



**Instituto Tecnológico y de Estudios Superiores de Monterrey**

**Campus Estado de México**

**Proyecto Integrador**  
**“Modelado de servicio de streaming”**

Programación Orientada a Objetos (TC1030)

Jesús Ángel Guzmán Ortega A01799257

Profesor: Roberto Martínez Román

Group 305

June, 2023

## Índice:

1. Portada.....	1
2. Índice de contenido.....	2
3. Introducción.....	3-4
4. Diagrama de clases explicando el diseño.....	5
5. Ejemplos de ejecución de todas las opciones.....	6-10
6. Identificación de casos que harían que el proyecto deje de funcionar.....	11
7. Conclusión.....	11-12
8. Referencias consultadas en formato APA.....	13

## *Introducción:*

En los últimos años, han proliferado los servicios de streaming de video bajo demanda por ejemplo Netflix, Disney Plus, HBO Max, entre otros. Algunos de ellos se especializan por el volumen de videos que proporcionan a sus usuarios mientras que otros se han puesto el reto de mostrar solamente videos de su propia marca. Una versión limitada para apoyar a un futuro proveedor de este tipo de servicios es la que se describe a continuación:

Se quiere trabajar con dos tipos de videos: películas y episodios de series.

Todo video tiene:

- ID.
- Nombre de la película o nombre del episodio de una serie.
- Género (Pueden ser varios separados por el carácter **\*\*&\*\***).
- Calificación en escala de **\*\*1** a **7\*\***.
- Duración en minutos.
- Fecha de estreno en formato **\*\*\*mm/dd/aaaa\*\*\***.

Las series tienen episodios y cada episodio tiene:

- ID.
- Nombre.
- Temporada a la que pertenece.
- Número de episodio.

Nos interesa construir un sistema de consulta para esta información que se encuentra en un archivo de texto.

El sistema debe ser capaz de mostrar el siguiente menú de manera iterativa, termina hasta que el usuario selecciona Salir:

1. Cargar el archivo de datos en formato **\*\*csv\*\***.
2. Mostrar los videos en general con:
  - Una calificación mayor o igual a un valor tecleado por el usuario.
  - Un género tecleado por el usuario.
3. Mostrar todos los episodios de una determinada serie que teclea el usuario.
4. Mostrar las películas con una calificación mayor a un número tecleado por el usuario.
5. Calificar un video.
  - Pedir título de película o nombre del capítulo de la serie a calificar.
  - Pedir valor otorgado y asignarlo (reemplaza la calificación anterior).
6. Calcular el promedio de la calificación de una serie (promedio de todos los episodios).
7. Salir. [1]

El sistema será implementado en C++ utilizando el paradigma orientado a objetos. La solución consta de 9 archivos en total, junto con un archivo CSV que servirá como base de datos para el sistema. De los 9 archivos, 5 serán archivos con extensión .cpp. A continuación, se describen brevemente los archivos y su función dentro del sistema:

`main.cpp`: Este archivo contendrá la función `main`, que es el punto de entrada del programa. Aquí se iniciará la ejecución del sistema y se gestionarán las interacciones con el usuario.

`pelicula.cpp` y `pelicula.h`: Estos archivos contendrán la implementación y la definición, respectivamente, de la clase `Pelicula`. La clase `Pelicula` representará una película dentro del sistema y contendrá los atributos y métodos necesarios para gestionar la información de una película.

`video.cpp` y `video.h`: Estos archivos contendrán la implementación y la definición, respectivamente, de la clase `Video`. La clase `Video` será una clase base de la cual la clase `Pelicula` heredará. La clase `Video` contendrá los atributos y métodos comunes a todos los tipos de videos dentro del sistema.

`episodio.cpp` y `episodio.h`: Estos archivos contendrán la implementación y la definición, respectivamente, de la clase `Episodio`. La clase `Episodio` será una clase derivada de la clase `Video` y representará un episodio de una serie. Contendrá atributos y métodos específicos para gestionar la información de un episodio.

`funciones_extras.cpp` y `funciones_extras.h`: Estos archivos contendrán la implementación y la definición, respectivamente, de funciones adicionales que no pertenecen directamente a ninguna de las clases anteriores. Estas funciones podrían incluir operaciones de lectura y escritura en el archivo CSV, funciones de búsqueda, entre otras.

