



Tecnológico de Monterrey

**Tecnológico de Monterrey
(Campus Guadalajara)**

Programación orientada a objetos

TC1030 Grupo 12

Proyecto Integrador

Propuesta de solución a Situación Problema

Profesor: Edgar Gerardo Salinas Gurrión

Alumno:

Misael Octavio Rodríguez Macías - A01639786

Fecha:

13 de Junio del 2021.

Índice

➤ Introducción	03
➤ Diagrama de clases	04
➤ Ejecución del código	05
➤ Justificaciones del código	10
➤ Casos de error	12
➤ Conclusión	13
➤ Referencias	

Situación Problema

Modelado de servicio de streaming

En los últimos años, se han proliferado los servicios de streaming de video bajo demanda por ejemplo Netflix, Disney, DC 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.

El secreto de los servicios de streaming es tener una colección amplia de manera que cualquier cosa que usted quiera escuchar esté disponible para cuando la quiera oír. Así, en la teoría la competencia se basa en quien ofrece la mejor experiencia de uso. Eso sería pensar en los usuarios y no en el bolsillo del dueño del servicio.

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 series. Todo video tiene un ID, un nombre, una duración y un género (drama, acción, misterio).

Las series tienen episodios y cada episodio tiene un título y temporada a la que pertenece. Nos interesa conocer la calificación promedio que ha recibido cada uno de los videos. Esta calificación está en escala de 1 a 5 donde 5 es la mejor calificación.

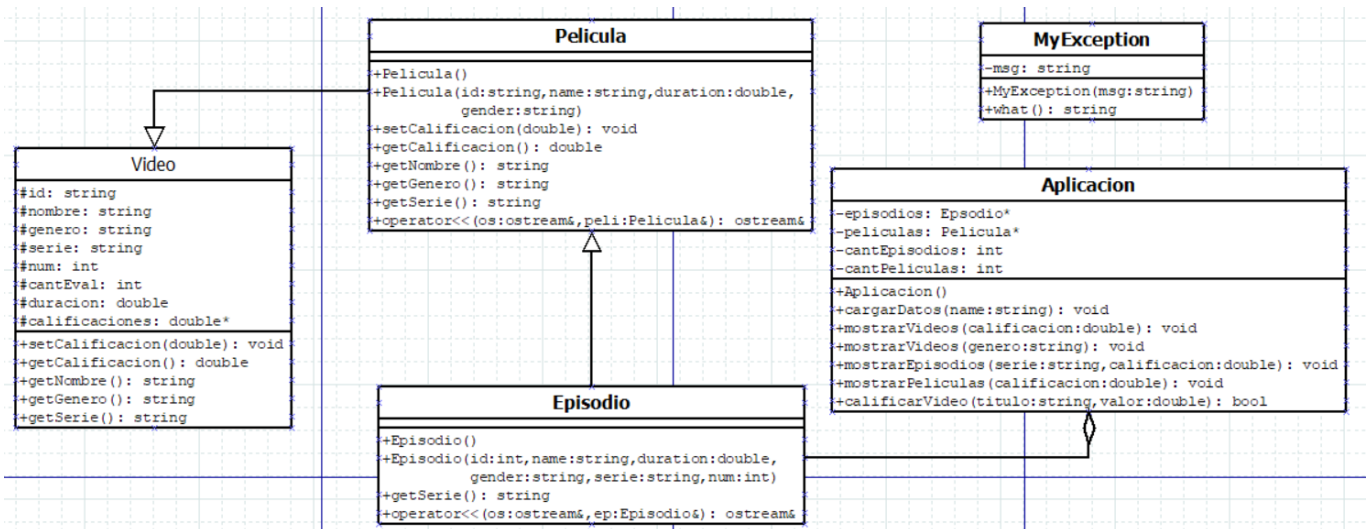
Este sistema debe ser capaz de :

- ✓ Cargar archivo de datos.
- ✓ Mostrar los videos en general.
- ✓ Mostrar los episodios de una determinada serie.
- ✓ Mostrar las películas.
- ✓ Calificar un video.

