Modelado de Servicio de Streaming

Allan Mauricio Brenes Castro

TC1030

Alma Patricia Fraga Mendoza

Campus CEM

Jun – Jul 2024

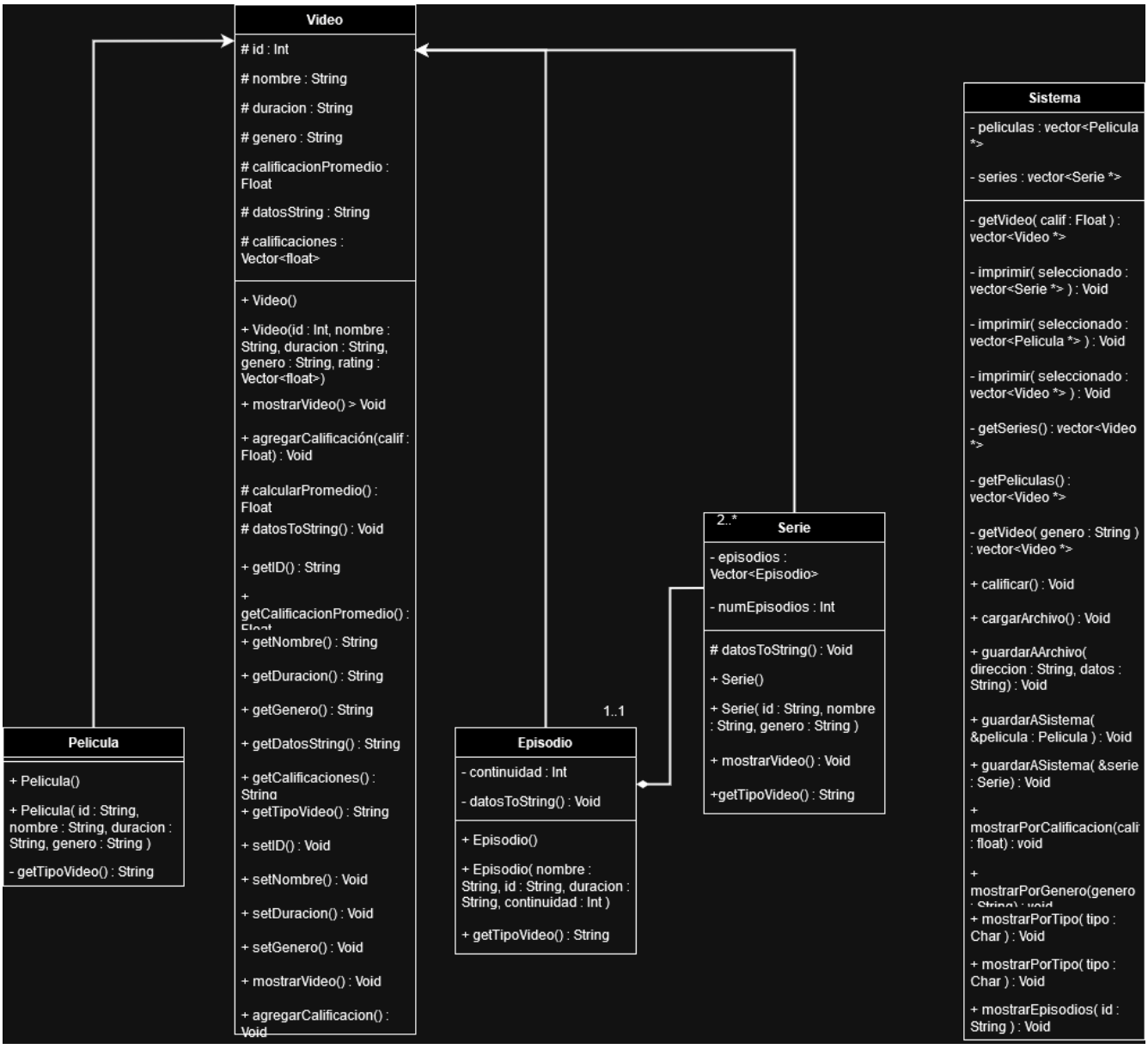
Introducción

Los servicios de streaming son importantes ahora más que nada. Llegaron para ser una alternativa para la televisión por cable, y se quedaron por ser mucho más convenientes. Estos servicios tienen como característica un orden impecable. Necesitan representar diferente información, como los tipos de contenido, cada uno con diferentes géneros, muestran información también del video, resumen, aparte del video en cuestión , los diferentes audios y subtítulos, entre muchas cosas.

En este proyecto, se modela un servicio de streaming bastante primitivo a comparación de los servicios de streaming en el mercado, ya que sólo se muestra el contenido que tiene y algo de información, como su nombre, género, ID, duración, etc. Se plantea aplicar lo aprendido en la materia TC1030, una materia enfocada en la programación orientada a objetos, donde se aprendieron cosas como el polimorfismo, herencia, sobrecarga de operadores, clases abstractas, entre otras cosas.

