



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Programación Orientada a Objetos

TC1030.303

Proyecto Integrador

Integrantes:

Hortencia Alejandra Ramírez Vázquez A01750150

Miguel Ángel Juárez Dorantes A01753328

Profesor: Antonio Víctor Mejía Olvera

Fecha de entrega:

11 de junio de 2021

Introducción	3
Diagrama UML de clases	3
Descripción	3
Ejecución del programa	6
Menú	6
Opción “Mostrar Catalogo”	6
Opción “Mostrar calificación de Películas y Episodios”	7
Opción “Mostrar calificación de episodios de una serie”	8
Opción “Mostrar Calificación de todas las películas”	9
Opción “Buscar películas y episodios por calificación”	9
Buscar episodios de una serie por calificación	10
Buscar películas por calificación	11
Calificar una película o un episodio	11
Buscar películas y series por género	12
Opciones “No válidas”	13
Argumentación del desarrollo del programa	13
Casos en los que el programa no funcionará	15
Lectura de archivo	15
Interacción de la aplicación con el usuario	16
Conclusión Personal	16
Hortencia Alejandra Ramírez Vázquez	16
Miguel Ángel Juárez Dorantes	17
Referencias	18

Introducción

De acuerdo con Mohn (2019) la programación orientada a objetos es, como dice su nombre un tipo de programación en la que los programadores se basan en la idea de objetos para crear código nuevo, este tipo de programación tiene muchos beneficios como poder insertar, quitar o reutilizar código, esto ayuda a disminuir el tiempo de trabajo y puede llegar a disminuir la cantidad de errores.

Durante el desarrollo de esta materia aprendemos y desarrollamos las bases de la programación orientada a objetos, para aplicar lo visto en clase se llevó a cabo el desarrollo de un proyecto integrador, en este proyecto se nos impuso el reto de crear una simulación de un servicio de streaming utilizando programación orientada a objetos y C++, además, fue necesario implementar todos los conceptos vistos en clase durante el desarrollo de este.

Diagrama UML de clases

En la figura 1 se puede observar el diagrama UML desarrollado para la solución del planteamiento del problema de la situación problema, simulando el desarrollo de un servicio de streaming, cuenta con cinco clases que se describen a continuación:

Descripción

- Video

La clase video es una clase abstracta, para crear clases derivadas que correspondan a un tipo de video como lo es: película y episodio, que contienen los siguientes atributos id, nombre, duración, género, calificación y fecha de estreno. Dentro de sus métodos se encuentran: imprimir, el cual corresponde un método virtual, lo que indica que las clases derivadas de Video deben contener este método, además de contener métodos get de todos sus atributos y set solo para calificación, la clase video además tiene el operador “==” en sobrecarga, el cual nos permite comprar un valor flotante con la calificación de un video.

- Película

Clase heredada de la clase video, película se considera como un tipo de video, esta clase tiene el método de imprimir, para mostrar la información de la película.

- Episodio

Clase heredada de la clase video, Episodio se considera un tipo de video, dentro de sus atributos además de contener los de video, tiene temporada y número de episodio, esta clase nos permite dentro de sus métodos mostrar la información de un episodio, además de tener los métodos get para sus atributos de temporada y número de episodio.

- Serie

La clase serie puede almacenar una lista de videos tipo episodios, además de tener un id de serie y un título. Dentro de sus métodos se encuentran los get y set de sus atributos de id de serie y de título, además de tener métodos como lo es agregarCapitulos, el cual está diseñado para ir agregando los capítulos a la serie, otro método es el imprimir, que muestra los datos y capítulos de la serie. Además esta clase tiene el operador “==” para comprar una variable tipo string con el nombre de una serie.

- ServicioStreaming

La clase ServicioStreaming sirve para hacer la simulación de las funciones que contendrá la aplicación, esta clase tiene como atributos nombre y una lista donde se almacenan las películas y otra para almacenar series, tiene métodos que nos permiten mostrar información de las películas y series de diferentes formas: mostrar el catalogo de ambos, mostrar episodios y películas con cierta calificación, así como también asignarle una nueva calificación al tipo de video, esto lo hace calculando el promedio entre la calificación con la que ya cuenta el video y con la calificación nueva, entre otras funciones, como lo es leer los datos de las películas y series desde un archivo con extensión csv.

