



Tecnológico de Monterrey

Instituto Tecnológico de Estudios Superiores de Monterrey
Campus Estado de México

Proyecto Integrador

Unidad de formación: TC1030 Programación orientada a objetos

Grupo: 305

Francisco Tonatihu Castro Flores

Nombre del profesor: Roberto Martínez Román

Indice de contenido:**1... Portada****2... Índice****3-4... Introducción****5... Diagrama de clases****6-...Ejecución de las opciones****7... Conclusiones****8...Referencias**

Introducción

En los últimos años, han proliferado los servicios de streaming de video bajo demanda por ejemplo Netflix, Disney Plus, HBO Max, entre otros. Algunos de ellos se especializan por el volumen de videos que proporcionan a sus usuarios mientras que otros se han puesto el reto de mostrar solamente videos de su propia marca. Una versión limitada para apoyar a un futuro proveedor de este tipo de servicios es la que se describe a continuación:

Se quiere trabajar con dos tipos de videos: películas y episodios de series.

Todo video tiene:

- ID.
- Nombre de la película o nombre del episodio de una serie.
- Género (Pueden ser varios separados por el caracter ****&****).
- Calificación en escala de ****1 a 7****.
- Duración en minutos.
- Fecha de estreno en formato ****mm/dd/aaaa****.

Las series tienen episodios y cada episodio tiene:

- ID.
- Nombre.
- Temporada a la que pertenece.
- Número de episodio.

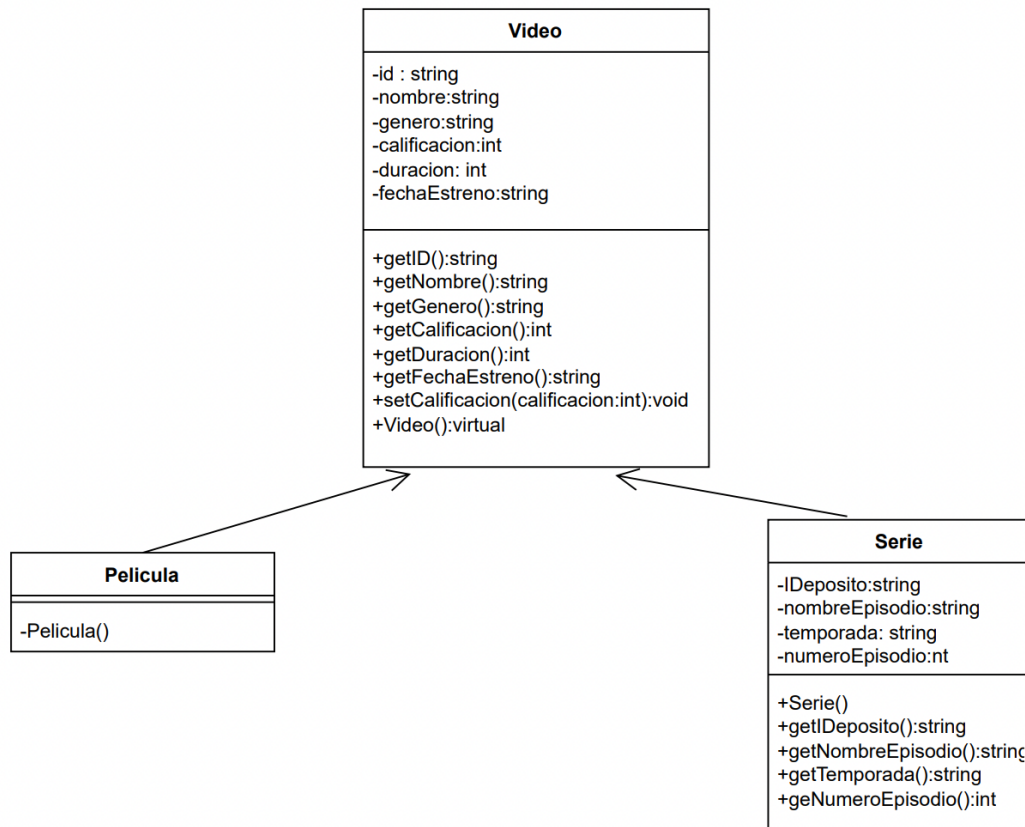
Nos interesa construir un sistema de consulta para esta información que se encuentra en un archivo de texto.