Además de estos archivos, el sistema hará uso de un archivo CSV como base de datos para almacenar la información de las películas, videos y episodios. Este archivo contendrá la información necesaria para mantener el estado del sistema y realizar operaciones de consulta y actualización de datos.

Se dará seguimiento y actualizaciones a través de un repositorio de control de versiones en GitHub. Esto permitirá mantener un registro de los cambios realizados en el código fuente y facilitará. El repositorio contendrá los archivos del sistema, incluyendo los archivos de código fuente (.cpp y .h) y el archivo CSV de la base de datos.

## Diagrama de Clases:

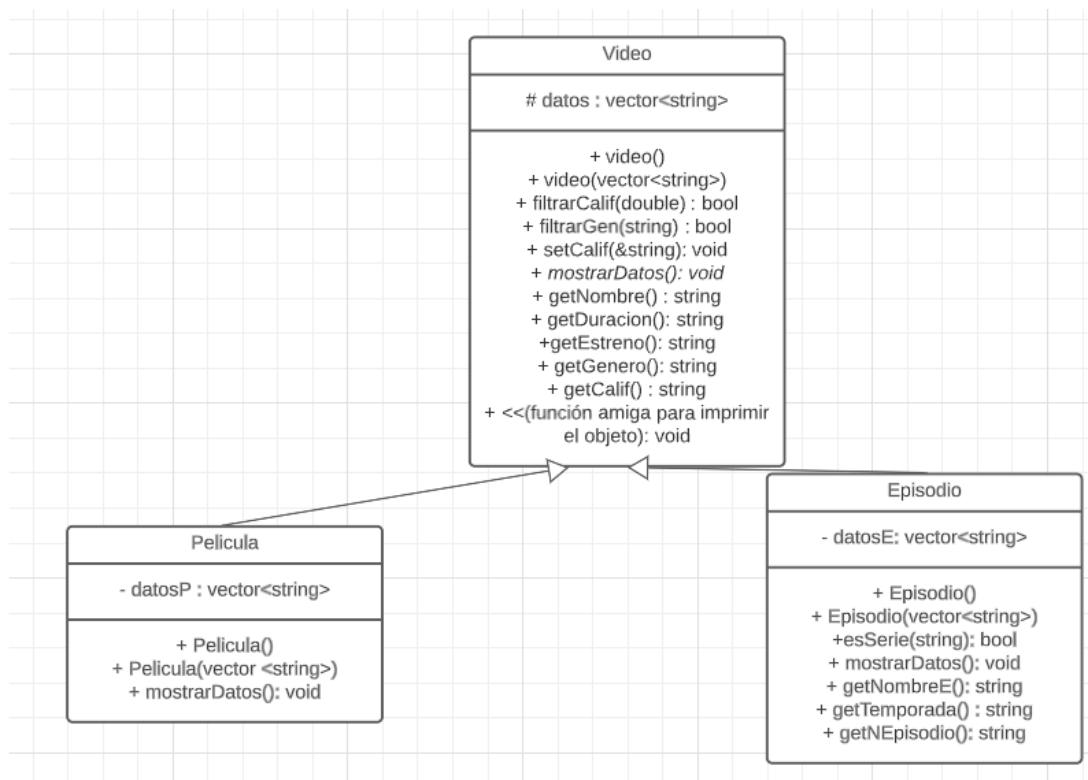


Figura [1] (“Diagrama de clases”)

El diagrama de clases presenta las clases Video, Pelicula y Episodio. En esta jerarquía, la clase Video es la superclase, lo que implica que Pelicula y Episodio son clases derivadas. Video se define como una clase abstracta debido a la presencia de métodos virtuales, lo que evita su instancia directa.

En cuanto a los atributos, la clase Video contiene un vector de strings, el cual se declara como protegido para permitir su acceso desde las clases hijas.

