

**Instituto Tecnológico y de Estudios Superiores de  
Monterrey.**

## Modelo de servicio de streaming

*Proyecto Integrador (POO)*



### Alumno:

- Josue Salvador Cano Martinez
  - A00829022

### Profesor:

- María del Carmen Pámanes Fernández

Monterrey, Nuevo León.  
05 de junio de 2020

**Objetivo:**

En los últimos años, han proliferado los servicios de streaming de video bajo demanda, por ejemplo, Netflix, Disney, DC, 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. Desarrollar una aplicación que permita modelar la situación planteada, construida a partir de clases y aplicando principios de la POO.

**Competencias:****SICT0301A**

Evalúa los componentes que integran una problemática asociada a sistemas computacionales simples, que puede encontrar en su vida cotidiana y para los cuales se tiene una descripción estructurada. Es capaz de usar conceptos básicos de las ciencias de ingeniería para detectar la condición normal de funcionamiento del sistema y posibles desviaciones sobre el mismo.

**SICT0302A**

Toma decisiones, en la solución de problemas que involucran tecnologías de información, relacionadas con el tipo de modelo (conceptual, matemático, físico) necesario para la solución del problema en cuestión, en el contexto de procesos productivos simples, que puede encontrar en su vida cotidiana y para los cuales se tiene una descripción estructurada. Es capaz de usar información disponible y modelos preseleccionados para obtener soluciones de problemas asociados con tecnologías de información; es capaz de identificar información necesaria para la generación de soluciones al problema

**SEG0702A**

Evalúa diversas tecnologías relevantes para la transformación de su práctica profesional. Es consciente de los conceptos relacionados a las tecnologías

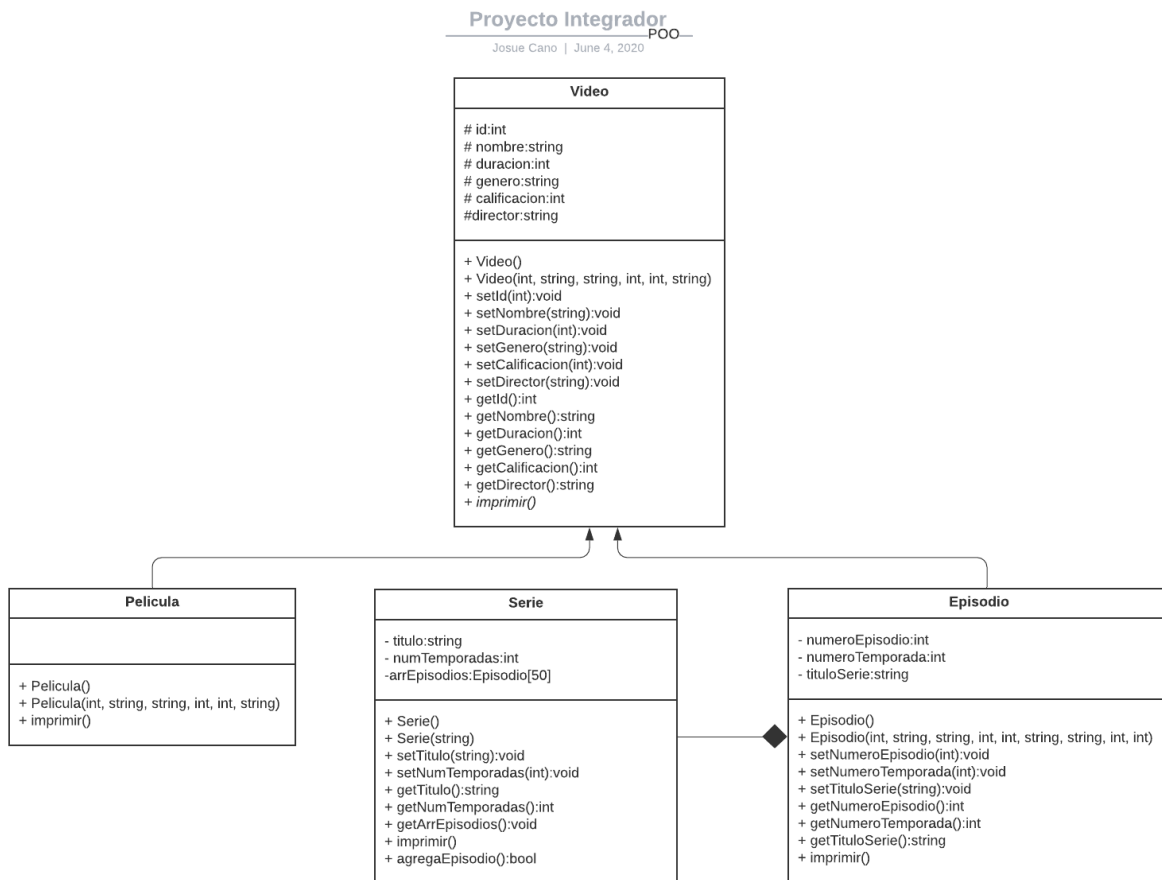
emergentes. Es usuario de diferentes sistemas computacionales ligados a su profesión. Toma conciencia de la importancia de la transformación digital. Para esto, aplica el razonamiento algorítmico para modelar situaciones reales ligadas a su actividad profesional.

