

Conceptualización



1. Definición de roles

- a. Encargado del repositorio → Jacobo Giraldo
- b. Encargado del trabajo interno → Matias Eduardo Leguer
- c. Encargado de relaciones exteriores → Diego Pablo Pinochet

2. Nombre

- a. El nombre que definimos para la app fue FyBuZz debido a que nos basamos en el origen de la música la cual se da principalmente en Grecia. FyBuZz nace de el líder de los dioses griegos de la música y el arte, llamado Febo en roma o Apolo (Φοῖβος en griego) en Grecia y que se pronuncia (fiibas) lo que nos llevó a escribirlo como FyBuZz la cual para una plataforma universal suena como Bass (Bajo en ingles) y le da un toque musical a nuestra aplicación.

3. Clases, interfaces y sus objetivos

- a. Multimedia
 - i. **Objetivo:** Definir características comunes de los archivos multimedia, incluyendo métodos para agregar, verificar y borrar elementos de este tipo. Además desde esta se podrá a acceder a la información de cada objeto.
 - ii. **Atributos:** Tiene ciertos atributos que son bases para las clases derivadas, tales como: name, format, gender, publisherYear, ranking, entre otros.
 - iii. **Métodos:** Tenemos tres métodos en la clase, denominados AddMult(), DeleteMult() y VerifyMult(). Estos van a ser utilizados para poder obtener información de manera más clara cuando se realice el proyecto y poder verificar que esta informacion este incorporada de manera correcta.
- b. Song: Multimedia
 - i. **Objetivo:** Definir características específicas de objetos de clase multimedia con formato “.mp3” o “.wav”.
 - ii. **Atributos:** tiene tres atributos propios únicos que son: artist, album y discography.
 - iii. **Métodos:** Hata el moemnto tenemos tres métodos que son ListInfoSongs(), InfoRep() y SearchedInfoSong() que entregan respetivamente, una lista con la iformacion relevante de la cancion, un numero con la cantidad de reproducciones de dicha cancion y un texto con la informacion de la cancion que esta siendo buscada..
- c. Video: Multimedia
 - i. **Objetivo:** Definir características específicas de objetos de clase multimedia con formato “.mp4” o “.mov”
 - ii. **Atributos:** Tiene varios atributos propios, estos son: videoDimension, quality, memorySize ,category, description, rated, image (Capacidad de otorgarle una imagen al álbum o Playlist).

- iii. **Métodos:** Por ahora tenemos tres metodos que cumplen las mismas funciones que los tres metodos de la clase Song, sin embargo con algunas diferencias de sus atributos que los diferencian.
- d. Playlist
 - i. **Objetivo:** Definir características específicas de objetos de clase playlist, en donde vamos a poder guardar los videos y canciones en listas. Estos se encontrarán separados debido a su formato.
 - ii. **Atributos:** Tiene dos atributos, estos son: formato y namePlayList.
 - iii. **Métodos:** Por ahora existen dos métodos, estos son: DisplayInfoPlayList(),InfoPlayList() y AddToPlaylist(). Sus objetivos respectivamente son: Mostrar la información de cada playlist de manera basica y añadir canciones o videos (dependiendo del formato) a los diccionarios que guardan respectivamente las playlist de canciones o videos.
- e. DataBase
 - i. **Objetivo:** El objetivo de esta clase consiste en tener un lugar en donde se encontrará todas las canciones, videos, playlists y todos los usuarios que ingresen a nuestra plataforma, es decir, esta clase guarda toda la información para que después puedas ser accedida de manera sencilla por las demás clases.
 - ii. **Atributos:** Los atributos que va a tener esta clase son cuatro listas y un diccionario: listSongsGlobal, listVideosGlobal, listPlaylistsGlobal, Gender (que contendrán todas las canciones, videos y playlists) y usersDataBase que es el diccionario que guarda toda la información del usuario relacionada a su cuenta en si .
 - iii. **Metodos:** Los métodos que tenemos en esta clase son principalmente aquellos que usan la serialización de las demás clases para guardar y cargar toda la información a otros archivos para poder tener la información incluso si cerramos el programa, dentro de estos archivos está incluida la información de todas los usuarios, canciones, videos, playlists. Ademas tambien estan los métodos AddUser, GetData, LogIn y AddMult que son los encargados de manejar toda la información del usuario y su interacción dentro del programa y la inserción de nueva multimedia por parte de estos usuarios.
- f. Server
 - i. **Objetivo:** Establecer una interacción entre DataBase y User mediante eventos, de esta manera tendremos información de todos los usuarios que ingresen en la plataforma.
 - ii. **Atributos:** Por ahora tenemos un atributo, este es database, tendrá la función de obtener mediante un “getter” el diccionario de la clase DataBase.
 - iii. **Métodos:** Este tiene dos métodos: Register(), su función es registrar al usuario en la plataforma y verifica que no existan más de un usuario con el mismo username o email. ChangePassword(), su objetivo es realizar el cambio de contraseña si es que el usuario lo necesita.

