

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE
TELECOMUNICACIÓN UNIVERSIDAD DE MÁLAGA



PROYECTO FIN DE CARRERA

APLICACIÓN ANDROID PARA
MEDICIÓN DE RUIDO AMBIENTAL
Y SU REPRESENTACIÓN
GEOLOCALIZADA

INGENIERÍA TÉCNICA DE
TELECOMUNICACIÓN

Málaga, 2015

Guillermo Orellana Ruiz

ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD DE MÁLAGA

Titulación: Ingeniería Técnica de Telecomunicación
Especialidad en Sonido e Imagen

Reunido el tribunal examinador en el día de la fecha, constituido por:

D./D^a. _____

D./D^a. _____

D./D^a. _____

para juzgar el Proyecto Fin de Carrera titulado:

**APLICACIÓN ANDROID PARA MEDICIÓN DE RUIDO
AMBIENTAL Y SU REPRESENTACIÓN
GEOLOCALIZADA**

del alumno *D./D^a. Guillermo Orellana Ruiz*
dirigido por *D./D^a. Enrique Márquez Segura*

ACORDÓ POR _____ OTORGAR LA

CALIFICACIÓN DE _____

y, para que conste, se extiende firmada por los componentes del tribunal, la presente diligencia.

Málaga, a ____ de ____ de ____

El Presidente:

El Vocal:

El Secretario:

Fdo.:_____ Fdo.:_____ Fdo.:_____

Universidad de Málaga
Escuela Técnica Superior de Ingeniería de
Telecomunicación

APLICACIÓN ANDROID PARA MEDICIÓN DE RUIDO AMBIENTAL Y SU REPRESENTACIÓN GEOLOCALIZADA

REALIZADO POR
Guillermo Orellana Ruiz

DIRIGIDO POR
Enrique Márquez Segura

Dpto. de: Ingeniería de Comunicaciones (DIC)

Palabras clave: palabras, clave, del, proyecto, separadas, por, coma

Titulación: Ingeniería de Telecomunicación

Resumen: El presente proyecto consiste en desarrollar una aplicación móvil para el sistema operativo Android, cuya funcionalidad es la medición del nivel de ruido asociado a un punto geográfico, y la representación del mismo superpuesto a un mapa geográfico

Málaga, 4 de enero de 2015

En caso de dedicatoria,
se realiza con esta página.
No es obligatoria, si bien es recomendable.

El autor

Acrónimos

ETSIT	Escuela Técnica Superior de Ingeniería de Telecomunicación
UMA	Universidad de Málaga
PFC	Proyecto Fin de Carrera
TFG	Trabajo Fin de grado
TFM	Trabajo Fin de Máster

Índice

Acrónimos	IX
Prólogo	1
I Introducción	3
Introducción y visión general	5
Resumen y objetivos del Proyecto	7
Motivación	7
Estado del arte	7
Metodología y directrices seguidas	7
Estructura del documento	8
Ámbito de aplicación	8
II Desarrollo del proyecto	9
1 Introducción teórica	11
1.1 Acústica	12
1.1.1 Nivel de presión sonora	12
1.1.2 Control automático de ganancia (CAG)	12
1.1.3 Modulación por impulsos codificados	12
1.1.4 Mapa de calor	12
1.2 Android	12
1.2.1 Interfaz	12
1.2.2 Componentes de una aplicación	13
1.2.3 Vistas	16
1.2.4 Procesos e hilos de ejecución	17
1.2.5 Servicios de Google Play	19
1.2.6 Manejo de archivos	19

2	Implementación	21
2.1	Análisis de requisitos	21
2.1.1	Dispositivo Android	21
2.1.2	Micrófono externo	22
2.1.3	Herramientas de calibración/dispositivo de calibrado	22
2.2	Android	22
2.2.1	Estructura del código	22
2.2.2	Interfaz gráfica	22
2.2.3	Captura de audio	23
2.2.4	Calibración	23
3	Plan de pruebas y verificación	25
3.1	Calidad del software	25
3.2	Precisión de las medidas	25
3.3	Realización de los objetivos	25
III	Parte tercera.	27
	Conclusiones y líneas de trabajo futuras	29
IV	Apéndices	31
A	Código del programa	33
A.1	Primera sección	33
	Bibliografía	35
	Índice alfabético	37

Índice de figuras

1.1	Diagrama de la estructura del sistema operativo Android	13
1.2	Diagrama del ciclo de vida de una actividad en Android. Tomado de [and14].	15
1.3	Diagrama del ciclo de vida de un servicio. Tomado de [and14]	17

Índice de Tablas

3.1	Tabla de comparacion de mediciones	26
-----	--	----

Prólogo

El presente proyecto consiste en desarrollar una aplicación móvil para el sistema operativo Android, cuya funcionalidad es la medición del nivel de ruido asociado a un punto geográfico, y la representación del mismo superpuesto a un mapa geográfico.

