

**Instituto Tecnológico y de Estudios Superiores de Monterrey (CCM)**



*Escuela de Ingeniería y Ciencias*

# Programación Orientada a Objetos

*(Gpo 326)*

## *Proyecto integrador*

### **INTEGRANTES:**

//Jennifer Avendaño Sánchez-A01656951

//Pastén Da Silva Julia - A01660665

### **DOCENTE:**

Jose Luis González

## Índice

---

Portada.....	1
Introducción .....	3
UML.....	4
Argumentación del Código.....	5
Ejecución del Código.....	12
Conclusión Individual.....	14

## Introducción

En el presente trabajo se apreciará la necesidad de un análisis sistemático de una situación problemática con la finalidad de identificar adecuadamente los factores que dan origen a las desviaciones observadas a la solución al reto el cual nos indica que debemos entender el funcionamiento básico de un servidor streaming. Para que este sea considerado un sistema básico debe ser capaz de :mostrar los videos en general con sus calificaciones,mostrar los episodios de una determinada serie con sus calificaciones,mostrar las películas con sus calificaciones. Las calificaciones esperadas de estos productos de entretenimiento deben de estar en una escala del 1 al 5 , donde se entiende que 1 es muy malo y 5 es demasiado bueno.

El fin de realizar este reto es poder aplicar todas las herramientas , contenidos conceptuales y prácticos vistos en clase :

- Herencia y Polimorfismo
- Diseño de POO
- Concepto de herencia
- Modificadores de acceso
- Sobrecarga y sobreescritura de métodos
- Concepto de polimorfismo
- Clases abstractas e interfaces
- Sobrecarga de operadores
- Introducción a la sobrecarga
- Operadores permitidos
- Sobrecarga de operadores dentro de una clase
- Manejo de excepciones
- Excepciones predefinidas
- Uso de excepciones

# UML

## Class Video

Decidimos estructurarlo así porque es de donde el menú y filtros de búsquedas obtendrán los datos de episodios y películas para rehacer los vectores del título a clasificar por el usuario.

## Class Pelicula

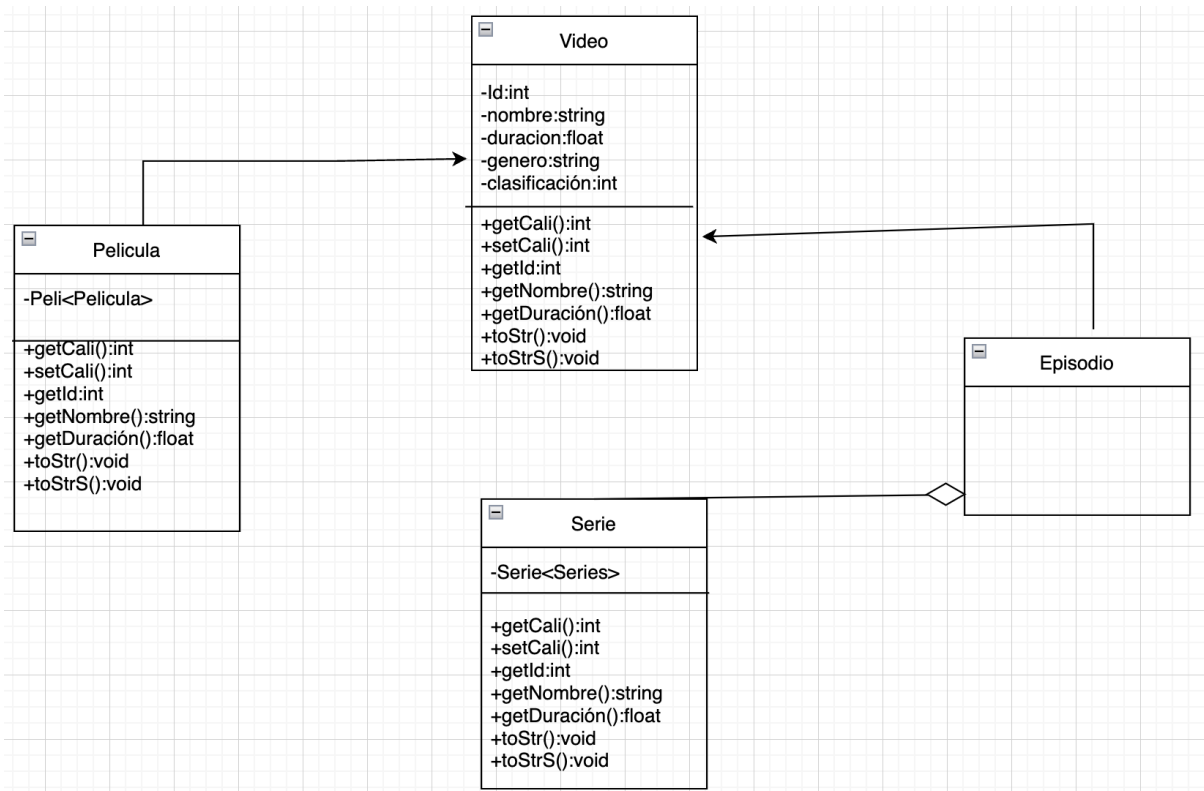
Le especificamos que sera la unica clase que tenga su propio vector privado y de donde la clase video heredera los parámetros para los objetos película

