

Evidencia Proyecto Integrador



Programación Orientada a Objetos

Equipo 9

A01721371 Gerardo Carlos Ferrigno Califa

A01194283 Marcelo Hernández Almada

Abril 25, 2021

Maestra: Ing. María Esperanza Garza Leal

Índice:

Introducción	3
Diagrama de clases UML	4
Casos de prueba	5-7
Argumentación	8
Casos en los que el proyecto deja de funcionar	9
Conclusión de Gerardo Ferrigno	10
Referencias	11

Introducción (planteamiento del problema).

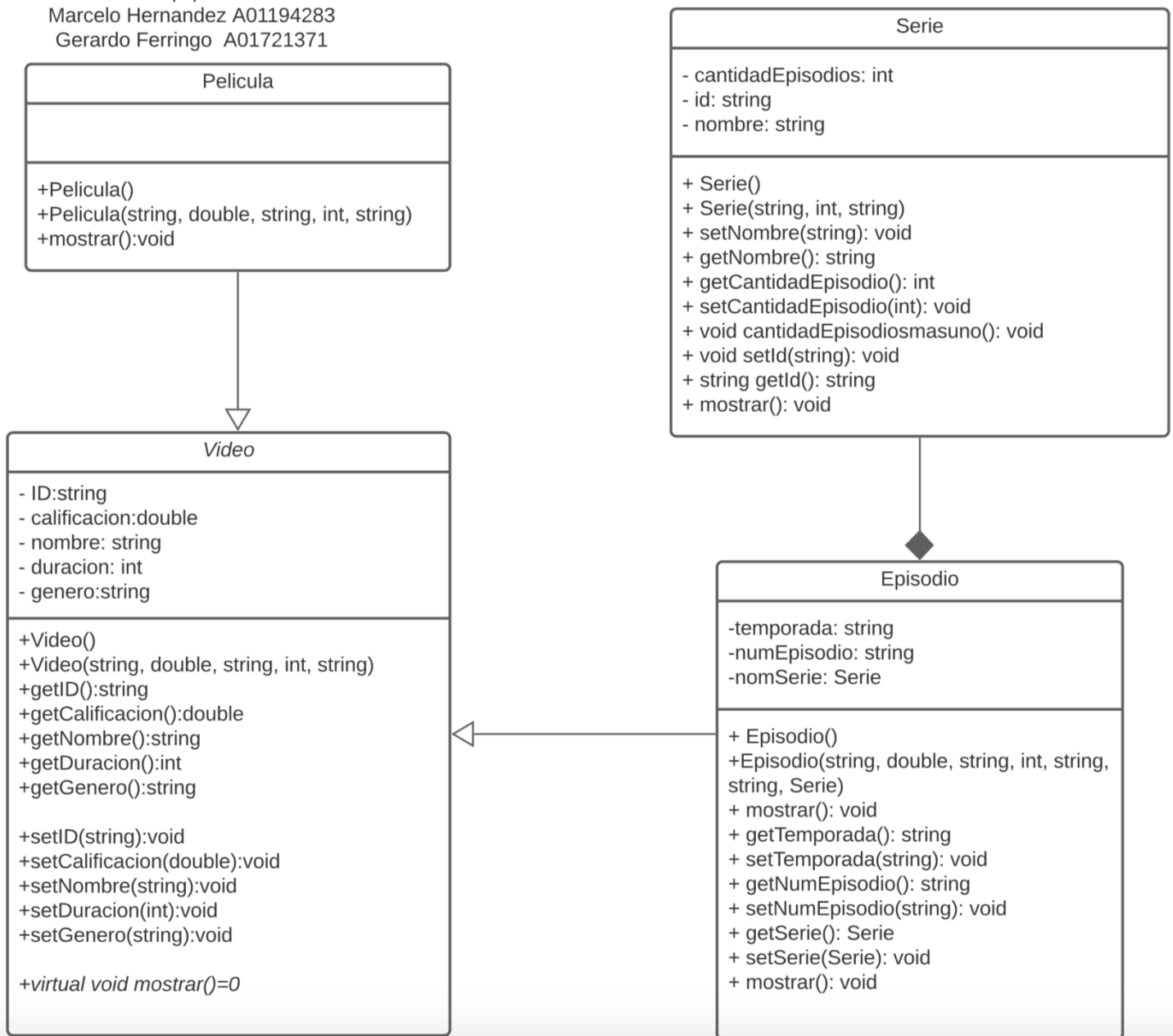
En el presente proyecto integrador se nos pidió desarrollar un servicio de streaming. Se requiere crear una versión de este para un proveedor. Este servicio contiene dos tipos de videos: series y películas. Los videos que se proporcionan tienen características como una duración, una identificación, un género y obviamente un nombre. A diferencia de las películas, las series también tienen episodios. Los episodios tienen un título y una temporada. La información sobre los videos se recibe en dos archivos. En uno se dan de alta las series, y en otro hay diferentes videos. El programa lee los archivos y crea los objetos.

