UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Título del Proyecto:

SoundFace

Autor:

Garzón Montesinos, Francisco José

Director:

Carbonell, Vicente

TESINA PARA LA OBTENCIÓN DEL TÍTULO DE:

Diploma de Especialización en Desarrollo de Aplicaciones para Android

Septiembre del 2015

Contenido

Introducción

Descripción del problema

Sounface pretende ser un proyecto en el que se ponga de manifiesto parte de lo aprendido en el diploma de especialización de desarrollo de aplicaciones en Android.

Consiste en una aplicación en la que se pueden generar imágenes con mensaje y/o videos a partir de datos independientes introducidos por el usuario.

El usuario puede generar un vídeo seleccionando el sonido y la imagen, una vez seleccionadas el sistema genera el video automáticamente.También puede generar una imagen a la que se le añadirá el texto que el usuario crea conveniente.

Estas generaciones se harán usando hilos de ejecución.

En cuanto a las imágenes, el usuario podrá elegir entre las que hay inicialmente almacenadas en la aplicación (y que se mostrarán usando fragments) o utilizar la cámara para seleccionar la imagen a través de una foto o usar una imagen de la galería.

En cuanto al sonido, el usuario podrá elegir entre las que hay inicialmente almacenados en la aplicación o utilizar sonidos almacenados en el dispositivo móvil.

El texto añadido a las imágenes será insertado a través de un campos de texto y se podrá seleccionas posición, color, estilo, etc.

Tanto las imágenes finales como el vídeo podrán ser compartidos en redes sociales.

Habrá notificaciones push que abrirán una actividad para bajar imágenes propuestas cada cierto tiempo.

Las imágenes se podrán modificar con diferentes filtros usando código nativo.

Se podrá dibujar usando el dedo sobre las imágenes generadas.

Objetivos

Es prácticamente imposible usar todos los módulos vistos en el diploma en un mismo proyecto, por lo que he optado por realizar un proyecto en el que pueda utilizar lo máximo posible con cierto sentido. Los objetivos consisten en poner de manifiesto lo aprendido, en este caso además de las vistas, layouts, actividades, intenciones, almacenamiento de datos, etc… vistos en la parte de fundamentos, he querido añadir procesado de imagen, hilos de ejecución, notificaciones push, código nativo, redes sociales…entre otras cosas.

En resumen, el objetivo es poner en práctica la mayor parte de los conocimientos adquiridos.

Motivación

La motivación se basa en poder realizar una aplicación Android usando los conocimientos adquiridos y que contenga una cierta utilidad.

Debido a que en la parte de fundamentos realicé un proyecto más orientado a un uso profesional, esta vez he preferido realizar una aplicación más orientada a la parte lúdica.

En este sentido, aunque la aplicación en esta versión inicial no tiene una funcionalidad muy potente, la intención es mejorarla para que pueda tener una funcionalidad más amplia. Sin embargo con el tiempo que he tenido pada poder dedicarle al desarrollo, creo que el objetivo de usar la mayor parte de lo aprendido queda cubierto.

Tecnologías utilizadas

Además de la tecnología descrita durante el Diploma, he usado las siguientes tecnologías:

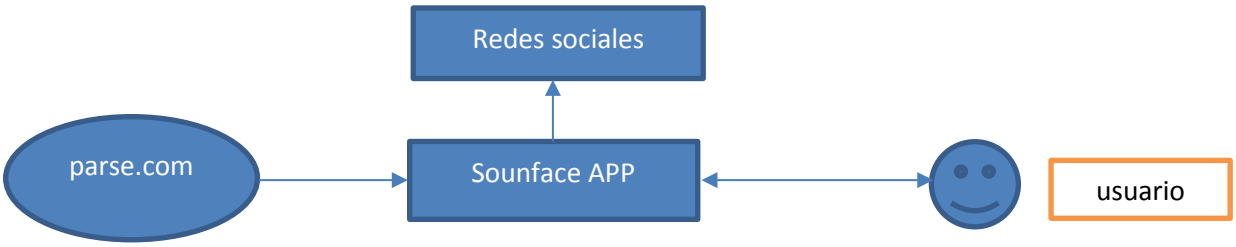
- Unas librerías “javacv”, “javaccp”, “ffmpeg”, “opencv” que me ha permitido la generación de videos a partir de imágenes y sonidos. (algunas de estas librerías incluyen código nativo). Para que estas librerías funcionen correctamente en el dispositivo móvil, este ha de tener instalado “OpenCV Manager”.
- Unas librerías “parse”, “bolts-android” que permite las notificaciones push usando un servidor proporcionado por la web parse.com y que se verá con más detenimiento en un apartado dedicado a este respecto.

Arquitectura de la aplicación

Esquema del diseño

La aplicación Soundface consta de varias secciones.

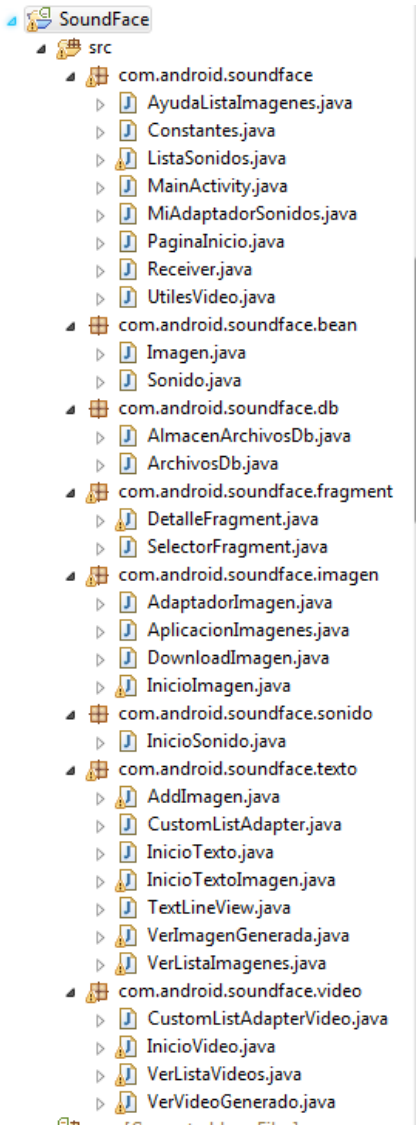
A gran escala podemos describir el siguiente esquema que representa la aplicación frente a los sistemas externos:



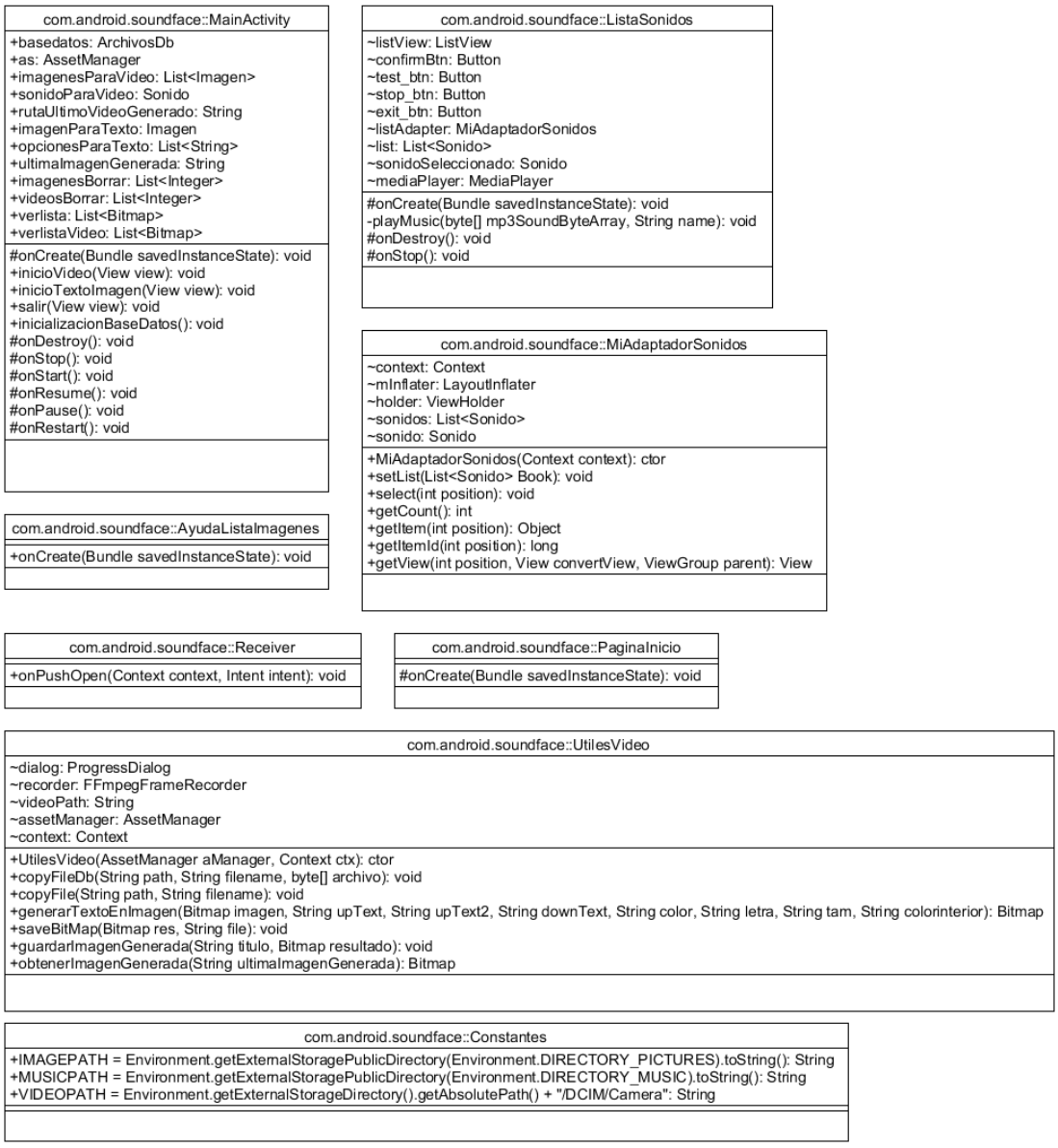
En nuestro caso, la aplicación sólo se comunica de manera externa con el usuario, con el que interactúa, con la web parse.com que nos proporciona las notificaciones push y con las redes sociales, a las que envía los datos generados por la propia aplicación

Si profundizamos más y nos metemos de lleno en el esquema de lo que es la aplicación propiamente dicha, podemos ver que el esquema es el siguiente:

1. Por un lado las clases .java



1.1 Las clases java se han separado en diferentes paquetes. Comenzando por el paquete raíz "com.android.soundface". Este paquete contiene las clases básicas usadas en la aplicación, alguna clase útil para el resto de paquetes y una clase constantes también utilizada por el resto de paquetes. Su diagrama de clases sería:

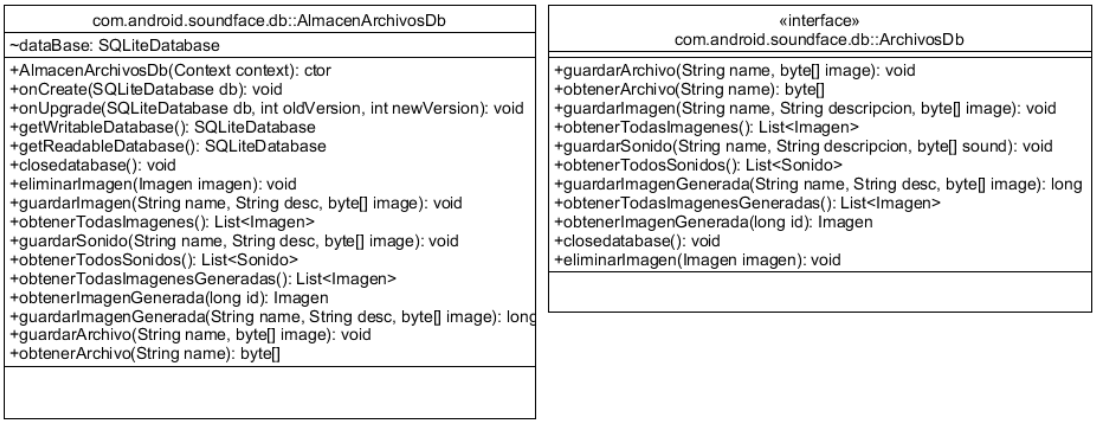


1.2 El siguiente paquete es el paquete bean, este paquete contiene dos objetos básicos en la aplicación "Imagen" y "Sonido" que son las clases básicas usadas en otras clases y que contienen información de las imágenes y sonido creados en la app. Su diagrama de clases sería:

