

# Plan de gestión, análisis, diseño y memoria del proyecto

Spread Your Music

March 3, 2018

## Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
1.1	Resumen . . . . .	2
<b>2</b>	<b>Organización del proyecto</b>	<b>2</b>
2.1	Equipo . . . . .	2
<b>3</b>	<b>Plan de gestión del proyecto</b>	<b>3</b>
3.1	Procesos . . . . .	3
3.1.1	Procesos de inicio de proyecto . . . . .	3
3.1.2	Procesos de ejecución y control del proyecto . . . . .	3
3.1.3	Procesos técnicos . . . . .	4
3.2	Planes . . . . .	4
3.2.1	Plan de gestión de configuraciones . . . . .	4
3.2.2	Plan de construcción y despliegue del software . . . . .	5
<b>4</b>	<b>Análisis y diseño del sistema</b>	<b>8</b>
4.1	Análisis de requisitos . . . . .	8
4.1.1	Sistema . . . . .	8
4.1.2	Backend . . . . .	8
4.1.3	Android . . . . .	9
4.1.4	Web . . . . .	11
4.2	Diseño del sistema . . . . .	12

# 1 Introducción

## 1.1 Resumen

La aplicación a desarrollar consistirá en un reproductor de música en *streaming* inspirado en *Soundcloud* y *Spotify*.

Es una aplicación orientada a todo tipo de usuario, tanto a los músicos que están empezando su carrera en el mundo de la música como a cualquier persona aficionada a la música. Nuestra aplicación permitirá a los usuarios subir canciones, crear listas de reproducción o escuchar canciones utilizando un reproductor propio entre otras funcionalidades. La versión Android permitirá a los usuarios descargar las canciones para poder escucharlas sin necesidad de estar conectado a Internet. También incluirá características sociales, permitiendo a los usuarios seguir a sus artistas favoritos para ver las novedades que publican o suscribirse a listas de reproducción creadas por otros usuarios para enterarse de cambios en esta. El sistema también poseerá integración con redes sociales, así como la posibilidad de autenticación mediante cuenta de Google.

El usuario tendrá recomendaciones personalizadas para el usuario tanto basadas en su historial de reproducción como en geolocalización y facilitará al usuario encontrar canciones, pudiendo buscar canciones por categorías, autor, nombre, así como también mostrando canciones populares dentro de la aplicación.

Nuestra aplicación tendrá soporte web (en navegadores Chrome y Firefox) y como aplicación Android, de manera que el usuario pueda usarla en web, en el móvil o en ambas ya que incorpora un sistema de sincronización de manera que el usuario puede seguir escuchando en cualquier dispositivo la misma canción justa en el momento en el que la dejó.

## 2 Organización del proyecto

### 2.1 Equipo

El equipo está formado por 7 estudiantes de Ingeniería Informática: Jorge Aznar López, ángel Cañal Muniesa, Abel Chils Trabanco, Alicia Yasmina Albero Escudero, óscar Fraca Ferrández, Alexandru Ioan Oarga Hategan y Jorge Pinilla López.

De estos, cinco poseen conocimientos sobre programación Web (*Frontend*) así como práctica en el desarrollo del *Backend* de una aplicación Web. Por otro lado, todos tienen experiencia en el diseño de base de datos, así como experiencia con la programación sobre la plataforma Android. Además, los integrantes del grupo ya poseen experiencia trabajando juntos, lo cual facilitará la comunicación entre ellos.

Para desarrollar esta aplicación se han creado 3 grupos y cada uno se centrará en una parte del desarrollo. Un equipo se encargará del *Backend*, otro de la interfaz Web (o *Frontend*) y por último otro de la aplicación Android. La asignación de los integrantes del grupo a cada una de las partes se ha realizado en base a la experiencia que posea el integrante en cuestión en dicha área.

El equipo de *Backend* está formado por óscar, ángel y Jorge Aznar. El equipo de la plataforma Android está formado por Abel y Yasmina. Por último, el equipo de *Frontend* está formado por Alexandru y Jorge Pinilla.