En este proyecto, se solicitó crear una clase base Video que tiene tres hijos: Pelicula, Episodio y Serie. Los videos deben de tener atributos de nombre, id, duración y género. Las clases hijo pueden tener también atributos que no tengan otras clases, como es el caso de Serie y Episodio. Serie tiene una lista de episodios, y Episodio tiene una continuidad en relación a sus otros episodios. Se debe de poder ver los detalles de cada video, ver videos de una calificación determinada o un género en específico y ver los episodios de una serie.

Diagrama de clases UML



El diseño se enfocó en que la clase base, Video, sea la que tenga la mayoría de los métodos y atributos. Esto se refleja cuando se comparan las similitudes entre un video y una película. De los atributos que tengan relevantes al proyecto, son muy similares. Tanto, que no se encontró un atributo

extra para la clase Pelicula. De haber pedido más detalles, se podría haber incluido atributos como “director”, “Productores”, “Escritores”, etc.

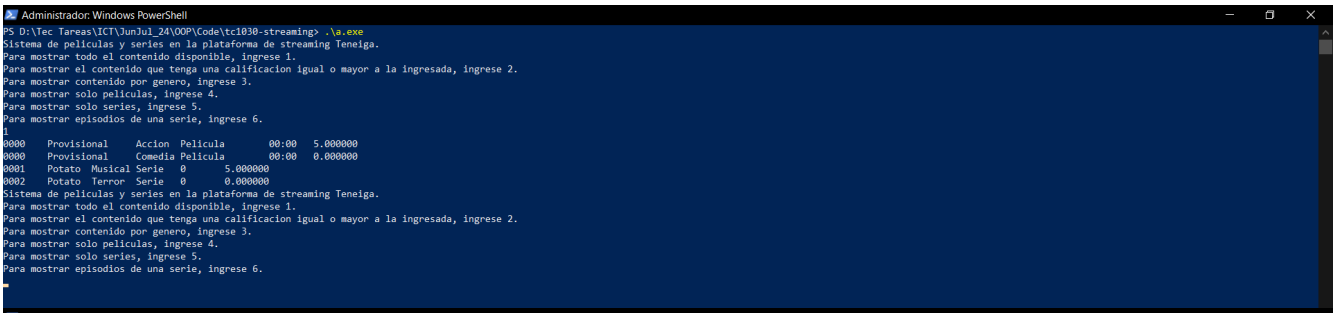
La mayoría de los métodos de Video son setters y getters, lo cual tiene sentido, ya que es la clase base y tiene más atributos que sus hijos.

