



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

Programación Orientada a Objetos

28/07/21

TC1030

Modelado de Servicio de Streaming

Aaron Inzunza Inzunza - A01114528
Julian Lawrence Gil Soares - A00832272
José Jezarel Sánchez Mijares - A01735226

Profesora encargada: Veronica Rodriguez Rodriguez

Índice

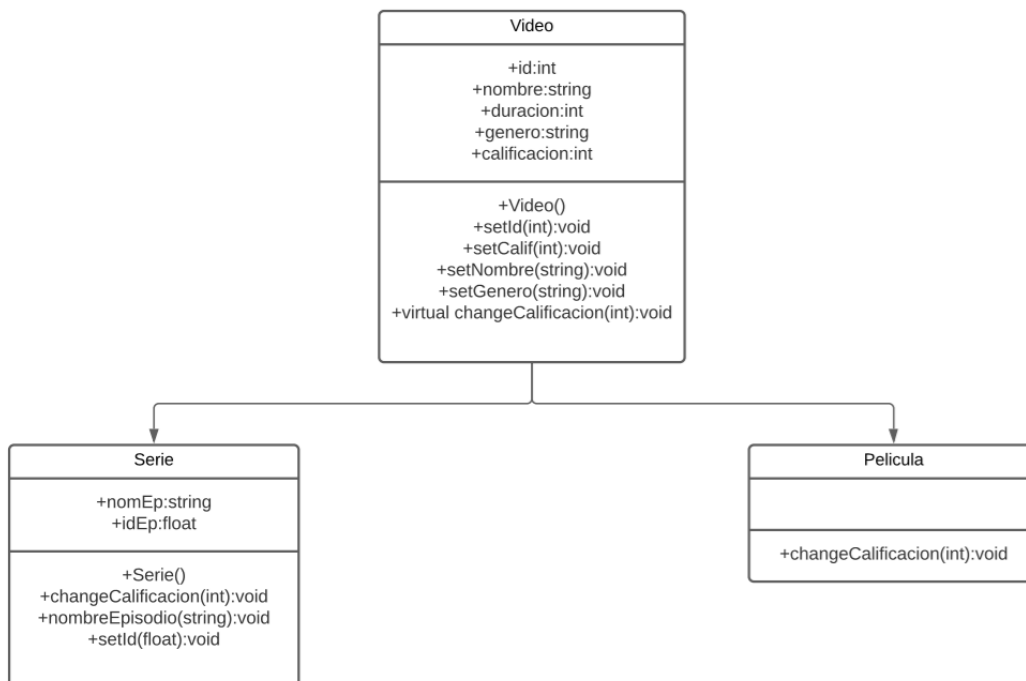
Introducción	3
Diagrama UML	3
Ejemplos de Ejecución	4
Argumentación	9
Casos de Error.....	10
Conclusion.....	10
Referencias.....	11

Introducción | Planteamiento del problema

Cada año más plataformas de streaming digitales son creadas y cada año se agregan más películas y series a estas plataformas. El objetivo de este proyecto fue utilizar el lenguaje de programación C++ y la programación orientada a objetos para crear un programa en el que se pueden guardar estos vídeos y que un usuario pueda fácilmente acceder a ellos para ver sus características como su duración o su género. Para este proyecto aplicamos la programación orientada a objetos este tipo de programación se enfoca en utilizar clases para reducir cuanto código se tiene que escribir.

Diagrama UML

Decidimos crear 3 clases la clase Video, Serie y Película las clases Serie y Película heredan atributos y métodos de la clase abstracta video esto lo hicimos para evitar tener código redundante en nuestro programa. La clase serie la utilizamos para crear los objetos que representan el episodio ya que tienen un ID de tipo float y cada uno tiene un nombre distinto a la serie. Podemos observar polimorfismo y sobreescritura con el método de changeCalificacion.



Ejemplos de Ejecución

Al correr el programa este es el menú que aparece.

```
Bienvenido usuario
---Menu---
1. Mostar Peliculas
2. Mostrar Series
3. Buscar Peliculas
4. Buscar Series
5. Cambiar Calificacion(Pelicula)
6. Cambiar Calificacion(Serie)
7. Terminar
█
```

Opción 1

Al seleccionar 1) se despliega la lista de películas.

NOTA: Es necesario desplegar las películas para que se pueda buscar o cambiar de calificación más adelante.

```
1 Red accion 120
2 Venom accion 140
3 Deadpool accion 230
4 Mechanic accion 80
5 Tennesse accion 150
6 Monday drama 200
7 Palmer drama 130
8 Capone drama 110
9 Unbroken drama 90
10 Parasite drama 100
11 Rebecca misterio 200
12 Jigsaw misterio 180
13 Inferno misterio 120
14 Arrival misterio 120
15 Sicario misterio
```

Opción 2

Al seleccionar la opción 2) se despliega la lista de series con sus respectivos episodios.

