

Clase Usuario:

- 1) OnEmailSent(); Pregunta al usuario si quiere verificar correo

Clase ProfileUser:

Atributos:

- 1) followedProfile (List<ProfileUser>): Será una lista donde se irá guardando cada persona a la que el usuario le ponga "seguir".
- 2) followers(List<ProfileUser>):
- 3) followedPlaylistVideo(List<PlaylistVideoEmptyClass>):
- 4) followedPlaylistSongs(List<Actor>)
- 5) followedDirectors(List<Director>)
- 6) followedDisk(List<Album>):
- 7) downloadsSongs(List<SongClass>):
- 8) likesSongs (List<SongClass>):
- 9) likesVideos(List<Video>)

Métodos:

- 1) Crear playlist canción: la playlist creada se guardará en la biblioteca de cada perfil.
- 2) Crear playlist vídeo: la playlist creada se guardará en la biblioteca de cada perfil.
- 3) Ver playlist.
- 4) Ver descargas.:
- 5) Editar gustos.
- 6) Eliminar playlist canción.
- 7) Eliminar playlist video.
- 8) Quitar canciones de favorito.
- 9) Eliminar seguidores.
- 10) Quitar artistas de favoritos.
- 11) Quitar álbumes de favoritos.
- 12) Buscar: Podrá buscar artistas, actores, cantantes, canciones, películas, etc.
- 13) Escuchar canción: indicará cada segundo que ya ha sido escuchado, para luego si se sale, la canción se reanude donde se quedó y no desde el inicio.
- 14) Download: Descargar canción.

Clase Video:

Atributos:

- 1) Categoría (lista string) : Donde se podrá encontrar una canción diferenciada por su tipo
- 2) Actores (lista string): Donde se podrá encontrar los actores que participan en el video
- 3) Descripcion(string): De que tratara la película
- 4) Subtitulos(string): Lo que hablan en el video

Métodos:

- 1) Agregar Datos().

Clase SongClass:

Atributos:

- 1) composer(string): Recibe al que compuso la canción.
- 2) singer(Singer): Artista que canta en dicha canción.
- 3) album(Album): Su discografía.
- 4) lyrics(string): Funcionalidad agregada a la aplicación (*)

Métodos:

- 1) AddData(): Agrega todos los datos pedidos por una canción.

Clase Playlist Spotify:

Atributos:

- 1) fileType(string): Se registra el tipo de archivo que está contenida la canción. (mp3, mp4)
- 2) Playlist: Lista de string de la playlist que desee el perfil.
- 3) PlaylisFav: Lista de string de las canciones favoritas del perfil.

Métodos

- 1) AddPlaylist(): Se añade una playlist hecha por el usuario
- 2) AddPlaylistFav(): Se agrega una playlist de favoritos según likes.
- 3) AddQueue(): Se crea una cola.

Clase PlaylistVideoEmptyClass:

Atributos:

- 1) typeFile(string): Se registra el tipo de archivo que es el video.
- 2) Playlist: Lista de string de la playlist que desee el perfil.
- 3) PlaylisFav: Lista de string de las canciones favoritas del perfil.
- 4) namePlaylist:

Métodos

- 1) AddPlaylist(): Se añade una playlist hecha por el usuario
- 2) AddPlaylistFav(): Se agrega una playlist de favoritos según likes.
- 3) AddQueue(): Se crea una cola de videos.

Searchclass:

Es una clase únicamente de métodos.

Métodos:

- 1)SearchForComposerr ();busca por director de video o compositor de la canción
- 2)SearchKeyWordSong(); buscar por palabra clave de la canción
- 3)SearchKeyWordVideo;buscar por palabra clave del video
- 4)SearchForCharacteristic(); busca canciones o videos por edad , sexo ..
- 5)SearchForResVideo();
- 6)SearchForEvaluation (); busca canciones o videos dependiendo de la calificación
- 7)SearchProfile(): Sirve para encontrar otros perfiles de la aplicación.
- 8)SearchSinger();Buscar canciones por cantante.
- 9)SearchForMember (); busca canciones o videos por participantes o integrantes
- 10)SearchDirector(); Busca videos por director

Clase Actor

Atributos:

movies: List<Video>
awards: List<string>
yearsActive: int

Clase Album

Atributos:

nameAlbum: Refiriendose al nombre del album
gender: Que tipo de género de musica pertenece
producer: Quien fue el productor de dicho album
listSongs: La lista de canciones que pertenecen al album
singer: Cantante principal del album.
publicationDate: Año de publicacion.

