

Explicación programa Spottflex Grupo 1

1) Main Program:

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

i) 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

ii) Menú de sesión: Al ingresar como usuario lo llevara a este menú

- Modificar Opciones del usuario: Aquí puedes editar los datos del Usuario tales como su nickname, contraseña, tipo de perfil, etc.
- Reproducir Canción: Aquí puedes reproducir alguna de las canciones que están en la librería de Spottflex
- Reproducir Video: Aquí puedes reproducir alguno de los videos que están en la librería de Spottflex
- Descargar Canción: Aquí puedes descargar alguna canción que esté en la librería
- Calificar: Aquí puedes calificar ya sea una canción o un video de 1 a 5 estrellas
- Ranking Actual: Aquí puedes ver un ranking de los mejores videos o las mejores canciones
- Búsqueda Personalizada: Con esta opción puedes buscar una canción, un video, una playlist, un usuario o una persona por diferentes filtros ya sea por nombre, género, personas involucradas, etc. Es posible buscar por más de un filtro, además cuando se encuentra algo puedes realizar acciones con lo encontrado.
- 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 mismo usuario.
- Quitar Canción: El usuario puede quitar una canción del sistema siempre y cuando lo haya subido el mismo usuario.
- 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.

iii) Menú de Modificar Usuario:

- Cambiar Nombre: Le da al usuario la posibilidad de cambiar el nombre
- Cambiar Edad: Le da al usuario la posibilidad de cambiar la edad
- Cambiar Accesibilidad del perfil: Le da al usuario la posibilidad de cambiar su cuenta de privado a público o viceversa
- Cambiar Tipo de Cuenta: Le da al usuario la posibilidad de ser cambiar el tipo de cuenta (Free a Premium)
- Cambiar Nickname: Le da al usuario la posibilidad de cambiar el nickname
- Cambiar Contraseña: Le da al usuario la posibilidad de cambiar su contraseña
- Cambiar Administrador: Le da la posibilidad al usuario ser administrador siempre y cuando tenga la clave asociada
- Ver Informacion del Usuario: Puedes ver tu informacion
- Ver Foto de perfil: Puedes ver tu foto de perfil, también te da la opcion de cambiarla
- Consultar Favoritos: Puedes consultar tus canciones y videos favoritos
- Borrar Usuario: Borrara tu sesión, si eres administrador te da la posibilidad de borrar a cualquier usuario, en caso de que borres tu sesión
- Volver al Menú Anterior: Te da la opcion de volver al menú principal

iv) Menú de Reproducir Canción:

- Playlist: El usuario podrá escuchar una playlist de canciones o ver una de videos
- Cola de Reproducción: Agregara elementos a la cola de canciones o de videos
- Reproducir lista de favoritos: El usuario podrá escuchar o ver su lista favorita de videos o canciones

v) Menú de Búsqueda: Acá se mostraran los criterios y las búsquedas posibles:

- Nombre: Con esta puedes buscar nombres de canciones, videos, playlists, usuarios y personas por medio de una palabra clave
- Género: Puedes buscar una canción o un video por algún genero
- Duración: Puedes buscar una canción o un video por duración, puedes buscar algo igual, mayor o menor al dato ingresado

- Fecha de Inclusión: Puedes buscar algún video/canción por la fecha de inclusión al sistema
- Fecha de Publicación: Puedes buscar algún video/canción por la fecha de creación de este en el computador
- Trabajo de la persona involucrada: Puedes buscar algún video/canción por el trabajo de la persona involucrada (ej.: cantante, director)
- Género de la persona involucrada: Puedes buscar canciones/videos por personas de un género (ej.: Buscar todas las canciones hechas por un hombre)
- Edad de la persona involucrada: Puedes buscar canciones/videos por personas de una edad (también se puede buscar por personas de mayor o menor edad)
- Nombre de la persona involucrada: Puedes buscar canciones/videos por nombres o apellidos de personas involucradas
- Información: Puedes buscar una canción/video por la descripción de esta
- Calidad: Puedes buscar una canción/video por su calidad (También puedes buscar algo mayor o menor a esa calidad)
- Calificación: Puedes buscar una canción/video por su calificación promedio (También puedes buscar algo mayor o menor a esa calificación)