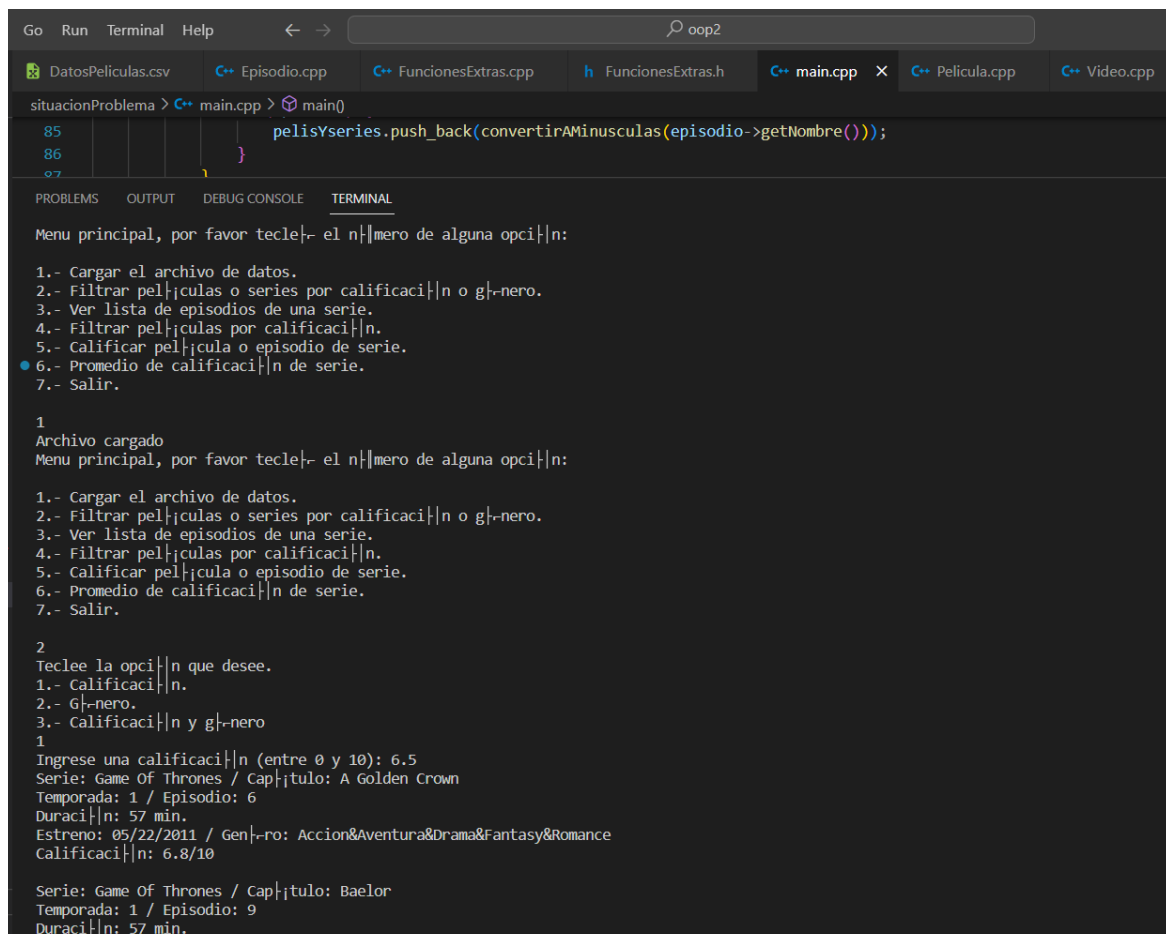
La funcionalidad de estas clases incluye varios métodos. Para acceder a las posiciones del vector de strings, se implementan getters correspondientes. Además, se proporciona un setter para que el usuario pueda calificar una película o un episodio en particular.

Dentro de los métodos que se reescriben y se utilizan de manera polimórfica, destaca el método `mostrarDatos`. Este método se adapta según el tipo de objeto, ya que las propiedades a mostrar difieren dependiendo si se trata de una película o un episodio.

En el caso específico de la clase Episodio, se incluye un método único llamado `esSerie`, el cual devuelve un valor booleano para determinar si el objeto es efectivamente un episodio.

Además, se define una función amiga para sobrecargar el operador `<<` y llamar al método `mostrarDatos` utilizando la sintaxis de `cout << objeto`.

