SUPUESTOS

1. Creamos la clase CONTENT con el objetivo de que sea abstracta para que SONGS y MOVIES hereden métodos en común, pero cada uno lo aplicará de distinta manera.
2. No mencionamos el tipo de variable del atributo photo debido a que, de momento, no sabemos cómo expresarlo y manipularlo en el código.
3. Creamos la clase ALBUM porque encontramos que este posee demasiada información para ser solamente un atributo de SONGS, y también porque nos facilita la gestión del método de búsqueda que se nos pide como requisito mínimo.
4. Creamos la clase ARTIST porque encontramos que este posee demasiada información para ser solamente un atributo de SONGS, y también porque nos facilita la gestión del método de búsqueda que se nos pide como requisito mínimo.
5. Creamos la clase ACTOR porque encontramos que este posee demasiada información para ser solamente un atributo de MOVIES, y también porque nos facilita la gestión del método de búsqueda que se nos pide como requisito mínimo.
6. Para la administración de usuarios nos basamos en el diagrama UML de Netflix para facilitarnos la manipulación y creación de perfiles en la plataforma por usuario.
7. Creamos las clases QUEUESONGS y QUEUEMOVIES ya que encontramos que, de esta manera, es más fácil manipular las listas de canciones y películas que se encuentran en la cola.
8. Creamos las clases PLAYLISTSONGS, PLAYLISTFAVOURITESSONGS, PLAYLISTMOVIES y PLAYLISTFAVOURITESMOVIES porque encontramos que es más fácil manipular las listas de canciones y películas de esta manera que si fuesen solamente atributos de SONGS y MOVIES, respectivamente.
9. La primera funcionalidad innovadora que le agregamos es la sugerencia inteligente tanto para SONGS como para MOVIES.
10. La segunda funcionalidad innovadora que le agregamos es la opcion de escuchar o ver la misma cancion o pelicula que un usuario esta consumiendo en el mismo momento.
11. Las funcionalidad de privado en las playlist aun no esta definido a la perfección para esta entrega.
12. Cuando se agregue una cancion o una película a Spotflix, este deberá tener el mismo nombre del archivo para que se pueda reproducir a la perfección, además de que este archivo este guardado de antes en la carpeta debug que se encuentra en bin.
13. La funcionalidad de poder salirse y volver al mismo lugar donde se salió del terminal aún no está definido correctamente para esta entrega.
14. Las funcionalidades nuevas están en fase de demostración, es decir, aún les falta para poder estar completas.
15. La funcionalidad de búsqueda está en fase de demostracion, osea aun le falta para poder estar completa. Por ende, si uno busca sin a ver agregado una canción o película antes, esta mostrará error.
16. Los archivos de videos deben ser archivos .mp4 y ,además, deben estar localizados en la carpeta DeBug, la cual está en Bin.
17. Los archivos de música deben ser archivos .wav y ,además, deben estar localizados en la carpteta DeBug, la cual esta en Bin.
18. Agregamos una carpeta en el repositorio, la cual contiene los archivos .wav y .mp4 que deben añadirse a la carpeta DeBug, para que el programa funcione a la perfección. Esto se hizo ya que el .gitignore no deja subir la carpeta bin ni la debug, por ende esto debe hacerce manualmente.

DESCRIPCIÓN DE CLASES

1. USER: clase encargada de administrar sus perfiles.
2. PROFILE: cara visible del usuario. Posee los datos privados de USER, como email y contraseña. Puede seguir películas, canciones, artistas, etc.
3. CONTENT: clase abstracta de la plataforma que permite compartir el método de busqueda entre SONGS y MOVIES.
4. MOVIES: clase que hereda de CONTENT. También se encarga de recopilar toda la información de las películas de la plataforma.
5. SONGS: clase que hereda de CONTENT. También se encarga de recopilar toda la información de las canciones de la plataforma.
6. ACTOR: clase encargada de recopilar la información de los actores de las películas.
7. ARTIST: clase encargada de recopilar la información de los artistas de las canciones.
8. PLAYLISTMOVIES: clase encargada de almacenar las playlists de películas seleccionadas por PROFILE.
9. PLAYLISTSONGS :clase encargada de almacenar las playlists de canciones seleccionadas por PROFILE.
10. PLAYLISTFAVOURITESMOVIES: clase encargada de almacenar la playlist de películas favoritas seleccionadas por PROFILE.
11. PLAYLISTFAVOURITESSONGS: clase encargada de almacenar la playlist de canciones favoritas seleccionada por PROFILE.
12. QUEUEMOVIES: clase encargada de almacenar las películas a reproducir seleccionadas por PROFILE.
13. QUEUESONGS: clase encargada de almacenar las canciones a reproducir seleccionada por PROFILE.
14. ALBUM: clase encargada de recopilar la información de los álbumes de la plataforma.