")
    ruta = input(Fore.LIGHTMAGENTA_EX+"-> Ingrese la direccion del archivo de entrada: "+Fore.CYAN)

    #abre el archivo modo escritura
    archivo = open(ruta, 'r')

    #contador de cuantas lineas tiene
    contadorlineas = 0
    for row in open(ruta) :
        contadorlineas += 1     #recuento de lineas

    for row in range(contadorlineas):
        lectura = archivo.readline().split(';')
        name = lectura[0]
        actor = lectura[1].split(',')
        lansamiento = lectura[2]
        categoria = lectura[3]

        #Rellenando la lista
        mi_peliculas.IncertarPelicula(str(name.strip()), actor, str(lansamiento.strip()),str(categoria.strip()))
except Exception as err:
    print(Fore.RED+"\n\tOcurrio un error con la carga del archivo :| del tipo: "+str(err))
    print(Fore.RED+"\tIntentalo de nuevo\n")
else:
    print(Fore.GREEN+"\n\tEl archivo se cargo exitosamente :) ")
```

Despues de ejecutar todo esto el programa mostrara de nuevo el menu principal gracias al while y el usuario podra elegir otra opcion, si el usuario selecciona la opcion 2 se llamala a la funcion…

def GestionadorPeliculas( )

Esta funcion contiene otro menu dentro de ella por lo mismo se implementa el mismo metodo para hacer que este se vuelva a mostrar cada que finalice de ejecutarse.

El usuario debera de elegir la opcion que desea en caso sea la opcion “ a ” el programa obtendra la cantidad de registros guardados propios de la clase de la lista, esto sera util para validar si ya se cargaron peliculas o no.

Despues de eso se imprime en pantalla el retorno de la funcion “ImprimirInfoPeliculas( )” propia del objeto tipo “ ListaPeliculas” la cual se explicara mas a delante su funcionamiento.

En caso el usuario seleccione la opcion 2 se llama tambien a una funcion del objeto y posterior se le pide al usuario un valor.

```
while True:
        print(Fore.CYAN +"\n-----------------------------------------------------")
        print(Fore.CYAN +"|             GESTIONADOR DE PELICULAS             |")
        print(Fore.CYAN +"|                                                  |")
        print(Fore.CYAN +"|              ¿Que desea realizar?                |")
        print(Fore.CYAN +"|    [a]. Ver las peliculas registrada             |")
        print(Fore.CYAN +"|    [b]. Ver los actores de las peliculas         |")
        print(Fore.CYAN +"|    [c]. Salir de este sub-menu                   |")
        print(Fore.CYAN +"|                                                  |")
        print(Fore.CYAN +"-----------------------------------------------------")
        menu_2 = input(Fore.CYAN+"Ingrese una opcion:  ")

        if menu_2 == "a":       #Opcion 1 del sebmenu2
            bandera = mi_peliculas.ContarNodos()
            if bandera != 0:
                print(Fore.LIGHTMAGENTA_EX+"\nlista de peliculas registradas: \n")
                print(Fore.YELLOW+ mi_peliculas.ImprimirInfoPelicula()) ##funcion imprimir peliculas
            else:
                print(mi_peliculas.ImprimirInfoPelicula())

        if menu_2 == "b":       #Opcion 2 del sebmenu2
            bandera = mi_peliculas.ContarNodos()
            if bandera != 0:
                print(Fore.LIGHTMAGENTA_EX+"\nlista de peliculas registradas: \n")
                print(Fore.YELLOW+ mi_peliculas.ImprimirPelicula())
                indice_Pelicula = input(Fore.YELLOW+"\nIngrese el numero de la pelicula de la que desea conocer sus actores: ")
                print(mi_peliculas.ImprimirActores(indice_Pelicula))
            else:
                print(mi_peliculas.ImprimirInfoPelicula())

        if menu_2 == "c":       #Opcion 3 del sebmenu2
            print(Fore.RED+"Nos vemos en el menu principal :)")
            break
```

Al momento que el usuario elija la tercera opcion el programa lo direcciona al menu principal donde se podra elejir otra opcion, en caso se elija la funcion 3 se llamra a la funcion…

def Filtros( )

Esta funcion tambien tiene un submenu donde el usuario podra elegir opciones y es muy similar a la funcion anterior, por cada opcion se llama a una funcion del objeto y se le pido al usuario que ingrese un valor el cual es util como identificador de datos

Modulo - Pelicula

En este modulo se definen dos clases, esto con el fin de crear nodos (datos abstractos) la primera clase con el nombre “Pelicula” se encarga de almacenar todos los datos, practicamente solo se asignan valores

```
class Pelicula:
    def __init__(self, id, nombre, actores, anio, genero):
        self.id = id
        self.nombre = nombre
        self.actores = actores
```

```
        self.anio = anio
        self.genero = genero
```

Posterior a esto se tiene la siguiente clase la cual creara los nodos con los datos, Esta funcion se encarga de recibir los datos desde una clase externa y los asigna a una variable y posteriormente se los pasa a la clase que se explico anteriormente.

En esta clase se encuentra todos los metodos gets y sets utilizados y algunos metodos para retornar datos especificos anidados con otros.

