



Instituto Tecnológico y de Estudios Superiores de Monterrey.

Campus Estado de México.

Programación orientada a objetos gpo 301

Profesores:

Antonio Víctor Mejía Olvera

Proyecto integrador

Alberto Limón Cancino A01749746

Fecha de entrega:

18 - Junio - 2023

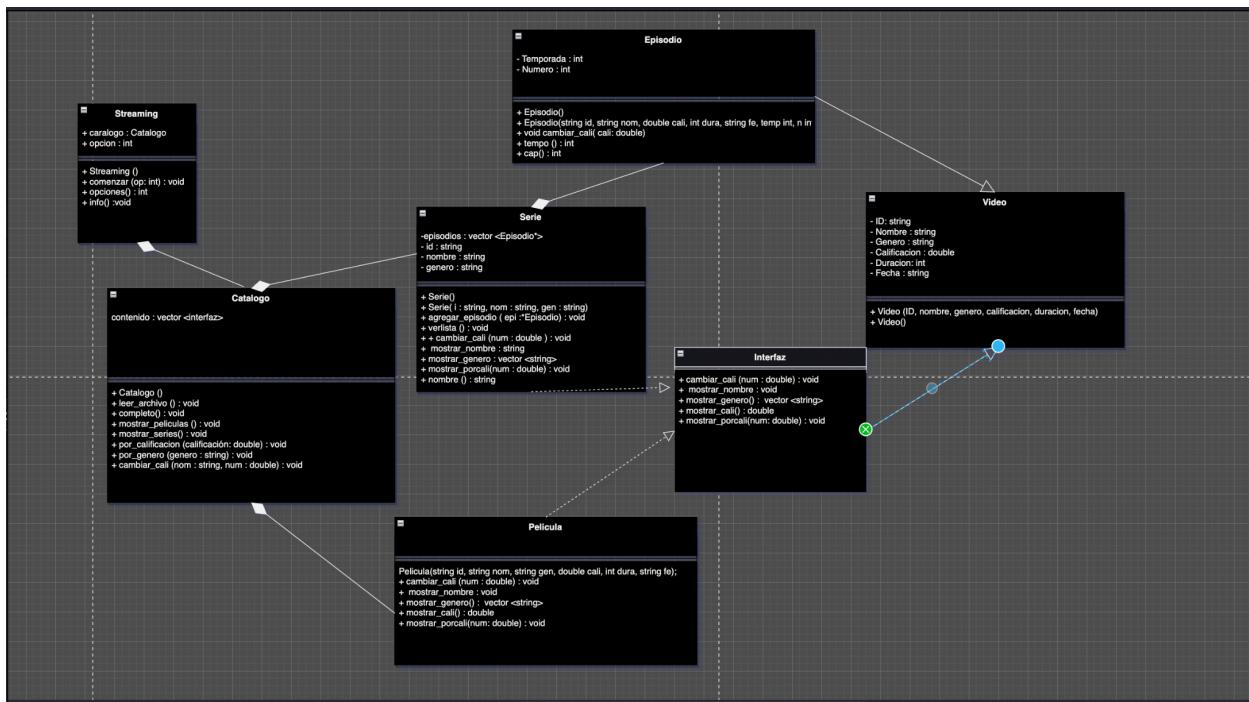
“En esta actividad me comprometo a aplicar mis conocimientos, a esforzarme en su desarrollo y a no servirme de medios no autorizados o ilícitos para realizarla.”

Introducción

A continuación se nos presentó el reto de desarrollar un sistema de streaming similar a las plataformas conocidas como Netflix, Prime, Paramount, entre otras. El objetivo del reto era desarrollar dicha plataforma haciendo uso de los temas vistos en clase como polimorfismo, herencia, agrupación, etc. La plataforma iba a funcionar con base en un catálogo proporcionado en csv con diversos contenidos entre películas y series. Cada serie tendría sus respectivos capítulos y la plataforma debía de ser capaz de resolver algunas tareas como la visualización del catálogo, y el uso de filtros de género y calificación. Finalmente, la plataforma también debía permitir el cambio de calificación en caso de ser pedido por el usuario.

UML

A continuación podremos observar el diagrama UML en donde podéis visualizar la planificación de la plataforma así como los método que se van a implementar para el funcionamiento correcto de la misma.



