



ENTREGA FINAL TP1

KMusic

Braulio Alpizar
Leslie Becerra
José David Chaverri

Contenido

Resumen ejecutivo	3
Descripción del problema.....	4
Requerimientos de la herramienta	5
Funcionales	5
No Funcionales	6
Descripción de la arquitectura.....	7
Descripción detallada.....	8
MVC	8
Diagrama de clases.....	9
Diagrama de componentes.....	10
Diagrama de despliegue.....	11
Interacción con sistemas externos	12
Wireframes	13
Manual de usuario	17

Resumen ejecutivo

La aplicación móvil se desarrollará en el sistema operativo Android, su función principal es la de poder realizar recomendaciones acerca de posibles artistas y géneros musicales, además se cuenta con la opción de poder realizar búsquedas personalizadas a las cuales se les podrá aplicar una serie de filtros como por ejemplo: Artista, Álbum, Canción y Género. Dependiendo de la búsqueda así se mostrarán los resultados, pero siempre se buscan los primeros 3 parámetros mencionados, por ejemplo en la búsqueda de un Artista se mostrará la información del Artista, los álbumes que ha compuesto, las canciones de dichos álbumes. También se contará con una pantalla en donde se tendrán enlaces a YouTube de los conciertos y covers de la búsqueda.

La arquitectura que se implementará será la de Rich Client, la cual corresponde a una arquitectura en donde la mayor carga de procesamiento se encuentra en el dispositivo que almacena la aplicación en cuestión, el modelo usualmente se basa en un cliente-servidor en donde la capa de presentación y lógica se encuentran en el cliente y la capa de datos en el servidor, sin embargo por la naturaleza de la aplicación la parte del servidor no es necesaria, en su lugar se cuenta con una serie de API a los cuales la aplicación consulta y muestra la información.

La interacción con sistemas externos se da con tres tipos de API's, el primer tipo corresponde al caso de obtener información acerca de artistas, géneros, álbumes y canciones se usarán 3 API's de distintos proveedores para obtener dicha información: Last.fm Web API, Spotify Web API y MusiXmatch API. El segundo tipo corresponde al servicio de compra de álbumes, el API que se usará es: Amazon Web API. El tercer tipo corresponde para la publicación de información en redes sociales los API's que se usarán respectivamente son: Facebook Web API y Twitter Web API.

Descripción del problema

En el mundo actual en el que vivimos la industria de la música se ha desarrollado de manera acelerada y en constante cambio, debido a esto cada vez más se producen gran cantidad de material musical todos los días y en diversos lugares del planeta, un ejemplo de esto es el nacimiento de nuevos artistas, géneros y álbumes cada vez más rápido y en mayor cantidad todos los días. La creación de esa gran cantidad de material musical genera problemas pues a las personas que les gusta la música se les dificulta el poder obtener información de sus artistas favoritos o bien el conocer acerca de nuevos artistas o agrupaciones, géneros y álbumes, además de esto el cómo poder comprar sus discos y el poder compartir con sus amigos y familiares sus gustos y tendencias en el ámbito musical son otros retos al que se enfrentan estas personas.

Debido a los problemas mencionados anteriormente es que nace la idea de desarrollar KMusic, esta aplicación pretende facilitar la búsqueda de información a las personas que disfrutan de la música, además de ofrecerle la opción de comprar álbumes y poder compartir sus gustos y tendencias en las redes sociales.

Requerimientos de la herramienta

Funcionales

- La aplicación deberá mostrar al usuario una lista canciones recomendadas según las siguientes páginas web: <http://www.last.fm>
- La aplicación deberá permitir al usuario buscar información por medio de ciertos criterios tales como: artista, género, canción, álbum, geografía y artistas similares, según las siguiente páginas web:
 - <https://www.spotify.com>
 - <http://musixmatch.com>
 - <http://echonest.com>
 - <http://www.last.fm>
- La aplicación brindará al usuario la opción de publicar en facebook.com y twitter.com cierta información obtenida en la búsqueda personalizada o selección de las recomendaciones.
- La aplicación permitirá al usuario ver por medio de un youtube.com el video de alguna canción.
- La aplicación brindará al usuario acceso a la página de Amazon en donde podrá realizar la compra del álbum mostrado por la aplicación.

No Funcionales

Confiabilidad

- La aplicación debe detectar si se cuenta con conexión a internet.

