

La Gramola

Proyecto de aplicación para Curso Android 2015-2016

Estévez Casado, Fernando
`fernando0estevez@gmail.com`

Laso Rodríguez, Rubén
`ruben.laso.riguez@gmail.com`

Universidade de Santiago de Compostela, Galicia

Domingo, 14 de febrero de 2016

Resumen

La Gramola es un reproductor de archivos de audio que te permitirá aprender las letras de las canciones que tengas en tu dispositivo, mostrándotelas cómodamente en la pantalla al mismo tiempo que se reproducen.

Índice

1. Introducción	2
2. Funcionamiento básico	2
3. Arquitectura bsica de la aplicación	5

1. Introducción

La Gramola es un reproductor de archivos de audio. Este muestra los archivos .mp3 que tenemos en nuestro dispositivo en una lista ordenada por autor, como se puede ver en la Figura 1. En dicha lista se muestra el título de cada canción, así como su autor y el álbum al que pertenece.



Figura 1: Captura de pantalla de la `MainActivity`, mostrando la lista de archivos de audio.

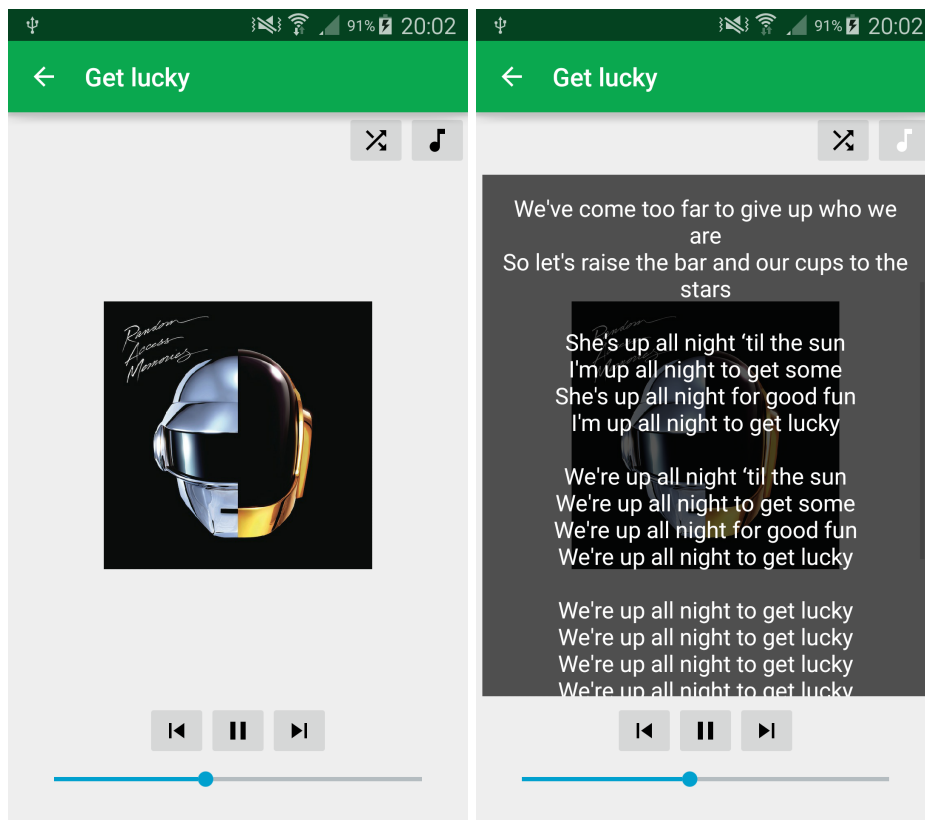
2. Funcionamiento básico

Pulsando sobre cualquier entrada se nos abrirá el reproductor y automáticamente comenzará a sonar la canción escogida, como se puede ver en la Figura 2a. Podremos pausar, reanudar, avanzar y retroceder en el audio con los botones universales que se proporcionan. A mayores, podremos avanzar a la siguiente canción o retroceder a la anterior.

El primer botón de la parte superior activará el modo aleatorio, con el cual el orden de las canciones será totalmente aleatorio.

La nota musical (el segundo de los iconos superiores) mostrará la letra de la canción en caso de que la app la pueda encontrar (ver Figura 2b). En otro caso mostrará un mensaje de error.

