Explicación programa Spotflex Grupo 1

1. Main Program:

El Main program será el menú, lo que llevará al usuario a las distintas funcionalidades del programa

* + 1. Menú de Inicio
* Ingresar como usuario: El usuario puede iniciar su sesión
* Crear Usuario: El usuario creará su sesión
* Salir: El usuario saldrá de la aplicación, guardando todo el progreso que se hizo en la plataforma
  + 1. Menú de sesión: Al ingresar como usuario lo llevara a este menú
* Modificar Opciones del usuario: Acá puedes editar los datos del Usuario tales como su nickname, contraseña, tipo de perfil, etc.
* Reproducir Canción: Acá puedes reproducir alguna de las canciones que están en tu librería
* Reproducir Video: Acá puedes reproducir alguno de los videos que están en el librería
* Descargar Canción: Acá puedes descargar alguna canción que esté en tu librería
* Calificar: Acá puedes calificar ya sea una canción o un video de 1 a 5 estrellas
* Ranking Actual: Acá puedes ver un ranking de los mejores videos o las mejores canciones:
* Búsqueda Personalizada: Con esta opción puedes buscar una canción, un video, una playita, un usuario o una persona por diferentes filtros ya sea por nombre, género, personas involucradas, etc. Es posible buscar por más de un filtro.
* Subir un video: El usuario puede subir videos a la plataforma
* Subir una canción: El usuario puede subir canciones a la plataforma
* Quitar Video: El usuario puede quitar un video del sistema siempre y cuando lo haya subido el
* Quitar Canción: El usuario puede quitar una canción del sistema siempre y cuando lo haya subido el
* Hacer Playlist: Con esta opción el usuario puede crear su playlist
* Modificar Playlist: Con esta opción el usuario puede modificar una playlist ya creada por el
* Cerrar Sesión: El Usuario cerrará su sesión
* Cerrar Programa: El programa se cerrará, guardando todo lo que se haya hecho en la plataforma.

1. Clase Usuario: Permite la creación de usuarios y también es la que almacena los nombres de usuario y contraseñas para el inicio de sesión

2.1) Atributos:

* + 1. private string Nombre: Contiene el nombre del usuario
    2. private int Edad: Contiene la edad del usuario
    3. private bool Tipo de usuario: Indica si el usuario es público o privado con true o false
    4. private bool Premium: Indica si el usuario es premium o no con true o false
    5. private string Nickname: Indica el nickname del Usuario
    6. private string Contraseña: Indica la contraseña del usuario
    7. private bool Administrador: Indica si el usuario es administrador con true o false
    8. private string Foto de perfil: Indica el path o directorio de la foto de perfil asociada al usuario
    9. private Lista de strings GenerosQueSigue: Indica los géneros que le gustan al usuario
    10. private Lista de Personas CantantesQueSigue: Indica los cantantes que sigue el usuario
    11. private Lista de Multimedia Favoritos: Indica las canciones y videos favoritas del Usuario
    12. private Lista de Multimedia Recomendaciones: Indica las canciones y videos que le podrían gustar al usuario
    13. private Lista de Multimedia UsuariosSeguidores: Indica los Usuarios que siguen a este usuario
    14. private int Contador: Indica a los usuarios las oportunidades que tienen para ganar el mini juego que les permite convertirse en premium si es que son usuarios free
    15. private Lista de Canciones CancionesAgregadas: Indica las canciones que agrego el usuario
    16. private Lista de Videos VideosAgregados: Indica los videos que agrego el usuario
    17. private Lista de Playlist Playlists Agregadas: Indica las playlists que agrego el usuario

2.2) Metodos:

1. Public Void Constructor de Personas: Este creara a las personas con los datos ingresados por el usuario
2. Public Void AgregarFotoPortada(): Ingresara una foto de portada al sistema, preguntándole al usuario el path de esta
3. Public Void CargarFotoPerfil(): Mostrara la foto de perfil actual del usuario
4. Public string GetFotoPerfil(): Retornara el path de la foto de perfil asociada
5. Public Void CambiarRango(): Con esta opción el usuario puede cambiar su tipo de Usuario (Free o Premium)
6. Public Void CambiarTipo(): Con esta opción el usuario puede cambiar su tipo de Usuario
7. Public Void Seguir(usuario Usu): Con esta opción se agregara a la lista de seguidores del usuario el Usuario usu, siempre y cuando no lo haya seguido
8. Public Void InformacionUsuario(): Con esta opción se mostrará la información del usuario, siempre y cuando el usuario sea público
9. Public Void Add\_Song\_To\_My\_List(Canción canción): Con esta opción el usuario agregara a su lista de canciones agregadas la canción
10. Public Void Add\_Video\_To\_My\_List(Video video): Con esta opción el usuario agregara a su lista de videos agregadas el video
11. Public Void Add\_Playlist\_To\_My\_List(Playlist playlist) Con esta opción el usuario agregara a su lista de playlists agregadas
12. Public string Get\_Name(): Retorna el nombre del usuario
13. Public int Get\_Edad(): Retorna la edad del usuario
14. Public bool Get\_TipoUsuario(): Retorna el tipo de usuario (público o privado)
15. Public bool Get\_Premium(): Retorna si el usuario es premium
16. Public string Get\_Nickname(): Retorna el nickname del usuario
17. Public string Get\_Password(): Retorna la contraseña del usuario
18. Public List<String> Get\_GenerosSeguidos(): Retorna la lista de los géneros que sigue el usuario
19. Public List<Multimedia> Get\_Favoritos(): Retorna la lista de favoritos
20. Public List<Multimedia> Get\_Recomendados(): Retorna la lista de recomendados
21. Public bool Get\_Admin(): Retorna si el usuario es Administrador o no
22. Public int Get\_Contador(): Retorna la cantidad de intentos que tiene para el mini juego
23. Public Void Cambiar\_Dato\_String(int caso, string palabra): Cambiará el nickname, el nombre o la contraseña del usuario según el caso usando la palabra
24. Public Void Cambiar\_Dato\_Booleano(int caso): Cambiará el tipo de usuario, si es premium o si es administrador según sea el caso
25. Public Void Cambiar\_Dato\_Edad(int edad): Cambiará la edad del usuario usando el parámetro
26. Public Void Lose\_BonusGame(): Descontará una oportunidad para jugar al mini juego
27. Public Void Add\_Favoritos(Multimedia multi): Agregara la canción o video a la lista de favoritos, si ya estaba, la quitará
28. Public Void Consultar\_Favoritos(): Mostrará todos los archivos multimedia que están en favoritos
29. Public List<Canciones> Get\_ListaCancionesAgregadas(): Retorna la lista de canciones agregadas
30. Public List<Video> Get\_ListaVideosAgregados(): Retorna la lista de videos agregadas
31. Public List<Playlist> Get\_ListaPlaylistAgregadas(): Retorna la lista de playlists agregadas
32. Clase Almacenamiento: La Clase Almacenamiento almacena todos los usuarios, videos, canciones, listas de reproducción, esta sería la clase cerebro del programa si se puede decir de alguna forma, ya que maneja gran parte del programa

3.1) Atributos:

1. Private Lista objeto Canciones Lista\_Canciones: Contiene todas las canciones
2. Private Lista objeto Videos Lista\_Videos: Contiene todos los videos
3. Private Lista objeto Usuarios Usuario: Contiene todos los usuarios
4. Private Lista objeto canción ColaDeReproduccionCancion: Contiene todas las canciones en cola
5. Private Lista objeto video ColaDeReproduccionVideo: Contiene todos los videos en cola
6. Lista de playlist multimedia: Guarda todas las playlists de canciones y videos
7. Lista de playlist de personas: Guarda todas las playlist de personas