Parte I

Introducción

Introducción y visión general

Contenido

Resumen y objetivos del Proyecto	7
Motivación	7
Estado del arte	7
Metodología y directrices seguidas	7
Estructura del documento	8
Ámbito de aplicación	8

Sinopsis

Éste es el capítulo de introducción, donde se explica todo lo que un lector externo necesita para entender el resto de la documentación.

El objetivo explica lo que persigue el proyecto, su finalidad.

El estado del arte explica la situación actual del entorno en el que este proyecto¹ se desenvuelve.

Las metodologías y directrices seguidas se centran en qué procedimientos se han utilizado durante el desarrollo del proyecto.

La estructura del documento describe los capítulos de los que se compone, incluyendo apéndices e información adicional.

Por último, el ámbito de aplicación completa el entorno de utilización del proyecto.

Aunque estos cinco apartados no son obligatorios, al menos es recomendable considerar estos conceptos en el capítulo de introducción como una guía básica.

Tampoco es obligatorio usar el entorno `LATEX minitoc` para cada capítulo.

En caso de no querer usarlo, tan sólo hay que comentar la línea `\minitoc`.

¹En adelante, se utiliza la palabra *proyecto* como sinónimo de TFG/TFM/PFC, según se aplique (Nota del autor).

Igualmente, esta sección inicial de Sinopsis no es obligatoria, se puede suprimir si no se desea.

Por extensión y en general, esta plantilla es una guía cuyo objetivo es facilitar la realización del proyecto, no un reglamento estricto ni rígido.

Resumen y objetivos del Proyecto

Cosas

Motivación

Es difícil encontrar actividades que no generen cierto nivel sonoro, ya sean naturales o producidas por el ser humano. Estos sonidos pueden ser categorizados en base a muchos parámetros, como lugar, duración, tipo, etc. pero cuando el sonido es molesto y no deseado, estamos hablando de ruido. Este ruido puede llegar a ser perjudicial para la audición, el físico y el psique de seres vivos, convirtiéndose en contaminación acústica.

La contaminación acústica es hoy día un factor clave en el deterioro de la calidad ambiental de un territorio. Según estudios de la Unión Europea (2005) “80 millones de personas están expuestas diariamente a niveles de ruido ambiental superiores a 65dBa y otros 170 millones, lo están a niveles entre 55-65dBa”

Normalmente, las mediciones de ruido conllevan el uso de múltiples y muy caros aparatos tales como sonómetro, calibrador, pantalla anti-viento, trípode, micrófono, preamplificador, etc. pero se puede tomar provecho de la alta penetración en el mercado que tienen los teléfonos inteligentes y conseguir, a costa de pérdida de precisión, una mayor facilidad y simplicidad a la hora de realizar mediciones.

La vasta mayoría de estos dispositivos incorporan de serie un receptor GPS, micrófono y almacenamiento, además de acceso a internet y pantalla táctil.

Estado del arte

Un proyecto se realiza sobre un estado de la técnica que debe explicarse para entender mejor conceptos tales como los problemas existentes o cuáles son las soluciones que se emplean hasta la fecha actual.

El estado del arte, a veces llamado estado de la técnica, suele estar presente en este tipo de documentos.

Metodología y directrices seguidas

Durante la elaboración del proyecto, se siguen procedimientos que el lector necesita conocer para entender de forma integral todo el documento.

Estructura del documento

Introducción teórica

En el capítulo 2 se hará una introducción a los conceptos teóricos requeridos para la comprensión del trabajo realizado. Se explicarán los parámetros acústicos tratados a lo largo de la memoria. A su vez, también se explicarán conceptos relativos a la programación en Android, tratados a lo largo del proyecto.

Implementación

En el capítulo 3 se explicará cómo se ha llevado a cabo toda la implementación de la aplicación, empezando por un análisis de los requisitos necesarios para la realización de esta, y siguiendo con los aspectos relevantes a la programación de la aplicación: métodos, interfaz gráfica, algoritmos, etc.

Plan de pruebas y verificación

Aquí se explicarán los procedimientos llevados a cabo para cerciorar el correcto funcionamiento de la aplicación.