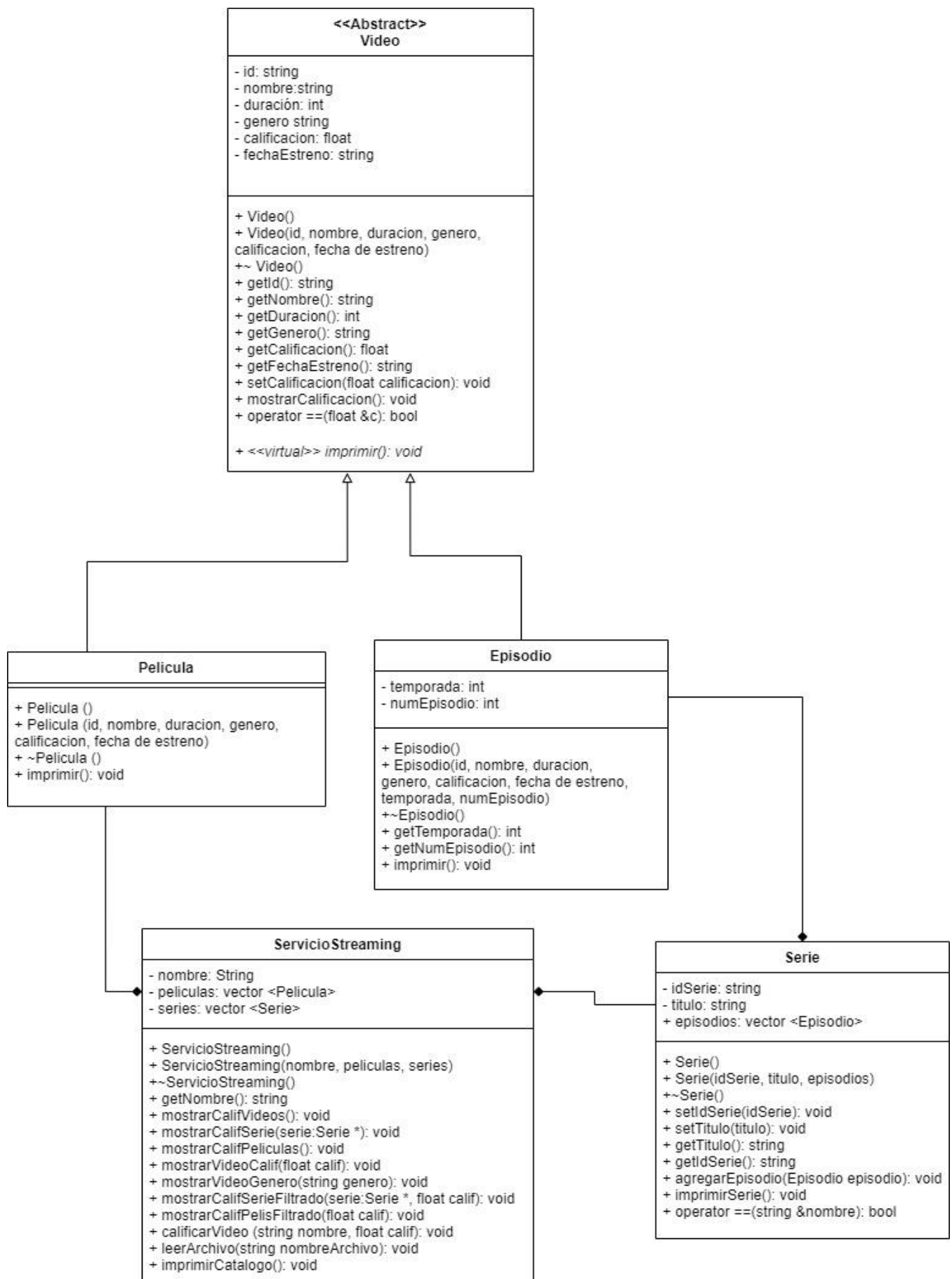


Figura 1. Diagrama UML de clases

Ejecución del programa

Menú

```
*****Bienvenido a Family Moment Video*****

Presione enter para continuar ...

*****Menú Principal*****

1. Mostrar Catalogo
2. Mostrar calificacion de películas y episodios
3. Mostrar calificación de los episodios de una serie
4. Mostrar calificación de todas las películas
5. Buscar películas y episodios por calificación
6. Buscar episodios de una serie por calificación
7. Buscar películas por calificación
8. Calificar una pelicula o un episodio
9. Buscar peliculas y series por genero
10. Salir

Introduzca el numero de la opción:
```

Figura 2. Menú principal del servicio de streaming.

Opción “Mostrar Catalogo”

Muestra toda la información (Id, nombre, duración, género, calificación y fecha de estreno) de las 17 películas y 5 series (Id de serie y título) con sus respectivos episodios (Id, nombre, duración, género, calificación y fecha de estreno), que están almacenados en el archivo que se está leyendo.

```
Introduzca el numero de la opción: 1

***PELICULAS***

Pelicula
Id: tt0107290
Nombre: Jurassic Park
Duración: 127
Genero: Action,Aventura,Ciencia Ficción y Suspenso
Calificación: 8.1
Fecha de estreno: 6/11/1993

Pelicula
Id: tt0796366
Nombre: Star Trek
Duración: 127
Genero: Action,Adventure,Sci-Fi
Calificación: 7.9
Fecha de estreno: 5/8/2009
```

Figura 3. Opción <1>, captura de una sección del catálogo.

```

***SERIES***

Serie
Id: tt0944947
Titulo: Game of Thrones

Episodio 1
  Id: tt1253648
  Nombre Episodio: Winter Is Coming
  Duración: 57
  Genero: Action,Adventure,Drama,Fantasy,Romance