Para gestionar el proyecto se ha designado un coordinador general y coordinadores específicos dentro de cada área. El coordinador general es Abel, el coordinador de *Backend* es ángel, el de Android es Abel y el de *Frontend* es Jorge Pinilla

## 3 Plan de gestión del proyecto

### 3.1 Procesos

#### 3.1.1 Procesos de inicio de proyecto

La aplicación móvil funcionará en dispositivos con Android 5.0, la aplicación web funcionará en los navegadores Firefox y Chrome. Para realizar las pruebas de la aplicación móvil se requerirá de un dispositivo con Android 5.0. Por otro lado, para las pruebas de la versión web será necesario un ordenador con el navegador instalado (Chrome o Firefox). El sistema se funcionará sobre un clúster, que contará con un almacenamiento bruto de 20GB. Se ha estimado este tamaño para el clúster teniendo en cuenta la arquitectura del sistema.

#### 3.1.2 Procesos de ejecución y control del proyecto

Las comunicaciones del grupo se van a realizar mediante un grupo en la aplicación de mensajería instantánea *WhatsApp* para tratar temas y comunicaciones poco importantes y eventuales. Sin embargo, para temas que deban ser permanentes y/o deba quedar constancia de esta comunicación, se usarán las *Issues* o incidencias de la plataforma de alojamiento de los proyectos *GitHub*. En esta plataforma de alojamiento se almacenará el código fuente del sistema desarrollado y todos los documentos generados durante el desarrollo. Entre estos documentos se encuentra, por ejemplo, las actas de las reuniones con los clientes, que serán redactadas por al menos un miembro del equipo durante dicha reunión. De forma similar se registrarán las reuniones del equipo, los contenidos y las decisiones que puedan tomarse en esas reuniones.

En todas las actas, tanto de reuniones del equipo como reuniones con los clientes se incluirá al menos la fecha y hora de la reunión, la duración de la reunión, los miembros presentes en la reunión y los temas y decisiones que se tomen en la reunión.

Todas las semanas tienen un conjunto de tareas asociadas. Al final de cada semana, el responsable de cada proyecto revisará las tareas que se han realizado esa semana y, si hay tareas que no se han cumplido, se asignarán automáticamente para la siguiente semana, siendo estas tareas las primeras que se deberán hacer. Además, cada semana el responsable del subproyecto revisará cuantas tareas se han realizado para la siguiente iteración como métrica de monitorización de la desviación según el plan original. Tras esta revisión el responsable del subproyecto asignará las tareas de la semana a todos los miembros del equipo que puedan trabajar esa semana.

Durante el desarrollo del proyecto puede haber problemas y disputas entre los miembros del equipo. Para tratar de resolverlos el responsable del subproyecto será el primero en mediar entre los miembros en disputa y, si hay alguna razón que haga imposible esta mediación será el resto del

equipo quien deberá mediar.

### 3.1.3 Procesos técnicos

Las herramientas utilizadas tanto para desarrollo del software (construcción, pruebas y despliegue) serán IntelliJ Idea y Android Studio debido a que ambas dan soporte para los lenguajes de programación Java y Kotlin además de su buena integración con git, herramienta que se utilizará para el control de versiones. En el caso especial del Frontend se usará como herramienta WebStorm por razones similares a los anteriores.

Para asegurar la calidad del software, al mismo tiempo que se vaya desarrollando se irán creando pruebas unitarias sobre el mismo con JUnit.

Siguiendo una metodología de diseo incremental, cada dos semanas se tiene planificada la entrega interna de una versión nueva del software. Estas versiones se basa en la versión anterior aadiéndole ciertas funcionalidades.

## 3.2 Planes

### 3.2.1 Plan de gestión de configuraciones

El código desarrollado deberá seguir una serie de estándares, siendo los siguientes los aceptados. Para el nombrado de métodos, variables, clases... se utilizará la filosofía **CamelCase** en los proyectos de Android y *Backend* y la filosofía **snake\_case** en el caso del *Frontend*. Además, para la escritura de código se seguirá el estándar recomendado por Google en el caso de Android y *Backend* (<https://google.github.io/styleguide/javaguide.html>) y el recomendado por la W3Schools para *Frontend* ([https://www.w3schools.com/html/html5\\_syntax.asp](https://www.w3schools.com/html/html5_syntax.asp)). Como medida adicional, para permitir que un mayor número de dispositivos puedan visualizar correctamente la página web, se va a desarrollar la web de forma Responsive.