Conclusiones y trabajo futuro

En este apartado se pretende representar las conclusiones obtenidas durante la realización de este proyecto así como plasmar posibles ideas que se consideran interesantes de cara a una futura continuidad en el desarrollo de la aplicación.

Ámbito de aplicación

Por último, completando los apartados anteriores, se explican las áreas de las que se compone el proyecto.

Parte II

Desarrollo del proyecto

Capítulo 1

Introducción teórica

Contenido

1.1	Acústica	12
1.1.1	Nivel de presión sonora	12
1.1.2	Control automático de ganancia (CAG)	12
1.1.3	Modulación por impulsos codificados	12
1.1.4	Mapa de calor	12
1.2	Android	12
1.2.1	Interfaz	12
1.2.2	Componentes de una aplicación	13
1.2.3	Vistas	16
1.2.4	Procesos e hilos de ejecución	17
1.2.5	Servicios de Google Play	19
1.2.6	Manejo de archivos	19

Sinopsis

1.1. Acústica

1.1.1. Nivel de presión sonora

1.1.2. Control automático de ganancia (CAG)

1.1.3. Modulación por impulsos codificados

1.1.4. Mapa de calor

1.2. Android

Android es un sistema operativo, en sus inicios concebido para dispositivos móviles con pantalla táctil, que ha evolucionado en una plataforma que actualmente también engloba relojes inteligentes, televisores, automóviles e incluso electrodomésticos.

Está basado en el núcleo de Linux, sobre el que corre el entorno de ejecución propio de Android (ya sea Dalvik o ART), y este a su vez ejecuta el código de las aplicaciones, escritas mayoritariamente en Java, aunque se permiten extensiones en C/C++

1.2.1. Interfaz

La interfaz de usuario por defecto de Android está basada en una manipulación directa, usando entrada táctil con gestos que vagamente corresponden a acciones físicas reales, tales como deslizar, golpear, pellizcar... para manipular objetos en pantalla. Además, la mayoría de dispositivos dispone de un teclado virtual, manipulado de la misma manera.

La respuesta del sistema a la entrada del usuario está diseñada para ser inmediata y dar una sensación de fluidez, utilizando las capacidades de vibración presentes en la mayoría de los dispositivos para proveer una respuesta háptica (no visual, no auditiva). Adicionalmente, algunas aplicaciones utilizan la información proveída por sensores tales como acelerómetros, giroscopios y sensores de proximidad para responder a interacciones adicionales, como por ejemplo ajustar la orientación de la pantalla cuando el dispositivo se encuentra apaisado o controlar alguna parte de la aplicación basándose en el azimut relativo del dispositivo.

Una parte importante de la interfaz general de Android son las notificaciones, presentes en la barra de estado

