



Tecnológico de Monterrey

Campus Guadalajara

Entrega final: Situación problema Netflix

Curso:

Programación orientada a objetos gpo (315)

Realizado por:

Victoria Ramírez Castro | A01640824

Liliana Solórzano Pérez | A01641392

Docente:

Fabiola Uribe

Fecha de entrega:

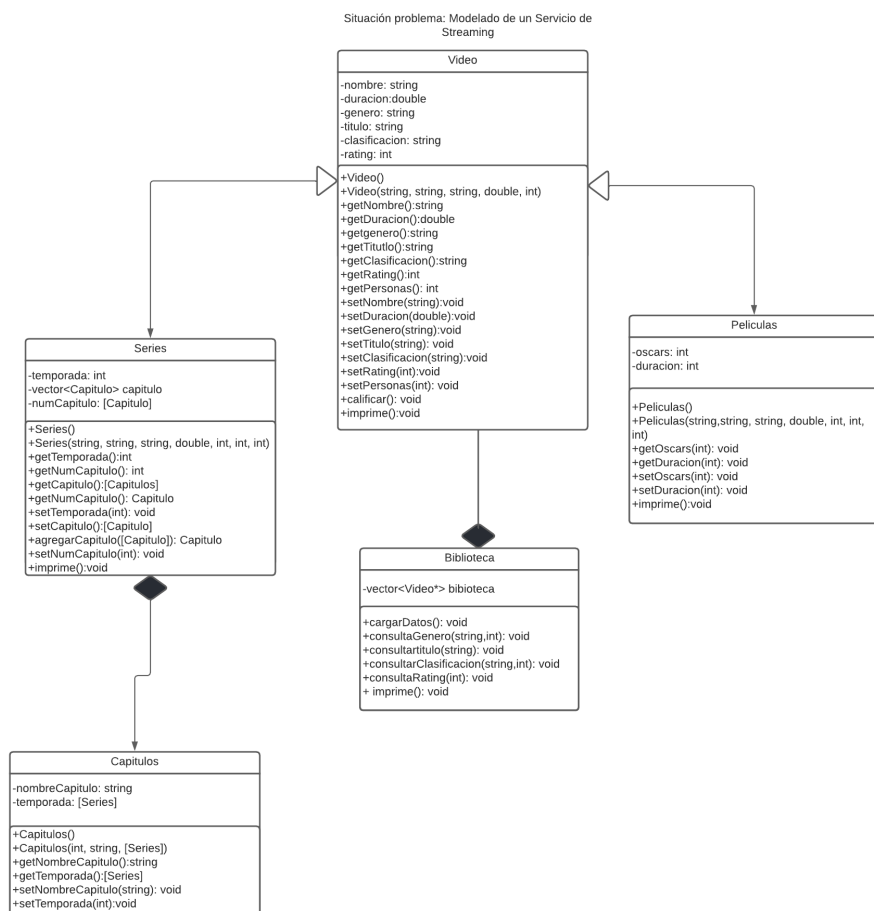
17 de Junio 2022

Introducción (planteamiento del problema)

A lo largo de este proyecto se implementará la programación orientada objetos en el lenguaje de programación c++. Se realiza un modelado de servicio de streaming, donde se implementan diversas clases, en este caso, la clase base Video, sus clases derivadas, películas y series, al igual que la clase capítulo (una serie está compuesta de capítulos) y biblioteca (de donde se hace la lectura de archivos).

Este proyecto tiene como objetivo crear una plataforma tipo Netflix, donde el usuario pueda consultar sus series o películas favoritas de acuerdo a una base de datos, previamente establecida, así como su respectiva información de cada una de ellas.

