



Situación Problema: Modelado de Servicio de Streaming

Brenda Elena Saucedo González – A00829855

Índice de Contenido

INTRODUCCIÓN.....	2
DIAGRAMA DE CLASES UML.....	3
EJEMPLO DE EJECUCIÓN.....	6
ARGUMENTACIÓN DE LAS PARTES DEL PROYECTO.....	11
IDENTIFICACIÓN DE CASOS QUE HARÍAN QUE EL PROYECTO DEJE DE FUNCIONAR.....	14
CONCLUSIÓN PERSONAL.....	15
REFERENCIAS CONSULTADAS.....	16

Introducción

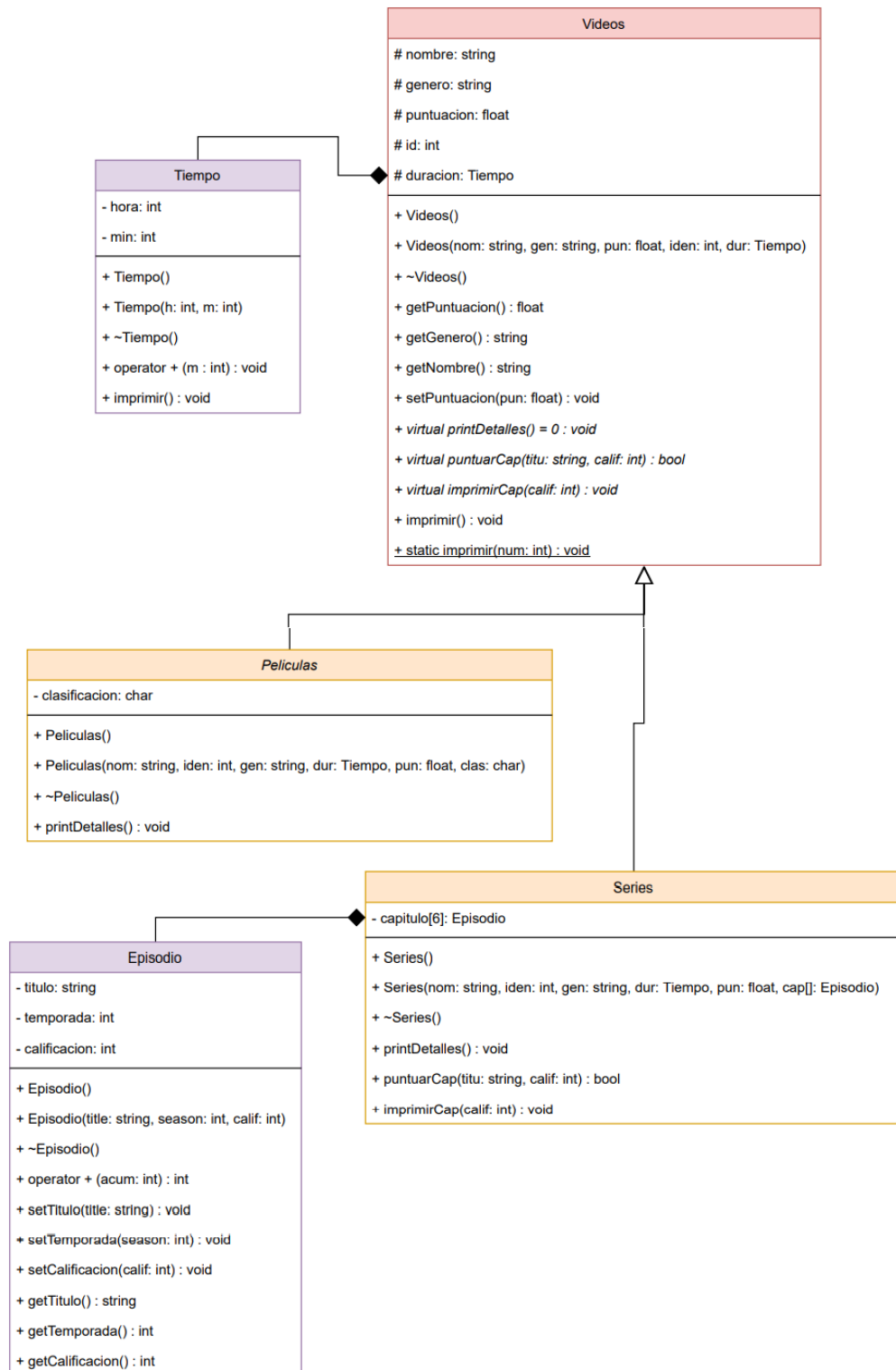
El Streaming es una tecnología que nos permite ver un archivo de audio o video directamente desde una página de Internet o una aplicación móvil, sin descargarlo completamente en nuestro dispositivo. Es decir, lo visualizamos a medida que se va descargando a la smart TV, computadora, tablet o teléfono.

Una de las alternativas más populares en los últimos años para ver televisión en México, es a través de plataformas de streaming. Las más utilizadas en la República Mexicana son Netflix, seguida de Claro Video, YouTube, Amazon Prime, HBO Go, entre muchas otras.

Últimamente, los servicios de streaming de video han cobrado mucha popularidad y demanda en el mercado, por ejemplo, grandes industrias como Netflix, Disney, DC entre otros, trabajan con este tipo de servicio. 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.

