

TC1030.303

Programación Orientada a Objetos

Proyecto Final

Luis Fernando De León Silva (A01754574)

Oswaldo Daniel Hernández De Luna (A01753911)

18 de Junio del 2022

Índice

Portada -----	1
Índice -----	2
Introducción -----	3
Diagrama -----	4
Ejemplo de Ejecución -----	5
Solución -----	7
Conclusiones -----	10
Referencias -----	11

Introducción

Los servicios de streaming se han convertido en una popularidad en los años recientes. Muchas personas acuden a este sistema en lugar de ir a un cine en física desde la comodidad de sus casas y lo hacen a través de un programa que ejecuta todo lo que el usuario desee para navegar dentro del mismo streaming y obtener una experiencia amigable.

Se calcula que en 2021 hubo casi 600 millones de suscriptores de streaming, el primer año completo en el que todos los grandes grupos audiovisuales se enfrentaron entre sí para captar espectadores para sus plataformas.

Dentro del mismo servicio de streaming, se encuentra el catálogo donde se almacenan todos los formatos de vídeo para que el usuario pueda reproducirlo. En este caso, ambos formatos serían los episodios y las películas. Claro, una serie se va a conformar de episodios, por lo que el usuario tendrá la posibilidad de escoger cual serie ver, y dentro de esa serie escoger una temporada y un episodio.

Y para esta situación, se debe crear un CLI (Command Line Interface) de un sistema completo de streaming que el usuario pueda acceder a ciertas películas, series, etc, pero que también tenga la posibilidad de acceder, obtener y cambiar la calificación de dichas películas, series y episodios.

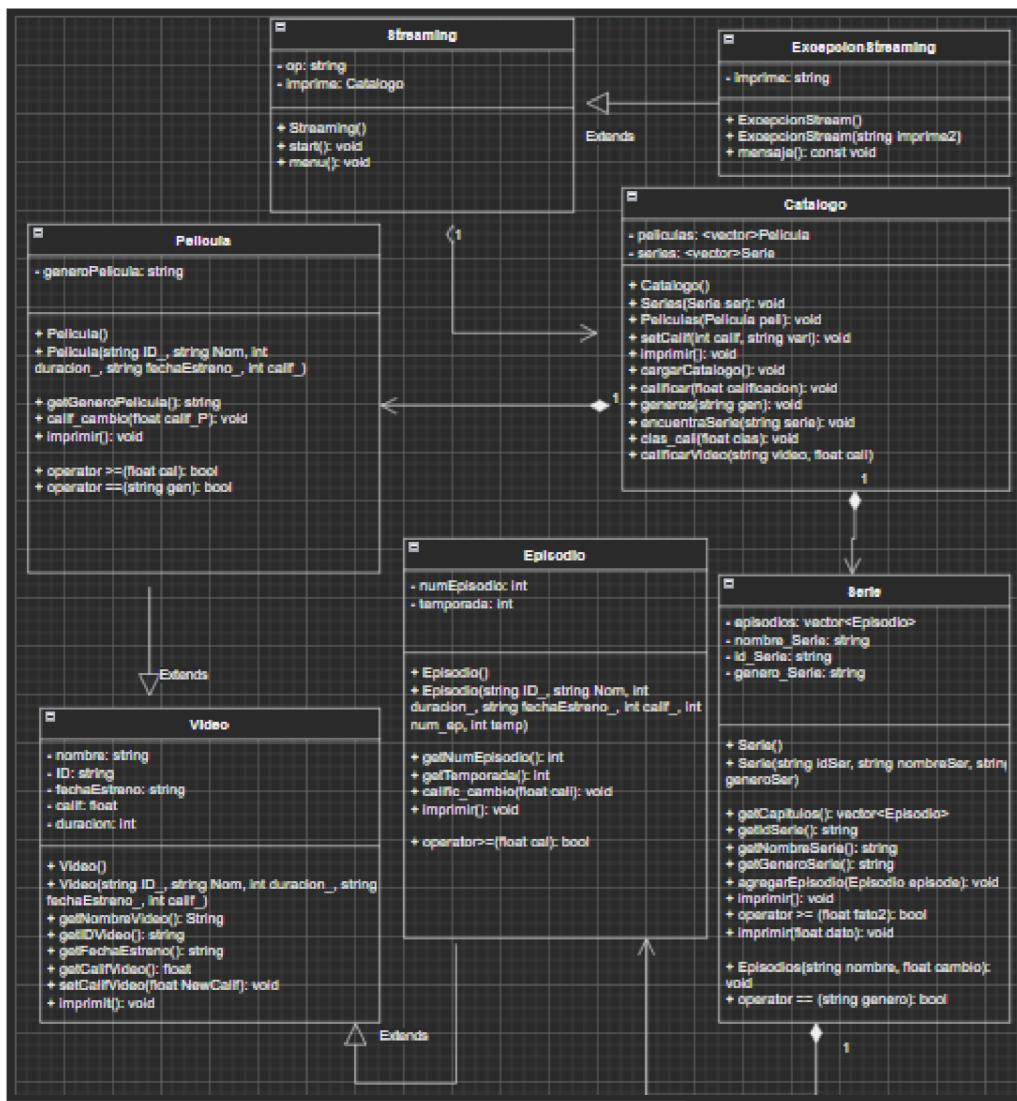
Toda esta información debe ser tomada para poder desarrollar la aplicación CLI y a su vez, ser implementada con conceptos de la Programación Orientada a Objetos en C++.