## Ejemplos de ejecución:



```
Go Run Terminal Help  oop2
DatosPeliculas.csv  C++ Episodio.cpp  C++ FuncionesExtras.cpp  h FuncionesExtras.h  C++ main.cpp X  C++ Pelicula.cpp  C++ Video.cpp

situacionProblema > C++ main.cpp > main()

85      pelisYseries.push_back(convertirAMinusculas(episodio->getNombre()));
86      }
87

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Menu principal, por favor teclee el número de alguna opción:

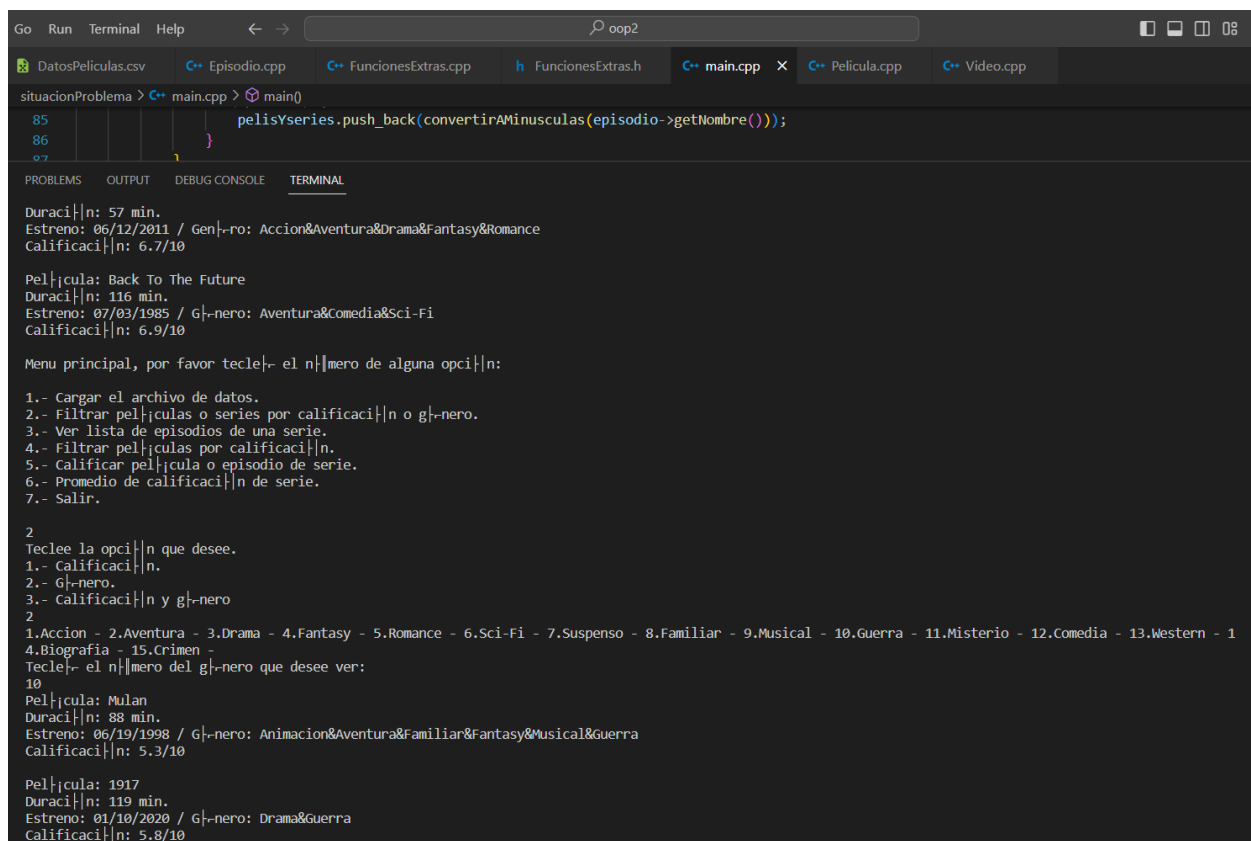
1.- Cargar el archivo de datos.
2.- Filtrar películas o series por calificación o género.
3.- Ver lista de episodios de una serie.
4.- Filtrar películas por calificación.
5.- Calificar película o episodio de serie.
6.- Promedio de calificación de serie.
7.- Salir.

1
Archivo cargado
Menu principal, por favor teclee el número de alguna opción:

1.- Cargar el archivo de datos.
2.- Filtrar películas o series por calificación o género.
3.- Ver lista de episodios de una serie.
4.- Filtrar películas por calificación.
5.- Calificar película o episodio de serie.
6.- Promedio de calificación de serie.
7.- Salir.

2
Teclee la opción que desee.
1.- Calificación.
2.- Género.
3.- Calificación y género
1
Ingrese una calificación (entre 0 y 10): 6.5
Serie: Game Of Thrones / Capítulo: A Golden Crown
Temporada: 1 / Episodio: 6
Duración: 57 min.
Estreno: 05/22/2011 / Género: Accion&Aventura&Drama&Fantasy&Romance
Calificación: 6.8/10

Serie: Game Of Thrones / Capítulo: Baelor
Temporada: 1 / Episodio: 9
Duración: 57 min.
```



```
Go Run Terminal Help  oop2
DatosPeliculas.csv  C++ Episodio.cpp  C++ FuncionesExtras.cpp  h FuncionesExtras.h  C++ main.cpp X  C++ Pelicula.cpp  C++ Video.cpp

situacionProblema > C++ main.cpp > main()

85      pelisYseries.push_back(convertirAMinusculas(episodio->getNombre()));
86      }
87

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Duración: 57 min.
Estreno: 06/12/2011 / Género: Accion&Aventura&Drama&Fantasy&Romance
Calificación: 6.7/10

Película: Back To The Future
Duración: 116 min.
Estreno: 07/03/1985 / Género: Aventura&Comedia&Sci-Fi
Calificación: 6.9/10

Menu principal, por favor teclee el número de alguna opción:

1.- Cargar el archivo de datos.
2.- Filtrar películas o series por calificación o género.
3.- Ver lista de episodios de una serie.
4.- Filtrar películas por calificación.
5.- Calificar película o episodio de serie.
6.- Promedio de calificación de serie.
7.- Salir.

2
Teclee la opción que desee.
1.- Calificación.
2.- Género.
3.- Calificación y género
2
1.Accion - 2.Aventura - 3.Drama - 4.Fantasy - 5.Romance - 6.Sci-Fi - 7.Suspense - 8.Familiar - 9.Musical - 10.Guerra - 11.Misterio - 12.Comedia - 13.Western - 14.Biografia - 15.Crimen
Teclee el número del género que desee ver:
10
Película: Mulan
Duración: 88 min.
Estreno: 06/19/1998 / Género: Animacion&Aventura&Familiar&Fantasy&Musical&Guerra
Calificación: 5.3/10

Película: 1917
Duración: 119 min.
Estreno: 01/10/2020 / Género: Drama&Guerra
Calificación: 5.8/10
```

```
Go Run Terminal Help  oop2
DatosPeliculas.csv  C++ Episodio.cpp  C++ FuncionesExtras.cpp  h FuncionesExtras.h  C++ main
situacionProblema > C++ main.cpp > main()
85      pelisYseries.push_back(convertirAMinusculas(episodio->getNombre
86      }
87