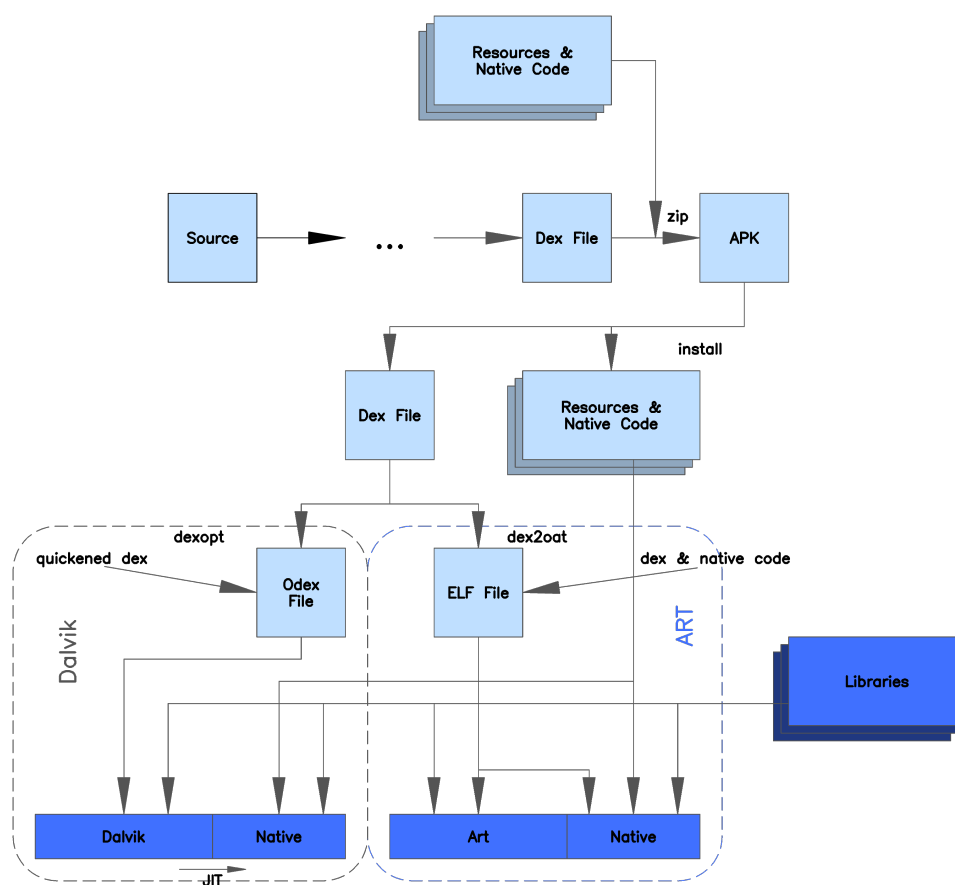


Figura 1.1: Diagrama de la estructura del sistema operativo Android

1.2.2. Componentes de una aplicación

Contexto

Uno de los conceptos más importantes cuando se utiliza la plataforma Android es el contexto, **Context**. La clase **Context** en si misma no es más que una interfaz a información global acerca del entorno de una aplicación, y como tal es abstracta. Sin embargo, es importante ser consciente de qué elementos representan un contexto válido y qué elementos no, ya que un contexto permite acceso a recursos y clases específicos de la aplicación, y llamadas al sistema para operaciones a nivel de aplicación, tales como lanzar actividades, emitir mensajes de difusión o recibirlos.

Actividades

En android, una actividad (**Activity**) representa una única cosa concreta que el usuario puede realizar en la aplicación. La mayoría de las actividades interaccionan con el usuario, por tanto la clase **Activity** se encarga de crear una ventana donde se puedan insertar los componentes de la interfaz de usuario. Aunque las actividades suelen ser vistas por el usuario como ventanas a pantalla completa, también pueden ser usadas en otras múltiples maneras, ya sea como ventanas flotantes, o incrustadas dentro de otra actividad (mediante un **ActivityGroup**)

Ciclo de vida de una actividad

Las distintas actividades de las distintas aplicaciones instaladas en el dispositivo Android son gestionadas en forma de una *pila de actividades*.

Cuando el sistema Android se inicia, se presenta al usuario una pantalla principal, desde donde puede lanzar varias acciones y aplicaciones. A partir de ahí, cuando una nueva actividad es empezada, se emplaza arriba de la pila y se convierte en la actividad en ejecución. Las actividades previas si las hubiera siempre permanecen por debajo en la pila, y no se traerán al frente hasta que la nueva actividad finalice.

Una actividad tiene cuatro estados básicos:

Activa

Una actividad está *activa* cuando está presente en primer plano en la pantalla, es decir, arriba de la pila. También se puede decir que la actividad está *En ejecución*.

Pausada

Si una actividad ha perdido el foco (ha dejado de estar en primer plano) pero todavía es visible, es decir, si una nueva actividad que no ocupa la totalidad de la pantalla o es transparente obtiene el primer plano, se encuentra *pausada*. Una actividad pausada se conserva completamente íntegra (mantiene todos los estados y se mantiene suscrita al gestor de ventanas) pero puede ser matada por el sistema en condiciones extremas de baja memoria disponible.

Parada

Si una actividad se encuentra oculta por completo, el sistema la deja *parada*. Mantiene estados e información de los miembros, pero sin embargo al no ser visible por el usuario es más probable que el sistema se deshaga de ella para liberar recursos cuando hagan falta.

Muerta

Cuando el sistema decide mover la actividad fuera de memoria, puede o bien

finalizarla o matar el proceso. Cuando sea mostrada de nuevo al usuario, debe ser completamente reiniciada y restaurada a su estado previo.