## Class Serie

Indicamos que debe heredar parámetros de video y construir su vector para compartirlo con episodio.

## Class Episodios

Le especificamos la unión con los parámetros de la clase video.



## Argumentación del Código

```
Get Started  C Episodio.h  C main.cpp  Pelicula  Episodios.txt  Pelicula.txt x  C Pelicula.h  C Serie.h  C Video.h
Pelicula.txt
1 1-Harry Potter y la piedra Filosofal-2.32-Fantasia/Aventura-5
2 2-Harry Potter y la camara secreta-2.41-Fantasia/Aventura-3
3 3-Harry Potter y el prisionero de Azkaban-2.22-Fantasia/Aventura-4
4 4-Harry Potter y el caliz de fuego-2.37-Fantasia/Aventura-5
5 5-Harry Potter y la Orden del Fenix-2.18-Fantasia/Aventura-4
6 6-Harry Potter y el misterio del Principe-2.33-Fantasia/Aventura-3
7 7-Harry Potter y las Reliquias de la Muerte-2.27-Fantasia/Aventura-5
8 8-Harry Potter y las Reliquias de la Muerte-2.10-Fantasia/Aventura-5
9 9-Animales fantásticos y dónde encontrarlos-2.16-Fantasia/Aventura-4
10 10-Animales fantásticos: los crímenes de Grindelwald-2.14-Fantasia/Aventura-4
11 11-Animales fantásticos: Los secretos de Dumbledore-2.23-Fantasia/Aventura-3
12 12-Crepusculo-2.02-Romance/Fantasia-3
13 13-Crepusculo:Luna Nueva-2.10-Romance/Fantasia-2
14 14-Crepusculo:Ecclipse-2.04-Romance/Fantasia-4
15 15-Crepusculo:Amanecer-1.57-Romance/Fantasia-5
16 16-Crepusculo:Amanecer2-1.55-Romance/Fantasia-5
17 17-A todos los chicos de los que me enamore-1.40-Comedia Romantica-5
18 18-A todos los chicos de los que me enamore 2-1.44-Comedia Romantica-5
19 19-A todos los chicos de los que me enamore 3-1.50-Comedia Romantica-5
20 20-El stand de los besos-1.46-Comedia Romantica-5
```