```
class NodoPelicula:
    def __init__(self, id, nombre, actores, anio, genero):
        self.Dato = Pelicula(id,nombre, actores, anio, genero)
        self.Siguiente = None

    def ObtenerId(self):
        return self.Dato.id

    def ObtenerSiguiente(self):
        return self.Siguiente

    def AsignarSiguiente(self, Nodo):
        self.Siguiente = Nodo

    def ObtenerPelicula(self):
        return self.Dato.nombre

    def ObtenerActor(self):
        actores = []
        for x in self.Dato.actores:
            actores.append(x.strip())
        return actores

    def ObtenerAnio(self):
        lanza = self.Dato.anio.strip()
        return lanza

    def ObtenerGenero(self):
        gene = self.Dato.genero.strip()
        return gene

    def prittInforMenu2a(self):
        return str(self.Dato.nombre)+ " - Fue estrenada en "+str(self.Dato.anio)+" - Se categorizo como: "+str(self.Dato.genero)

    def prittInfoMenu2b(self):
        acrto1 = ""
        for i in self.Dato.actores:
            acrto1 += " - "+i
        return str(acrto1)

    def BuscarActor(self, name):
        for i in self.Dato.actores:
            sin_espacion = i.strip()
            if name == sin_espacion:
                return str(sin_espacion)

    def prueba(self, name):
        for i in self.Dato.actores:
            sin_espacion = i.strip()
            if name == sin_espacion:
                return sin_espacion
```

Modulo - ListaPeliculas

En este modulo se crea una clase la cual se definieron los apuntadores de los nodos, un apuntador para el inicio de la cola y otro para el final, tambien se definio un contador para tener un recuento de la cantidad de datos que contiene la

lista.

```
def __init__(self):
        self.Inicio = None
        self.Final = None
        self.Limite = 1
```

def ContarNodos(self)

En este metodo de la clase se retorna la canditdad de nodos almacenados en la lista

```
def ContarNodos(self):
    return self.Limite
```

def IncertarPelicula(self, nombre, actores, anio, genero)

En esta funcion de la clase es donde se recolecta la informacion del archivo y se le envia a esta funcion y esta funcion crea un objeto con esta informacion y luego se la pasa a las clases que se explicaron anteriormente, despues se pasa por validaciones para ver si ya existe un nodo o la lista esta vacia, en caso lo este se le asigna el apuntador inicio y final al objeto creado en este caso “NuevoNodo” y en caso contrario ya existan nodos entonces se crea una variable auxiliar y se el asigna el puntero Inicio, este nos era util para recorrer los nodos almacenados en la lista, se crea un while para hacer y dentro de el una validacion, en donde se asegura que no se repira un parametro de los datos recolectados, en caso no se repita entonces se conecta el nuevo nodo a la lista y se hace que auxiliar avance una posicion de memoria para verificarlo con el siguiente nodo

```
def IncertarPelicula(self, nombre, actores, anio, genero):
        NuevoNodo = NodoPelicula(self.Limite, nombre, actores, anio, genero)
        if self.Inicio == None:
            self.Inicio = NuevoNodo
            self.Final = NuevoNodo
            self.Limite +=1
        else: #and self.Inicio.Siguiente != None
            numero = 0
            Actual = self.Inicio
            bandera = False
            while Actual != None:
                numero += 1
                if NuevoNodo.ObtenerPelicula() == Actual.ObtenerPelicula():
                    print(Fore.RED+str("\tLa pelicula "+str(numero)+" se encuentra repetida"))
                    bandera = True
                    Actual = Actual.Siguiente
                else:
                    Actual = Actual.Siguiente

            if bandera == False:
                self.Limite += 1
                self.Final.AsignarSiguiente(NuevoNodo)
                self.Final = NuevoNodo
```

def ImprimirInfoPelicula(self)

En esta funcion se hacen validaciones en caso no existan nodos retornar un mensaje de error y en caso contrario establecer un while como en la funcion anterior para poder recorrer la lista y obtener los datos que se necesitas