Para asegurar que el desarrollo sigue los estándares de calidad y de nombrado, el coordinador de cada proyecto se encargará de la revisión de los commits de su proyecto y la aceptación o no aceptación de código nuevo. Del despliegue y la puesta en marcha del sistema se encargará Jorge Pinilla y de la correcta administración del sistema de control de versiones (VCS) Git Ángel Cañal.

Este Sistema de Control de Versiones está compuesto por 4 repositorios, uno en el que se alojan todos los documentos del sistema (e.g. "Propuesta Económica" o "Plan de Gestión") y 3 repositorios más, uno por cada uno de los proyectos (Android, *Backend* y *Frontend*). Para intentar reducir conflictos o problemas, sólo los miembros más experimentados tendrán acceso total a los repositorios (estos son Ángel Cañal y Abel Chils). El resto del equipo sólo tendrá acceso al repositorio de Documentos y al del proyecto en el que esté trabajando.

Para reducir problemas causados por modificación concurrente del mismo fichero de código fuente, se va a usar un flujo de trabajo en Git denominado Git Flow (<https://danielkummer.github.io/git-flow-cheatsheet/>) por el que hay dos ramas principales, una para la última versión estable (*master*) y otra para la versión en desarrollo (*develop*). Además, para cualquier nueva característica a añadir habrá que crear una nueva rama en Git y, cuando se haya terminado, se volcará esa rama a la rama

de desarrollo. No se permite envíar código directamente a las ramas *develop* ni *master*. Como este flujo de trabajo puede resultar complejo, se simplifica haciendo uso del comando "git flow".

Como el sistema Git soporta la gestión de incidencias, se va a aprovechar este sistema para gestionar todas las situaciones inesperadas que pudieran suceder. Además, con la gestión de incidencias se va a utilizar una incidencia especial, llamada "tarjeta" por la que se va a poder gestionar las tareas pendientes por hacer. Para ello, el responsable de cada equipo creará una incidencia por tarea a realizar (lo más pequeña posible). Estas incidencias podrán estar en 3 estados, "Abierto", "En progreso" y "Terminado" además de estar asignado a un punto en el tiempo en el que deberán estar terminadas por completo. Para pasar de "En progreso" a "Terminado" el coordinador del equipo debe dar el visto bueno e incorporar el nuevo código al resto (rama *develop*).