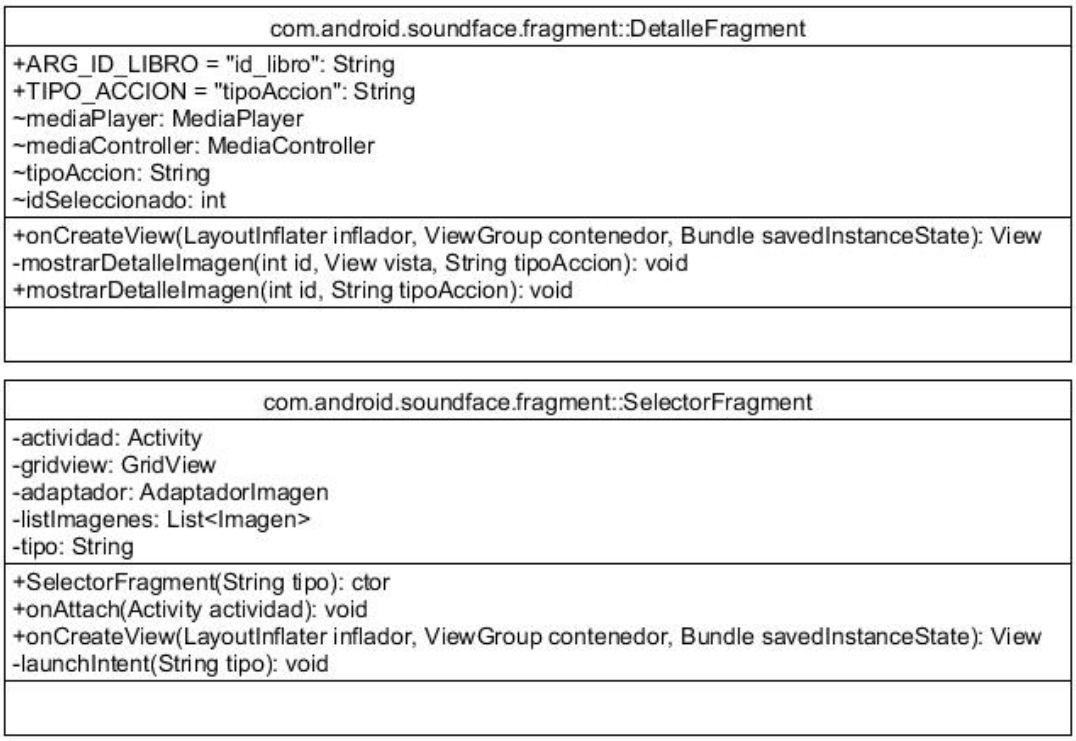
com.android.soundface.bean::Sonido
-serialVersionUID = 1L: long -idSonido: Integer -nombreSonido: String -descSonido: String -sonido: byte[] -isSelected = false: boolean
+Sonido(Integer idSonido, String nombreSonido, String descSonido, byte[] sonido): ctor +getNombreSonido(): String +setNombreSonido(String nombreSonido): void +getIdSonido(): Integer +setIdSonido(Integer idSonido): void +getDescSonido(): String +setDescSonido(String descSonido): void +getSonido(): byte[] +setSonido(byte[] sonido): void +isSelected(): boolean +setSelected(boolean isSelected): void

com.android.soundface.bean::Imagen
-serialVersionUID = 1L: long -idImagen: Integer -nombreImagen: String -descImagen: String -imagen: byte[]
+Imagen(Integer idImagen, String nombreImagen, String descImagen, byte[] imagen): ctor +getNombreImagen(): String +setNombreImagen(String nombreImagen): void +getIdImagen(): Integer +setIdImagen(Integer idImagen): void +getDescImagen(): String +setDescImagen(String descImagen): void +getImagen(): byte[] +setImagen(byte[] imagen): void

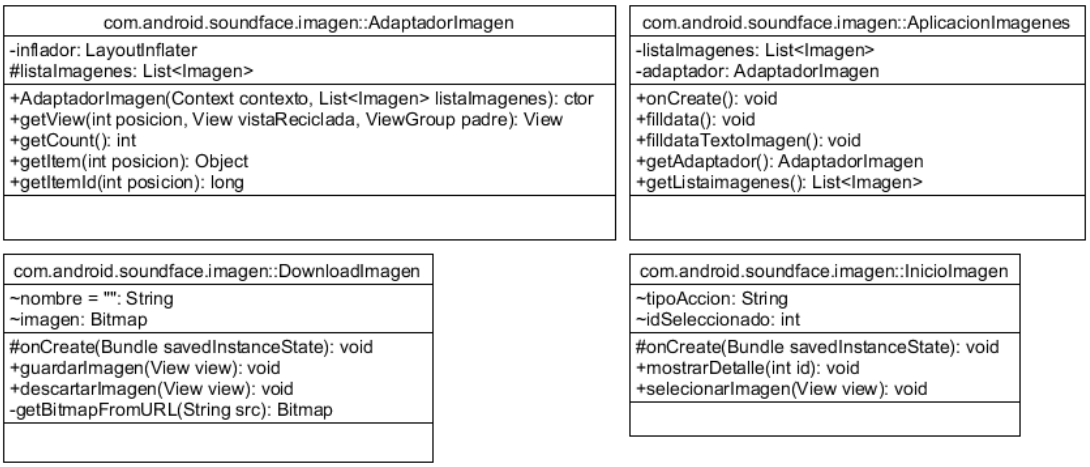
1.3 El paquete db contiene las clases asociadas a la interacción con la base de datos SQLite que contiene la aplicación. El modelo de base de datos será explicado más adelante.Su diagrama de clases sería:



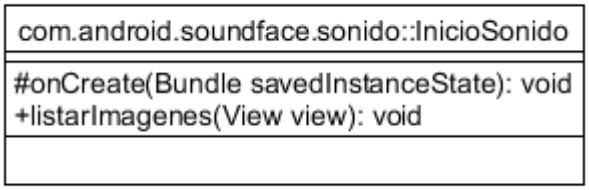
1.4 El paquete fragment contiene las clases asociadas a la creación de fragments en la app. En nuestro caso hemos usado fragments para la visualización de las imágenes precargadas que la app va a usar para la creación de vídeos e imágenes más texto. El diagrama de clases es:



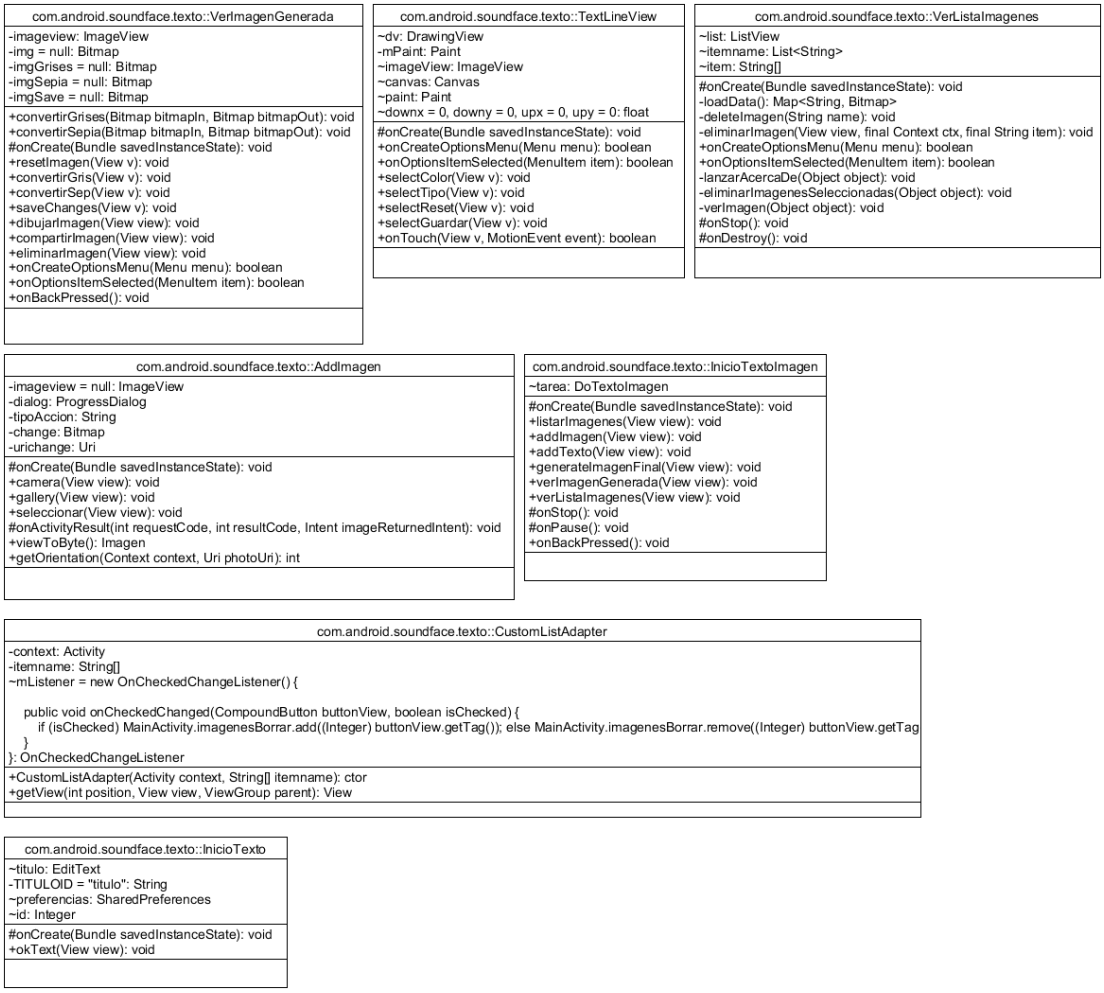
1.5 El paquete imagen contiene las clases asociadas al tratamiento de imágenes, tanto para su creación como su generación y visualización. El diagrama de clases es:



1.6 El paquete sonido contiene las clases asociadas al tratamiento de sonidos. El diagrama de clases es:



1.7 El paquete texto contiene las clases asociadas al tratamiento de textos para la generación de imágenes con texto incrustado. El diagrama de clases es:



1.8 El paquete vídeo contiene las clases asociadas al tratamiento de vídeos, tanto para su creación como su generación y visualización. El diagrama de clases es:

com.android.soundface.video::VerListaVideos
~list: ListView ~itemname: List<String> ~item: String[]
#onCreate(Bundle savedInstanceState): void -loadData(): Map<String, Bitmap> -deleteVideo(String name): void -eliminarVideo(View view, final Context ctx, final String item): void +onCreateOptionsMenu(Menu menu): boolean +onOptionsItemSelected(MenuItem item): boolean -lanzarAcercaDe(Object object): void -eliminarVideosSeleccionados(Object object): void -verVideo(Object object): void #onStop(): void #onDestroy(): void

com.android.soundface.video::InicioVideo
~tarea: DoVideo
#onCreate(Bundle savedInstanceState): void +listarImagenes(View view): void +addImagen(View view): void +listarSonidos(View view): void +verVideo(View view): void +verListaVideo(View view): void +generateVideo(View view): void #onStop(): void #onPause(): void

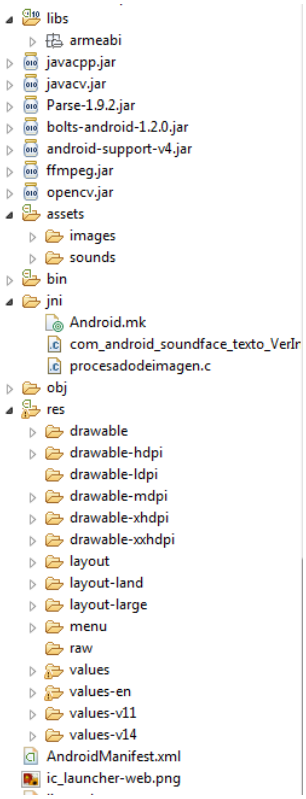
com.android.soundface.video::CustomListAdapterVideo

-context: Activity
-itemname: String[]
~mListener = new OnCheckedChangeListener() {
 	
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
if (isChecked) MainActivity.videosBorrar.add((Integer) buttonView.getTag()); else MainActivity.videosBorrar.remove((Integer) buttonView.getTag
}
}: OnCheckedChangeListener

+CustomListAdapterVideo(Activity context, String[] itemname): ctor +getView(int position, View view, ViewGroup parent): View

com.android.soundface.video::VerVideoGenerado
-mVideoView: VideoView
#onCreate(Bundle savedInstanceState): void +compartirVideo(View view): void #onDestroy(): void #onStop(): void +onBackPressed(): void

2. <u>Por otro lado el resto de componentes:</u>



Estos componentes son:

2.1 Libs : se trata de las librarías usadas para el tratamiento de imágenes y generación de vídeos : “javacv”, “javaccp”, “ffmpeg”, “opencv”. Además de las librerías usadas para las notificaciones push : “parse”, “bolts-android”. También contiene la carpeta armeabi, esta carpeta contiene los archivos .so que son necesarios para la generación de vídeos asociados a opencv y ffmpeg. También contiene el archivo “libprocesadodeimagen.so” que se ha generado para el tratamiento de imágenes, en concreto, para conversión de los colores de una imagen en blanco/negro y en sepia.

2.2 Assests : esta carpeta contiene otras subcarpetas images y sounds que contienen las imágenes y sonidos precargados por la aplicación.

2.3 Jni: contiene otros archivos necesarios para el tratamiento de imágenes, en concreto, para conversión de los colores de una imagen en blanco/negro y en sepia. En este caso se trata de los archivos de código nativo usados:
- Android.mk
- com_android_soundface_texto_VerImagenGenerada.h
- procesadodeimagen.c

2.4 res: es la carpeta generada por los proyectos Android. En nuestro caso tenemos layouts y layouts-land para la orientación de la pantalla. También tenemos values y values-en para poder usar dos idiomas (inglés y español). Contiene también algunos menús contextuales

2.5 AndroidManifest.xml : archivo descriptivo de android

Otros aspectos a destacar del diseño:

La capa de presentación se ha diseñado para las dos orientaciones posibles, sólo para teléfonos móviles (no se han incluido layouts para tablets), se han tenido que modificar algunas activities para que se adapten a ambas orientaciones y se ha hecho necesario que haya un paso de parámetros entre orientaciones para que no se pierda manejabilidad.

Se ha intentado hacer lo más manejable posible aunque mi ausencia de conocimientos en usabilidad ha desembocado en una interpretación personal de manejabilidad, creo que la solución tomada es bastante amigable.

Se han incluidos idioma inglés y castellano.

La capa UI contiene todas las clases java. Se han usado Activities, Adapters, BaseAdapters, ArrayAdapter, SQLITE, Fragments, FragmentActivity, Application, Views. También se ha usado código nativo.

