




Proyecto Integrador

Victor Alatorre

A00227368



INDICE

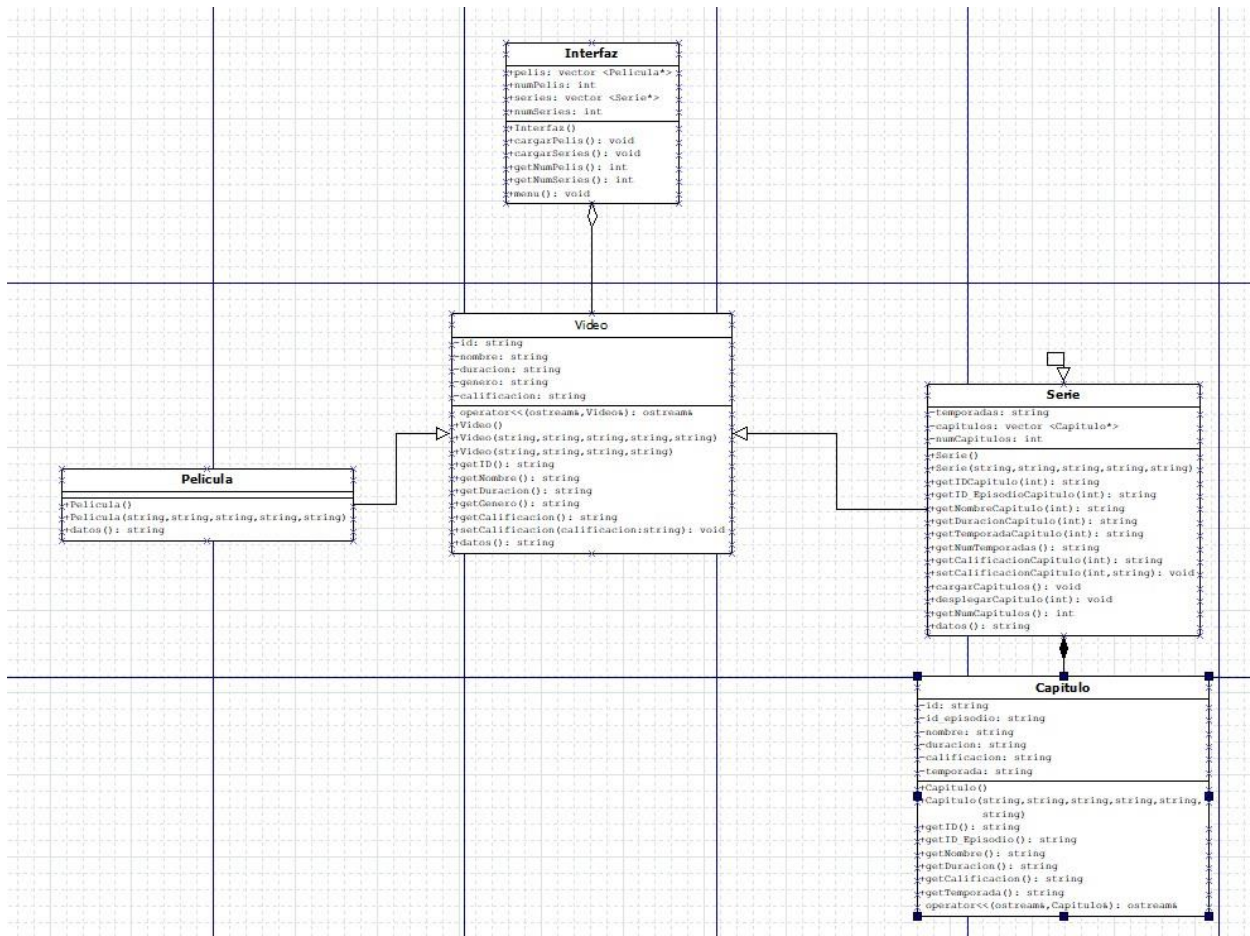
INTRODUCCION.....	2
DIAGRAMA UML	3
DEMOSTRACION DE EJECUCION.....	4
ARGUMENTACIÓN DEL DISEÑO DEL PROGRAMA.....	6
IDENTIFICACION DE POSIBLES ERRORES.....	7
CONCLUSION.....	8
REFERENCIAS.....	9