### 3.2.2 Plan de construcción y despliegue del software

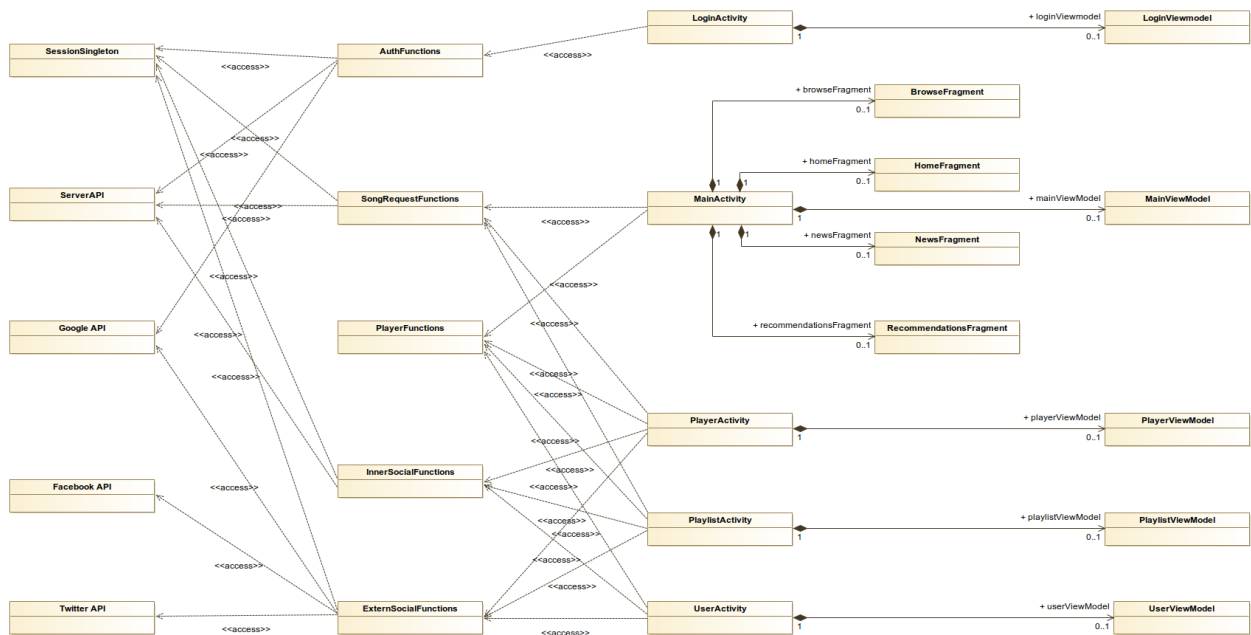
- Cómo se construye e integra el software: si hay scripts de construcción automatizada o no (en ese caso qué se usa, y cómo se garantiza que todos los participantes compilan igual y con las mismas dependencias), qué se incluye en la construcción (descarga y actualización de dependencias, compilación, ejecución de tests automáticos...) y cada cuánto se construye (compila, integra, prueba) el sistema completo, cómo se configuran los computadores de los desarrolladores.
- Cómo se despliega el software más allá de las máquinas de desarrollo: contenedores, máquinas virtuales, servidor en cloud etc. y cómo se configuran esos entornos (rutas, usuarios y contraseñas, puertos y otros elementos).

#### Arquitectura de la aplicación Android

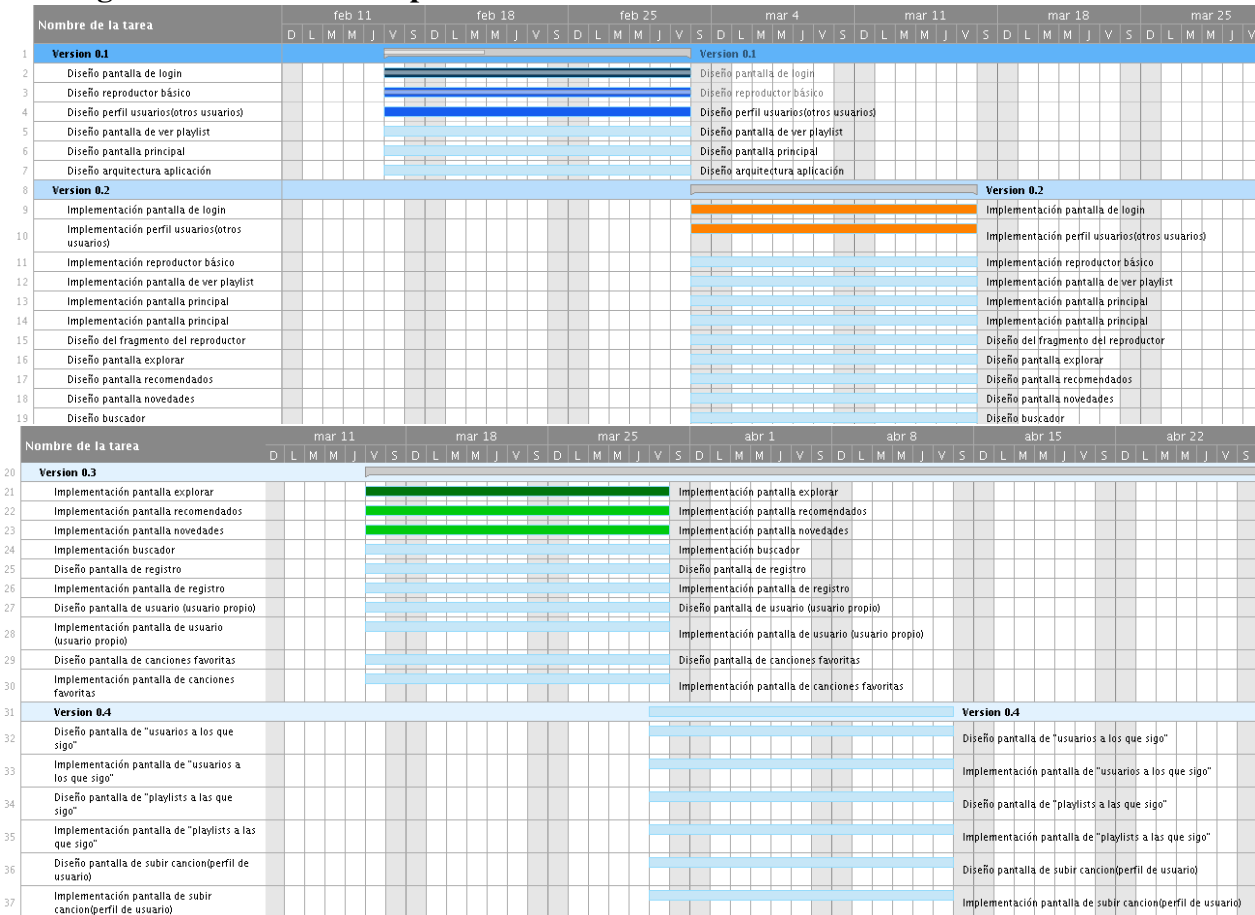
La aplicación android está estructurada en 3 capas, aunque a su vez una de estas está dividida en dos.

