Texto

Descripción generada automáticamente con confianza baja

**Situación problema: modelado de servicio de streaming**

Mariana Balderrábano Aguilar A01749581

17 de junio de 2023

Programación Orientada a objetos

Roberto Martínez Román

Índice

[Introducción 3](#_Toc137930866)

[Diagrama de clases 3](#_Toc137930867)

[Ejemplos de ejecución 4](#_Toc137930868)

[Identificación de casos que harían que el proyecto deje de funcionar. 10](#_Toc137930869)

[Conclusión personal 10](#_Toc137930870)

[Referencias 12](#_Toc137930871)

# Introducción

Con el paso del tiempo, las plataformas de videos para ver series o películas (de streaming) han crecido exponencialmente, de acuerdo con BBVA, “El crecimiento de plataformas como Netflix, Twitch, Amazon Prime en los últimos años ha alcanzado tal punto que llegan a representar un 25% del mercado televisivo” (Plataformas de Streaming: ¿Cómo Ha Sido Su Crecimiento Con El Pasar de Los Años?, 2022). Es por ello por lo que este proyecto se centra en el desarrollo de un programa en el lenguaje de c++ con ayuda de programación orientada a objetos. Modelar este tipo de plataformas nos ayudará en un futuro a poder seguir innovando, a su vez nos podría ayudar a hacer más eficientes los procesos de los códigos y de esta manera finalmente brindar la mejor calidad de plataformas de streaming a los usuarios.

# Diagrama de clases

Imagen que contiene Diagrama

Descripción generada automáticamente

Figura 1. Diagrama de clases en lenguaje UML

En la Figura 1 podemos apreciar que para este proyecto se decidió trabajar con tres grandes clases, las cuales fueron, la clase Video, Película y Episodio.

La clase Video es la superclase o clase base, es decir que la clase Película y Episodio heredarán más adelante los atributos y métodos de la clase Video, esta misma clase contiene los atributos en común en todos los videos (película y episodio). En la parte de la superclase se definieron los atributos que describen a un objeto de tipo Video, es por ello que tenemos el id, título o nombre del video, el género, la calificación y la duración del video, en este caso, todos los atributos mencionados anteriormente son de tipo string, posteriormente tenemos a los getters (nos ayudarán a obtener el string de los atributos y en el caso de getCaliDouble, nos regresará la calificación en tipo double) y setCalificacion, que nos ayuda a definir el valor de calificación con la calificación dada en los parámetros, así como el constructor de Video con todos los atributos dentro del mismo y tenemos al método abstracto ImprimeDatos, lo cual hace que la clase Video, sea una clase abstracta. Más adelante, el último método mencionado se sobrescribirá de acuerdo a su tipo de dato.

Posteriormente tenemos la subclase Película, la cual dentro del recuadro de atributos no tiene nada, puesto que todos los atributos que necesita ya han sido heredados de la clase Video, por la parte de los métodos, tenemos el constructor con todos los atributos dentro, así como el getCalificacion que nos regresará el valor de la calificación con el tipo de dato double y por último ImprimeDatos, que es un método sobrescrito que nos ayudará a imprimir los datos, en este caso de la Película con un formato específico y de esta forma usamos polimorfismo.

Por otra parte, tenemos a la subclase Episodio, la cual heredó de Video, por lo que tendrá esos datos, pero de igual manera necesita más datos, por lo que en los atributos agregué idE, el cual es el id del episodio, así como tituloE, título del episodio, también temporada y numEp (número de episodios de la serie), todo con tipo de dato string. Por la parte de los métodos, tenemos el constructor Episodio, el cual tiene dentro los datos necesarios, los cuales son los atributos heredados de Video y los descritos anteriormente (id del episodio, título del episodio, temporada de la serie y número de episodios). Asimismo, tenemos el método imprimeDatos que es sobrescrito para que imprima los datos de tipo Episodio de una serie con formato y de esta manera usar polimorfismo. Finalmente, tenemos el método getTituloE, el cual nos regresará el título del episodio.