INTRODUCCION

El mundo esta en constante cambio, demandando cada vez tecnologías más nuevas e innovadoras, en muchas áreas, incluidas las de entretenimiento, lo que era la empresa BlockBuster ya no existe debido a esta constante evolución, y se han creado servicios de streaming que ahora han reemplazado a esta empresa y creado un nuevo modelo de negocio con este servicio innovador. Se presenta la situación que se debe diseñar un programa que funcione como plataforma de interfaz para servicios de streaming, como Netflix, Hulu, Etc. En este programa se tendrá que usar conceptos de programación orientada a objetos, para un diseño óptimo y eficiente.

Dentro de este proyecto se usan conceptos de programación como Herencia, polimorfismo, clases abstractas sobrecarga de operadores, excepciones, que serán explicados su utilidad y justificación más adelante en el documento, y por qué nos funcionan dentro de este contexto, así como también se cubrirán en que casos no funcionaría este programa.

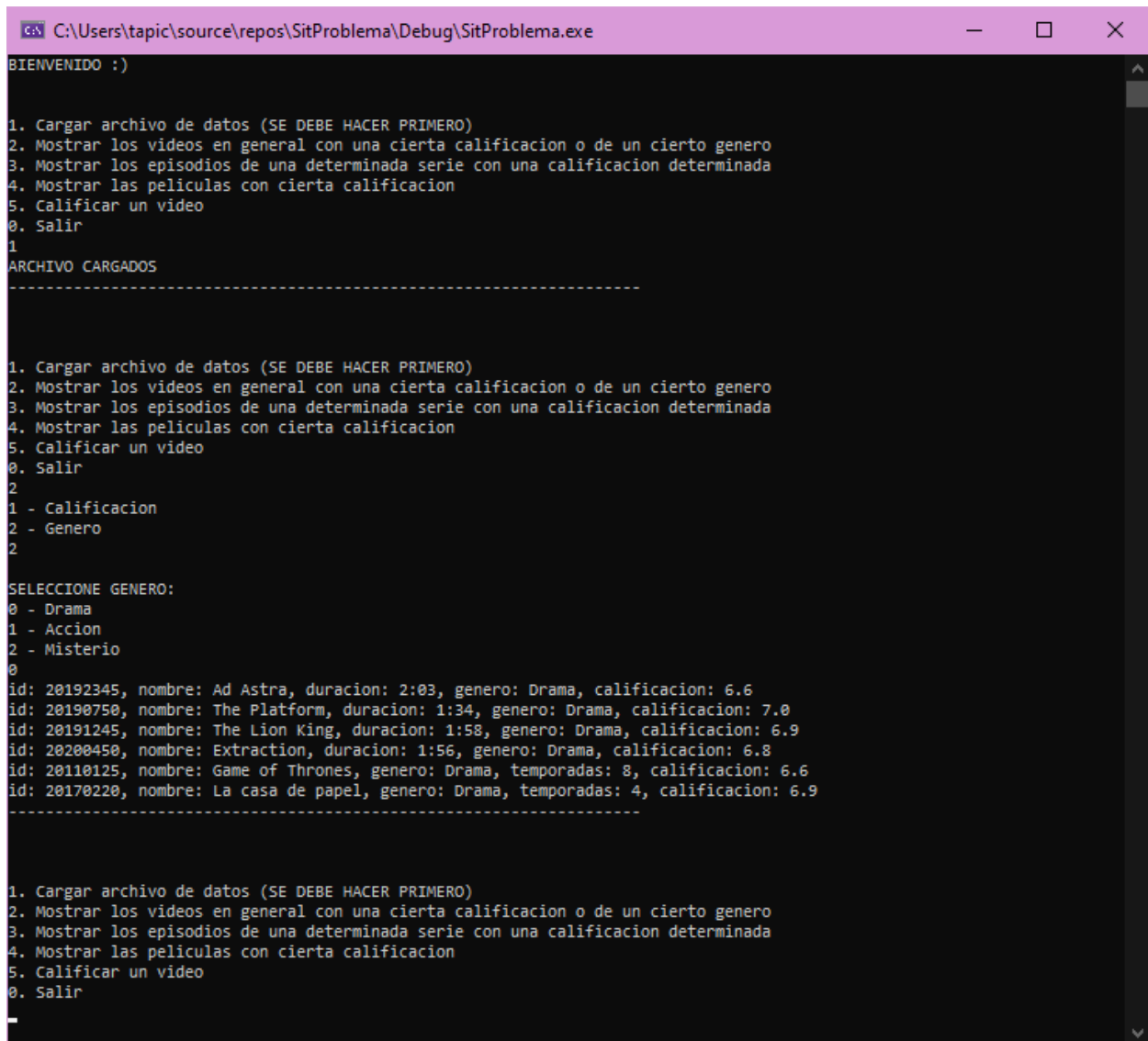
DIAGRAMA DE CLASES UML



-Este diseño permite, en pocas palabras, una reutilización de código y un funcionamiento ideal, ya que se hace uso de la herencia para ahorrarnos el trabajo de crear una clase desde cero para película y otra para serie, teniendo que implementar todos los getters y setters y las demás funciones que tanto Pelicula y Serie comparten. Como una serie esta formada de capítulos, se usa una relación de composición para unirlos a las series, donde los datos de los capítulos tienen que ser accedidos a través de su serie.

Todas las películas y las series se almacenan en la clase interfaz, que es donde ocurre la interacción con el usuario y con los datos.

DEMOSTRACION DE EJECUCION



```
C:\Users\tapic\source\repos\SitProblema\Debug\SitProblema.exe
BIENVENIDO :)

1. Cargar archivo de datos (SE DEBE HACER PRIMERO)