En el presente UML podemos ver la planificación como fue previamente mencionado. En él podemos observar el diseño y posteriormente una explicación del porqué de dicho diseño.

Empezamos con el servicio de streaming que funcionará como un paso anterior a la función principal, dicha clase nos permitirá desplegar las opciones de la plataforma al usuario así como llamar al catálogo donde podremos continuar con las demás funciones.

En catálogo podemos observar un vector con el contenido, dicho vector será llenado con apuntadores a la interfaz, dicha interfaz será explicada a continuación pero básicamente es una herramienta que nos permitirá guardar tanto series como películas en el mismo vector y manejarlas. Por otro lado, tenemos la

función que lee el archivo y genera dicho vector. Y finalmente, contamos con las funciones encargadas de llevar a cabo las tareas ofrecidas en el streaming para el usuario como ver el catálogo, únicamente las series, películas, filtros, entre otras.

Como previamente se mencionó se hizo uso de una interfaz que permitiera a las series y películas compartir métodos y tener una relación con la interfaz. Los métodos de la interfaz se encargan de obtener los datos para posteriormente utilizarlos dependiendo de la situación en la llamada del catálogo. La interfaz cuenta con métodos de cambiar la calificación, obtener los géneros, las calificaciones, cambiar la calificación, etc. El verdadero propósito de la interfaz fue que al momento de llamar funciones del vector contenido este pudiera ver si era una serie o una película y tomar el camino correcto.

Asimismo, tenemos las clases de Serie y Película que como previamente se mencionó utilizan los métodos de la interfaz; sin embargo, Serie cuenta con otros métodos que permiten guardar los episodios así como manipularlos directamente para cambiar calificaciones.

Con base en el archivo con el catálogo se observó que tanto Película como Episodio contaban con los mismos atributos por lo que se optó por crear una clase Video que permitiera heredar a ellos dichas variables. Dicha clase cuenta únicamente con sus variables que hereda a película y episodio.

Ejemplos de ejecución

Bienvenido a Netflix Selecciona una de las siguientes opciones 1. Ver Catalogo completo 2. Ver películas 3. Ver series 4. Filtrar por calificacion 5. Filtrar por genero 6. Cambiar una calificacion 7. Ver episodios de una serie 8. Salir Ingresa un número: 1										
ID	Nombre	Duracion	Género	Calificación	Estreno	Nombre Episodio	Temporada	Num. Episodio	ID Episodio	
tt0944947	Game Of Thrones	57	Acción/Aventura/Drama/Fantasy/Romance	9	04/17/2011	Winter Is Coming	1	1	tt1668899	
tt0944947	Game Of Thrones	57	Acción/Aventura/Drama/Fantasy/Romance	8.8	04/02/11	The Kingsroad	1	2	tt1668746	
tt0944947	Game Of Thrones	57	Acción/Aventura/Drama/Fantasy/Romance	8.7	05/01/11	Lord Snow	1	3	tt1829962	
tt0944947	Game Of Thrones	57	Acción/Aventura/Drama/Fantasy/Romance	8.8	05/08/11	Cripples Bastards And Broken Things	1	4	tt1829969	
63										
tt0944947	Game Of Thrones	57	Acción/Aventura/Drama/Fantasy/Romance	9.1	05/12/11	The Wolf And The Lion	1	5	tt1829964	
tt0944947	Game Of Thrones	57	Acción/Aventura/Drama/Fantasy/Romance	9.2	5/22/2011	A Golden Crown	1	6	tt1837862	
tt0944947	Game Of Thrones	57	Acción/Aventura/Drama/Fantasy/Romance	9.2	5/29/2011	You Win Or You Die	1	7	tt1837863	
tt0944947	Game Of Thrones	57	Acción/Aventura/Drama/Fantasy/Romance	9	05/06/11	The Pointy End	1	8	tt1837864	
tt0944947	Game Of Thrones	57	Acción/Aventura/Drama/Fantasy/Romance	9.6	12/06/11	Baelor	1	9	tt1851398	

