

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. No especificamos cómo manipularemos los archivos multimedia porque, de momento, no poseemos esos conocimientos.
7. 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.
8. 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.
9. 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.
10. La primera funcionalidad innovadora que le agregamos es la sugerencia inteligente tanto para SONGS como para MOVIES.
11. La segunda funcionalidad innovadora que le agregamos es la opción de escuchar o ver la misma canción o película que un usuario está consumiendo en el mismo momento.

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 ciertos métodos 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.
15. SUGGESTIONSONG: clase encargada de recopilar los aspectos en común de las canciones escuchadas
16. SUGGESTIONMOVIE: clase encargada de recopilar los aspectos en común de las películas vistas.
17. JOINSONG: clase encargada de poder escuchar lo que un usuario, al que sigues, esta escuchando en ese momento.
18. JOINMOVIE: clase encargada de poder ver lo que un usuario, al que sigues, esta viendo en ese momento.