



**TECNOLÓGICO
DE MONTERREY®**

Instituto Tecnológico y de Estudios Superiores de Monterrey

Escuela de Ingeniería y Ciencias

Eva Denisse Vargas Sosa A01377098

Omar Rodrigo Sorchini Puente A01749389

Modelado de servicio de streaming

TC1030 Programación Orientada a Objetos

Grupo 308 y 309

Profesor:

Antonio Víctor Mejía Olvera

Junio 2020



**TECNOLÓGICO
DE MONTERREY®**

Índice

Situación Problema.....	3
Descripción de función del programa de Python.....	4
Ejecución del programa de Python.....	5
Argumentos.....	11
Casos en los que el programa dejaría de funciones.....	14
Conclusiones.....	15
Referencias.....	15



Situación Problema:

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. 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 series.

Todo video tiene:

- *ID*
- *Nombre Pelicula / Serie*
- *Género*
- *Calificación*
- *Duración*
- *Fecha Estreno*

Las series tienen episodios y cada episodio tiene:

- *ID*
- *Nombre*
- *Temporada a la que pertenece.*
- *Número de episodio*

*Nos interesa conocer la **calificación promedio** que ha recibido cada uno de los **videos**. Esta calificación está en escala de 1 a 10 donde 10 es la mejor calificación*



Descripción:

1. Se identificaron las clases a utilizar así como los atributos que le correspondía a cada una de ellas y se determinaron las relaciones respectivas tanto de composición como de herencia para cada una de estas como se muestra en la Imagen 1.
2. Una vez especificado lo anterior, procedimos a programarlo:
 - a. Se programa una clase abstracta de tipo de Video que esta clase la heredará la clase Episodio y Película
 - b. Se tiene que la clase MiCatalogo contiene un relación de composición tanto con Películas como para Series y a la vez la clase Serie está compuesta por los episodios, los cuales son clases de tipo Episodio
3. La forma en la que funciona el programa:
 - a. **Le muestra un menú al usuario y lo primero que tiene que hacer es cargar el archivo** con la opción como se muestra en la Imagen 2.
 - b. Una vez cargado el archivo, el usuario puede escoger que hacer a continuación dependiendo del menú que se le presenta
 - i. Si escoge la opción 2, este **desplegará los videos** que tiene, aquí hay tres opciones:
 1. **Calificación:** Donde se le pedirá al usuario que ingrese un valor numérico de los videos que desee filtrar arriba de número.
 2. **Género:** Se le mostrará al usuario los géneros disponibles y se le pedirá al usuario el género de los videos que desea ver.
 3. **Ambas:** Donde el usuario tendrá que ingresar el género y la calificación de los videos que desea filtrar.Esto se muestra detalladamente desde la Imagen 3 a la Imagen 5
 - ii. Si escoge la opción 3, el programa **mostrará las series** que tenga disponibles, el usuario tendrá que escoger uno de títulos mostrados y se desplegará la información de los episodios que de la serie. Imagen 6.
 - iii. Si escoge la opción 4, el programa **mostrará las películas** dependiendo de la calificación del usuario, esto es para filtrar las películas que tengan la calificación arriba de las que da el usuario. Imagen 7.
 - iv. Si escoge la opción 5, el programa le pedirá el usuario el tipo de video que desea **calificar** (película o episodio), después se le pedirá al usuario que ingrese la calificación que desea darle a ese video. Imagen 8 y 9.
 - v. Si escoge el 6, **saldrá del menú**, haciendo que el programa termine.

