Instituto Tecnológico y de Estudios Superiores de Monterrey

Programación Orientada a Objetos

Grupo 300

Periodo 3
Reporte Proyecto Integrador
Profesor: MTI Gabriela Zamora Leal

Elaborado por:

Luis Antonio Barajas Ramírez
Campus Monterrey
A01235589
A01235589@tec.mx

15 de junio del 2022

Monterrey, Nuevo León

INDICE

Introducción	3
UML	4
Diseño	5
Ejecución	8
Conclusión	10
Referencias	10

INTRODUCCION

En los últimos años, han proliferado los servicios de streaming de video bajo demanda por ejemplo Netflix, Disney, 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. Una versión limitada para apoyar a un futuro proveedor de este tipo de servicios.

Se requiere crear un prototipo de software de videos: películas y series. Este prototipo guardara y mostrara estos videos hacia el usuario que son ingresados por este mismo. Las películas tendrán su título, duración, id, género y calificación. Las series tendrán los mismos atributos, además, las series tendrán sus propios episodios los cuales tendrán sus títulos y calificaciones. Se mostrará el siguiente menú:

1. Cargar archivo de datos
2. Mostrar los videos en general
3. Mostrar los episodios de una serie seleccionada
4. Mostrar las películas con una cierta calificación
5. Calificar un video
0. Salir

1. Se cargarán las películas con sus títulos, ids, duraciones, géneros y calificaciones. Se agregarán las películas que el usuario quiera. Después, el usuario ingresara la información de las series (id, titulo, genero, calificación, numero de episodios) y a continuación ingresara los datos que son: título y calificación por cada episodio, de los episodios de su respectiva serie

2. Se mostrarán todos los videos, es decir, todas las películas y todas las series con sus respectivos episodios.

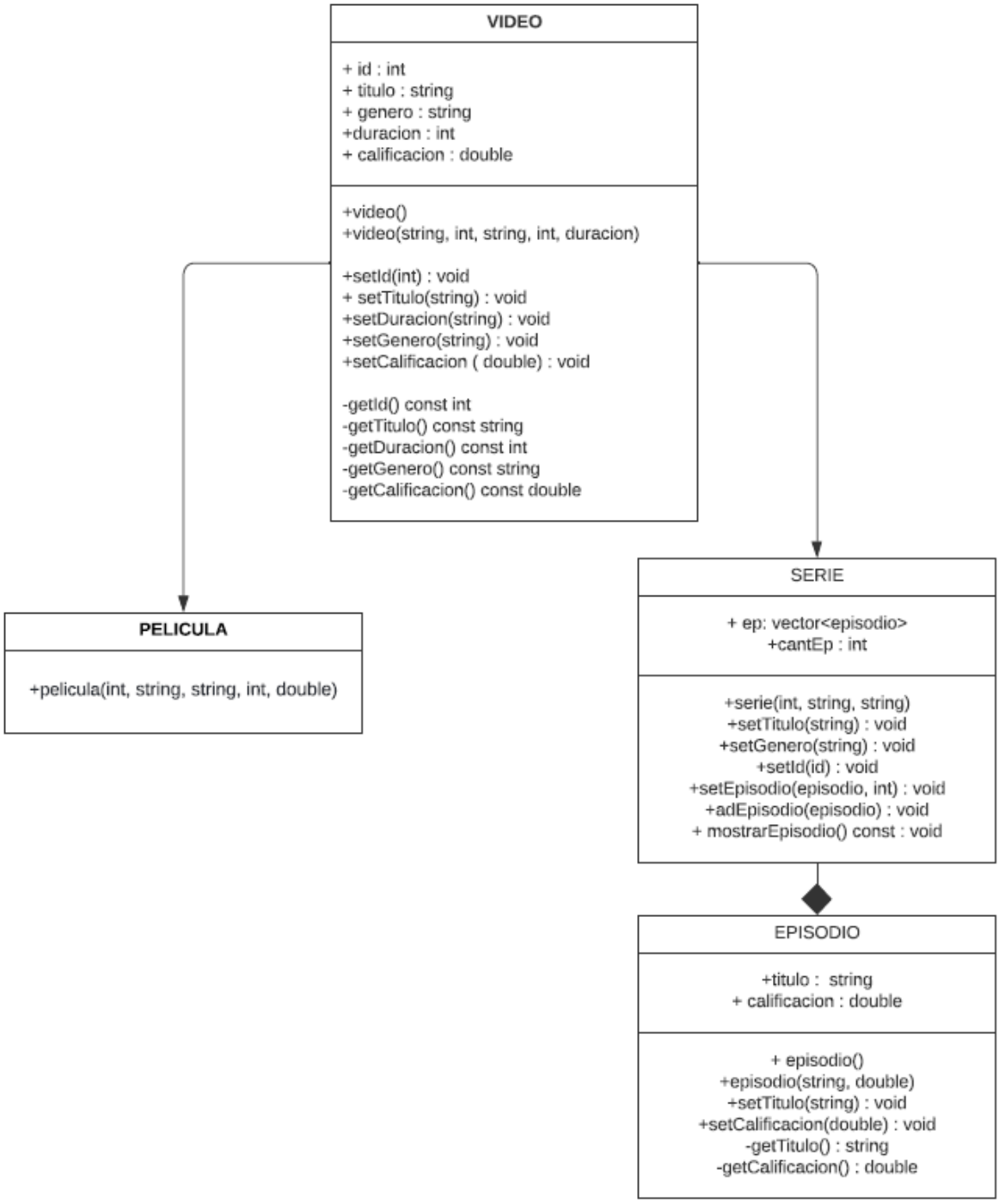
3. Se mostrarán los episodios, con su título y calificación, de una serie que el usuario quiera.