Para este sistema se pondrán en práctica todos los conocimientos y temas vistos durante el curso de 5 semanas, con el fin de obtener una versión eficiente, mucho mejor implementada y con una sintaxis más entendible a la vista.

Este sistema está pensado para apoyar a los futuros servicios de streaming, sin embargo, solamente de manera limitada, debido a la misma simplicidad del sistema, el tiempo y conocimientos que se disponen.

Diagrama de clase



Para el desarrollo de la aplicación que daría solución a la situación problema, se optó por este diseño de clases, el cual cuenta con 5 clases en total, entre ellas una es abstracta y 2 son clases hijas.

Explicando un poco acerca de este diseño, la clase **Video** es abstracta, debido a que es una clase padre, esta no se va a instanciar (solamente sus hijas) e incluso algunos de sus métodos carecen de claridad dentro de esa clase, pero tienen más sentido en sus clases hijas, por lo que solamente se toma como un modelo/plantilla para considerar.

La clase **Película** es la clase hija de **Video**, en este se implementan todos los métodos de **Video** y se crea un constructor para instanciarse, la clase **Episodio** es hija de **Película**, aunque bien podría ser hija de **Video** directamente para mantener conceptos más claros y evidentes, se realiza así con el objetivo de reutilizar el código de los métodos ya implementados en **Película**.

Finalmente están dos clases que no manejan estos conceptos de herencia ni polimorfismo, la clase **Aplicación** tiene como única función unificar las clases anteriores para producir los métodos necesarios en el menú de la aplicación, de esta manera se obtiene una mayor claridad y organización en caso de ocurrir algún error o necesitar modificar algún procedimiento específico y por último la clase **MyException** es independiente de las demás, ya que solamente se utiliza en el archivo `main` para verificar y validar datos de entrada principalmente.

De estas clases, aquellas solicitadas e indispensables para la solución del problema son las que poseen herencia y polimorfismo, es decir las clases **Video**, **Película** y **Episodio** son las clases base en las cuales se basa toda la solución, sin embargo, se decidió agregar la clase **MyException** para poder crear propios casos de excepciones dentro del programa, lo cual ayudaría a la disminución de errores principalmente y obtención de resultados correctos esperados, en cuanto a la clase **Aplicación**, se desarrolló para aumentar la organización de los procedimientos de cada una de las opciones del menú del sistema, logrando así que el acceso a estos métodos sea mucho más rápido y el archivo `main` que contiene el menú del sistema, este mucho más simplificado y sea simple de comprender.

Ejecución del código

- Menú principal del sistema:

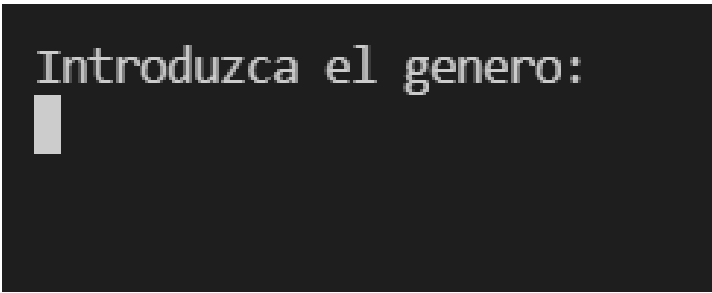
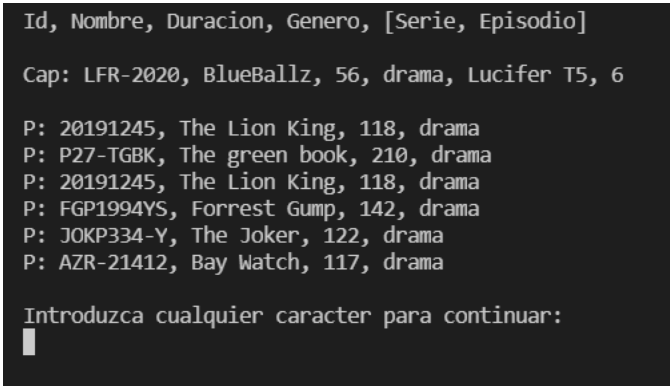
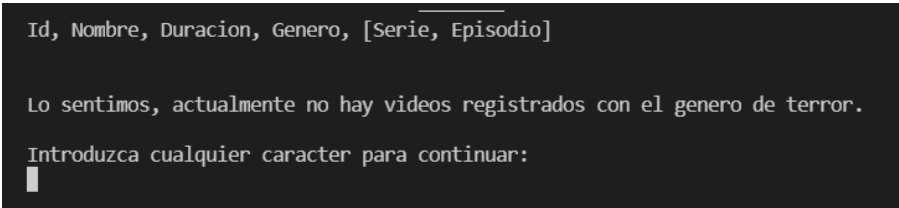
```
-----Menu para servicios de streaming-----  
1. Cargar archivo de datos  
2. Mostrar los videos en general con una cierta calificacion o de un cierto genero  
3. Mostrar los episodios de una determinada serie con una calificacion determinada  
4. Mostrar las peliculas con cierta calificacion  
5. Calificar un video  
0. Salir  
  
Escriba el numero de la opcion que desee:  
█
```

- ❖ Opción 1 del menú (cargar archivo de datos):

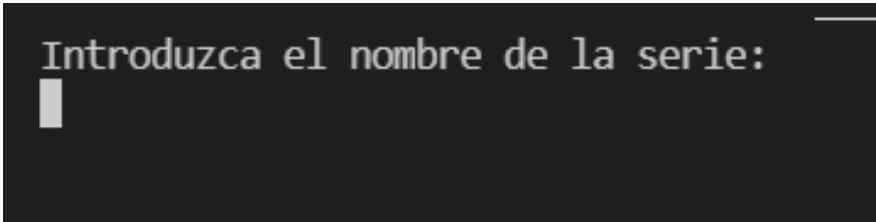
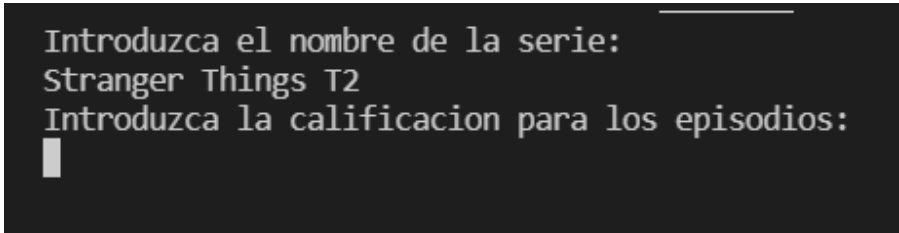
Solicitud de datos	<pre>Nombre del archivo (sin extension): █</pre>
Proceso terminado con éxito	<pre>Nombre del archivo (sin extension): data Introduzca cualquier caracter para continuar: █</pre>
Proceso terminado sin éxito	<pre>Nombre del archivo (sin extension): datos No se puede acceder al archivo. Verifique el nombre del archivo y su ubicacion. Introduzca cualquier caracter para continuar: █</pre>

❖ Opción 2 del menú (Videos en general con una cierta calificación o cierto género):

