**Conceptualización Entrega 2**

Grupo:14

Para esta entrega tomamos mas en cuenta las bibliotecas disponibles para realizar acciones, tal como nos recomendó nuestro tutor al revisar el primer diagrama UML, también gracias a el tuvimos una idea mas clara sobre como conectar las clases entre si y de realizar un modelo más detallado y organizado.

Lista de clases:

Media: Es una clase abstracta que contiene todo lo que se puede relacionar a videos y canciones en lo que se refiere a sus atributos y métodos, con tal de que luego se pudieran crear las clases que hereditaria en estos mismos.

Song: Esta clase se refiere a las canciones en sí, donde añadimos atributos que no pertenecerían a la clase Movie y que tengan los métodos que sean propios de esta, como la posibilidad de descargar las canciones por medio de una interfaz y mantener el minuto en el que se encontraba la canción antes de cerrar la aplicación, con tal de hacer esto se le agrego la clase serializable para poder recordar el minuto en específico.

Movie: La otra clase que hereda de Media, y que tal como song se le agregan sus atributos propios, sin embargo, no se le añade la interfaz de descargar por lo que se menciona el enunciado

Person: La creamos como una clase normal y no abstracta, ya que solo serán atributos los que serán heredados por las clases hijas.

Actor: Se refiere al trabajador o persona que actuó en el video.

Director: Se refiere a la persona que grabo el video.

Artista: Se refiere a la persona que trabajó en la canción.

Playlist: La clase abstracta playlist sirve como base para crear las demás playlist, que como ya se ha dicho antes se usará el método de herencia, aunque a diferencia del anterior esta no será abstracta, pues aquí sí pueden crearse objetos y rellenar los métodos y debido a que el concepto de playist es uno más concreto que abstracto.

Playlist\_song: Es la clase que se encarga de contener a las canciones en una playlist

Playlist\_movie: Es la clase que se encarga de guardar los videos/películas en una playlist

User: Se refiere a la clase del usuario de la aplicación, donde se guarda la información de este y se usa los números del 0 al 2 para definir el tipo de usuario que es. Si es gratuito, premium o administrador. Para los administradores se crea una interfaz en el que puedan usar las funcionalidades de este. Debido a que no necesita los atributos de Person no es necesario que herede de este, a pesar de que a un usuario se le considera una persona

Controlador: En esta clase se tiene aquellas acciones realizadas por el usuario dentro de la aplicación, lo que le permite moverse dentro de esta e interactuar con la media deseada, ya sea para darle un rating, descargar, buscar entre otras cosas.

Interacción: Es la clase encargada de darle notificaciones al usuario, le manda mensajes indicándole alguna clase de error o bien darle un aviso de cuando se sigue a una persona o media realiza un post. Debido a que no se necesita de crear objetos es una clase estática.

Searcher: Es la clase estática encargada de buscar por medio de filtros (usando para estos una interfaz). El usuario escribe lo que desea buscar y la clase se encarga de encontrarlo.