

**Proyecto Integrador - Modelado de servicio de streaming  
Programación Orientada a Objetos (TC1030.3)**

Alumno:

Alejandro Daniel González Carrillo A01570396

Profesora:

Ma. Guadalupe Roque Díaz de León

© 2020 Derechos reservados: Ninguna parte de esta obra puede ser reproducida o transmitida, mediante ningún sistema o método, electrónico o mecánico, sin conocimiento por escrito del autor.

Monterrey, Nuevo León.  6 de junio de 2020.

Índice

Introducción…………………………………………………………………………………………3

Diagrama UML……………………………………………………………………………………...4

Ejemplos de Ejecución…………………………………………………………………………5-12

Argumentación…………………………………………………………………………………….13

Identificación de Casos Erróneos………………………………………………………………14

Conclusión Personal……………………………………………………………………………..14

Referencias Consultadas………………………………………………………………………..14

**Introducción (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. (Situación Problema, 2020)

**Instrucciones**

En este caso se nos a solicitado desarrollar una versión limitada de servicio de streaming para un futuro proveedor. Este servicio de streaming se compone de dos tipos de videos: películas y series.

* Todo video tiene un ID, un nombre, una duración y un género (drama, acción, misterio).
* Las series tienen episodios y cada episodio tiene un título y temporada a la que pertenece.
* Nos interesa conocer la calificación promedio que ha recibido cada uno de los videos.
* Esta calificación está en escala de 1 a 5 donde 5 es la mejor calificación.

El sistema debe ser capaz de:

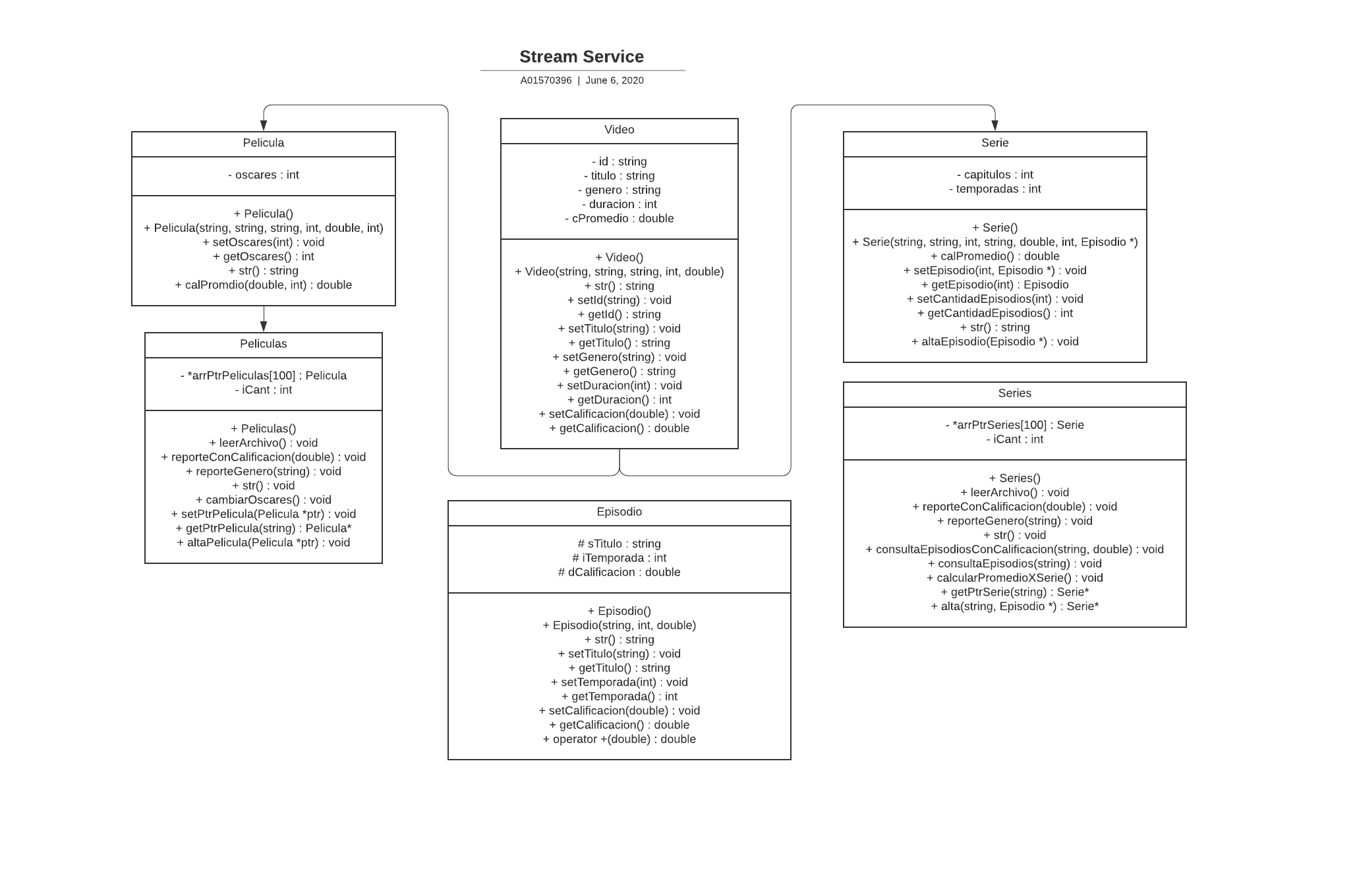
- Mostrar los episodios de una determinada serie con sus calificaciones.