Diagrama

Las clases se representan de dicha manera en base a los atributos y métodos que se emplearon para la situación problema.

Se usó este formato de diagrama como una jerarquía de clases, esto con el fin de que se trabaje de manera más óptima y sencilla de realizar el programa. Además consideramos que las funciones propuestas son las correctas para una ejecución correcta.

Mediante los conocimientos de Herencia, Composición y Agregación, implementamos las flechas correctas para su correcto uso en cada una de ellas. Aquí, se utilizaron los conceptos principales de la POO que posteriormente, se emplearon en el código



Ejemplo de Ejecución

```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\oswad\Documents\GitHub\proyectointegrador-A01753911> ./myapp

* BIENVENIDOS A TEC-FLIX *

1. Cargar archivo de datos (Excel o csv)
2. Mostrar los videos de cierto genero
3. Mostrar los episodios de una determinada serie.
4. Mostrar las películas con una calificación mayor a cierto número
5. Calificar un video
6. Salir

Elige una opcion: 1

```

```

* BIENVENIDOS A TEC-FLIX *

1. Cargar archivo de datos (Excel o csv)
2. Mostrar los videos de cierto genero
3. Mostrar los episodios de una determinada serie.
4. Mostrar las películas con una calificación mayor a cierto número
5. Calificar un video
6. Salir

Elige una opcion: 2
1. Una calificación mayor
2. Un cierto genero
introduce la opcion que deseas saber: 1
Escoge una calificación del 1 al 10: 9

Las listas de Películas con mayor calificación:
Nombre : The Dark Knight
ID: tt0468569
Fecha Estreno: 7/18/2008
Calificación: 9
Duración: 152

*PELICULA*
Genero Pelicula: Action' Crime' Drama' Thriller

Series y Episodios con calificación mayor:
ID: tt1480055
Nombre: Game Of Thrones

```

```

*PELICULA*
Genero Pelicula: Action' Adventure' Sci-Fi
Nombre : The Danish Girl
ID: tt0810819
Fecha Estreno: 1/22/2016
Calificación: 7
Duración: 119

*PELICULA*
Genero Pelicula: Biography ' Drama' Romance
Nombre : Spider-Man: Into The Spider-Verse
ID: tt4633694
Fecha Estreno: 12/14/2018
Calificación: 8
Duración: 117

*PELICULA*
Genero Pelicula: Animation' Action' Adventure' Family' Sci-Fi

* BIENVENIDOS A TEC-FLIX *

1. Cargar archivo de datos (Excel o csv)
2. Mostrar los videos de cierto genero
3. Mostrar los episodios de una determinada serie.
4. Mostrar las películas con una calificación mayor a cierto número
5. Calificar un video
6. Salir

Elige una opcion: []