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 peliculas con cierta calificacion
5. Calificar un video
0. Salir
1
ARCHIVO CARGADOS
-----

1. Cargar archivo de datos (SE DEBE HACER PRIMERO)
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 peliculas con cierta calificacion
5. Calificar un video
0. Salir
2
1 - Calificacion
2 - Genero
2

SELECCIONE GENERO:
0 - Drama
1 - Accion
2 - Misterio
0
id: 20192345, nombre: Ad Astra, duracion: 2:03, genero: Drama, calificacion: 6.6
id: 20190750, nombre: The Platform, duracion: 1:34, genero: Drama, calificacion: 7.0
id: 20191245, nombre: The Lion King, duracion: 1:58, genero: Drama, calificacion: 6.9
id: 20200450, nombre: Extraction, duracion: 1:56, genero: Drama, calificacion: 6.8
id: 20110125, nombre: Game of Thrones, genero: Drama, temporadas: 8, calificacion: 6.6
id: 20170220, nombre: La casa de papel, genero: Drama, temporadas: 4, calificacion: 6.9
-----

1. Cargar archivo de datos (SE DEBE HACER PRIMERO)
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 peliculas con cierta calificacion
5. Calificar un video
0. Salir
-
```

-En este ejemplo, mediante la interfaz desplegamos videos que sean de género drama.

```
C:\Users\tapic\source\repos\SitProblema\Debug\SitProblema.exe
1. Cargar archivo de datos (SE DEBE HACER PRIMERO)
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 peliculas con cierta calificacion
5. Calificar un video
0. Salir
3
0 - Game of Thrones
1 - Dark
2 - Stranger Things
3 - La casa de papel
0
INGRESE LIMITE INFERIOR:
5
INGRESE LIMITE SUPERIOR:
7
id: 20110125, id episodio: 20110125-S08E04, nombre: The Last of the Starks, duracion: 1:18, calificacion: 5.5, temporada: 8
id: 20110125, id episodio: 20110125-S08E05, nombre: The Bells, duracion: 1:18, calificacion: 6, temporada: 8
-----
1. Cargar archivo de datos (SE DEBE HACER PRIMERO)
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 peliculas con cierta calificacion
5. Calificar un video
0. Salir
```