  Calificación: 9
  Fecha de estreno: 4/17/2011
  Numero de episodio: 1
  Temporada:1

Episodio 2
  Id: tt1668746
  Nombre Episodio: The Kingsroad
  Duración: 57
  Genero: Action,Adventure,Drama,Fantasy,Romance
  Calificación: 8.8
  Fecha de estreno: 2/4/2011
  Numero de episodio: 2
  Temporada:1

```

Figura 4. Opción <1>, captura de una sección del catálogo.

Opción “Mostrar calificación de Películas y Episodios”

Esta opción muestra el nombre y calificación de todas las películas y episodios de las series, en este caso muestra la información de 17 películas y de los capítulos de las 5 series.

```

Introduzca el numero de la opción: 2

Película 1
Nombre película: Jurassic Park
Calificación: 8.1

Película 2
Nombre película: Star Trek
Calificación: 7.9

Película 3
Nombre película: Mulan
Calificación: 7.6

Película 4
Nombre película: The Lion King 2: Simba's Pride
Calificación: 6.5

Película 5
Nombre película: Superman
Calificación: 7.3

Película 6
Nombre película: Batman
Calificación: 7.5

```

Figura 5. Opción <2>, captura de una sección del catálogo.

```

Serie 2
Friends

Episodio 1
Nombre Episodio: The One Where Monica Gets a Roommate
Calificacion: 8.4

Episodio 2
Nombre Episodio: The One with the Sonogram at the End
Calificacion: 8.1

Episodio 3
Nombre Episodio: The One with the Thumb
Calificacion: 8.2

Serie 3
Modern Family

Episodio 1
Nombre Episodio: Piloto
Calificacion: 8.4

Episodio 2
Nombre Episodio: The Bicycle Thief
Calificacion: 8.2

Episodio 3
Nombre Episodio: Come Fly with Me
Calificacion: 8

```

Figura 6. Opción <2>, captura de una sección del catálogo.

