



Instituto Tecnológico y de Estudios Superiores de Monterrey

CAMPUS PUEBLA



PENSAMIENTO ORIENTADO A OBJETOS

TC1030.1

Prof. Daniel Perez

Francisco Rocha Juaréz // A01730560
Jesús Jiménez Aguilar // A01735227

07 DE JUNIO DE 2021

Contenidos

Introducción	03
Solución Propuesta	04
Descripción general	04
Elementos	04
Diagrama de clases UML	05
Ejemplo de ejecución	06
Justificación de solución	08
Dificultades presentadas	09
Casos en los que el programa no funciona	10
Conclusión general	10
Conclusión de Francisco	11
Conclusión de Jesús	11
Elaborado por	12
Referencias	12



Introducción

MODELADO DE SERVICIO DE STREAMING

En los últimos años, se han proliferado los servicios de streaming de video bajo demanda por ejemplo Netflix, Disney+, Prime Video, DC 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, así como algunos una mezcla de ambos. Una versión limitada para apoyar a un futuro proveedor de este tipo de servicios es dar a conocer la calificación promedio que ha recibido cada uno de los videos en el catalogo para atraer la atención del



publico. Esta calificación está en escala de 1 a 5 donde 5 es la mejor calificación.

Se quiere trabajar con dos tipos de videos: películas y series. Las series se encuentran divididas en episodios.

Es por ello que se diseñó un modelo de datos y de aplicaciones de software sencillas que sirvan como base para que los usuarios puedan consultar la calificación de los videos de su interés así como poder otorgar también una calificación.



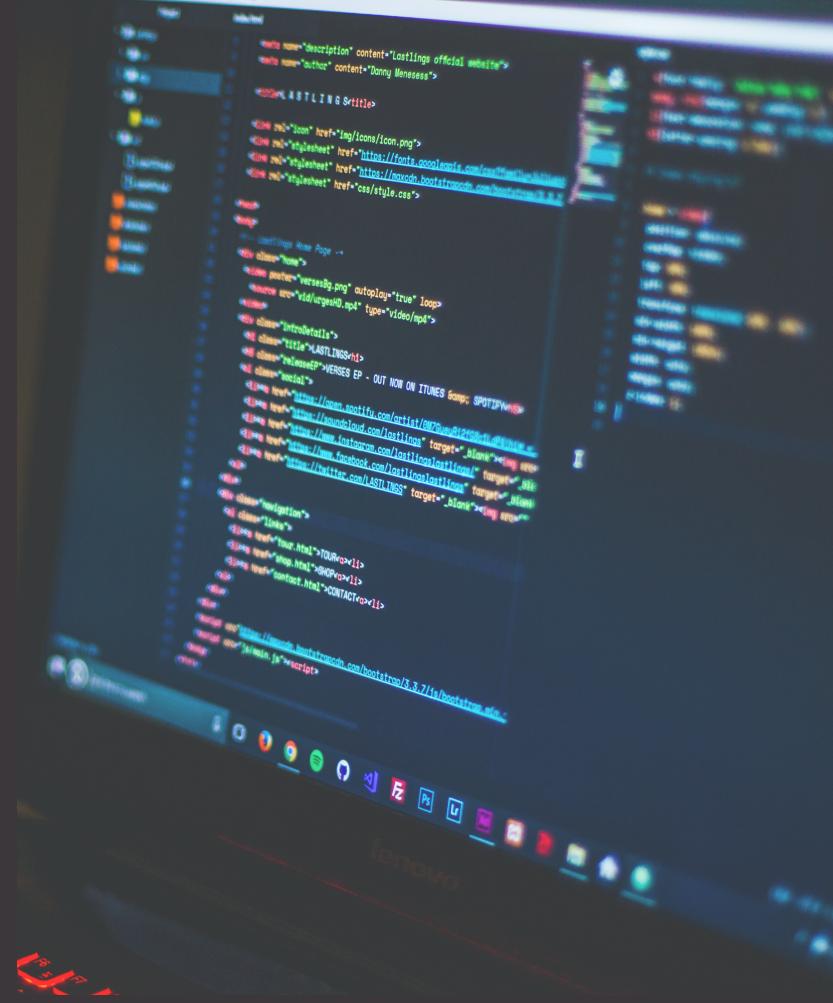
Solución propuesta:

DESCRIPCIÓN GENERAL

El programa está compuesto por un main principal, cuatro archivos .cpp y cuatro archivos .h (header).

El código básicamente lo que hace es tener registro de diversas películas y series en la que el usuario puede conocer la calificación promedio recibida por otros usuarios, así como poder darles una calificación alterando así el promedio inicial (en el caso de las series la calificación es por episodio).