Se ha intentado usar el mayor número de componentes de los estudiados en el diploma de modo que se justifique lo aprendido en el proyecto.

Modelo de datos

Esquema de la base de datos:

En nuestro caso la base de datos de ha usado simplemente para almacenar los sonidos e imágenes que en un principio están en assets, de modo que el sistema siempre tenga los sonidos e imágenes en base de datos. Esto es porque en la funcionalidad de bajar imágenes desde la notificación push enviada, la imagen bajada de la red, se almacena en base de datos, de esta manera el sistema siempre sabe que las imágenes disponibles están en el mismo lugar, en base de datos.

La base de datos contiene dos tablas: IMÁGENES y SONIDOS

Los script de creación son:

```
CREATE TABLE imagenes (id_imagen INTEGER PRIMARY KEY AUTOINCREMENT,
                   nombre_imagen TEXT, desc_imagen TEXT, imagen BLOB)

CREATE TABLE sonidos (id_sonido INTEGER PRIMARY KEY AUTOINCREMENT,
                   nombre_sonido TEXT, desc_sonido TEXT, sonido BLOB)
```

Ambas tablas son similares, contienen:

id_imagen/id_sonido : identificados único de cada registro

nombre_imagen/nombre_sonido : nombre de la imagen/sonido

desc_imagen/desc_sonido: descripción de la imagen/sonido

imagen/sonido : imagen/sonido propiamente dichos, es decir, el BLOB que contiene la información de la imagen o el sonido

Al instalar la aplicación se hace una carga inicial de datos en ambas tablas con los registros precargados, tanto de imágenes como de sonidos.

No se han usado servicios web.

Vistas

Pantalla inicial:

IMAGEN 1

VIDEO

Pulsando “Acciones Vídeo” se muestra la pantalla principal de esta sección:

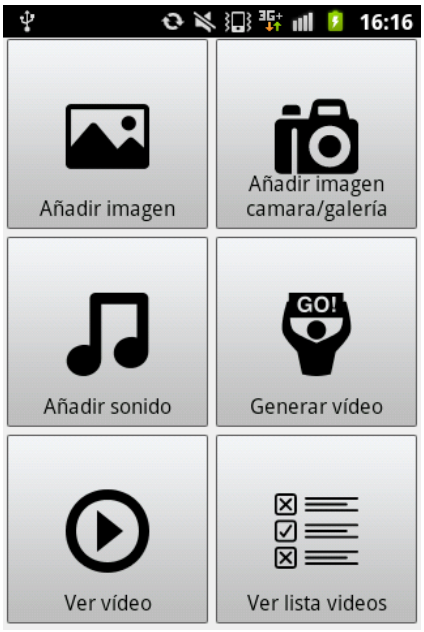


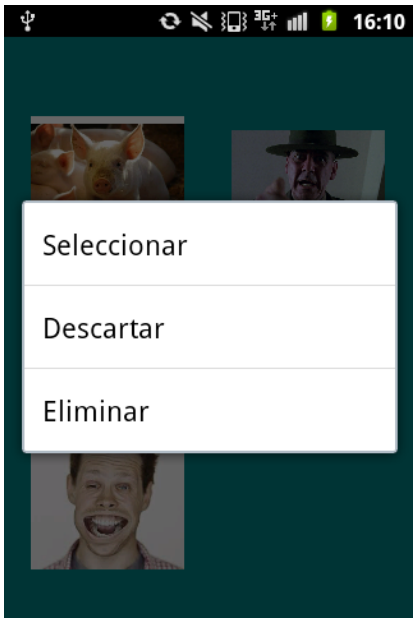
IMAGEN 1.1

Si desde la IMAGEN 1.1 (menú principal de acciones vídeo) pulsamos en "Añadir Imagen":

Aparecen las imágenes preseleccionadas y cargadas por la app. Si pulsamos en una de las imágenes se ampliará y aparecerá un botón "Seleccionar" que permitirá escoger la imagen para posteriormente generar un vídeo con ella.

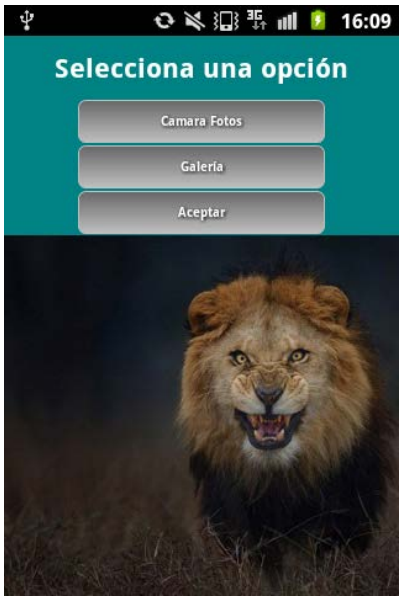
IMAGEN 1.2

Si sobre esta última IMAGEN 1.2 pulsamos largo (onlongclick) no aparece un menú:



En este menú se nos ofrece la posibilidad de "Seleccionar" la imagen igualmente que lo explicado en el punto anterior, "Descartar" para seguir navegando y "Eliminar" si queremos borrar esa imagen de la app.

Si desde la IMAGEN 1.1 (menú principal de acciones vídeo) pulsamos en añadir Imagen Cámara/galería:

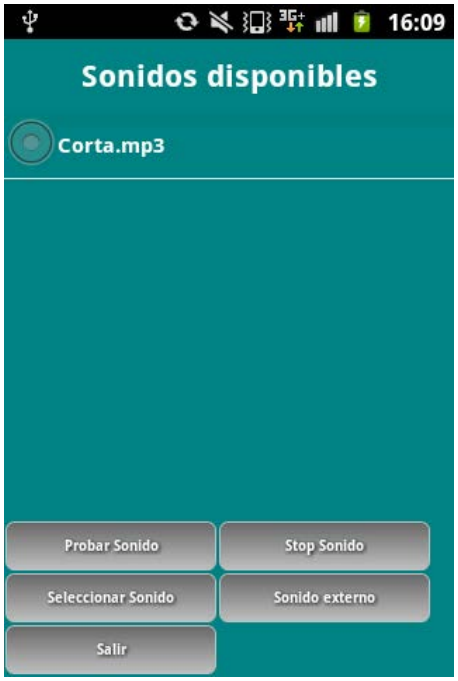


Si elegimos “Cámara fotos” o “Galería”, podremos escoger una foto o imagen. Al seleccionarla se nos mostrará tal y como lo vemos en la captura.

Si pulsamos “Aceptar” se seleccionará la imagen para la generación del vídeo.

