



Evidencia 02

Programación orientada a objetos (Gpo 308)

Presentan:

Ricardo Chapa Perez	A01722755
Greta Paola Guagnelli González	A01722531
Marcelo Cárdenas Leal	A00836049

\

Docente asesor:

Maria Teresa Garza Garza

Monterrey, N.L.

18 junio 2023

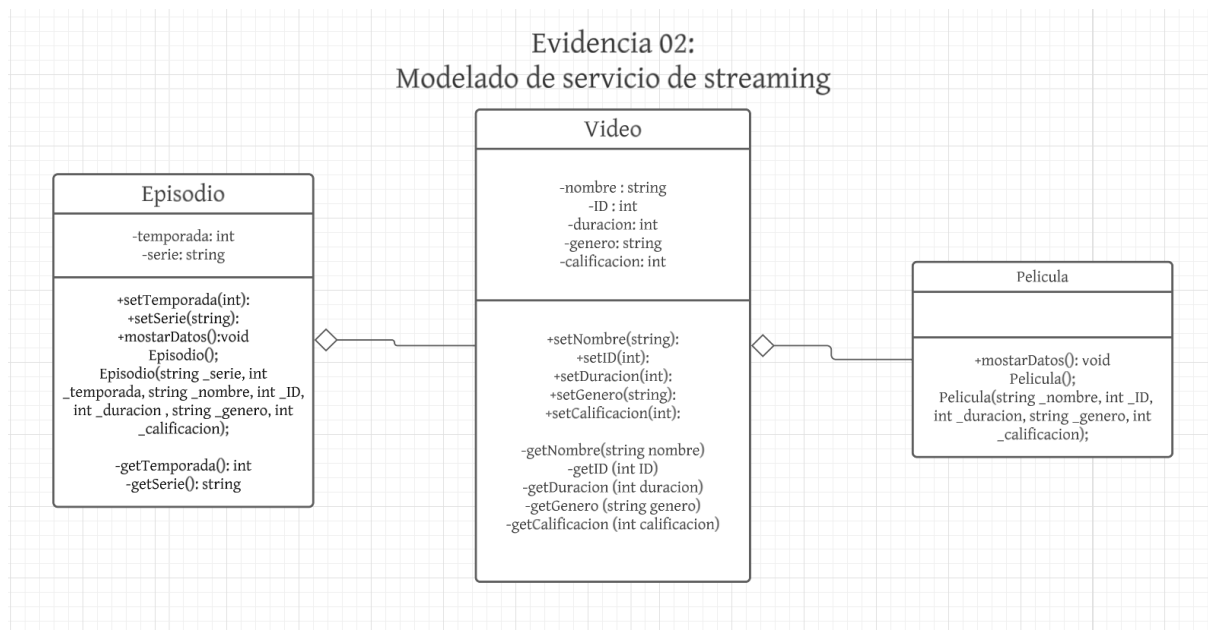
Índice:

Introducción.....	2
Diagrama de clases.....	2
Explicación de diagrama de clases.....	2
Explicación código.....	5
Ejemplo de ejecución.....	6
Conclusiones.....	12

Introducción:

Utilizando los conceptos de la programación orientada a objetos, podemos construir un sistema de clases que represente correctamente la situación planteada y facilite la gestión de vídeos, películas y series al igual que cumpla con la información sobre los géneros y calificaciones de estas. Esto nos permitirá organizar y estructurar nuestro código de manera eficiente para facilitar la manipulación de los datos.

Diagrama de clases UML:



El UML proporcionado describe tres clases: "Episodio", "Video" y "Pelicula", junto con sus respectivos atributos y métodos.

La clase "Episodio" tiene dos atributos privados: "temporada" de tipo entero (int) y "serie" de tipo cadena de texto (string). También tiene dos métodos públicos para establecer estos atributos: "setTemporada" que recibe un valor entero y "setSerie" que recibe una cadena de texto. Además, tiene un método público "mostrarDatos" que no devuelve ningún valor (void), que podría utilizarse para imprimir los datos del episodio.

La clase "Episodio" también tiene dos constructores: uno sin parámetros y otro que toma como argumentos una cadena de texto para la serie, un entero para la temporada, una cadena de texto para el nombre, un entero para el ID, otro entero para la duración, una cadena de texto para el género y un entero para la calificación.

La clase "Video" representa un video genérico y tiene cinco atributos privados: "nombre" de tipo cadena de texto, "ID" de tipo entero, "duracion" de tipo entero, "genero" de tipo cadena de texto y "calificacion" de tipo entero. Además, tiene cinco métodos públicos para establecer estos atributos: "setNombre" que recibe una cadena de texto, "setID" que recibe un valor entero, "setDuracion" que recibe un valor entero, "setGenero" que recibe una cadena de texto y "setCalificacion" que recibe un valor entero. También tiene cinco métodos públicos de obtención (get) para cada uno de los atributos mencionados anteriormente.