Bienvenido a Netflix Selecciona una de las siguientes opciones 1. Ver Catalogo completo 2. Ver películas 3. Ver series 4. Filtrar por calificacion 5. Filtrar por genero 6. Cambiar una calificacion 7. Ver episodios de una serie 8. Salir Ingresa un número: 2										
ID	Nombre	Duracion	Género	Calificación	Estreno	Nombre Episodio	Temporada	Num. Episodio	ID Episodio	
tt0107290	Jurassic Park	127	Acción/Aventura/Sci-Fi/Suspense	8.1	11/06/93					
tt0796366	Star Trek	127	Acción/Aventura/Sci-Fi	7.9	08/05/09					
tt0120762	Mulan	88	Animación/Aventura/Familiar/Fantasy/Musical/Guerra	7.6	6/19/1998					
tt0120131	The Lion King 2: Simba Pride	81	Animación/Aventura/Drama/Familiar/Musical/Romance	6.5	10/27/1998					
tt0078346	Superman	143	Acción/Aventura/Drama/Sci-Fi	7.3	12/15/1978					
tt0096895	Batman	126	Acción/Aventura	7.5	6/23/1989					
tt0848228	The Avengers	143	Acción/Aventura/Sci-Fi	8	04/05/12					
tt0415778	End Game	93	Acción/Crimen/Drama/Misterio/Suspense	5.1	7/21/2006					
tt1853728	Django Unchained	165	Drama/Western	8.4	12/25/2012					
tt1638002	Love Rosie	102	Comedia/Romance	7.2	10/22/2014					
tt0088763	Back To The Future	116	Aventura/Comedia/Sci-Fi	8.5	03/07/85					

Bienvenido a Netflix
Selecciona una de las siguientes opciones
1. Ver Catalogo completo
2. Ver peliculas
3. Ver series
4. Filtrar por calificacion
5. Filtrar por genero
6. Cambiar una calificacion
7. Ver episodios de una serie
8. Salir
Ingresa un número: 3
ID Nombre Duracion Genero Calificación Estreno Nombre Episodio Temporada Num. Episodio ID Episodio
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 9 04/17/2011 Winter Is Coming 1 1 tt1668899
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 8.8 04/02/11 The Kingsroad 1 2 tt1668746
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 8.7 05/01/11 Lord Snow 1 3 tt1829962
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 8.8 05/08/11 Cripples Bastards And Broken Things 1 4 tt18299
63
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 9.1 05/12/11 The Wolf And The Lion 1 5 tt1829964
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 9.2 5/22/2011 A Golden Crown 1 6 tt1837862
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 9.2 5/29/2011 You Win Or You Die 1 7 tt1837863
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 9 05/06/11 The Pointy End 1 8 tt1837864
tt0944947 Game Of Thrones 57 Action/Aventura/Drama/Fantasy/Romance 9.6 12/06/11 Baelor 1 9 tt1851398
tt0108778 Friends 22 Comedia/Romance 8.4 9/22/1994 The One Where Monica Gets A Roommate 1 1 tt0583459

Bienvenido a Netflix
Selecciona una de las siguientes opciones
1. Ver Catalogo completo
2. Ver peliculas
3. Ver series
4. Filtrar por calificacion
5. Filtrar por genero
6. Cambiar una calificacion
7. Ver episodios de una serie
8. Salir
Ingresa un número: 4
Ingresa la calificacion deseada:
8
ID Nombre Duracion Genero Calificación Estreno Nombre Episodio Temporada Num. Episodio ID Episodio
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 9 04/17/2011 Winter Is Coming 1 1 tt1668899
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 8.8 04/02/11 The Kingsroad 1 2 tt1668746
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 8.7 05/01/11 Lord Snow 1 3 tt1829962
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 8.8 05/08/11 Cripples Bastards And Broken Things 1 4 tt18299
63
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 9.1 05/12/11 The Wolf And The Lion 1 5 tt1829964
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 9.2 5/22/2011 A Golden Crown 1 6 tt1837862
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 9.2 5/29/2011 You Win Or You Die 1 7 tt1837863
tt0944947 Game Of Thrones 57 Accion/Aventura/Drama/Fantasy/Romance 9 05/06/11 The Pointy End 1 8 tt1837864

Bienvenido a Netflix
Selecciona una de las siguientes opciones
1. Ver Catalogo completo
2. Ver peliculas
3. Ver series
4. Filtrar por calificacion
5. Filtrar por genero
6. Cambiar una calificacion
7. Ver episodios de una serie
8. Salir
Ingresa un número: 5
Ingresa el genero deseado:
Tenemos: Accion/Aventura/Drama/Fantasy/Romance/Sci-Fi/Suspeno/Familiar/Musical/Guerra/Romance/Misterio/Crimen/Western/Comedia/Terror/Biografia/Animacion/Aventura
ID Nombre Duracion Genero Calificación Estreno Nombre Episodio Temporada Num. Episodio ID Episodio
tt0107290 Jurassic Park 127 Accion/Aventura/Sci-Fi/Suspeno 8.1 11/06/93
tt0078346 Superman 143 Accion/Aventura/Drama/Sci-Fi 7.3 12/15/1978
tt0848228 The Avengers 143 Accion/Aventura/Sci-Fi 8 04/05/12