4. Se mostraran las películas con su id, titulo y su clasificación correspondiente.

5. Se le cambiara la calificación a una película que el usuario quiera.

0. Saldrá del menú y el programa terminara.

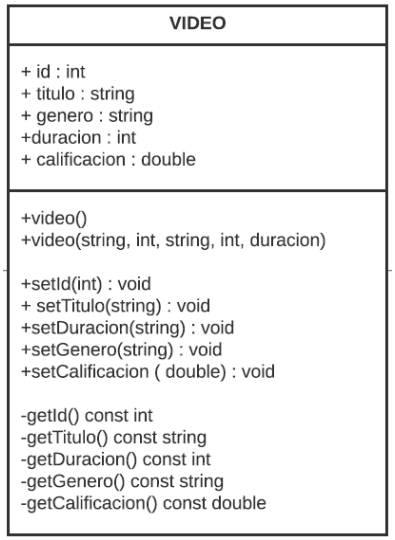
UML


VIDEO
+ id : int
+ titulo : string
+ genero : string
+duracion : int
+ calificacion : double
+video()
+video(string, int, string, int, duracion)
+setId(int) : void
+ setTitulo(string) : void
+setDuracion(string) : void
+setGenero(string) : void
+setCalificacion (double) : void
-getId() const int
-getTitulo() const string
-getDuracion() const int
-getGenero() const string
-getCalificacion() const double
PELICULA
+pelicula(int, string, string, int, double)
SERIE
+ ep: vector<episodio>
+cantEp : int
+serie(int, string, string)
+setTitulo(string) : void
+setGenero(string) : void
+setId(id) : void
+setEpisodio(episodio, int) : void
+adEpisodio(episodio) : void
+ mostrarEpisodio() const : void
EPISODIO
+titulo : string
+ calificacion : double
+ episodio()
+episodio(string, double)
+setTitulo(string) : void
+setCalificacion(double) : void
-getTitulo() : string
-getCalificacion() : double

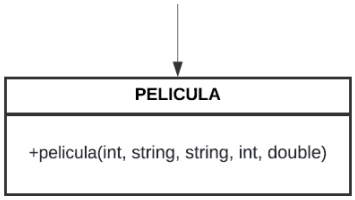
DISEÑO

Para el desarrollo de este prototipo se basó en programación Orientada a Objetos debido a la estructura del proyecto y para su simplicidad. Se utilizan : clases, objetos, herencia , polimorfismo, constructores, sobrecarga de constructores, composición, etc.