Diagrama de clases UML



Ejemplo de ejecución (capturas de pantalla)

```
-----Bienvenidos a NETFLIX-----  
Este programa esta hecho por Victoria y Liliana  
Antes de iniciar, vamos a crear su cuenta  
Ingrese su nombre de usuario  
>> Victoria  
  
Bienvenid@ Victoria , gracias por preferir nuestra aplicacion  
  
Seleccione la opcion que prefiera  
1) Mostrar catalogo  
2) Seleccionar pelicula  
3) Seleccionar serie  
4) Calificar video  
5) Salir  
>> █
```

```
DATOS DE LA PELÍCULA  
+++++  
Título: Sombras tenebrosas  
Genero: Comedia  
Clasificación: PG-13  
Rating: 5.000  
Duracion de la pelicula: 113 minutos  
Cantidad de Oscars de la pelicula: 0
```

```
DATOS DE LA PELÍCULA  
+++++  
Título: El proyecto Adam  
Genero: Ciencia ficcion  
Clasificación: PG-13  
Rating: 4.000  
Duracion de la pelicula: 106 minutos  
Cantidad de Oscars de la pelicula: 1
```

```
DATOS DE LA PELÍCULA  
+++++  
Título: No mires arriba  
Genero: Temas sociales  
Clasificación: R  
Rating: 3.200  
Duracion de la pelicula: 138 minutos  
Cantidad de Oscars de la pelicula: 3
```

```
DATOS DE LA SERIE
+++++
Título: Atypical
Genero: Drama
Clasificación: TV-14
Rating: 4.700
Cantidad de Temporadas: 4
Total de Capítulos: 38
- Temporada: 1
+ Capítulo: Antartida
+ Duracion: 38 minutos
- Temporada: 1
+ Capítulo: Una hembra humana
+ Duracion: 33 minutos
- Temporada: 1
+ Capítulo: Julia dijo
+ Duracion: 31 minutos

DATOS DE LA SERIE
+++++
Título: Stranger things
Genero: Ciencia ficcion
Clasificación: TV-14
Rating: 3.600
Cantidad de Temporadas: 4
Total de Capítulos: 32
- Temporada: 1
+ Capítulo: Capitulo uno: La aparicion de Will Byers
+ Duracion: 49 minutos
- Temporada: 1
```

```
Buscar peliculas por:
1) Titulo
2) Genero
3) Clasificacion
4) Rating
>> 1
Pelicula que desea buscar:
>> Crush

DATOS DE LA PELÍCULA
+++++
Título: Crush
Genero: Romance
Clasificación: PG-13
Rating: 1.700
Duracion de la pelicula: 102 minutos
Cantidad de Oscars de la pelicula: 2

¿Quiere regresar al menu principal? (S/N)
>> █
```

```
Buscar serie por:
1) Titulo
2) Genero
3) Clasificacion
4) Rating
>> █
```

```
Buscar serie por:
1) Título
2) Género
3) Clasificación
4) Rating
>> 2
Género que desea buscar:
>> Comedia

DATOS DE LA SERIE
+++++
Título: One day at a time
Género: Comedia
Clasificación: TV-PG
Rating: 4.540
Cantidad de Temporadas: 4
Total de Capítulos: 43
- Temporada: 1
+ Capítulo: No hay mas
+ Duración: 31 minutos
- Temporada: 1
+ Capítulo: Bobos y mamitas
+ Duración: 30 minutos
- Temporada: 1
+ Capítulo: Misa
+ Duración: 32 minutos
¿Quiere regresar al menú principal? (S/N)
>> █
```

```
Escriba el título del video a calificar
>> Sombras tenebrosas
Ingrese la calificación
>> 4█
```

```
La calificación tiene que estar entre 1 y 5
Ingrese la calificación
>> █
```

Argumentación sobre el diseño de tus clases, el porqué elegiste esa estructura de clases, herencia, decisión sobre funciones polimórficas, sobre el operador sobrecargado, el manejo de excepciones que se implementó.

- **Clase base -> Video:** La clase video se establece como la clase base, ya que, tanto película como serie, son videos, ambos poseen las mismas características de un video, en este caso, título, género, clasificación, rating y personas.
- **Clases derivada de Video con herencia -> Serie y película:** La clase serie y película posee la herencia de los atributos de la clase video, ya que, se comporta como uno.

Sin embargo, en el caso de una serie, posee sus atributos propios como son los capítulos y las temporadas. Por otra parte, se tiene la clase película, la cual cuenta con dos atributos propios de esta, oscars y duración.

- **Clase de compuesta para Serie -> Capítulo:** Una serie está compuesta por varios capítulos, y no una serie se deriva en capítulos. En este caso, se crean varios objetos de la clase capítulo, y esos objetos componen la clase serie.
- **Archivo txt -> Biblioteca:** Este archivo contiene la base de datos de películas y series, capítulos y temporadas que cuenta nuestro navegador.
- **Lector de archivo txt -> Clase biblioteca:** Esta clase se encarga de leer el archivo txt, haciendo uso de funciones polimórficas, apuntadores y vectores. En esta clase, también se implementan filtros para mostrar ciertos videos.
- **Implementación del menú -> main:** Se genera el menú de nuestro buscador de películas y series, creando las opciones que puede seleccionar el usuario. Si es que desea consultar el catálogo, buscar series o películas por género, título, rating, entre otros.
- **Polimorfismo:** Nosotras utilizamos polimorfismo entre la clase video, serie y películas (ya que solo se puede aplicar polimorfismo cuando hay herencia), en la función de “imprime”. Solo la hicimos en esta función, debido a que queremos que tanto la clase video pueda imprimir todos los videos que tiene y su información, pero si es una serie, se impriman los atributos especiales de serie, y si es película se impriman los atributos especiales de película.
- **Sobrecarga:** Para este proyecto, consideramos que un buen lugar para hacer la sobrecarga seria a la hora de calificar un video, ya que queremos que se sume la calificación actual + nueva calificación. La sobrecarga se hizo en el operador “+” para que cuando se quiera sumar la calificación que ya tiene el video + la nueva calificación, lo que en realidad haga es multiplicar la calificación del video por el