Además de que el usuario puede agregar más series/películas a través de la carga de archivos.



ELEMENTOS

- Clases: se ocuparon cuatro clases (info, video, película, episodio) cada una localizada en su respectivo archivo .h. Más aparte la clase principal en el archivo Main.
- Objetos: Los objetos se usaron en todos los archivos del código.
- Atributos: Se definieron en los archivos .h y se utilizaron en los archivos .cpp
- Métodos: Se definieron en los archivos .h y se utilizaron en los archivos .cpp
- Constructores: Se definieron en los archivos .h y se utilizaron en los archivos .cpp
- Arreglos: Se localizan en el menú principal alterando todo el programa.
- Composición: Se utilizó en los archivos .cpp.

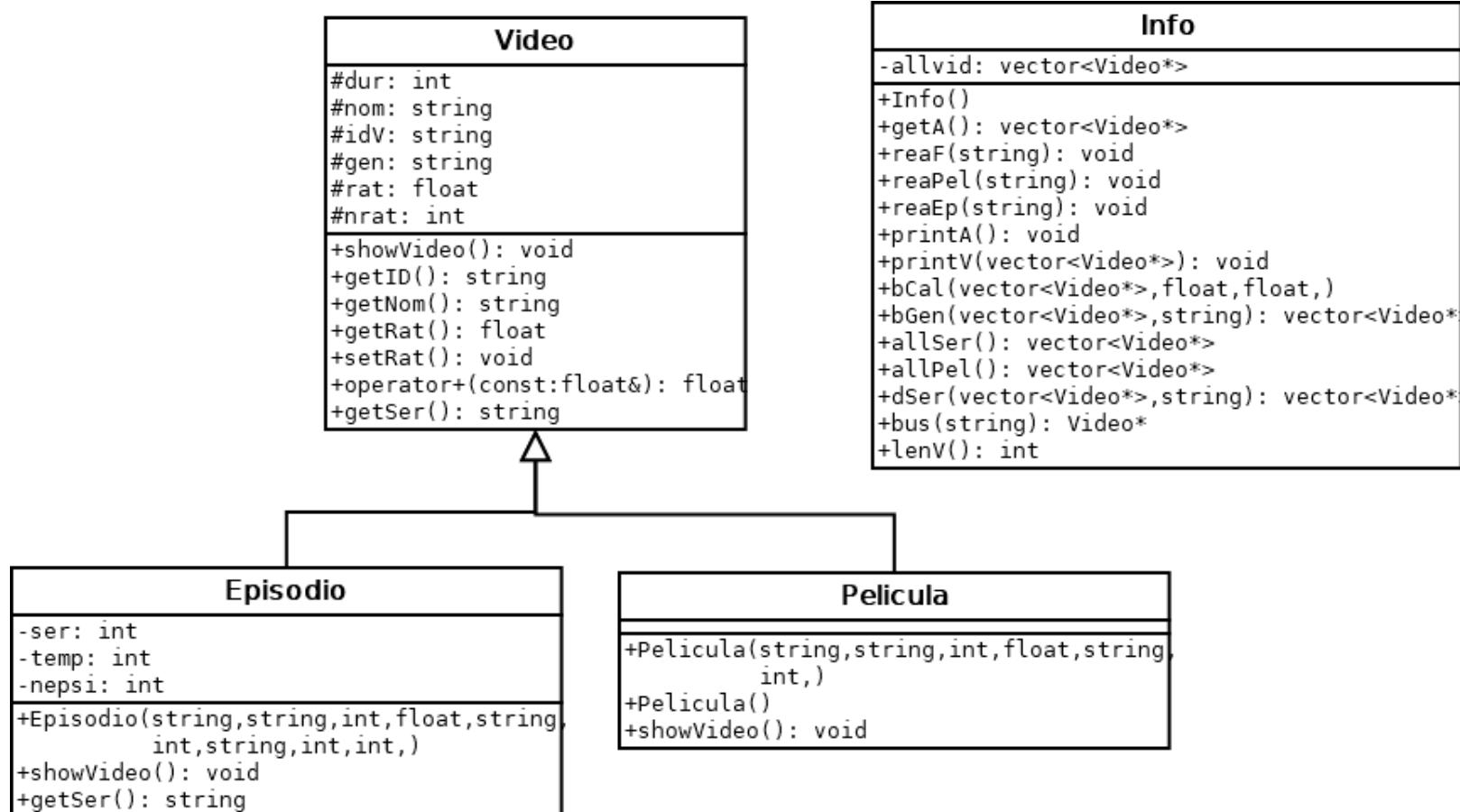


Diagrama de Clases UML

DISEÑADO EN DIA



Ejemplo de ejecución

```
PS C:\Users\jesus\Desktop\ProyectoS> g++ mainNetflix.cpp
PS C:\Users\jesus\Desktop\ProyectoS> ./a.exe

*****BIENVENIDO A TU PLATAFORMA*****
*****FAVORITA DE STREAMING*****

*****MENU PRINCIPAL*****
1. Cargar archivo de datos
2. Mostrar los videos en general, con una cierta calificacion o de un cierto genero
3. Mostrar los episodios de una determinada serie con una calificacion determinada
4. Mostrar las pelis;culas con cierta calificacion
5. Calificar un video
0. Salir
*****1

Ingrera el nombre del archivo. (Por favor incluye la terminacion .txt)
archivo2.txt
Los datos se han cargado correctamente.
```

Al ejecutar el código aparecerá el menú principal.

Seleccionando la opción 1 el usuario puede cargar un archivo de texto (.txt).

Por default se comprende que seleccionando la opción 0 se desplegará un mensaje de despedida.

```
*****MENU PRINCIPAL*****
1. Cargar archivo de datos
2. Mostrar los videos en general, con una cierta calificacion o de un cierto genero
3. Mostrar los episodios de una determinada serie con una calificacion determinada
4. Mostrar las pelis;culas con cierta calificacion
5. Calificar un video
0. Salir
*****2

Seleccione una opcion:
1. Mostrar todos los videos
2. Filtrar por genero
3. Filtrar por calificacion
1
Todos los videos:
1. ''The Lion King''
ID: M0190753 Duracion: 118 min. Rating: 3.45 Genero: drama

2. ''Onward''
ID: M0190754 Duracion: 102 min. Rating: 3.75 Genero: accion

3. ''Happy Death Day''
ID: M0190755 Duracion: 96 min. Rating: 3.4 Genero: misterio

4. ''Beauty and the Beast''
ID: M0190756 Duracion: 129 min. Rating: 3.55 Genero: accion

5. ''Miss Americana''
ID: M0198913 Duracion: 86 min. Rating: 4.13 Genero: drama

6. Outlander. Temporada 1, Episodio 1: ''Sassenach''
ID: S07704010101 Duracion: 51 min. Rating: 4.1 Genero: drama

7. Outlander. Temporada 1, Episodio 2: ''Castle Leoch''
ID: S07704010102 Duracion: 44 min. Rating: 4.05 Genero: drama
```

Seleccionando la opción 2 el usuario podrá ver los videos cargados.

Para ello aparecerá otro menú como filtro.

Seleccionando la opción 1 del submenú, podrá ver todos los videos sin importar la calificación, el género o el tipo.

Seleccionando la opción 2 del submenú, podrá ver todos los videos de cierto género sin importar la calificación o el tipo.

```
*****MENU PRINCIPAL*****
1. Cargar archivo de datos
2. Mostrar los videos en general, con una cierta calificacion o de un cierto genero
3. Mostrar los episodios de una determinada serie con una calificacion determinada
4. Mostrar las pelis;culas con cierta calificacion
5. Calificar un video
0. Salir
*****2

Seleccione una opcion:
1. Mostrar todos los videos
2. Filtrar por genero
3. Filtrar por calificacion
2
Escriba el genero:
-Drama
-Accion
-Misterio
Accion

Resultados:
1. ''Onward''
ID: M0190754 Duracion: 102 min. Rating: 3.75 Genero: accion

2. ''Beauty and the Beast''
ID: M0190756 Duracion: 129 min. Rating: 3.55 Genero: accion

3. Stranger Things. Temporada 2, Episodio 1: ''Chapter One: MADMAX''
ID: S07404010201 Duracion: 48 min. Rating: 4.15 Genero: accion

4. Stranger Things. Temporada 2, Episodio 2: ''Chapter Two: Trick or Treat Freak''
ID: S07404010202 Duracion: 56 min. Rating: 4.2 Genero: accion
```