3.2) Métodos

1. Public Void Quitar\_Cancion(Canción canción): Sacara la canción de la plataforma
2. public Void Quitar\_Video(Video video): Sacara el video de la plataforma
3. public Void Add\_Cancion(Canción canción): Agregara la canción al sistema
4. public void Add\_Video(Video video): Agregara el video al Sistema
5. public void AgregarUsuario(Usuario usu): Agregara el usuario al sistema
6. public void AgregarPersona(Persona persona): Agregara la persona al sistema
7. public void RankingDeCanciones(): Mostrara el ranking de canciones
8. public void RankingDeVideos(): Mostrara el ranking de videos
9. public void Reproducir\_Multimedia(Multimedia multi): Reproducirá el archivo multimedia
10. public void Crear\_Playlist\_Canciones(string nombre\_playlist): Creará una playlist de canciones con ese nombre, acá el usuario podrá meter las canciones que quiera
11. public void Crear\_Playlist\_Videos(string nombre\_playlist): Creará una playlist de videos con ese nombre, acá el usuario podrá meter los videos que quiera
12. public void Modificar\_Playlist(): El usuario podrá modificar una de sus playlists creadas, puede agregar canciones, quitarlas o borrar la playlist
13. public void ColaReproduccionCanciones(): El usuario puede agregar las canciones a la cola de reproducción
14. public void ColaReproduccionVideos(): El usuario puede agregar los videos a la cola de reproducción de videos
15. public void PortadaCancion(Canción canción): El usuario puede agregar la portada a la canción con una imagen
16. public void PortadaVideo(Video video): El usuario puede agregar la portada a la canción con un video
17. public void MostrarPortadaCancion(Canción canción): El usuario puede ver la portada de la canción
18. public void MostrarPortadaVideo(Video video): El usuario puede ver la portada del video
19. public void Filtro\_busqueda(List<string> buscado, List<int>criterio, Usuario usu, List<int> condicion\_extra): Usando una lista de criterio y una lista de palabras claves, cada una asociada con el índice (es decir el primer criterio está asociado a la primera palabra clave) buscará todas las canciones, videos, playlists, usuarios o personas según sea el caso que cumpla con las condiciones (además para las variables numéricas habrá una condición extra si quiere buscar algo igual, menor o mayor al valor), de ahí se mostrará una lista enumerada de todo lo que cumpla con los criterios, el usuario elegirá qué elemento quiere observar y dependiendo de lo que sea se le darán opciones ya sea reproducir en el caso de las canciones o videos, ver su información, etc.
20. Public void ComprobarUsuarios(string nombre, string contra): Verifica si al ingresar al sistema, el usuario ingresó bien su nickname y contraseña
21. Public Void Bonus\_Game(Usuario usu): El usuario jugará un minijuego que consiste en adivinar la canción mostrando la letra de esta, si adivina el usuario pasará a ser premium
22. Public List<Canciones> Get\_ListaCanciones(): Retorna la lista de canciones
23. Public List<Video> Get\_ListaVideos(): Retorna la lista de videos
24. Public List<Usuario> Get\_Usuarios(): Retorna la lista de usuarios
25. Public List<Canciones> Get\_ColaCanciones(): Retorna la cola de canciones
26. Public List<Video> Get\_ColaVideos(): Retorna la cola de videos
27. Public List<Playlist> Get\_Playlists(): Retorna las playlists
28. Public List<Personas> Get\_Personas(): Retorna la lista de personas
29. Clase Padre Multimedia: Esta clase contendrá todo lo que puede tener un archivo multimedia ya sea canción o video, como también lo que puede hacer de por si un archivo de estas características

