

## **SUPUESTOS DE ALAINID ENTREGA 2**

1. Cada usuario podrá tener solo un perfil, es decir, no se podrá hacer lo que hace netflix que con un usuario/cuenta, se pueden tener varios perfiles.
2. Además del administrador, el usuario premium podrá descargar archivos.
3. Solo el admin podrá importar canciones y videos.
4. En una película puede haber máximo un director, pero cada director puede participar en varias películas.
5. El ranking será una nota de 1 a 10 en la que solo se podrán poner notas con número enteros, pero luego el promedio podrá ser con decimales.
6. La nueva funcionalidad KARAOKE, será una categoría especial en la que habrá solo una cantidad limitada de canciones. Es decir, no todas las canciones podrán ponerse en modo Karaoke.
7. La nueva funcionalidad HISTORIAL DE REPRODUCCIÓN, consistirá en registrar un historial en el que se pueda ver cada canción y video que se escuchó y vio en la sesión.
8. El usuario podría descargar solo un archivo a la vez.
9. Una canción solo puede pertenecer a un álbum independiente de que tenga a más de un cantante.
10. La lista inteligente solo se armara con un único criterio.
11. No se puede reproducir videos, estos se podrá hacer en la entrega 3 utilizando forms
12. Un usuario puede hacer más de 1 rating a la misma canción (cambia de opinión)

## Lista de Clases

1. **ALAINID**: En esta clase se controlará gran parte de la aplicación.  
Ej: ver videos y canciones, descargar, hacer playlist, agregar canciones, videos, etc.  
Es la clase que tendrá los métodos que controlarán a las demás clases.
2. **Song**: Esta clase define lo que será una canción, tendrá atributos como el nombre, duración, cantante etc..Cada una de las canciones que se suban dentro de la plataforma se les podrá implementar el método para ver su información donde mostrará datos como el tamaño, calidad, tipo, información básica de esta.
3. **Video**: Esta clase define lo que será una video, tendrá atributos como el nombre, duración, director, actores etc..Cada uno de estos videos que se suban dentro de la plataforma se les podrá implementar el método para ver su información donde mostrará datos como el tamaño, calidad, tipo, información básica de esta.
4. **Admin**: Esta clase contendrá los métodos necesarios para que el admin pueda hacer todos los cambios que quiera, ya sea importar, descargar, administrar usuarios, etc...
5. **Search**: Aquí se implementan los métodos para poder hacer búsqueda tanto simples como compuestas. Además se guardaran las búsquedas hechas para poder ir teniendo un registro de las preferencias del usuario.
6. **Playlist**: Clase madre abstracta de PlaylistVideo y PlaylistSong, ésta contendrá los atributos de estas.
7. **PlaylistVideo**: Aquí se podrán ir registrando y guardando las diferentes playlist de videos que el usuario quiera crear. (Hereda Playlist)
8. **PlaylistSong**: Aquí se podrán ir registrando, guardando y las diferentes playlist de música que el usuario quiera crear. (Hereda Playlist)
9. **Studio**: Clase madre abstracta de FilmStudio y RecordCompany, ésta contendrá los atributos de estas.
10. **Filmstudio**: Esta clase servirá para poder almacenar estudios de rodaje de videos o películas en una lista.
11. **RecordCompany**: Esta clase servirá para poder almacenar nuevos estudios de grabacion de musica en una lista. (Hedera Studio)

12. **Artist:** Clase madre de Actor, Director, Singer, Composer. Esta contendrá específicamente el nombre, edad, fecha de nacimiento y nacionalidad del artista.
13. **Singer:** Esta clase es poder almacenar los distintos cantantes de la música que se suba a la plataforma, esto dentro de una lista.
14. **Composer:** Esta clase es poder almacenar los distintos compositores de la música que se suba a la plataforma, esto dentro de una lista.(Hedera Studio).
15. **Album:** Esta clase servirá para poder almacenar álbumes de los distintos cantantes en una lista.
16. **FavoriteSong:** Esta será una static class que representará la playlist única FAVORITOS de canciones. Contiene una lista tipo canción dentro.
17. **FavoriteVideo:**Esta será una static class que representara la playlist única FAVORITOS de videos. Contiene una lista tipo video dentro.
18. **Director:** Esta servirá para almacenar los distintos directores de la plataforma, que serán de tipo artist.
19. **Actor:**Esta clase servirá para poder almacenar actores de los distintos videos en una lista.
20. **Play:** Gracias a esta clase, podremos reproducir los videos y canciones que queramos, esta controla las reproducciones de la aplicación de forma completa. Tendrá métodos como el historial y el poder reproducir canciones o videos en cola.
21. **SingUporSingin:** Con los métodos de esta clase se podrá dar la opción de registrarse con una nueva cuenta o de ingresar a la plataforma con una cuenta creada anteriormente.
22. **User:** Esta clase define lo que será un usuario, tendrá atributos como el nombre de usuario, nombre de usuario, email, etc..Cada uno de los usuarios que existan dentro de la plataforma se les podrá implementar el método para ver su información donde mostrará datos como el nombre, fecha de nacimiento, email, información básica de este.
23. **inputoutput:** Esta clase será la que permita y gestione todo lo que tenga que ver con el input y output entre la consola y la aplicación ALAINID (Llamará a todos los métodos de ALAINID). Esta clase será la que esté directamente conectada con la consola. Se ha creado para evitar que en Program se llene de cosas y dejarlo solo con una variable tipo inputoutput que llame al método comenzar.