Opción “Mostrar calificación de episodios de una serie”

Esta opción muestra la calificación de los episodios de una serie, tras ingresar el nombre de la serie, cuando la serie que se ingresa no se encuentra en la búsqueda, el programa muestra un mensaje diciendo “No se encontraron coincidencias”.

```

Introduzca el numero de la opción: 3
Introduzca el nombre de la serie: Friends

Serie
Friends

Episodio 1
Nombre Episodio: The One Where Monica Gets a Roommate
Calificacion: 8.4

Episodio 2
Nombre Episodio: The One with the Sonogram at the End
Calificacion: 8.1

Episodio 3
Nombre Episodio: The One with the Thumb
Calificacion: 8.2
Presione enter para continuar ...

```

Figura 7. Opción <3>, muestra la calificación de los episodios de la serie “Friends”


```
Introduzca el numero de la opción: 3
Introduzca el nombre de la serie: Hello

¡No se encontraron coincidencias!

Presione enter para continuar ...
```

Figura 8. Opción <3>, muestra mensaje “No se encontraron coincidencias”.

Opción “Mostrar Calificación de todas las películas”

Esta opción muestra la calificación de todas las películas que han sido almacenadas.

```
Introduzca el numero de la opción: 4

Película 1
Nombre película: Jurassic Park
Calificación: 8.1

Película 2
Nombre película: Star Trek
Calificación: 7.9

Película 3
Nombre película: Mulan
Calificación: 7.6

Película 4
Nombre película: The Lion King 2: Simba's Pride
Calificación: 6.5
```

Figura 9. Opción <4>, captura de una sección del catálogo de las calificaciones de películas.

Opción “Buscar películas y episodios por calificación”

Esta opción muestra las películas y episodios de una serie, a partir de ingresar una calificación, muestra aquellas películas y/o episodios iguales a dicha calificación, cuando no se encuentra en la búsqueda el valor ingresado, el programa muestra un mensaje diciendo “No se encontraron coincidencias”.

```
Introduzca el numero de la opción: 5
Introduzca la calificación: 7.4

The Simpsons
Nombre Episodio: Homer's Odyssey
Calificación: 7.4
Presione enter para continuar ...
```

Figura 10. Opción <5>, muestra las películas y/o episodios con calificación igual a 7.4.

```

Introduzca el numero de la opción: 5
Introduzca la calificación: 4

¡No se encontraron coincidencias!

Presione enter para continuar ...

```

Figura 11. Opción <5>, muestra mensaje “No se encontraron coincidencias”

Buscar episodios de una serie por calificación

Esta opción permite mostrar episodios de una serie con cierta calificación, esto a través de introducir el nombre de la serie y el número de la calificación, cuando se ingresa un nombre de serie que no está en la base de datos, el programa muestra el mensaje “No se encontraron coincidencias”.

```

Introduzca el numero de la opción: 6
Introduzca el nombre de la serie: Friends
Introduzca la calificación: 8.4

Friends
Episodio
Nombre Episodio: The One Where Monica Gets a Roommate
Calificación: 8.4
Presione enter para continuar ...

```

Figura 12. Opción <6>, muestra los episodios de la serie “Friends” con calificación igual a 8.4.

```

Introduzca el numero de la opción: 6
Introduzca el nombre de la serie: Modern Family
Introduzca la calificación: 8

Modern Family
Episodio
Nombre Episodio: Come Fly with Me
Calificación: 8
Presione enter para continuar ...

```

Figura 13. Opción <6>, muestra los episodios de la serie “Modern Family” con calificación igual a 8.

```

Introduzca el numero de la opción: 6
Introduzca el nombre de la serie: Friends
Introduzca la calificación: 8

¡No se encontraron coincidencias!

Presione enter para continuar ...

```

Figura 14. Opción <6>, muestra mensaje “No se encontraron coincidencias”.