En la primera de estas se encuentran las apis con servicios externos (backend, google, facebook y twitter) y una clase que se encargará de mantener la sesión de la aplicación (incluyendo el manejo de la persistencia de esta sesión). Luego existe una segunda capa en la cual se encuentran funciones para manipular la capa citada anteriormente. Por último está la capa que se encarga del manejo de la interfaz gráfica. Esta última capa se divide a su vez en dos. Una de ellas se encarga del manejo de la interfaz en las pantallas y la otra se encarga de la persistencia de los datos que se están mostrando. Esta segunda capa es requerida para evitar tener que repetir alguna operaciones cuando se cambia la orientación del dispositivo o cuando se apaga la pantalla.

#### Diagrama de clases de la aplicación Android



## Diagrama de Gantt de la aplicación Android



Nombre de la tarea		abr 8							abr 15							abr 22							abr 29							may 6							may 13							may 20																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
		D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
38	Version 0.5																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						</

## Versiones de la aplicación Android

Versión 0.1 (3 marzo) Diseo pantalla de login Diseo reproductor básico Diseo perfil usuarios(otros usuarios) Diseo pantalla de ver playlist Diseo pantalla principal (Aparecerán recomendaciones, novedades de artistas o playlists a las que se esté suscrito y canciones,playlists o artistas populares). Diseo arquitectura aplicación

Versión 0.2 (17 marzo) Implementación pantalla de login Implementación perfil usuarios(otros usuarios) Implementación reproductor básico Implementación pantalla de ver playlist Implementación pantalla principal (Aparecerán recomendaciones, novedades de artistas o playlists a las que se esté suscrito y canciones,playlists o artistas populares). Diseo del fragmento del reproductor que aparecerá cuando se está escuchando música en una pantalla que no es la pantalla de reproductor. Diseo pantalla explorar (pantalla en la que aparecen todos los generos, artistas del momento y recomendaciones basadas en la ubicación). Diseo pantalla recomendados Diseo pantalla novedades Diseo buscador

Versión 0.3 (31 marzo) Implementación pantalla explorar (pantalla en la que aparecen todos los generos, artistas del momento y recomendaciones basadas en la ubicación). Implementación pantalla recomendados Implementación pantalla novedades Implementación buscador Diseo pantalla de registro Implementación pantalla de registro Diseo pantalla de usuario (usuario propio) Implementación pantalla de usuario (usuario propio) Diseo pantalla de canciones favoritas Implementación pantalla de canciones favoritas