```

```

PS C:\Users\oswad\Documents\GitHub\proyectointegrador-A01753911> ./myapp

* BIENVENIDOS A TEC-FLIX *

1. Cargar archivo de datos (Excel o csv)
2. Mostrar los videos de cierto genero
3. Mostrar los episodios de una determinada serie.
4. Mostrar las películas con una calificación mayor a cierto número
5. Calificar un video
6. Salir

Elige una opcion: 1
Series:

ID: tt1480055
Nombre: Game Of Thrones
Genero: Action' Adventure' Drama' Fantasy' Romance
Episodios:
Nombre : Winter Is Coming
ID: tt1668899
Fecha Estreno: 04/17/2011
Calificación: 9
Duración: 57
*EPISODIO*
Número Episodio: 1
Temporada: 1

Nombre : The Kingsroad
ID: tt1668746

```

```

* BIENVENIDOS A TEC-FLIX *

1. Cargar archivo de datos (Excel o csv)
2. Mostrar los videos de cierto genero
3. Mostrar los episodios de una determinada serie.
4. Mostrar las películas con una calificación mayor a cierto número
5. Calificar un video
6. Salir

Elige una opcion: 2
1. Una calificación mayor
2. Un cierto genero
introduce la opcion que deseas saber: 1

```

```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

1. Cargar archivo de datos (Excel o csv)
2. Mostrar los videos de cierto genero
3. Mostrar los episodios de una determinada serie.
4. Mostrar las películas con una calificación mayor a cierto número
5. Calificar un video
6. Salir

Elige una opcion: 2
1. Una calificación mayor
2. Un cierto genero
introduce la opcion que deseas saber: 2
Genero disponibles -> las opciones son:

Action
Adventure
Drama
Fantasy
Romance
Sci-Fi
Animation
Family
Musical
War
Crime
Mystery
Thriller
Comedy
Western
Biography
Horror
¿Qué deseas buscar? introduce aqui: Action

```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

```

Nombre : The Dark Knight
ID: tt0468569
Fecha Estreno: 7/18/2008
Calificacion: 9
Duracion: 152

*PELICULA*
Genero Pelicula: Action' Crime' Drama' Thriller
Nombre : Hancock
ID: tt0448157
Fecha Estreno: 02/07/2008
Calificacion: 6
Duracion: 92

*PELICULA*
Genero Pelicula: Action' Fantasy
Nombre : Pacific Rim
ID: tt1663662
Fecha Estreno: 12/07/2013
Calificacion: 6
Duracion: 131

*PELICULA*
Genero Pelicula: Action' Adventure' Sci-Fi
Nombre : The Danish Girl
ID: tt0810819
Fecha Estreno: 1/22/2016
Calificacion: 7
Duracion: 119

*PELICULA*

```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

```

* BIENVENIDOS A TEC-FILX *

1. Cargar archivo de datos (Excel o csv)
2. Mostrar los videos de cierto genero
3. Mostrar los episodios de una determinada serie.
4. Mostrar las peliculas con una calificacion mayor a cierto numero
5. Calificar un video
6. Salir

Elige una opcion: 3
¿Que serie quieres que te mostremos?
The Big Bang Theory

ID: tt0898266
Nombre: The Big Bang Theory
Genero: Comedia' Romance
Episodios:
Nombre : The Emotion Detection Automation
ID: tt6414316
Fecha Estreno: 02/02/2017
Calificacion: 7
Duracion: 19
*EPISODIO*
Numero Episodio: 14
Temporada: 10

Nombre : The Lizard-Spock Expansion
ID: tt1256039
Fecha Estreno: 01/05/2006

```

* BIENVENIDOS A TEC-FILX *

1. Cargar archivo de datos (Excel o csv)
2. Mostrar los videos de cierto genero
3. Mostrar los episodios de una determinada serie.
4. Mostrar las peliculas con una calificacion mayor a cierto numero
5. Calificar un video
6. Salir