Analizando esta situación, se nos planteó realizar una versión limitada para apoyar a un futuro proveedor de este tipo de servicios, en donde mostraremos los dos tipos de streaming de videos que existen, los cuales son películas y series. En dicho proyecto se desplegarán los videos con sus respectivos títulos, tipo de género, la calificación, el ID y la duración del video. También se desplegará la información a más detalle dependiendo del tipo de video que se esté buscando, esto para facilitar al usuario su búsqueda de mejores videos dependiendo su calificación o género, e incluso llegar a modificar dicha calificación. En el presente proyecto se mostrará un simulador limitado y básico de servicio de streaming.

Diagrama de Clases UML



El diagrama se decidió diseñar de la presente manera, ya que se sabía que los videos constaban de dos tipos, películas y series, las cuales compartían algunas características en común, como lo son el nombre, género, la calificación (puntuación), la duración y el ID del video. Se hizo un recuadro para la clase base, que es la clase Videos y dos recuadros para las clases derivadas, las cuales son las clases Películas y Series, y para hacer referencia que son clases derivadas se usó su respectiva flecha que los une con la clase base.

En la clase Videos, se escribieron los atributos que comparten las películas y series, y en los métodos se escribieron los constructores, el destructor, los métodos de acceso y modificación. Fue necesario agregar otros métodos para que se desplegara la información solicitada u obtener los resultados necesarios para que el programa pudiera continuar y hallar los resultados esperados.

Estos otros métodos declarados, son para desplegar la información de los videos, uno de manera general, y el otro de manera más detallada, el cual se programó para ser un virtual puro, ya que este no contiene los detalles de las películas y series. También se programaron dos métodos virtuales, uno para calificar los capítulos de las series, y otro para desplegar los capítulos con una calificación determinada de una serie determinada, en las cuales solo la clase Series puede acceder a los capítulos.

También se declaró un método por sobrecarga parametrizada, en los cuales el original es imprimir() y el de sobrecarga parametrizada es un método estático imprimir(num: int) que lleva como parámetro un dato de tipo entero para poder hacer la distinción.

Luego, se hizo un nuevo recuadro para la clase Tiempo, ya que la duración de un video consta de horas y minutos, por lo que se optó por crear una clase Tiempo que guardara estos atributos y que la duración del video fuera del tipo de esta clase. Para el diseño, se usó su respectiva línea con un rombo para hacer referencia a la composición que se está haciendo, ya que la clase Video tiene un atributo de tipo Tiempo.

En esta clase, se escribieron los métodos que se pensó que iban a ser útiles, así como los demás métodos que se fueron declarando en el programa, como lo es la función de sobrecarga de operadores, la cual se agregó al programa y al diseño porque la información que se recibe por parte del archivo de texto es en minutos, por lo que se optó que sería una manera más sencilla sumar los minutos a una variable tipo Tiempo para que este guardara los valores. También se agregó el método imprimir para que desplegara los datos guardados en sus atributos, es decir las horas y minutos, esto para ahorrar código cuando se necesite saber la duración del video.

En cuanto a la clase derivada Películas, esta tiene una característica que la clase Series no tiene, como lo es la clasificación de la película, por lo que se agregó este

atributo al diseño de la clase Películas. Por otra parte, en esta misma clase, aparte de los métodos constructores y destructor, se definió el método printDetalles(), para desplegar los detalles de la película (nombre, genero, duración, id, puntuación y clasificación).

En cuanto a la clase derivada Series, esta tiene una característica la cual no comparte con las películas, característica la cual son capítulos. Esta característica se decidió hacer de ella un arreglo, ya que una serie posee varios capítulos, pero también se decidió que fuera de tipo Episodio, ya que todo capítulo contiene información distinta, como lo son su título, temporada y calificación, por lo que se procedió a realizar un recuadro para la clase Episodio, que contuviera estas características (título, temporada y calificación), los cuales fueron agregados al diagrama de clases y se usó su respectiva línea con un rombo para hacer referencia a la composición que se está haciendo, ya que la clase Series tiene un atributo de tipo Episodio.

En la clase Episodio también se declaró una función de sobrecarga de operadores para facilitar el cálculo del promedio de la calificación de la serie, razón por la cual esta función igualmente fue agregada al diseño, así como también sus debidos métodos de acceso y modificación, constructores y destructor.