Este fue el text de nuestro catálogo de películas que escogimos en el cual podemos ver sus datos.

```
Get Started  C Episodio.h  C main.cpp  Pelicula  Episodios.txt x  C Pelicula.h  C Serie.h  C Video.h
Episodios.txt
1 T01S01C01-Euphoria:Piloto-0.53-Drama-4
2 T01S01C02-Euphoria:Stuntin Like My Daddy-1.00-Drama-5
3 T01S01C03-Euphoria:Made You Look-0.56-Drama-5
4 T01S01C04-Euphoria:Shock One-0.56-Drama-4
5 T01S01C05-Euphoria:03 Bonnie and Clyde-0.52-Drama-5
6 T01S01C06-Euphoria:The Next Episode-0.50-Drama-3
7 T01S01C07-Euphoria:The Trials and Tribulations of Trying to Pee While Depressed-0.59-Drama-5
8 T01S01C08-Euphoria:And Salt the Earth Behind You-1.04-Drama-2
9 T01S02C01-Euphoria:Trying to Get to Heaven Before They Close the Door-1.01-Drama-5
10 T01S02C02-Euphoria:Out of Touch-0.57-Drama-5
11 T01S02C03-Euphoria:Ruminations: Big and Little Bullys-1.00-Drama-5
12 T01S02C04-Euphoria:You Who Cannot See, Think of Those Who Can-1.02-Drama-4
13 T01S02C05-Euphoria:Stand Still Like the Hummingbird-0.54-Drama-5
14 T01S02C06-Euphoria:A Thousand Little Trees of Blood-1.00-Drama-3
15 T01S02C07-Euphoria:The Theater and It's Double-0.58-Drama-5
16 T01S02C08-Euphoria:All My Life, My Heart Has Yearned for a Thing I Cannot Name-1.01-Drama-5
17 T02S01C01-Los Juegos del Calamar:Luz Roja,Luz Verde-1-Suspense-4
18 T02S01C02-Los Juegos del Calamar:Infierno-1.03-Suspense-4
19 T02S01C03-Los Juegos del Calamar:El hombre del paraguas-0.54-Suspense-3
20 T02S01C04-Los Juegos del Calamar:No abandones el equipo-0.54-Suspense-4
21 T02S01C05-Los Juegos del Calamar:Un mundo justo-0.52-Suspense-4
22 T02S01C06-Los Juegos del Calamar:Gganbu-1.02-Suspense-4
23 T02S01C07-Los Juegos del Calamar:Los VIP-0.58-Suspense-4
24 T02S01C08-Los Juegos del Calamar:El líder-0.32-Suspense-4
25 T02S01C09-Los Juegos del Calamar:Un día de suerte-0.56-Suspense-4
26 T03S01C01-Destino La Saga Winx:Con un Hada , de la mano-0.53-Fantasia-1
27 T03S01C02-Destino La Saga Winx:Aqui no hay extraños-0.48-Fantasia-2
28 T03S01C03-Destino La Saga Winx:Pesadas esperanzas mortales-0.52-Fantasia-2
29 T03S01C04-Destino La Saga Winx:Angeles caidos-0.47-Fantasia-3
30 T03S01C05-Destino La Saga Winx:Marchitarse en la verdad-0.50-Fantasia-3
31 T03S01C06-Destino La Saga Winx:Un corazon fanatico-0.54-Fantasia-3
32 T04S01C01-Recursos Humanos:El parto-0.25-Comedia-4
33 T04S01C02-Recursos Humanos:Dia de la capacitación-0.26-Comedia-3
34 T04S01C03-Recursos Humanos:Mamis malas-0.26-Comedia-4
35 T04S01C04-Recursos Humanos:Rutgers, un paraíso de amor-0.26-Comedia-4
36 T04S01C05-Recursos Humanos:El amor en los tiempos de postparto-0.26-Comedia-4
37 T04S01C06-Recursos Humanos:El angel de la adicción-0.26-Comedia-3
38 T04S01C07-Recursos Humanos:Convencion internacional de criaturas-0.28-Comedia-3
39 T04S01C08-Recursos Humanos:La luz-0.27-Comedia-3
40 T04S01C09-Recursos Humanos:El fin se acerca-0.28-Comedia-3
41 T04S01C10-Recursos Humanos:Tormenta de Mierda-0.28-Comedia-3
42
```

Este fue el text de nuestro catálogo de episodios de las series que escogimos en el cual podemos ver sus datos.

```
Get Started  main.cpp  Pelicula.h  Serie.h  Video.h  Episodio.h
C Pelicula.h
1  //Jennifer Avendaño Sánchez-A01656951
2  //Pastén Da Silva Julia - A01660665
3  #include <vector>
4
5  #ifndef _video_
6  #define _video_
7  #include "Video.h"
8  #endif
9
10 class Pelicula: public Video{
11     std::vector <Video> peliculas;
12     public:
13     Pelicula(std::string i, std::string n, float d, std::string g, int c): Video(i, n, d, g, c){};
14 };
15
```