Nuestra clase “padre” o clase de la que se heredaran los atributos y métodos será “Video”, ya que las películas y series ambas son videos que comparten los mismos atributos y métodos. Los atributos serán: id, titulo, genero, duración y calificación; todos estos serán compartidos por todas las películas y series que se creen. Tendrá su constructor por default y su constructor con parámetros inicializar los atributos. Los atributos tendrán sus respectivos setter y getters para darles un valor y regresar el atributo que se solicita.



Después se crean las clases “hijas” o que heredan de “video” las cuales son: películas y serie. En la clase película únicamente se utiliza un constructor debido a que todos los atributos y métodos que requiere ya los tiene la clase “padre” que en este caso se Video: id, titulo , genero , duración y calificación.

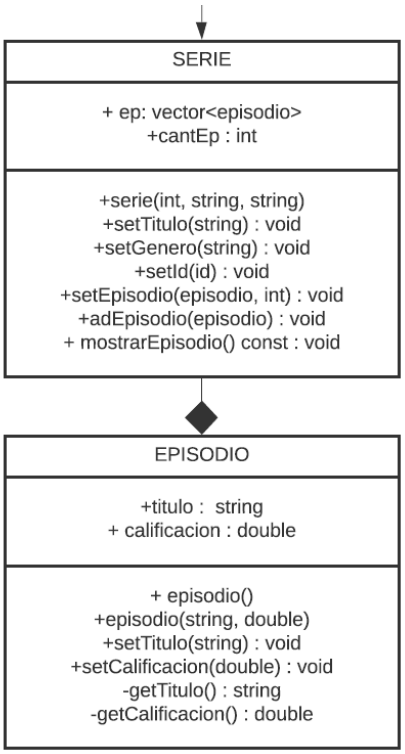


La clase SERIE es un poco mas compleja debido a que cada serie contiene un numero de episodios con su respectivo título y calificación, por lo que seria un arreglo dentro de

episodios dentro de una serie. Sin embargo, debido a que son muchas series (no solamente 1), también habrá un arreglo de series, por lo que podemos concluir que se crearía un arreglo dentro de un arreglo, en otras palabras: muchas series y cada serie, con muchos episodios.

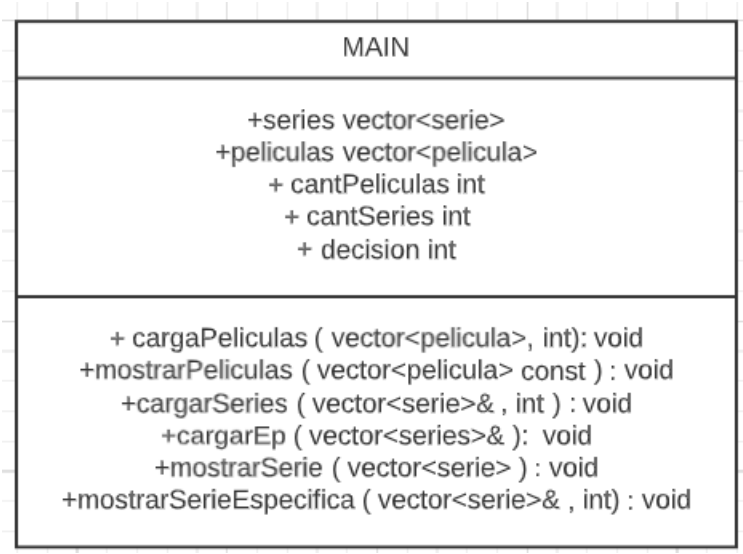
Los atributos de nuestra clase SERIE son: un vector de tipo episodio para poder guardar los episodios (ya con sus atributos de título y calificación) de una serie en su respectiva serie, y la cantidad de episodios. Se crea su constructor inicializar su nombre, genero y id de la serie, continuando con sus respectivos métodos para asignarles un valor. Después los métodos para establecer un episodio (setEpisodio), agrégalo al vector de episodios (adEpisodio) y al final mostrar todos los episodios de la serie (mostrarEpisodios).