IMAGEN 1.3

Si desde la IMAGEN 1.1 (menú principal de acciones vídeo) pulsamos en “Añadir sonido” accedemos a:



En esta pantalla aparecerá la lista de sonidos disponibles. Si lo seleccionamos tenemos la opción: de “Probar sonido” que hará sonar la melodía “Stop sonido”, que parará la melodía, “Sonido externo” que nos abrirá los sonido del teléfono para poder seleccionar uno de ellos, “Seleccionar sonido” para seleccionar un sonido ofrecido por la app o seleccionado del teléfono y finalmente “Salir”, para volver a la pantalla anterior

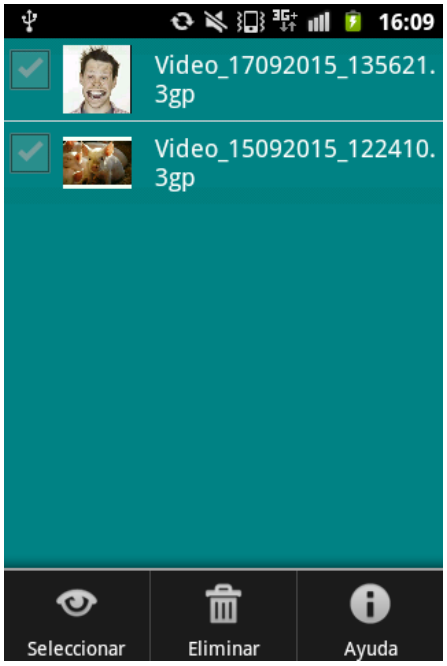
IMAGEN 1.4

Si desde la IMAGEN 1.1 (menú principal de acciones vídeo) pulsamos en “Generar Vídeo”, se generará el vídeo con la imagen y el sonido seleccionados, si pulsamos “Ver vídeo”, se mostrará:

Se muestra el video generado, con unos controles típicos de control de vídeo. Si se pulsa "Compartir" se podrá compartir el vídeo en redes sociales, correo, etc…

IMAGEN 1.5

Si desde la IMAGEN 1.1 (menú principal de acciones vídeo) pulsamos en "Ver lista vídeos", se mostrará:



Se muestran una lista de los vídeos generados hasta el momento con un menú contextual con las opciones "Seleccionar" para ver el vídeo, "Eliminar", para eliminar el vídeo y "Ayuda" para ver una pequeña ayuda.

IMAGEN 1.6

IMAGEN+TEXTO

Pulsando “Acciones imagen” desde la pantalla principal IMAGEN 1 se muestra la pantalla principal de esta sección:

IMAGEN 2

Si pulsamos en “Imagen predeterminada” y/o “Añadir imagen cámara/galería”, se desencadenan la mismas acciones que para la sección de vídeo, sólo que la selección de imagen no tendrá como finalidad la generación de un vídeo, sino la generación de una imagen+texto.

Si pulsamos sobre “Añadir texto” no aparece la pantalla:

En esta pantalla añadiremos el texto a poner sobre la imagen seleccionada. Hay dos textos superiores (uno debajo de otro) y un texto inferior, todos opcionales.

Un título para la imagen y las diferentes opciones para enriquecer el texto (ver manual de usuario).

Al pulsar aceptar se guarda el texto para la generación final.

IMAGEN 2.1

Si desde el menú principal de imagen+texto (IMAGEN 2) pulsamos “Generar imagen” se generará la imagen y al pulsar en “Ver imagen” podremos ver el resultado:

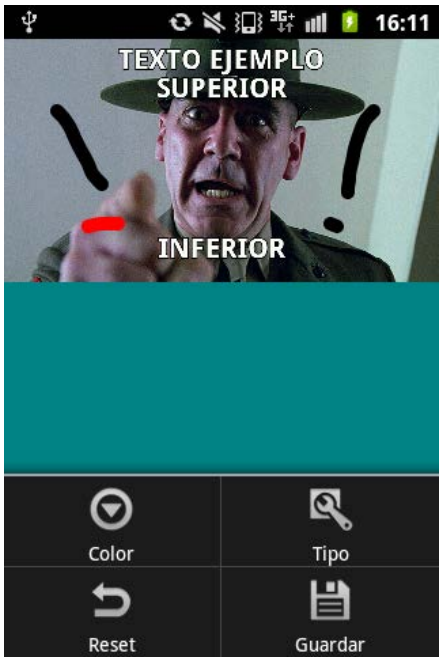
Al ver el resultado podremos “Eliminar/Descartar imagen”, de este modo no se guardará (si no se pulsa se guardará por defecto). “Compartir” para compartir en redes sociales.

Contiene un menú contextual para pasar a blanco y negro, sepia, deshacer los cambio y/o guardar los cambios.

Si pulsamos sobre la imagen esta se amplía, para volver se vuelve a pulsar

IMAGEN 2.2

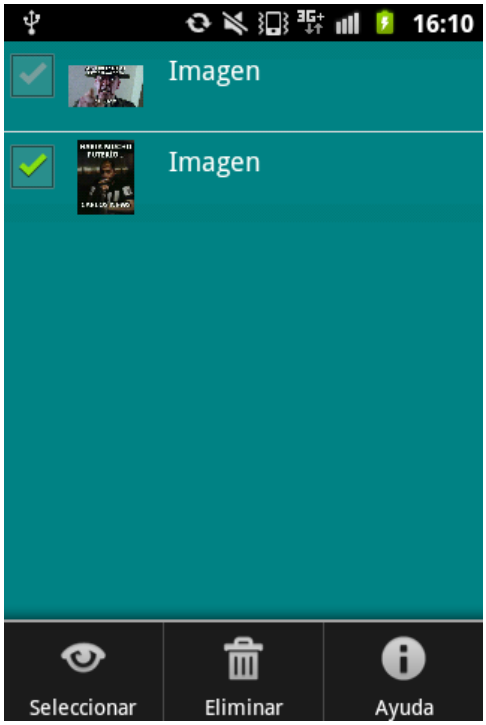
Si desde esta última pantalla pulsamos “Dibujar”, nos lleva a la pantalla:



En esta pantalla podremos dibujar con el dedo. Contiene un menú para elegir color y grosor del trazo. Podemos guardar o descartar los cambios.

IMAGEN 2.3

Si desde la IMAGEN 2 (menú principal de acciones imagen) pulsamos en "Ver lista imágenes", se mostrará una lista de imágenes generadas hasta el momento:



Si se selecciona una imagen a través del menú tendrá el mismo flujo que la parte explicada a través de las imágenes 2.2 y 2.3.

IMAGEN 2.4

DOWNLOAD

Si se recibe una notificación push obtendremos la notificación de este modo:

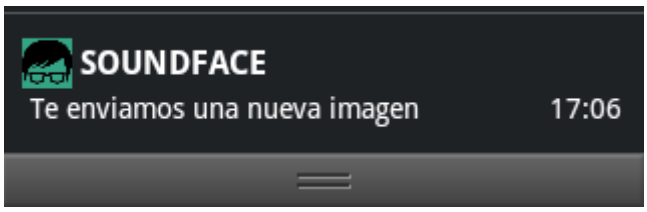


IMAGEN 3

Al pulsar en ella se abrirá una pantalla:

En esta pantalla existe la opción de guardar la imagen enviada (se mostrará a partir de ese momento como imagen predeterminada) o descartarla.