Versión 0.4 (14 abril) Diseo pantalla de "usuarios a los que sigo" Implementación pantalla de "usuarios a los que sigo" Diseo pantalla de "playlists a las que sigo" Implementación pantalla de "playlists a las que sigo" Diseo pantalla de subir cancion(perfil de usuario) Implementación pantalla de subir cancion(perfil de usuario)

Versión 0.5 (28 abril) Diseo pantalla de mis playlists(perfil usuario) Implementación pantalla de mis playlists(perfil usuario) Diseo pantalla de mis canciones(perfil usuario) Implementación pantalla de mis canciones(perfil usuario)

Versión 0.6 (última) (12 mayo) Integración del reproductor en todas las pantallas Reproductor avanzado(ecualizador + letras)

## **4 Análisis y diseño del sistema**

### **4.1 Análisis de requisitos**

#### **4.1.1 Sistema**

##### **Requisitos no funcionales**

1. El sistema tendrá una versión para Android y otra para Web.
2. El sistema soporta los ficheros MP3, WAV, OGG.
3. Existe un servidor para el almacenamiento de canciones.
4. La aplicación móvil soportará Android 5.0 y la Web la última versión de Firefox y Chrome.

#### **4.1.2 Backend**

##### **Requisitos funcionales**

1. El sistema se compone de Usuarios, Canciones y Listas de reproducción.
2. El sistema permite 3 tipos de usuarios:
  - Usuario registrado (usuario que esté registrado en el sistema)
  - Usuario no registrado (que es el cual el sistema no puede identificar con ninguna cuenta)
  - Usuario administrador (aquel que es como un usuario registrado pero unos privilegios especiales, como es verificar cuentas)
3. Una canción se compone de un título, un audio, la transcripción de dicho audio, el país de la canción, cuántos me gusta tiene y el número de reproducciones.
4. Una lista de reproducción o categoría es una lista de canciones generadas por el sistema agrupadas por género, éxito, país, situación para la que es propicia o generada por el usuario.
5. Un usuario registrado se compone de un nombre, un nick único (es decir, que dos usuarios no pueden tener el mismo nick), un correo electrónico, una contraseña para acceder a la aplicación, la fecha de nacimiento, una biografía, una foto de perfil, qué canciones ha reproducido, cuántos me gusta han recibido las canciones que ha subido, cuántos me gusta ha dado, sus canciones, el país y las redes sociales que quiera añadir de forma opcional (Twitter, Facebook e Instagram).
6. El sistema permite obtener todos los datos de una canción.
7. El sistema permite que los usuarios se registren en el sistema.
8. El sistema permite que los usuarios se identifiquen tanto con usuario y contraseña así como con una cuenta de Google.
9. El sistema permite que los usuarios registrados registren nuevas canciones en el sistema.
10. El sistema permite que los usuarios registrados creen y borren listas de reproducción formadas por canciones que pueden ser o no del propio usuario, las cuales serán públicas.



11. El sistema permite que los usuarios registrados añadan una canción a una lista de reproducción que haya creado él mismo.
12. El sistema permite obtener todas las canciones de una lista de reproducción.
13. El sistema permite que los usuarios registrados tengan una lista de canciones favoritas, la cual será privada.
14. El sistema permite que los usuarios registrados aadan o eliminen canciones de favoritos.
15. El sistema permite que los usuarios registrados obtengan la lista de sus canciones favoritas.
16. El sistema permite obtener las letras de una canción.
17. El sistema permite obtener recomendaciones (canciones, listas de reproducción y usuarios), canciones más populares, novedades sobre los usuarios o listas de reproducción que un usuario sigue.
18. El sistema permite que los usuarios registrados modifiquen su nombre, su correo electrónico, su contraseña, su fecha de nacimiento, su biografía, su foto de perfil, su país y sus redes sociales.
19. El sistema permite que los usuarios registrados sigan a una lista de reproducción.
20. El sistema permite buscar canciones mediante palabras clave (título), género, país, me gusta, reproducciones o una combinación de varios.
21. El sistema permite ver las canciones más populares de un país.
22. El sistema permite ver las canciones más populares del país de un usuario.
23. El sistema permite buscar artistas mediante palabras clave (nick, nombre y biografía), país o una combinación de varios.
24. El sistema permite buscar listas de reproducción mediante palabras clave (título), canciones de la lista, tipo de orden de la lista o una combinación de varios.
25. Las búsquedas por palabras clave deberán contener al menos una palabra y, si contiene sólo una palabra, que sea de longitud mayor o igual a 3 caracteres.

