

Instituto Tecnológico de estudios superiores de Monterrey
Campus Estado de México
Departamento de Ingeniería

TC1030

Programación orientada a objetos

Grupo: 302

Dr. Jorge Adolfo Ramírez Uresti

“Situación problema”

Fecha: 10 / 06/ 2024

Estrella Lolbeth Téllez Rivas
Karen Vásquez Medina

A01750496
A01748805

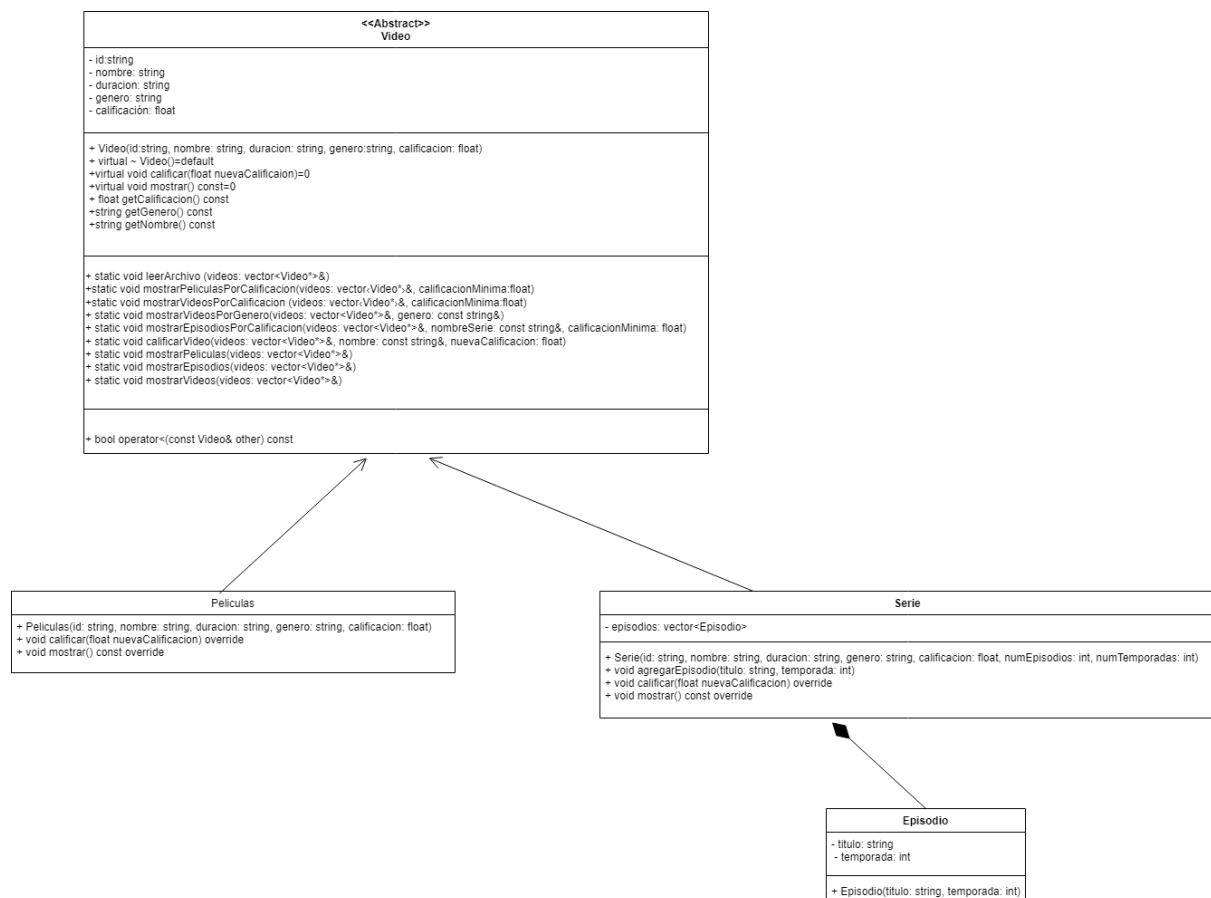
Índice

Índice.....	2
Introducción.....	3
Argumentación del Diseño y UML.....	3
Clase Abstracta Video:.....	4
Herencia:.....	4
Polimorfismo:.....	4
Clases Abstractas y Métodos Abstractos:.....	5
Modificadores de Acceso:.....	5
Sobrecarga y Sobreescritura de Métodos:.....	5
Sobrecarga de Operadores:.....	5
Excepciones:.....	5
Capturas de las corridas del código.....	6
Identificación de Casos Críticos.....	7
Conclusiones Personales.....	7
Referencias.....	7

Introducción.