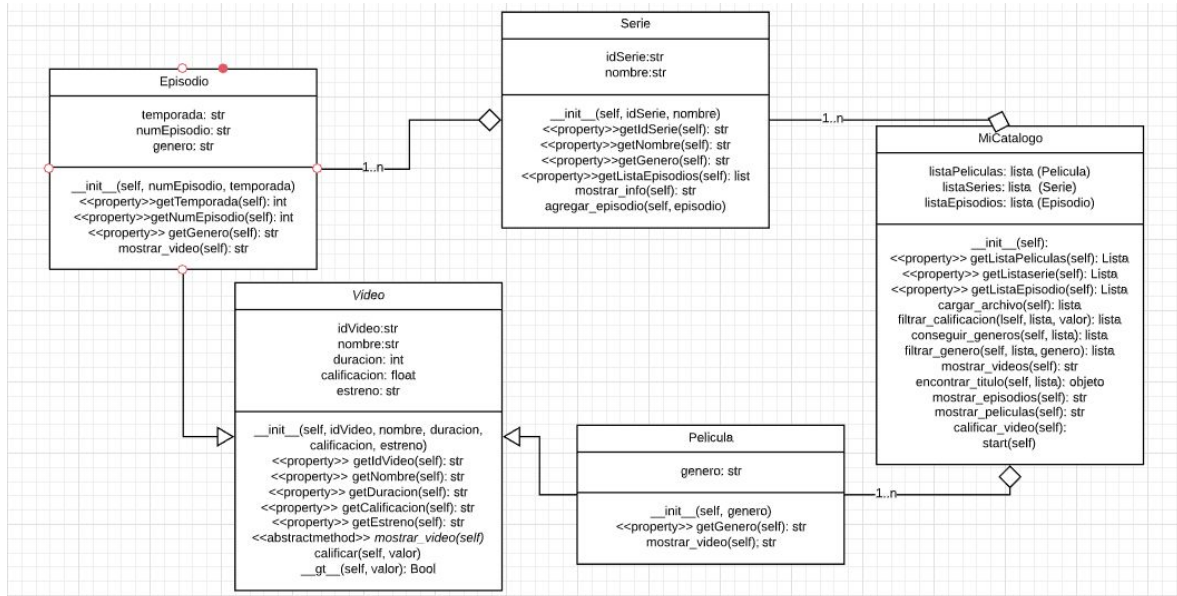


Imagen 1. Diagrama UML utilizado en este proyecto

Ejecución del programa de Python

```
>>> %Run MiCatalogo.py

Bienvenido a Ome Video

Escriba una de las siguientes numeros:

1 - Cargar catalogo
2 - Mostrar videos
3 - Mostrar episodios de una serie
4 - Mostrar peliculas
5.- Calificar video
6.- Salir

Qué es lo que desea ver a continuación:
1
La opcion fue 1
Catalogo cargado con éxito
```

Imagen 2. Ejecución del menú y selección de cargar catalogo



**TECNOLOGICO
DE MONTERREY®**

Escriba una de las siguientes numeros:

- 1 - Cargar catalogo
- 2 - Mostrar videos
- 3 - Mostrar episodios de una serie
- 4 - Mostrar peliculas
- 5.- Calificar video
- 6.- Salir

Qué es lo que desea ver a continuación:

2

La opcion fue 2

Como desea filtrar?

Por calificacion, genero o ambas?

calificacion

Arriba de que calificacion quiere los videos?: 9.5

Pelicula: Game of Thrones

Duracion: 57 minutos

Género: Action, Adventure, Drama, Fantasy, Romance

Calificacion: 9.6

Fecha de estreno: 06/12/2011

Imagen 3. Ejecución de mostrar videos de acuerdo a su calificación



TECNOLOGICO
DE MONTERREY®

Escriba una de las siguientes numeros:

- 1 - Cargar catalogo
- 2 - Mostrar videos
- 3 - Mostrar episodios de una serie
- 4 - Mostrar peliculas
- 5.- Calificar video
- 6.- Salir

Qué es lo que desea ver a continuación:

2

La opcion fue 2

Como desea filtrar?

Por calificacion, genero o ambas?

genero

Géneros disponibles

Action

Adventure

Sci-Fi

Animation

Family

Fantasy

Musical

War

Drama

Romance

Crime

Mystery

Thriller

Western

Comedia

Comedy

Biografía

Accion

Horror

Biography

Que genero quiere ver?:

War

Pelicula: Mulan

Duracion: 88 minutos

Género: Animation, Adventure, Family, Fantasy, Musical, War

Calificacion: 7.6

Fecha de estreno: 6/19/1998

Pelicula: 1917

Duracion: 119 minutos

Género: Drama, War

Calificacion: 8.3

Fecha de estreno: 10/1/2020

Imagen 4. Ejecución de mostrar videos de acuerdo a su género



**TECNOLÓGICO
DE MONTERREY®**

Qué es lo que desea ver a continuación:

2

La opción fue 2

Como desea filtrar?