## **Introducción:**

El avance del desarrollo tecnológico ha proliferado el uso de servicios digitales de entretenimiento entre la sociedad, claros ejemplos como lo son: Netflix, Disney, Spotify, y DC, entre otros más. Dichos servicios streaming se caracterizan por la gran cantidad de contenido digital (video y música) que ponen al alcance de sus usuarios. Como bien se sabe, no todas las personas comparten iguales gustos e intereses, esto impulsa, entre otras cosas, el auge del análisis de datos sobre el comportamiento de los usuarios con el fin de lograr proveerles un servicio personalizado que los convierta en clientes fieles. Para la situación problema se tiene dos tipos de video: películas y series; dichos videos tienen un ID, un nombre, una duración y un género; las series tienen episodios, donde cada episodio, además de tener los parámetros que tiene la película, también tiene un título y una temporada. Se busca determinar la calificación promedio de cada video (en escala de 1 a 5). Debido a que, como se mencionó antes, se busca ofrecer al cliente un servicio donde él mismo pueda determinar qué ver, se requiere elaborar un sistema que permita crear ciertos frameworks con los datos que se tienen, esto con el fin de lograr brindar la posibilidad de ejecutar distintas funciones selectivas: mostrar videos con sus calificaciones, mostrar episodios de una serie con sus calificaciones y mostrar películas con sus calificaciones son algunos de ellos.

Durante este proyecto se trabajó en el desarrollo del sistema mencionado, haciendo uso de clases y aplicando ciertos conceptos como polimorfismo y herencia en la programación orientada a objetos es como se logró conseguir el desarrollo del objetivo planteado.

## Diagrama de clases UML:



## Argumentación del diagrama UML:

Como es posible de visualizar en el diagrama, para el funcionamiento de la aplicación principal (main.cpp) fue necesaria la implementación de 4 clases: Video.h, Pelicula.h, Episodio.h y Serie.h

Si se vuelve a analizar la descripción del planteamiento de la problemática es posible percatarse que se tienen Videos, tanto las películas como los episodios que conforman las series son videos, de ello parte el planteamiento a partir del cual se define a la clase Video.h como clase padre por heredar ciertos métodos compartidos tanto a película.h como a Episodio.h; de la

misma manera, la clase Video.h es una clase abstracta, esto es posible de argumentar al visualizar que la función imprime está inicializada como un método pure virtual, tanto Pelicula.h como también Episodio.h las utilizarán para desplegar sus datos, sin embargo, cada clase lo implementará de una forma distinta (debido a que no son los mismos datos que se imprimen para cada caso); se puede confirmar que el método es virtual al observar la palabra reservada 'virtual' en su declaración, además de que se encuentra igualado a cero. Cabe agregar que en la clase video se desarrollan ciertos atributos que son mencionados en la descripción problema como propios de los videos: el ID, el nombre, la duración, el género y la calificación, los cuales, a su vez son heredados a las clases hijas; como es posible percatarse, también se tiene el atributo 'director' con sus respectivos métodos e inicializado en sus respectivos constructores, esto se trata de una función extra que se implementó al programa, la cual permite validar si el usuario conocer quién fue el director de cierto video encontrado en los registros.