En los últimos años, los servicios de streaming de video bajo demanda han proliferado con plataformas como Netflix, Disney+, y DC Universe. Estos servicios ofrecen una variedad de videos, desde películas hasta series, cada uno con diferentes características. Este proyecto se enfoca en crear una versión limitada de un sistema de streaming que gestione dos tipos de videos: películas y series, cada una con un ID, nombre, duración y género. Las series tendrán episodios con títulos y temporadas, y se evaluarán por calificaciones en una escala de 1 a 5.

Argumentación del Diseño y UML.



El diseño del diagrama de clases se centra en la abstracción y la reutilización de código. A continuación, se argumentan las decisiones tomadas:

Clase Abstracta Video:

Se utiliza una clase abstracta **Video** para definir los atributos y métodos comunes a todas las clases derivadas, como **Películas** y **Serie**. Esto permite la reutilización de código y garantiza que todas las clases derivadas implementen los métodos esenciales definidos en **Video**.

Herencia:

Peliculas y Serie heredan de Video, lo que refleja una relación "es-un" entre estas clases. Esto permite que Peliculas y Serie compartan atributos y métodos comunes definidos en Video, evitando la duplicación de código.

Polimorfismo:

La utilización de métodos virtuales y sobrescritos en las clases derivadas permite el polimorfismo. Los métodos como calificar y mostrar son definidos en Video y pueden ser sobrescritos en Peliculas y Serie para proporcionar implementaciones específicas.

Clases Abstractas y Métodos Abstractos:

La clase Video contiene métodos abstractos que deben ser implementados por las clases derivadas. Esto garantiza que todas las clases derivadas tengan una implementación concreta de estos métodos.

Modificadores de Acceso:

Los modificadores de acceso (public, protected, private) se utilizan para controlar el acceso a los atributos y métodos de las clases. Esto asegura el encapsulamiento y la protección de los datos internos de las clases.

Sobrecarga y Sobreescritura de Métodos:

Se implementan métodos sobrecargados y sobrescritos para proporcionar diferentes formas de ejecutar una operación similar. Por ejemplo, calificar puede tener diferentes parámetros o comportamientos en Peliculas y Serie.

Sobrecarga de Operadores:

Se sobrecarga el operador `+ bool operator<(const Video& other) const` para comparar objetos de Video directamente, lo cual puede ser útil para ordenar o clasificar videos según su calificación.

Excepciones:

Se usan las excepciones para que al leer y convertir datos del archivo .csv el programa pueda gestionar errores inesperados durante la ejecución:

```
} catch (const invalid_argument& e) {  
    cerr << "Error al convertir los datos: " << e.what() << endl; //Ma  
} catch (const out_of_range& e) {  
    cerr << "Valor fuera de rango: " << e.what() << endl;  
}
```

Se optó por esta solución porque permite una organización clara y eficiente del código a través de la reutilización y la extensión de clases. El uso de una clase abstracta como `Video` asegura que todos los tipos de videos (películas y series) compartan atributos y comportamientos comunes, lo que reduce la duplicación de código y facilita el mantenimiento. La herencia y el polimorfismo permiten manejar objetos de diferentes tipos de manera uniforme, aumentando la flexibilidad y extensibilidad del sistema. Además, los modificadores de acceso aseguran el encapsulamiento y la protección de datos, manteniendo la integridad y seguridad del sistema. Estas decisiones, en conjunto, minimizan la complejidad y mejoran la claridad del código.

Capturas de las corridas del código.

```
Menu:
1. Cargar archivo de datos
2. Mostrar los videos
3. Mostrar las peliculas
4. Mostrar los episodios
5. Mostrar los videos con una cierta calificacion o de un cierto genero
6. Mostrar los episodios de una determinada serie con una calificacion determinada
7. Mostrar las peliculas con cierta calificacion
8. Calificar un video
0. Salir
Opcion: 8
Ingrese el nombre del video a calificar: The Office
Ingrese la nueva calificacion: 9
La calificacion nueva del video es: 9
```

```
Menu:
1. Cargar archivo de datos
2. Mostrar los videos
3. Mostrar las peliculas
4. Mostrar los episodios
5. Mostrar los videos con una cierta calificacion o de un cierto genero
6. Mostrar los episodios de una determinada serie con una calificacion determinada
7. Mostrar las peliculas con cierta calificacion
8. Calificar un video
0. Salir
Opcion: 3
Pelicula: The Dark Knight, Duracion: 152, Genero: Accion, Calificacion: 9
Pelicula: Inception, Duracion: 148, Genero: Ciencia Ficcion, Calificacion: 8.8
Pelicula: The Godfather, Duracion: 175, Genero: Crimen, Calificacion: 9.2
Pelicula: The Shawshank Redemption, Duracion: 142, Genero: Drama, Calificacion: 9.3
Pelicula: Pulp Fiction, Duracion: 154, Genero: Crimen, Calificacion: 8.9
Pelicula: The Matrix, Duracion: 136, Genero: Ciencia Ficcion, Calificacion: 8.7
Pelicula: Interstellar, Duracion: 169, Genero: Ciencia Ficcion, Calificacion: 8.6
Pelicula: The Lion King, Duracion: 88, Genero: Animacion, Calificacion: 8.5
Pelicula: Forrest Gump, Duracion: 142, Genero: Drama, Calificacion: 8.8
Pelicula: Fight Club, Duracion: 139, Genero: Drama, Calificacion: 8.8
```

```

Menu:
1. Cargar archivo de datos
2. Mostrar los videos
3. Mostrar las peliculas
4. Mostrar los episodios
5. Mostrar los videos con una cierta calificacion o de un cierto genero
6. Mostrar los episodios de una determinada serie con una calificacion determinada
7. Mostrar las peliculas con cierta calificacion
8. Calificar un video
0. Salir
Opcion: 7
Ingrese la calificación mínima: 9
Pelicula: The Dark Knight, Duracion: 152, Genero: Accion, Calificacion: 9
Pelicula: The Godfather, Duracion: 175, Genero: Crimen, Calificacion: 9.2
Pelicula: The Shawshank Redemption, Duracion: 142, Genero: Drama, Calificacion: 9.3