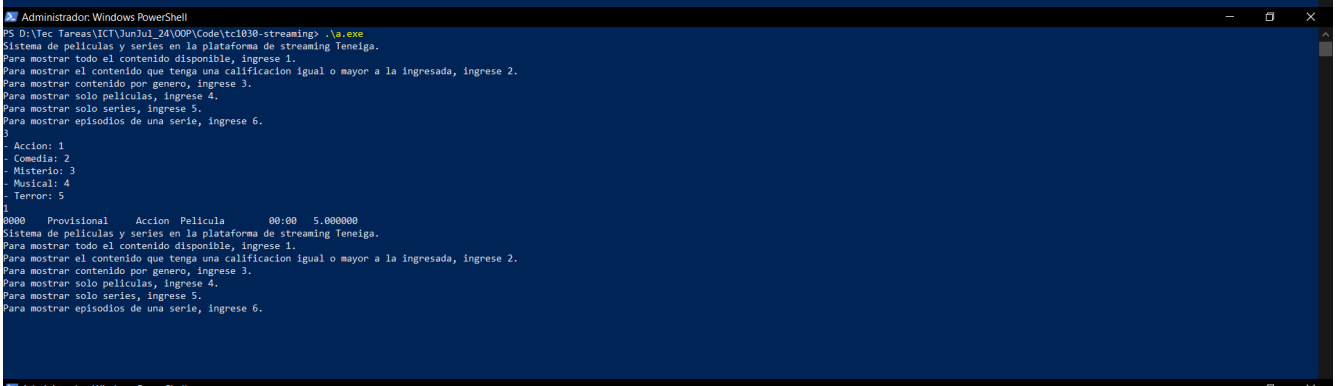
Serie y Episodio se consideró que tienen atributos relevantes que no se encuentran en la clase base, como continuidad, en el caso de la clase Episodio, y episodios, en el caso de la clase Serie. Aunque una serie y un video son muy diferentes, se optó por permitir que Serie herede de Video por sus similitudes entre atributos solicitados.

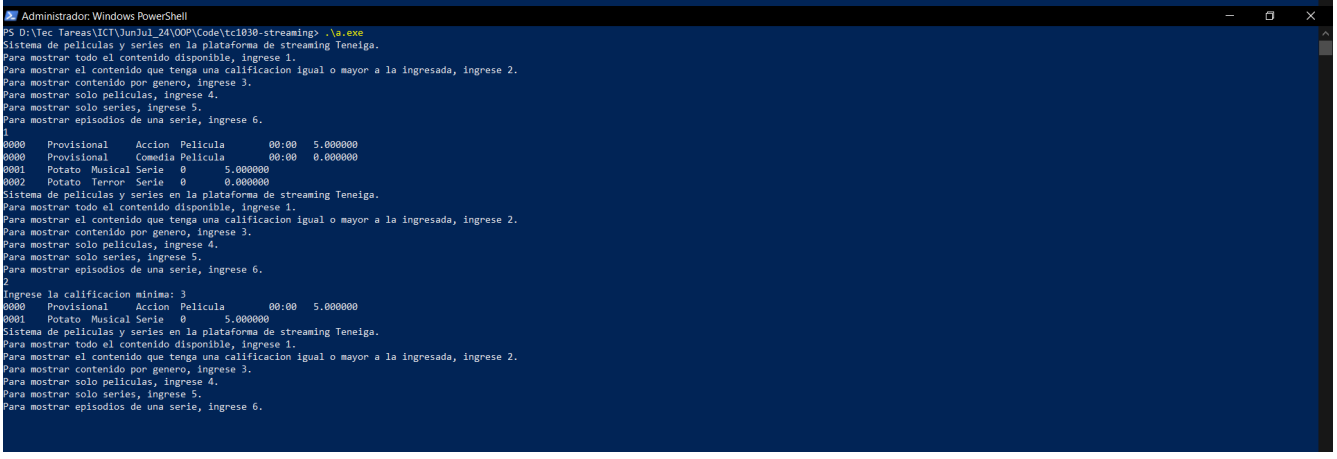
Para la lógica del programa se decidió usar una clase maestra llamada Sistema. Esta clase controla el programa y llama a las clases de Video y sus hijos. La mayoría de los miembros de esta clase se conforman de métodos, lo que tiene sentido considerando su propósito. Esta clase permite que main.cpp esté lo más limpio posible.