Seleccionando la opción 3 del submenú, podrá ver todos los videos que se encuentren en cierto rango de calificación sin importar el género o el tipo.

```
*****MENU PRINCIPAL*****
1. Cargar archivo de datos
2. Mostrar los videos en general, con una cierta calificación o de un cierto género
3. Mostrar los episodios de una determinada serie con una calificación determinada
4. Mostrar las películas con cierta calificación
5. Calificar un video
0. Salir
*****
2

Seleccione una opción:
1. Mostrar todos los videos
2. Filtrar por género
3. Filtrar por calificación
3
Escriba la calificación mínima (0-5): 4
Escriba la calificación máxima (0-5): 4.7

Resultados:
1. "Miss Americana"
ID: M0198913 Duración: 86 min. Rating: 4.13 Genero: drama

2. Outlander. Temporada 1, Episodio 1: "Sassenach"
ID: S07704010101 Duración: 51 min. Rating: 4.1 Genero: drama

3. Outlander. Temporada 1, Episodio 2: "Castle Leoch"
ID: S07704010102 Duración: 44 min. Rating: 4.05 Genero: drama
```

```
*****MENU PRINCIPAL*****
1. Cargar archivo de datos
2. Mostrar los videos en general, con una cierta calificación o de un cierto género
3. Mostrar los episodios de una determinada serie con una calificación determinada
4. Mostrar las películas con cierta calificación
5. Calificar un video
0. Salir
*****
3

Escriba el nombre de la serie que desea filtrar: Once Upon A Time
Escriba la calificación mínima (0-5): 4.5
Escriba la calificación máxima (0-5): 5

Resultados:
1. Once Upon A Time. Temporada 2, Episodio 2: "We Are Both"
ID: S07704010202 Duración: 54 min. Rating: 4.5 Genero: misterio

2. Once Upon A Time. Temporada 2, Episodio 3: "Lady of the Lake"
ID: S07704010203 Duración: 56 min. Rating: 4.6 Genero: misterio
```

Seleccionando la opción 4 el usuario podrá ver todas las películas que se encuentren en cierto rango de calificación sin importar el género.

Seleccionando la opción 3 el usuario podrá ver todos los episodios (de una determinada serie) que se encuentren en cierto rango de calificación.

```
*****MENU PRINCIPAL*****
1. Cargar archivo de datos
2. Mostrar los videos en general, con una cierta calificación o de un cierto género
3. Mostrar los episodios de una determinada serie con una calificación determinada
4. Mostrar las películas con cierta calificación
5. Calificar un video
0. Salir
*****
4

Escriba la calificación mínima (0-5): 3
Escriba la calificación máxima (0-5): 4

Resultados:
1. "The Lion King"
ID: M0190753 Duración: 118 min. Rating: 3.45 Genero: drama

2. "Onward"
ID: M0190754 Duración: 102 min. Rating: 3.75 Genero: accion
```

```
*****MENU PRINCIPAL*****
1. Cargar archivo de datos
2. Mostrar los videos en general, con una cierta calificación o de un cierto género
3. Mostrar los episodios de una determinada serie con una calificación determinada
4. Mostrar las películas con cierta calificación
5. Calificar un video
0. Salir
*****
5

Escriba el nombre o ID del título a calificar: Miss Americana
Introduzca calificación: 5
Título calificado con éxito.

Nueva calificación del título: 4.13967
```

Seleccionando la opción 5 el usuario podrá asignar una calificación a un episodio o película, misma que se promediará con las calificaciones otorgadas anteriormente.

Justificación

¿POR QUÉ ESTA SOLUCIÓN Y NO OTRAS?

- **Clases a utilizar:** En este caso decidimos dividir las clases en Vídeo, Episodio, Película e Info que no es un objeto como tal, si no una clase que contiene métodos para manipular los vectores con la información de cada una de las películas y episodios.
- **Herencia:** La herencia está presente en nuestro programa con la clase padre abstracta vídeo y sus clases derivadas episodio y película. Decidimos que esta opción era la más fácil, ya que Episodio y Película, ambas son videos y comparten atributos en común y métodos para su visualización.
- **Modificadores de acceso:** En nuestro programa ocupamos modificadores de acceso protegidos y privados para los atributos y métodos públicos. No se optó por atributos públicos porque esto no es una buena práctica en la programación.
- **Sobrecarga y sobrescritura de métodos:** En nuestro programa sobreescrivimos los métodos virtuales de video en nuestras clases derivadas y se definen como queremos que actué en cierto caso.
- **Polimorfismo:** En nuestro programa se aprecia el concepto de polimorfismo en nuestros métodos. Por ejemplo, la manera de visualizar un episodio no es la misma en la que se visualiza una película, el método es el mismo, pero este se comporta de maneras diferentes, ya que ambos son videos, pero cada uno se comporta como un episodio o una película. El polimorfismo nos facilita mucho en la programación orientada a objetos.
- **Clases Abstractas:** La clase abstracta en nuestro programa es la clase video, ya que es una clase que no queremos instanciar, no queremos tener un objeto video, queremos videos que estén definidos como episodio o película. Es por ello que definimos esta clase como abstracta.
- **Sobrecarga:** Tuvimos problemas al inicio en como sumar las calificaciones y sacar el promedio de una manera fácil, es por ello que decidimos sobre cargar el operador + para poder hacer una operación simple y poder obtener un promedio de manera más fácil.
- **Excepciones:** Un ejemplo de excepciones en nuestro programa es que cuando el vector esta vacío hay ciertas instrucciones para que el programa no se rompa.