La clase "Película" es una subclase de la clase "Video" y no tiene atributos adicionales. Hereda todos los atributos y métodos de la clase "Video". Además, la clase "Película" tiene dos constructores: uno sin parámetros y otro que toma como argumentos una cadena de texto para el nombre, un entero para el ID, otro entero para la duración, una cadena de texto para el género y un entero para la calificación. La clase "Película" también tiene un método público "mostrarDatos" que no devuelve ningún valor (void), que podría utilizarse para imprimir los datos de la película.

Esta implementación de código demuestra un diseño coherente y eficiente al utilizar principios clave de programación orientada a objetos. El uso de encapsulación garantiza que los datos estén protegidos y solo se acceda a través de métodos públicos, lo que promueve una mejor organización y modularidad. La herencia se utiliza adecuadamente al establecer una jerarquía entre las clases "Video" y "Película", lo que facilita la reutilización de código y promueve la coherencia en la estructura del programa. Los constructores proporcionan una forma clara y flexible de inicializar los objetos con valores predefinidos o personalizados. En general, esta implementación se destaca por su claridad, legibilidad y mantenibilidad, lo que facilita la comprensión y el desarrollo de nuevas funcionalidades en el futuro.

Explicación código:

En el código se identifican las siguientes clases: Video, Película, Episodio. Cada una de estas clases representan entidades clave dentro del sistema.

Se utiliza el concepto de herencia en las clases Película y Episodio, que heredan de la clase base Video. Esto permite compartir atributos y comportamientos comunes, y al mismo tiempo, extender y especializar dichas clases.

Los modificadores de acceso se utilizan adecuadamente en las clases y sus miembros. Los atributos y métodos relevantes son declarados con los modificadores public, protected y private según corresponda, para controlar la visibilidad y el acceso a los mismos.

Se utiliza la sobreescritura de métodos en las clases Video, Película y Episodio. Esto permite definir comportamientos específicos para cada clase y operar con ellas de manera adecuada.

El polimorfismo se emplea al utilizar punteros de la clase base Video para referenciar objetos de las clases derivadas Película y Episodio. Esto permite tratar a los objetos de diferentes clases de forma uniforme y realizar operaciones polimórficas sobre ellos.

En tu código no se utiliza una clase abstracta en sí, pero la clase Video se acerca a ser una clase abstracta, ya que contiene un método virtual puro (mostrarDatos()) que debe ser implementado por las clases derivadas. Esto permite definir un comportamiento básico común para todos los videos y requerir su implementación en las clases hijas.

Ejemplos de ejecución:

caso 1

```
1) Cargar Archivo de Datos
2) Mostrar la lista de Peliculas
3) Mostrar la lista de Episodios
4) Mostar los videos con cierto genero
5) Mostrar los episodios de una determinada serie
6) Mostrar las peliculas con cierta calificacion
7) Calificar un video
8) Salir
Seleccione una opcion: 1

Nombre: The Super Mario Bros. Movie
ID: 1001
Duración: 92
Genero: Comedia
Calificación: 9

Nombre: Black Widow
ID: 1002
Duración: 133
Genero: Acción
Calificación: 8

Nombre: Cruella
ID: 1003
Duración: 134
Genero: Comedia
Calificación: 8

Nombre: A Quiet Place Part II
ID: 1004
Duración: 97
Genero: Terror
Calificación: 8
```

caso 2

```
Menu de opciones

1) Cargar Archivo de Datos
2) Mostrar la lista de Peliculas
3) Mostrar la lista de Episodios
4) Mostar los videos con cierto genero
5) Mostrar los episodios de una determinada serie
6) Mostrar las peliculas con cierta calificacion
7) Calificar un video
8) Salir
Seleccione una opcion: 2

Nombre: The Super Mario Bros. Movie
ID: 1001
Duración: 92
Genero: Comedia
Calificación: 9

Nombre: Black Widow
ID: 1002
Duración: 133
Genero: Acción
Calificación: 8

Nombre: Cruella
ID: 1003
Duración: 134
Genero: Comedia
Calificación: 8

Nombre: A Quiet Place Part II
ID: 1004
Duración: 97
Genero: Terror
Calificación: 8
```

caso 3

```

Menu de opciones

1) Cargar Archivo de Datos
2) Mostrar la lista de Peliculas
3) Mostrar la lista de Episodios
4) Mostrar los videos con cierto genero
5) Mostrar los episodios de una determinada serie
6) Mostrar las peliculas con cierta calificacion
7) Calificar un video
8) Salir
Seleccione una opcion: 3

Serie: The Office
Temporada: 1
Nombre: Piloto
ID: 2001
Duración: 42
Genero: Accion
Calificación: 9

Serie: The Office
Temporada: 1
Nombre: El dia de la diversidad
ID: 2002
Duración: 37
Genero: Comedia
Calificación: 3

Serie: The Office
Temporada: 1
Nombre: Me multaron
ID: 2003
Duración: 45
Genero: Comedia
Calificación: 3

Serie: The Office
Temporada: 1
Nombre: Me multaron
ID: 2004