Buscar películas por calificación

Esta opción muestra las películas con cierta calificación, mediante un valor que introduce el usuario mediante la consola, , cuando no se encuentra en la búsqueda el valor ingresado, el programa muestra un mensaje diciendo “No se encontraron coincidencias”.

```
Introduzca el numero de la opción: 7
Introduzca la calificacion: 7.6

Nombre película: Mulan
Calificacion: 7.6

Nombre película: The Greatest Showman
Calificacion: 7.6
Presione enter para continuar ...
█
```

Figura 15. Opción <7>, muestra las películas con calificación igual a 7.6.

```
Introduzca el numero de la opción: 7
Introduzca la calificacion: 4

¡No se encontraron coincidencias!

Presione enter para continuar ...
█
```

Figura 16. Opción <7>, muestra mensaje “No se encontraron coincidencias”.

Calificar una película o un episodio

Esta opción permite al usuario asignarle una calificación, calcula un promedio entre la calificación con la que ya contaba la película o episodio con la calificación ingresada por el usuario, el valor que se debe ingresar debe estar entre el 1 al 10, , cuando no se encuentra en la búsqueda el título de una película o episodio, el programa muestra un mensaje diciendo “No se encontraron coincidencias”.

```
Introduzca el numero de la opción: 8
Introduzca el nombre del episodio o película: Mulan
Introduzca la calificacion que le otorgará (1 a 10): 10
La nueva calificacion de la pelicula es: 8.8
Presione enter para continuar ...
█
```

Figura 17. Opción <8>, asigna una calificación a la película de Mulan.

```

Introduzca el numero de la opción: 8
Introduzca el nombre del episodio o película: Mulan
Introduzca la calificación que le otorgará (1 a 10): 11

¡La calificación ingresada NO es valida!

Presione enter para continuar ...

```

Figura 18. Opción <8>, muestra mensaje “La calificación NO es válida”

```

Introduzca el numero de la opción: 8
Introduzca el nombre del episodio o película: Hello
Introduzca la calificación que le otorgará (1 a 10): 8

¡No se encontraron coincidencias con el titulo ingresado!

Presione enter para continuar ...

```

Figura 19. Opción <8>, muestra mensaje “Título no encontrado”

Buscar películas y series por género

Esta opción muestra las películas y/o series que correspondan a cierto género, esto tras ingresar por parte del usuario un género en específico, cuando no se encuentran títulos con el género ingresado se muestra un mensaje al usuario “No se encontraron coincidencias”.

```

Introduzca el numero de la opción: 9
Introduzca el genero: Mystery

Película
Id: tt0415778
Nombre: End Game
Duración: 93
Genero: Action,Crime,Drama,Mystery,Thriller
Calificación: 5.1
Fecha de estreno: 7/21/2006

Película
Id: tt2935510
Nombre: Ad Astra
Duración: 123
Genero: Adventure,Drama,Mystery,Sci-Fi,Thriller
Calificación: 6.6
Fecha de estreno: 9/20/2019

The 100
Id: tt2912494
Nombre Episodio: Pilot
Duración: 43
Genero: Drama,Mystery,Sci-Fi
Calificación: 7.5
Fecha de estreno: 3/19/2014
Numero de episodio: 1
Temporada:1

```

Figura 20. Opción <9>, busca películas y/o series correspondientes al género “Mystery”

```
Introduzca el numero de la opción: 9
Introduzca el genero: Ciencia

¡No se encontraron coincidencias!

Presione enter para continuar ...
```

Figura 21. Opción <9>, muestra mensaje “No se encontraron coincidencias”.

Opciones “No válidas”

Cuando el usuario ingresa parámetros que no se encuentran como opción del menú, se implementó el manejo de excepciones, tras ingresar una valor fuera del rango de 1 a 10, se muestra el mensaje “Opción no válida”, ya sea un carácter o un valor numérico.

```
Introduzca el numero de la opción: 11
Opción no válida
Presione enter para continuar ...
```

Figura 22. Opciones no válidas.

```
Introduzca el numero de la opción: e
Opción no válida
Presione enter para continuar ...
```

Figura 23. Opciones no válidas.

