



Instituto Tecnológico de Estudios Superiores de Monterrey
Campus Estado de México

Situación Problema:

Modelado de Servicio de Streaming

Autor:

Mario Alberto Pérez Barrera

A01799928

Unidad de formación:

Programación Orientada a Objetos II

TC1030.305

Profesor:

Roberto Martínez Román

Fecha:

17 de junio de 2023

Índice.

Introducción-----	3
Diagrama de Clases-----	4
Explicación del diagrama de clases -----	5
Ejemplos de Ejecución -----	6
Casos en los que no podría funcionar-----	9
Conclusión -----	10
Fuentes de Información -----	10

Introducción

En los últimos años, han proliferado los servicios de streaming de video bajo demanda por ejemplo Netflix, Disney Plus, HBO Max, entre otros. 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. Nos interesa construir un sistema de consulta para esta información que se encuentra en un archivo de texto.

Una versión limitada para apoyar a un futuro proveedor de este tipo de servicios es la que se describe a continuación:

Se quiere trabajar con dos tipos de videos: películas y episodios de series.

Todo video tiene:

- ID.
- Nombre de la película o nombre del episodio de una serie.
- Género (Pueden ser varios separados por el caracter &).
- Calificación en escala de 1 a 7.
- Duración en minutos.
- Fecha de estreno en formato mm/dd/aaaa.

Las series tienen episodios y cada episodio tiene:

- ID.
- Nombre.
- Temporada a la que pertenece.
- Número de episodios.

Nos interesa construir un sistema de consulta para esta información que se encuentra en un archivo de texto.

El sistema debe ser capaz de mostrar el siguiente menú de manera iterativa, termina hasta que el usuario selecciona Salir:

- 1) Cargar el archivo de datos en formato csv.
- 2) Mostrar los videos en general con:
 - Una calificación mayor o igual a un valor tecleado por el usuario.
 - Un género tecleado por el usuario.
- 3)Mostrar todos los episodios de una determinada serie que teclea el usuario.
- 4) Mostrar las películas con una calificación mayor a un número tecleado por el usuario.

5) Calificar un video.

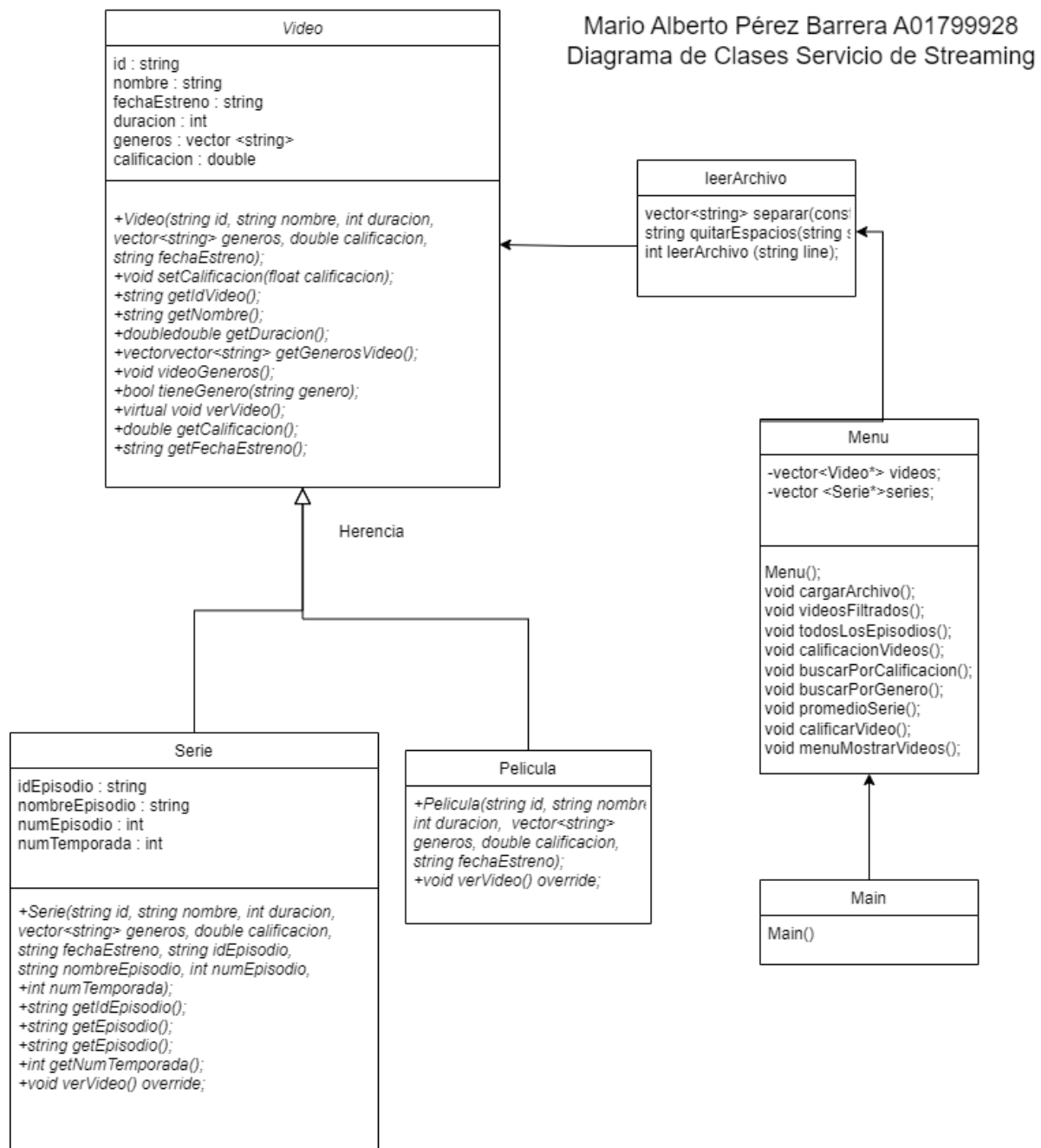
-Pedir título de película o nombre del capítulo de la serie a calificar.