Construimos la Clase Película en la cual van heredar la clase de video ya que los parámetros del video son los mismos que necesita la película . Esta clase tiene su propio vector privado.

```
Get Started  main.cpp  Pelicula.h  Serie.h  Video.h  Episodio.h
C Serie.h > ...
1  //Jennifer Avendaño Sánchez-A01656951
2  //Pastén Da Silva Julia - A01660665
3  #include <vector>
4  #include "Episodio.h"
5
6  class Serie: public Episodio{
7     std::vector <Video> series;
8     public:
9     Serie(std::string i, std::string n, float d, std::string g, int c): Episodio(i, n, d, g, c){};
10 };

```

Construimos la Clase Serie en la cual van heredar la clase de episodio ya que los parámetros del video son los mismos que necesita la serie . A diferencia del vector de película , el vector de serie se junta con el de episodio ya que lo necesita.

```
Get Started  main.cpp  Pelicula.h  Serie.h  Video.h  Episodio.h
C Episodio.h > ...
1  //Jennifer Avendaño Sánchez-A01656951
2  //Pastén Da Silva Julia - A01660665
3  #include <vector>
4
5  #ifndef _video_
6  #define _video_
7  #include "Video.h"
8  #endif
9
10 class Episodio: public Video{
11     public:
12     Episodio(std::string i, std::string n, float d, std::string g, int c): Video(i, n, d, g, c){};
13 };

```

Declaramos el episodio constructor como polimorfismo del video dándoles los mismos parámetros, que básicamente es unirlos.

```
Get Started  C Episodio.h  C main.cpp x  C Video.h  C Serie.h
main.cpp > ...
1  // Jennifer Avendaño Sánchez A01656951
2  // Julia Pasten Da Silva A01660665
3  #include <iostream>
4  #include <fstream>
5  #include <vector>
6  #ifndef _video_
7  #define _video_
8  #include "Video.h"
9  #endif
10
11 using namespace std;
12 //Películas
13 void cargarPelis(vector<Video> &);
14 //Episodio
15 void cargarSeries(vector<Video> &);
16 //Otras
17 void filVidDur(vector<Video>& vec);
18 void filVidGen(vector<Video>& vec);
19 void filVidCal(vector<Video>& vec);
20 void filVidSin(vector<Video>& vec);
21 void filEpi(vector<Video>& vec);
22 void mosVidS(vector<Video>& vec);
23 void calVids(vector<Video>& vec);
24 void mosVidP(vector<Video>& vec);
25 void calVidP(vector<Video>& vec);
26
27 int main(){
28     vector <Video> series;
29     cargarSeries(series);
30     vector <Video> peliculas;
31     cargarPelis(peliculas);
32
33     // Qué hacer
34     string hacer;
35     cout << "¿Qué quieres hacer?" << endl;
36     cout << "a. Ver películas o series" << endl;
37     cout << "b. Calificar películas o series" << endl;
38     cout << "c. Salir" << endl;
39     cin >> hacer;
40     cout << "-----" << endl;
41     while(hacer != "c"){
42         if (hacer=="a"){
43             // Busca película o serie
44             string buscar;
45             cout << "¿Qué quieres ver?" << endl;
46             cout << "a. Películas" << endl;
47             cout << "b. Series" << endl;
48             cin >> buscar;
```