Si se prosigue con analizar la clase Pelicula.h es posible percatarse que hereda atributos de su clase padre (Video.h), además de que desarrolla la función 'imprimir()' para desplegar los datos correspondientes a cada película. Para el caso de esta clase no es posible identificar atributos privados debido a que los datos que identifican a una película se encuentran ya descritos en la clase Video.h (los cuales heredó), de esta manera, sólo es necesario inicializar los constructores (aunque estos no concentren asignaciones de variables) y desarrollar la función imprimir().

Si se analiza la clase Episodio.h es posible percatarse que también incluye los atributos que hereda de la clase padre Video.h, sin embargo, aunado a esto, también incluye ciertos atributos privados que son propios de un episodio, que no comparte con una película y, por tanto, no fueron incluidos en la clase Video.h Dichos atributos corresponden al título de la serie a la cual pertenece el episodio, el número de episodio del cual estamos hablando y en número de la temporada a la cual pertenece, para estos atributos es

necesario definir sus métodos set y get, incluirlos en los constructores y tomarlos en cuenta al desplegar los valores de los datos dentro de la función 'imprimir()' (he aquí porque el método imprimir es diferente para cada caso, dependiendo de la clases en la cual se esté trabajando).

Por último, se tiene la clase Serie.h, como se sabe, un conjunto de episodios conforma una serie, por tanto, en dicha clase es necesario incluir una llamada a la clase Episodio.h previamente creada. Los atributos propios de esta clase son: el título de la serie, el número de temporadas que lo componen y el arreglo de los episodios que lo conforman; estos atributos tienen definidos sus métodos set y get, son inicializados e incluidos en los constructores de esta clase; de igual manera se implemente una función que permite agregar episodios y se personaliza la función 'imprimir()' acorde a lo que se busca desplegar ( el nombre de la serie con sus respectivos episodios).

### **Características de la App:**

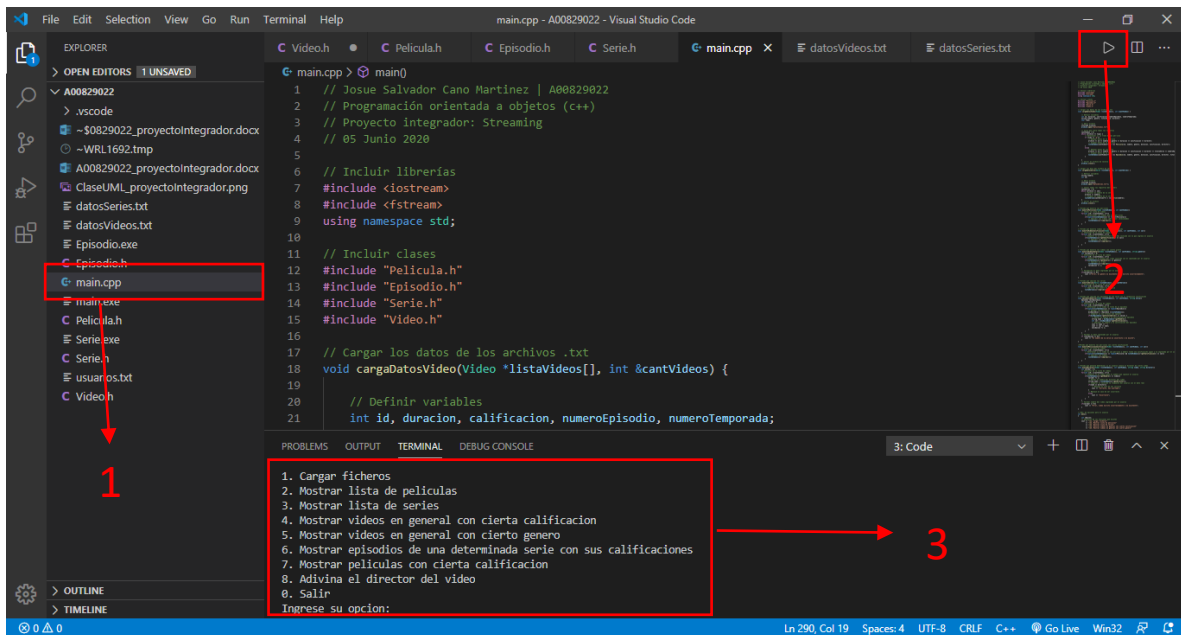
1. Cargar fichero
2. Mostrar lista películas
3. Mostrar lista series
4. Mostrar videos en general con cierta calificación
5. Mostrar videos en general de cierto género
6. Mostrar episodios de una serie con sus calificaciones
7. Mostrar películas con cierta calificación
8. Adivina el director del video
1. Salir