Ejecución


Administrador: Windows PowerShell
PS D:\Tec Tareas\ICT\JunJul_24\OOP\Code\tc1030-streaming> .\a.exe
Sistema de peliculas y series en la plataforma de streaming Teneiga.
Para mostrar todo el contenido disponible, ingrese 1.
Para mostrar el contenido que tenga una calificacion igual o mayor a la ingresada, ingrese 2.
Para mostrar contenido por genero, ingrese 3.
Para mostrar solo peliculas, ingrese 4.
Para mostrar solo series, ingrese 5.
Para mostrar episodios de una serie, ingrese 6.


Administrador: Windows PowerShell
PS D:\Tec Tareas\ICT\JunJul_24\OOP\Code\tc1030-streaming> .\a.exe
Sistema de peliculas y series en la plataforma de streaming Teneiga.
Para mostrar todo el contenido disponible, ingrese 1.
Para mostrar el contenido que tenga una calificacion igual o mayor a la ingresada, ingrese 2.
Para mostrar contenido por genero, ingrese 3.
Para mostrar solo peliculas, ingrese 4.
Para mostrar solo series, ingrese 5.
Para mostrar episodios de una serie, ingrese 6.
1
0000 Provisional Accion Pelicula 00:00 5.000000
0000 Provisional Comedia Pelicula 00:00 0.000000
0001 Potato Musical Serie 0 5.000000
0002 Potato Terror Serie 0 0.000000
Sistema de peliculas y series en la plataforma de streaming Teneiga.
Para mostrar todo el contenido disponible, ingrese 1.
Para mostrar el contenido que tenga una calificacion igual o mayor a la ingresada, ingrese 2.
Para mostrar contenido por genero, ingrese 3.
Para mostrar solo peliculas, ingrese 4.
Para mostrar solo series, ingrese 5.
Para mostrar episodios de una serie, ingrese 6.


Administrador: Windows PowerShell
PS D:\Tec Tareas\ICT\JunJul_24\OOP\Code\tc1030-streaming> .\a.exe
Sistema de peliculas y series en la plataforma de streaming Teneiga.
Para mostrar todo el contenido disponible, ingrese 1.
Para mostrar el contenido que tenga una calificacion igual o mayor a la ingresada, ingrese 2.
Para mostrar contenido por genero, ingrese 3.
Para mostrar solo peliculas, ingrese 4.
Para mostrar solo series, ingrese 5.
Para mostrar episodios de una serie, ingrese 6.
3
- Accion: 1
- Comedia: 2
- Misterio: 3
- Musical: 4
- Terror: 5
1
0000 Provisional Accion Pelicula 00:00 5.000000
Sistema de peliculas y series en la plataforma de streaming Teneiga.
Para mostrar todo el contenido disponible, ingrese 1.
Para mostrar el contenido que tenga una calificacion igual o mayor a la ingresada, ingrese 2.
Para mostrar contenido por genero, ingrese 3.
Para mostrar solo peliculas, ingrese 4.
Para mostrar solo series, ingrese 5.
Para mostrar episodios de una serie, ingrese 6.