Bienvenido a Netflix

Selecciona una de las siguientes opciones

1. Ver Catalogo completo
2. Ver peliculas
3. Ver series
4. Filtrar por calificacion
5. Filtrar por genero
6. Cambiar una calificacion
7. Ver episodios de una serie
8. Salir

Ingresá un número: 7

Ingresá la serie de la cual gustás ver los capítulos:

Friends

ID sodio ID Episodio	Nombre	Duracion	Genero	Calificación	Estreno	Nombre Episodio	Temporada	Num. Epi
tt0108778	Friends	22	Comedia/Romance	8.4	9/22/1994	The One Where Monica Gets A Roommate	1 1	tt0583459
tt0108778	Friends	22	Comedia/Romance	8.1	9/29/1994	The One With The Sonogram At The End	1 2	tt0583647
tt0108778	Friends	22	Comedia/Romance	8.2	06/10/94	The One With The Thumb	1 3	tt0583653

Bienvenido a Netflix

Selecciona una de las siguientes opciones

1. Ver Catalogo completo
2. Ver peliculas
3. Ver series
4. Filtrar por calificacion
5. Filtrar por genero
6. Cambiar una calificacion
7. Ver episodios de una serie
8. Salir

Ingresá un número: 6

Ingresá el título a cambiar:

Friends

Ingresá la nueva calificación: 8

- 1) The One Where Monica Gets A Roommate
- 2) The One With The Sonogram At The End
- 3) The One With The Thumb

A cuál capítulo deseas cambiarle la calificación ?

3

Calificación actualizada

Argumentación

Para el primer punto (lectura de archivo) se optó por una solución estándar en donde dividimos las variables por una referencia; en este caso una coma. Una solución sencilla de implementar y efectiva.

```
22 void Catalogo::leer_archivo() {
23     ifstream catalogo;
24     catalogo.open("/Users/betolimon/desktop/integrador-A01749746-1/BasePeliculas.csv");
25
26     if (!catalogo.is_open()) {
27         cout << "No se pudo abrir el archivo." << endl;
28         return;
29     }
30
31     string line;
32     while (getline(catalogo, line)) {
33         stringstream token(line);
34         string id;
35         string nombre;
36         string duracion;
37         string genero;
38         string calificacion;
39         string fecha;
40         string nom_episodio;
41         string temp;
42         string cap;
43         string id_episodio;
44
45         getline(token, id, ',');
46         getline(token, nombre, ',');
47         getline(token, duracion, ',');
48         getline(token, genero, ',');
49         getline(token, calificacion, ',');
50         getline(token, fecha, ',');
51         getline(token, id_episodio, ',');
52         getline(token, nom_episodio, ',');
53         getline(token, temp, ',');
54         getline(token, cap, ',');
55
56         try {
57             if (id_episodio == "") {
58                 Pelicula *p = new Pelicula(id, nombre, genero, stod(calificacion), stoi(duracion), fecha);
59                 contenido.push_back(p);
60             } else {
61
62                 Episodio *e = new Episodio(id_episodio, nom_episodio, stod(calificacion), stoi(duracion), fecha, stoi(temp), stoi(cap));
63                 bool x = false;
64                 for (int i = 0; i < contenido.size(); i++) {
65
66                     if (auto serie = dynamic_cast<Serie*>(contenido[i])) {
67                         if (contenido[i]->nom() == nombre) {
68                             serie->agregar_episodio(e);
69                             x = true;
70                         }
71                     }
72                 }
73
74                 if (x == false){
75
76                     Serie *s = new Serie(id, nombre, genero);
77                     s->agregar_episodio(e);
78                     contenido.push_back(s);
79
80                 }
81             }
82         } catch (...) {
83             cout << "Error atrapado " << endl;
84         }
85     }
86 }
```