Por calificación, género o ambas?

ambas

Arriba de que calificación quiere los videos?: **8**

Géneros disponibles

Action

Adventure

Sci-Fi

Animation

Family

Fantasy

Musical

War

Drama

Romance

Crime

Mystery

Thriller

Western

Comedia

Comedy

Biografía

Accion

Horror

Biography

Que género quiere ver?:

War

Película: 1917

Duración: 119 minutos

Género: Drama, War

Calificación: 8.3

Fecha de estreno: 10/1/2020

Imagen 5. Ejecución de mostrar videos de acuerdo a su calificación y género



**TECNOLOGICO
DE MONTERREY®**

Qué es lo que desea ver a continuación:

3

La opción fue 3

Serie: Game Of Thrones

Serie: Friends

Serie: Modern Family

Serie: The 100

Serie: The Simpsons

Serie: The Big Bang Theory

Serie: Memories Of The Alhambra

Serie: Death Note

Serie: Fate Zero

Serie: Unbreakable Kimmy Schmidt

Serie: Black Mirror

Serie: Chilling Adventures of Sabrina

Serie: Stranger Things

Serie: Maniac

Que serie quieres ver?: **Stranger Things**

Nombre: Chapter Six: The Monster

Duración: 51 minutos

Fecha de estreno: 7/15/2016

Calificación: 8.8

Genero: Drama, Fantasy, Horror, Mystery, Sci-Fi, Thriller

Temporada: 1

Numero de episodio: 6

Imagen 6. Ejecución de mostrar series.



**TECNOLÓGICO
DE MONTERREY®**



Imagen 7. Ejecución de mostrar películas de acuerdo a las calificación que da el usuario



**TECNOLÓGICO
DE MONTERREY®**

Qué es lo que desea ver a continuación:
5
La opción fue 5
Que desea calificar? (película/episodio): película

Película: Star Trek
Duración: 127 minutos
Género: Action, Adventure, Sci-Fi
Calificación: 7.9
Fecha de estreno: 8/5/2009

Película: Mulan
Duración: 88 minutos
Género: Animation, Adventure, Family, Fantasy, Musical, War
Calificación: 7.6
Fecha de estreno: 6/19/1998

Que título quieres calificar?: Pacific Rim
Que puntaje le das del 1 al 10?: 8
Su calificación ha sido subida con éxito

Imagen 8. Ejecución de calificar de acuerdo a las películas



**TECNOLOGICO
DE MONTERREY®**

Qué es lo que desea ver a continuación:

5

La opción fue 5

Que deseas calificar? (película/episodio): episodio

Serie: Game Of Thrones

Nombre: Winter Is Coming

Duración: 57 minutos

Fecha de estreno: 04/17/2011

Calificación: 9

Genero: Action, Adventure, Drama, Fantasy, Romance

Temporada: 1

Numero de episodio: 1

Nombre: The Kingsroad

Duración: 57 minutos

Fecha de estreno: 4/2/2011

Calificación: 8.8

Genero: Action, Adventure, Drama, Fantasy, Romance

Temporada: 1

Numero de episodio: 2

Serie: Maniac

Nombre: The Lake Of The Clouds

Duración: 10 minutos

Fecha de estreno: 9/21/2018

Calificación: 7.8

Genero: Comedy, Drama, Sci-Fi

Temporada: 1

Numero de episodio: 8

Que título quieres calificar?: The Lake Of The Clouds

Que puntaje le das del 1 al 10?: 4

Su calificación ha sido subida con éxito

Imagen 9. Ejecución de de calificar de acuerdo a los episodios



Argumentación:

- Clases

Todas las clases mantienen una relación, dada y principalmente se tiene composiciones y herencias entre las mismas, por ejemplo:

La clase **Video** es una clase abstracta ya que tiene métodos de tipo abstractos que en casos futuros utilizaremos para crear videos tangibles con su respectivo nombre lo cual sucederá a continuación con las clases **Película** y **Episodio**, las cuales son clases instanciadas no abstractas debido a la implementación del método *mostrar_video()*, método abstracto, en ambas clases que heredan directamente de **Video**.

Las demás clases tienen relaciones de tipo composición con las otras, teniendo así que la clase principal **MiCatalogo** está compuesta por clases de tipo **Película** y **Serie** que a su vez las clase de tipo **Serie** tiene objetos de tipo

Episodio contenidos dentro de sí.

Se optó por seguir una clase abstracta ya que al ser un catalogo de películas y series, necesitamos un comportamiento general a seguir por los videos de la plataforma pero que tuvieran un distinta forma de representación entre los mismos, lo cual fue razón suficiente para poder dejar uno de los métodos como abstractos, además como se puede apreciar, se crearon las relaciones con base al razonamiento de : un catálogo de vídeos tiene series y películas, pero las series tienen episodios.

- Herencia

Las clases **Película** y **Episodio** heredan de la superclase **Video** ya que al ser video una clase que cumple con todos los requisitos de un dado comportamiento de una película y un episodio (pues un episodio y una película son un vídeo, se realiza una herencia puesto que la clase abstracta **Video** fue diseñada para proporcionar el comportamiento general de lo que debería de hacer un vídeo.