-Pedir valor otorgado y asignarlo (reemplaza la calificación anterior).

6) Calcular el promedio de la calificación de una serie (promedio de todos los episodios).

7) Salir.

DIAGRAMA DE CLASES



Explicación del diagrama de Clases

En este caso, el diagrama muestra las clases **Video**, **Película** y **Serie**, junto con los archivos **leerArchivo**, **Menu** y **main**.

La clase base es **Video**, que contiene atributos como **idVideo**, **nombre** y **calificacion**, y también incluye métodos para acceder y modificar estos atributos. Tanto **Película** como **Serie** heredan de la clase **Video**, lo que significa que heredan sus atributos y métodos, pero también pueden tener atributos y métodos adicionales específicos de cada una de ellas. Por ejemplo, la clase **Serie** tiene atributos adicionales como **idEpisodio**, **nombreEpisodio**, **numTemporada**, **numEpisodio**.

La clase **leerArchivo** y **Menú** están asociadas con las clases **Video**, **Película** y **Serie**, interactúan con ellas, **leerArchivo** se encarga de leer los datos del archivo **.csv** además de separar sus líneas y evitar contar los espacios en las cadenas. Además de que crea instancias para las clases **Video**, **Película** y **Serie**.

En la clase **Menú** se implementan todos los métodos solicitados en la Situación Problema, los cuales son los que realizan todas las acciones relacionadas con los videos. Después las funciones o métodos creados en esta clase serán utilizados en el menú principal el **Main**.

Finalmente, el archivo **main** utiliza la clase **Menu** para crear y ejecutar el menú principal del programa. Esto implica que el flujo del programa se inicia en **Main**, donde se crea una instancia de **Menú** y se llama a sus métodos según las opciones seleccionadas por el usuario.

Ejemplos de ejecución

1. Se compila el archivo usando `g++ -std=c++11 *.cpp` (en caso de Windows)
2. Se ingresa `./a.exe` para crear un archivo a
3. El programa correrá esto:

(Cuando se ingresa algún acento el programa saca símbolos extraños, eso puede variar dependiendo del dispositivo del usuario).

```
| Soy Mario Alberto Pérez Barrera |
| ¿En que puedo ayudarte?        |
|-----|
| 1. Cargar datos                 |
| 2. Mostrar videos               |
| 3. Mostrar episodios de serie   |
| 4. Mostrar peliculas por califi |
| 5. Calificar video              |
| 6. Calcular promedio de una se |
| 7. Salir                        |
|-----|
```

4. Debemos de ingresar el número 1 para poder cargar los datos. Si los datos no son cargados no podremos continuar.

```
Ingrese una opción: 1
El archivo se cargo de manera exitosa
```

5. Ahora ingresamos la Opción 2:

```
1. Mostrar videos por califi |
2. Mostrar videos por g |
3. Mostrar todos los videos  |
4. Regresar                  |
Ingrese una opción: 2
```

6. Pediremos los videos por Género e ingresamos 'Aventura'

```
Los generos que hay son:
1) Acción
2) Aventura
3) Drama
4) Fantasy
5) Romance
6) Sci-Fi
7) Suspenso
8) Animación
9) Familiar
10) Musical
11) Guerra
12) Crimen
13) Misterio
14) Comedia
15) Western
16) Biografía
-----
Ingrese el genero que desea buscar, recuerde escribirlo tal cual: Aventura
```