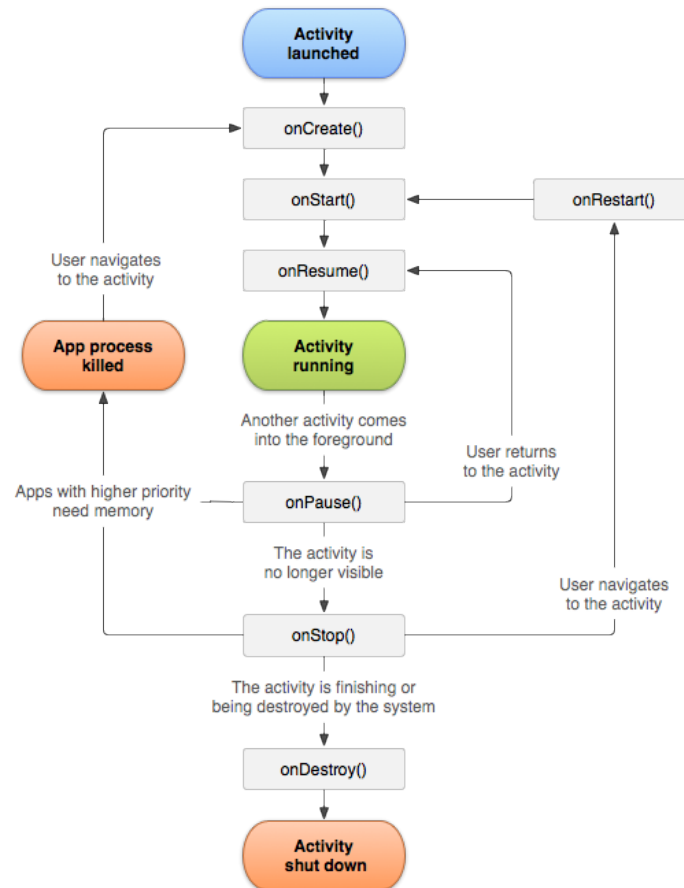


Figura 1.2: Diagrama del ciclo de vida de una actividad en Android. Tomado de [and14].

Servicios

Un Servicio (**Service**) es un componente de la aplicación que puede realizar tareas de larga duración en segundo plano y no provee ninguna interfaz de usuario. Otro componente de la aplicación puede iniciar un servicio y este continuará en marcha en segundo plano, incluso si el usuario cambia a otra aplicación distinta. Además, un componente puede adherirse (**bind**) a un servicio para interactuar con él, e incluso realizar comunicación inter-procesos (IPC por sus siglas en inglés, Inter-Process Communication). Por ejemplo, un servicio puede manejar llamadas de red,

reproducir música, realizar operaciones en el sistema de archivos, todo en segundo plano.

Un servicio puede tomar dos estados:

Started

Un servicio está en estado **Started** (empezado) cuando un componente de la aplicación, por ejemplo una actividad, lo empieza llamando al método `startService()`. Una vez empezado de esta manera, un servicio puede continuar en segundo plano de manera indefinida, incluso si el componente que lo empezó ha sido destruido. Normalmente, suelen ser servicios que realizan una única operación y no devuelven ningún resultado. Por ejemplo, puede descargar o subir un archivo en la red. Cuando la operación ha sido completada, el servicio debe pararse a si mismo.

Bound

Un servicio está en estado **Bound** (adherido) cuando un componente de la aplicación se adhiere a él llamando al método `bindService()`. Un servicio adherido ofrece una interfaz servidor-cliente que permite a los componentes interactuar con el servicio, mandar peticiones, obtener resultados, e incluso hacerlo entre distintos procesos mediante comunicación inter-proceso (IPC). Un servicio adherido solamente es activo durante el tiempo que otro componente esté adherido a él. Varios componentes pueden estar adheridos en un momento dado al servicio, pero cuando todos se desadhieren del servicio, el servicio es destruido.

Hay que tener cuidado, dado que un servicio corre en el hilo principal de ejecución del proceso que lo llama, a no ser que se especifique lo contrario. Esto implica que, si el servicio va a realizar alguna tarea intensiva en CPU, o alguna operación bloqueante, se debe de crear un nuevo hilo de ejecución dentro del servicio para ese propósito. De no hacerlo, se corre el riesgo de que la aplicación deje de responder y sea matada por el sistema operativo.

1.2.3. Vistas

La interfaz gráfica de usuario en una aplicación Android está construida usando una jerarquía de vistas, objetos de la clase `View`, y grupos de vistas, objetos de la clase `ViewGroup`. Los objetos `View` suelen ser artilugios (widgets) de la interfaz de usuario, tales como botones o campos de texto, y los objetos `ViewGroup` son contenedores invisibles que definen cómo se posicionan las vistas que dependen de ellos, por ejemplo dispuestas en forma de rejilla o lista vertical.