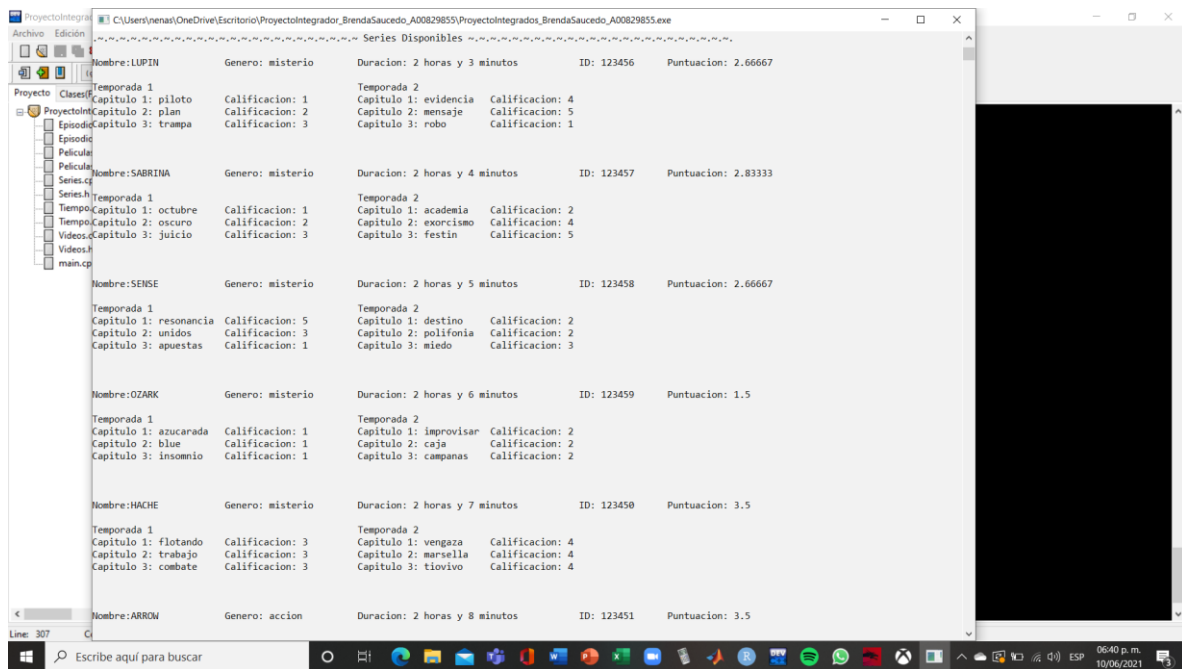
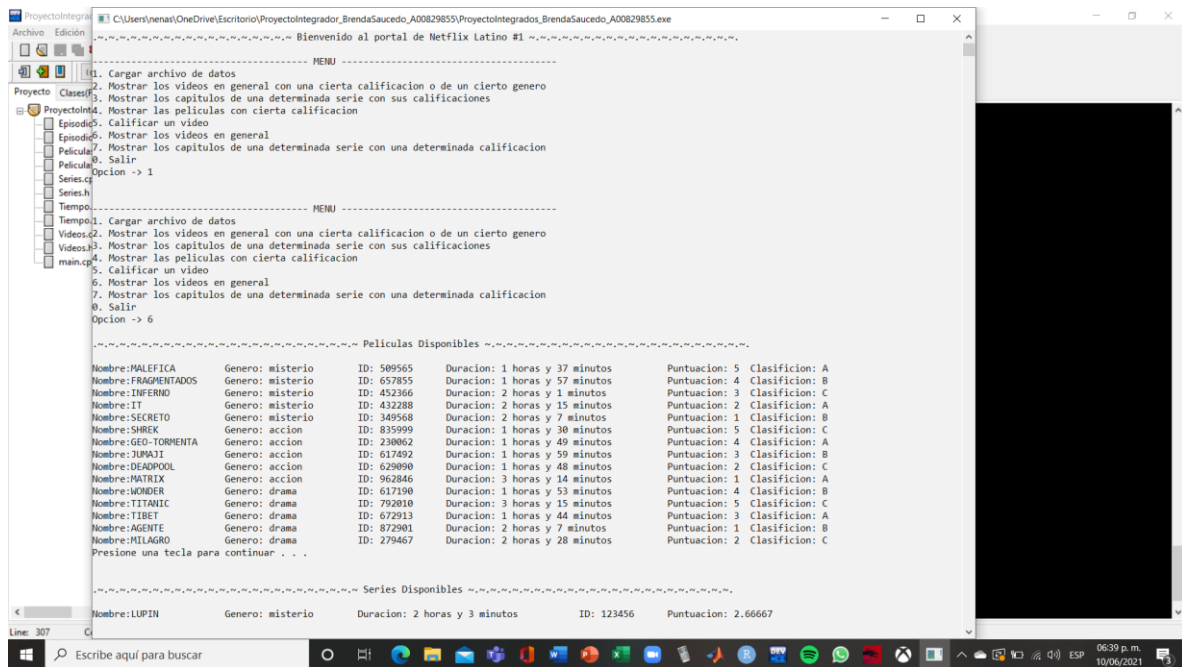
En la clase Series también se tuvo que agregar al diseño los métodos para puntuar un capítulo e imprimir determinados capítulos de una determinada serie. Estos métodos no fueron utilizados en otras clases, ya que la clase Series es la única que maneja capítulos, por lo que en el diseño solo se agregaron al recuadro de la clase Series.

En la clase Series también se definió el método printDetalles(), para desplegar los detalles de la serie (nombre, genero, duración, id, puntuación, el título de los capítulos, su temporada y calificación).

En cada una de las clases, así como a sus debidos recuadros en el diseño de clases UML, se agregaron sus respectivos métodos constructores y destructor, y en algunos se agregaron sus métodos de acceso y modificación, ya que no en todas las clases fue necesario el uso de este tipo de métodos.

Con esta explicación, fue la razón por la que se agregaron estas clases, métodos y atributos al diseño, ya que el diagrama de clases UML es la base para empezar a construir el proyecto y conforme se hacen avances, estas modificaciones se agregan al diseño, ya que el diagrama de clases es una representación visual de lo que se esta programando o lo que se quiere programar.