Elige una opcion: 5
 Ingresa el nombre del capitulo o pelicula que te gustaria calificar:
 Stranger Things
 Ingresa la calificacion que quieres darle:
 9

* BIENVENIDOS A TEC-FILX *

1. Cargar archivo de datos (Excel o csv)
2. Mostrar los videos de cierto genero
3. Mostrar los episodios de una determinada serie.
4. Mostrar las peliculas con una calificacion mayor a cierto numero
5. Calificar un video
6. Salir

Elige una opcion:

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

* BIENVENIDOS A TEC-FILX *

1. Cargar archivo de datos (Excel o csv)
2. Mostrar los videos de cierto genero
3. Mostrar los episodios de una determinada serie.
4. Mostrar las peliculas con una calificacion mayor a cierto numero
5. Calificar un video
6. Salir

Elige una opcion: 4
 ¿Desde que calificacion quieres que te muestre las peliculas (maximo es 10)?
 8
 Peliculas con igual o mayor calificacion
 Nombre : Jurassic Park
 ID: tt0107290
 Fecha Estreno: 11/06/1993
 Calificacion: 8
 Duracion: 127

PELICULA
Genero Pelicula: Action' Adventure' Sci-Fi' Thriller
Nombre : The Avengers
ID: tt08848228
Fecha Estreno: 04/05/2012
Calificacion: 8
Duracion: 143

PELICULA
Genero Pelicula: Action' Adventure' Sci-Fi
Nombre : Django Unchained

* BIENVENIDOS A TEC-FILX *

1. Cargar archivo de datos (Excel o csv)
2. Mostrar los videos de cierto genero
3. Mostrar los episodios de una determinada serie.
4. Mostrar las peliculas con una calificacion mayor a cierto numero
5. Calificar un video
6. Salir

Elige una opcion: 6
 Hasta la proxima
 PS C:\Users\oswad\Documents\GitHub\proyectointegrador-A01753911>

Solución

Como se mencionó anteriormente, la situación problema es desarrollar una aplicación de comandos de línea (CLI), que será una plataforma de streaming y los conceptos básicos de la POO a emplear en este proyecto son la encapsulación, la abstracción, herencia, polimorfismo, pero también se emplea el uso de excepciones y sobrecarga de operadores.

Como parte de la solución, las clases de la aplicación fueron presentadas antes de realizar el diagrama UML. Sin embargo, en dichas clases se declararon tanto los atributos privados, como los constructores, métodos, getters, setters, sobrecargas, etc. Cabe mencionar que dentro de la implementación de los conceptos de POO, hubo ciertas modificaciones que se tuvieron que realizar para agregar estos conceptos a nuestro diagrama y posteriormente a nuestro código.

La funcionalidad del proyecto final (clases) funciona de la siguiente manera: Las clases episodio y película heredan de la clase video, ya que se toma en cuenta solamente que episodio y película son formatos de video. Ahora bien, la clase serie se compone de episodios, por lo que la clase catálogo va a ser la encargada de mostrarle al usuario todas las películas y series junto con sus respectivos episodios que hay en existencia. Hay que tener en cuenta que todas las películas y series se encuentran en una base de datos de un archivo .csv, por lo que la clase catálogo también es la encargada de leer este archivo utilizando la función getline() para poder leer strings como flujo de entrada.

Ahora bien, la clase streaming es el menú de todo el programa. El usuario puede interactuar con el programa y decidir qué acción realizar. Para esto, se tuvieron que llamar las funciones declaradas en catálogo para imprimir el mismo, o si bien, si el usuario sólo desea ver las películas, una serie o película en específico, agregar y/o ver la calificación de dicho video, etc. Dentro de este menú se utilizaron excepciones, para que en caso de que el

usuario errara una de las opciones a escoger y regresar al menú principal. Para esto, se tuvo que hacer otra clase extra llamada excepcionStream, que es la encargada de lanzar y atrapar el error en caso de que el usuario coloque una opción incorrecta. Con esto, se evitaba la necesidad de utilizar recursiones en las funciones. Y por último, el main, que se encargará de empezar el programa con un método start().