Rendimiento

- La aplicación deberá mostrar los resultados de las búsquedas del usuario en menos de 10 segundos. *
- La aplicación deberá brindar acceso a la página web de Amazon en menos de 10 segundos.*
- La aplicación deberá brindar acceso a la página de YouTube en donde pueda visualizar los videos de las canciones resultantes de la búsqueda en menos de 10 segundos.*

Mantenimiento

- La aplicación deberá tener documentación interna que describa las funcionalidades de cada clase, así como los métodos usados, esto para facilitar el mantenimiento a la aplicación.
- La aplicación deberá tener documentación externa que describa las decisiones de diseño tomadas con respecto a manejo de interfaz, lógica y datos, esto para facilitar el mantenimiento a la aplicación.

Usabilidad

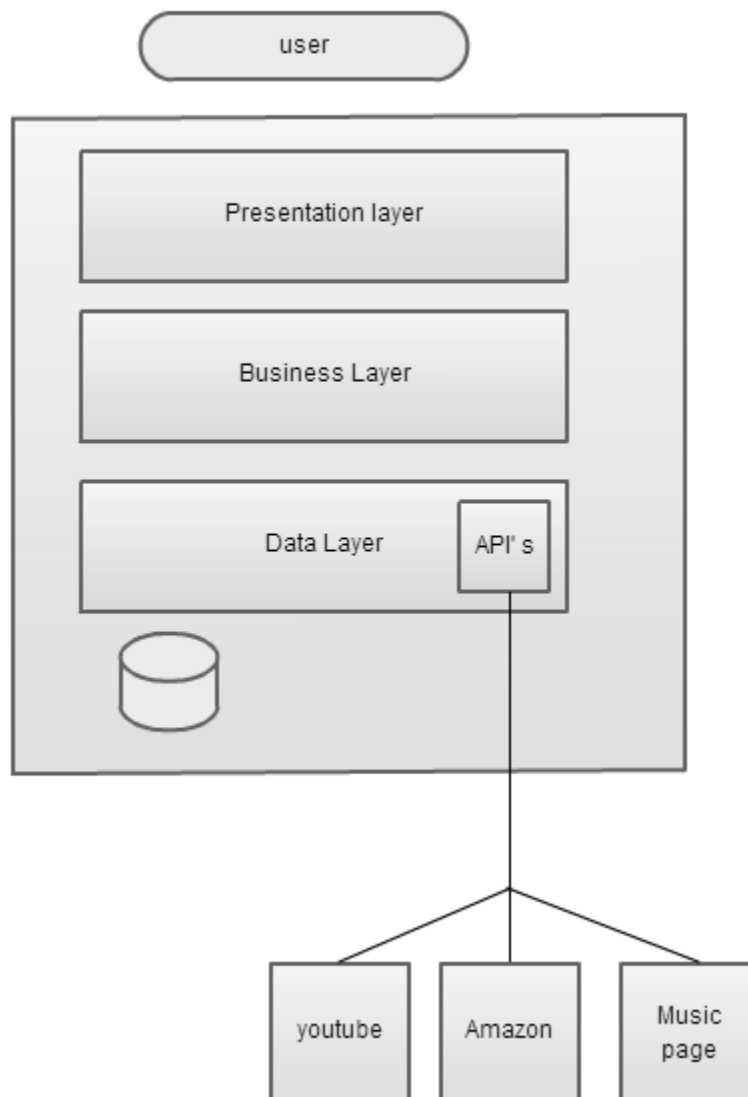
- Debido a que la aplicación es fuertemente dependiente de conexión a internet, el sistema debe alertar al usuario sino se cuenta con conexión.
- El sistema debe mostrar al usuario un icono de lupa en las opciones que generan acciones de búsqueda.
- El sistema debe mostrar al usuario un icono de compartir contenido en las opciones que generan acciones para compartir en redes sociales.
- En caso de error el sistema debe indicar al usuario que ha ocurrido determinado error.

*. Nota: La cantidad de 10 segundos, se basa en pruebas realizadas a ciertas aplicaciones como: Shazam entre otras, las cuales hacen llamadas a otras aplicaciones como: Facebook, Twitter, YouTube y al browser predeterminadas, así como opciones de búsqueda similares a la aplicación que se desea desarrollar. Estas pruebas dieron como tiempo promedio de llamada 10 segundos en escenarios con distintas conexiones a internet y bajo distintas cargas de trabajo en el dispositivo.

Descripción de la arquitectura

La arquitectura de la aplicación es Rich Client sin capa propia de servidor, la única parte del servidor es la conexión mediante API' s a algunas páginas de música de terceros.

Consiste en una aplicación que contiene la capa presentación, la capa de modelo de negocio y la capa de datos, en esta última es donde se acceden a los datos mediante los Agent Services (API' s) los cuales reciben ciertas entradas y brindan a la aplicación determinados datos de acuerdo a dichas entradas o realizan alguna acción específica como ejecutar otra aplicación.