Nota: se deben de validar las entradas del usuario

### **Descripción: ¿Cómo ejecutar el programa?**

Para la ejecución del programa (que simula la interacción que el usuario llegaría a tener con el sistema desarrollado), es necesario solamente compilar el código del archivo main.cpp (la aplicación); no se requiere

efectuar cierto proceso o acción dentro de una clase, debido a que el archivo main.cpp ya las incluye todo se realizará dentro del mismo. A continuación, se muestra una descripción de cómo debería de ejecutarse en Visual Studio:



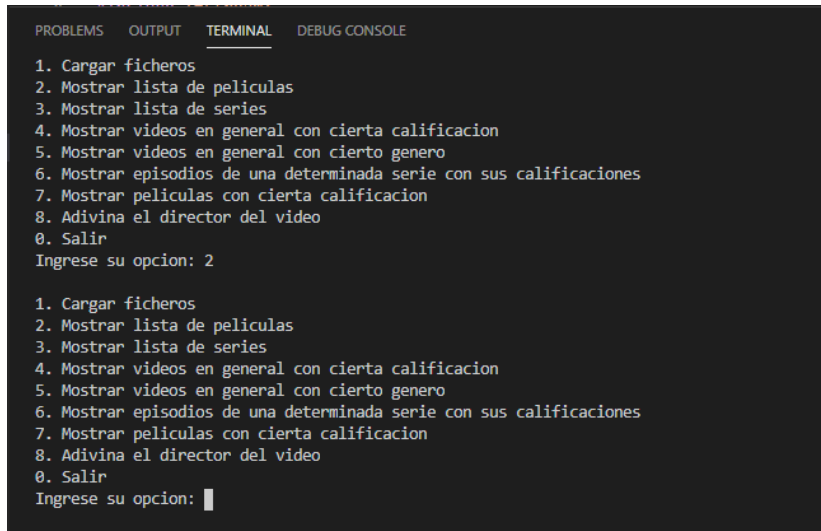
1. Abrir Visual Studio Code y posicionarse en la carpeta donde se encuentra el archivo main.cpp junto a las demás clases y archivos de texto (con datos). Una vez abierta la carpeta, seleccionar en la barra de navegación de la izquierda el archivo que lleva extensión .cpp, en nuestro caso: main.cpp
2. Al abrir el archivo de la aplicación (.cpp) se abrirá una ventana donde se podrá visualizar el código, hacer clic en el botón llamado: 'Run code' (con geometría de un triángulo) localizado en la parte superior derecha de la interfaz. También se puede usar el comando: `ctr+alt+n`
3. Se abrirá una terminal en la parte inferior, en ella se mostrará un menú que nos permitirá interactuar con el sistema.

### Ejemplo de ejecución:

A continuación, se efectuará la demostración del funcionamiento de cada acción que ofrece el menú, esto con el fin de lograr determinar cómo utilizarlo e identificar posibles errores que pueden llegar a presentarse durante su uso (sobre todo por mal ingreso en la sintaxis de los datos que se solicitan).

#### 1. Cargar ficheros

Permite cargar los datos correspondientes a los vídeos y a las series que se encuentran en los archivos .txt Si no se efectúa este proceso no se podrá efectuar ninguno de las demás funciones debido a que realmente no se tendrían datos para trabajar, por ello, antes de probar con las demás opciones, es necesario ingresar la opción 1.