<p>Submenú</p>	<p>Desea mostrar videos basados en calificacion(1) o en genero(2)? Escriba el numero de la opcion que desee: █</p>
<p>Submenú Opción 1</p>	<p>Introduzca la calificacion: █</p>
<p>Opción 1 con éxito</p>	<pre>Id, Nombre, Duracion, Genero, [Serie, Episodio] Cap: 20160150-S02E09, Chapter Nine: The Gate, 62, accion, Stranger Things T2, 9 Cap: T1-TBYS01, El nombre del juego, 40, accion, The boys, 1 Cap: T1-TBYS02, Cherry, 42, accion, The boys, 2 Cap: S37BK8D, Piloto, 41, Crimen, Breaking bad, 1 Cap: VJSRBN574, Ya era hora, 48, accion, Invencible, 1 Cap: BKD5BM-4B, Aquí vamos, 44, accion, Invencible, 2 Cap: FR1995-T1, El intoxicado, 22, comedia, Friends T1, 15 Cap: XDJAIAJA, Internment, 43, accion, The Walking Dead T4, 5 Cap: LFR-2020, BlueBallz, 56, drama, Lucifer T5, 6 P: 20191245, The Lion King, 118, drama P: P25-TGMS, The greatest showman, 140, musical P: P27-TGBK, The green book, 210, drama P: 20191245, The Lion King, 118, drama P: A2302344, Sin límites, 200, accion P: AVSEGS933, Avengers: Endgame, 320, ciencia ficcion P: FGP1994YS, Forrest Gump, 142, drama P: JOKP334-Y, The Joker, 122, drama P: AVS-6BT5H, Who killed captain Alex, 74, accion P: 12874124A, Never Back Down, 127, accion P: JGV-124512, Avatar, 150, accion P: AZR-21412, Bay Watch, 117, drama Introduzca cualquier caracter para continuar: █</pre> <pre>Id, Nombre, Duracion, Genero, [Serie, Episodio] Lo sentimos, actualmente no hay videos registrados con una calificacion igual o mayor a 3 Introduzca cualquier caracter para continuar: █</pre>
<p>Opción 1 sin éxito</p>	<p>El valor de la calificacion debe estar entre 1 y 5, por favor intentelo de nuevo.</p> <p>Desea mostrar videos basados en calificacion(1) o en genero(2)? Escriba el numero de la opcion que desee: █</p> <p>Entrada invalida, por favor intentelo de nuevo.</p> <p>Desea mostrar videos basados en calificacion(1) o en genero(2)? Escriba el numero de la opcion que desee: █</p>

<p>Submenú Opción 2</p>	
<p>Opción 2 con éxito</p>	 

❖ Opción 3 del menú (Episodios de una determinada serie y calificación determinante):

<p>Solicitud de datos</p>	 
----------------------------------	--

<p>Proceso terminado con éxito</p>	<pre> Id, Nombre, Duracion, Genero, Serie, Episodio Cap: 20160150-S02E09, Chapter Nine: The Gate, 62, accion, Stranger Things T2, 9 Introduzca cualquier caracter para continuar: █ Id, Nombre, Duracion, Genero, Serie, Episodio Lo sentimos, actualmente no hay episodios de la serie Pokemon con una califiacion mayor o igual a 4 Introduzca cualquier caracter para continuar: █ </pre>
<p>Proceso terminado sin éxito</p>	<pre> El valor de la calificacion debe estar entre 1 y 5, por favor intentelo de nuevo. Introduzca el nombre de la serie: █ </pre>

❖ Opción 4 del menú (Películas con una cierta calificación):

<p>Solicitud de datos</p>	<pre> Introduzca la calificacion minima para las peliculas: █ </pre>
<p>Proceso terminado con éxito</p>	<pre> Id, Nombre, Duracion, Genero P: 20191245, The Lion King, 118, drama P: P25-TGMS, The greatest showman, 140, musical P: P27-TGBK, The green book, 210, drama P: 20191245, The Lion King, 118, drama P: A2302344, Sin límites, 200, accion P: AVSEGS933, Avengers: Endgame, 320, ciencia ficcion P: FGP1994YS, Forrest Gump, 142, drama P: JOKP334-Y, The Joker, 122, drama P: AVS-6BT5H, Who killed captain Alex, 74, accion P: 12874124A, Never Back Down, 127, accion P: JGV-124512, Avatar, 150, accion P: AZR-21412, Bay Watch, 117, drama Introduzca cualquier caracter para continuar: █ Id, Nombre, Duracion, Genero Lo sentimos, actualmente no hay peliculas registradas con una califiacion mayor o igual a5 Introduzca cualquier caracter para continuar: █ </pre>