- Mostrar las películas con sus calificaciones.

- Mostrar las películas con su género.

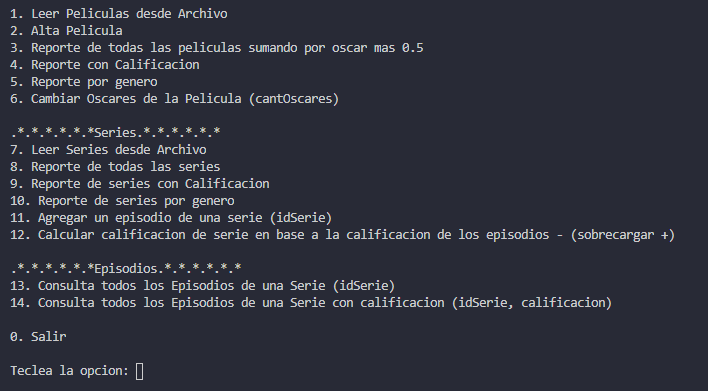
- Mostrar los episodios de una determinada serie con sus géneros

**Diagrama UML**

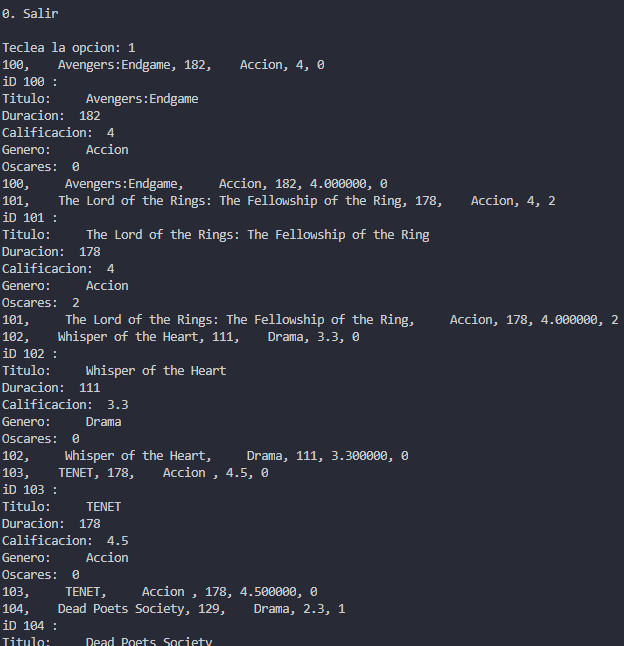


**Ejemplos de Ejecución**

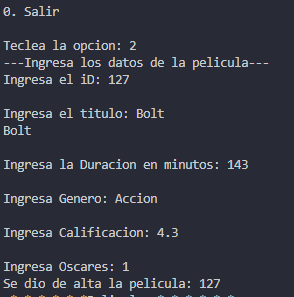
Menú al correr el código:



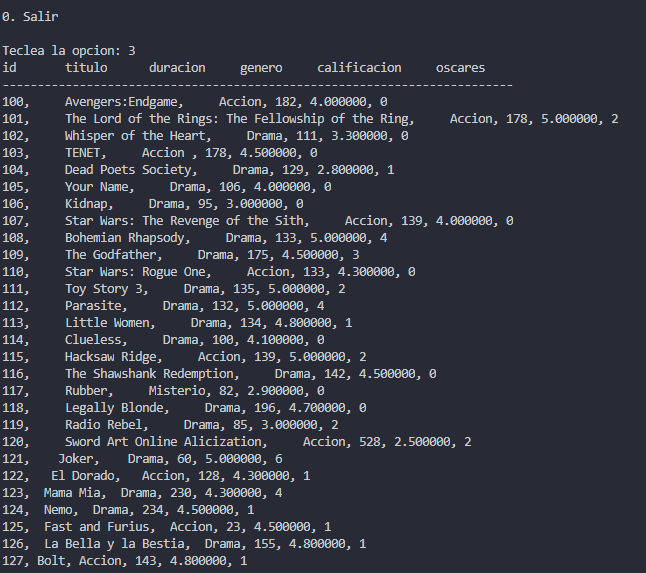
Comenzamos con la opción 1. Leer Películas desde archivo



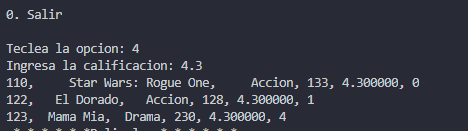
2. Alta de Película



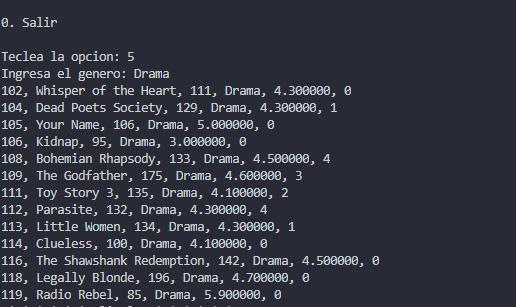
3. Reporte de todas las películas sumando por Oscar más 0.5



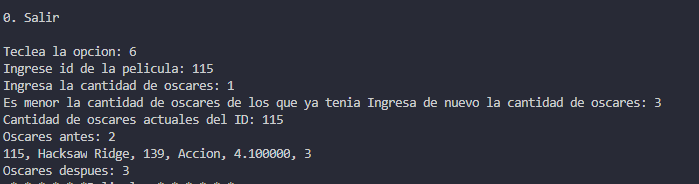
4. Reporte con Calificación (Es necesario haber cargado el punto 3 para que el valor de las calificaciones se actualizara).



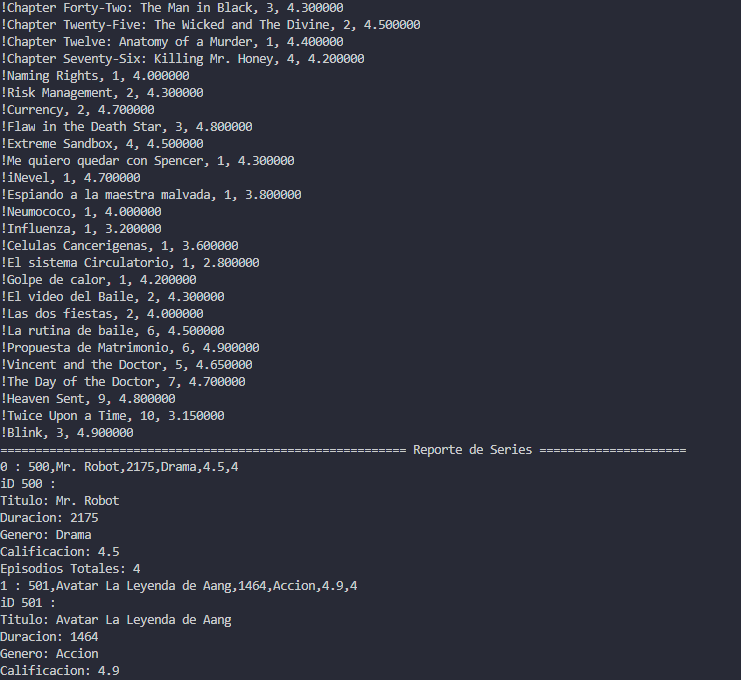
5. Reporte por genero (Es necesario haber cargado el punto 3 para que el valor de las calificaciones se actualice).



