Etapa 1 del Reto. Investigación
Nombre de la unidad de formación: programación Orientada a objetos.
Fecha de entrega: 10 de junio de 2021
Estudiante:
Alejandro Ruiz García Rojas A01611451

Indice de contenidos:

Introducción:

Este proyecto consistió en la elaboración de un código que representaría una plataforma de streaming en donde se muestran varias series y películas que un usuario pueda ver, y muestre información relevante sobre ella, como su cantidad de capítulos, calificación de usuarios, nombre, duración, etc.

En el proyecto se busca que se muestre un uso conciso de sistema de clases, Herencia Polimorfismo, y conocimiento de ciclos para la elaboración de un menú que pueda dejar al usuario no solo consultar la información sino también calificar la película dependiendo de como fue su experiencia con ella.

A continuación, en las siguientes secciones mostraremos, como es que nosotros pudimos demostrar el conocimiento de estos conceptos, al igual que generar una propuesta de un programa que haga lo que se nos fue indicado, y le de la habilidad a un cliente de ver información sobre las Películas y series que existen en el sistema y las califique.

Código y Ejemplos de Ejecución:

A continuación, se presentarán imágenes en las que podremos ver la estructura que tiene el código aplicado para la elaboración de la situación problema planteada, con una pequeña explicación de su función, al igual que imágenes que nos muestra la ejecución correcta del código como tal.

En la siguiente imagen se muestra como esta declarada la clase principal Video, con sus respectivos constructores, al igual que sus métodos correspondientes y sus atributos. Puede ser observado que la función Muestra Datos es declarada como virtual, para que se pueda utilizar en clases que tengan herencia a esta, aplicando el respectivo polimorfismo.

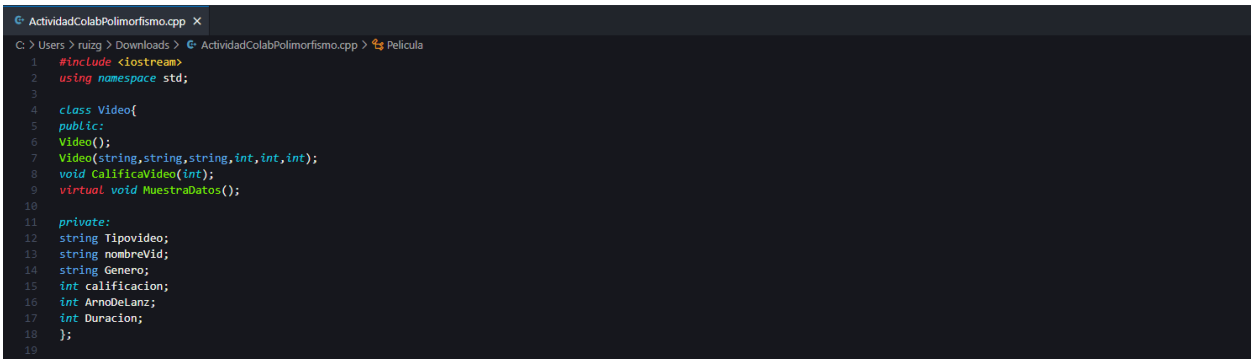


Ilustración 1:Declaracion de la clase principal con respectivos atributos y métodos.

La siguiente imagen nos muestra la asignación de nuestro constructor Video, designando que variables son respectivas a los diferentes componentes de el objeto instanciado, es decir, se le asignan los atributos respectivos al objeto con el constructor.

```
Video::Video(string _tipo,string _nombre,string _genero,int _calificacion,int _arno,int _duracion)
{
Tipovideo=_tipo;
nombreVid=_nombre;
Genero=_genero;
calificacion=_calificacion;
 ArnoDeLanz=_arno;
 Duracion=_duracion;

}

```

Ilustración 2: Constructor Video.

Podemos aquí ver la asignación de las funciones o métodos de la clase Video, y que es lo que harán. Estos fueron declarados fuera de la clase para mantener un orden y que el código fuera más limpio.

```
void Video::CalificaVideo(int _calificacion){
calificacion=_calificacion;
}
void Video::MuestraDatos(){
    cout<<"Video de tipo: "<<Tipovideo<<endl;
    cout<<"Nombre: "<<nombreVid<<endl;
    cout<<"Genero: "<<Genero<<endl;
    cout<<"Calificacion: "<<calificacion<<endl;
    cout<<"Duracion: "<<Duracion<<endl;
    cout<<"Arno de Lancamiento: "<<ArnoDeLanz<<endl;

}
```

Ilustración 3: Métodos de la clase Video.