Proceso terminado sin éxito	<p>El valor de la calificación debe estar entre 1 y 5, por favor intentelo de nuevo.</p> <p>Introduzca la calificación mínima para las películas:</p>
--	---

❖ Opción 5 del menú (Calificar un video):

Solicitud de datos	<p>Introduzca el titulo del video a calificar:</p> <p>Introduzca el titulo del video a calificar: The Lion King Introduzca el valor de calificación para el video:</p>
Proceso terminado con éxito	<p>Introduzca cualquier caracter para continuar:</p> <p>Lo sentimos, el video que usted intenta calificar no se encuentra registrado o tiene un nombre diferente. Introduzca cualquier caracter para continuar:</p>
Proceso terminado sin éxito	<p>El valor de la calificación debe estar entre 1 y 5, por favor intentelo de nuevo.</p> <p>Introduzca el titulo del video a calificar:</p>

❖ Opción 0 del menú (Salir):

Muchas gracias por utilizar nuestros servicios.
PS C:\Users\kayse\Documents\Tec\Programación orientada a objetos\Código\Situación Problema>

Justificaciones del código

a) Clases a utilizar:

Como ya se mencionó anteriormente, la clase Video es abstracta porque no es necesario que se instancie y funcionaría como una plantilla o modelo para sus clases hijas, las clases Episodio y Película son “tipos” de video que utilizan este modelo para funcionar, estas 3 clases son por excelencia las clases totalmente necesarias para llevar a cabo la solución del problema de cualquier manera, sin embargo, la clase MyException se utiliza con el fin de crear excepciones propias para casos específicos definidos por el programador, utilizándolas para situaciones como la verificación de datos de entrada.

Finalmente, la clase Aplicación se creo para implementar los métodos de procesamiento de datos de cada una de las opciones en el menú del sistema, si bien no es absolutamente necesario tener esta clase, implementarla otorgara una mayor claridad y organización del código al momento de modificar código o solucionar algún error en específico.

b) Herencia:

Para esta solución se utilizo el concepto de herencia con 3 de las 5 clases utilizadas, la clase de Video se estableció como padre, debido a que se trataba de una “generalización” de un objeto del cual pueden derivar otros, pero con diferentes características, es decir, las clase hija de esta fue primero la clase Película, ya que esta se trata de un tipo específico de “Video”, por lo que realizar herencia en este caso es correcto, además que utiliza la mayoría de los métodos y atributos de Video.

En el caso de la clase Episodio, esta podría ser directamente una clase hija de Video, ya que igualmente se trata de una clase de “Video”, pero con diferentes características a una película, sin embargo, se implementa directamente como hija de Película, con la finalidad de reutilizar código de los métodos que ya se implementaron en esta clase, además al hacer esto no solamente sería indirectamente hija de Video, ya que Película es hija de esta, sino que también se puede reutilizar el código de esta.

c) Modificadores de acceso:

Los modificadores de acceso de las clases utilizadas se definen principalmente dependiendo del uso que se les dará a sus atributos y métodos. Los métodos de todas las clases son públicos, debido a que de esta manera se puede acceder y utilizar la clase desde un ambiente exterior, de caso contrario la clase sería totalmente aislada del exterior y no se podría acceder a ella, por lo que prácticamente no se le podría dar un uso para la solución del problema, solamente desde clases con herencia de ella.