Ahora, dentro de streaming, lo que se hizo fue hacer una excepción para el menú, como se comentó anteriormente, pero dentro del menú se encuentra la función de cambiar una calificación de un título que el usuario desee. En esta ocasión, también se decidió usar excepciones para mejorar la interactividad con el usuario y así ayudarlo a entender mejor el programa. Si el usuario coloca una calificación incorrecta, el método le pedirá de nuevo que ingrese una calificación correcta. Para herencia y polimorfismo, la clase video es la clase que va a heredar dichas instancias a película y episodio, y también se considera una clase abstracta. Por último, los operadores == y >= se sobrecargan tanto en película como en serie para encontrar el video que el usuario desea.

Optamos por estas soluciones debido a que son las maneras más prácticas de realizar un código debido a que nos evitamos de estar declarando variables con vectores, se pudo optimizar todo lo que más se pudo, asimismo heredar estas clases nos ayuda mucho en el reciclaje de variables, métodos, etc.

El sobrecargar los operadores hace que no tengamos que hacer códigos innecesarios además de que es práctico ya que con la misma instrucción puedes imprimir diferentes tipos de cosas con polimorfismo.

Identificación de casos que harían que el proyecto deje de funcionar

Los casos que harían que el proyecto deje de funcionar es el siguiente:

Si se mueve una variable del código en específico si es la clase video, esta hará que tanto la clase de película como serie se vean afectadas, e incluso en la clase catalogo los token no reconoceran las variables y provocará que existan al menos 3 o 4 clases que puedan ser detectadas como error, provocando que ni siquiera compile en la terminal.

Otro de los errores o haría que el código deje de funcionar es si en la clase de excepciones que nosotros agregamos, es que si las variables no son determinadas correctamente, provocará que la clase catalogo que es una de las más importantes dentro del proyecto integrador, marqué más de 9 errores y esto se debe a las excepciones que nosotros implementamos dentro de esta.

Muchos pueden ser los casos por los cuales el código o el programa deje de funcionar pero se considera que estos dos son los principales porque provocaría un programa lleno de errores.

CONCLUSIONES

Con este proyecto nos pudimos dar cuenta que es importante tener los conocimientos de herencia, polimorfismo, sobrecarga, excepciones, ya que son prácticos para cualquier programa de un nivel más complejo, además de que, podemos observar cómo todos estos temas pueden lograr crear una pequeña base de datos simulando un menú de una plataforma de streaming, lo cual es un buen ejemplo para poner en práctica todo lo visto en estas 5 semanas del curso, a pesar de tener algunos tropiezos durante la elaboración del proyecto, considero que gracias a esto, puedo decir que tengo un poco más de conocimiento tanto de herencia como de polimorfismo y de cómo implementarlos.

- Oswaldo Daniel Hernández De Luna

Este proyecto es un claro ejemplo de cómo usar los conceptos aprendidos básicos de la POO, concretamente en C++, pero sobre todo afinar dichos conocimientos creando una CLI totalmente funcional para los usuarios y gracias a esta situación problema, se pudieron adquirir ciertos conocimientos y reforzarlos, para así volver a aplicarlos en un nuevo proyecto. Inclusive hasta poder mejorar el cómo implementar todos los conceptos revisados y hacer una optimización de código, pero esto se podrá lograr mediante la práctica.

- Luis Fernando De León Silva

Referencias:

Cáceres, I. (2022, enero 15). *Netflix podría comprar una cadena de cines, Apple un gran estudio para impulsar sus contenidos y otras 3 predicciones para 2022 de los expertos en entretenimiento*. Business Insider España.

<https://www.businessinsider.es/tendencias-guerra-streaming-2022-expertos-993409>

Leer archivo (.txt) C++. (s/f). Stack Overflow en español. Recuperado el 15 de junio de 2022, de

<https://es.stackoverflow.com/questions/63961/leer-archivo-txt-c>

(Revisamos conceptos de como leer archivos en c++ , ya que nos marcaba errores constantes en poder abrir un archivo csv)