Ejemplo de Ejecución



ProjectoIntegrador.exe

Archivo Edición

Nombre: ARROW Genero: accion Duracion: 2 horas y 8 minutos ID: 123451 Puntuacion: 3.5

Temporada 1
 Capitulo 1: inicio Calificacion: 5
 Capitulo 2: honra Calificacion: 5
 Capitulo 3: solitarios Calificacion: 5

Temporada 2
 Capitulo 1: inocente Calificacion: 2
 Capitulo 2: mercancia Calificacion: 2
 Capitulo 3: legado Calificacion: 2

Nombre: OUTLANDER Genero: accion Duracion: 2 horas y 9 minutos ID: 123452 Puntuacion: 1.83333

Temporada 1
 Capitulo 1: escocia Calificacion: 2
 Capitulo 2: castillo Calificacion: 2
 Capitulo 3: salida Calificacion: 1

Temporada 2
 Capitulo 1: encuentro Calificacion: 2
 Capitulo 2: renta Calificacion: 2
 Capitulo 3: comandante Calificacion: 2

Nombre: MERLIN Genero: accion Duracion: 2 horas y 10 minutos ID: 123453 Puntuacion: 3.33333

Temporada 1
 Capitulo 1: llamado Calificacion: 3
 Capitulo 2: valiant Calificacion: 3
 Capitulo 3: envenenar Calificacion: 3

Temporada 2
 Capitulo 1: lancelot Calificacion: 2
 Capitulo 2: remedio Calificacion: 4
 Capitulo 3: avalon Calificacion: 5

Nombre: VIKINGOS Genero: accion Duracion: 2 horas y 11 minutos ID: 123454 Puntuacion: 2.33333

Temporada 1
 Capitulo 1: iniciacion Calificacion: 2
 Capitulo 2: furia Calificacion: 1
 Capitulo 3: desposieido Calificacion: 5

Temporada 2
 Capitulo 1: juicio Calificacion: 1
 Capitulo 2: asalto Calificacion: 2
 Capitulo 3: entierro Calificacion: 3

Nombre: FLASH Genero: accion Duracion: 2 horas y 12 minutos ID: 123455 Puntuacion: 3

Temporada 1
 Capitulo 1: comienzo Calificacion: 4
 Capitulo 2: veloz Calificacion: 2
 Capitulo 3: escape Calificacion: 5

Temporada 2
 Capitulo 1: renegados Calificacion: 1
 Capitulo 2: plastico Calificacion: 1
 Capitulo 3: nacimiento Calificacion: 5

Nombre: PALACIO Genero: drama Duracion: 2 horas y 13 minutos ID: 123460 Puntuacion: 3

Temporada 1

Temporada 2

Line: 307

Escribe aquí para buscar

06:40 p.m. 10/06/2021

AutoSave ProjectoIntegrador.exe

File Home

Nombre: PALACIO Genero: drama Duracion: 2 horas y 13 minutos ID: 123460 Puntuacion: 3

Temporada 1
 Capitulo 1: amigo Calificacion: 3
 Capitulo 2: adios Calificacion: 3
 Capitulo 3: rico Calificacion: 2

Temporada 2
 Capitulo 1: ascenso Calificacion: 1
 Capitulo 2: competencia Calificacion: 4
 Capitulo 3: secreto Calificacion: 5

Nombre: BRIDGERTON Genero: drama Duracion: 2 horas y 14 minutos ID: 123461 Puntuacion: 2.83333

Temporada 1
 Capitulo 1: diamante Calificacion: 5
 Capitulo 2: sorprende Calificacion: 2
 Capitulo 3: desmayo Calificacion: 1

Temporada 2
 Capitulo 1: honor Calificacion: 3
 Capitulo 2: duque Calificacion: 4
 Capitulo 3: frufu Calificacion: 2

Nombre: YOU Genero: drama Duracion: 2 horas y 15 minutos ID: 123462 Puntuacion: 2.33333

Temporada 1
 Capitulo 1: piloto Calificacion: 3
 Capitulo 2: agradable Calificacion: 3
 Capitulo 3: capitan Calificacion: 1

Temporada 2
 Capitulo 1: enemigo Calificacion: 2
 Capitulo 2: totalidad Calificacion: 4
 Capitulo 3: tuyo Calificacion: 1

Nombre: DARK Genero: drama Duracion: 2 horas y 16 minutos ID: 123463 Puntuacion: 4

Temporada 1
 Capitulo 1: secretos Calificacion: 5
 Capitulo 2: mentiras Calificacion: 3
 Capitulo 3: tiempo Calificacion: 4

Temporada 2
 Capitulo 1: dobles Calificacion: 4
 Capitulo 2: verdades Calificacion: 3
 Capitulo 3: despertar Calificacion: 5

Nombre: RAIN Genero: drama Duracion: 2 horas y 17 minutos ID: 123464 Puntuacion: 3.33333

Temporada 1
 Capitulo 1: contacto Calificacion: 3
 Capitulo 2: verdad Calificacion: 4
 Capitulo 3: control Calificacion: 3

Temporada 2
 Capitulo 1: salvense Calificacion: 3
 Capitulo 2: organcense Calificacion: 4
 Capitulo 3: sobrevivir Calificacion: 3