IMAGEN 3.1

Capítulos adicionales

Breve descripción de aspectos del trabajo

La aplicación SoundFace consiste en una aplicación capaz de generar vídeos e imágenes con texto.

1. Generalidades

Se ha tratado de usar la mayor parte de componentes que se han visto a lo largo de este diploma de especialización. Algunos otros componentes (como por ejemplo animaciones) no se han usado puesto que se pusieron de manifiesto en la práctica entregada en la asignatura de fundamentos, se ha hecho más hincapié en la parte de avanzado.

2. Vistas

En las vistas, tanto de inicio de las acciones de imágenes como en las de inicio de vídeo se ha usado GridView. En la vista que muestra las imágenes predeterminadas se han usado fragments. Para las listas de datos se ha utilizado BaseAdapter y ArrayAdapter

3. Vídeos

Para la generación de vídeos, se ha optado por la utilización de las librerías “opencv” y “javacv”.

La generación de vídeo propiamente dicha se hace usando threads, en concreto AsyncTask.

Describimos más en detalle la generación de vídeos de la siguiente manera:

- Se toma la imagen seleccionada y se guarda en el almacenamiento externo del teléfono, concretamente en el directorio “Pictures”.
- Se toma de manera análoga el sonido seleccionado y se guarda en el almacenamiento externo del teléfono, concretamente en el directorio “Music”.
- Se utilizan las clases FFmpegFrameGrabber para obtener estos datos y pasarlos a FFmpegFrameRecorder, este objeto se encarga de realizar el vídeo “juntando” ambos objetos.

4. Imágenes con texto

Para la generación de imágenes con texto se han utilizado el objeto Paint, Rect, Canvas, Color, Typeface… entre otros. La generación de la imagen final propiamente dicha, se hace usando threads, en concreto AsyncTask.

Para la conversión de la imagen a escala de grises o sepia se ha utilizado programación en código nativo Android NDK. Por lo que se hace necesaria la instalación de OpenCVManager.

Para poder dibujar con el dedo sobre la imagen generada se ha usado la clase DrawingView.

Describimos más en detalle la generación imagen más texto:

- Se seleccionan las áreas para "pintar" el texto con el objeto "android.graphics.Rect".
- Se calcula la posición donde colocar cada texto introducido por el usuario.
- Se aplican las opciones seleccionadas de tamaños, color, tipo de letra, etc…
- Se utilizan la clase "android.graphics.Canvas" para "pintar" el texto sobre la imagen.

5. Download por notificación recibida

La aplicación puede recibir una notificación con el texto:

"SOUNDFACE: Te enviamos una nueva imagen"

Si pulsamos la notificación se abre la aplicación con la imagen que se envía, se puede descargar (y formará parte de las imágenes de la aplicación), o se puede descartar.

Para el envío de las notificaciones PUSH a través de parse.com hay un apartado específico.

Describimos más en detalle la obtención de imagen por notificación:

- Leemos el objeto JSON que nos llega de la notificación push.
- Obtenemos un BitMap de la url proporcionada en JSON y la mostramos en el ImageView de la vista.
- Si el usuario lo considera oportuno de guarda la imagen transformada en byte[] en la base de datos.

Conclusiones

Grado de cumplimiento de los objetivos planteados.

Considero que siendo mi objetivo la puesta en práctica de conocimientos adquiridos en este diploma, el grado de cumplimiento es alto. Se han utilizado la mayoría de tecnologías explicadas en dicho diploma. Si algunas de ellas se han quedado fuera ha sido o por la imposibilidad de llevarla a cabo por no encajar en el proyecto (dispositivos wearable, Android

TV, Google Glass…) o porque se llevaron a cabo en el proyecto realizado para la asignatura de Fundamentos (animaciones, preferencias…) o porque son demasiado específicas.

Líneas abiertas.

Tras la realización del proyecto quedan como líneas abiertas las siguientes:

- Generación de vídeos con más de una imagen. La implementación es sencilla, pero no he tenido tiempo de probarlo con rigor. Habría que hacer pruebas para ver el rendimiento.
- Sección de texto sobre imágenes: Añadir mayor flexibilidad al generar texto sobre imagen. Sería interesante poder situar el texto justo donde el usuario desea. Actualmente se ofrecen unas alternativas generales.
- Generación de vistas con un aspecto visual más cuidado. He priorizado la funcionalidad sobre el aspecto visual. Se podría mejorar mucho.

Consideraciones personales.

Considero que el proyecto generado cumple con lo requerido. Sin embargo me resulta sólo un punto de partida para un proyecto final más ambicioso.

Quizá en el posterior desarrollo del master se puedan abordar más cuestiones al respecto.

Anexos

Código fuente en GitHub

Url del proyecto SoundFace. Contiene las fuentes del proyecto:

https://github.com/kakoplay/SoundFace

Manual de usuario

La aplicación SoundFace se presenta como una aplicación capaz de generar vídeos e imágenes con texto a partir de imágenes predeterminadas, sonidos predeterminados, imágenes de galería, imágenes por cámara e imágenes por descarga. Para la explicación del funcionamiento separaremos la casuística en dos partes: vídeos e imágenes.

1. Vídeos

Para la generación de vídeos, desde la pantalla principal se accederá a través del botón "Acciones vídeo". Aparecerán una serie de botones con el fin de generar un vídeo, para ello primero deberemos seleccionar una imagen y después un sonido.

Para seleccionar una imagen tenemos dos opciones:

- A través del botón "Añadir imagen": aparecerán las imágenes cargadas por defecto por la aplicación, si hacemos click sobre una imagen, aparecerá dicha imagen más grande en detalle y un botón "Seleccionar imagen", si se pulsa ese botón la imagen que seleccionada para la generación del video.
  Si en lugar de click, hacemos un click largo aparecen tres opciones: "Seleccionar", si se pulsa la imagen que seleccionada para la generación del video, "Descartar", se usa para salir de ese submenú y "Eliminar", si se pulsa eliminará esa imagen de nuestra aplicación definitivamente.
- A través del botón "Añadir imagen cámara galería": aparecen tres botones. El primero "Cámara fotos" permite abrir la funcionalidad de la cámara de fotos del teléfono y hacer una foto que quedará pre-seleccionada. El segundo botón "Galería", accede a la galería de imágenes del teléfono y pre-seleccionará la imagen deseada. El tercer botón "Aceptar" selecciona la imagen elegida, bien sea por cámara o por galería.

Una vez hemos seleccionado la imagen, podemos volver y pulsar "Añadir sonido", aparecerá una lista de sonidos precargados por la aplicación, se puede seleccionar uno y probar su sonido. Una vez hayamos elegido, pulsamos "Seleccionar sonido" y el sonido quedará seleccionado para el vídeo. También podemos pulsar "Sonido externo" de este modo podemos seleccionar un sonido que tengamos almacenado en el móvil. Luego pulsamos salir y volvemos a la pantalla principal de la utilidad vídeo.