```

```

Menu:
1. Cargar archivo de datos
2. Mostrar los videos
3. Mostrar las peliculas
4. Mostrar los episodios
5. Mostrar los videos con una cierta calificacion o de un cierto genero
6. Mostrar los episodios de una determinada serie con una calificacion determinada
7. Mostrar las peliculas con cierta calificacion
8. Calificar un video
0. Salir
Opcion: 6
Ingrese el nombre de la serie: The Office
Ingrese la calificación mínima: 8
Serie: The Office, Duracion: 22, Genero: Comedia, Calificacion: 8.8, Episodios: 205
    Episodio: Episodio 1, Temporada: 2
    Episodio: Episodio 2, Temporada: 9
    Episodio: Episodio 3, Temporada: 9
    Episodio: Episodio 4, Temporada: 7
    Episodio: Episodio 5, Temporada: 9
    Episodio: Episodio 6, Temporada: 7
    Episodio: Episodio 7, Temporada: 2

```

Identificación de Casos Críticos

-Lectura de Archivo:

Si el archivo de videos no se encuentra o está corrupto, la función leerArchivo fallará.

-Calificaciones Inválidas:

Si se intentan asignar calificaciones fuera del rango permitido, esto podría causar errores.

-Acceso a Episodios:

Intentar acceder a episodios en una serie que no existen causaría errores de índice.

Conclusiones Personales.

Karen.

En este proyecto, realmente quise asegurarme de que aplicamos a fondo los principios de la programación orientada a objetos. Utilicé la herencia, el polimorfismo y el encapsulamiento para construir un sistema que no solo es modular y extensible, sino también fácil de entender

y mantener. La estructura de clases que diseñe, junto con los métodos abstractos y sobrescritos, hacen que el código sea reutilizable y fácil de trabajar. Además, el manejo adecuado de excepciones y la sobrecarga de operadores no son solo detalles técnicos; realmente fortalecen y flexibilizan el sistema. Estoy convencida de que este enfoque proporciona una base sólida para la gestión de videos y simplifica cualquier futura ampliación o mantenimiento.

Estrella

Este proyecto es un ejemplo de la aplicación de principios fundamentales de programación orientada a objetos (POO), como herencia, polimorfismo y encapsulamiento. Me ayudó a tener un entendimiento completamente práctico de estas técnicas dentro de un sistema sencillo que se mantiene fácil de entender y mantener, reutilizable, flexible, sólido y confiable. En general, el enfoque de POO que adopte en este proyecto proporciona una base sólida para el entendimiento de importantes conceptos de la programación orientada a objetos, con un proyecto fácil de entender y que me impulsó a aplicar los conocimientos adquiridos durante el curso.

Referencias.

GeeksforGeeks. (2014, July 15). Pure Virtual Functions and Abstract Classes in C. GeeksforGeeks; GeeksforGeeks. <https://www.geeksforgeeks.org/pure-virtual-functions-and-abstract-classes/?ref=gcse>

GeeksforGeeks. (2017, May 25). Inheritance in C. GeeksforGeeks; GeeksforGeeks. https://www.geeksforgeeks.org/inheritance-in-c/?ref=header_search

GeeksforGeeks. (2017, May 22). C Polymorphism. GeeksforGeeks; GeeksforGeeks. https://www.geeksforgeeks.org/cpp-polymorphism/?ref=header_search

GeeksforGeeks. (2014, May 28). Function Overloading in C. GeeksforGeeks; GeeksforGeeks. https://www.geeksforgeeks.org/function-overloading-c/?ref=header_search

GeeksforGeeks. (2013, December 23). Exception Handling in C. GeeksforGeeks; GeeksforGeeks. https://www.geeksforgeeks.org/exception-handling-c/?ref=header_search