Después se asigna la clase con herencia a la clase video, con el nombre de clase Serie, y se declaran sus respectivos métodos, constructor y atributos nuevos, que en este caso solo es su número de capítulos.

```
class Serie:public Video{
    public:
    Serie(string,string,string,int,int,int,int);
    void MuestraDatos();
    private:
    int capitulos;
};
```

Ilustración 4: Clase Hija Serie

Igualmente, se declaran los métodos de la clase al igual que el constructor fuera de esta por limpieza. Dado a que necesitamos de valores usados en clases anteriores, se especifica en la asignación de el constructor que este es caso, con los dos puntos y llamando a la clase video de la siguiente forma. De igual forma, al usar polimorfismo, simplemente llamamos a la función inicial de la clase Video, y añadimos un cout que nos muestra la cantidad de capítulos.

```
Serie::Serie(string _tipo,string _nombre,string _genero,int _calificacion,int _arno,int _duracion,int _capitulos):Video(_tipo,_nombre,_genero,_calificacion,_arno,_duracion){
    capitulos=_capitulos;
}

void Serie::MuestraDatos(){
    Video::MuestraDatos();
    cout<<"Capitulos: "<<capitulos<<endl;
}
```

Ilustración 5: Métodos de la clase Serie.

Se declara la clase Hija Película que tiene herencia de la clase Video, de la misma manera que se hace anteriormente. Se considera un nuevo atributo de nominaciones.

```
class Pelicula:public Video{
    public:
    Pelicula(string,string,string,int,int,int,string);
    void MuestraDatos();
    private:
    string nominaciones;
};
```

Ilustración 6: Clase Película.

Se asignan sus métodos y constructor.

En este caso no fue posible tomar una captura completa, entonces se insertó como texto el código:

```
Pelicula::Pelicula(string _tipo,string _nombre,string _genero, int _calificacion,
 int _arno, int _duracion, string _nominaciones):Video(_tipo,_nombre,_genero,_cal
ificacion,_arno,_duracion){
    nominaciones=_nominaciones;
}
void Pelicula::MuestraDatos(){
    Video::MuestraDatos();
    cout<<"La Pelicula es: "<<nominaciones<<endl;
}
```

Finalmente se declara la función main, en donde manejamos nuestro menú, y instanciamos nuestros objetos de la siguiente manera, ciclando todo con un ciclo while,

que busca hacer comparaciones del valor de elección para mostrar los respectivos videos o calificarlos dependiendo del valor ingresado por el usuario.

Esto se hizo de la siguiente manera:

En este caso no fue posible tomar una captura completa, entonces se insertó como texto el código:

```
int main(){
int decision;
    int elecionvid;
    decision=0;
    elecionvid=0;
    int contador=0;
    while(contador==0){
        cout<<"Hola Bienvienido. Oprima 1 para Calificar o  2 para Consular datos
."<<endl;
        cin>>decision;


        Pelicula Pelicula1("Pelicula","Capitan America: El primer vengador","Supe
rheroes",10,2011,124,"Oscar a Mejor actuaci□n");
        Pelicula Pelicula2("Pelicula","Enredados","Animacion",10,2010,100, "Oscar
 a Mejor animaci□n");

        Serie Serie1("Serie","Love, Death + Robots","Animacion para adultos",9,20
19,6-17,26);
        Serie Serie2("Serie", "El mundo oculto de Sabrina", "Terror", 9, 2018,50,
36);
        Serie Serie3("Serie","El mundo de Riley","Adolescente", 10, 2014, 22, 72)
;


        if (decision==1){
            cout<<"Presione 1 para calificar la Pelicula 1. Presione 2 para calif
icar la Pelicula 2. Presione 3 para calificar la serie 1. Presione 4 para califica
r la serie 2. Presione 5 para calificar la serie 3."<<endl;
            cin>>elecionvid;

            if(elecionvid==1){
                cout<<"Ingrese la Calificacion que desea asignarle al video."<<en
dl;

                int cali;
                cin>>cali;
                Pelicula1.CalificaVideo(cali);
                }

            if(elecionvid==2){
```

```
                cout<<"Ingrese la Calificacion que desea asignarle al video."<<en
dl;
                int cali;
                cin>>cali;
                Pelicula2.CalificaVideo(cali);
                }

            if(elecionvid==3){
                cout<<"Ingrese la Calificacion que desea asignarle al video."<<en
dl;
                int cali;
                cin>>cali;
                Serie1.CalificaVideo(cali);
                }

            if(elecionvid==4){
                cout<<"Ingrese la Calificacion que desea asignarle al video."<<en
dl;
                int cali;
                cin>>cali;
                Serie2.CalificaVideo(cali);
                }
            if(elecionvid==5){
                cout<<"Ingrese la Calificacion que desea asignarle al video."<<en
dl;
                int cali;
                cin>>cali;
                Serie3.CalificaVideo(cali);
                }

            }

        if (decision==2){
            cout<<"Presione 1 para consultar los datos de la pelicula 1. Presione
 2 para consultar los datos de la pelicula 2. Presione 3 para consultar los datos
 de la serie 1. Presione 4 para consultar los datos de la serie 2. Presione 5 par
a consultar los datos de la serie 3."<<endl;
            cin>>elecionvid;

            if(elecionvid==1){
                Pelicula1.MuestraDatos();
                }

            if(elecionvid==2){
                Pelicula2.MuestraDatos();
```

```
                }

            if(elecionvid==3){
                Serie1.MuestraDatos();
                }

            if(elecionvid==4){
                Serie2.MuestraDatos();
                }

             if(elecionvid==5){
                Serie3.MuestraDatos();
                }
            }



cout<<"Oprima 0 si desea hacer otro movimiento.Oprima 1 si este no es el caso.";
cin>>contador;
    }
}
```