Argumentación del desarrollo del programa

a) Diseño de clases

Durante el análisis del problema se identificaron cinco clases: video, serie, película, episodio y servicio de streaming, además de la identificación de las clases, nos fue posible identificar la mayoría de los atributos y métodos de estas, gracias a la descripción del problema y al análisis de este pudimos establecer las bases de nuestras clases así como las relaciones entre ellas, ya sea herencia, composición, agregación, etc. Durante todo el proceso de codificación creamos, quitamos y adaptamos métodos y atributos de acuerdo a nuestras necesidades, los resultados fueron clases funcionales, correlacionadas entre sí y con los métodos y atributos necesarios para desempeñar sus funciones de manera eficaz .

b) Concepto de Herencia

El concepto de herencia fue indispensable y de gran ayuda para el desarrollo del proyecto, mediante la aplicación de este concepto nos fue posible reducir la carga de trabajo ya que nos enfocamos en programar la clase base, en este caso video, una vez programada la clase base el desarrollo de sus derivadas (episodio y película) fue más rápido y sencillo puesto que la mayoría de los atributos y métodos de estas fueron heredados.

c) Modificadores de acceso

En la programación orientada a objetos es importante identificar la información que queremos compartir entre clases y aquella que es exclusiva para cierto tipo de clases, en nuestro caso la mayoría de los métodos de nuestras clases son públicos mientras que sus atributos son privados, a excepción de un atributo de la clase serie, ya que la clase “serviciostreaming” utiliza este atributo en sus métodos fue necesario establecer este como público, de esta manera no hay ningún problema con que otra clase acceda a este atributo y trabajar con él.

d) Sobrecarga y sobreescritura de métodos

En nuestra clase abstracta (video) declaramos un método virtual “imprimir”, este método es virtual ya que se sobrecribirá en las clases derivadas de video en este caso “episodio” y “película”, en cada una de estas clases el método se encargará de mostrar en pantalla la información del objeto, ya que esta información es distinta en cada clase es necesario que el método sea codificado de manera distinta. Crear un método virtual nos permitió crear un solo método que desempeñe funciones distintas de acuerdo al tipo de objeto que lo mande a llamar, ahorrando así líneas de código y evitando crear un método distinto para cada clase.

e) Polimorfismo

Con la aplicación del polimorfismo en nuestro programa nos fue posible heredar los métodos y atributos de la clase abstracta “video” a las clases “episodio” y “película”, ya que video es una clase abstracta no podremos instanciar objetos de este tipo, no obstante los objetos que creamos de tipo “película” y “episodio” se comportan y tienen propiedades de un objeto de tipo video. Utilizar el concepto de polimorfismo nos permite trabajar con dos tipos de objetos distintos, con características en común sin necesidad de programar todas estas características en ambos.

f) Clases Abstractas

Se decidió que la clase abstracta de nuestro programa fuera la clase “video” ya que las clases “película” y “episodio” heredan gran parte de los atributos y métodos de esta, no obstante cada una tiene atributos y métodos propios que las distinguen una de otra. Definir nuestra clase

“video” como una clase abstracta nos permitió ahorrar tiempo y facilitar el desarrollo de las otras dos clases ya que codificamos los atributos y métodos que estas tienen en común una sola vez.

g) Sobrecarga de al menos un operador en el conjunto de clases

Para la aplicación del concepto de sobrecarga de operadores se contemplaron diversas opciones: comparar videos, sumarlos entre sí, sumarlos con flotantes, entre otras; al final todas estas opciones fueron descartadas, principalmente porque no encontramos la manera de utilizarlas en nuestro programa. La manera más eficiente de aplicar la sobrecarga de operadores fue para comparar videos con strings sobrecargando el operador “==”, utilizando este operador nos permitió realizar búsquedas de datos eficientes y de manera sencilla.

h) Uso de excepciones

En el caso de las excepciones se optó por implementarlas en el menú con el que interactúa el usuario, de esta forma nos aseguramos de que el usuario pueda seguir utilizando el programa aún si este comete algún error durante la manipulación del mismo, si este es el caso el programa le indicará al usuario y este podrá volver a intentarlo. Gracias a la implementación de excepciones en el menú este se vuelve más interactivo, guía al usuario para que este aprenda a usar el programa aún si este se equivoca.