- Modificadores de acceso

Todas las variables de clase que se tienen dentro de todas las clases, tienen un método de tipo 'getter' con el cual podemos regresar las variables de clase necesitadas de la clase que se especifica.

- Sobreescritura de métodos

__init__: Sobre cargamos este inicializador para poder agregar género a clase **Película** y a la clase **Episodio**.



calificar: Le pedimos al usuario que diera un título y la calificación que le daba al video y con base a esto se sobrescribe la calificación original del video.

__gt__: Sobrecargamos el método para que pudiera comparar la calificación de los vídeos regresando un Bool (True/False)

- Poliformismo

En este caso, las clases **Película** y **Serie** utilizan el polimorfismo para poder ser instanciadas como clases no abstractas ya que heredan el método abstracto *mostrar_video()*, por lo que ambas clases deben de implementarlo para que muestren su respectiva información en ambas clases y así poder utilizar dichas clases como objetos para **MiCatalogo**

- Clase Abstracta

Declaramos la clase **Video** como abstracta, porque necesitábamos una clase que tuviera un comportamiento muy general que pudiera ser heredado para las clases que contuviera un comportamiento de video con especificaciones propias de la nueva clase.

- Excepciones

Se implementaron desde el loop principal y de acuerdo a los *inputs* que pedíamos al usuario fuimos agregando errores que lanzaban excepciones conforme al tipo de error, teniendo así procesos a seguir conforme si son errores de tipo ValueError o Errores en general, cabe decir que estos errores no para el funcionamiento del programa.

Casos donde el programa deje de funcionar

Caso 1:

Debido a que la base de datos se carga de un archivo llamado BasePelículas el cual es un csv conteniendo toda la información separada por comas, se necesitará de dicho archivo para que el método *cargar_archivo()* pueda ser capaz de tener datos que permitan el correcto funcionamiento del programa pues sin este operaria con listas vacías y simplemente no funcionará como se planeó, aunque seguiría corriendo por las excepciones.

Caso 2:

Cargar una base de datos errónea no permitirá que el programa cargue la información por lo que los métodos tendrán errores que detendrán el proceso del programa.

Conclusiones:

Eva Denisse Vargas Sosa:

Este proyecto me ayudó a entenderlo más la programación orientada objetos, el semestre pasado si esta un poco perdida. Al principio se nos hizo algo complicado por la relación que tenía que tener cada clase, después de unas



cuantas asesorías, quedó un poco más claro el cómo debía funcionar, con mi compañero estuvimos hablando el funcionamiento de los métodos que se utilizaron en el programa. Tuvimos uno que otro problema con las excepciones porque no entendíamos cómo funcionaba ese tema, así que causaba algunos conflictos con el funcionamiento de nuestro menú principal, igual tuvimos que asistir a asesoría. A final de cuentas, si funciono el programa, realmente ayudó a reforzar los temas que vimos en clase.

Omar Rodrigo Sorchini Puente:

Mediante el desarrollo de este proyecto pudimos apreciar todos los métodos, convenciones y atributos propios que conlleva la programación orientada a objetos de manera que ahora podemos usar las relaciones de herencia y los conceptos de abstracción para poder generalizar objetos a necesitar que para proyectos a gran escala tales como un videojuego serán muy útiles puesto que cada objeto del juego tiene ciertos comportamientos a seguir, por lo que en mi caso de que voy para una concentración de videojuegos el haber aprendido la base de la POO fue de gran ayuda para desarrollarme profesionalmente.

Además, el uso de la herencia y del polimorfismo será una gran propiedad que podrá ser usada incluso en todo tipo de programas en donde se pueda usar una temática de objetos entre clases y con ellas los métodos a utilizar.

Referencias

- [1] Alvarez, M. (2014, 20 mayo). Polimorfismo en Programación Orientada a Objetos. Recuperado de <https://desarrolloweb.com/articulos/polimorfismo-programacion-orientada-objetos-concepto.html>
- [2] Ramirez, R. (2017, 27 enero). ¿Qué es variable de clase y una variable de instancia? Recuperado de <https://es.stackoverflow.com/questions/46454/qu%C3%A9-es-variable-de-clase-y-una-variable-de-instancia>
- [3] Miguel, P. E. (2016, 8 enero). UML (. Recuperado de <http://patriciaemiguel.com/poo/2016/01/08/uml.html>