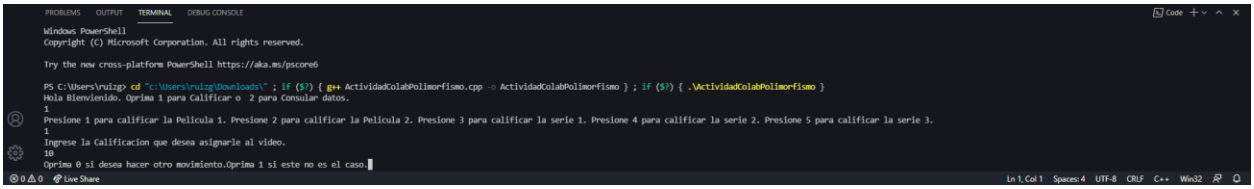
Finalmente, se presentan capturas de que el código funciona correctamente:

Confirmamos que existe la elección de calificar o Consultar datos.

El usuario tiene la elección de objeto que desea calificar.

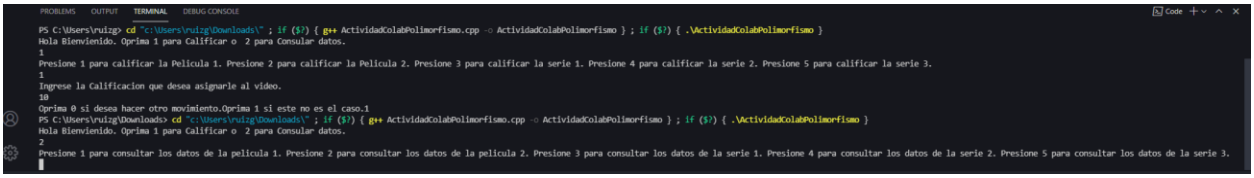
Se le pide al usuario insertar una calificación:

Se le pide al usuario si desea continuar o no.

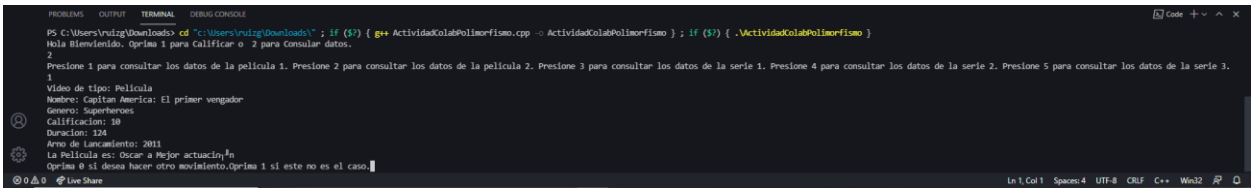


Ahora observamos que el código mostrara los datos respectivos a la película solicitada.

Se nos pide seleccionar el objeto a consultar:



Se nos muestra la información de la película solicitada, con la calificación actualizada de 10.



Argumentación:

¿Como se identifican la manera correcta de las clases a utilizar?