El sistema diseñado debe de poder llevar a cabo las siguientes acciones:

- Cargar el archivo de datos (para asignar la información).
- Mostrar todas las películas.
- Mostrar todas las series.
- Mostrar los videos con una cierta calificación.
- Mostrar los videos de un género pedido.
- Mostrar los episodios de cierta serie con sus calificaciones.
- Mostrar películas que tienen cierta calificación.
- Salir del programa.

Diagrama de clases UML

Equipo 9
Marcelo Hernandez A01194283
Gerardo Ferringo A01721371



Este es el diagrama UML con los cambios hechos en el desarrollo completo. Inicialmente se tenía una clase de más, la clase de Temporada, diferentes atributos, y diferentes atributos en episodio y en serie. Ahora solamente episodio y película heredan de video.

Casos de prueba (capturas de pantalla).

El usuario carga el archivo y pide la segunda opción (la primera es la carga de archivo):

```
1: Cargar archivo de datos:
2: Mostrar la lista de películas:
3: Mostrar la lista de series:
4: Mostrar los videos en general con una cierta calificación:
5: Mostrar los videos en general de un cierto género:
6: Mostrar los episodios de una determinada serie con sus calificaciones
:
7: Mostrar las películas con cierta calificación:
8: Salir

2
Nombre de pelicula: Avatar
ID: 110
Calificacion: 5
Duracion: 161
Genero: Ciencia_ficcion

Nombre de pelicula: Batman_El_caballero_de_la_noche
ID: 111
Calificacion: 4
Duracion: 152
Genero: Accion

Nombre de pelicula: Guason
ID: 112
Calificacion: 5
Duracion: 121
Genero: Drama
```

El usuario pide la tercera opción (esto después de la carga de archivo):

```
1: Cargar archivo de datos:
2: Mostrar la lista de películas:
3: Mostrar la lista de series:
4: Mostrar los videos en general con una cierta calificación:
5: Mostrar los videos en general de un cierto género:
6: Mostrar los episodios de una determinada serie con sus calificaciones
:
7: Mostrar las películas con cierta calificación:
8: Salir

3
Nombre de la serie: Flash
ID: 501
Cantidad Episodios: 6

Nombre de la serie: Friends
ID: 502
Cantidad Episodios: 6

Nombre de la serie: New_Amsterdam
ID: 503
Cantidad Episodios: 3
```

El usuario pide la cuarta opción con cierta calificación (en este caso fue 3), esto después de la carga de archivo:

```
1: Cargar archivo de datos:
2: Mostrar la lista de películas:
3: Mostrar la lista de series:
4: Mostrar los videos en general con una cierta calificación:
5: Mostrar los videos en general de un cierto género:
6: Mostrar los episodios de una determinada serie con sus calificaciones
:
7: Mostrar las películas con cierta calificación:
8: Salir

4
Sobre cual calificacion deseas buscar?
3
Nombre de pelicula: Guardianes_de_la_galaxia
ID: 120
Calificacion: 3
Duracion: 118
Genero: Ciencia_ficcion

Se encontraron 1 videos
```