g. ProfilePreferences: DataBase

- i. **Objetivo:** Establecer las preferencias de cada perfil, estas se basan en las búsquedas y reproducciones que se hacen de todos los archivos multimedia de la plataforma.
- ii. **Atributos:** Por ahora tenemos dos atributos, estos son searchHistorySongs y searchHistoryVideos. Sus objetivos son establecer una lista de Playlist (de canciones o videos respectivamente) que su contenido será todas las búsquedas realizadas por el usuario.
- iii. **Métodos:** Los métodos establecidos son: DisplayHistory(), BrowserHistorySongs(), BrowserHistoryVideos(), ProfilePreferencesSongs(), ProfilePreferencesVideo(), ProfilePreferencesPlaylist(). Sus objetivos respectivamente son: Mostrar el historial completo de búsqueda y así poder generar Playlist con los gustos de cada perfil; Generar una Lista de las últimas 10 busquedas, "busquedas recientes". Los últimos 3 métodos entregarán la preferencia respecto al atributo de su respectiva clase, es decir, se le entrega un parámetro: nombre, artista, album, etc. y entrega la inclinación del perfil respecto a este. (diferenciando si es cancion, video o playlist).

h. User

- i. **Objetivo:** El objetivo de esta clase es poder tener distintas cuentas en nuestro proyecto, en donde cada usuario tendrá su mail, su username y contraseña única. También, poder diferenciar el tipo de usuario que se va a crear, ya sea premium, standard, administrador. Además, desde esta se obtendrá la verificación del usuario, sus seguidores y sus seguidos.
- ii. **Atributos:** Los atributos van a ser la información básica de las cuentas: username, email, password, accountType, followers, following, verified. Cabe resaltar que los distintos tipos de usuarios funcionan de manera distinta y tienen sus privilegios, es decir, el admin puede eliminar y banear usuarios (aunque no tiene motivo para eliminar/banear usuarios, sin embargo el metodo se implementa de manera aleatoria), los usuarios premium pueden tener más de un solo perfil y escuchar canciones que ellos eligen y los standard solo pueden escuchar música aleatoria y tener un solo perfil.
- iii. **Métodos:** Por ahora tenemos tres métodos importantes para el funcionamiento del programa: CreateProfile(), GetVerification() y AccountSettings(). Ademas tenemos otros metodos que cumplen con funcionalidades no tan especiales, los cuales son AdminDeleteUser() y AdminBanUser().

i. Profile: User

- i. **Objetivo:** Crea perfiles, los que tendrán distintos tipos y funciones específicas dentro de la cuenta de la aplicación, tales como nombre de perfil, imagen de perfil, Playlist en cola, etc.
- ii. **Atributos:** Existen variedad de atributos dentro de esta clase, estos son: `profileName`, `profilePic`, `profileType`, `profileMail`, `playlistEnCola`, `playlistFavoritos`, `gender`, `age`. Todos tiene como objetivo principal diferenciar un perfil dentro de otro mediante características modificables por el usuario, sin embargo se debe hacer la diferencia que estas playlist se separan para videos o canciones.
- iii. **Métodos:** Tenemos distintos métodos para esta clase con distintas funcionalidades, dentro de estos se encuentran los siguientes: `ChangeName()`, `InfoProfile()`, entre otros. La utilidad de estos métodos se debe a que el perfil de un usuario tiene los permisos para poder modificar información de su cuenta, seguir a otras cuentas, etc.

j. Menu

- i. **Objetivo:** Esta clase tiene el principal objetivo de comunicarse con las otras clases y crear la interfaz en donde se va a poder ver el programa.
 - *Como no conocemos la materia todavía, no podemos desarrollar de forma concreta esta clase, pero escribimos lo que pensamos que debería tener.
- ii. **Atributos:** En esta clase por ahora hay un solo atributo que es la lista de todos los filtros que se pueden tener para realizar una búsqueda, y estos corresponder a los filtros en el enunciado general.
- iii. **Métodos:** Para los métodos tenemos solos dos por el momento ya que creemos que nos faltan conocimientos adecuados para generar y mostrar un menú de una aplicación, sin embargo tenemos `SearchEngine()` (la búsqueda de multimedia en sí) y `DisplayPlaylist()` (capacidad de mostrar la playlist en la plataforma) que son los que creemos que debería tener por ahora nuestro menú. Más adelante se generarán nuevos métodos que puedo mostrar todo lo que queremos implementar a partir de nuestras demás clases.

4. Funcionalidades extras

- a. **Restricción de edad:** Lee los lyrics de las canciones y oculta canciones con contenido explícito, al igual que los video con su rating.
- b. **Ponerle subtítulos a los videos,** a parte de los lyrics de las canciones.