Casos en los que el programa no funcionará

Lectura de archivo

El programa se diseñó para que el usuario pueda manejarlo y sea susceptible a los errores que se puedan cometer a la hora de introducir datos en la aplicación del programa.

Sin embargo si se desea cargar un archivo diferente al cargado en la ejecución del programa, esta base de datos debe contener los mismos parámetros que el archivo “BasePelículas4.csv”, en el orden en el que se encuentran guardados.

De igual manera el archivo que se cargue en el programa debe de ser un archivo con extensión “csv”, puesto que es la extensión que el programa puede aceptar para hacer la lectura para la base de datos.

Interacción de la aplicación con el usuario

El programa es sensible a mayúsculas y minúsculas, por lo que si se ingresan nombres de películas, series y episodios con mayúsculas y minúsculas diferente a como está guardado en la base de datos y aunque se trate del mismo título, el programa no encontrará la opción buscada.

Después de ejecutar algunas funciones el programa muestra una pantalla en negro como se muestra en la *figura 24*, en espera de que el usuario ejecute una acción, en estos casos lo único que se debe hacer es dar enter para que la aplicación continúe con la ejecución del programa.

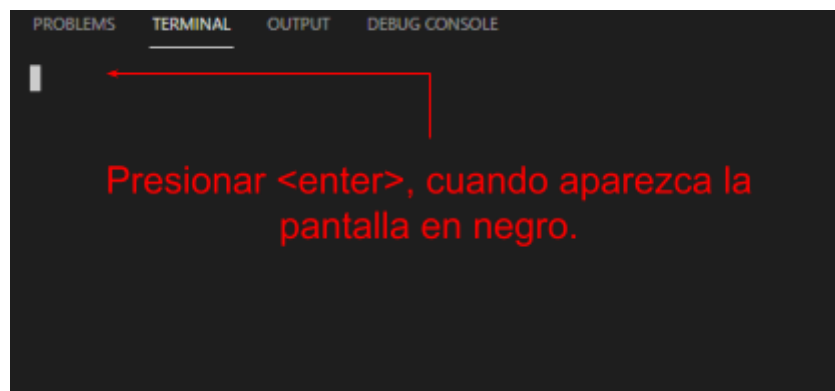


Figura 24. Ejemplo de pantalla negra después de la ejecución de algunas funciones.

Conclusión Personal

Hortencia Alejandra Ramírez Vázquez

La elaboración del proyecto integrador, nos permitió visualizar la importancia de realizar un buen diseño de diagrama UML de clases, al tener un problema planteado, pues de esta forma es más sencillo hacer la manipulación tanto del programa, de las clases, de los métodos, como de la información guardada. Relacionamos todos los conceptos vistos durante la unidad de formación, para después poder comprender la aplicación que tienen en una situación real o en un problema planteado. El aplicar los diferentes conceptos que tiene la programación orientada a objetos, nos permite elaborar aplicaciones con un nivel más alto de programación que hacen más eficientes el desarrollo de programas.

Miguel Ángel Juárez Dorantes

Durante el desarrollo de este proyecto se pusieron en práctica los conocimientos adquiridos en clase, gracias a esto nos percatamos de que la programación orientada a objetos nos permite ahorrar tiempo y líneas de código gracias a todas las relaciones que se pueden crear entre clases, además, nos permite visualizar las soluciones de los problemas de una manera más sencilla, al trabajar con objetos y clases la relación que existe entre estas y situaciones reales facilitan la comprensión y el desarrollo de un problema.

Referencias

- Ma, E. (2020). How to split and iterate a string separated by a specific character in C++? - SysTutorials.
<https://www.systutorials.com/how-to-split-and-iterate-a-string-separated-by-another-string-in-c/>
- Mohn, E. (2019). Object-Oriented Programming (OOP). Salem Press Encyclopedia of Science.