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Menu principal, por favor teclee el nmero de alguna opci:
1.- Cargar el archivo de datos.
2.- Filtrar pelculas o series por calificaci|n o g|nero.
3.- Ver lista de episodios de una serie.
4.- Filtrar pelculas por calificaci|n.
5.- Calificar pelcula o episodio de serie.
6.- Promedio de calificaci|n de serie.
7.- Salir.
3
Teclee el nombre por favor: game of thrones
Serie: Game Of Thrones / Capitulo: Winter Is Coming
Temporada: 1 / Episodio: 1
Duraci|n: 57 min.
Estreno: 04/17/2011 / G|nero: Accion&Aventura&Drama&Fantasy&Romance
Calificaci|n: 6.3/10
Serie: Game Of Thrones / Capitulo: The Kingsroad
Temporada: 1 / Episodio: 2
Duraci|n: 57 min.
Estreno: 02/04/2011 / G|nero: Accion&Aventura&Drama&Fantasy&Romance
Calificaci|n: 6.2/10
Serie: Game Of Thrones / Capitulo: Lord Snow
Temporada: 1 / Episodio: 3
Duraci|n: 57 min.
Estreno: 01/05/2011 / G|nero: Accion&Aventura&Drama&Fantasy&Romance
Calificaci|n: 6.1/10
Serie: Game Of Thrones / Capitulo: Cripples Bastards And Broken Things
Temporada: 1 / Episodio: 4
Duraci|n: 57 min.
Estreno: 08/05/2011 / G|nero: Accion&Aventura&Drama&Fantasy&Romance
Calificaci|n: 6.2/10
Serie: Game Of Thrones / Capitulo: The Wolf And The Lion
Temporada: 1 / Episodio: 5
```

```
Go Run Terminal Help  oop2
DatosPeliculas.csv  C++ Episodio.cpp  C++ FuncionesExtras.cpp  h FuncionesExtras.h  C++ main.cpp
situacionProblema > C++ main.cpp > main()
85      pelisYseries.push_back(convertirAMinusculas(episodio->getNombre())
86      }
87
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Estreno: 06/05/2011 / G|nero: Accion&Aventura&Drama&Fantasy&Romance
Calificaci|n: 6.3/10
Serie: Game Of Thrones / Capitulo: Baelor
Temporada: 1 / Episodio: 9
Duraci|n: 57 min.
Estreno: 06/12/2011 / G|nero: Accion&Aventura&Drama&Fantasy&Romance
Calificaci|n: 6.7/10
Menu principal, por favor teclee el nmero de alguna opci:
1.- Cargar el archivo de datos.
2.- Filtrar pelculas o series por calificaci|n o g|nero.
3.- Ver lista de episodios de una serie.
4.- Filtrar pelculas por calificaci|n.
5.- Calificar pelcula o episodio de serie.
6.- Promedio de calificaci|n de serie.
7.- Salir.
3
Teclee el nombre por favor: DEATH NOTE
Serie: Death Note / Capitulo: Confrontation
Temporada: 1 / Episodio: 2
Duraci|n: 24 min.
Estreno: 10/21/2007 / G|nero: Animacion&Crimen&Drama&Fantasy&Misterio&Suspenso
Calificaci|n: 6.3/10
```

```
DatosPelículas.csv C++ Episodio.cpp C++ FuncionesExtras.cpp h FuncionesExtras.h C++ main.cpp x C
situacionProblema > C++ main.cpp > main()
85         pelisYseries.push_back(convertirAMinusculas(episodio->getNombre()));
86     }
87
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Temporada: 1 / Episodio: 2
Duraci|n: 24 min.
Estreno: 10/21/2007 / G|nero: Animacion&Crimen&Drama&Fantasy&Misterio&Suspense
Calificaci|n: 6.3/10
Menu principal, por favor tecle|- el n|mero de alguna opci|n:
1.- Cargar el archivo de datos.
2.- Filtrar pel|culas o series por calificaci|n o g|nero.
3.- Ver lista de episodios de una serie.
4.- Filtrar pel|culas por calificaci|n.
5.- Calificar pel|cula o episodio de serie.
6.- Promedio de calificaci|n de serie.
7.- Salir.
3
Teclee el nombre por favor: SERIE QUE NO EXISTE
La serie no est|i en la base de datos, verifique c|mo la escribi| e int|ntelo nuevamente.
