



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Aplicación para la gestión del
sonido en un dispositivo móvil**



Presentado por Jorge Zamora Marqués
en Universidad de Burgos — 14 de febrero
de 2019

Tutor: Carlos López Nodal



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Carlos López Nodal, profesor del departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Jorge Zamora Marqués, con DNI 72895344-H, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Aplicación para la gestión del sonido en un dispositivo móvil.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 14 de febrero de 2019

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Carlos López Nodal

D. nombre co-tutor

Resumen

Se desarrollará una aplicación móvil que permitirá la configuración de unos perfiles de audio los cuales serán establecidos en nuestro dispositivo móvil por una serie de eventos configurados por el usuario, como eventos manuales, periódicos, wifi, GPS y de calendario.

Descriptores

Servicio, volumen, perfil, GPS, calendario, wifi, observador, concurrencia, Android . . .

Abstract

A mobile application will be developed that will allow the configuration of some audio profiles which will be established in our mobile device by a series of events configured by the user, such as manual, periodic, wifi, GPS and calendar events.

Keywords

Service, volume, profile, GPS, calendar, wifi, observer, concurrency, Android.

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	3
2.1. Objetivos técnicos del proyecto	3
2.2. Objetivos propios de la aplicación	3
Conceptos teóricos	5
3.1. Servicios Android - Services	5
3.2. Listener	6
3.3. Concurrencia	6
Técnicas y herramientas	7
4.1. Herramientas utilizadas	7
4.2. Técnicas utilizadas	8
Aspectos relevantes del desarrollo del proyecto	11
5.1. Inicio del proyecto	11
5.2. Metodologías	11
5.3. Herramientas para la construcción y versiones de Android . .	12
5.4. Aprendizaje e investigación	12
5.5. Decisiones propias durante el desarrollo de la aplicación . . .	13
5.6. Localización de la App	16

5.7. Pruebas	16
5.8. Control de la calidad de código	16
Trabajos relacionados	17
6.1. No molestar (Android)	17
6.2. Netroken Volume Control	17
6.3. Go Bees	17
6.4. Comparativa y opinión	18
Conclusiones y Líneas de trabajo futuras	19
7.1. Conclusiones	19
7.2. Nuevas líneas de trabajo	19
Bibliografía	23

Índice de figuras

Índice de tablas

4.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	9
---	---

Introducción

En los días en los que vivimos, mas de cinco mil millones de personas usan en su vida diaria un SmarthPhone, con lo que a más de una le ha pasado y le pasará, que su terminal suene en situaciones indebidas o que este no suene cuando se quería recibir cualquier tipo de aviso. Lo que quiero desarrollar en este Trabajo de Fin de Grado es una aplicación para SmartPhone, más concreta mente para los que usan el Sistema Operativo Android, para evitar en lo más posible, el que el terminal suene en momentos indebidos, y con ello ahorrarnos alguna que otra vergüenza.

El principal problema que encontramos, es que llevamos encima los terminales, ya sea en el trabajo, durante el estudio, durante las horas de dormir ect. Lo que quiero conseguir es posibilitar el configurar una serie de condiciones para que el terminal cambie solo de estado de volumen, los cuales añadiremos nosotros.

Para facilitar el uso a los usuarios, podremos utilizar los eventos de los calendarios del móvil, así como nuestra situación GPS, por las conexiones Wifi, y por eventos que añadamos manualmente dentro de la aplicación.

Los elementos que podremos configurar serán el volumen general del móvil, el volumen de la música, de la alarma, de las notificaciones y el volumen de llamada.

Objetivos del proyecto

Objetivo principal El primero objetivo que se busca es posibilitar la automatización del control del volumen de nuestro terminal móvil con una sencilla aplicación, y que con el mínimo esfuerzo el usuario pueda configurar distintas configuraciones de audio, para luego activarlas con unos eventos específicos, como eventos de calendario o la localización GPS.

2.1. Objetivos técnicos del proyecto

- Se estudiarán las diferentes librerías para el control de calendarios, GPS y wifi.
- Se estudiará el funcionamiento de las aplicaciones Android así como el ciclo de vida de estas.
- El estudio de diferentes formas de serializar datos en dispositivos móviles.
- El estudio de las diferentes versiones de Android, así como de las ventajas que nos ofrece cada una.
- La generación de documentación con Latex.