-En este ejemplo, mostramos episodios de una determinada serie especificando la calificación requerida.

```
C:\Users\tapic\source\repos\SitProblema\Debug\SitProblema.exe
-----
1. Cargar archivo de datos (SE DEBE HACER PRIMERO)
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 peliculas con cierta calificacion
5. Calificar un video
0. Salir
5
1 - PELICULA
2 - SERIE
2
SELECCIONE TITULO A CALIFICAR
0 - id: 20110125, nombre: Game of Thrones, genero: Drama, temporadas: 8, calificacion: 6.6
1 - id: 20170120, nombre: Dark, genero: Misterio, temporadas: 2, calificacion: 7.0
2 - id: 20160150, nombre: Stranger Things, genero: Accion, temporadas: 3, calificacion: 6.7
3 - id: 20170220, nombre: La casa de papel, genero: Drama, temporadas: 4, calificacion: 6.9
2
INGRESE PUNTAJE: 7.0
CAMBIO REALIZADO: id: 20160150, nombre: Stranger Things, genero: Accion, temporadas: 3, calificacion: 7.0
-----
1. Cargar archivo de datos (SE DEBE HACER PRIMERO)
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 peliculas con cierta calificacion
5. Calificar un video
0. Salir
```

-En este ejemplo se califica un título.

ARGUMENTACIÓN DEL DISEÑO DEL PROGRAMA

Este programa funciona desde una interfaz, donde se interactúa con el usuario y se trabajan también con los archivos de texto que contienen los datos de las películas, series y sus capítulos. Se utilizan métodos para cargar y leer los datos dentro de los archivos, y se almacenan los datos procesados en nuestros vectores que los contienen, así permitiendo almacenar de manera dinámica la cantidad de videos que se tienen, el uso de punteros permite este funcionamiento dinámico, sin tener que especificar el tamaño/cantidad de videos que se tengan.

De una clase base abstracta llamada video, donde se crean las funciones que serán comunes en nuestras clases derivadas Pelicula y Serie, donde Pelicula es diseñado solamente con constructores, y Serie tiene métodos extras para trabajar y acceder a los capítulos, de esta manera se restringe un acceso no deseado a los capítulos, siendo que solamente se pueden acceder a través de su serie.

Como polimorfismo en este programa, una función virtual pura y la sobrecarga de operadores se combinan para sobrecargar el operador <<, simplificando el código y permitiendo su lectura más simple y clara, se utilizan funciones virtuales puras ya que, no se necesita un objeto de la clase base, y en las clases derivadas el método virtual puro despliega información de la película o la serie, y deseamos que la información sea presentada diferente dependiendo de si es película o serie, debido a que se tienen diferentes datos.

Se utilizan excepciones para verificar que, al momento de calificar un título, se ingrese un valor de calificación que sea correcto, se utilizan excepciones para facilitar el funcionamiento, así como la lectura del código.

El programa funciona a través de un menú para facilitar la interacción con el usuario, aunque existen posibles errores de input de parte del usuario, que serán cubiertos a continuación.

IDENTIFICACION DE POSIBLES ERRORES

Este programa funciona a partir de los archivos cargados al programa, por lo que, si no se cargan de manera preliminar, el programa no tendrá los datos con los que se trabajan y no funcionarán los métodos.

También, como el programa funciona a través de un menú, el usuario puede ingresar accidentalmente un input inválido, que puede generar errores en el funcionamiento.

También existe la posibilidad que, cuando se quiera calificar un título, aunque el programa verifique que el numero este dentro del rango de nuestra calificación, el usuario puede ingresar un input que no sea un número, y por lo tanto generar errores de funcionamiento.

CONCLUSIÓN

Este programa, aunque con una funcionalidad simple, tiene un diseño sólido y bien optimizado, haciendo uso de las herramientas que permitan diseñar un programa de mejor calidad y un uso/desarrollo más simple y fácil.

Es gracias a proyectos como estos donde realmente puedo ensamblar mis conocimientos y darles un uso práctico, donde aprecio los retos que se me presentan para desarrollar mis habilidades y así poder superarme como ingeniero y como programador.

Me pareció enriquecedor que, durante el desarrollo del programa tuve que aprender con mis propios métodos ciertas herramientas y funciones que no están explícitamente requeridas para el funcionamiento, ya que esto alienta a un pensamiento más ingenieril donde tenga que buscar las soluciones por mi cuenta.

REFERENCIAS

<https://www.geeksforgeeks.org/pure-virtual-functions-and-abstract-classes/>

<https://www.geeksforgeeks.org/inheritance-in-c/>

<http://www.cplusplus.com/reference/string/string/getline/>