Se utilizo un vector debido a su facilidad de modificación de datos ya que no se le tiene que asignar un tamaño definido, es decir, puede ir aumentando conforme se ingresan mas datos. Como no sabemos la cantidad de episodios que tiene una serie, es por eso por lo que utilizo un vector.



La tercera clase es EPISODIO. Esta clase se utiliza en la clase SERIE como composición para crea los episodios de cada serie. Los episodios tendrán dos atributos: el título del episodio y la calificación de ese episodio. Se crean sus ambos constructores: default y con parámetros para la inicialización. Después sus métodos setter y getter para modificar su título y calificación.

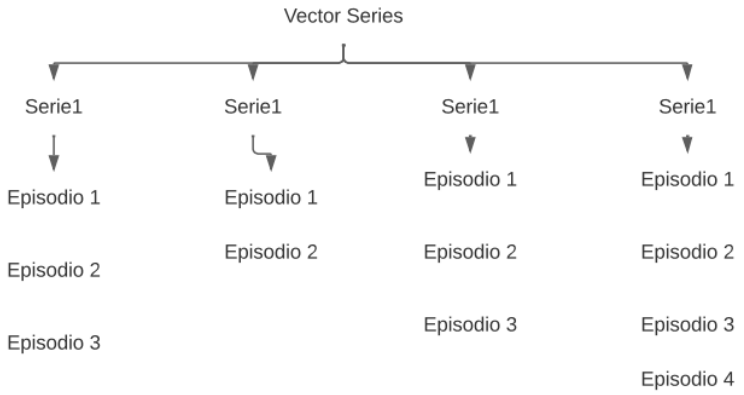
Con esto podemos entender mejor lo que anteriormente se explicaba en la clase VIDEO. El vector episodio, será un vector de varios episodios y cada uno tendrá su título y calificación. Pero como son varias series las que existen, será un vector de episodios por cada serie.

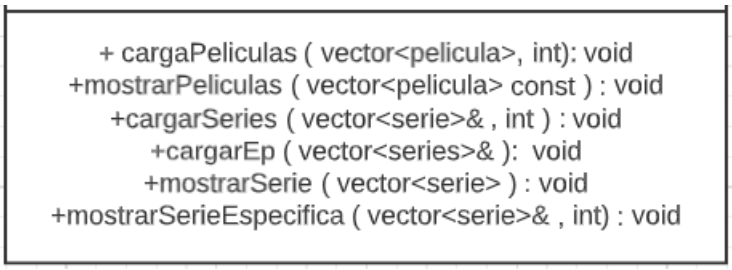


En el apartado MAIN, se juntarán todas las clases. De inicio, se crean dos objetos tipo vectores de tipo película y serie con la finalidad de guardar todas las películas y series. Además se ingresan las cantidades de películas y series que se usan en las funciones de mostraPeliculas() y cargarSerie(), y una variables de decisión del menú.

El vector de tipo películas, será un arreglo que guardara las películas con sus atributos los cuales son: id, titulo, genero, duración y calificación.

Lo complejo es el vector de tipo serie. El vector de tipo serie guardara todas las series con sus atributos de id, titulo, género y calificación; pero además se guardaran los episodios de cada una de las series como lo comentamos anteriormente. Serán un vector de episodios en una serie, que esa serie estará dentro que otro vector que contiene mas series.