número de personas que han votado, después sume la nueva calificación y ese resultado dividirlo entre el número de personas que calificaron más uno (porque ya aumentó el número de personas).

- **Manejo de excepciones:** El manejo de excepciones (aunque pudimos haber hecho más excepciones) solo la utilizamos a la hora de pedir la calificación del video, ya que la calificación solo puede ser un número entre 1 y 5. Si la calificación ingresada por el usuario sale de este rango, le va a arrojar un error de que está fuera del rango y vuelve a pedir la calificación hasta que el usuario escriba un valor entre 1 y 5. Igualmente, un manejo de excepciones parecido se utilizó para encontrar videos con un rating entre 1 y 5, y si se ingresa un valor fuera de ese rango, manda un error y vuelve a pedir el rating.

Identificación de casos que harían que el proyecto deje de funcionar

1. Escribir erróneamente el título de alguna serie o película que se desea buscar en la biblioteca.
2. Seleccionar algún número que no se encuentre en el menú
3. Escribir todo con mayúsculas o minúsculas, se debe de escribir el género, título de película, entre otro, tal cual como sale en el catálogo de películas y series
4. No seguir las instrucciones del funcionamiento del programa, es decir, una vez que se accede a alguna sección del menú, ignorar las indicaciones a seguir para poder navegar dentro de este.

Conclusión personal

Liliana Solórzano: A lo largo de este proyecto me di cuenta de la importancia de la creación de clases al momento de realizar un proyecto en c++. Desde mi perspectiva, programar con clases es de cierta forma más fácil y efectivo, ya que, al momento de tratar de corregir errores

o incluso checar la funcionalidad del proyecto, es más fácil corregirlo. Por otra parte, se me hizo un poco complicado el trabajo de polimorfismo y siento que me falta un poco de más práctica al momento de tratar con apuntadores, sin embargo, logré dominar los temas de sobrecarga, herencia y excepciones, en su mayoría. Por otra parte, me di cuenta del gran trabajo de programación que hay detrás de las plataformas de streaming de videos, como Netflix, HBO, Disney Plus entre otros. Fue un proyecto que retó mis conocimientos en programación orientada a objetos en c++. A su vez que los conocimientos impartidos durante las clases fueron más que suficientes para poder resolver la situación problema.

Victoria Ramírez: Este proyecto fue realmente retador para mi, porque me cuesta algo de trabajo abstraer conceptos para poder ponerlos en código. También, me costó trabajo plasmar ciertas ideas, ya que cuando estaba escribiendo parte del código, me funcionaba una parte y otra dejaba de funcionar, o me salían errores muy básicos pero que de tantos errores que salían me abrumaba y me bloqueaba. Pero una vez que lograba corregir esos errores, el resto del código salía mas rapido. Realmente me gusto mucho este proyecto, y a comparación con el curso pasado puedo decir que entendí mejor tanto programar en c++ como entender mejor programación orientada a objetos. Aparte, me gusto mucho trabajar en equipo, porque nos complementamos entre los temas que no entendíamos, y si no entendíamos algún tema las 2, buscamos una forma de solucionarlo pero siempre entre las dos nos apoyamos. En general, creo que este proyecto me ayudó a unificar todos los conocimientos que adquirí en clases, aparte que me dio un mejor panorama de cómo funcionan plataformas de streaming como Netflix, lo cual se me hace muy interesante.

Referencias consultadas

- Diapositivas brindadas por el docente

- Deitel, H. M., & Deitel, P. J. (2004). Cómo programar en C/C++ y Java. Pearson educación
- Deitel, H. M., & Deitel, P. J. (2003). C++ how to program. Pearson Educación.
- Programazsi3n ATS. (2016). 82. Programaci3n en C++ || Punteros || Declaraci3n de Punteros. Recuperado el 10 de Junio 2020 de: https://youtu.be/_pcfFMFs9-g