El sistema debe ser capaz de mostrar el siguiente menu de manera iterativa, termina hasta que el usuario selecciona Salir:

1. Cargar el archivo de datos en formato ****csv****.
2. Mostrar los videos en general con:

- Una calificación mayor o igual a un valor tecleado por el usuario.
 - Un género tecleado por el usuario.
3. Mostrar todos los episodios de una determinada serie que teclea el usuario.
 4. Mostrar las películas con una calificación mayor a un número tecleado por el usuario.
 5. Calificar un video.
 - Pedir título de película o nombre del capítulo de la serie a calificar.
 - Pedir valor otorgado y asignarlo (reemplaza la calificación anterior).
 6. Calcular el promedio de la calificación de una serie (promedio de todos los episodios).
 7. Salir.

Diagrama de clases



El diagrama de clases se compone de un primer rectángulo que contiene a la clase padre “Video”, en él se incluyen tanto sus atributos como sus métodos. Heredan de la clase padre “Película” y “Video” la relación de herencia se ve remarcada por las flechas. La clase “Película” hereda sus atributos de “Video”, “Serie” también hereda de “Video” y además contiene sus atributos y métodos específicos.

Ejecución de las opciones

No se logró ejecutar el código de manera correcta. Se agradecería retroalimentación.

Conclusiones

En general realizar este código fue todo un reto para mí, hubo muchas partes difíciles de implementar, pero sin duda lo que más problema me cause fue el main, ya que el lograr que pudiera leer de manera correcta el archivo .csv fue algo que hasta el momento no pude lograr. El diagrama de clases siempre es de gran importancia, porque este te ayuda a tener una idea e ir modelando lo que será el programa y te permite realizar correcciones o mejoras a tu código. La codificación fue complicada hasta cierto punto, como ya mencioné, lo más complicado fue realizar el programa main y declarar las clases fue en realidad lo más sencillo. Me queda mucho como aprendizaje tras este proyecto, desde conceptos hasta experiencias que considero poder aplicar en un futuro, un código que me causo muchos problemas, sin dudad fue una gran oportunidad para aprender y obtener conocimiento de los errores.

Referencias

Cabrera, Luis. (2021). Leer archivo csv en c++. Recuperado de:

<https://www.youtube.com/watch?v=bal7cn-YvfU>

Microsoft(2023). Const (C++). Recuperado de: [https://learn.microsoft.com/es-](https://learn.microsoft.com/es-es/cpp/cpp/const-cpp?view=msvc-170)

[es/cpp/cpp/const-cpp?view=msvc-170](https://learn.microsoft.com/es-es/cpp/cpp/const-cpp?view=msvc-170)

Microsoft (2023). Operadores integrados de c++, precedencia y asociatividad. Recuperado

de: [https://learn.microsoft.com/es-es/cpp/cpp/cpp-built-in-operators-precedence-and-](https://learn.microsoft.com/es-es/cpp/cpp/cpp-built-in-operators-precedence-and-associativity?view=msvc-170)

[associativity?view=msvc-170](https://learn.microsoft.com/es-es/cpp/cpp/cpp-built-in-operators-precedence-and-associativity?view=msvc-170)

Uchicago. (s.f). std::stoi, std::stof, std::stoll. Recuperado de:

http://naipc.uchicago.edu/2014/ref/cppreference/en/cpp/string/basic_string/stol.html

Zator systems. (2016). Tipos básicos: representación interna, rango. Recuperado de:

[https://www.zator.com/Cpp/E2_2_4.htm#:~:text=%C2%A77%20La%20clase%20numeric_limits,incluyendo%20los%20booleanos\)%20y%20fraccionarios.](https://www.zator.com/Cpp/E2_2_4.htm#:~:text=%C2%A77%20La%20clase%20numeric_limits,incluyendo%20los%20booleanos)%20y%20fraccionarios.)