Descripción detallada

MVC

El patrón ha de usar en la aplicación en general es MVC (Model View Controller). Con este patrón se busca mantener separadas cada una de las capas y mantener ciertas interfaces entre las mismas las cuales se encargarán de comunicar cada capa y que estas soliciten y brinden servicios entre si siguiendo el flujo que se muestra en la siguiente imagen.

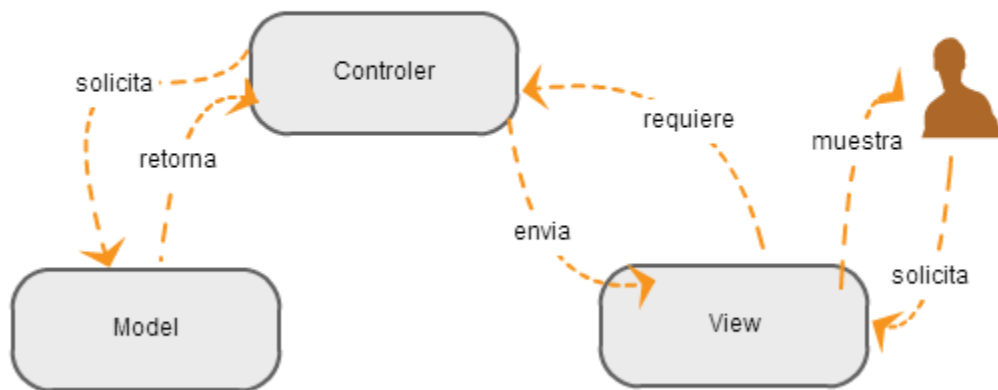
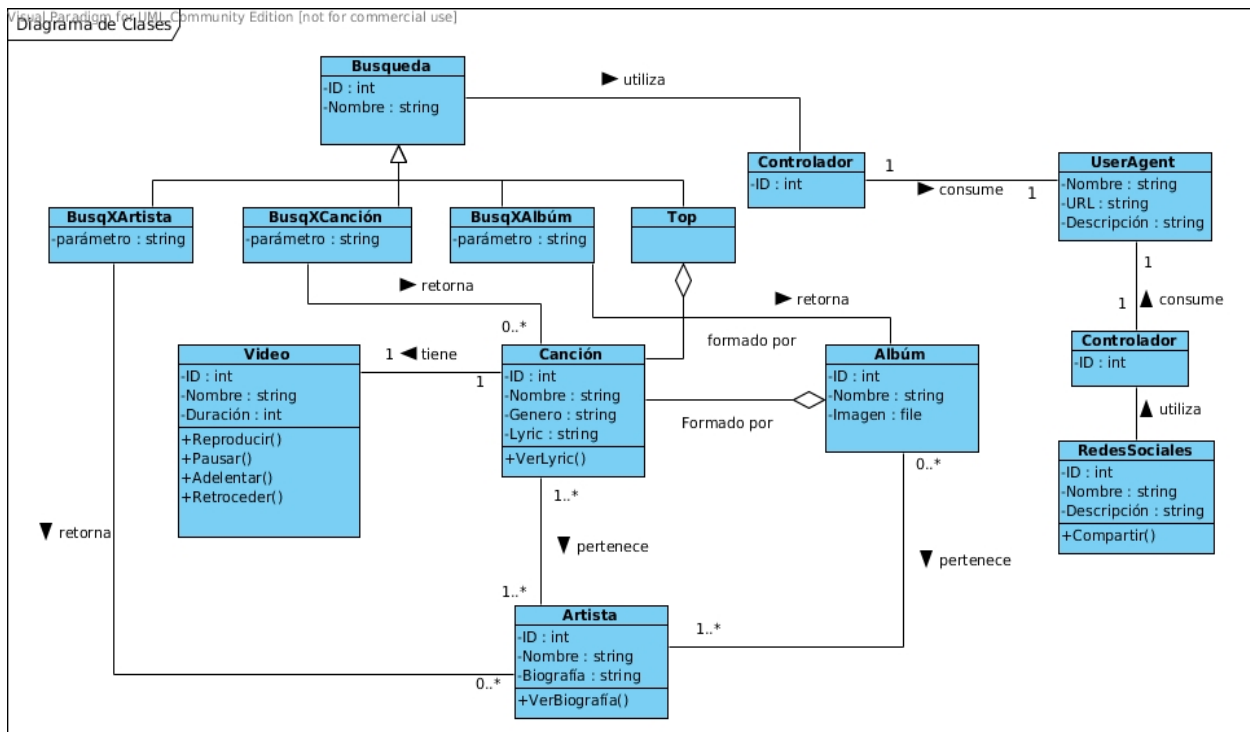