Presione 0 para continuar o cualquier otro n|mero para salir.
0
Teclee el nombre por favor: STRANGER things
Serie: Stranger Things / Cap|tulo: Chapter Six: The Monster
Temporada: 1 / Episodio: 6
Duraci|n: 51 min.
Estreno: 07/15/2016 / G|nero: Drama&Fantasy&Terror&Misterio&Sci-Fi&Suspense
Calificaci|n: 6.2/10
Menu principal, por favor tecle|- el n|mero de alguna opci|n:
1.- Cargar el archivo de datos.
2.- Filtrar pel|culas o series por calificaci|n o g|nero.
3.- Ver lista de episodios de una serie.
4.- Filtrar pel|culas por calificaci|n.
5.- Calificar pel|cula o episodio de serie.
6.- Promedio de calificaci|n de serie.
7.- Salir.
```

```
Go Run Terminal Help oop2
DatosPelículas.csv C++ Episodio.cpp C++ FuncionesExtras.cpp h FuncionesExtras.h C++ main.cpp
situacionProblema > C++ main.cpp > main()
85         pelisYseries.push_back(convertirAMinusculas(episodio->getNombre()));
86     }
87
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
4.- Filtrar pel|culas por calificaci|n.
5.- Calificar pel|cula o episodio de serie.
6.- Promedio de calificaci|n de serie.
7.- Salir.
3
Teclee el nombre por favor: hvuyi
La serie no est|i en la base de datos, verifique c|mo la escribi| e int|ntelo nuevamente.
Presione 0 para continuar o cualquier otro n|mero para salir.
1
Menu principal, por favor tecle|- el n|mero de alguna opci|n:
1.- Cargar el archivo de datos.
2.- Filtrar pel|culas o series por calificaci|n o g|nero.
3.- Ver lista de episodios de una serie.
4.- Filtrar pel|culas por calificaci|n.
5.- Calificar pel|cula o episodio de serie.
6.- Promedio de calificaci|n de serie.
7.- Salir.
4
Ingrese una calificaci|n (entre 0 y 10): 6.5
Pel|cula: Back To The Future
Duraci|n: 116 min.
Estreno: 07/03/1985 / G|nero: Aventura&Comedia&Sci-Fi
Calificaci|n: 6.9/10
```



```
Go Run Terminal Help  oop2
DatosPeliculas.csv  C++ Episodio.cpp  C++ FuncionesExtras.cpp  h FuncionesExtras.h  C++ main.c
situacionProblema > C++ main.cpp > main()
24 while(opc != "7"){
25     menu(); // Mostrar el menú principal
26     cin >> opc;
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL
• Menu principal, por favor teclee el número de alguna opción:
1.- Cargar el archivo de datos.
2.- Filtrar películas o series por calificación o género.
3.- Ver lista de episodios de una serie.
4.- Filtrar películas por calificación.
5.- Calificar película o episodio de serie.
6.- Promedio de calificación de serie.
7.- Salir.
5
Teclee el nombre por favor: game of thrones
La serie no está en la base de datos, verifique cómo la escribió e inténtelo nuevamente.
Presione 0 para continuar o cualquier otro número para salir.
0
Teclee el nombre por favor: jurassic park
Ingrese una calificación (entre 0 y 10): 9
Calificación guardada exitosamente.
Menu principal, por favor teclee el número de alguna opción:
1.- Cargar el archivo de datos.
2.- Filtrar películas o series por calificación o género.
3.- Ver lista de episodios de una serie.
4.- Filtrar películas por calificación.
5.- Calificar película o episodio de serie.
6.- Promedio de calificación de serie.
7.- Salir.
5
Teclee el nombre por favor: winter is COMING
Ingrese una calificación (entre 0 y 10): 9
Calificación guardada exitosamente.
```

```
Go Run Terminal Help  Funcio
DatosPeliculas.csv  C++ Episodio.cpp  C++ FuncionesExtras.cpp  h Funcio