#### **Requisitos no funcionales**

1. El nick de un usuario estará compuesto por entre 4 y 32 caracteres.

#### **4.1.3 Android**

##### **Requisitos funcionales**

1. El sistema permite un solo tipo de usuario, y es el usuario registrado (usuario que esté registrado en el sistema).
2. El sistema permite que los usuarios se identifiquen tanto con usuario y contraseña así como con una cuenta de Google.

3. El sistema permite que los usuarios suban canciones a la aplicación
4. El sistema permite que los usuarios se registren en la aplicación.
5. El sistema permite a los usuarios tener una lista de canciones favoritas, la cual será privada.
6. El sistema permite añadir a favorito una canción que se está escuchando.
7. El sistema permite desde cualquier pantalla escuchar canciones y ver los controles básicos de canciones (play, pause y pasar canción)
8. El sistema permite desde una pantalla específica acceder a más controles sobre las canciones.
  - Desde la pantalla específica el sistema permite avanzar o retroceder en la canción a un momento exacto de esta.
  - Desde la pantalla específica el sistema permite ver la letra de la canción dinámica (va al ritmo de la canción) en el caso que la posea.
  - Desde la pantalla específica el sistema permite ver la onda de sonido de la canción.
  - Desde la pantalla específica el sistema permite cambiar el orden de muestra de las canciones (aleatorio o lineal).
9. El sistema posee una pantalla principal en la que se mostrarán recomendaciones (canciones, listas de reproducción y usuarios), novedades sobre los usuarios y listas de reproducción seguidos y canciones más populares.
10. El sistema posee una pantalla de usuario (otros usuarios) en la que se puede ver su información, canciones, listas de reproducción creadas, su número de seguidores, la opción de seguirlo así como un enlace a sus redes sociales.
11. El sistema posee una pantalla de usuario (usuario propio) en la que se puede ver las canciones del propio usuario y sus listas de reproducción creadas, así como añadir más canciones, listas de reproducción o eliminar alguna de las dos.
12. El sistema permite crear listas de reproducción formadas por canciones de las que puedes ser autor o no, las cuales serán públicas.
13. El sistema permite modificar los datos de usuario una vez creado.
14. El sistema posee una pantalla específica en la que se puede ver las canciones que posee una lista de reproducción así como seguirla.
15. El sistema permite buscar las canciones más populares de un género.
16. El sistema permite ver las canciones más populares en el país desde el que se conecta el dispositivo.
17. El sistema permite buscar canciones, artistas y listas de reproducción por su nombre mediante un buscador.

### **Requisitos funcionales**

1. El sistema posee una pantalla principal en la que se mostrarán recomendaciones (canciones, listas de reproducción y usuarios), novedades sobre los usuarios y listas de reproducción seguidos y canciones más populares.

2. El sistema posee una pantalla de usuario (otros usuarios) en la que se puede ver su información, canciones, listas de reproducción creadas, su número de seguidores, la opción de seguirlo así como un enlace a sus redes sociales.
3. El sistema posee una pantalla de usuario (usuario propio) en la que se puede ver las canciones del propio usuario y sus listas de reproducción creadas, así como añadir más canciones, listas de reproducción o eliminar alguna de las dos.

#### **4.1.4 Web**

##### **Requisitos funcionales**

1. El sistema permite registrarse mediante usuario o a través de Google.
2. Verificación de cuenta. Esto supone un indicador gráfico en el perfil de usuario que anteriormente a verificado su identidad. La finalidad es garantizar la verdadera identidad del usuario.
3. La reproducción actual de un usuario se sincroniza en todos los dispositivos.
4. Existen 3 perfiles de usuarios: registrados, no registrados y administrador.