2) 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:

- i) private string Nombre: Contiene el nombre del usuario
- ii) private int Edad: Contiene la edad del usuario
- iii) private bool Tipo de usuario: Indica si el usuario es público o privado con true o false
- iv) private bool Premium: Indica si el usuario es premium o no con true o false
- v) private string Nickname: Indica el nickname del Usuario
- vi) private string Contraseña: Indica la contraseña del usuario
- vii) private bool Administrador: Indica si el usuario es administrador con true o false
- viii) private string Foto de perfil: Indica el path o directorio de la foto de perfil asociada al usuario
- ix) private Lista de strings GenerosQueSigue: Indica los géneros que le gustan al usuario

- x) private Lista de Personas CantantesQueSigue: Indica los cantantes que sigue el usuario
- xi) private Lista de Canciones Favoritas: Indica las canciones favoritas del Usuario
- xii) private Lista de Videos Favoritas: Indica los videos favoritos del usuario
- xiii) private Lista de Multimedia Recomendaciones: Indica las canciones y videos que le podrían gustar al usuario
- xiv) private Lista de Multimedia UsuariosSeguidores: Indica los Usuarios que siguen a este usuario
- xv) 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
- xvi) private Lista de Canciones CancionesAgregadas: Indica las canciones que agrego el usuario
- xvii) private Lista de Videos VideosAgregados: Indica los videos que agrego el usuario
- xviii) private Lista de Playlist Playlists Agregadas: Indica las playlists que agrego el usuario

2.2) Métodos:

- i. Public Void Constructor de Personas: Este creara a las personas con los datos ingresados por el usuario
- ii. Public Void AgregarFotoPerfil(): Ingresara una foto de portada al sistema, preguntándole al usuario el path de esta
- iii. Public Void CargarFotoPerfil(): Mostrara la foto de perfil actual del usuario
- iv. Public string GetFotoPerfil(): Retornara el path de la foto de perfil asociada
- v. 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
- vi. Public Void InformacionUsuario(): Con esta opción se mostrará la información del usuario, siempre y cuando el usuario sea público
- vii. Public Void Add_Song_To_My_List(Canción canción): Con esta opción el usuario agregara a su lista de canciones agregadas la canción
- viii. Public Void Add_Video_To_My_List(Video video): Con esta opción el usuario agregara a su lista de videos agregadas el video
- ix. Public Void Add_Playlist_To_My_List(Playlist playlist) Con esta opción el usuario agregara a su lista de playlists agregadas
- x. Public string Get_Name(): Retorna el nombre del usuario
- xi. Public int Get_Edad(): Retorna la edad del usuario

- xii. `Public bool Get_TipoUsuario():` Retorna el tipo de usuario (público o privado)
- xiii. `Public bool Get_Premium():` Retorna si el usuario es premium
- xiv. `Public string Get_Nickname():` Retorna el nickname del usuario
- xv. `Public string Get_Password():` Retorna la contraseña del usuario
- xvi. `Public List<String> Get_GenerosSeguidos():` Retorna la lista de los géneros que sigue el usuario
- xvii. `Public List<Multimedia> Get_Favoritos():` Retorna la lista de favoritos
- xxviii. `Public List<Multimedia> Get_Recomendados():` Retorna la lista de recomendados
- xix. `Public List<Usuarios> Get_Seguidores():` Retorna la lista de seguidores
- xx. `Public bool Get_Admin():` Retorna si el usuario es Administrador o no
- xxi. `Public int Get_Contador():` Retorna la cantidad de intentos que tiene para el mini juego
- xxii. `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
- xxiii. `Public Void Cambiar_Dato_Booleano(int caso):` Cambiará el tipo de usuario, si es premium o si es administrador según sea el caso
- xxiv. `Public Void Cambiar_Dato_Edad(int edad):` Cambiará la edad del usuario usando el parámetro
- xxv. `Public Void Lose_BonusGame():` Descontará una oportunidad para jugar al mini juego
- xxvi. `Public Void Add_FavoritosCan(Canción cancion):` Agregara la canción a la lista de favoritos, si ya estaba, la quitará
- xxvii. `Public Void Add_FavoritosVid(Video video):` Agregara el video a la lista de favoritos, si ya estaba, la quitará
- xxviii. `Public List<Canciones> Get_Lista_Canciones_Fav():` Retorna la lista favoritas de canciones del usuario
- xxix. `Public List<Canciones> Get_Lista_Videos_Fav():` Retorna la lista favoritas de videos del usuario
- xxx. `Public Void Consultar_Favoritos():` Mostrará todos los archivos multimedia que están en favoritos
- xxxi. `Public List<Canciones> Get_ListaCancionesAgregadas():` Retorna la lista de canciones agregadas
- xxxii. `Public List<Video> Get_ListaVideosAgregados():` Retorna la lista de videos agregadas
- xxxiii. `Public List<Playlist> Get_ListaPlaylistAgregadas():` Retorna la lista de playlists agregadas