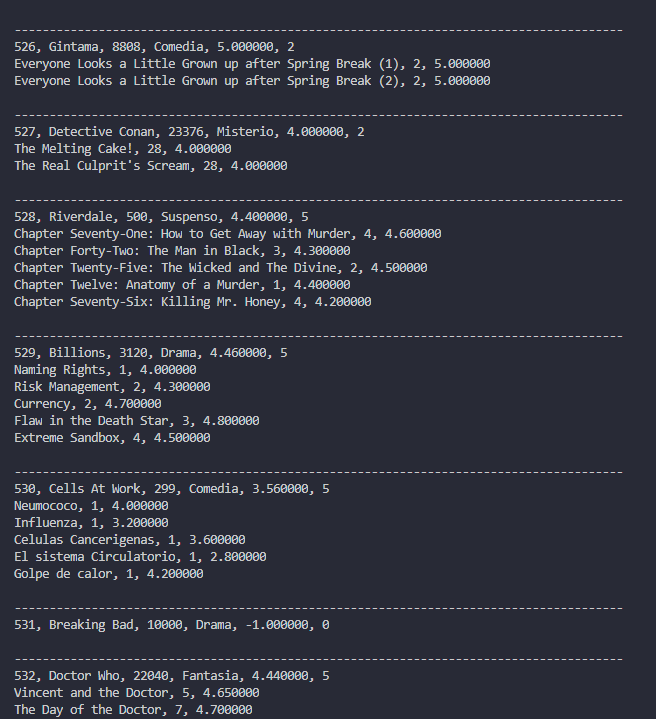
6. Cambiar Oscares de la Película (cantOscares) (Al hacer el cambio de Oscares no se verá reflejado el cambio de valor hasta que corramos el punto 3).



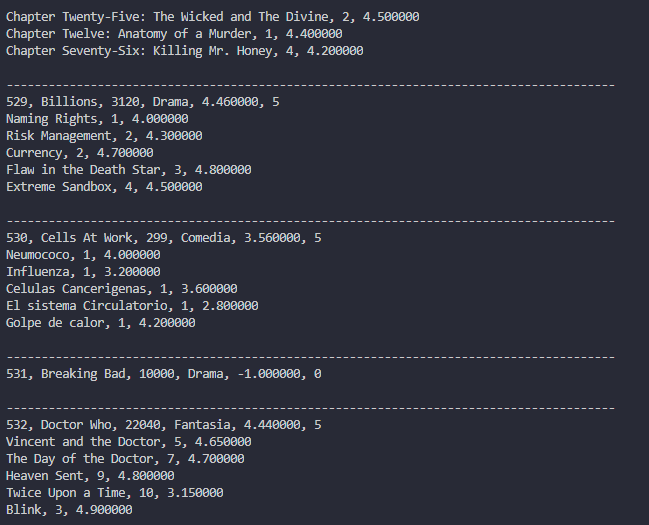
7. Leer Series desde Archivo



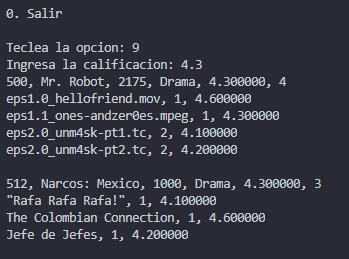
12. Calcular calificación de serie en base a la calificación de los episodios - (sobrecargar +) (LEER ANTES QUE TODOS LOS DEMAS PUNTOS DESPUES DEL 7 DEBIDO A QUE LA CALIFICACION NO SERA LA CORRECTA SINO SE EVALUE ESTE PUNTO PRIMERO).