MENU

1. Cargar archivo de datos

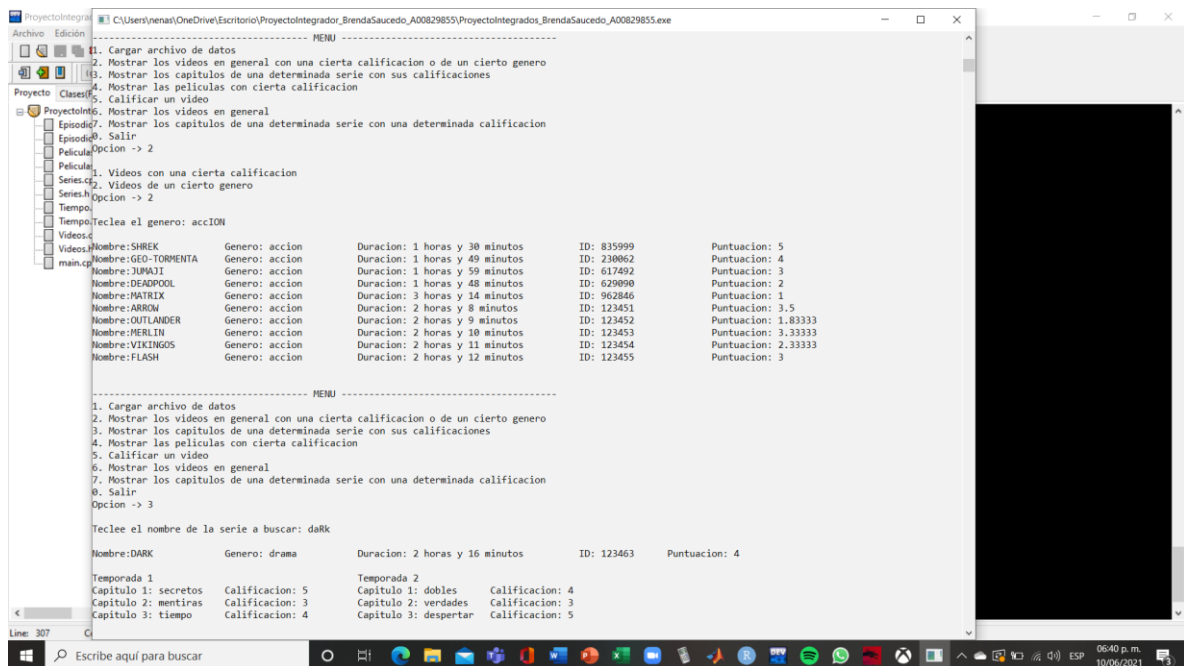
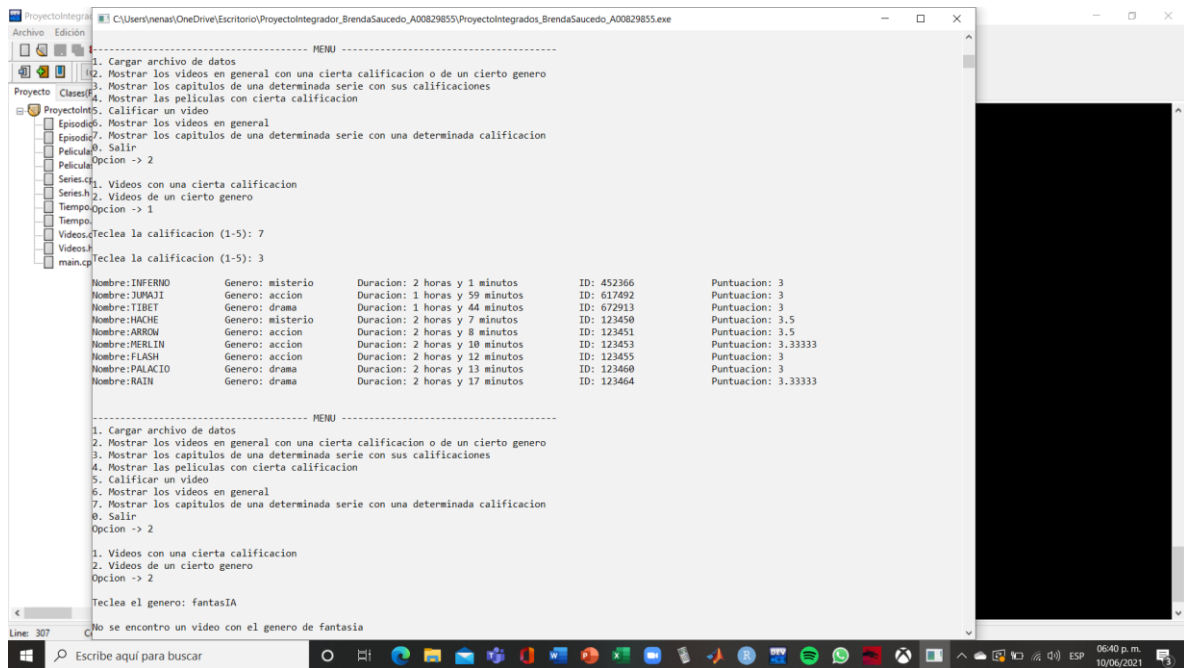
2. Mostrar los videos en general con una cierta calificacion o de un cierto genero