NOTA: Es necesario desplegar las series para que se pueda buscar o cambiar de calificación más adelante.

```
1 Invincible accion
1.1 Super 20
1.2 Crash 20
2.1 Stop 20
2 You drama
1.1 Joe 40
1.2 Beck 40
2.1 Love 40
3 Dark misterio
1.1 sun 40
1.2 moon 40
2.1 star 40
```

Opción 3

Al seleccionar la opción 3), te genera una lista de cómo quisieras buscar una película, (ID, Género o Calificación).

```
Como quiere buscar peliculas
1. ID
2. Genero
3. Calificaion(1-5)
```

Si eliges la opción 1) de ID, te preguntará el número ID que quisieras elegir y dependiendo de tu respuesta, te desplegará el nombre de la película.

```
Como quiere buscar peliculas
1. ID
2. Genero
3. Calificaion(1-5)1
Que ID quiere ver?
5
Tennet
```

Al elegir la opción 2) de Género, te despliega las opciones que puedes elegir: 1) Acción, 2) Drama y 3) Misterio. Dependiendo de qué número elijas, se despliegan todas las películas que se encuentren dentro del género que selecciones.

```
Como quiere buscar peliculas
1. ID
2. Genero
3. Calificaion(1-5)
2
Que genero quiere ver?
1. Accion
2. Drama
3. Misterio
```

Al elegir la opción 3) de Calificación te preguntará con qué número de estrellas del 1-5 quisieras buscar la película. Dependiendo de que elijas se desplegarán las películas.

```
Como quiere buscar peliculas
1. ID
2. Genero
3. Calificaion(1-5)
3
Que calificacion quiere buscar?4
Venom tiene claificacion 4
Sicario tiene claificacion 4
```

Opción 4

Al seleccionar la opción 4) de la lista, te preguntará cómo quisieras buscar la serie.

```
Como quiere buscar series
1. ID
2. Genero
3. Calificaion(1-5)1
Que ID quiere ver?(1-3)
1
Invincible
```

Al elegir 2) por género se despliega las opciones acción, drama y misterio al igual que la última vez, pero al elegir, esta vez se despliega el nombre de la serie y el nombre de los episodios dentro de esa serie. Y sucede lo mismo con la opción 3).

```
Como quiere buscar series
1. ID
2. Genero
3. Calificaion(1-5)2
Que genero quiere ver?
1. Accion
2. Drama
3. Misterio
3
Dark
sun moon star
```

Opción 5

Al seleccionar la opción 5) , te pregunta sobre la película que quisieras cambiar su calificación tomando su ID como número de referencia. Después te pregunta a qué calificación quieres cambiar y ya que el cambio esté hecho se despliega el menú de nuevo.

```
Bienvenido usuario
---Menu---
1. Mostar Peliculas
2. Mostrar Series
3. Buscar Peliculas
4. Buscar Series
5. Cambiar Calificacion(Pelicula)
6. Cambiar Calificacion(Serie)
7. Terminar
5
De que pelicula quiere cambiar calificacion? (ID)
4
Mechanic
A que calificaion lo quiere cambiar?
3
```

Opcion 6

La opción 6 es lo mismo que la 5 pero de series. Sigues el mismo proceso que la opción 5 pero después de pedir cual serie quiere calificar pide a qué episodio le quieres cambiar la calificación.

```
Bienvenido usuario
---Menu---
1. Mostar Peliculas
2. Mostrar Series
3. Buscar Peliculas
4. Buscar Series
5. Cambiar Calificacion(Pelicula)
6. Cambiar Calificacion(Serie)
7. Terminar
6
De que serie quiere cambiar calificacion? (ID)(1-3)
2
Eligio cambiar la calificacion de You
Que episodio quiere cambiar? (1-3)
2
Va a cambiar la calificacion de Beck
A que lo quieres cambiar?(1-5)4
```

Opción 7

Al elegir la opción 7, se termina el programa.

```
Bienvenido usuario
---Menu---
1. Mostar Peliculas
2. Mostrar Series
3. Buscar Peliculas
4. Buscar Series
5. Cambiar Calificacion(Pelicula)
6. Cambiar Calificacion(Serie)
7. Terminar
7

GRACIAS!! :)>
```


Argumentacion

A) Se identifican de manera correcta las clases a utilizar

Decidimos utilizar una clase de Videos, otra de Series y otra de Películas. La clase Video tiene atributos que puedan ser utilizados en las otras dos clases. Y las clases de Series y Películas tienen sus atributos y métodos en público.

B) Se plantea de manera correcta el concepto de Herencia

Como se mencionó en el inciso a) la clase Video hereda sus atributos a la clase Película y clase Series para de esta manera disminuir la cantidad de código y que sea más funcional.

C) Se emplea de manera correcta los modificadores de acceso

Se han implantado distintos modificadores públicos por todo el programa. Casi todos los atributos fueron heredados de la Clase Video pero también cada clase tiene sus propios métodos y atributos

D) Se emplea de manera correcta la sobreescritura de métodos

En la clase Video se utilizó un “virtual void” que lo convierte en una clase abstracta. Se sobreescriben los valores del virtual void en las otras clases.