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

1. Cargar ficheros
2. Mostrar lista de películas
3. Mostrar lista de series
4. Mostrar videos en general con cierta calificación
5. Mostrar videos en general con cierto genero
6. Mostrar episodios de una determinada serie con sus calificaciones
7. Mostrar películas con cierta calificación
8. Adivina el director del video
0. Salir
Ingrese su opcion: 2

1. Cargar ficheros
2. Mostrar lista de películas
3. Mostrar lista de series
4. Mostrar videos en general con cierta calificación
5. Mostrar videos en general con cierto genero
6. Mostrar episodios de una determinada serie con sus calificaciones
7. Mostrar películas con cierta calificación
8. Adivina el director del video
0. Salir
Ingrese su opcion: █
```

Como se puede apreciar, en este ejemplo se decidió comenzar directamente con la opción 2, sin embargo, no se mostró ninguna película debido a que no hay datos cargados.



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

1. Cargar ficheros
2. Mostrar lista de peliculas
3. Mostrar lista de series
4. Mostrar videos en general con cierta calificacion
5. Mostrar videos en general con cierto genero
6. Mostrar episodios de una determinada serie con sus calificaciones
7. Mostrar peliculas con cierta calificacion
8. Adivina el director del video
0. Salir
Ingrese su opcion: 1

1. Cargar ficheros
2. Mostrar lista de peliculas
3. Mostrar lista de series
4. Mostrar videos en general con cierta calificacion
5. Mostrar videos en general con cierto genero
6. Mostrar episodios de una determinada serie con sus calificaciones
7. Mostrar peliculas con cierta calificacion
8. Adivina el director del video
0. Salir
Ingrese su opcion: 2
Id: 110
Nombre: Avatar
Genero: Ciencia_ficcion
Duracion: 161
Calificacion: 5
```

Para este segundo caso, primero se ingresó la opción 1 (carga de datos) y posteriormente la opción 2 (mostrar películas). Como se puede apreciar, el sistema es funcional.

## 2. Mostrar lista de películas

Dentro del archivo .txt se tienen datos de dos tipos de video: películas y episodios; con esta opción lo que se hace es extraer aquellos videos que pertenezcan a películas para posteriormente imprimirlos. Una vez que se han cargado los ficheros lo único que se debe de hacer es ingresar '2' en el campo: 'ingrese su opción', de esta forma, se desplegaran todas aquellas películas registradas en el fichero.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

1. Cargar ficheros
2. Mostrar lista de peliculas
3. Mostrar lista de series
4. Mostrar videos en general con cierta calificacion
5. Mostrar videos en general con cierto genero
6. Mostrar episodios de una determinada serie con sus calificaciones
7. Mostrar peliculas con cierta calificacion
8. Adivina el director del video
0. Salir
Ingrese su opcion: 2
Id: 110
Nombre: Avatar
Genero: Ciencia_ficcion
Duracion: 161
Calificacion: 5
El director del video es: james
Id: 111
Nombre: Batman_El_caballero_de_la_noche
Genero: Accion
Duracion: 152
Calificacion: 4
El director del video es: christopher
Id: 112
Nombre: Guason
Genero: Drama
Duracion: 121
```

Despliegue de los datos correspondientes a las películas registradas en el fichero .txt

### 3. Mostrar lista de series

Aparte del archivo que contiene los videos se tiene otro archivo más que incluye en su contenido las series de las cuales se desglosan los Episodios albergados en el fichero de videos. En esta opción lo que se busca es desplegar los una lista con los nombres de las series que se tienen, por tanto, lo único que se deberá hacer será ingresar '3' en el campo de: 'ingrese su opción:' y el programa automáticamente desplegará una lista con las series registradas.

```
1. Cargar ficheros
2. Mostrar lista de peliculas
3. Mostrar lista de series
4. Mostrar videos en general con cierta calificacion
5. Mostrar videos en general con cierto genero
6. Mostrar episodios de una determinada serie con sus calificaciones
7. Mostrar peliculas con cierta calificacion
8. Adivina el director del video
0. Salir
Ingrese su opcion: 3
Serie: Stranger_things
Serie: How_i_met_your_mother
Serie: La_casa_de_las_flores
```

Como puede apreciarse, en el fichero datosSeries.txt se tienen 3 registros correspondientes a las 3 series (de las cuales son parte los episodios albergados en el fichero de videos).

#### 4. Mostrar videos en general con cierta calificación

Esta opción permite al usuario visualizar aquellos videos que tengan cierto promedio de calificación definido por las personas que los han visualizado (el cual va de un rango de 1 a 5). Para lograr visualizar su funcionalidad lo único que debe de hacerse es escribir '4' en el campo correspondiente a: 'ingrese su opción', posterior a ello deberá de escribirse la calificación con la cual se busca efectuar la filtración de los videos.