```

caso 4

```

Menu de opciones

1) Cargar Archivo de Datos
2) Mostrar la lista de Peliculas
3) Mostrar la lista de Episodios
4) Mostrar los videos con cierto genero
5) Mostrar los episodios de una determinada serie
6) Mostrar las peliculas con cierta calificacion
7) Calificar un video
8) Salir
Seleccione una opcion: 4
Seleccione el género:
1) Comedia
2) Animación
3) Drama
4) Acción
5) Terror
3
Videos de género Drama:

Nombre: Nomadland
ID: 1014
Duración: 108
Genero: Drama
Calificación: 7

Nombre: The Father
ID: 1015
Duración: 97
Genero: Drama
Calificación: 8

Nombre: Minari
ID: 1017
Duración: 115
Genero: Drama
Calificación: 7

```

caso 5

```

Menu de opciones
1) Cargar Archivo de Datos
2) Mostrar la lista de Peliculas
3) Mostrar la lista de Episodios
4) Mostrar los videos con cierto genero
5) Mostrar los episodios de una determinada serie
6) Mostrar las peliculas con cierta calificacion
7) Calificar un video
8) Salir
Seleccione una opcion: 5
Seleccione la serie:
1) The Office
2) Suits
2
Episodios de la serie Suits:

Serie: Suits
Temporada: 1
Nombre: Pilot
ID: 2010
Duración: 30
Genero: Drama
Calificación: 6

Serie: Suits
Temporada: 1
Nombre: Errors and Omissions
ID: 2011
Duración: 42
Genero: Drama
Calificación: 7

Serie: Suits
Temporada: 1
Nombre: Inside Track
ID: 2012
Duración: 45
Genero: Drama

```

caso 6

```

Menu de opciones
1) Cargar Archivo de Datos
2) Mostrar la lista de Peliculas
3) Mostrar la lista de Episodios
4) Mostrar los videos con cierto genero
5) Mostrar los episodios de una determinada serie
6) Mostrar las peliculas con cierta calificacion
7) Calificar un video
8) Salir
Seleccione una opcion: 6
Seleccione la calificación:
a) 1
b) 2
c) 3
d) 4
e) 5
f) 6
g) 7
h) 8
i) 9
j) 10
9
Peliculas con calificación 9:

Nombre: The Super Mario Bros. Movie
ID: 1001
Duración: 92
Genero: Comedia
Calificación: 9
Menu de opciones
1) Cargar Archivo de Datos
2) Mostrar la lista de Peliculas
3) Mostrar la lista de Episodios
4) Mostrar los videos con cierto genero
5) Mostrar los episodios de una determinada serie
6) Mostrar las peliculas con cierta calificacion
7) Calificar un video
8) Salir
Seleccione una opcion: █

```

caso 7

```

Seleccione una opcion: 7
Su video es una pelicula o un episodio de una serie? (p/e): 2
Ingrese el nombre del video: 3
Ingrese la duracion del video: 4
Ingrese el genero del video: 1
Ingrese la calificacion: 2

Video agregado exitosamente

```


Identificación de casos que harían que el proyecto deje de funcionar

Entrada incorrecta del usuario: Cuando a el usuario se le pide que ingrese un número del 1 al 8 en nuestro menú si se le llega a poner una Letra o un símbolo nuestro código dejará de funcionar. De igual manera si se le pide ingresar una palabra e ingresa un número tendrá un error.

Al mencionar estos errores buscamos que el usuario tenga la experiencia más limpia, segura y confiable. Esto para ayudar a prevenir cualquier error que pueda tener el usuario y así que se sienta más seguro y tenga una mejor experiencia.

Conclusiones personales:

Ricardo: Durante las cinco semanas de clase de C ++, he adquirido un profundo conocimiento sobre los conceptos de herencia y polimorfismo. Estos temas son fundamentales en la programación orientada a objetos y me han proporcionado una base sólida para el desarrollo de aplicaciones más eficientes y flexibles. El trabajo final, realizado en equipo, me permitió aplicar estos conceptos en un proyecto práctico y trabajar en colaboración con mis compañeros. Esta experiencia ha mejorado mis habilidades de comunicación y resolución de problemas, y estoy seguro de que los conceptos aprendidos serán de gran utilidad en mi trayectoria profesional como programador.

Greta: Durante este proyecto, he aprendido la importancia de una estructura de clases y la jerarquía de herencia para representar de manera correcta los diferentes tipos de vídeos, películas, series y episodios. Este proyecto me ha proporcionado una muy buena base para abordar proyectos similares en el futuro de mi carrera.

Marcelo: A lo largo de este proyecto, he desarrollado un conocimiento profundo sobre la importancia de utilizar una estructura de clases y una jerarquía de herencia correcta para representar los diversos tipos de contenido como videos, películas, series y episodios. Mediante el uso de clases, pude definir atributos y comportamientos específicos para cada tipo de vídeo, lo que facilitó la manipulación y gestión de la información relacionada.