Diagrama de clases



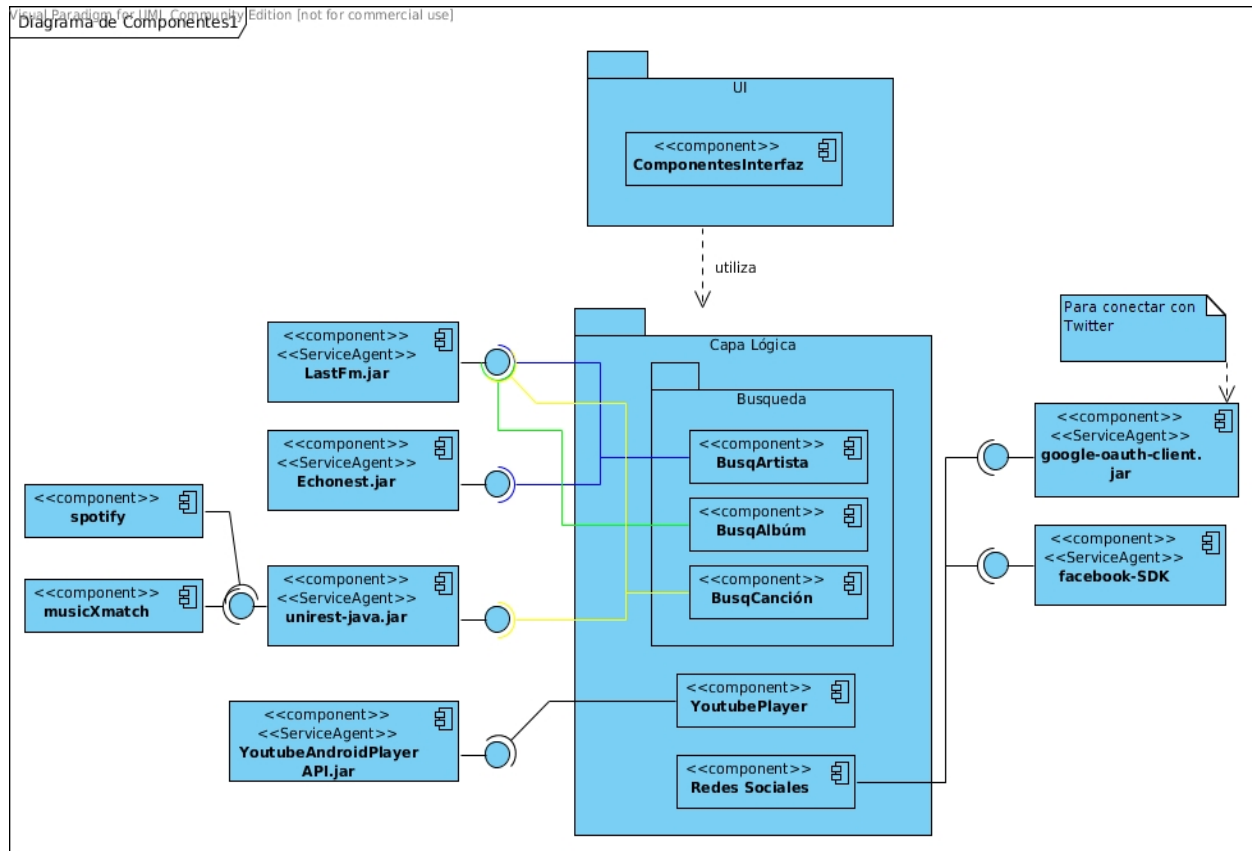
Como se observa en el diagrama de clases anterior, las clases que se tienen pertenecen al modelo de dominio de aplicaciones relacionadas con música, por ejemplo la clase canción tiene atributos como: ID, Nombre, Género y Lyric, además del método `verLyric()` el cual se encuentra de mostrar la letra de la canción.

La clase álbum tiene una agregación con respecto a la clase Canción, esto quiere decir que un álbum está formado por canciones, lo mismo sucede con la clase Top que también está conformado por objetos de tipo Canción.

La clase **UserAgent** representa a los API's usados por la aplicación, cada objeto de esa clase será reflejado en el consumo de la aplicación al API respectivo según lo que se desea hacer, esta clase es usada por dos clases distintas, la clase búsqueda y la clases **RedesSociales**, cada una con su controlador respectivo, esta controlador contendrá ciertos métodos para poder enviar la solicitud al **UserAgent** y procesar la información recibida del mismo.