4.1) Atributos

* 1. Protected int Duración: Indica la duración del archivo (se instaló librería NuGet WMP.Lib para la ejecución de este atributo)
  2. Protected string Título: Indica el título del archivo
  3. Protected string Fecha\_Inclusion: Indica la fecha en la que se incluye el archivo a la plataforma
  4. Protected long Size: Indica el tamaño del archivo en bytes
  5. Protected string Estudio: Indica el estudio donde proviene el archivo
  6. Protected string CarpetaArchivo: Indica el directorio del archivo
  7. Protected string Fecha\_Publicacion: Indica la fecha de publicación del archivo
  8. Protected string Descripción: Indica la descripción del archivo
  9. Protected string Portada: Indica el directorio de la portada del archivo
  10. Protected int NumeroReproducciones: Indica el número de reproducciones del archivo
  11. Protected List<string> Comentarios: Indica los comentarios que tiene el archivo
  12. Protected List<int> Ranking: Indica las calificaciones que tiene el archivo
  13. Protected List<string> Genero: Indica los géneros de la canción
  14. Protected List<Usuario> Seguidores: Indica los Usuarios que siguen al archivo

4.2) Metodos

1. Constructor: Con el constructor se generara el objeto multimedia de acuerdo lo que ingresa el usuario
2. Public string Get\_Portada(): Retorna el directorio de la portada
3. Public void Agregar\_Comentarios(string comentario): Agrega el comentario a la lista de comentarios
4. Public void Seguir(Usuario usu): Agrega al usuario a la lista de seguidores
5. Public void Ranked(int estrellas): Agrega la calificación (Estrellas) a la lista de calificaciones
6. Public double Get\_Mean\_Tier(): Entrega la calificación promedio del archivo multimedia
7. Public void Cambiar\_Ubicacion(string newpath): Cambia el directorio del camino con newpath
8. Public void Sumar\_Reproducciones(): Sumará una reproducción más al archivo
9. Public string Get\_Titulo(): Retornara el título del archivo
10. Public int Get\_Duracion(): Retornará la duración del archivo
11. Public string Get\_FechaInclusion(): Retornará la fecha de inclusión
12. Public long Get\_Size(): Retornara el tamaño del archivo
13. Public string Get\_Estudio(): Retornara el estudio de donde proviene el archivo
14. Public string Get\_Carpeta\_Archivo(): Retornara el path de donde proviene el archivo
15. Public string Get\_Fecha\_Publicacion(): Retorna la fecha en la que se public el archive
16. Public string Get\_Descripcion(): Retorna la descripción del archivo
17. Public int Get\_Reproducciones(): Retornará el número de reproducciones que tiene el archivo
18. Public List<String> Get\_Comentarios(): Retorna los comentarios que tiene el archivo multimedia
19. Public List<int> Ranking(): Retorna la lista de notas del archivo multimedia
20. Public List<string> Get\_Genero(): Retorna los géneros que tiene el archivo multimedia
21. Public List<Usuario> Get\_Seguidores(): Retorna la lista de Usuarios que sigue la canción
22. Clase Canciones: Clase hija de Multimedia, acá se tienen los atributos específicos de la clase canciones (esta clase puede usar todo lo que está en multimedia)

5.1) Atributos

* 1. Private string Letra: Indica la letra de la canción
  2. Private string Álbum: Indica el álbum/disco de la canción
  3. Private List<Personas> PersonasCancion: Indica los artistas de la canción

5.2) Métodos

* 1. Constructor: Se generará un objeto de la clase canción según las características dadas por el usuario
  2. Public void Información(): Dará toda la información de la canción
  3. Public string Get\_Letra(): Retornara la letra de la canción
  4. Public string Get\_Album(): Retornara el álbum de la canción
  5. Public List<Personas> Get\_Artistas(): Retornara los artistas de la canción

1. Clase Videos: Clase hija de Multimedia, acá se tienen los atributos específicos de la clase videos (esta clase puede usar todo lo que está en multimedia)

6.1) Atributos

* 1. Private string Calidad: Indica la calidad del video
  2. Private List<Personas> PersonaVideo: Indica las personas que están involucradas en la producción del video