7. El programa mostrara todos los videos con el genero de 'Aventura' (Un ejemplo de salida)

```
Id: tt0107290
Nombre: Jurassic Park
Duracion: 127
Generos: Accion Aventura Sci-Fi Suspenso
Calificacion: 5.7
Fecha de estreno: 06/11/1993

Id: tt0796366
Nombre: Star Trek
Duracion: 127
Generos: Accion Aventura Sci-Fi
Calificacion: 5.5
Fecha de estreno: 05/08/2009

Id: tt0120762
Nombre: Mulan
Duracion: 88
Generos: Animacion Aventura Familiar Fantasy Musical Guerra
Calificacion: 5.3
Fecha de estreno: 06/19/1998
```

8. Ahora, pediremos las peliculas con calificación arriba de 5

```
Ingrese una opcion: 4
Ingrese la calificacion: 5
Peliculas con calificacion mayor o igual a 5
Id: tt0107290
Nombre: Jurassic Park
Duracion: 127
Generos: Accion Aventura Sci-Fi Suspenso
Calificacion: 5.7
Fecha de estreno: 06/11/1993

Id: tt0796366
Nombre: Star Trek
Duracion: 127
Generos: Accion Aventura Sci-Fi
Calificacion: 5.5
Fecha de estreno: 05/08/2009

Id: tt0120762
Nombre: Mulan
Duracion: 88
Generos: Animacion Aventura Familiar Fantasy Musical Guerra
Calificacion: 5.3
Fecha de estreno: 06/19/1998
```

9. Ahora queremos calificar un video, yo escojo Fate Zero, pero hay un error donde no el nombre de la serie no es encontrado. En cambio si yo escribo el nombre del episodio que quiero calificar en este caso llamado 'Masp No Yaiba' el programa mostrará su calificación actual y pedirá al usuario que ingrese una nueva calificación.

```
Ingrese una opcion: 5
Ingrese el nombre de la pelicula/serie: Fate Zero
No se encontro el video

-----
| 1. Cargar datos(Solo se cargan una vez)|
| 2. Mostrar videos                       |
| 3. Mostrar episodios de serie          |
| 4. Mostrar peliculas por calificacion  |
| 5. Calificar video                     |
| 6. Calcular promedio de una serie     |
| 7. Salir                              |
-----

Ingrese una opcion: 5
Ingrese el nombre de la pelicula/serie: Masp No Yaiba
Calificacion actual: 5.8
Ingrese la calificacion: 6.4
Calificacion actualizada: 6.4
```

10. Ahora pediremos el promedio de una serie, escojo Game Of Thrones debido a que es la que tiene mas capitulos.

```
El promedio de la serie Game Of Thrones es 6.37778
```

11. Y para finalizar, si presionamos la opción 7) el programa terminará con su ejecución.

```
Ingrese una opcion: 7
PS C:\Users\mario\OneDrive\Escritorio\situacionProblema>
```


Casos en los que el programa no podría funcionar

1. Si hay errores en el código fuente, el proyecto no se compilará correctamente y no se generará un archivo ejecutable. Estos errores pueden incluir sintaxis incorrecta, referencias a clases o funciones inexistentes, o problemas de inclusión de archivos.
2. Si llegara a haber problemas con el compilador el cual llega a negar el acceso al .exe del programa, que impide que este corra.
3. Si faltara alguna biblioteca el programa no correrá.
4. Si no se ingresa un archivo correcto, como el utilizado para el ejemplo este no servirá, me refiero al número de columnas, si ingresamos más columnas se les deberá agregar nuevos métodos al programa para que este pueda servir. En cuanto al número de vídeos, ese no importa
5. Si llegara a cambiar la indentación del programa este podría mover algún ciclo y evitar que el programa termine o no funcione adecuadamente.
6. Lo más importante, todos los archivos deben de estar dentro de la misma carpeta para que el programa pueda ejecutarse.

Conclusión

Lo que más se me complicó y me tarde más en conseguir fue el leer el archivo, implementamos una función durante clases, pero como sentía que no me servía del todo la tuve que rehacer. Sin embargo no quedaba como la necesitaba.

Durante la implementación de este programa se complicó un poco debido a la gran cantidad de métodos que se deben implementar, sin embargo la obtención de los videos no era algo tan complicado debido que era lo mismo que habíamos realizado en clase.

El diagrama de clases lo fui modificando conforme iba implementando el código ya que necesitaba agregar nuevos métodos, quitar o modificar los ya existentes, fue de ayuda aunque también me confundió un poco.

En cuanto al aprendizaje, pude poner a prueba cosas que había aprendido durante cursos anteriores así como lo visto en este, pero hará falta un estudio por cuenta propia para poder mejorar lo que he visto hasta ahora.

Fuentes de información

Tc1030-Rmroman. (n.d.). *GitHub - TC1030-rmroman/situacionProblema: Situación problema para Programación orientada a objetos.* GitHub.
<https://github.com/TC1030-rmroman/situacionProblema>

Marioooba. (n.d.). *situacionProblema/README.md at main · Marioooba/situacionProblema.* GitHub.
<https://github.com/Marioooba/situacionProblema/blob/main/README.md>