El mecanismo para encontrar las letras es simple. Se toman los datos de la pista (título y autor) y se realiza una consulta a la API de Musixmatch. Primero se busca la pista en su base de datos con la información de la que disponemos. Si conseguimos obtener una coincidencia, usamos el ID que se nos devuelve para obtener los *lyrics*.



(a) Vista vertical del reproductor.

(b) Vista vertical con la letra de la canción.

Figura 2: Vistas verticales del reproductor

El reproductor dispone de una vista vertical (Figuras 2a y 2b) y otra apaisada (Figuras 3 y 4). Cada una de ellas optimiza la disposición de la información mostrada por pantalla.

Desde la pantalla principal con la lista de canciones podemos acceder al menú de ajustes pulsando sobre el engranaje. Este menú nos permite configurar la disposición automática de los *lyrics*, así como el orden aleatorio de las pistas. La opción de cambiar el tema se muestra bloqueada por el momento, puesto que está en fase de construcción y en futuras versiones se permitirá elegir entre

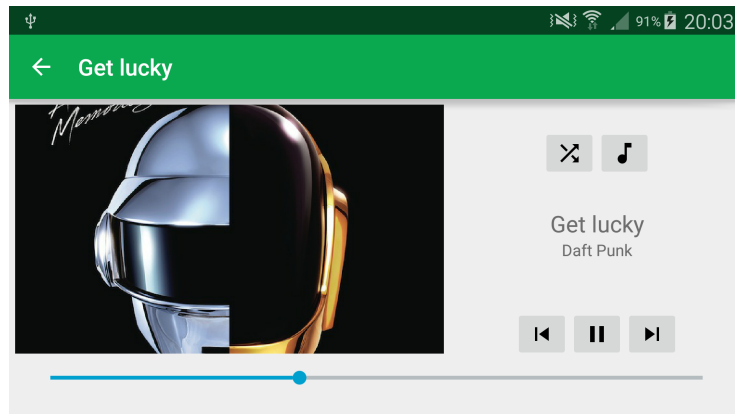


Figura 3: Captura de pantalla de la PlayerActivity (vista apaisada).

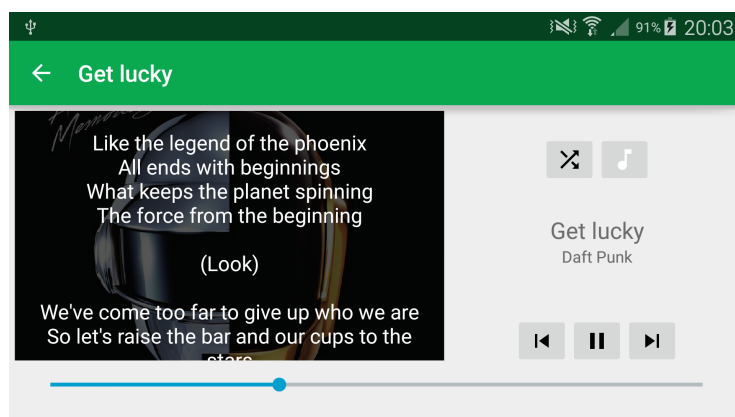


Figura 4: Captura de pantalla de la PlayerActivity (vista apaisada), mostrando la letra de la canción.

los temas Azul, Rojo y Verde. Por ahora se usa el Verde de forma estática.

3. Arquitectura básica de la aplicación

A continuación se detalla brevemente la función de cada una de las clases incluídas en el proyecto:

- **GetLyrics**: es una **AsyncTask** que se conecta a la API y devuelve la letra de las canciones. Solicita como parámetros dos **String** (el título y el autor). La base de datos de las canciones es bastante completa, no obstante, al ser gratuita permite consultar hasta 500 *lyrics* al día y, por temas legales, algunos de ellos se muestran parcialmente.
- **GetPreferences**: se encarga de administrar las preferencias en tiempo de ejecución. Es decir, recoge y aplica los cambios introducidos en la pantalla de reproducción.
- **MainActivity**: actividad encargada de mostrar la lista de archivos de audio del dispositivo.
- **PlayerActivity**: clase controladora del reproductor de canciones.
- **SettingsActivity**: clase para la actividad del panel de ajustes. Extiende **PreferenceActivity**.
- **SongViewHolder**: clase encargada de adaptar la información a las **views** de la lista de archivos.
- **SongListAdapter**: esta clase cogerá los datos del **Cursor** y, con la colaboración del **ViewHolder**, creará la lista de archivos de la **MainActivity**.