La clase **Búsqueda** se especializa en 4 tipos de búsqueda, tres de las cuales buscan mediante un parámetro específico, siendo la clase **Top** la única que busca sin recibir un parámetro.

Diagrama de componentes



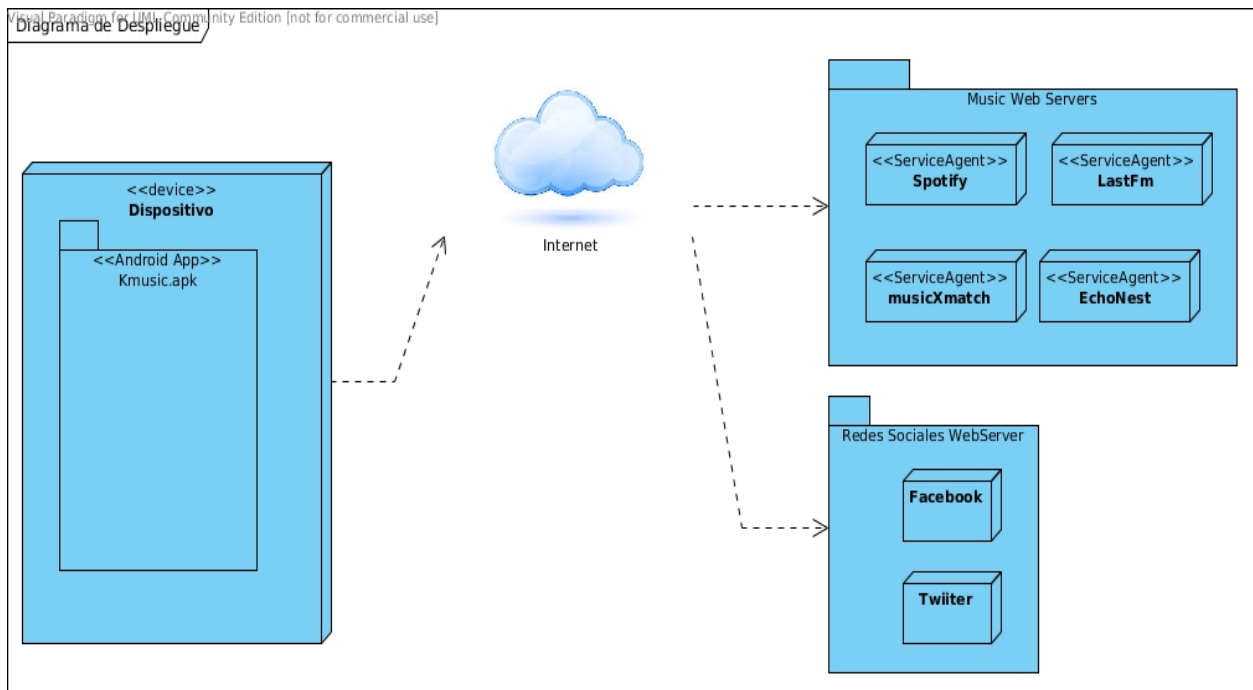
Como se observa en el diagrama de componentes anterior, se tiene una capa de lógica y una de Interfaz, no se cuenta con capa de datos, debido a que los datos que se manejan son pocos y la aplicación no requiere del uso de una Base de datos.

En la capa de IU (User Interface) se tiene los componentes de interfaz proveídos por Android.

En la capa de lógica, se tiene los componentes de búsqueda, manejo de redes sociales y el reproductor de YouTube.

Cada búsqueda hace uso de componentes UserAgent *.jar los cuales tienen funcionalidades específicas.

Diagrama de despliegue



Como se observa en el diagrama de despliegue la aplicación corre en un dispositivo con una versión de Android 4.1.1 o superior, la aplicación consume los servicios de los API's mediante Internet, la aplicación no cuenta con un BackEnd propio.

Interacción con sistemas externos

Para el funcionamiento de la aplicación se necesitará la interacción con otros sistemas, que este caso será API's necesarias tanto para obtener los datos necesarios como para proporcionar servicio de compra y publicación de información.

En el caso de obtener información acerca de artistas, géneros, álbumes y canciones se usarán 3 API's de distintos proveedores para obtener dicha información, estos son los API's que serán utilizados:

- Last.fm Web API
- Spotify Web API
- MusiXmatch API

Con respecto al servicio de compra de álbumes el API que se usará es:

- Página Web de Amazon mediante el browser predeterminado del dispositivo.