6.2) Métodos

* 1. Constructor: Creará un objeto de la clase videos con los datos que le ingresa el usuario
  2. Public void Información(): Dara toda la informacion del video
  3. Public string Get\_Calidad(): Retorna la calidad del video
  4. Public string Get\_Staff(): Retorna la lista de personas involucradas en el video

1. Clase Playlist: Clase que almacena listas de canciones o videos que luego son guardadas en el almacenamiento

7.1) Atributos

* 1. Private List<Canciones> Playlist\_Canciones: Indica las canciones de la playlist
  2. Private List<Video> Playlist\_Videos: Indica los videos de la playlist
  3. Private string nombre: Indica el nombre de la playlist
  4. Private List<Usuarios> Seguidores: Indica los usuarios que siguen esta playlist

7.2) Métodos

* 1. Constructor: Generara un objeto de la clase playlist acorde a los datos ingresados por el usuario
  2. Public void Seguir(Usuario Usu): Se agregara al usuario a la lista de seguidores
  3. Public void MostrarCanciones(): Mostrara las canciones de la playlist
  4. Public void MostrarVideos(): Mostrara los videos de la playlist
  5. Public List<Canciones> GetPlaylistCanciones(): Retornara la lista de canciones
  6. Public List<Video> GetPlaylistVideos(): Retornara la lista de videos
  7. Public string Get\_Name(): Retorna el nombre de la playlist
  8. Public List<Personas> Get\_Seguidores(): Retorna la lista de seguidores
  9. Public Add\_Music\_To\_Playlist(Canción canción): Agrega la canción a la lista de canciones
  10. Public Add\_Video\_To\_Playlist(Video video): Agrega el video a la lista de videos

1. Clase Persona: Clase que tiene la información de las personas involucradas ya sea en una canción o en un video

8.1) Atributos

* 1. Private string Nombre: Indica el nombre de la persona
  2. Private string Apellido: Indica el apellido de la persona
  3. Private int Edad: Indica la edad de la persona
  4. Private string Sexo: Indica el sexo de la persona
  5. Private string PuestoTrabajo: Indica el puesto de trabajo de la persona
  6. Private List<Usuario> Seguidores: Indica los usuarios que siguen a la persona

8.2) Metodos

* 1. Constructor: Se genera un objeto de la clase persona con los datos ingresados por el usuario
  2. Public void Seguir(Usuario usu): Agrega al usuario usu a la lista de seguidores
  3. Public void Informacion\_Persona(Lista<video> Videos, Lista<canciones> Canción): Muestra la informacion personal de la persona y además sus trabajos realizados
  4. Public string Get\_Nombre(): Retorna el nombre de la persona
  5. Public string Get\_Apellido(): Retorna el apellido de la persona
  6. Public int Get\_Edad(): Retorna la edad de la persona
  7. Public string Get\_Sexo(): Retorna el sexo de la persona
  8. Public string Get\_PuestoTrabajo(): Retorna el puesto de trabajo de la persona
  9. Public List<Usuario> Get\_Seguidores(): Retorna la lista de seguidores de la persona

1. Clase Importacion-Exportacion: Esta clase se encarga de llevar todas las canciones y videos a almacenamiento, también se encarga de descargar las canciones y quitarlas del sistema

9.1) Métodos

* 1. Public void AgregarCanción (Almacenamiento alm, Usuario usu): Le pide al usuario usu todos los datos de la canción y la agrega a la plataforma
  2. Public void AgregarVideo (Almacenamiento alm, Usuario usu): Le pide al usuario usu todos los datos del video y la agrega a la plataforma
  3. Public void QuitarCanción (Canción canción, Almacenamiento alm): Borra la Canción de la lista de Canciones de la Clase almacenamiento.
  4. Public void QuitarVideo (Video videos, Almacenamiento alm): Borra el Video de la lista de la lista de la Clase almacenamiento.
  5. Public void Descarga(Canción canción): Descarga la canción a la carpeta de descargas del programa