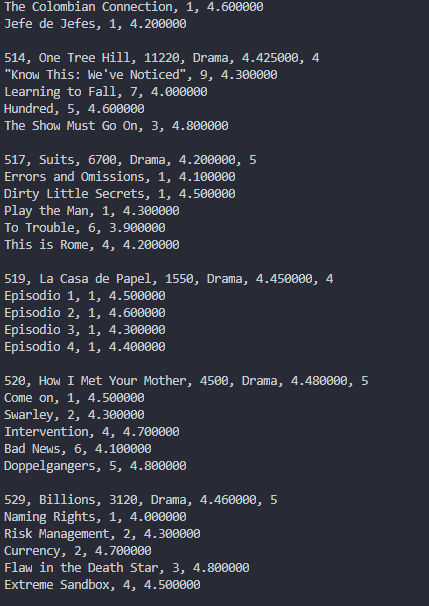
8. Reporte de todas las series (NECESARIO CORRER EL PUNTO 12)



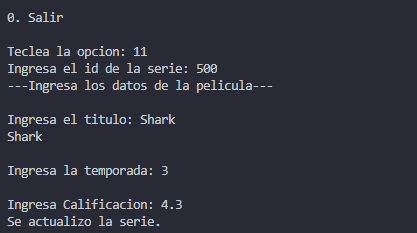
9. Reporte de series con calificación

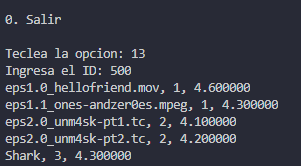


10. Reporte de series por genero (Drama, Accion, Comedia, Suspenso, Misterio, Fantasia)

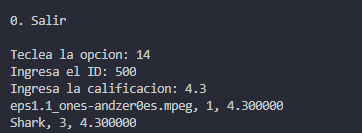


11. Agregar un episodio de una serie (idSerie)



13. Consulta todos los Episodios de una Serie (idSerie) 

14. Consulta todos los Episodios de una Serie con calificacion (idSerie, calificacion)



**Argumentación**

1. Si divide con el UML de manera ordenada cada uno de los puntos que requería, en caso de requerir alguna otra cosa además el dividirlo en esas clases podemos hacer una manera mas eficiente y ordenada de realizar el código.
2. Si al momento de pasarle los datos y ahorrar código película y series heredan las características de video.
3. En este caso considero que si se mantienen en privado todas las variables excepto las del video que se protegen y los métodos se dejan en publico para que puedan ser usados por las demás clases.
4. Si utilice bien la sobrecarga en este caso solo sobrecargue un operador el +. Se utilizo en la suma de calificaciones en los episodios. Al usarlo validamos correctamente la calificación de la serie.
5. Utilice polimorfismo al momento de llamar las variables de una clase en los métodos de otra clase.
6. Se comentaron casi todo el código, el main, las clases sus métodos y atributos. Y como funciona cada parte de cada método.
7. Se sobrecarga el operador +.

**Identificación de Casos Erróneos**

En el desarrollo del código se hicieron validaciones para cada una de las entradas del usuario. En tal caso de que crasheara pudiera ser que tratara de introducir algún otro valor corrupto dentro de los inputs. En este caso veo muy extraño que el usuario pueda corromper el código. Nomás que si pudiera llegar a introducir información incorrecta en el ID debido a que es un string.

**Conclusión Personal**

Durante estas 5 semanas logre comprender muchos temas de orientado a objetos como lo son Polimorfismo, el uso de pointers, sobrecarga de operadores, herencia, entre otras más. Estos temas los pude poner en practica en un proyecto el cual me ha retado del todo y me ha dado respuestas a muchas de mis dudas que tenía antes sobre orientado a objetos. Este proyecto me tomo alrededor de 1 día completo o 24 horas. Considero que lo que mas se me dificulto fueron los pointers y manejar las referencias, en general esa parte. De la misma manera el uso de métodos en los pointers me confundió un poco pero mientras más le dedicaba tiempo, más comprendía como funcionaban. Finalmente, me despido de esta materia con un gran sabor de boca definitivamente todos estos conocimientos sé que me servirán mucho más adelante en mi carrera y me han desarrollado más como un programador profesional.

**Referencias**

Roque, M. (2020). Programación Orientada a Objetos (TC1030.3). ITESM.

Situacion Problema. (2020). Recuperado el 6 de Junio del 2020 en

https://experiencia21.tec.mx/courses/23728/pages/situacion-problema