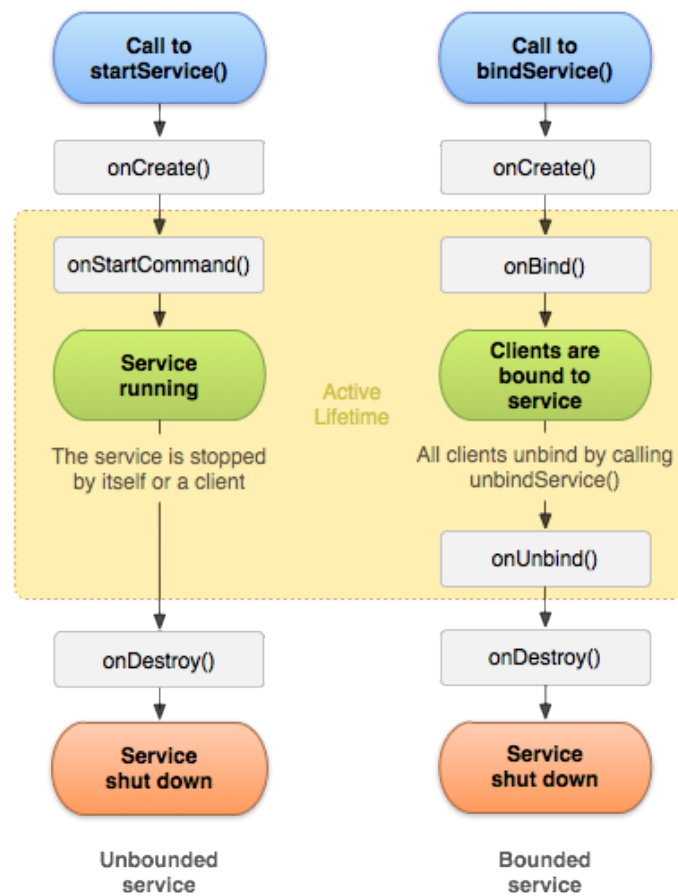


Figura 1.3: Diagrama del ciclo de vida de un servicio. Tomado de [and14]

Android provee un vocabulario XML que se corresponde con las subclases de `View` y `ViewGroup`, y permite definir la interfaz de usuario en XML usando una jerarquía de elementos de interfaz de usuario.

1.2.4. Procesos e hilos de ejecución

En sistemas operativos, son básicos los conceptos de proceso e hilo de ejecución. En Android, el sistema operativo comienza un nuevo proceso Linux por cada primer componente de cada aplicación, con un único hilo de ejecución. Por defecto, todos los componentes de la misma aplicación corren en los mismos proceso e hilo de ejecución, el hilo principal de ejecución (*main thread* en inglés).

En caso de que un componente de una aplicación sea inicializado, y ya exista un proceso para dicha aplicación en el sistema que otro componente de la misma aplicación ha inicializado, el nuevo componente se inicializa dentro del proceso original.

y usa el mismo hilo de ejecución. Sin embargo, se puede configurar una aplicación de manera que diferentes componentes corran en procesos separados, y siempre se pueden crear hilos de ejecución adicionales para cualquier proceso.

Procesos

Como ya ha sido expuesto anteriormente, por defecto todos los componentes de la misma aplicación corren en el mismo proceso, y la mayoría de las aplicaciones no deberían de cambiarlo. Sin embargo, es posible controlar qué proceso pertenece a qué componente de ser necesario.

El sistema operativo puede decidir apagar un proceso, cuando la cantidad de memoria disponible sea baja y haya requerimiento de ella por otro proceso que sirva de manera más inmediata al usuario. En este caso, los componentes dentro de dicho proceso que es apagado, son destruidos. Cuando estos componentes sean necesarios de nuevo, un nuevo proceso será comenzado por el sistema operativo para ello.

Hilos de ejecución

Previamente se ha mencionado que todos los componentes de la misma aplicación corren en el mismo hilo de ejecución, el hilo principal de ejecución o *main thread*. Éste hilo es de suma importancia, dado que carga con la responsabilidad de despachar los eventos al widget de la interfaz de usuario que sea pertinente, incluyendo los eventos de dibujado en pantalla. Es también el hilo de ejecución en el que la aplicación interactúa con los componentes básicos de interfaz de usuario de Android, también conocidos como *Android UI toolkit* (ubicados dentro de los paquetes java `android.widget` y `android.view`). Por todo esto, no es extraño encontrar denominado este hilo de ejecución como el hilo de la interfaz de usuario, o *UI Thread*.

Dado que todos los componentes que corren en el mismo proceso son instanciados en el hilo de ejecución principal, las llamadas del sistema operativo a cada componente son despachadas desde dicho hilo. En consecuencia, todos los métodos que responden a retrollamadas del sistema (*system callbacks*), como por ejemplo para indicar que una tecla ha sido pulsada, siempre corren en el hilo principal de ejecución del proceso.