situacionProblema > C++ main.cpp > ...
1 /**
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Menu principal, por favor teclee el número de alguna opción:
1.- Cargar el archivo de datos.
2.- Filtrar películas o series por calificación o género.
3.- Ver lista de episodios de una serie.
4.- Filtrar películas por calificación.
5.- Calificar película o episodio de serie.
6.- Promedio de calificación de serie.
7.- Salir.
2
Teclee la opción que desee.
1.- Calificación.
2.- Género.
3.- Calificación y género
1
Ingrese una calificación (entre 0 y 10): 9
Serie: Game Of Thrones / Capítulo: Winter Is Coming
Temporada: 1 / Episodio: 1
Duración: 57 min.
Estreno: 04/17/2011 / Género: Accion&Aventura&Drama&Fantasy&Romance
Calificación: 9.000000/10
Película: Jurassic Park
Duración: 127 min.
Estreno: 06/11/1993 / Género: Accion&Aventura&Sci-Fi&Suspense
Calificación: 9.000000/10
```

```
Go Run Terminal Help
DatosPeliculas.csv C++ Episodio.cpp C++ FuncionesExtras.cpp h Funcio
situacionProblema > C++ main.cpp > ...
1 /**
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1.- Cargar el archivo de datos.
2.- Filtrar películas o series por calificación o género.
3.- Ver lista de episodios de una serie.
4.- Filtrar películas por calificación.
5.- Calificar película o episodio de serie.
6.- Promedio de calificación de serie.
7.- Salir.

4
Ingrese una calificación (entre 0 y 10): 9
Película: Jurassic Park
Duración: 127 min.
Estreno: 06/11/1993 / Género: Accion&Aventura&Sci-Fi&Suspenso
Calificación: 9.000000/10

Menu principal, por favor teclee el número de alguna opción:

1.- Cargar el archivo de datos.
2.- Filtrar películas o series por calificación o género.
3.- Ver lista de episodios de una serie.
4.- Filtrar películas por calificación.
5.- Calificar película o episodio de serie.
6.- Promedio de calificación de serie.
7.- Salir.

6
Teclee el nombre por favor: game of THROnes
El promedio es: 6.67778/10
```

```
DatosPeliculas.csv C++ Episodio.cpp C++ FuncionesExtras.cpp h Funciones
situacionProblema > C++ main.cpp > ...
1 /**
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

6
Teclee el nombre por favor: game of THROnes
El promedio es: 6.67778/10

Menu principal, por favor teclee el número de alguna opción:

1.- Cargar el archivo de datos.
2.- Filtrar películas o series por calificación o género.
3.- Ver lista de episodios de una serie.
4.- Filtrar películas por calificación.
5.- Calificar película o episodio de serie.
6.- Promedio de calificación de serie.
7.- Salir.

5445
Menu principal, por favor teclee el número de alguna opción:

1.- Cargar el archivo de datos.
2.- Filtrar películas o series por calificación o género.
3.- Ver lista de episodios de una serie.
4.- Filtrar películas por calificación.
5.- Calificar película o episodio de serie.
6.- Promedio de calificación de serie.
7.- Salir.

6554656
Menu principal, por favor teclee el número de alguna opción:

1.- Cargar el archivo de datos.
2.- Filtrar películas o series por calificación o género.
3.- Ver lista de episodios de una serie.
4.- Filtrar películas por calificación.
5.- Calificar película o episodio de serie.
6.- Promedio de calificación de serie.
7.- Salir.

7
Gracias.
PS C:\Users\trato\Documents\oop2\situacionProblema>
```