```
1. Cargar ficheros
2. Mostrar lista de peliculas
3. Mostrar lista de series
4. Mostrar videos en general con cierta calificacion
5. Mostrar videos en general con cierto genero
6. Mostrar episodios de una determinada serie con sus calificaciones
7. Mostrar peliculas con cierta calificacion
8. Adivina el director del video
0. Salir
Ingrese su opcion: 4
Ingrese la calificacion (entero entre 1 y 5): 4
Id: 111
Nombre: Batman_El_caballero_de_la_noche
Genero: Accion
Duracion: 152
Calificacion: 4
El director del video es: christopher
Id: 113
Nombre: La_La_Land
Genero: Musical
Duracion: 127
Calificacion: 4
El director del video es: damien
Id: 114
Nombre: Toy_Story_3
Genero: Animacion
```

Impresión de videos con calificación de 4

## 5. Mostrar videos en general con cierto género

De todos los videos registrados en el fichero datosVideos.txt se seleccionan aquellos donde su género coincide con el proporcionado como dato de entrada por el usuario. Es importante ser cauteloso con la sintaxis ya que se trata de comparación de strings y, por tanto, el valor que escribe el usuario debe de coincidir con el dato del registro que se encuentra en el fichero. Lo que se debe de tener en cuenta es: el género debe de escribirse sin acentos ni caracteres especiales, los espacios (en caso de que los incluya) deben de sustituirse con un guion bajo ('\_'), y si el género se compone de dos o más palabras, sólo se escribirá con mayúscula la inicial de la primera palabra.

```
1. Cargar ficheros
2. Mostrar lista de peliculas
3. Mostrar lista de series
4. Mostrar videos en general con cierta calificacion
5. Mostrar videos en general con cierto genero
6. Mostrar episodios de una determinada serie con sus calificaciones
7. Mostrar peliculas con cierta calificacion
8. Adivina el director del video
0. Salir
Ingrese su opcion: 5
Ingrese el genero (sintaxis: Genero | Mi_genero): ciencia ficcion
Error, el genero no existente o fue escrito incorrectamente
```

Ejemplo de un error debido a una mala sintaxis en la escritura del género; nótese que la primera letra de la primera palabra se escribió con minúscula y debería de haber sido escrita con mayúscula.

```
Ingrese su opcion: 5
Ingrese el genero (sintaxis: Genero | Mi_genero): Ciencia_ficcion
Id: 110
Nombre: Avatar
Genero: Ciencia_ficcion
Duracion: 161
Calificacion: 5
El director del video es: james
Id: 117
Nombre: Avengers_Endgame
Genero: Ciencia_ficcion
Duracion: 181
Calificacion: 5
```

Mismo género ingresado con una sintaxis correcta, esta vez si se despliegan los registros.

#### 6. Mostrar episodios de una determinada serie con sus calificaciones

En este caso el usuario ingresa el nombre de una serie y el programa despliega los episodios que la conforman, incluyendo sus respectivas calificaciones. Para su correcto funcionamiento, se debe de ingresar el nombre de la serie tomando en cuenta la siguiente sintaxis: en lugar de espacios se utiliza guion bajo, no incluir caracteres especiales (tildes, por ejemplo), todo se escribe con minúscula (excepto la primera letra de la primera palabra, la cual se escribe con mayúscula).

```
1. Cargar ficheros
2. Mostrar lista de peliculas
3. Mostrar lista de series
4. Mostrar videos en general con cierta calificacion
5. Mostrar videos en general con cierto genero
6. Mostrar episodios de una determinada serie con sus calificaciones
7. Mostrar peliculas con cierta calificacion
8. Adivina el director del video
0. Salir
Ingrese su opcion: 6
Ingrese la serie (sintaxis: Stranger_things): strangerthings
El nombre de la serie es incorrecto o no existe
```

Como puede apreciarse, en este caso la sintaxis fue incorrecta: no se usó el guion bajo para separar las palabras, y la primera letra de la primera palabra no fue escrita con mayúscula, por tanto, marca error y no despliega los episodios correspondientes.