Los modificadores de acceso de los atributos en las clases suelen ser privados o protegidos, no públicos, ya que, de esta manera, prácticamente se tendría total acceso a su visualización, pero también a su modificación. Que sean privados o protegidos, depende principalmente de si manejan herencia o no, ya que, atributos protegidos son inaccesibles desde el exterior, pero accesibles en cuanto a visualización desde clases con herencia de esta y atributos privados son totalmente inaccesibles desde otro cualquier medio que no sea la misma clase. Es recomendable que sean privados, si no poseen herencia y si se necesita acceder a ellos de alguna manera, se hará desde un método público.

d) Sobrecarga y sobreescritura de métodos:

En esta solución se utiliza tanto la sobrecarga como la sobreescritura de métodos, la sobrecarga se empleo para la implementación de una de las opciones en el menú del sistema que requería buscar videos de 2 formas distintas, buscar por calificación o buscar por género, esta decisión se haría por el usuario, por lo que se definieron 2 métodos con el mismo nombre, pero con diferencia en sus atributos, ya que uno recibe calificación y otro recibe género, de esta manera con la elección del usuario no hace falta llamar a métodos distintos, solamente hace falta cambiar el argumento que recibe, de otra manera haría falta definir 2 métodos con nombres distintos, que puede llegar a confundir al usuario.

En cuanto a sobreescritura de métodos, se empleo entre las clases con herencia al implementar un método que poseía diferentes interpretaciones o propósitos dentro de sus clases hijas. El método `getSerie` se definió como virtual y abstracto en la clase `Video`, por lo que debe ser definido en sus clases hijas, pero a diferencia de los demás métodos, este en específico tiene diferentes propósitos en la clase `Película` y `Episodio`, ya que la clase `Película` no posee de una serie, entonces regresa un string que especifique que no aplica para esta clase, pero la clase `Episodio` si posee serie, por lo que regresa el valor almacenado en su atributo `serie`. De esta manera el método se debe de sobrescribir entre estas 2 clases.

e) Polimorfismo:

En la solución para este problema, no se lleva a cabo el polimorfismo, debido a que no se trata de una característica imprescindible y las operaciones que se pueden realizar con este concepto, no son necesarias llevarlas a cabo y puedes obtener mismos resultados con la implementación de código sin el uso de este. Principalmente debido a que no es necesario instanciar la clase padre `Video` de la herencia que posee los métodos virtuales, además por esto mismo no se recurre tampoco a un apuntador de tipo `Video` que pueda utilizar los métodos definidos por sus clases derivadas, haciendo que el polimorfismo no se lleve a cabo y por ejemplo, creando apuntadores de tipo `Película` y `Episodio` que utilizan sus propios métodos en vez de un apuntador `Video`, mejora la organización de los objetos de estos tipos en 2 arreglos diferentes, pudiendo acceder a los episodios sin necesidad de acceder a las películas.

f) Clases Abstractas:

Dentro de la herencia realizada en la solución la primera clase `Padre` se define como abstracta, debido a que esta misma clase `Video` no es necesaria para instanciarse, además de que algunos de los atributos y métodos en la estructura no llegan a tener mucho sentido o uso si se implementan en esta misma, pero si se implementa como abstracta, entonces las clases hijas pueden implementar los métodos como estas los requieran, como por ejemplo en el método `getSerie` que tiene diferentes implementaciones en las clases `Película` y `Episodio`.

g) Sobrecarga de operadores:

En el desarrollo de la solución si se implemento una sobrecarga de operador, específicamente la sobrecarga del operador `<<`, esta se realizo en las 2 clases base para el desarrollo del sistema, es decir, la clase `Película` y la clase `Episodio`, cada una de estas define el método de la sobrecarga como le

convenga dependiendo de los atributos que se deseen imprimir con un “cout”, aunque este proceso de impresión se puede realizar con un método convencional, que se defina como virtual desde la clase Video y se sobrescriba en ambas clase como les convenga, realizarlo de esta manera con la sobrecarga del operador <<, le da mayor claridad a la vista y entendimiento al código cuando se necesita realizar la impresión en pantalla de las características de uno de estos objetos Video, simplificando código y pudiendo modificar el orden de salida de los atributos de manera eficiente.

h) Excepciones:

Como una medida de prevención de errores y resultados inesperados en el sistema, se implementaron excepciones propias definidas, para los casos en los que se requerían, como la validación de entrada de los datos para continuar con el programa, o situaciones a realizar en un caso diferente a los predefinidos.