Generamos una Clase video en el cual establecemos los atributos de nuestros objetos para que sean inicializados y guardados , con la finalidad de que sea el inicio para nuestra función y objetivo de escoger entre un filtro cual película o episodio de serie quiero ver o calificar. De igual manera es el formato de nuestro catálogo a exhibir. La función declaradas al inicio son para filtrar y para mostrar los episodios y películas.

```

Get Started  C main.cpp X  C Pelicula.h  C Serie.h  C tempCodeRunnerFile.h  C Video.h  C Episodio.h
C main.cpp > ...
1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4  #ifndef _video_
5  #define _video_
6  #include "Video.h"
7  #endif
8
9  using namespace std;
10 //Películas
11 void cargarPelis(vector<Video> &);
12 //Episodio
13 void cargarSeries(vector<Video> &);
14 //Otras
15 void filVidDur(vector<Video>& vec);
16 void filVidGen(vector<Video>& vec);
17 void filVidCal(vector<Video>& vec);
18 void filVidSin(vector<Video>& vec);
19 void mosVidS(vector<Video>& vec);
20 void calVids(vector<Video>& vec);
21
22 int main(){
23     vector <Video> series;
24     cargarSeries(series);
25     vector <Video> peliculas;
26     cargarPelis(peliculas);
27
28     // Qué hacer
29     string hacer;
30     cout << "¿Qué quieres hacer?" << endl;
31     cout << "a. Ver películas o series" << endl;
32     cout << "b. Calificar películas o series" << endl;
33     cout << "c. Salir" << endl;
34     cin >> hacer;
35     cout << "-----" << endl;
36     while(hacer != "c"){
37         if (hacer=="a"){
38             // Busca película o serie
39             string buscar;
40             cout << "¿Qué quieres ver?" << endl;
41             cout << "a. Películas" << endl;
42             cout << "b. Series" << endl;
43             cin >> buscar;
44             cout << "-----" << endl;
45             if (buscar=="a"){
46                 // Busca por un filtro
47                 string filtro;
48                 cout << "Filtrar en orden por:" << endl;
49                 cout << "a. Duración" << endl;
50                 cout << "b. Género" << endl;
51                 cout << "c. Calificación" << endl;
52                 cout << "d. Sin orden" << endl;
53                 cin >> filtro;
54                 cout << "-----" << endl;
55                 if (filtro=="a"){filVidDur(peliculas);}
56                 else if (filtro=="b"){filVidGen(peliculas);}
57                 else if (filtro=="c"){filVidCal(peliculas);}
58                 else if (filtro=="d"){filVidSin(peliculas);}
59                 else{cout<<"Valor inválido, reingrese valor"<<endl;}
60             }
61             else if(buscar=="b"){
62                 // Busca por un filtro

```

Definimos un menú alternativo donde el usuario le indicará al programa, que es lo que quiere hacer; ver películas o series, calificar películas o series. A Partir de lo que le indique el usuario lo llevará al menú de dicha elección que al seleccionar su elección correctamente el código ejecutará la función establecida.



```

69         cout << "c. Calificación" << endl;
70         cout << "d. Sin orden" << endl;
71         cout << "-----" << endl;
72         cin >> filtro;
73
74         if (filtro=="a"){filVidDur(series);}
75         else if (filtro=="b"){filVidGen(series);}
76         else if (filtro=="c"){filVidCal(series);}
77         else if (filtro=="d"){filVidSin(series);}
78         else{cout<<"Valor inválido, reingrese valor"<<endl;}
79     }
80     cout<<endl;
81     cout << "¿Qué quieres hacer?" << endl;
82     cout << "a. Ver películas o series" << endl;
83     cout << "b. Calificar películas o series" << endl;
84     cout << "c. Salir" << endl;
85     cin >> hacer;
86     cout << "-----" << endl;
87
88 }
89 else if(hacer=="b") {
90     // Busca película o serie
91     string buscar;
92     cout << "¿Qué quieres calificar?" << endl;
93     cout << "a. Películas" << endl;
94     cout << "b. Series" << endl;
95     cout << "-----" << endl;
96     cin >> buscar;
97     if (buscar=="a"){
98         mosVidS(películas);
99         calVids(películas);
100     }
101     else if(buscar=="b"){
102         mosVidS(series);
103         calVids(series);
104     }
105     cout<<endl;
106     cout << "¿Qué quieres hacer?" << endl;
107     cout << "a. Ver películas o series" << endl;
108     cout << "b. Calificar películas o series" << endl;
109     cout << "c. Salir" << endl;
110     cin >> hacer;
111     cout << "-----" << endl;
112 }
113 }
114 if(hacer=="c"){
115     cout<<"bye"<<endl;

```

Con la función void cargarPelis(vector<Video> &), void cargarSeries(vector<Video> &) se declaran los vectores para series y películas, además a esto se utiliza la función ifstream MyReadEpi ("Episodios.txt") y ifstream MyReadPeli("Película.txt") para cargar el archivo .txt de episodios y películas.