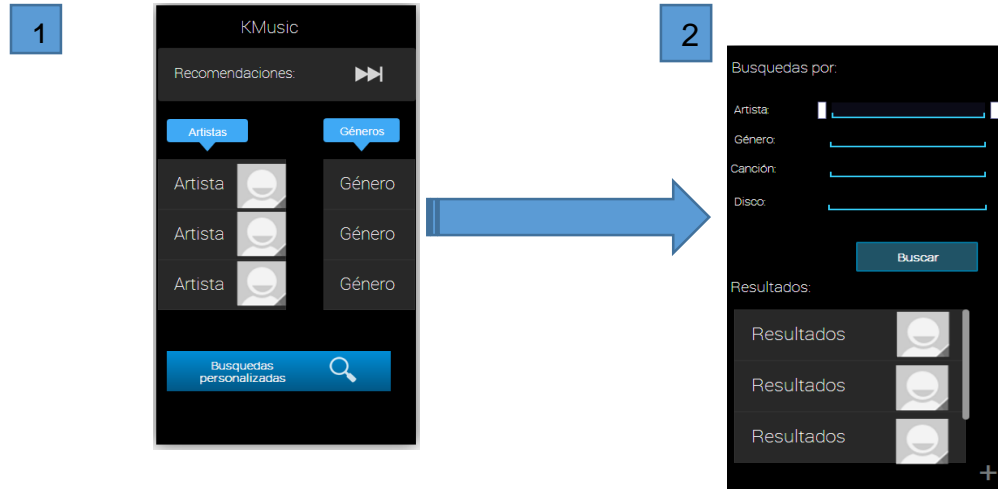
Para la publicación de información en redes sociales los API's que se usarán respectivamente son:

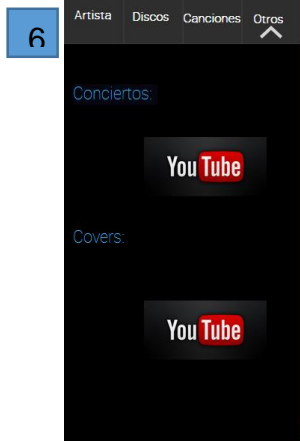
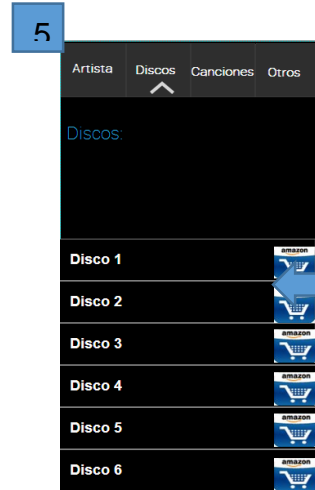
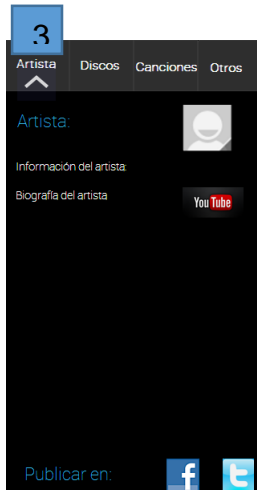
- Facebook Web API
- Twitter Web API

Estos son los API's que serán utilizados por la aplicación para realizar las funciones mencionadas anteriormente.

Wireframes

A continuación se muestran los Wireframes propuestos para la aplicación.





Flujo primario de la aplicación: La navegación inicia por la pantalla mostrada en la imagen número 1, en donde por medio del botón de búsquedas personalizada, se hace el cambio a la pantalla mostrada en la imagen número 2, en donde a través de los resultados mostrados por la búsqueda, se salta hacia la pantalla número 3, una vez que se encuentra en esta esta pantalla se realiza la navegación por los tabs mostrados en las imágenes 3, 4,5 y 6. De igual manera para devolverse a las imágenes 1 y 2, se hace uso del botón de atrás físico del dispositivo móvil.

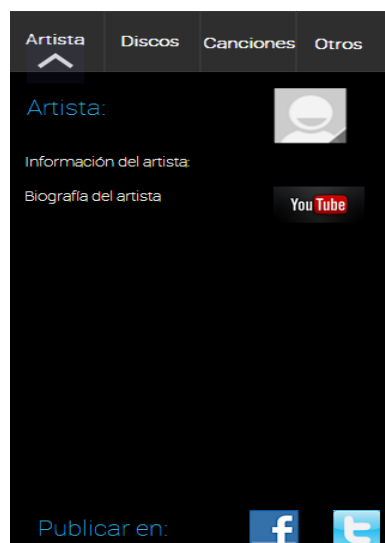
Flujo secundario de la aplicación: La navegación inicia en la imagen 1, en donde a través de resultados mostrados por las recomendaciones se hace el cambio a la imagen número3, una vez que se encuentra en esta esta pantalla se realiza la navegación por los tabs mostrados en las imágenes 3,4,5 y 6. De igual manera para devolverse a las imágenes 1 y 2, se hace uso del botón de atrás físico del dispositivo móvil.



