

# Instituto Tecnológico de Estudios Superiores de Monterrey



## ***Situación Problema***

Programación Orientada a Objetos - TC 1030.800

### **Profesor:**

Gustavo Emilio De La Rosa Guerra

### **Nombres:**

Carlos Iñaki Román Martínez - A01702712

Yahir Rivera Huerta - A00572029

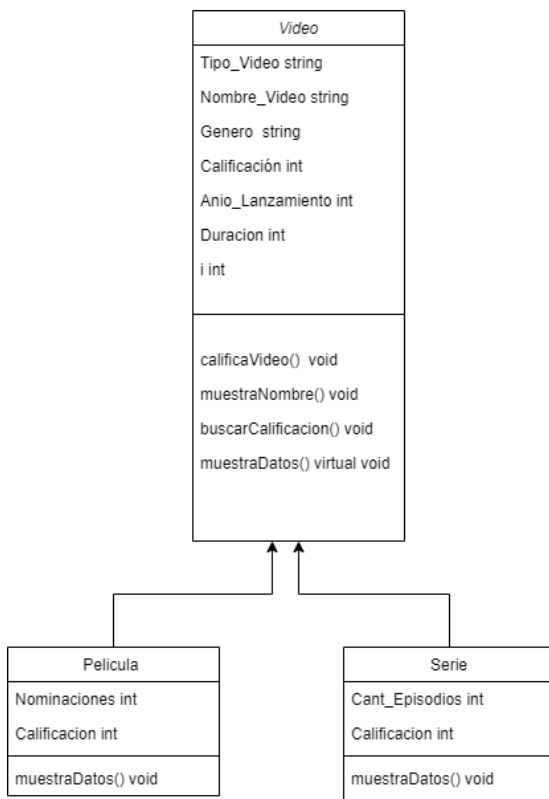
## ***Índice***

1. Introducción (planteamiento del problema)
2. Diagrama de clases UML
3. Ejemplo de ejecución
4. Argumentación de las partes del proyecto (a-e)
5. Identificación de casos que harían que el código falle
6. Conclusiones Personales

## Introducción

A lo largo de la historia el ser humano ha encontrado formas diferentes de entretenimiento, teniendo ejemplos como los deportes en vivo, la televisión, el cine y más recientemente el streaming de películas y series. El streaming es una nueva forma de consumir contenido en video, en audio e incluso de videojuegos, abriendo así un nuevo horizonte a las posibilidades del uso del internet. Por eso se nos dio la tarea de realizar un software que simule esto, un servicio de streaming el cual tenga un catálogo de películas y series.

## Diagramas de clases



## ***Ejemplos de ejecución***

Menú principal:

En el menú principal se muestran cinco opciones, las cuales van desde mostrar el catálogo de títulos, hasta calificar cada uno de esos títulos.

```
**** MENU ****
1.- Mostrar Videos en genral
2.- Mostrar series con una calificacion
3.- Mostrar peliculas con una calificacion
4.- Calificar Video
0.- Salir
█
```

Opción 1:

En la opción 1 nos muestra una serie de títulos de las películas y series que la plataforma tiene a nuestra disposición, teniendo tres películas y dos series.

```
**** MENU ****
1.- Mostrar Videos en genral
2.- Mostrar series con una calificacion
3.- Mostrar peliculas con una calificacion
4.- Calificar Video
0.- Salir
1

*** Mostrar Videos en general ***

Pelicula uno:
El nombre del video es: Hola Mundo

Pelicula dos:
El nombre del video es: Adios Mundo

Pelicula tres:
El nombre del video es: Mundo

Serie uno:
El nombre del video es: Hola

Serie dos:
El nombre del video es: Mundo
```

Opción 2:

En esta opción nos pide que ingresemos una calificación por la cual queramos ver un video, en este caso se ingresa el diez y nos muestra que las dos series que hay en el catálogo tienen esa calificación.

```
**** MENU ****
1.- Mostrar Videos en genral
2.- Mostrar series con una calificacion
3.- Mostrar peliculas con una calificacion
4.- Calificar Video
0.- Salir
2
Introduce la calificacion a buscar:
10

El tipo de video es: Serie
El nombre del video es: Hola
El genero del video es: Panico
La calificacion del video es: 10
El ano de lanzamiento es: 2021
La duracion del video es: 100
La cantidad de episodios es: 8

El tipo de video es: Serie
El nombre del video es: Mundo
El genero del video es: Amistad
La calificacion del video es: 10
El ano de lanzamiento es: 2021
La duracion del video es: 100
La cantidad de episodios es: 6
```

Ahora si ingresamos una calificación la cual no esté en el catálogo nos enviará el mensaje de que no hay algún video de esa calificación.

```
**** MENU ****
1.- Mostrar Videos en genral
2.- Mostrar series con una calificacion
3.- Mostrar peliculas con una calificacion
4.- Calificar Video
0.- Salir
2
Introduce la calificacion a buscar:
7
No hay un video con esa calificacion
```

Opción 3:

Esta opción es exactamente igual a la pasada pero ahora solo tomando en cuenta las películas que tiene el catálogo.

```
**** MENU ****
1.- Mostrar Videos en genral
2.- Mostrar series con una calificacion
3.- Mostrar peliculas con una calificacion
4.- Calificar Video
0.- Salir
3
Introduce la calificacion a buscar:
10

El tipo de video es: Pelicula
El nombre del video es: Hola Mundo
El genero del video es: Terror
La calificacion del video es: 10
El anio de lanzamiento es: 2021
La duracion del video es: 100
La cantidad de nominaciones es: 10
```

De igual manera nos muestra que no hay ninguna película con una calificación de cinco.

```
**** MENU ****
1.- Mostrar Videos en genral
2.- Mostrar series con una calificacion
3.- Mostrar peliculas con una calificacion
4.- Calificar Video
0.- Salir
3
Introduce la calificacion a buscar:
5
No hay un video con esa calificacion
```

Opción 4:

En esta opción nos encontramos con otro menú el cual nos pregunta si queremos calificar las películas o las series.

```
**** MENU ****
1.- Mostrar Videos en genral
2.- Mostrar series con una calificacion
3.- Mostrar peliculas con una calificacion
4.- Calificar Video
0.- Salir
4

*** Menu de calificacion ***
1.- Pelicula
2.- Serie
0.- Salir
█
```

Al seleccionar las películas nos irá preguntando una por una qué calificación será la que tú le des.

```
*** Menu de calificacion ***
1.- Pelicula
2.- Serie
0.- Salir
1

Pelicula uno:
Cual es la calificacion que le das al video:
10

Pelicula dos:
Cual es la calificacion que le das al video:
10

Pelicula tres:
Cual es la calificacion que le das al video:
10
```

De igual manera pasa lo mismo con las series.

```
*** Menu de calificacion ***
1.- Pelicula
2.- Serie
0.- Salir
2

Serie uno:
Cual es la calificacion que le das al video:
8

Serie dos:
Cual es la calificacion que le das al video:
9
```

Después de esto nos regresará al menú principal donde puedes repetir todas las acciones pasadas o salir del sistema.

### ***Argumentación de las partes del proyecto***

- a) Para solucionar la situación problema optamos por utilizar clases de forma que podamos reutilizar código. Creamos una clase llamada video donde se encuentra la mayoría de los atributos y métodos que utilizaremos en el programa. Después se creó la clase película y la clase serie que pertenecen y heredan de la clase video. Cada clase tiene sus modificadores de acceso, constructores y las funcionalidades están en archivos diferentes para un mejor entendimiento del código.
- b) Para el programa utilizamos el concepto de herencia para declarar los atributos en común y el constructor en la clase padre y reutilizar código en las clases hijo. Para evitar problemas con los modificadores de acceso declaramos que las clases hijo sean parte del acceso público de la clase padre. En este caso utilizamos la clase padre como video ya que nos permite heredar los atributos genéricos que se utilizan en



ambas clases hijo. En la clase padre existe un constructor con parámetros para inicializar todos los atributos genéricos y en la clase hijo llamamos el constructor de la clase padre y le añadimos los atributos específicos de cada clase.

- c) Los modificadores de acceso nos han permitido que los atributos y métodos de las clases sean encapsulados y de esta forma utilizarlos solo de la manera correcta. Para este programa en cada clase hemos declarado los atributos en la zona de privado para que partes externas de la clase no puedan modificarlos. El constructor, métodos, getters y setters fueron declarados en la sección pública para que podamos acceder a atributos y métodos fuera de la clase.
- d) Para este programa nos resultó muy útil utilizar polimorfismo y funciones virtuales. En la clase padre Video creamos una función virtual llamada *muestraDatos()* que nos permite imprimir en la pantalla los datos de un cierto video. Como en la clase padre tenemos solo los datos genéricos y en las clases hijo tenemos más datos incluyendo los genéricos, decidimos utilizar la misma función en las clases heredadas con más atributos que son exclusivos de la clase heredada. La clave de una función polimórfica es que si en la clase heredada no tenemos declarada la misma función con otros parámetros, el programa utilizará la función declarada en la clase padre.
- e) Al hablar de sobrecarga de operadores en programación orientada a objetos es una herramienta muy importante ya que estamos creando tipos de datos abstractos que con los operadores convencionales no se podría hacer operaciones. En el programa podríamos sobrecargar el operador igual para poder comparar si el título de una serie y el de una película son iguales.

### ***Identificación de casos de fallos***

Al momento de estar calificando el programa no acepta números negativos y se cicla infinitamente.

## ***Conclusiones***

Este proyecto nos ha permitido desarrollar nuestras habilidades en la programación en c + + diseñando estructuras de información pensando en objetos. Este proyecto nos ha permitido buscar soluciones eficientes a problemas de manejo de información abstracta y reutilización de código. Sin duda después de este proyecto estamos más confiados para desarrollar programas con este tipo de estructura que a la vez eficiente el desarrollo de software de forma considerable.

## ***Liga al código***

[https://drive.google.com/drive/folders/17AzJcnBFEjnyK5qOL\\_6ozQ1drKa2xpBD?usp=sharing](https://drive.google.com/drive/folders/17AzJcnBFEjnyK5qOL_6ozQ1drKa2xpBD?usp=sharing)