```
def ImprimirInfoPelicula(self):
        if self.Inicio == None:
            return Fore.RED+"""\n\tNo hay peliculas cargadas en el sistema
    Puede realizarlo en la opcion [1] del MENU PRINCIPAL"""
        c_pelicula = 0
        Retorno = ""
        Auxiliar = self.Inicio
        while Auxiliar != None:
            c_pelicula +=1
            Retorno += "-->La pelicula ["+str(c_pelicula)+"] es "+str(Auxiliar.prittInforMenu2a())
            if Auxiliar.Siguiente != None:
                Retorno += "\n"
            Auxiliar = Auxiliar.Siguiente
            Retorno += ""
        return Retorno
```

Las demas funciones se basan practicamente en lo mismo, se establece un ciclo con el que se puede iterar la lista de Nodos y validar dependiendo de lo que encesite el programa y unicamente retornamos los datos necesarios

def GraficoRelacion(self)

En esta funcion se crea una lista la cual nos servira para almacenar actores de todas las peliculas, pero sin que se repita

hacemos uso de nuevo del Auxiliar para recorrer la lista de nodos y dentro del while agregamos la informacion de cada iteracion en la lista.

En este punto ya se tienen todos los actores, pero repetidos para evitar esto se creo un for que iterara la lista y dentro un while el cual tendra como condicion que si su cuenta de repetidos es mayor a 1 entonces elimine ese item de la lista.

Una vez listo los actores se procede a graficarlos con las funciones del modulo graphviz, se crea el nombre del nodo y se crear un texto multilinea donde se define una tabla y gracias al formato de texto “f” se pueden introducir los datos de los actores.

Se realiza lo mismo para dibujar las peliculas con la diferencia que este ciclo es para la lista de Nodos.

Dentro de este ciclo de nodos se establece la relacion que tienen las peliculas con los actores, esto mediante otro for el cual recorrera la lista de actores y si el nodo actual tiene el mismo nombre que el de la lista de actores sin repetir entonces se procede a crear un vinculo con la funcion edge.

Por utilimo se llama a la funcion view( ) para poder generar todo lo que se creo en documento pdf

```
def GraficaRelacion(self):
        #Para recoger todo los autores en las peliculas
        total_actores = []
        Auxiliar = self.Inicio

        #Obtener todos los actores
        while Auxiliar != None:
            temp_actores = Auxiliar.ObtenerActor()
            for x in temp_actores:
                total_actores.append(x)
            Auxiliar = Auxiliar.Siguiente

        #Eliminando los actores repetidos
        for act in total_actores:
            while(total_actores.count(act) > 1):
                total_actores.remove(act)

        ##Para graficar los ACTORES Y RESISTENCIAS
        imgRelacion = graphviz.Digraph('Relacion', filename="ImagenRelacion")
        imgRelacion.attr(rankdir="LR")
```

```
        Auxiliar = self.Inicio
        #para generar los actores
        for actor in total_actores:
            imgRelacion.node(str(actor), f'''<
            <TABLE BORDER="0" CELLBORDER="1" CELLSPACING="0" CELLPADDING="6">
                <TR>
                    <TD COLSPAN="2"><FONT COLOR="black">{str(actor)}</FONT></TD>
                </TR>
            </TABLE>>''', shape="none", fillcolor="#33ceff", style = "filled")

        #Para generar todas las peliculas
        while Auxiliar != None:
            imgRelacion.node(str(Auxiliar.ObtenerId()), f'''<
            <TABLE BORDER="0" CELLBORDER="1" CELLSPACING="0" CELLPADDING="4">
                <TR>
                    <TD COLSPAN="2"><FONT COLOR="green">{str(Auxiliar.ObtenerPelicula())}</FONT></TD>
                </TR>
                <TR>
                    <TD><FONT COLOR="black">{Auxiliar.ObtenerAnio()}</FONT></TD>
                    <TD><FONT COLOR="black">{Auxiliar.ObtenerGenero()}</FONT></TD>
                </TR>
            </TABLE>>''', shape="box", fillcolor="#a52e0c", style = "")

            for conexion in total_actores:
                if conexion == Auxiliar.BuscarActor(conexion):
                    imgRelacion.edge(str(Auxiliar.ObtenerId()), str(conexion) )


            Auxiliar = Auxiliar.Siguiente

        imgRelacion.view()
```