Cuando el usuario toca un botón en la pantalla, el hilo principal de la aplicación despacha el evento de toque al widget pertinente, que reacciona cambiando su estado a presionado y manda una petición de invalidación a la cola de eventos. El hilo de ejecución principal entonces desencola la petición y notifica al widget que debe de redibujarse.

Cuando una aplicación realiza trabajo intensivo en respuesta a una interacción del usuario, el modelo de hilo de ejecución único puede resultar en una falta de

rendimiento. Concretamente, de suceder todo el procesamiento en el hilo principal de ejecución e iniciar tareas de larga ejecución tales como acceso a red o consultas a bases de datos, resultará en un bloqueo completo de la interfaz de usuario y su correspondiente hilo principal. Cuando el hilo de ejecución está bloqueado, no se pueden despachar eventos, eventos de dibujado en pantalla incluidos. Desde el punto de vista del usuario, esto se traduce en una aplicación que parece colgarse. En peores casos, en los que el hilo principal de ejecución está bloqueado por más de unos cuantos segundos (cinco segundos en la actualidad) el sistema operativo entrará en acción y mostrará una pantalla explicando que la aplicación ha dejado de responder, y matará la aplicación bloqueada.

Para evitar dicha penalización en el rendimiento, deben usarse hilos de ejecución alternativos para toda tarea que bloquee la ejecución o sea de alta carga procedural.

1.2.5. Servicios de Google Play

1.2.6. Manejo de archivos

Capítulo 2

Implementación

2.1. Análisis de requisitos

Para el desarrollo de este proyecto se han tenido en cuenta una serie de requisitos previos mínimos, necesarios para una correcta implementación del mismo. Los requisitos funcionales de la aplicación fueron acordados en los siguientes:

- Capacidad de medir la magnitud del ruido ambiente
- Capacidad de determinar la posición del dispositivo
- Capacidad de asociar ambas mediciones
- Capacidad de almacenar los datos obtenidos
- Capacidad de mostrar los datos obtenidos sobre un mapa

Adicionalmente, los siguientes requisitos no funcionales fueron considerados:

2.1.1. Dispositivo Android

El desarrollo de este proyecto ha sido realizado y testeado en dos dispositivos. El primero, modelo HTC Desire HD poseedor de la versión 4.2.2 de Android y el segundo Google Nexus 5, bajo la versión Android 5.0, lo cual garantiza que la aplicación conserva toda su funcionalidad en los modelos más modernos, tanto en software como en hardware.

No obstante, también se ha comprobado su funcionalidad en una rango de dispositivos más amplio, tales como Samsung Galaxy S3, Samsung Galaxy Nexus, Sony Xperia P, no mostrando pérdida alguna de funcionalidad.

La versión mínima de Android requerida para el correcto funcionamiento de esta aplicación, es el nivel de API 15, correspondiente a Android 4.0.3. Es posible

hacerla funcionar en niveles más bajos (antiguos), pero requiere esfuerzo adicional, tal y como se esboza en el apartado de Conclusiones y Trabajo Futuro.

2.1.2. Micrófono externo

Los micrófonos empotrados en los teléfonos móviles tienen un objetivo muy claro y marcado, que es la transmisión de voz vía redes celulares, y muestran cierto sesgo en diseño cuando se les intenta usar para otro propósito.

Un perfecto ejemplo de ello es el control automático de ganancia (CAG). EL CAG es de verdadera utilidad para mejorar los niveles sonoros realizando una llamada, pero entra en conflicto con el propósito de este proyecto, ya que necesita de una señal sin pre-procesamiento alguno. El efecto de compresión que realiza el CAG, proporciona unas medidas sin sentido alguno.

Adicionalmente, el tamaño y posición empotrada del mismo, los hace muy sensibles a interferencias indeseadas tales como la vibración del propio teléfono.

Uno de los modelos utilizados en el desarrollo, el Google Nexus 5, permite la desactivación del CAG, por tanto se obtienen medidas aceptables. No obstante, para mejores resultados, se debe de usar un micrófono externo sin CAG.

2.1.3. Herramientas de calibración/dispositivo de calibrado

2.2. Android

2.2.1. Estructura del código

La estructura de la aplicación sigue la estructura estándar de proyecto Android bajo el entorno de desarrollo Android Studio. Se diferencia tres grupos principales: código, organizado en paquetes de Java, recursos, los cuales se clasifican por tipo y versión a la que van orientados, y varios archivos de instrucciones de compilación escritos en el lenguaje específico de dominio (en inglés DSL, domain specific language) de Gradle.