Para el segundo punto tenemos el motivo principal del uso de una interfaz, en donde podemos mandar a llamar el mismo método para cualquier elemento del vector y este sabrá si deberá hacerlo con respecto al método definido en Serie o en Película.

```
127 void Catalogo::por_calificacion(double cali){
128
129     if ((cali> 10)or(cali<0)){
130         string s = "Favor de ingresar un numero del 0 al 10 ";
131         throw(s);
132     }
133
134
135     for (int i = 0; i< contenido.size();i++){
136         if (auto pelicula = dynamic_cast<Pelicula*>(contenido[i])){
137             if(pelicula->calificacion >= cali){
138                 pelicula->mostrar_nom();
139             }
140         }
141         else if (auto serie = dynamic_cast<Serie*>(contenido[i])){
142             serie->mostrar_porcali(cali);
143         }
144     }
145 }
146
147 }

148 void Catalogo::por_genero(string gen){
149
150     if (gen != "Aventura" && gen != "Accion" && gen != "Drama" && gen != "Fantasy" && gen != "Romance" && gen != "Sci-Fi" && gen != "Suspeno"
151         string s = "Genero invalido " + gen;
152         throw(s);
153     }
154
155
156     for (int i =0; i< contenido.size(); i++) {
157
158         for (int i = 0 ; contenido[i]->mostrar_gen().size(); i++){
159
160             if(contenido[i]->mostrar_gen()[i] == gen){
161                 contenido[i]->mostrar_nom();
162             }
163         }
164     }
165
166
167 }
168
169 }
```

Para el tercer punto se utilizaron únicamente métodos exclusivos de la serie, esto es debido a que de haberlos incluido en la interfaz estos deberían de haber sido implementados en película y esta clase no tiene ninguna relación con episodios de una serie.

```
117 void Catalogo::mostrar_episodios(string seri){
118
119     for (int i = 0; i< contenido.size();i++){
120         if (auto serie = dynamic_cast<Serie*>(contenido[i])){
121             serie->ver_lista(seri);
122         }
123     }
124
125 }
126
```

Para el punto final se optó por la misma solución de la interfaz en donde se decidirá si implementarlo en serie o en película dependiendo del contenido de ese vector en ese elemento. Exclusivamente para la serie dicho método es más específico y conlleva más métodos ya que de ahí debe preguntarle al usuario qué capítulo y la serie debe de buscar ese capítulo.

```
171 void Catalogo::cambiar_cali(string nom, double num){  
172     if ((num> 10)or(num<0)){  
173         string s = "Favor de ingresar un numero del 0 al 10 ";  
174         throw(s);  
175     }  
176  
177     bool x = false;  
178  
179     for (int i = 0; i< contenido.size();i++){  
180         if (contenido[i]->nom() == nom)  
181             x = true;  
182     }  
183  
184     if (x == false){  
185         string y= "Ingresa correctamente el nombre ";  
186         throw(y);  
187     }  
188  
189     for (int i = 0; i< contenido.size();i++){  
190         if (contenido[i]->nom() == nom){  
191             contenido[i]->cambiar_cali(num);  
192         }  
193     }  
194  
195 }  
196  
197 cout << "Calificacion actualizada" << endl;  
198  
199  
200 }
```

Conclusión

El programa presentado ofrece una solución al problema presentado. Se hizo uso de errores y ciclos que permitían que el usuario volviera a poner su deseo en caso de haber escrito algo mal. Asimismo, se cometen errores en caso de escribir un número cuando era una palabra y viceversa. Finalmente, se considera que el programa podría llegar a fallar si el orden el csv es incorrecto y no va de acorde a las variables en el programa. Asimismo, es importante tener cuidado con los apuntadores y no intentar llevar elementos a donde no es su lugar ya que podría existir un error en la memoria.

Con base en el proyecto realizado puedo concluir que fue una gran prueba para poner en práctica el conocimiento adquirido a lo largo del semestre. Asimismo, fue un proyecto retador en el que fue necesario llegar cada vez más lejos para su desarrollo completo. Se reforzaron los temas aprendidos e incluso se aprendieron nuevos al momento de encontrar problemas en el camino. Finalmente, considero el proyecto un éxito porque te fuerza a implementar el conocimiento de una manera directa y a encontrar diversas soluciones hasta llegar a la correcta, asimismo, divertido ya que conforme se avanza hay que cambiar cosas pasadas, agregar nuevos métodos e incluso eliminar algunos.