**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.

Texto

Descripción generada automáticamente

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.

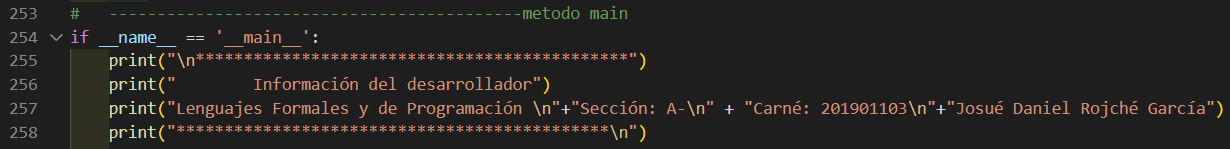
Primero se importan las librerías necesarias, las cuales se utilizarán más adelante, por ejemplo: la librería graphviz, la cual servirá para la opción de graficar.

Texto

Descripción generada automáticamente

**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().



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().

Texto

Descripción generada automáticamente

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.

**Texto

Descripción generada automáticamente**

**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.

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.

**Texto

Descripción generada automáticamente**

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

Interfaz de usuario gráfica

Descripción generada automáticamente

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.

**Texto

Descripción generada automáticamente**

**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.

**Texto

Descripción generada automáticamente**

**Método mostrar películas**

Se crea

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Método mostrar actores**

Se crea

**Texto

Descripción generada automáticamente**

**Menú Filtrado**

Se crea

**Texto

Descripción generada automáticamente**

**Método filtrar actor**

Se crea

Texto

Descripción generada automáticamente

**Método filtrar año**

Se crea

Texto

Descripción generada automáticamente

**Método filtrar genero**

Se crea

Texto

Descripción generada automáticamente

**Método graficar**

Se crea

**Texto

Descripción generada automáticamente**