Una vez seleccionados imagen y sonido podemos pulsar "Generar vídeo", se generará automáticamente el vídeo. Una vez generado si pulsamos "Ver vídeo" podremos ver el resultado, habrá un botón compartir que permite compartir el vídeo a través de redes sociales, mail, etc..

Si pulsamos "Ver lista vídeos" podremos ver una lista con los vídeos generados hasta el momento. Si seleccionamos uno de ellos y usamos el menú contextual, tendremos la opción de seleccionarlo para verlo y/o compartirlo, la opción de eliminarlo y una opción de ayuda.

2. Imágenes con texto

Para la generación de imágenes con texto, desde la pantalla principal se accederá a través del botón "Acciones imágenes". Aparecerán una serie de botones con el fin de generar una imagen con texto, para ello primero deberemos seleccionar una imagen y después un texto.

Para seleccionar una imagen tenemos dos opciones:

- A través del botón "Imagen predeterminada": aparecerán las imágenes cargadas por defecto por la aplicación, si hacemos click sobre una imagen, aparecerá dicha imagen más grande en detalle y un botón "Seleccionar imagen", si se pulsa ese botón la imagen que seleccionada para la generación del video.
  Si en lugar de click, hacemos un click largo aparecen tres opciones: "Seleccionar", si se pulsa la imagen que seleccionada para la generación del video, "Descartar", se usa para salir de ese submenú y "Eliminar", si se pulsa eliminará esa imagen de nuestra aplicación definitivamente.
- A través del botón "Añadir imagen cámara galería": aparecen tres botones. El primero "Cámara fotos" permite abrir la funcionalidad de la cámara de fotos del teléfono y hacer una foto que quedará pre-seleccionada. El segundo botón "Galería", accede a la galería de imágenes del teléfono y pre-seleccionará la imagen deseada. El tercer botón "Aceptar" selecciona la imagen elegida, bien sea por cámara o por galería.

Una vez hemos seleccionado la imagen, podemos volver y pulsar "Añadir texto", aparecerá una pantalla con los siguientes campos:

- Un "título" por defecto que es modificable y que identificará la imagen final generada, este título puede ser modificado e inicialmente tiene el formato "Imagen_X" (es obligatorio). La X es un secuencial que usa las preferencias para guardar el último número utilizado y aumenta uno más.
- Dos campos "texto superior", estos campos permiten introducir dos textos que aparecerían en la parte superior de la imagen final, uno debajo del otro. (es opcional)
- "Texto inferior" permite introducir un texto que aparecerían en la parte inferior de la imagen final. (es opcional)
- "Color borde", establece el color del borde de la letra que conformará el texto.
- "Color interior", establece el color del interior de la letra que conformará el texto.
- "Tipo de letra", establece el tipo de letra que conformará el texto.
- "Tamaño", establece el tamaño que conformará el texto.

Una vez seleccionados los datos para el texto, pulsamos "Aceptar" y quedará el texto seleccionado para la generación de la imagen final.

Una vez seleccionados imagen y texto podemos pulsar "Generar imagen", se generará automáticamente la imagen final. Una vez generada si pulsamos "Ver imagen" podremos ver el resultado, habrá un botón "Compartir" que permite compartir el vídeo a través de redes sociales, mail, etc. Otro botón "Eliminar/Descartar imagen" que hará que la imagen no se guarde. Y un tercer botón "Dibujar" que al pulsarlo nos ofrecerá la opción de dibujar con el dedo sobre la imagen generada. Al ver la imagen final generada, también tenemos un menú contextual que nos permite convertir la imagen a escala de grises o a sepia y guardar esos cambios.

Si pulsamos sobre la imagen esta se amplía, para volver se vuelve a pulsar.

En esta última opción “Dibujar” hay un menú contextual para poder seleccionar el color y el grosor del trazo para dibujar. También una opción reset, para eliminar lo dibujado y una opción guardar para sobreescribir la imagen con los trazos dibujados.

Si pulsamos “Ver lista imágenes” podremos ver una lista con las imágenes generadas hasta el momento. Si seleccionamos una de ellas y usamos el menú contextual, tendremos la opción de seleccionarlo para verlo, dibujar, eliminar y/o compartirlo, la opción de eliminarlo y una opción de ayuda.

3. **Download por notificación recibida**

La aplicación puede recibir una notificación con el texto:

“SOUNDFACE: Te enviamos una nueva imagen”

Si pulsamos la notificación se abre la aplicación con la imagen que se envía, se puede descargar (y formará parte de las imágenes de la aplicación), o se puede descartar.

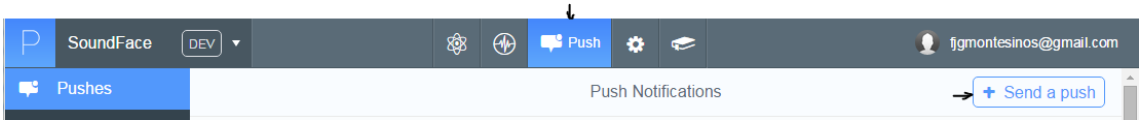
Notificaciones Push

Para la generación de notificaciones PUSH se ha usado la web preparada para ello parse.com, a través de esta web se pueden enviar este tipo de notificaciones.

Para su uso, primero hemos de registrarnos en la web, una vez dentro damos de alta nuestra APP, al darla de alta se nos proporciona una Aplication-key y una client-key, esos dos valores son los que usaremos en nuestra aplicación para registrarla para la recepción de las notificaciones que enviemos desde esa página.

Para probar el funcionamiento de las notificaciones se proponen dos maneras:

- Que el testeador de las notificaciones push se dé de alta en parse.com, y cree una aplicación, de esta manera podrá obtener una aplication-key y una client-key. Estos valores los podrá sustituir en la clase
  Una vez sustituidos, el testeador podrá enviar las notificaciones a través de la web parse.com. En esta aplicación en envío se hace usando JSON. Si entramos en la sección de envío de una notificación:



  Y posteriormente activamos la opción JSON:

Podremos enviar una notificación push, aquí se especifica un ejemplo de envío:

```
{
  "alert": "Te enviamos una nueva imagen",
  "badge": "http://1.bp.blogspot.com/-
zr4CYIlq5qg/VfHQgmHi1II/AAAAAAACpEo/Q0PfdL6VkPM/s1600/fond
os%2By%2Bwallpapers%2Bde%2Bmascotas%2Bperro%2Bjugando%2B-
%2Bdog%2Bplaying.jpg",
  "name": "perrito.jpg",
  "title": "SOUNDFACE"
}
```

Una vez introducido este JSON sólo hay que darle a “Enviar”.

- Que el testeador, una vez instalada la aplicación, establezca una/s fecha/s y hora en las que se deban enviar las notificaciones push. El alumno enviará esas notificaciones a través de parse.com en los horarios establecidos.