- 3) 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 de este.

3.1) Atributos:

- i. Private Lista objeto Canciones Lista_Canciones: Contiene todas las canciones
- ii. Private Lista objeto Videos Lista_Videos: Contiene todos los videos
- iii. Private Lista objeto Usuarios Usuario: Contiene todos los usuarios
- iv. Private Lista objeto canción ColaDeReproduccionCancion: Contiene todas las canciones en cola
- v. Private Lista objeto video ColaDeReproduccionVideo: Contiene todos los videos en cola
- vi. Lista de playlist multimedia: Guarda todas las playlists de canciones y videos
- vii. Lista de playlist de personas: Guarda todas las playlist de personas

3.2) Métodos

- i. Public Void Quitar_Cancion(Canción canción): Sacara la canción de la plataforma
- ii. public Void Quitar_Video(Video video): Sacara el video de la plataforma
- iii. public Void Add_Cancion(Canción canción): Agregara la canción al sistema
- iv. public void Add_Video(Video video): Agregara el video al Sistema
- v. public void Add_Playlist(Playlist playlist): Agrega una playlist al sistema
- vi. public void AgregarUsuario(Usuario usu): Agregara el usuario al sistema
- vii. public void CrearPersona(Persona persona): Agregara la persona al sistema
- viii. public void RankingDeCanciones(): Mostrara el ranking de canciones
- ix. public void RankingDeVideos(): Mostrara el ranking de videos
- x. public void Reproducir_Multimedia(Multimedia multi): Reproducirá el archivo multimedia con el reproductor por defecto del equipo, para futuro se implementara este método en el mismo programa.
- xi. 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
- xii. 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

- xiii. `public void Modificar_Playlist():` El usuario podrá modificar una de sus playlists creadas, puede agregar canciones, quitarlas o borrar la playlist
- xiv. `public void ColaReproduccionCanciones():` El usuario puede agregar las canciones a la cola de reproducción
- xv. `public void ColaReproduccionVideos():` El usuario puede agregar los videos a la cola de reproducción de videos
- xvi. `public void PortadaCancion(Canción canción):` El usuario puede agregar la portada a la canción con una imagen
- xvii. `public void PortadaVideo(Video video):` El usuario puede agregar la portada a la canción con un video
- xviii. `public void MostrarPortadaCancion(Canción canción):` El usuario puede ver la portada de la canción (Implementar a futuro junto al forms)
- xix. `public void MostrarPortadaVideo(Video video):` El usuario puede ver la portada del video (Implementar a futuro junto al forms)
- xx. `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 numerada 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.
- xxi. `Public void ComprobarUsuarios(string nombre, string contra):` Verifica si al ingresar al sistema, el usuario ingresó bien su nickname y contraseña
- xxii. `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
- xxiii. `Public Void Save():` Guarda todos los datos y cambios en el programa como canciones, videos, playlists, usuarios y personas incluidas
- xxiv. `Public List<Canciones> Get_ListaCanciones():` Retorna la lista de canciones
- xxv. `Public List<Video> Get_ListaVideos():` Retorna la lista de videos
- xxvi. `Public List<Usuario> Get_Usuarios():` Retorna la lista de usuarios
- xxvii. `Public List<Canciones> Get_ColaCanciones():` Retorna la cola de canciones
- xxviii. `Public List<Video> Get_ColaVideos():` Retorna la cola de videos
- xxix. `Public List<Playlist> Get_Playlists():` Retorna las playlists