3. Mostrar los capitulos de una determinada serie con sus calificaciones

Page 7 of 13 256

Escribe aquí para buscar

06:40 p.m. 10/06/2021



```
ProjectoIntegrador
C:\Users\nenas\OneDrive\Escritorio\ProjectoIntegrador_BrendaSaucedo_A00829855\ProjectoIntegrador_BrendaSaucedo_A00829855.exe

Archivo Edición
----- MENU -----
1. Cargar archivo de datos
2. Mostrar los videos en general con una cierta calificacion o de un cierto genero
3. Mostrar los capitulos de una determinada serie con sus calificaciones
4. Mostrar las peliculas con cierta calificacion
5. Calificar un video
6. Mostrar los videos en general
7. Mostrar los capitulos de una determinada serie con una determinada calificacion
0. Salir
Opcion -> 5

Teclee el titulo del video a calificar: Dark
Teclee el titulo del capitulo: verdades
Teclee la calificacion a otorgar (1-5): 5

Calificacion modificada con exito!

Nombre: DARK          Genero: drama          Duracion: 2 horas y 16 minutos          ID: 123463          Puntuacion: 4.33333
Temporada 1          Capitulo 1: secretos          Calificacion: 5
Temporada 2          Capitulo 1: dobles          Calificacion: 4
Capitulo 2: mentiras          Calificacion: 3
Capitulo 2: verdades          Calificacion: 5
Capitulo 3: tiempo          Calificacion: 4
Capitulo 3: despertar          Calificacion: 5

----- MENU -----
1. Cargar archivo de datos
2. Mostrar los videos en general con una cierta calificacion o de un cierto genero
3. Mostrar los capitulos de una determinada serie con sus calificaciones
4. Mostrar las peliculas con cierta calificacion
5. Calificar un video
6. Mostrar los videos en general
7. Mostrar los capitulos de una determinada serie con una determinada calificacion
0. Salir
Opcion -> 4

Teclee la calificacion a buscar (1-5): 5

Nombre: MALEFICA          Genero: misterio          ID: 509565          Duracion: 1 horas y 37 minutos          Puntuacion: 5          Clasificion: A
Nombre: SHREK          Genero: accion          ID: 835999          Duracion: 1 horas y 30 minutos          Puntuacion: 5          Clasificion: C
Nombre: TITANIC          Genero: drama          ID: 792010          Duracion: 3 horas y 15 minutos          Puntuacion: 5          Clasificion: C

Line: 307
```

```
ProjectoIntegrador
C:\Users\nenas\OneDrive\Escritorio\ProjectoIntegrador_BrendaSaucedo_A00829855\ProjectoIntegrador_BrendaSaucedo_A00829855.exe

Archivo Edición
----- MENU -----
1. Cargar archivo de datos
2. Mostrar los videos en general con una cierta calificacion o de un cierto genero
3. Mostrar los capitulos de una determinada serie con sus calificaciones
4. Mostrar las peliculas con cierta calificacion
5. Calificar un video
6. Mostrar los videos en general
7. Mostrar los capitulos de una determinada serie con una determinada calificacion
0. Salir
Opcion -> 5

Teclee el titulo del video a calificar: MALEfica
Teclee la calificacion a otorgar (1-5): 3

Calificacion modificada con exito!

Nombre: MALEFICA          Genero: misterio          ID: 509565          Duracion: 1 horas y 37 minutos          Puntuacion: 3          Clasificion: A

----- MENU -----
1. Cargar archivo de datos
2. Mostrar los videos en general con una cierta calificacion o de un cierto genero
3. Mostrar los capitulos de una determinada serie con sus calificaciones
4. Mostrar las peliculas con cierta calificacion
5. Calificar un video
6. Mostrar los videos en general
7. Mostrar los capitulos de una determinada serie con una determinada calificacion
0. Salir
Opcion -> 7

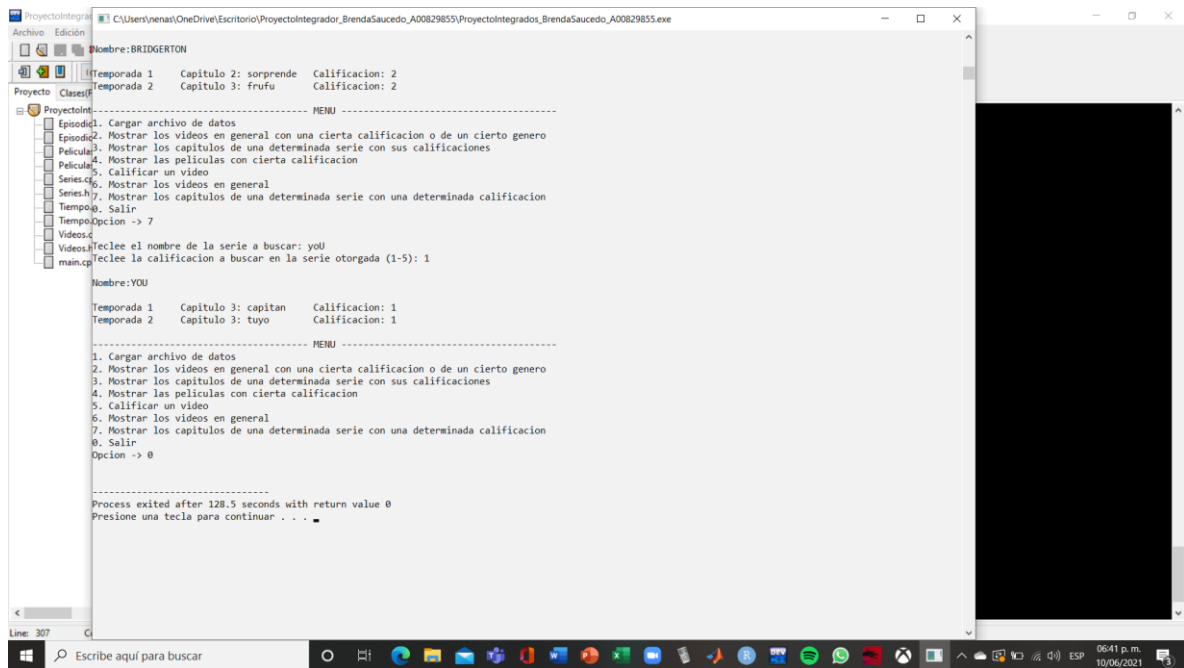
Teclee el nombre de la serie a buscar: bridgerton
Teclee la calificacion a buscar en la serie otorgada (1-5): 9
Teclee la calificacion a buscar en la serie otorgada (1-5): 2

Nombre: BRIDGERTON

Temporada 1          Capitulo 2: sorprende          Calificacion: 2
Temporada 2          Capitulo 3: frufu          Calificacion: 2

----- MENU -----
1. Cargar archivo de datos
2. Mostrar los videos en general con una cierta calificacion o de un cierto genero
3. Mostrar los capitulos de una determinada serie con sus calificaciones
4. Mostrar las peliculas con cierta calificacion
5. Calificar un video
6. Mostrar los videos en general
7. Mostrar los capitulos de una determinada serie con una determinada calificacion
0. Salir
Opcion -> 0

Line: 307
```