El usuario pide la quinta opción con cierto género (en este caso fue comedia):

```
1: Cargar archivo de datos:
2: Mostrar la lista de películas:
3: Mostrar la lista de series:
4: Mostrar los videos en general con una cierta calificación:
5: Mostrar los videos en general de un cierto género:
6: Mostrar los episodios de una determinada serie con sus calificaciones
:
7: Mostrar las películas con cierta calificación:
8: Salir

5
Sobre que genero deseas buscar? (Primera laetra en mayuscula)
Comedia
Numero de temporada: T1
Nombre de Episodio: The_One_with_The_Thumb
ID: 216
Calificacion: 4
Duracion: 39
Genero: Comedia

Numero de temporada: T1
Nombre de Episodio: The_One_Where_Nana_Dies_Twice
ID: 217
Calificacion: 4
Duracion: 42
Genero: Comedia
```

Finalmente el programa muestra cuantos videos se encontraron que cumplían con lo que ingresó el usuario.

El usuario pide la sexta opción con el nombre de cierta serie (en este caso fue Friends):

```
1: Cargar archivo de datos:
2: Mostrar la lista de películas:
3: Mostrar la lista de series:
4: Mostrar los videos en general con una cierta calificación:
5: Mostrar los videos en general de un cierto género:
6: Mostrar los episodios de una determinada serie con sus calificaciones
:
7: Mostrar las películas con cierta calificación:
8: Salir

6
Que serie deseas buscar? (Primera letra en mayuscula y los espacios con guion bajo por favor)
Friends
Numero de temporada: T1
Nombre de Episodio: The_One_with_The_Thumb
ID: 216
Calificacion: 4
Duracion: 39
Genero: Comedia

Numero de temporada: T1
Nombre de Episodio: The_One_Where_Nana_Dies_Twice
ID: 217
Calificacion: 4
Duracion: 42
Genero: Comedia
```

Finalmente muestra la cantidad de episodios que se tienen de esa serie.

El usuario pide la séptima opción con una cierta calificación (en este caso fue 5):

```
5: Mostrar los videos en general de un cierto género:
6: Mostrar los episodios de una determinada serie con sus calificaciones
:
7: Mostrar las películas con cierta calificación:
8: Salir

7
Que calificacion de pelicula deseas buscar?
5
Nombre de pelicula: Avatar
ID: 110
Calificacion: 5
Duracion: 161
Genero: Ciencia_ficcion

Nombre de pelicula: Guason
ID: 112
Calificacion: 5
Duracion: 121
Genero: Drama

Nombre de pelicula: Avengers_Endgame
ID: 117
Calificacion: 5
Duracion: 181
Genero: Ciencia_ficcion

Se encontraron 3 peliculas
```

Argumentación de las partes del proyecto relacionadas con cada uno de los puntos a) a e) mencionados arriba, con una argumentación de el por qué se optó por esa solución y no por otras.

Para desarrollar el diagrama de clases decidimos usar lucidChart, esto porque es una herramienta que permite que la gente trabaje en un canvas al mismo tiempo. Logrando que la colaboración sea sencilla y fluida. También, porque es una herramienta que hemos utilizado previamente y ya sabemos la manera en la que funciona.

Para nombrar las clases, las variables y las funciones de nuestro programa, decidimos utilizar la nomenclatura sugerida por los estándares de programación, esto para que nuestro código sea entendible, debido a que de esta manera sabes diferenciar entre, variables, clases y funciones, y puedes deducir su propósito a partir de la manera en la que fueron nombradas.