Clase Database

Atributos

registered: diccionario donde se guardan cada registro en el siguiente formato
[usuario, correo, password, linkVerificacion, fecha, numero].

Metodos:

ChangePassword(string usr, string newpasswd): Metodo para cambiar la contraseña de usr por newpasswd.

LogIn: Método para iniciar sesión.

AddUser: Método para agregar un nuevo usuario, verificando que además no exista.

Clase Director

Atributos:

movies: List<Video>

awards: List<string>

yearsActive: int

Clase MailSender

Metodos:

OnRegistered:

OnPasswordChanged:

OnEmailSented:

Clase Person

Atributos:

name:

age:

lastName

gender:

nationality

occupation: Sirve para identificar quien es actor usuarios, director.

Clase RegisterEventArgs

Método

OnPasswordChanged:

Clase Reproductor

Métodos:

- 1) Iniciar(): Inicia un cronometro que nos indicara los minutos y segundos que lleva la cancion seleccionada.
- 2) Detener():Para el cronometro
- 3) Tic (): Metodo reservado de timer

Clase Singer

Hereda de la clase Person.

Atributos:

songs

Álbum

awards

voice Type

genders

yearsActive:

discographic

Clase ProfileManagment

Clase Profile

Explicación de la solución en cada uno de los puntos pedidos.

1) Importar canciones y vídeos: Agregar nuevos archivos multimedia de forma sencilla.

Soportar al menos los formatos más comunes (mp3, wav, mp4, etc).

El Perfil tiene la opción de importar canciones y videos. Para ello se creó la interfaz importar. Esto se hizo mediante la serialización de las clases SongClass y video, donde se pide toda la información necesaria.

2) Información de las canciones (y vídeos)

a) Se crea la clase canción y la clase video con todos sus atributos respectivos

b) Se agrega métodos a la clase canciones y video para agregar datos

c)

d) Se agregaron métodos a la clase abstracta para esto. "agregar reproducción", "Agregar calificación"

3) Search Class tiene diferentes métodos para buscar canciones o videos.

4) Playlists. Se agregó a la cada clase (video y canción) su clase hija.

Se encargará de tener una lista de las canciones y videos respectivos.

La clase usuario va a tener el método 'Ver Biblioteca' donde podrá encontrar las playlist favoritas todo.

Se hizo serialización de las playlist en la clase program, el perfil podrá crear playlist y guardar canciones.

5) En cola. Se agregó en la clase Playlist de cada uno el método de agregar cola.

Esto creará listas que el usuario podrá ver.

Cuando se acabe la última canción esta se borra queda la lista vacía. Se termina. (Cómo va a clase playlist va a biblioteca el usuario lo podrá ver)

6) Playlist específica de favoritos

Dentro de la misma clase playlist, se agregó el método para poder ver por orden de calificación las canciones y los videos.

La playlist consta con una lista distinta que serán las favoritas.

Cada vez que un usuario le de like a una canción se va a guardar automáticamente en la playlist.

7) Buscar y seguir. La clase búsqueda tendrá este método. Además, los atributos de la clase perfil permiten seguir y tener seguidores. Clase Search implementa diferentes métodos.

8) Agregar imágenes

Agregar un método a la clase abstracta --> Agregar imagen (SOLO USUARIOS PREMIUM)

9) Administración de usuarios

Profile Management puede administrar perfiles.

10) Diferenciar usuarios pagos de no pagos

Nuevo Atributo Usuario 'Pago'. Al inicio del programa, se ve quien pago o no pago por el, diferenciando 3 planes: Básico, Premium, Familiar

11) Las canciones pueden o no ser descargadas

Método Descargar en canción. (interfaz) (Solo Para usuarios Premium.)

Profile user tiene el método para descargar canciones.

12) Se crea cronómetro y se implementa en la clase program.