```
Ingrese su opcion: 6
Ingrese la serie (sintaxis: Stranger_things): La_casa_de_las_flores
La_Casa_de_las_Flores 5
Es_Mejor_Asi 5
Mio 5
Tormento 4
Tímido 4
Senora 3
Mi_Gran_Noche 4
Nieva_Nieva 5
El_Triste 5
```

Puede visualizarse un correcto uso de la sintaxis en el nombre de la serie, obteniendo así un despliegue de los episodios que la conforman y de la calificación respectiva a cada episodio.

## 7. Mostrar películas con cierta calificación

Para esta opción lo que hace el programa es obtener todos aquellos registros correspondientes a películas (encontrados en el archivo .txt de datosVideo) y posteriormente selecciona aquellos que tienen un valor para 'Calificación' igual al valor que ingresa el usuario, de esta manera, se puede hacer una búsqueda filtrada que arroje registros con cierto valor específico en su calificación. Es importante mencionar que se debe de ingresar un valor entre 1 y 5.

```
1. Cargar ficheros
2. Mostrar lista de peliculas
3. Mostrar lista de series
4. Mostrar videos en general con cierta calificacion
5. Mostrar videos en general con cierto genero
6. Mostrar episodios de una determinada serie con sus calificaciones
7. Mostrar peliculas con cierta calificacion
8. Adivina el director del video
0. Salir
Ingrese su opcion: 7
Ingrese la calificacion (entero entre 1 y 5): 10
ERROR, dato fuera de parametrosIngrese la calificacion (entero entre 1 y 5):
```

Error en la ejecución debido a que se ingresó un valor fuera de los parámetros permitidos.

```
Ingrese su opcion: 7
Ingrese la calificacion (entero entre 1 y 5): 10
ERROR, dato fuera de parametros
Ingrese la calificacion (entero entre 1 y 5): 5
Id: 110
Nombre: Avatar
Genero: Ciencia_ficcion
Duracion: 161
Calificacion: 5
El director del video es: james
Id: 112
Nombre: Guason
Genero: Drama
Duracion: 121
```

Al ingresar un valor entero dentro del rango permitido (en este caso 5) el programa despliega aquellas películas con esta calificación

## 8. Adivina el director del video

Esta opción pone a prueba los conocimientos del usuario respecto a quiénes son los autores de cada video encontrado en la lista; para esto, el usuario deberá ingresar el nombre del video del cual quiere adivinar su director, debe cuidar la sintaxis con la cual ingresa el nombre del video: la primera letra de cada palabra con mayúscula (a excepción de los conectores, los cuales en su totalidad se escriben con minúscula) y las palabras deben de estar separadas por un guion bajo (sin usar espacios). Una vez se ha ingresado el nombre del video se pedirá ingresar el nombre del director, el cual debe de ser escrito en su totalidad con minúsculas, sin caracteres especiales y debe de escribirse sólo un nombre (sin apellidos no espacios). El programa determinará si el usuario acertó, si no acertó o si ingreso incorrectamente algún dato, para lo cual desplegará un mensaje correspondiente a cada caso.

```
Ingrese su opcion: 8
Ingrese el nombre del video:Nota: primera letra de cada palabra con mayuscula, conectores
se escriben con minusculas, palabras separadas por guion bajo: toystory3
Ingrese solo el primer nombre del director (en minusculas): lee
Error, video escrito incorrectamente o no existente
```

El programa marca error porque a pesar de que al nombre del director es correcto el nombre del vídeo no cumple la sintaxis establecida.

```
Ingrese su opcion: 8
Ingrese el nombre del video:Nota: primera letra de cada palabra con mayuscula, conectores
se escriben con minusculas, palabras separadas por guion bajo: Toy_Story_3
Ingrese solo el primer nombre del director (en minusculas): lee
Correcto, has acertado
```

Para este caso, tanto los datos como la sintaxis son correctos; ¡hemos acertado!