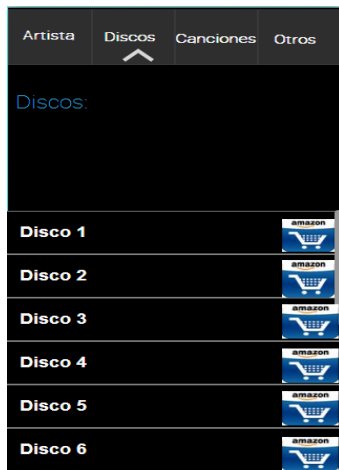
Esta imagen corresponde a una posible representación de la pantalla de inicio de la aplicación móvil kmusic, en donde se muestra una tabla de artista, la cual contendrá el Top 10 de artistas musicales según el API determinado. A su vez también se muestran los géneros más escuchados según el API. También se cuenta con un botón de búsquedas personalizadas el cual redirecciona a una segunda pantalla en donde se especifican los criterios de búsqueda.

artista, género, canción y disco, los cuales funcionan haciendo uso de los API determinados,

luego en la parte inferior de la imagen se muestran todas las posibles coincidencias con los parámetros de la búsqueda realizada. Para poder ver detenidamente toda la información del resultado



Esta imagen corresponde con la pantalla de artista, en donde una vez que se dio "clic" en algún resultado, se muestra la información del artista que corresponde a la búsqueda realizada, también se cuenta con botones de enlace a YouTube, Facebook y Twitter con los cuales se pueden buscar videos del artista en la página de Youtube y publicar en las redes sociales el artista buscado. Dicha pantalla cuenta con barra con tabs la cual sirve para navegar entre las distintas pantallas, de la forma en que se muestra en el diagrama de navegación.

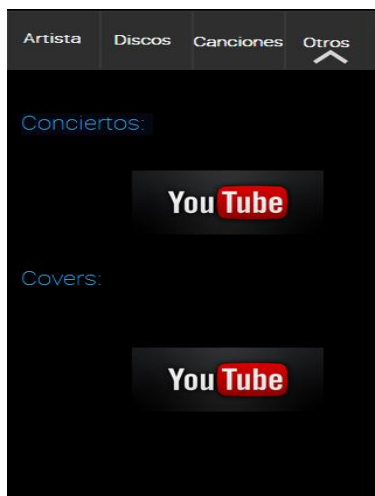


En la siguiente imagen corresponde a la pantalla de discos en donde se muestran los discos correspondientes a la búsqueda realizada, también se da la opción de que por medio de botón que se encuentra en cada resultado, se pueda comprar el álbum en la página de Amazon. Dicha pantalla cuenta con barra con Tabs la cual sirve para navegar entre las distintas pantallas, de la forma en que se muestra en el diagrama de navegación

En la siguiente imagen se muestran la pantalla de canciones la cual despliega las canciones resultantes de la búsqueda, también se tienen botones de enlace a YouTube, Facebook y Twitter con los cuales se pueden buscar videos de la canción en la página de Youtube y publicar en las redes sociales la canción buscada. Dicha pantalla cuenta con barra con tabs la cual sirve para navegar entre las distintas pantallas, de la forma en que se muestra en el diagrama de navegación



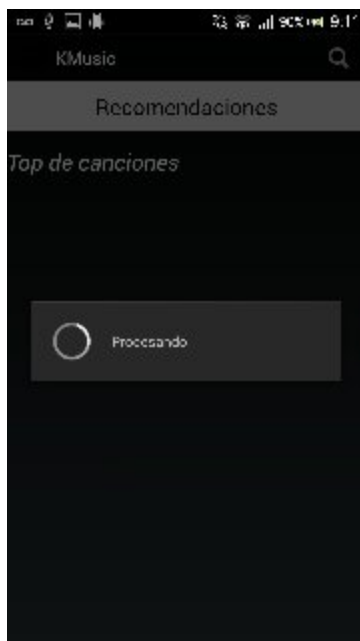
En la siguiente pantalla se tienen enlaces YouTube con los posibles conciertos y covers de la búsqueda realizada. Dicha pantalla cuenta con barra con tabs la cual sirve para navegar entre las distintas pantallas, de la forma en que se muestra en el diagrama de navegación