Dificultades presentadas:

C++ ¿ALIADO O ENEMIGO?

- Había sintaxis que no estaban muy claras por lo que se tuvieron que repasar los apuntes de clase.
- El tiempo en el que coincidíamos era poco ya que tenías otros proyectos que realizar.
- Cuando se quiso promediar la nueva calificación se presentó error de sintaxis.
- Trabajar con vectores y sentencias if anidadas en ocasiones causaba confusión y no se cerraban los corchetes correctamente.



- Se trató de realizar el menú con la sentencia switch pero se presentaron dificultades por lo que se optó trabajar con la sentencia if en lo que se resolvía la situación.
- En ocasiones se definían mal las variables o se llamaban incorrectamente.
- El nombre de una clase coincidía con el de un atributo lo cual generaba error.

Casos en los que no funciona el programa

Un caso que provoca error en nuestro programa es cuando calificas un título y pones un número extremadamente grande, de igual manera te permite poner números mayores a 5 cuando no debería ser así, igual al buscar. También cuando buscas un título y no respetas las mayúsculas no lo encuentra, obligandote a escribir todo con sus respectivas mayúsculas y minúsculas.

Conclusión general

Es muy importante que las plataformas de streaming cuenten con un programa donde los usuarios puedan ver los videos de su interés clasificados en una puntuación de 1 a 5 así como ellos puedan dar su propia puntuación, ya que así el resto del público podrá tener una idea de que tan recomendado es ese video.

A pesar de que la calificación de cada video parezca irrelevante, tiene mucha importancia debido a que si la mayor parte del catálogo tiene una puntuación alta los usuarios se sentirán atraídos a suscribirse a dicha plataforma pues considerará que tiene contenido de calidad lo cual generará grandes ganancias para la compañía todo dependerá de su adaptabilidad al cambio, ya que son empresas con un gran potencial que deben aprovechar cada oportunidad para crecer día con día.



Conclusión de Francisco

El curso de programación orientada a objetos 2 fue de gran aprendizaje para mí, debido a que pude aprender sobre herencia y polimorfismo que es lo que realmente define a este tipo de programación, no fue una materia fácil, requirió de mucho esfuerzo y de ver las clases varias veces hasta tener los conceptos muy en claro. Sin embargo es de mis materias favoritas y considero que todos los aprendizajes obtenidos durante este curso van a ser de gran ayuda en mi vida profesional.



Conclusión de Jesús

La manera en la que se impartieron las clases durante este zoomsemestre fue muy buena, ya que sirvió para entender la programación orientada a objetos y no solo se enfocó en enseñar, sino en organizarnos en clases y saber reutilizar los códigos de manera correcta y eficiente. Además, en cada clase se mantenía activa la atención de la mayoría de los alumnos para no perder el hilo de la clase. Aunque creemos que de manera presencial la experiencia hubiera sido más dinámica ya que en ocasiones se observó que no todos estaban presentes en las clases.



Elaborado por:

- Francisco:
 - Archivo main.cpp
 - Archivos Video (.cpp y .h)
 - Archivos Episodio (.cpp y .h)
 - Diagrama UML
- Jesús:
 - Archivo main.cpp
 - Archivos Pelicula (.cpp y .h)
 - Archivos Info (.cpp y .h)
 - Archivos .txt



```
* @var $OOLICEN
*/
define('PSI_INTERNAL_XML', false);
if (version_compare("5.2", PHP_VERSION
    die("PHP 5.2 or greater is requir
}) if (!extension_loaded("pcre")) {
    die("phpSysInfo requires the pcr
        properly.");
}
require_once APP_ROOT.'/includes/a
// Load configuration
require_once APP_ROOT.'/config.ph
if (!defined('PSI_CONFIG_FILE'))
    $tpl = new Template("/templa
        echo $tpl->fetch();
    }
}
```

Referencias

Canvas. (s. f.). Situación Problema. Recuperado 07 de junio de 2021, de <https://experiencia21.tec.mx/courses/140654/pages/situacion-problema>