```
Ingrese su opcion: 8
Ingrese el nombre del video:Nota: primera letra de cada palabra con mayuscula, conectores
se escriben con minusculas, palabras separadas por guion bajo: Toy_Story_3
Ingrese solo el primer nombre del director (en minusculas): james
Incorrecto
```

En este caso, la sintaxis de los datos de entrada es correcta, sin embargo, el nombre del director no es correcto, por ello despliega el mensaje: "Incorrecto"

## 0. Salir

El menú se encuentra dentro de un ciclo que no terminará hasta que se introduzca 0 como opción, lo que se hace simplemente es romper la condición que mantiene el ciclo y, por ende, el programa termina.

```
1. Cargar ficheros
2. Mostrar lista de películas
3. Mostrar lista de series
4. Mostrar videos en general con cierta calificación
5. Mostrar videos en general con cierto genero
6. Mostrar episodios de una determinada serie con sus calificaciones
7. Mostrar películas con cierta calificación
8. Adivina el director del video
0. Salir
Ingrese su opción: 0
PS C:\Users\Josué\Documents\Programación orientada a Objetos\A00829022>
```

Programa terminado al ingresar la opción '0'

## Conclusión personal:

Nos encontramos en una era donde los servicios están migrando a prestarse en un formato digitalizado y donde los nuevos modelos de negocio se acoplan en este formato; la competencia empresarial busca ofrecer el mejor servicio posible a sus clientes y basado en ello es como se busca efectuar el manejo de datos de los usuarios. Cuando hablamos de un servicio que involucra un impacto nacional e incluso internacional el tamaño de los datos es muy grande y resultaría imposible manejarlos de no ser por las herramientas que nos ofrece la programación orientada a objetos. Las aplicaciones que puede llegar a tener son bastas, en diferentes áreas de nuestra vida: salud, construcción y prácticamente todo aquello de la vida real que pueda ser representado como un objeto. El objetivo principal de los procesos, métodos y herramientas que proporciona la POO a este tipo de casos es la optimización (gastar menos recursos y ofrecer un resultado más eficiente), el control (tener un orden y estructuración de las grandes magnitudes de datos) y la simplificación (aquello que resultaría ser tedioso y laborioso resulta simplificarse bastante incorporando el uso de clases, métodos y funciones).



## Referencias consultadas:

- Alday, D. (2018). *superaficionados*. Recuperado el 04 de 06 de 2020, de <https://www.superaficionados.com/mejores-peliculas-historia-cine/>
- Caviaro, J. L. (18 de 01 de 2020). *espinof*. Recuperado el 04 de 06 de 2020, de <https://www.espinof.com/listas/mejores-peliculas-2019-ahora>
- Deitel, P. J. (2014). *C++ : how to program* (9th ed ed.). Prentice Hall. Recuperado el 01 de 06 de 2020
- ecartelera. (s.f.). *ecartelera*. Recuperado el 04 de 06 de 2020, de <https://www.ecartelera.com/listas/mejores-peliculas/>
- formatv. (s.f.). *formatv*. Recuperado el 04 de 06 de 2020, de <https://www.formatv.com/series/la-casa-de-las-flores/capitulos/>
- GeeksforGeeks. (s.f.). *GeeksforGeeks*. Recuperado el 01 de 06 de 2020, de <https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/>
- infobae. (21 de 05 de 2020). *infobae*. Recuperado el 04 de 06 de 2020, de <https://www.infobae.com/america/entretenimiento/2020/05/22/estas-son-las-50-mejores-peliculas-de-la-historia-elegidas-por-los-actores-y-directores-de-hollywood/>
- Netflix. (2018). *Netflix*. Recuperado el 04 de 06 de 2020, de <https://www.netflix.com/mx/title/80160935>
- Rosén, L., Torres, D., & Zambra, D. (02 de 06 de 2020). *finde*. Recuperado el 04 de 06 de 2020, de <https://finde.latercera.com/series-y-peliculas/50-mejores-peliculas-en-netflix-2020-junio/>
- Zorilla, M. (23 de 05 de 2020). *espinof*. Recuperado el 04 de 06 de 2020, de <https://www.espinof.com/listas/mejores-peliculas-viajes-tiempo>