# Ejemplos de ejecución

1.- Carga el archivo csv

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

2.- Muestra videos con calificación mayor a la dada

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente  
  
2.- Muestra videos con el género dado

Interfaz de usuario gráfica, Texto, Aplicación

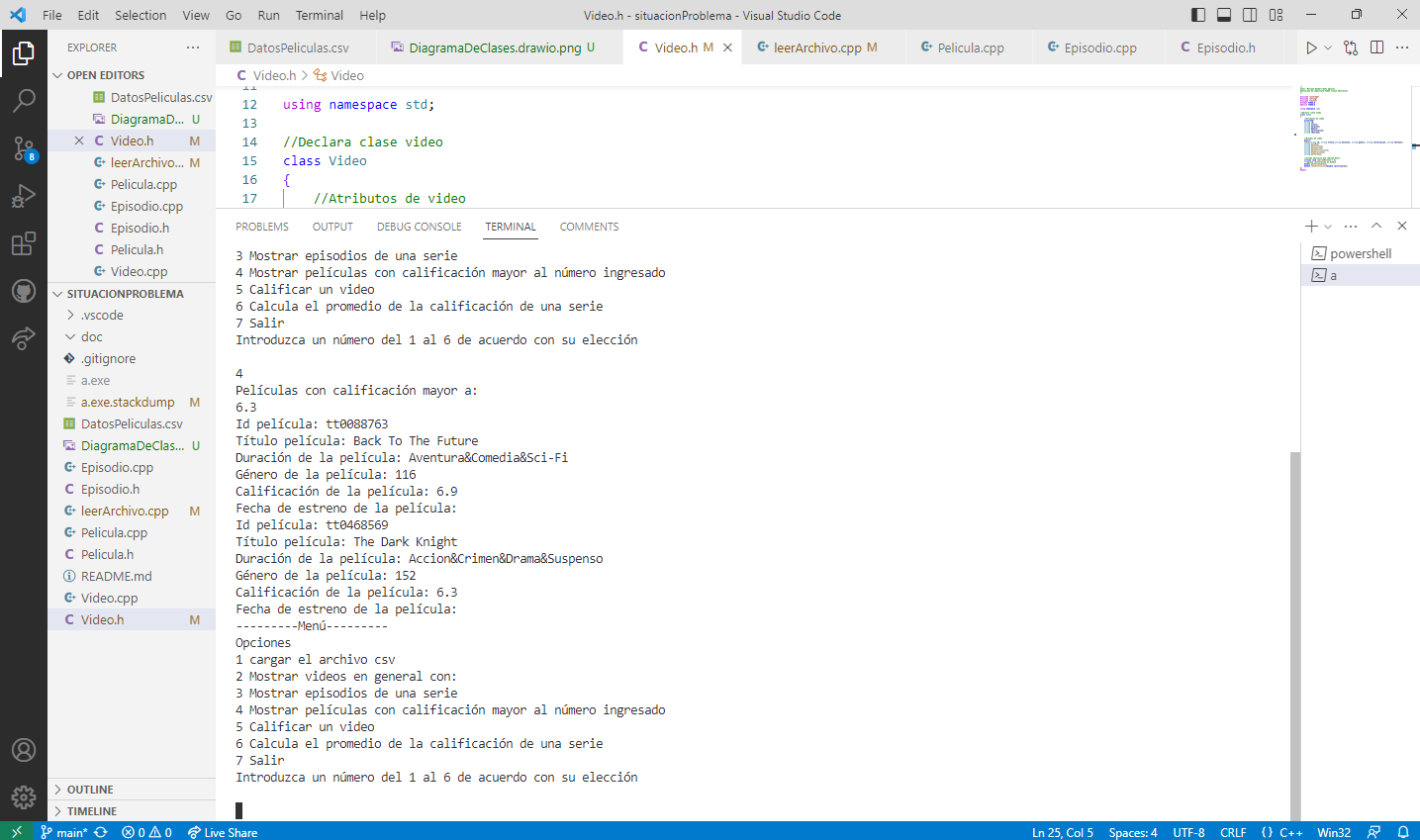
Descripción generada automáticamente

3.- Muestra episodios de una serie

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

4.- Muestra películas con una calificación mayor o igual a la dada



5.- Califica un video y define la nueva calificación

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente6.- Calcula el promedio de una serie y la regresa

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

7.- Salir

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Cuando un argumento no es válido, regresa a menú.

# Identificación de casos que harían que el proyecto deje de funcionar.

Algunos de los casos que harían que el proyecto deje de funcionar sería que el orden del archivo csv cambiara, que cambiara el número de datos que tiene un episodio y una película, pues no se logrará distinguir un episodio de una película, que existieran más valores por separar, que el usuario escribiera mal el género o lo escribiera con todas mayúsculas o todas minúsculas, que pusieran algún espacio de más al ingresar los títulos de las películas o ponerle acento, pues puede que en el documento csv no tengan acento.

# Conclusión personal

Esta situación problema me ayudó a mejorar mis habilidades de programación, con ella pude distinguir cosas que se me dificultan y de igual manera reforzar lo aprendido. En lo personal, la lectura de archivos se me complicó, gracias a la actividad noté que necesito practicarlo más, me pareció interesante la solución propuesta por mí, asimismo, esta actividad me desafió, pues a veces producía errores que tuve que investigar para poder resolver, gracias a esta actividad pude mejorar mi habilidad para leer y comprender los errores para poder así corregirlos. El diagrama de clases en lenguaje UML me pareció de suma importancia hacerlo al inicio, ya que me ayudó demasiado a poder tener una idea más clara del código, pero como es inevitable, a lo largo de la actividad se tuvieron que hacer algunas modificaciones. Me sorprendió mucho el avance que pude tener de primer semestre a este segundo semestre ya que pude comprender mejor los códigos y de igual forma logré entender muchas cosas que no podía el semestre pasado de programación orientada a objetos, sinceramente para mí representó un logro personal poder desarrollar este código de manera individual, ya que el semestre pasado, estoy casi segura de que no lo hubiera logrado yo sola. Gracias a la actividad entendí la importancia del diagrama de clases, pude aplicar los conocimientos aprendidos en clase y logré obtener más conocimiento gracias a los errores que se generaban a lo largo del código.

# Referencias

DuHadway, L. (2020). The string npos static constant [YouTube Video]. In YouTube. <https://www.youtube.com/watch?v=BqnXcsZ7-6w>

fajardo, eldonedgar. (2020). 02 Progamación C++ -- Getline Cin.Ignore [YouTube Video]. In YouTube. <https://www.youtube.com/watch?v=3bbZkofM594>

Luis Cabrera Benito. (2021). Leer archivo CSV con C++ [YouTube Video]. In YouTube. <https://www.youtube.com/watch?v=bal7cn-YvfU>

Plataformas de streaming: ¿Cómo ha sido su crecimiento con el pasar de los años? (2022, July 15). BBVA.CH. <https://www.bbva.ch/noticia/plataformas-de-streaming-como-ha-sido-su-crecimiento-con-el-pasar-de-los-anos/>

Programación ATS. (2016). 12. Programación en C++ || Condicionales || La sentencia switch [YouTube Video]. In YouTube. <https://www.youtube.com/watch?v=hjDB7ev9ePM>

The Cherno. (2019). Dynamic Casting in C++ [YouTube Video]. In YouTube. <https://www.youtube.com/watch?v=CiHfz6pTolQ>

‌

‌