xxx. Public List<Personas> Get_Personas(): Retorna la lista de personas

- 4) 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

- i. Protected int Duración: Indica la duración del archivo (se instaló librería NuGet WMP.Lib para la ejecución de este atributo)
- ii. Protected string Título: Indica el título del archivo
- iii. Protected string Fecha_Inclusion: Indica la fecha en la que se incluye el archivo a la plataforma
- iv. Protected long Size: Indica el tamaño del archivo en bytes
- v. Protected string Estudio: Indica el estudio donde proviene el archivo
- vi. Protected string CarpetaArchivo: Indica el directorio del archivo
- vii. Protected string Fecha_Publicacion: Indica la fecha de publicación del archivo
- viii. Protected string Descripción: Indica la descripción del archivo
- ix. Protected string Portada: Indica el directorio de la portada del archivo
- x. Protected int NumeroReproducciones: Indica el número de reproducciones del archivo
- xi. Protected List<string> Comentarios: Indica los comentarios que tiene el archivo
- xii. Protected List<int> Ranking: Indica las calificaciones que tiene el archivo
- xiii. Protected List<string> Genero: Indica los géneros de la canción
- xiv. Protected List<Usuario> Seguidores: Indica los Usuarios que siguen al archivo

4.2) Metodos

- i. Constructor: Con el constructor se generara el objeto multimedia de acuerdo lo que ingresa el usuario
- ii. Public string Get_Portada(): Retorna el directorio de la portada
- iii. Public void Agregar_Comentarios(string comentario): Agrega el comentario a la lista de comentarios
- iv. Public void Seguir(Usuario usu): Agrega al usuario a la lista de seguidores
- v. Public void Ranked(int estrellas): Agrega la calificación (Estrellas) a la lista de calificaciones
- vi. Public double Get_Mean_Tier(): Entrega la calificación promedio del archivo multimedia
- vii. Public void Cambiar_Ubicacion(string newpath): Cambia el directorio del camino con newpath

- viii. `Public void Sumar_Reproducciones():` Sumará una reproducción más al archivo
 - ix. `Public string Get_Titulo():` Retornará el título del archivo
 - x. `Public int Get_Duracion():` Retornará la duración del archivo
 - xi. `Public string Get_FechaInclusion():` Retornará la fecha de inclusión
 - xii. `Public long Get_Size():` Retornará el tamaño del archivo
 - xiii. `Public string Get_Estudio():` Retornará el estudio de donde proviene el archivo
 - xiv. `Public string Get_Carpeta_Archivo():` Retornará el path de donde proviene el archivo
 - xv. `Public string Get_Fecha_Publicacion():` Retorna la fecha en la que se public el archive
 - xvi. `Public string Get_Descripcion():` Retorna la descripción del archivo
 - xvii. `Public int Get_Reproducciones():` Retornará el número de reproducciones que tiene el archivo
 - xviii. `Public List<String> Get_Comentarios():` Retorna los comentarios que tiene el archivo multimedia
 - xix. `Public List<int> Ranking():` Retorna la lista de notas del archivo multimedia
 - xx. `Public List<string> Get_Genero():` Retorna los géneros que tiene el archivo multimedia
 - xxi. `Public List<Usuario> Get_Seguidores():` Retorna la lista de Usuarios que sigue la canción
- 5) 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

- i. `Private string Letra:` Indica la letra de la canción
- ii. `Private string Álbum:` Indica el álbum/disco de la canción
- iii. `Private List<Personas> PersonasCancion:` Indica los artistas de la canción

5.2) Métodos

- i. Constructor: Se generará un objeto de la clase canción según las características dadas por el usuario
- ii. `Public void Información():` Dará toda la información de la canción
- iii. `Public string Get_Letra():` Retornará la letra de la canción
- iv. `Public string Get_Album():` Retornará el álbum de la canción
- v. `Public List<Personas> Get_Artistas():` Retornará los artistas de la canción

- 6) 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

- i. Private string Calidad: Indica la calidad del video
- ii. Private List<Personas> PersonaVideo: Indica las personas que están involucradas en la producción del video