Las clases se usan correctamente en el programa, asignando cada una sus respectivos atributos y funciones. Existe la clase Video que es la principal, y sus clases hijas Película y Serie. En las clases se busca describir o “clasificar” a un objeto, denotando lo que lo hace ese objeto o lo diferencia de otros, y estableciendo su función. Después, en la función main, se busca instanciarlo, es decir crear un objeto que se del tipo de la clase asignada. En el programa podemos ver que esto ocurre. Declaramos lo que es que hace el video y lo que lo hace un video, y luego instanciamos o creamos un video después en el código que tiene esas mismas características y funciones.

¿Como se emplea el concepto de Herencia?

La herencia se utiliza mucho para este programa y de hecho diría que seria una de las partes mas importantes para que nuestro programa funcione. En este programa se crearon dos clases hijas de la clase video que son la clase Serie y Película, que ambos toman atributos de la clase principal Video. Estos son su nombre, su calificación, su duración, entre otros. La herencia busca utilizar y reusar atributos y métodos para no hacer una doble asignación de algo que sería prácticamente lo mismo. Es decir, se busca hacer el código más limpio y efectivo.

¿Como se utilizan los modificadores de acceso?

En este caso solo fue necesario usar los modificadores de acceso public y private, y no protected, aunque podrían ser implementados fácilmente. En el programa podemos ver que estos son usados apropiadamente, dado a que funciones y métodos que eran necesarios fuera de sus clases respectivas fueron usados, y todo aquello que no era necesario mostrar o hacer publico al usuario como por ejemplo los atributos de cada clase, se mantuvieron en privado, para que solo un administrador del sistema pudiera usarlos y consultarlos.

¿Como se empleó el concepto de Polimorfismo en el programa?

En el programa se aplicó polimorfismo para una función sencilla que era la función de mostrar datos. El polimorfismo es poder usar una función de mismo nombre dentro de varias clases, y como se muestra en el código, este es el caso. En la función principal se asigna la función de mostrar datos, y luego se uso en otras clases, agregándole información respectiva a ellas.

¿Como podríamos usar el concepto de sobrecarga en este programa?

Las indicaciones para la elaboración de este programa no señalaron el uso de sobrecarga de operadores, sin embargo si es que este fuera implementado, considero que sería de buen uso para sistemas en el que se busque sumar valores de varios objetos, es decir de varias series por ejemplo. Un ejemplo que yo podría dar para usar el conocimiento de sobrecarga de operadores es, buscar la calificación promedio de una selección de series. Es decir, mostrar que las películas mas populares tienen un promedio de calificación alto o algo similar. También, se podría usar para comparar una serie de otra. Igualmente, por su calificación.

Problemas Posibles:

El problema principal al que se podría correr con este sistema es que no es posible agregar más películas o series de una manera efectiva al sistema, dado a que se debe de insertar cada una individualmente al instanciar el objeto dentro del programa.

Igualmente, otro problema al que podríamos correr es que la calificación no es un promedio asignado, sino que es una calificación que cambia por cada usuario que se ingresa.

Yo considero que el contenido aprendido no aborda como podríamos llegar a una solución precisa de este problema, dado a que considero que seria necesario el uso de una base de datos o sistemas de MySQL para que esto funcionara de la manera en la que se desea.

Igualmente, se nos fue mencionado que podríamos usar un formato de switch case para el menú, desafortunadamente, desconozco su sintaxis, y su función, por lo que no pude implementarlo al programa como fue solicitado.

Conclusión:

Esta actividad la verdad fue muy enriquecedora. Me ayudo mucho a tener un mejor entendimiento de la funcionalidad de clases, y como estas actúan y pueden usarse para organizar y simplificar el código que se realiza, al igual que para describir y ilustrar situaciones que son muy reales y llevadas mas a un contexto cotidiano. Aprendí sobre el uso de polimorfismo, u como este puede ser útil para códigos largos o de una gran cantidad de objetos instanciados, dado a que podemos usar una misma función para todas nuestras clases.

Aprendí sobre la sobrecarga, que nos deja hacer operaciones entre varios objetos. Aprendí sobre apuntadores, brevemente, y me gustaría ver como podría usarlos de una manera efectiva para hacer que el uso de memoria sea eficaz cuando se programa.

Para finalizar, creo que esta actividad me ayudo no solo en el sentido de programación, sino que también me brindo habilidades muy esenciales para la vida real, como es el trabajo en equipo, el poder lidiar con situaciones bajo presión, el poder utilizar información proporcionada y brindarle uso para la solución de algo entre muchas otras más.