## *Casos que harían que el programa dejará de funcionar:*

Entre los casos que harían que el programa dejará de funcionar, se incluyen los siguientes:

1. Modificación de las columnas de la base de datos en formato CSV: Esto causaría problemas porque los getters y setters asumen un orden específico de los datos en cada columna de la base de datos.
2. Archivo CSV vacío o con nombre incorrecto: Si el archivo CSV está vacío o si el nombre del archivo no coincide con el que se especifica en el código, el programa podría devolver un valor de -1, lo que indica un error al intentar leer los datos.
3. Duplicidad de nombres entre películas y episodios: Si se introduce una película y un episodio con el mismo nombre, el programa no leerá correctamente cada identificador (id) y devolverá ambos de forma incorrecta al realizar una solicitud.
4. En varias secciones del código, se utiliza el operador `dynamic_cast` para convertir un puntero a Video a un puntero a Episodio o Pelicula. Sin embargo, el operador `dynamic_cast` debe usarse con precaución, ya que devuelve un puntero nulo si la conversión no es válida. No se realiza una verificación adecuada de los punteros resultantes antes de acceder a los miembros de los objetos convertidos. Esto puede provocar errores y comportamiento indefinido si se accede a punteros nulos al recopilar la información del csv.
5. El último error sería que si el usuario teclea una calificación mayor a cualquier Video de la base de datos, el programa no muestra un mensaje indicando que no se encontraron resultados.

Es importante abordar y manejar estos casos de error para garantizar el correcto funcionamiento del programa y la integridad de los datos.

## *Conclusión:*

En conclusión, para desarrollar correctamente este proyecto personalmente, se necesito representar de manera estructurada la información relacionada con cada elemento, esto se hizo en el diagrama de clases.

El paradigma de programación orientada a objetos utilizado en este proyecto permite organizar y gestionar eficientemente los datos, aprovechando la herencia y polimorfismo para lograr un diseño flexible y escalable.

El uso de un archivo CSV como base de datos proporciona una forma sencilla de almacenar y acceder a la información de los videos, facilitando su carga y consulta dentro del sistema.

Durante el desarrollo del proyecto, se identificaron algunos posibles casos de error que podrían afectar el funcionamiento del programa, como modificaciones en la estructura del archivo CSV, archivos vacíos o con nombres incorrectos, y la posibilidad de duplicidad de nombres entre películas y episodios. Es importante considerar y manejar estos casos para garantizar la integridad de los datos y el correcto desempeño del sistema.

Por esto, puedo decir que la mejor manera de aprender programación y el paradigma orientado a objetos es mediante la práctica y el desarrollo de proyectos.

### ***Referencias:***

1. Tc1030-Rmroman. (s. f.). GitHub - TC1030-rmroman/situacionProblema: Situación problema para Programación orientada a objetos. GitHub. <https://github.com/TC1030-rmroman/situacionProblema>
2. Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd ed.). Prentice Hall
3. Stroustrup, B. (2013). The C++ Programming Language (4th ed.). Addison-Wesley.