Argumentación de las Partes del Proyecto

En cuanto a las clases que se utilizaron y sobre el concepto de herencia, por el planteamiento de la situación problema se decidió que debía de haber una clase padre, la cual es la clase Videos, que contiene todos los atributos que comparten sus clases derivadas, es decir, las clases hijas, las cuales son las películas y series. Esto se realizó de esta forma para no tener que declarar los atributos en cada clase derivada, ya que al declarar los atributos de manera protected en la clase padre, las clases hijas heredan estos atributos, así como también los métodos públicos.

Para un mejor manejo de la duración de los videos, se decidió crear una clase Tiempo que fuera capaz de manipular la duración de una forma más sencilla. No se decidió utilizar los minutos directamente, ya que, aunque el tiempo total se da en minutos, el formato de la duración de los videos viene en horas y minutos.

También se creó la clase Episodio que contuviera la información de los capítulos, para que la clase Series pudiera manipular de manera más sencilla sus capítulos. No se crearon directamente como atributos en la clase Series, ya que, a la hora de obtener la información de los capítulos, los cuales son varios, causaría confusión a la hora de programar, razón por la cual se optó por declarar una clase que guardara esta información extra.

A la hora de programar las clases, fue necesaria una determinada implementación de los modificadores de acceso para que las clases derivadas pudieran heredar de manera correcta los atributos de la clase padre, razón por la cual en la clase padre los atributos se declararon de tipo protected y los métodos de tipo public. Si no se hubieran declarado de esta forma, las clases derivadas no podrían heredar los atributos y métodos de la clase base.

En las demás clases, los atributos se declararon de tipo private, ya que la información que contiene cada una de estas clases es ajena al usuario, es decir, son datos que no deben de ser modificados por el usuario a menos que se solicite hacerlo, como se hace exclusivamente a la hora de calificar un video, razón por la cual no se declararon como atributos públicos. Tampoco se optó por declarar los atributos como protected, ya que estas clases no le heredan sus atributos a otras clases, a excepción de la clase Videos.

También fue necesaria la sobrecarga de métodos en algunas de las clases, como la clase Videos y la clase Tiempo, ambos con el método imprimir(), para que se pudiera ahorrar código a la hora de solicitar la duración del video y desplegar sus datos, ya que ambos despliegan información distinta. No se optó por otra solución,

ya que esta fue la manera más sencilla que se encontró para desplegar los datos, al mismo tiempo que se ahorran líneas de código. Se pudo haber programado también un método imprimir() en la clase Episodio, pero por cuestiones de formato, se decidió omitir. En cuanto a la sobrecarga parametrizada, esta fue utilizada en la clase Video, en donde se utiliza el mismo método imprimir pero con parámetros "imprimir(num: int)", ya que al igual, el programa despliega un texto, la diferencia es que no despliegan la misma información. Este método también se declaró de tipo static para que no fuera necesario declarar un objeto de esta clase y poder desplegar un texto de bienvenida con el simple uso del nombre de la clase. No se decidió declarar un método distinto, ya que ambos imprimen un texto, pero con diferente contenido.

En el proyecto, fue necesario utilizar el polimorfismo por sobre escritura de métodos con ayuda de la implementación de métodos virtuales (virtuales puros y comunes). En la clase base se declararon métodos de tipo virtual, para que las clases hijas pudieran implementar sus propios métodos. Como, por ejemplo, fue necesario declarar un método virtual para calificar los capítulos de las series, y otro para desplegar los capítulos con una determinada calificación de una serie determinada, en las cuales solo la clase Series puede acceder a los capítulos. Este método no fue declarado como virtual puro, ya que la clase Películas no cuenta con capítulos.

El uso del virtual puro también fue necesario, para hacer de la clase base, una clase abstracta, en la cual no es necesario definir el contenido del método, pero si declarar el encabezado en la clase padre, así como también es forzoso que se declare y defina en las demás clases derivadas. Esta solución se implementó para declarar el método de printDetalles(), en la cual todas las clases derivadas pueden desplegar todos los detalles de sus atributos, en donde la clase padre no tiene acceso a los atributos definidos en sus clases hijas, razón por la cual no fue declarado como un virtual común.

En la clase Tiempo se utilizó la sobrecarga de operadores, ya que la información que se recibe por parte del archivo de texto es en minutos, por lo que se optó que sería una manera más sencilla sumar los minutos de tipo entero a un objeto de tipo Tiempo para que este guardara los valores correspondientes y bien calculados a sus debidos atributos, y así podernos evitar calcular para cada atributo su debida duración en horas y minutos.

También se utilizó la sobrecarga de operadores en la clase Episodio para calcular más fácilmente el acumulado y el promedio de las calificaciones de los capítulos de una determinada serie, y evitarnos el tener que calcular y programar manualmente declarando nuevas variables que pudieran aceptar un tipo de dato y operación en específico. De esta manera se volvió más flexible el realizar operaciones con otros tipos de datos definidos por el programador.