##### **(a) Registrados**

- i. Los usuarios no registrados pueden acceder a la página principal, páginas de los diferentes usuarios, páginas de canciones además de poder realizar búsquedas.
- ii. Los usuarios no registrados pueden reproducir canciones de usuarios o de listas de reproducción públicas.

##### **(b) Registrados**

- i. Los usuarios registrados pueden realizar todas las funcionalidades de un usuario no registrado.
- ii. El sistema permite a los usuarios registrados identificarse.
- iii. El sistema permite a los usuarios registrados modificar su información personal.
- iv. El sistema permite a los usuarios registrados subir canciones.
- v. El sistema permite administrar una lista de reproducción privada de favoritos.
- vi. El sistema permite añadir cualquier canción a favoritos.
- vii. El sistema permite crear listas de reproducción públicas
- viii. El sistema permite añadir cualquier canción a una lista de reproducción previamente creada por el usuario.
- ix. El sistema permite eliminar una canción de sus listas de reproducción o de la lista favoritos.
- x. El sistema permite seguir/ dejar de seguir a un usuario.
- xi. El sistema permite seguir/ dejar de seguir a una lista de reproducción.
- xii. El sistema permite modificar sus enlaces a redes sociales.
- xiii. (RNF) La pantalla principal de un usuario registrado contiene recomendaciones de artistas, listas o usuarios en base a sus gusto y novedades de sus artistas/listas seguidos.

##### **(c) Administrador**

- i. El usuario administrador se identifica en la aplicación con un usuarios y contrasea predefinidos.
- ii. El sistema permite al usuario administrador puede verificar la identidad de un usuario.
- iii. El sistema permite al usuario administrador modificar la información de un usuario, canción o playlist.
- iv. (RNF) La pantalla de administrador incluye un buscador para la búsqueda de un usuario, lista o canción y una lista con el resultado de la búsqueda.
- v. (RNF) Cada canción, lista o usuario tiene una pantalla de administrador desde donde el sistema permite al administrador la modificación.

### **Requisitos no funcionales**

1. El sistema tiene una pantalla principal donde se mostraran novedades de artistas y canciones y canciones mas populares.
2. El sistema tiene una pantalla de usuario pública para todo el mundo donde aparece su información, canciones, listas de reproducción públicas, número de seguidores y enlaces a sus redes sociales
3. El sistema cuenta en todas las pantallas con un reproductor de audio que incluye información sobre la canción que se esta reproduciendo y las opciones de avanzar, retroceder a un momento de la canción; avanzar o retroceder a la siguiente o anterior canción si se esta reproduciendo una lista; parar la reproducción y cambiar el orden de reproducción a aleatorio.
4. El sistema tiene en todas las pantallas un buscador para realizar una búsqueda de canciones, listas y usuarios.
5. El sistema tiene una pantalla para cada canción donde se puede ver la información de la canción, la onda de sonido y si tuviese las letras de
6. forma dinámica si se estuviese reproduciendo.
7. El sistema tiene una pantalla para identificarse/ registrarse.

## **4.2 Diseño del sistema**

- Diagramas arquitecturales (de módulos, de componentes y conectores, de distribución), patrones de diseño y estilos arquitecturales que se aplicarán. Las interfaces (de módulos y de componentes) son especialmente importantes. También lo son los protocolos de comunicación entre componentes.
- Tecnologías elegidas (lenguajes de programación, componentes que se integrarán, API web externas con las que se conectará etc.).
- Otros aspectos técnicos de interés (p.ej. si hay base de datos si va a ser SQL o NoSQL, si hay una API Web va a ser RESTful o no, si algunas de las operaciones van a ser asíncronas o no, si va a ser una aplicación móvil o de escritorio será nativa o se van a usar tecnologías web, cómo se van a considerar los requisitos de seguridad o de prestaciones, cómo y dónde se harán las instalaciones y despliegues etc.)

Hay que justificar todas las decisiones de diseño. Esto exige contestar a dos preguntas sobre cada decisión: qué alternativas se barajaron? y por qué se eligió una y no las otras?