6.2) Métodos

- i. Constructor: Creará un objeto de la clase videos con los datos que le ingresa el usuario
- ii. Public void Información(): Dara toda la informacion del video
- iii. Public string Get_Calidad(): Retorna la calidad del video
- iv. Public string Get_Staff(): Retorna la lista de personas involucradas en el video

- 7) Clase Playlist: Clase que almacena listas de canciones o videos que luego son guardadas en el almacenamiento

7.1) Atributos

- i. Private List<Canciones> Playlist_Canciones: Indica las canciones de la playlist
- ii. Private List<Video> Playlist_Videos: Indica los videos de la playlist
- iii. Private string nombre: Indica el nombre de la playlist
- iv. Private List<Usuarios> Seguidores: Indica los usuarios que siguen esta playlist

7.2) Métodos

- i. Constructor: Generara un objeto de la clase playlist acorde a los datos ingresados por el usuario
- ii. Public void Seguir(Usuario Usu): Se agregara al usuario a la lista de seguidores
- iii. Public void MostrarCanciones(): Mostrara las canciones de la playlist
- iv. Public void MostrarVideos(): Mostrara los videos de la playlist
- v. Public List<Canciones> GetPlaylistCanciones(): Retornara la lista de canciones
- vi. Public List<Video> GetPlaylistVideos(): Retornara la lista de videos
- vii. Public string Get_Name(): Retorna el nombre de la playlist
- viii. Public List<Personas> Get_Seguidores(): Retorna la lista de seguidores
- ix. Public Add_Music_To_Playlist(Canción canción): Agrega la canción a la lista de canciones
- x. Public Add_Video_To_Playlist(Video video): Agrega el video a la lista de videos

- 8) Clase Persona: Clase que tiene la información de las personas involucradas ya sea en una canción o en un video

8.1) Atributos

- i. Private string Nombre: Indica el nombre de la persona
- ii. Private string Apellido: Indica el apellido de la persona
- iii. Private int Edad: Indica la edad de la persona
- iv. Private string Sexo: Indica el sexo de la persona
- v. Private string PuestoTrabajo: Indica el puesto de trabajo de la persona
- vi. Private List<Usuario> Seguidores: Indica los usuarios que siguen a la persona

8.2) Metodos

- i. Constructor: Se genera un objeto de la clase persona con los datos ingresados por el usuario
- ii. Public void Seguir(Usuario usu): Agrega al usuario usu a la lista de seguidores
- iii. Public void Informacion_Persona(Lista<video> Videos, Lista<canciones> Canción): Muestra la informacion personal de la persona y además sus trabajos realizados
- iv. Public string Get_Name(): Retorna el nombre de la persona
- v. Public string Get_Apellido(): Retorna el apellido de la persona
- vi. Public int Get_Edad(): Retorna la edad de la persona
- vii. Public string Get_Sexo(): Retorna el sexo de la persona
- viii. Public string Get_PuestoTrabajo(): Retorna el puesto de trabajo de la persona

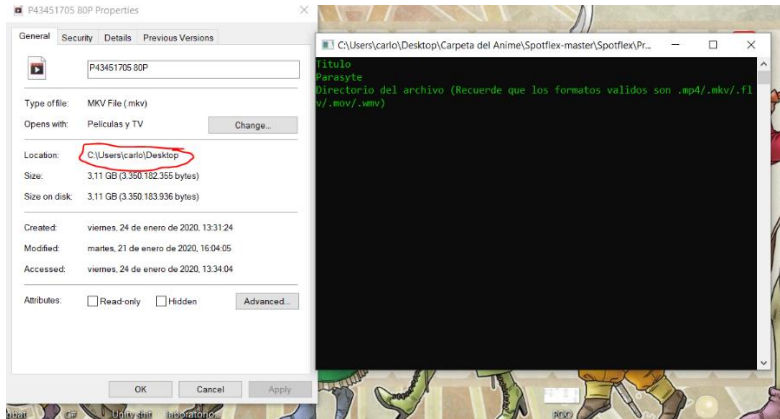
- 9) Clase Importacion-Exportacion: Esta clase se encarga de llevar todas las canciones y videos a almacenamiento, también se encarga de descargar las canciones y quitarlas del sistema