Administrador: Windows PowerShell
PS D:\Tec Tareas\ICT\JunJul_24\OOP\Code\tc1030-streaming> .\a.exe
Sistema de peliculas y series en la plataforma de streaming Teneiga.
Para mostrar todo el contenido disponible, ingrese 1.
Para mostrar el contenido que tenga una calificacion igual o mayor a la ingresada, ingrese 2.
Para mostrar contenido por genero, ingrese 3.
Para mostrar solo peliculas, ingrese 4.
Para mostrar solo series, ingrese 5.
Para mostrar episodios de una serie, ingrese 6.
1
0000 Provisional Accion Pelicula 00:00 5.000000
0000 Provisional Comedia Pelicula 00:00 0.000000
0001 Potato Musical Serie 0 5.000000
0002 Potato Terror Serie 0 0.000000
Sistema de peliculas y series en la plataforma de streaming Teneiga.
Para mostrar todo el contenido disponible, ingrese 1.
Para mostrar el contenido que tenga una calificacion igual o mayor a la ingresada, ingrese 2.
Para mostrar contenido por genero, ingrese 3.
Para mostrar solo peliculas, ingrese 4.
Para mostrar solo series, ingrese 5.
Para mostrar episodios de una serie, ingrese 6.
2
Ingrese la calificacion minima: 3
0000 Provisional Accion Pelicula 00:00 5.000000
0001 Potato Musical Serie 0 5.000000
Sistema de peliculas y series en la plataforma de streaming Teneiga.
Para mostrar todo el contenido disponible, ingrese 1.
Para mostrar el contenido que tenga una calificacion igual o mayor a la ingresada, ingrese 2.
Para mostrar contenido por genero, ingrese 3.
Para mostrar solo peliculas, ingrese 4.
Para mostrar solo series, ingrese 5.
Para mostrar episodios de una serie, ingrese 6.

Argumentación

En la situación problema se solicitan cuatro clases diferentes:
La clase base Video, y tres clases que heredan de Video (Pelicula, Episodio y Serie).
En el programa se usan esas mismas tres clases y tienen los atributos solicitados (como nombre, id, duracion, genero, etc.) Esto, junto con la herencia solicitada de la clase base, cumple con los requisitos. En este ámbito no había de otra sin romper con lo que plantea la situación problema.

Los modificadores de acceso se usan adecuadamente. Los métodos que utilizan otras clases, como la clase Sistema, se mantienen públicas. Estas mismas se permiten estar públicas porque son clases que se podrían utilizar en el futuro en caso de que se llegue a expandir el programa. También hubieron métodos que se alojaron en el modificador de acceso Protected, como es el caso de Video::calcularPromedio(). Esto es porque no es un método que otra clase debería de usar, aparte de la clase base y sus hijos. Que permanezca en Protected es necesario para que los hijos puedan usarlo, algo que no sería posible si se asignara en Private. Los atributos de todas las clases se quedan en Private para asegurarse de que no se puedan modificar fácilmente.

La sobreescritura de métodos resultó muy útil para cuando se necesita cambiar la funcionalidad de una función para una clase en específico. Esto permite no tener que memorizar una nueva función, algo que se utilizó en las clases hijas, aunque resultó más útil el polimorfismo y las clases abstractas. Para las funciones que no tendrían sentido que heredaran un funcionamiento, se utilizó el polimorfismo y las clases abstractas, como getTipoVideo() o datosToString()

Se utilizan las excepciones, especialmente en aquellas donde se requiere que el usuario ingrese información para que reciba información en pantalla.

Identificación de casos que podrían hacer que el programa deje de funcionar

No deja de funcionar, pero actualmente hay un error en el que, al mostrar los episodios de una serie, el nombre del episodio muestra valores basura.

Conclusión

Hay áreas de oportunidad definitivamente. Se podría implementar una entrada y salida de archivos de texto para guardar los videos permanentemente y no tener que generarlos en el main cada vez que se inicia el programa. También se tiene que solucionar el error respecto a los valores basura en el nombre de los episodios de una serie.

Aún así, se logró crear un sistema lo suficientemente bueno como prototipo para poder expandirlo si se decidiera así. Con el conocimiento de materias posteriores se podría implementar más con estructuras de datos, acceso desde el internet, o incluso crear una aplicación de streaming formal con una interfaz gráfica más amigable para el usuario.

Referencias

https://stackoverflow.com/questions/15745045/how-do-i-resolve-git-saying-commit-your-changes-or-stash-them-before-you-can-me

https://www.geeksforgeeks.org/pointers-and-references-in-c/