No se optó por usar la sobrecarga de operadores en otras clases, ya que las clases que se definieron en este proyecto manejan en su mayoría datos de tipo string o

bien datos los cuales no deben de ser modificados por el usuario (titulo, id, duración, genero, etc.), a excepción de la calificación que se les da a los videos, pero a esta calificación solo se le sustituye su valor por otro, por lo que no puede ser utilizada la sobrecarga de operadores en estas clases.

El uso de excepciones no fue utilizado en este proyecto, ya que en la ejecución del programa no presento errores, los archivos los lee correctamente, así como también los datos que el usuario teclea, a excepción de que por un error de dedo teclee otro tipo de dato, en este caso el programa terminaría. Si el usuario teclea el tipo de dato que corresponde, no debería marcar error el programa. El uso de excepciones hubiera hecho más fácil la continuación del programa y evitar que este terminara abruptamente por un error de dedo, aun así, se decidió omitir esta solución.

En cuanto a la interfaz del programa, se optó por dejar el menú que venía por default en las instrucciones, así como también se optó por agregar unas cuantas opciones más al menú establecido, para que este estuviera lo más completo posible, como lo es la opción que despliega todos los videos. Esta opción fue puesta para podernos asegurar de que se guardaron correctamente los datos de los archivos.

En cuanto al despliegue de resultados, se insertaron varios saltos de línea para que no interfirieran y se prestara a confusiones con datos anteriores. Se agregaron uno que otro detalle de diseño para que la interfaz pudiera ser más visual y de fácil comprensión.

Identificación de Casos que Harían que el Proyecto deje de Funcionar

Algunos de los casos que haría que el proyecto deje de funcionar sería cambiar el archivo de texto por otro, ya que el programa sigue el orden que está establecido en cada uno de los archivos. También si se borran algunos datos en los archivos, haría que el programa dejará de funcionar correctamente.

Otro de los errores comunes del programa es que, si no se teclea en un inicio la opción 1 de cargar los archivos de datos, haría que todo el programa terminara, es decir, al teclear otra opción que no fuera la 1 y teclear lo que se pide en la opción seleccionada, al final no encontraría ningún dato al que hacerle la modificación o desplegar sus resultados de la búsqueda, por lo que el programa terminaría abruptamente.

También, al momento de teclear la opción 2 de “Mostrar los videos con una cierta calificación o de un cierto género” o la opción 5 de “Calificar un video”, al momento que el usuario teclee el género, el título del video o el título del capítulo, debe de tener cuidado de no teclear espacios, ya que, si el usuario hace esto, el programa se ciclaría y dejaría de funcionar.

Otro de los casos, sería que cuando se le pide al usuario que introduzca alguno de los datos solicitados, necesariamente tengan que ser del tipo de dato requerido, ya que, si teclea otro tipo de dato, el programa se ciclaría o terminaría. Por ejemplo, cuando al usuario se le pide la calificación del video, la cual para facilitar la búsqueda se hizo de tipo entero, y el usuario teclea un carácter, el programa dejaría de funcionar.

Igualmente, el programa deja de funcionar cuando se pide un dato de tipo entero y el usuario teclea un dato de tipo flotante, como lo es en el caso de dar una calificación a una película o a un capítulo de la serie, ya que las calificaciones son de tipo entero en los videos y los capítulos. Lo mismo sucede para buscar videos con una cierta calificación, ya que la variable que guardará el valor recibido por el usuario fue declarada como entero, esto para facilitar la búsqueda, ya que las calificaciones se manejan como enteros, a excepción de la calificación promedio de la serie, por lo cual en esta parte se utilizó un ciclo y una condicional para encontrar las series con el mismo entero, sin importar los decimales.

Conclusión Personal

En este proyecto pude aplicar gran parte de los materiales y temas que aprendí en esta unidad de formación, muchos los cuales no conocía y que ahora se.

Al principio de la unidad de formación batallaba un poco para comprender los temas nuevos que se nos impartían y sobre todo en un corto tiempo, por lo que a la hora de poner en práctica esos nuevos conocimientos en las tareas, tuve que hacer un esfuerzo para ponerme a estudiar de nuevo el material de clase, así como también buscar en otras fuentes para comprender un poco mejor sobre los temas y sacar adelante las actividades de clase. Personalmente me gusta mucho lo que es la programación, no descanso hasta poder visualizar un buen avance o buenos resultados y siempre trato de dar lo mejor.

En conclusión, al poner en práctica los conocimientos adquiridos en esta unidad de formación y plasmarlos en el proyecto, pude observar que comprendo la mayoría de los temas que se impartieron en el curso, pero también siento que necesito estudiar y practicar un poco más en aquellos detalles y temas que siento que todavía me falta profundizar o conocer más a fondo para poder aplicarlos correctamente. En sí, la unidad de formación me lleno de muchos conocimientos que sé que aplicare en cursos posteriores y en la vida profesional.

Referencias Consultadas

Anónimo. (2021). Streaming: qué es, cómo funciona, precios y periodo de prueba 2021. mayo 7, 2021, de Selectra Sitio web: <https://selectra.mx/streaming>

Anónimo. (2021). Situacion problema. mayo 7, 2021, de Instituto Tecnológico y de Estudios Superiores de Monterrey Sitio web: <https://experiencia21.tec.mx/courses/138942/pages/situacion-problema>