9.1) Métodos

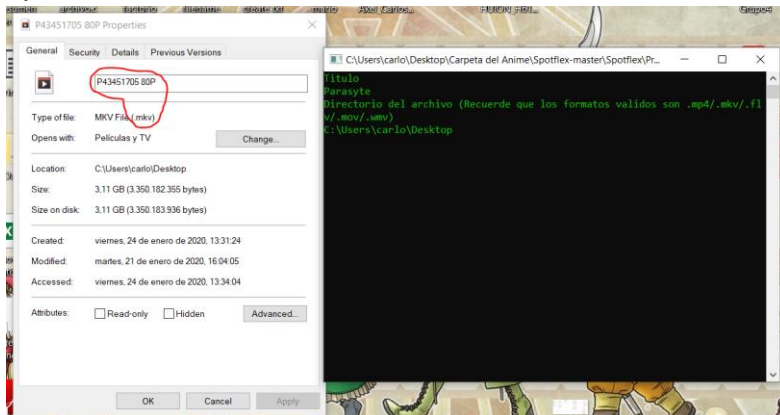
- i. Public void AgregarCanción (Almacenamiento alm, Usuario usu): Le pide al usuario usu todos los datos de la canción y la agrega a la plataforma
- ii. Public void AgregarVideo (Almacenamiento alm, Usuario usu): Le pide al usuario usu todos los datos del video y la agrega a la plataforma
- iii. Public void QuitarCanción (Canción canción, Almacenamiento alm): Borra la Canción de la lista de Canciones de la Clase almacenamiento.
- iv. Public void QuitarVideo (Video videos, Almacenamiento alm): Borra el Video de la lista de la lista de la Clase almacenamiento.
- v. Public void Descarga(Canción canción): Descarga la canción a la carpeta de descargas del programa

10)Mini guía de Lectura de los Archivos

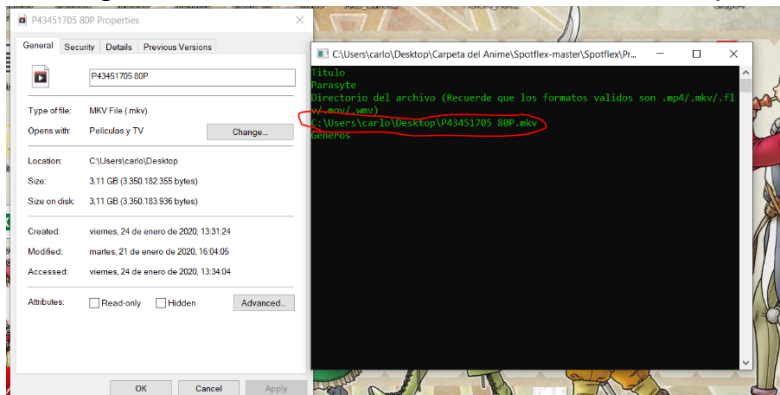
1.-Abre las propiedades del archivo y mira su ubicación.



2.-Luego de ingresar la ubicación escribe un “\” y el nombre del archivo con su tipo dentro de la consola.



3.-Luego de tener todos los datos de la ubicación presione Enter.



11)Consideraciones:

Hay un usuario que es administrador para probar sus funcionalidades, su nickname es "dark" y su clave es "lol", aparte esta clase de usuarios tienen la facultad de eliminar cualquier usuario.

El usuario free solo puede agregar como máximo 5 canciones y videos a la plataforma. Por otro lado esta clase de usuarios al tratar de importar más canciones puede jugar un minijuego y si lo gana será usuario premium, sin embargo, solo tienen 3 oportunidades.

12)Ideas a Futuro

- i) Los Usuarios pueden hacer una Tier List de canciones y/o videos
- ii) Agregar las portadas a los videos y canciones usando forms
- iii) Reproductor propio con opciones para pausar, adelantar, etc.
- iv) Mejorar el minijuego de tal forma que en vez de mostrar la letra escucha un fragmento de la cancion
- v) Obtener los datos de las canciones y videos de una forma más amigable y rápida
- vi) Ver que al traspasar el programa, se traspase las canciones y videos que este tiene (esto más que nada porque el gitignore no dejaba traspasarlos)