

```
nt()
nt()
ion = int(input(Fore.BLUE+"Ingrese una opción: "+Fore.WHITE))
option == 1:
    i=0
    file = filedialog.askopenfilename(initialdir="C:/Users/sebas/Docum
",title="Elige un archivo de entrada", filetypes=(("Archivos de Dat
with open(file,"r") as archivo:
    lector = csv.reader(archivo, delimiter=";")
    for fila in lector:
        nombre = str(fila[0])
        actores = str(fila[1])
        año = int(fila[2])
        genero = str(fila[3])
        if i == 0:
            id = id+1
            movies.append(Pelicula(id,nombre,actores,año,genero))
            i=1
        else:
            varval=1
            for movie in movies:
                if movie.nombre == nombre:
                    print()
                    print(Fore.RED+"La película "+nombre+" se ha i
                    varval = 0
            if varval == 1:
                id = id+1
                movies.append(Pelicula(id,nombre,actores,año,genero))
```

# MANUAL / TÉCNICO

Geovanny Sebastián Herrera Claudio

# ANALIZADOR

Básicamente lo que hace el analizador es guardar todo el texto en un string desde el primer "{" hasta el primer "}" que encuentre, de esta manera encontrara los datos de una operación, así mismo con los corchetes, luego con un conjunto de if's y for's, esta programado para que pueda realizar las opciones que se indiquen en el texto cargado.

```
with open(filename,mode="r") as archivo:
    dot = graphviz.Digraph(comment="Example")
    inicial = 0
    text = ""
    for linea in archivo:
        if "{" in linea:
            text = text+linea+"\n"
        if "}" in linea:
            text = text+linea+"\n"
            if "Operacion:"+"Suma" in text:
                remove("operacion.txt")
                file = open("operacion.txt", "w")
                file.write(text)
                file.close()
                inicial=inicial+1
                var1="a"+str(inicial)
                dot.node(var1,"SUMA"+"\\n",{ "color":background,"fontcolor":font,"style":"filled","shape":form})
                digit = 1
                num = 0
                file = open("operacion.txt", "r")
                for valores in file:
                    const = "Valor'+str(digit)+'":'
                    if const in valores:
                        digit = digit+1
                        print(valores.rstrip())
                        texto = ""
                        if const+"[" in valores.replace(" ", ""):
                            for datos in file:
                                if "[" in datos:
                                    texto = texto+datos+"\n"
                                if "]" in datos:
                                    texto = texto+datos+"\n"
                                    remove("operacionNew.txt")
                                    archivo = open("operacionNew.txt", "w")
                                    archivo.write(texto)
                                    archivo.close()
                                    digit = 1
```

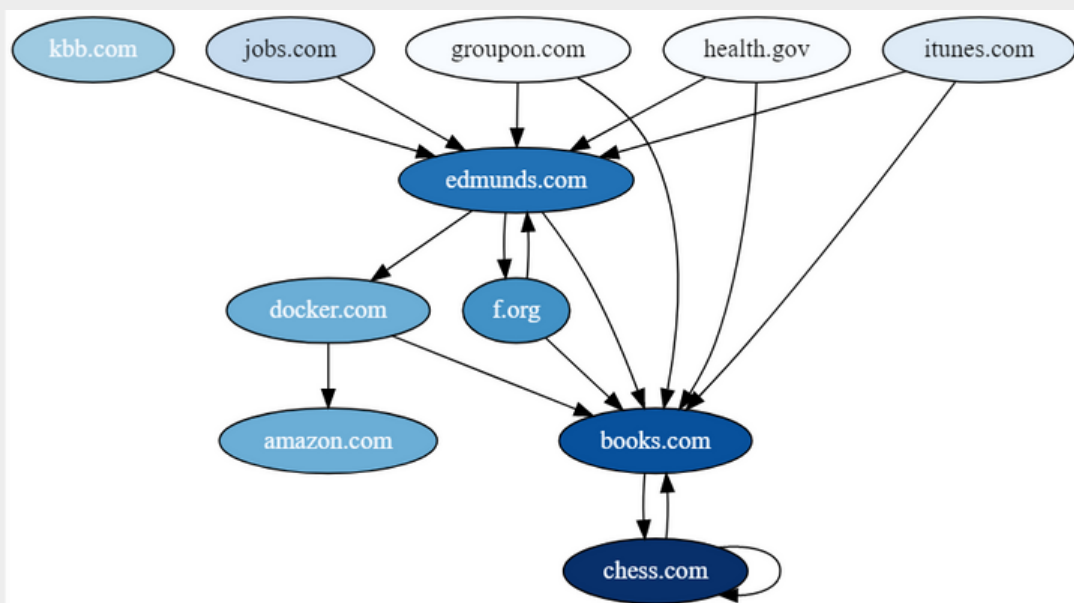
# ERRORES

Básicamente funciona igual que el analizador, aunque en este caso utiliza el conocimiento de un alfabeto de caracteres considerados errores, entonces con unos cuantos for's y un if es capaz de detectar que error sucedio, y en que línea del documento fue encontrado.

```
errores.py > errors
1 from os import remove
2
3 def errors(filename):
4     errores = ["#", "$", "%", "&", "(", ")", "*", "+", "/", "\\", "@", "!", ":", ";", "?", "\n"]
5     with open(filename, mode="r") as archivo:
6         text = ""
7         counter = 0
8         count = 0
9         text=text+"\n"
10        for linea in archivo:
11            counter=counter+1
12            for error in errores:
13                if error in linea:
14                    count = count+1
15                    text=text+"\nNo. ":" "+str(count)+"\n"+"Descripcion-Token":(linea+"Lexema": "error+"\n"+"Tipo":Error\n"+"Linea":'+str(counter)+'\n)\n"
16            text=text+"}"
17        try:
18            remove("ERRORES_202110588.txt")
19            file = open("ERRORES_202110588.txt", "w")
20            file.write(text)
21            file.close()
22        except:
23            file = open("ERRORES_202110588.txt", "w")
24            file.write(text)
25            file.close()
```

# GRAPHVIZ

Graphviz is open source graph visualization software. Graph visualization is a way of representing structural information as diagrams of abstract graphs and networks. It has important applications in networking, bioinformatics, software engineering, database and web design, machine learning, and in visual interfaces for other technical domains.



# WEBBROWSER

The webbrowser module provides a high-level interface to allow displaying web-based documents to users. Under most circumstances, simply calling the `open()` function from this module will do the right thing.

Esta librería se uso en la práctica para poder abrir un documento PDF.