Con la palabra hacer entramos en un ciclo de condicionales donde solamente se hace lo que el usuario gusta realizar, y para salir le damos al usuario la opción de que rompa el ciclo con "c". Este ciclo se realiza tanto para películas como para episodios, lo que significa que dependiendo de cual el usuario elija, mostrará sus datos y calificación previamente establecida.

```
Get Started  C Episodio.h  G+ main.cpp X  C Video.h  C Serie.h
G+ main.cpp > ...
156 void filVidGen(vector<Video>& vec){
157     std::cout<< "Estos son los géneros: ";
158     std::cout<<std::endl;
159     int tam =vec.size();
160     for (int i=0;i<tam;i++){
161         if(vec[i].getGenero()!= vec[i+1].getGenero()){
162             std::cout<< vec[i].getGenero()<<std::endl;
163         }
164     }
165     string gen;
166     std::cout << "Dame el género que buscas: " << std::endl;
167     cin >> gen;
168     for(Video v : vec){
169         if(v.getGenero() == gen){
170             v.toStr();
171         }
172     }
173 };
174 void filVidCal(vector<Video>& vec){
175     int cal;
176     std::cout << "Dame la calificación que buscas: " << std::endl;
177     cin >> cal;
178     for(Video v : vec){
179         if(v.getCalificacion() >= cal){
180             v.toStr();
181         }
182     }
183 };
184 void filVidSin(vector<Video>& vec){
185     for(Video v : vec){
186         v.toStr();
187     }
188 };
189 void filEpi(vector<Video>& vec){
190     cout<<endl;
191     //Reciber Id de clase
192     string idinci;
193     cout <<"Dame el Id de la serie cuyos datos quieres filtrar"<<endl;
194     cin>>idinci;
195
196     //Filtra
197     cout<<endl;
198     string filtro;
199     cout << "Filtrar en orden por:" << endl;
200     cout << "a. Duración" <<endl;
201     cout << "b. Género" <<endl;
202     cout << "c. Calificación" <<endl;
```

La calificación se separo entre serie y película. Con las variables vacías de “string myEpi” y “string myPeli” nos permite guardarr la información de los parámetros de cada título en las clases de vídeo. En el apartado de separación de datos (donde empieza el while) tanto de Películas como de Episodios , básicamente estamos ordenando nuestro catálogo descartando los guiones para mantenerlo de manera limpia y ordenada en la terminal, ubicando las posiciones de los datos del título que son; id, nombre, duración, género y clasificación, igual le ordenamos convertir los string a float.

Para los ingresos de valores de video estamos declarando la clase video como video\_temp que le estamos ordenando los parámetros a como vamos ordenar nuestro constructor de video y al darle el push\_back le damos la orden para inicializar el vector.

```

Get Started  main.cpp x Pelicula.h Serie.h tempCodeRunnerFile.h
main.cpp > ...
182 };
183 void filVidCal(vector<Video>& vec){
184     int cal;
185     std::cout << "Dame la calificación que buscas: " << std::endl;
186     cin >> cal;
187     for(Video v : vec){
188         if(v.getCalificacion() >= cal){
189             v.toStr();
190         }
191     }
192 };
193 void filVidSin(vector<Video>& vec){
194     for(Video v : vec){
195         v.toStr();
196     }
197 };
198 void mosVidS(vector<Video>& vec){
199     for(Video v : vec){
200         v.toStrS();
201     }
202 }
203 void calVids(vector<Video>& vec){
204     // Da el id del objeto que quiere calificar
205     cout<<endl;
206     string seleccion;
207     cout << "Ingresa el id para calificar: ";
208     cin >> seleccion;
209     int tam =vec.size();
210     for (int i=0;i<tam;i++){
211         if(vec[i].getId()== seleccion){
212             std::cout<< "Id enocntrado, su título es: "<<vec[i].getNombre()<<std::endl;
213             //Calificar el id
214             int califica;
215             cout << "¿Cuál es tu calificación para este título?(en entero): ";
216             cin >> califica;
217             vec[i].setCalificacion(califica);
218         }
219     }
220 }

```

Aquí empezamos con buscar el título que el usuario gusta calificar y lo hace por medio de un filtro dependiendo del parámetro que el usuario ingrese, y le mostraremos el catálogo de este filtro con la ayuda de la con fusión calVids. Posteriormente a esto en vec.size leemos el

tamaño del vector , llegando al if que aquí lo único que hacemos es adentrarnos en el vector del id que el usuario seleccionó , para poder calificarlo y obtenerlo con el set.