La función cargaPeliculas() se usa para ingresar los datos de todas las películas y mostraPeliculas() imprimir el vector de películas. La función de cargaSeries() se usa para ingresar los datos de las series: id, titulo y género. Después se usa la función de cargaEp() en el cual se ingresan los datos de los episodios de una serie y se agregan al vector episodio que después se guarda en su respectiva serie. La función mostrarSerie() imprime todas las series con sus respectivos episodios. Finalizando, la función mostrarSeriesEspecifica() muestra una serie que el usuario selecciono con sus respectivos episodios.

EJECUCION

```
1. Carga archivo de datos
2. Mostrar los videos en general
3. Mostrar los episodios de una serie seleccionada
4. Mostrar las peliculas con una cierta calificacion
5. Calificar un video
0. Salir
1
```

Primero se muestra el menú de decisión, para saber que es lo que usuario quiere hacer. Seleccionando la primera opción (cargar los archivos de película y serie) pasamos a cargar los datos de cada película y cada serie.

```
PELICULAS

Cantidad de peliculas: 2
ID: 1
Nombre de la pelicula: pelicula1
Calificacion: 10
Duracion: 120
Genero: accion
ID: 2
Nombre de la pelicula: pelicula2
Calificacion: 9
Duracion: 125
Genero: drama
```

```
SERIES

Cantidad de series: 2
ID de la serie: 3
Nombre de la serie: serie1
Genero de la serie: suspenso
ID de la serie: 4
Nombre de la serie: serie2
Genero de la serie: comedia
```

```
Cantidad de episodios: 2
nombre del episodio: episodio1
Calificacion para este episodio: 8
nombre del episodio: episodio2
Calificacion para este episodio: 6
Cantidad de episodios: 3
nombre del episodio: episodio3
Calificacion para este episodio: 5
nombre del episodio: episodio4
Calificacion para este episodio: 10
nombre del episodio: episodio4
Calificacion para este episodio: 9
```

En las imágenes de arriba, ingresamos los datos de dos películas, 2 series y 5 episodios.

Al volver al menú, si seleccionamos la opción 2, nos mostrara todos los videos: películas y series.

```
PELICULAS

1 pelicula1 10
2 pelicula2 7

SERIES

serie1 suspenso
episodio1 4
episodio2 7
serie2 comedia
episodio3 2
episodio4 9
episodio5 10
```

Nos muestra las películas con sus respectivos ids, títulos y calificación. Después se imprimen las series con su título y género, por debajo sus episodios.

Regresamos al menu y si seleccionamos la opción 3, tendremos se ingresar el id de la serie que buscamos, en este caso dimos el ID 3, lo que nos arrojo la serie “serie1” con sus episodios y calificaciones.

```
ID de la serie que buscas:
3
serie1
episodio1 8
episodio2 6
```

De vuelta al menú, ahora elegimos la opción 4, la cual nos imprime las películas con únicamente su título y su calificación.

```
PELICULAS

1 pelicula1 10
2 pelicula2 9
```

En la última opción 5, nos permite modificar la calificación de alguna película que elijamos o queramos, proporcionando el Id de la película. En la imagen de la derecha podemos ver que la calificación de “pelicula2” cambio a 7 después de haberla modificarla.

```
ID de pelicula a calificar: 2
Nueva Calificacion: 7
Calificacion cambio
```

```
PELICULAS

1 pelicula1 10
2 pelicula2 7
```

CONCLUSION

Con uso de las propiedades y características la programación orientada a objeto se desarrollo un prototipo de software sobre una plataforma de videos. La herencia y polimorfismo de clase nos facilito el desarrollo y redujo el tamaño del código y de programación. Se determino que un programa que incluya todos los aspectos de programación orientada a objetos es muy potente para poder desarrollar prototipos de plataformas entre muchas otras cosas.

REFERENCIAS

C++ OOP (Object-Oriented Programming). (2019). W3schools.

https://www.w3schools.com/cpp/cpp_oop.asp

IMB. (2021, August 17). *Programación orientada a objetos*. IBM.

https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=language-object-oriented-programming