Todas nuestras clases incluyen sus atributos, constructores default y con parámetros, y las funciones de esta misma como los sets and gets. Los constructores son sumamente importantes para que se puedan crear objetos de la clase con la definición correcta sin necesidad de utilizar los sets para cada atributo. Las funciones dentro de la clase sirven para que estas mismas no estén en tu main y el archivo esté más ordenado.

Dentro del programa nos dimos cuenta que para que este funcione de manera correcta eran necesario el polimorfismo, la herencia, arreglos de apuntadores a objetos, la composición y arreglos de objetos. El polimorfismo era necesario para que la clase abstracta de videos pueda tener una función mostrar() distinta en el tipo película o el tipo episodio. La herencia se daba entre la clase video, esta heredaba a las clases episodio y película. El arreglo de apuntadores existe para poder tener un arreglo de la clase abstracta Video que incluya películas y episodios. La composición era necesaria dentro de la clase episodio, esta tenía una variable del tipo serie. Por último creamos un arreglo de objetos para que el programa lea el documento de las series y obtenga las distintas series hechas objeto. No utilizamos la sobrecarga de operadores ni el manejo de excepciones porque no fue necesario para que nuestro proyecto funcionara.

Decidimos situar el menú y las funciones en diferentes documentos para que nuestro programa sea más legible.

Identificación de casos que harían que el proyecto deje de funcionar.

Primeramente, el programa no deja de funcionar si el usuario ingresa algo que no sea un número, solamente despliega un mensaje que dice que esta no es una función válida; para esto se usa `cin.fail()`. En el caso de que el usuario proporcione algún número que se encuentre fuera del rango de opciones, el programa vuelve a desplegar el menú.

Además del menú, si en el caso 4 y 7 se proporciona algo que no sea un número, el programa despliega que se encontraron cero videos. Igualmente se muestra lo mismo si la calificación se encuentra fuera del 1-5 o si no hay videos con la calificación pedida. Si en la opción 5 se proporciona un género inválido, el programa muestra que se encontraron cero videos de ese género. Si el usuario no carga los archivos antes de seleccionar alguna de las opciones, no se mostrará nada, debido a que no se leyó la información del archivo.

Otro incidente que haría que deje de funcionar el proyecto tiene que ver con los archivos. Primeramente, si existe una serie en el archivo de videos, que no existe en el archivo de series, esta no se guardaría, primero se dan de alta las series y luego se relacionan con sus episodios. Igualmente, si hay un error en el orden (o en los espacios, o en los tipos de datos) del archivo en el que se escriben los videos o las series, el programa dejaría de funcionar o no leería la información de manera adecuada. Lo que se escribe en el archivo debe de tener el mismo orden que en el que se carga en la función `leerArchivo()` para que funcione correctamente y debe de tener todos los valores.

Conclusión Personal. (Gerardo Ferrigno)

Para concluir, creo que logramos la manera más simple y directa de realizar lo que se nos planteó en la situación problema. Mi compañero Marcelo y yo colaboramos para obtener las mejores ideas y aplicarlas a nuestro programa. El proyecto funcionó para aplicar todo lo que hemos aprendido en las clases ya en un proyecto por nuestra cuenta. Creo que debido a lo que aprendí en las horas de clase, junto con mi compañero, pudimos crear el sistema de manera correcta. Creo que este proyecto me ayudó mucho a reforzar lo ya aprendido para en un futuro aplicarlo conforme avanza mi carrera.

Referencias

Object Oriented Programming in C++. GeeksforGeeks. (2020, April 28).
<https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/>.

Pointers in C and C++: Set 1 (Introduction, Arithmetic and Array). GeeksforGeeks. (2021, April 21).
<https://www.geeksforgeeks.org/pointers-in-c-and-c-set-1-introduction-arithmetic-and-array/>.

Applications of Pointers in C/C++. GeeksforGeeks. (2021, February 3).
<https://www.geeksforgeeks.org/applications-of-pointers-in-c-cpp/>.

References in C++. GeeksforGeeks. (2020, October 26).
<https://www.geeksforgeeks.org/references-in-c/>.