## Ejecución del Código

```
5 warnings generated.
¿Qué quieres hacer?
a. Ver películas o series
b. Calificar películas o series
c. Salir
a
-----
¿Qué quieres ver?
a. Películas
b. Series
a
-----
Filtrar en orden por:
a. Duración
b. Género
c. Calificación
d. Sin orden
c
-----
Dame la calificación que buscas:
4
*****
El id: 1
El nombre: Harry Potter y la piedra Filosofal
La duración: 2.32
El género: Fantasía/Aventura
La calificación: 5
*****
El id: 3
El nombre: Harry Potter y el prisionero de Azkaban
La duración: 2.22
El género: Fantasía/Aventura
La calificación: 4
*****
El id: 4
El nombre: Harry Potter y el caliz de fuego
La duración: 2.37
El género: Fantasía/Aventura
La calificación: 5
*****
El id: 5
El nombre: Harry Potter y la Orden del Fenix
La duración: 2.18
El género: Fantasía/Aventura
La calificación: 4
*****
El id: 7
El nombre: Harry Potter y las Reliquias de la Muerte
La duración: 2.27
El género: Fantasía/Aventura
La calificación: 5
*****
El id: 8
El nombre: Harry Potter y las Reliquias de la Muerte2
La duración: 2.1
El género: Fantasía/Aventura
La calificación: 5
*****
El id: 9
El nombre: Animales fantásticos y dónde encontrarlos
La duración: 2.16
El género: Fantasía/Aventura
La calificación: 4
*****
El id: 10
El nombre: Animales fantásticos: los crímenes de Grindelwald
La duración: 2.14
El género: Fantasía/Aventura
La calificación: 4
*****
El id: 14
El nombre: Crepusculo:Ecclipse
La duración: 2.04
El género: Romance/Fantasía
La calificación: 4
*****
El id: 15
El nombre: Crepusculo:Amanecer
La duración: 1.57
El género: Romance/Fantasía
La calificación: 5
*****
El id: 16
El nombre: Crepusculo:Amanecer2
La duración: 1.55
El género: Romance/Fantasía
La calificación: 5
*****
El id: 17
El nombre: A todos los chicos de los que me enamore
La duración: 1.4
El género: Comedia Romantica
La calificación: 5
*****
El id: 18
El nombre: A todos los chicos de los que me enamore 2
La duración: 1.44
El género: Comedia Romantica
La calificación: 5
*****
El id: 19
El nombre: A todos los chicos de los que me enamore 3
La duración: 1.5
El género: Comedia Romantica
La calificación: 5
*****
El id: 20
El nombre: El stand de los besos
La duración: 1.46
El género: Comedia Romantica
La calificación: 5
*****
¿Qué quieres hacer?
a. Ver películas o series
b. Calificar películas o series
c. Salir
```

```

Ingresa el id para calificar: T04S01C03
Id encontrado, su título es: Recursos Humanos:Mamis malas
¿Cuál es tu calificación para este título?(en entero): 5
*****
El id: T01S01C01
El nombre: Euphoria:Piloto
La calificación: 5
*****
El id: T01S01C02
El nombre: Euphoria:Stuntin Like My Daddy
La calificación: 5
*****
El id: T01S01C03
El nombre: Euphoria:Made You Look
La calificación: 5
*****
El id: T01S01C04
El nombre: Euphoria:Shock One
La calificación: 5
*****
El id: T01S01C05
El nombre: Euphoria:03 Bonnie and Clyde
La calificación: 5
*****
El id: T01S01C06
El nombre: Euphoria:The Next Episode
La calificación: 5
*****
El id: T01S01C07
El nombre: Euphoria:The Trials and Tribulations of Trying to Pee While Depressed
La calificación: 5
*****
El id: T01S01C08
El nombre: Euphoria:And Salt the Earth Behind You
La calificación: 5
*****
El id: T01S02C01
El nombre: Euphoria:Trying to Get to Heaven Before They Close the Door
La calificación: 5
*****
El id: T01S02C02
El nombre: Euphoria:Out of Touch
La calificación: 5
*****
El id: T01S02C03
El nombre: Euphoria:Ruminations: Big and Little Bulls
La calificación: 5
*****
El id: T01S02C04
El nombre: Euphoria:You Who Cannot See, Think of Those Who Can
La calificación: 5
*****
El id: T01S02C05
El nombre: Euphoria:Stand Still Like the Hummingbird
La calificación: 5
*****
El id: T01S02C06
El nombre: Euphoria:A Thousand Little Trees of Blood
La calificación: 5