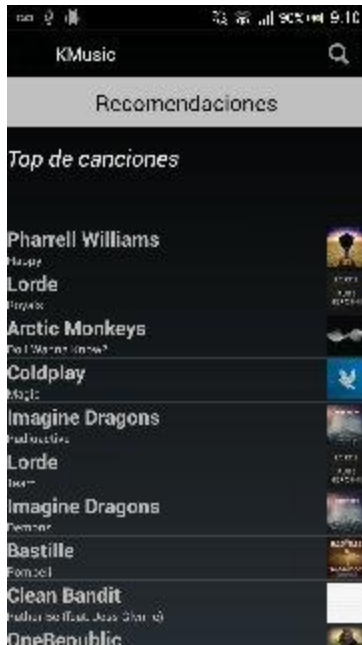
Manual de usuario



Esta es la pantalla de inicio de la aplicación, en esta parte se debe esperar que la aplicación ejecuta ciertas tareas en segundo plano.



En esta pantalla la aplicación está a punto de mostrar cierta información así que se debe esperar.



Esta es la lista de recomendaciones, como se puede observar es una lista con los siguientes elementos:

- Nombre del artista
- Nombre de la canción
- Imagen representativa

En esta parte se puede mantener presionado de dos a tres segundos y aparecerá el menú que se explica a continuación.



Una vez que se deja presionado aparece el menú que se muestra a la izquierda, el menú tiene las siguientes funciones:

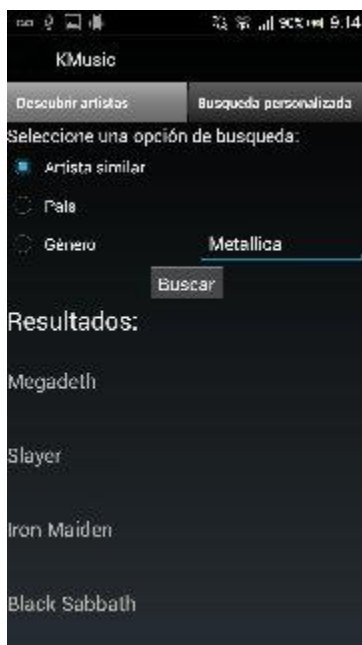
- Ver video: Abre la pantalla que se explica a continuación.
- Compartir en Facebook: Inicia un ventana de la aplicación de Facebook y se ejecuta el procedimiento normal, una vez terminado se da presiona el botón “Compartir” o “Share” presente en la aplicación de Facebook.



Esta pantalla surge una vez que se da clic en ver video. En la parte superior se muestra el video y en la parte inferior se muestra la letra de la canción.

En la parte superior derecha está el menú con la opción BUSCAR y la opción compartir la cual corresponde al icono que está a la derecha de BUSCAR, esta opción permite compartir en la red social Twitter.

Si se da Clic en la opción BUCAR la aplicación va a la pantalla que se muestra a continuación.



En esta pantalla se muestran dos pestañas, a la izquierda Descubrir artistas y Búsqueda personalizada.

En la pestaña de Descubrir artistas, con las opciones Artista Similar, País y Generó estas son mutuamente exclusivas por lo que solo se puede seleccionar una de ellas, a la derecha donde dice "Metallica" es un espacio para ingresar el criterio de búsqueda, una vez ingresado se presiona el botón Buscar, con esto se muestra una lista en la parte inferior, estos son los resultados, para acceder a cada uno de ellos se presiona.



En el paso anterior se presionó un elemento de la lista, ante esto surge esta pantalla, como se puede observar esta muestra tres pestañas, de izquierda a derecha son: Biografía, Álbums y Concierto.

En la pestaña de Biografía que es la que se muestra a la derecha, esta presenta la biografía del artista.



En esta pantalla de Álbums se muestra un listado de los álbums que pertenecen al artista.

Basta con presionar en algún elemento de la lista para que se muestre la siguiente pantalla.



En esta pantalla se muestra la información del álbum, en la parte superior se muestra el nombre del álbum, a la derecha la imagen del portada del disco, justo debajo de la imagen se encuentra el botón Comprar el cual abre la página de Amazon en el Browser predeterminado del dispositivo.

En la parte inferior se muestran las canciones del disco, si se presiona o da clic en alguna de estas se muestra el video de YouTube respectivo.



Volviendo a las pestañas anteriores, se encuentra la pestaña Concierto, la cual muestra el video en YouTube de algún concierto en vivo del artista.



Volviendo a la pantalla de búsquedas, la búsqueda Personalizada cuenta en la parte superior con entradas de texto para ingresar en orden de arriba abajo, Artista, Álbum y Canción, luego se muestra el botón Buscar, una vez presionado este botón en la parte inferior se muestran los resultados. Basta con presionar alguno de estos elementos y se mostrara la pantalla con las pestañas de Biografía, Álbumes y Concierto.