E) Se emplea de manera correcta el concepto de Polimorfismo

El polimorfismo se encuentra dentro del programa, por ejemplo el “changeCalificación” es una función heredada, sin embargo es distinta para cada una de las clases.

F) Se emplea de manera correcta el concepto de Clases Abstractas

Utilizamos un método virtual llamado changeCalificacion decidimos que este método lo haríamos virtual ya que para las series se califican los episodios y no las series pero si se califican las películas entonces podemos observar sobrescritura y el uso de clases abstractas en este método.

G) En el program principal se utilizan funciones para cada opción del menú

Para cada opción del menú, se utilizaron los métodos para llamar las diferentes clases dentro de cada if statement.

H) Dentro del programa manejamos excepciones a la hora de calificar videos. Si el usuario introduce un número que está fuera del rango de calificaciones que maneja el programa (1-5) se despliega un mensaje de error también manejamos excepciones a la hora de buscar películas y series. Cuando el programa pide que atributo quiere utilizar para buscar videos si el usuario introduce un número que está fuera del rango (1-3) se despliega un mensaje de error similar

Casos de Error

Si el usuario no carga correctamente los archivos de series y películas las otras opciones no funcionarían de manera correcta.

Si el usuario busca una película o serie por ID y introduce un número de ID que es muy grande o pequeño el programa no despliega nada y simplemente regresa al menú.

Si el usuario character o un string en lugar de un número cuando el programa le pregunta cómo quiere buscar películas el programa entra a un ciclo infinito.

Conclusion personal

Aaron:

En la elaboración de este proyecto use los conceptos básicos que aprendimos en la clase de Programación Orientada a Objetos como herencia, polimorfismo, etc. para elaborar un program útil para la vida práctica en este caso siendo elaborando un sistema que despliega las calificaciones de películas y series de una plataforma de streaming. Al principio se me hizo difícil porque no sabía por dónde empezar y luego si me atoraba en algo me frustraba y ya no quería terminarlo pero con dedicación y esfuerzo pude terminar el proyecto en tiempo y forma.

Julian:

A lo largo de este curso aprendimos acerca de la programación orientada a objetos que es como dice su nombre una forma de programar en el que se enfoca en el uso de objetos para desarrollar el programa. En este proyecto creamos una base de datos en donde se guardan películas y series y un interfaz que el usuario puede utilizar para poder buscar videos por sus diferentes atributos como su género o su calificación. Para poder lograr esto fue muy importante el manejo de clases y objetos y que redujera la cantidad de código por mucho. El poder crear un objeto que guarde todos estos diferentes atributos hace que un programador pueda crear códigos mucho más complejos sin tener que escribir veinte mil líneas de código.

Jeza:

Durante el proceso de elaboración de este proyecto hice uso de todos los conceptos llevados a lo largo del curso de Programación Orientada a Objetos (POO), destacando el uso de la herencia y el polimorfismo lo cual facilitó la realización del proyecto, ya que, permitió ir relacionando las clases sin mayor dificultad permitiendo bastante ahorro en las líneas de código, se hizo un poco complicado la incorporación de los archivos de texto sin embargo al final se logró obtener los resultados deseados y nos permitió mejorar el programa. Aunque este proyecto tuvo sus dificultades, fue gracias a las lecciones de programación tomadas durante el curso que no solo se hizo más fácil el proyecto, si no, que se pudo mejora la calidad de nuestra programación

Referencias consultadas

Coding Games and Programming Challenges to Code Better. (2018). CodinGame.

<https://www.codingame.com/playgrounds/50747/herencia-en-c-practica-3/concepto-de-herencia>

adrianvaca. (2011). *Polimorfismo.* Programacionenc.net.

https://programacionenc.net/index.php?option=com_content&view=article&id=70:polimorfismo&catid=37:programacion-cc&Itemid=55#:~:text=El%20polimorfismo%20se%20implementa%20por,est%C3%A1%20asociada%20con%20ese%20objeto.

ztizzlegaming. (2019). *ztizzlegaming/movie-recommendation-system.* GitHub.

<https://github.com/ztizzlegaming/movie-recommendation-system>

Coding Games and Programming Challenges to Code Better. (2018). CodinGame.

<https://www.codingame.com/playgrounds/50557/clases-y-objetos-en-c-practica-1/mi-embros-de-clase-en-c-variables-y-metodos>

Moisset, D. (2021). *Herencia en C++.* Tutorialesprogramacionya.com.

<https://www.tutorialesprogramacionya.com/cmasmasya/detalleconcepto.php?punto=27&codigo=158&inicio=15>

Conceptos básicos de la Programación Orientada a Objetos. (2021). Sc.ehu.es.

<http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/clases1/clases.htm>

designing a simple movie rating guide pr - C++ Forum. (2013). Cplusplus.com.

<http://www.cplusplus.com/forum/beginner/96274/>

C++ Object-Oriented Curriculum-Film Review Information Management Program on

Movie Watching Website - Programmer Sought. (2017). Programmersought.com.

<https://www.programmersought.com/article/8489431008>