```

```

El id: T01S02C07
El nombre: Euphoria:The Theater and It's Double
La calificación: 5
*****
El id: T01S02C08
El nombre: Euphoria:All My Life, My Heart Has Yearned for a Thing I Cannot Name
La calificación: 5
*****
El id: T02S01C01
El nombre: Los Juegos del Calamar:Luz Roja,Luz Verde
La calificación: 4
*****
El id: T02S01C02
El nombre: Los Juegos del Calamar:Infierno
La calificación: 4
*****
El id: T02S01C03
El nombre: Los Juegos del Calamar:El hombre del paraguas
La calificación: 4
*****
El id: T02S01C04
El nombre: Los Juegos del Calamar:No abandones el equipo
La calificación: 4
*****
El id: T02S01C05
El nombre: Los Juegos del Calamar:Un mundo justo
La calificación: 4
*****
El id: T02S01C06
El nombre: Los Juegos del Calamar:Gganbu
La calificación: 4
*****
El id: T02S01C07
El nombre: Los Juegos del Calamar:Los VIP
La calificación: 4
*****
El id: T02S01C08
El nombre: Los Juegos del Calamar:El lider
La calificación: 4
*****
El id: T02S01C09
El nombre: Los Juegos del Calamar:Un día de suerte
La calificación: 4
*****
El id: T03S01C01
El nombre: Destino La Saga Winx:Con un Hada , de la mano
La calificación: 1
*****
El id: T03S01C02
El nombre: Destino La Saga Winx:Aqui no hay extraños
La calificación: 2
*****
El id: T03S01C03
El nombre: Destino La Saga Winx:Pesadas esperanzas mortales
La calificación: 2
*****
El id: T03S01C04
El nombre: Destino La Saga Winx:Angeles caidos
La calificación: 3

```

```

El id: T03S01C04
El nombre: Destino La Saga Winx:Angeles caidos
La calificación: 3
*****
El id: T03S01C05
El nombre: Destino La Saga Winx:Marchitarse en la verdad
La calificación: 3
*****
El id: T03S01C06
El nombre: Destino La Saga Winx:Un corazon fanatico
La calificación: 3
*****
El id: T04S01C01
El nombre: Recursos Humanos:El parto
La calificación: 4
*****
El id: T04S01C02
El nombre: Recursos Humanos:Día de la capacitación
La calificación: 3
*****
El id: T04S01C03
El nombre: Recursos Humanos:Mamis malas
La calificación: 5
*****
El id: T04S01C04
El nombre: Recursos Humanos:Rutgers, un paraíso de amor
La calificación: 4
*****
El id: T04S01C05
El nombre: Recursos Humanos:El amor en los tiempos de postparto
La calificación: 4
*****
El id: T04S01C06
El nombre: Recursos Humanos:El angel de la adicción
La calificación: 3
*****
El id: T04S01C07
El nombre: Recursos Humanos:Convencion internacional de criaturas
La calificación: 3
*****
El id: T04S01C08
El nombre: Recursos Humanos:La luz
La calificación: 3
*****
El id: T04S01C09
El nombre: Recursos Humanos:El fin se acerca
La calificación: 3
*****
El id: T04S01C10
El nombre: Recursos Humanos:Tormenta de M*erda
La calificación: 3

```

¿Qué quieres hacer?

- Ver películas o series
- Calificar películas o series
- Salir

c

-----

bye

(base) iulscascelli@MacBook-Air-de-Julia proyecto 5 %

## Conclusión Individual

Jennifer

Se practicó toda la teoría de Herencia y Polimorfismo vista en clase llegando a la demostración de un código base de streaming en el cual el usuario pueda no solo ver el catálogo sino aportar a su calificación llegando a la conclusión de que realizamos el reto de POO con éxito, ratificando nuestro aprendizaje y nivel del curso.

Julia

Se pudo llegar a la meta de realizar un mini servicio de streaming con las funciones básicas, lo cual nos hace llegar a la conclusión de que al retornos en aplicar nuestros conocimientos obtenidos del curso en este proyecto sugestionando y evaluando nuestro trabajo como eficiente , obteniendo la certeza de que estamos a un buen nivel de aplicación de Herencia y Polimorfismo.