No se utilizaron excepciones predefinidas por el lenguaje, debido a la falta de entendimiento y experiencia con el uso de estos, por lo que definir excepciones propias sería la opción más viable y con menos errores. Al utilizar excepciones, en vez de verificaciones típicas con estructuras condicionales, se reduce la cantidad de código empleado y es común que las excepciones puedan funcionar en más de un caso de error.

Casos de error

El sistema está programado para que otorgue los resultados esperados solamente si se le ingresan valores correctos, es decir, no otorga resultados por default si no se le ingresan datos correctos. Sin embargo, para el caso del ingreso de valores inválidos por el usuario para continuar con el proceso, se implementaron las excepciones en la mayoría de estas situaciones, estas excepciones pueden realizar dos procesos diferentes:

- Pedir la terminación del proceso, haciendo que el usuario ingrese un carácter cualquier y regresándolo al menú principal.
- Volver a pedir los datos de búsqueda, pero continuando dentro de la opción seleccionada en el menú, sin salir al menú principal hasta que ingreses los datos correctos.

Para visualizar de mejor manera estos procesos de excepciones, mirar los apartados de “proceso terminado sin éxito” de cada opción del menú en la tercera parte del documento: Ejecución del código. (Véase página 05).

Con todo lo anterior mencionado, aunque se implementaron situaciones de excepción, aún existen algunos casos en los cuales el sistema puede fallar o dar resultados inesperados, como los siguientes:

- Por ejemplo, en el caso de ingresar un carácter cuando te piden una cantidad, es probable que el programa deje de funcionar en ese momento, por lo que intenta asegurar que el usuario introduzca al menos el mismo tipo de dato que se pide.

- No introducir un archivo para cargar en la base de datos e intentar acceder a las opciones del menú que requieren de esta base de datos, puede ocasionar errores que fuercen la terminación del programa o entregar resultados inesperados sin ningún valor para el usuario que desea utilizar el sistema.

Para evitar estos, se debe de intentar que el usuario ingrese como mínimo el mismo tipo de dato que se pide y cargar un archivo de datos sea la primera acción a realizar siempre que se inicie el sistema.

Conclusión

En base a la situación que se presentó, se implementó una solución que consistía en un sistema que apoyara a aquellos con los problemas presentados, aunque este sistema sea bastante simple y no abarque todos los aspectos de un sistema para apoyar realmente a la solución del problema, puede servir como una base para realizar un programa más especializado, que si pueda ser utilizado realmente por futuros agentes de servicios de streaming.

Con este proyecto integrador, logre reafirmar muchos de los conceptos que no tenía del todo claros, además que ayudo a seguir mejorando mi pensamiento lógico de programación, velocidad y eficiencia de implementación. En general, me pareció un buen proyecto que no te hacía implementar en código absolutamente todo lo visto durante el curso, pero si te obligaba a comprender y evaluar lo visto durante este curso para seleccionar la mejor opción considerada para la implementación de este sistema, además considerando las diferentes características o aspectos que se mencionaba en la situación que se requerían necesariamente para la evaluación.

Referencias

- Awaldraff. (2016). ESTE ES EL MAYOR PROBLEMA DE LOS SERVICIOS DE STREAMING. junio 12, 2021, de techcetera Sitio web: <https://techcetera.co/este-es-el-mayor-problema-de-los-servicios-de-streaming/>
- tutorialspoint. (2015). C++ Exception Handling. junio 12, 2021, de tutorialspoint Sitio web: https://www.tutorialspoint.com/cplusplus/cpp_exceptions_handling.htm
- Camilo, C. (2019). Polimorfismo. junio 13, 2021, de codingame Sitio web: <https://www.codingame.com/playgrounds/51214/manejo-dinamico-de-memoria-y-polimorfismo-practica-4/polimorfismo>
- Dukevin. (2011). std::cin input with spaces?. junio 12, 2021, de stackoverflow Sitio web: <https://stackoverflow.com/questions/5838711/stdcin-input-with-spaces>