2.2.2. Interfaz gráfica

La interfaz gráfica de esta aplicación se ha desarrollado siguiendo las guías de diseño vigentes para la versión del sistema KitKat. Estas no son las más modernas, ya que recientemente han sido actualizadas a Lollipop, y las guías de diseño cambiadas hacia el llamado “Material Design” (diseño material), pero se ha desestimado seguirlas dadas su novedad, que implica una nueva curva de aprendizaje y posible escasez de recursos de apoyo.

No obstante, la interfaz es sencilla e intuitiva, y resulta familiar para todo usuario del sistema operativo Android. Además se han incluido explicaciones y guías de usuario dentro de la aplicación, para mejorar su usabilidad y asegurar el correcto uso de la misma por parte del usuario.

En Android hay dos maneras de definir una interfaz gráfica: programáticamente o por archivos de recurso XML. Se ha optado por la segunda opción, la cual disminuye el acoplamiento en el código, aumenta la reusabilidad y la claridad del mismo, y por otra parte el entorno de desarrollo permite previsualizar el resultado de dichos archivos XML con bastante fidelidad.

2.2.3. Captura de audio

2.2.4. Calibración

Capítulo 3

Plan de pruebas y verificación

3.1. Calidad del software

3.2. Precisión de las medidas

Para cerciorarse de que los valores de nivel de presión sonora mostrados en la aplicación se ajustan a la realidad, es necesario comparar los valores para una misma fuente con los obtenidos por un aparato de medición calibrado.

Tras realizar dichas medidas, la aplicación deberá de ser configurada conforme a los resultados obtenidos, introduciendo los parámetros pertinentes en la pantalla dispuesta a dicho efecto.

Dicho proceso es necesario para cada micrófono distinto que sea usado con la aplicación, ya sea por usarla en un teléfono distinto o por utilizar un micrófono externo, ya que las características de cada uno varían, y no puede garantizarse la fidelidad de los resultados de un micrófono con los parámetros de otro.

Para las pruebas de calibrado del nivel de presión sonora, se han realizado en el laboratorio medidas de nivel de presión sonora emitido por un monitor de estudio, primero por un sonómetro calibrado, y después por la aplicación.

El sonómetro utilizado ha sido el Svantek SVAN 977.

El micrófono utilizado por la aplicación ha sido el integrado en el teléfono, modelo LG Nexus 5. Según [LG], es un micrófono del tipo microelectromecánico, o MEMS según sus siglas en inglés, de la marca Goertek. Sin embargo, la hoja de características del micrófono no está disponible.

3.3. Realización de los objetivos

Aparato	Sonómetro	Teléfono
Tono 440 Hz	68.7 dB	68.6 dB
Ruido ambiente	37.4 dB	43.2 dB
Ruido blanco	69.3 dB	62.6 dB
Ruido Rosa	69.2 dB	64.1 dB
Canción	63.1 dB	59.3 dB

Tabla 3.1: Tabla de comparacion de mediciones

Parte III

Parte tercera.

Conclusiones y líneas de trabajo futuras

Guillermo Orellana Ruiz
4 de enero de 2015

Parte IV

Apéndices

Apéndice A

Código del programa

Contenido

A.1 Primera sección	33
-------------------------------	----

A.1. Primera sección

Bibliografía

- [and14] Android developers guide, 2014. <http://developer.android.com/>.
- [CPCA13] Eduardo Casilari Pérez and José Antonio Cortés Arrabal. *Breves notas de estilo para la redacción de Proyectos Fin de Carrera y Trabajos Fin de Grado*. ETSIT, Universidad de Málaga, 2013. http://www.uma.es/media/files/Manual_de_Estilo_TFG_ETSIT.pdf.
- [Ent05] J.T. Entrambasaguas. *Ingeniería de Desarrollo de sistemas de Telecomunicación*, apuntes de clase. ETSIT-UMA, Curso 2004-2005.
- [ETS01] ETSIT. *Programación docente*. ETSIT-UMA, Curso 2000-2001.
- [ETS08] ETSI. MTS: Methods for Testing Specifications. <http://portal.etsi.org/mbs/Testing/testing.htm>, 2008.
- [Lam99] Leslie Lamport. *TEX: a Document Preparation System*. AddisonWesley Longman, Inc, 2nd edition, August 1999. It includes user's Guide and Reference manual. ISBN: 0-201-52983-1.
- [LG] LG. Service manual nexus 5 lg d821.
- [min14] Minted: a highlighted source code for latex. <https://code.google.com/p/minted/>, 2014.
- [Wik04] Wikipedia. Plagiarism — Wikipedia, the free encyclopedia, 2004. [Online; accessed 22-July-2004].

