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 2](#_Toc137930866)

[Diagrama de clases 2](#_Toc137930867)

[Ejemplos de ejecución 2](#_Toc137930868)

[Identificación de casos que harían que el proyecto deje de funcionar. 2](#_Toc137930869)

[Conclusión personal 2](#_Toc137930870)

[Referencias 2](#_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 para finalmente brindar mejor calidad de plataformas de streaming a los usuarios.

# Diagrama de clases

Interfaz de usuario gráfica, Sitio web

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, contiene los atributos em 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 del video, el título o nombre del video, el género del video, la calificación del video y la duración del video, en este caso, todos los atributos mencionados anteriormente son de tipo string pues fueron obtenidos de leer un archivo, posteriormente tenemos al vector, este es un vector de punteros a objetos de tipo Video. Posteriormente contamos con los métodos, los cuales incluyen getters (nos ayudarán a obtener el string de los atributos y en el caso de getCaliDouble, nos regresará la calificación en tipo double) y setters (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.

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 sobreescrito que nos ayudará a imprimir los datos, en este caso de la Película con un formato específico y de esta forma utilizar 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, el cual es el título del episodio, también temporada y numEp, que son el 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 de la serie, 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) y asimismo tenemos el método imprimeDatos que es sobreescrito para que imprima los datos de tipo Episodio de una serie con formato y de esta manera usar polimorfismo.

# Ejemplos de ejecución

1.- Carga el archivo csv y despliega el contenido

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Al concluir regresa al menú

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

6.-Al salir, ya no muestra el menú

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Cuando ingresa un número no válido, imprime argumento no válido y muestra el menú una vez más.

Una captura de pantalla de una computadora

Descripción generada automáticamente

# 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 tienen un episodio de una película, pues no se logrará distinguir un episodio de una pelicula, que existieran más valores por separar.

# 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, pues considero practicarlo más, de igual forma no logré modificar un dato directo en el archivo csv. Asimismo, encontré desafiante implementar vectores, considero que es algo que necesito practicar más. Por otra parte, logré darme cuenta de la importancia de hacer un diagrama de clases, pues finalmente es tu guía y a pesar de que se modificaba con el tiempo, considero que, si no hubiera creado el diagrama de clases, no hubiera podido implementar todo lo necesario.

# Referencias

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/>

‌