2.2. Objetivos propios de la aplicación

- Poder guardar configuraciones de sonido propias, para el uso de estas dentro de la aplicación.

- Poder generar eventos propios por el usuario, ya sean de forma periódica o para momentos concretos.
- Poder obtener la red wifi a la que el usuario está conectado con su dispositivo.
- Poder leer los eventos de los calendarios de un usuario.
- Poder obtener la posición GPS de nuestro dispositivo Android.
- Poder configurar ciertas particularidades de los eventos, como distancia de aproximación en el GPS o la lectura de la actual red wifi.
- Poder obtener una rápida ayuda desde la propia aplicación para utilización de esta.
- Poder saber cual es el perfil de sonido activo en cualquier momento.
- Universalizar los layouts de la aplicación.
- Internacionalización de la aplicación.
- Poder activar o desactivar un tipo determinado de eventos.
- Que la aplicación gestione de manera automática los eventos creados y cambiar automáticamente los perfiles de sonido.

Conceptos teóricos

A continuación se expondrán los principales conceptos teóricos, los cuáles son la piedra angular del funcionamiento del proyecto los cuales son los servicios, los listener y la propia concurrencia en si.

3.1. Servicios Android - Services

[Un Service es un componente de una aplicación que puede realizar operaciones de larga ejecución en segundo plano y que no proporciona una interfaz de usuario. Otro componente de la aplicación puede iniciar un servicio y continuará ejecutándose en segundo plano aunque el usuario cambie a otra aplicación. Además, un componente puede enlazarse con un servicio para interactuar con él e incluso realizar una comunicación entre procesos (IPC). Por ejemplo, un servicio puede manejar transacciones de red, reproducir música, realizar I/O de archivos o interactuar con un proveedor de contenido, todo en segundo plano.

Un servicio puede adoptar esencialmente dos formas:

Servicio iniciado Un servicio está iniciado cuando un componente de aplicación (como una actividad) lo inicia llamando a `startService()`. Una vez iniciado, un servicio puede ejecutarse en segundo plano de manera indefinida, incluso si se destruye el componente que lo inició. Por lo general, un servicio iniciado realiza una sola operación y no devuelve un resultado al emisor. Por ejemplo, puede descargar o cargar un archivo a través de la red. Cuando la operación está terminada, el servicio debe detenerse por sí mismo.

Servicio de enlace Un servicio es de enlace cuando un componente de la aplicación se vincula a él llamando a `bindService()`. Un servicio de enlace ofrece una interfaz cliente-servidor que permite que los componentes interactúen con

el servicio, envíen solicitudes, obtengan resultados e incluso lo hagan en distintos procesos con la comunicación entre procesos (IPC). Un servicio de enlace se ejecuta solamente mientras otro componente de aplicación está enlazado con él. Se pueden enlazar varios componentes con el servicio a la vez, pero cuando todos ellos se desenlazan, el servicio se destruye.] [7]

3.2. Listener

Los listener son clases que se encargan de controlar los eventos que se generan dentro de nuestra aplicación. Estos listeners podrán manejar los eventos para su posterior procesamiento, pudiendo capturar datos e información de dichos eventos.

3.3. Concurrencia

La concurrencia es una propiedad que permite al sistema el poder realizar trabajos de manera simultánea. En nuestro caso, la concurrencia se realizará a través de los diferentes hilos de ejecución de nuestra aplicación, como el hilo principal de la aplicación (Main Thread), nuestro observador GPS el cual mirará en todo momento si se hacen cambios en el GPS, y el hilo del observador genérico, el cual es un hilo que se lanza cada x tiempo.

El principal problema que se trató con el tema de la concurrencia fueron el bloqueo del hilo principal por medio del observador genérico, ya que al utilizar un Thread no había control sobre el bloqueo del hilo principal. Como alternativas se utilizaron los AsincThread, hilos asíncronos que nos permitían la ejecución de tareas en segundo plano, aunque al final se decidió la utilización del TimerTask, ya que está preparado para la repetición de acciones programadas con un lapso x de tiempo.

Técnicas y herramientas

4.1. Herramientas utilizadas

Las herramientas utilizadas durante el proceso del desarrollo del trabajo han sido, el IDE Android Studio para la generación de código de Java, utilizando las librerías de Android, y a su vez el desarrollo de las interfaces gráficas de nuestra aplicación. Para el control del código generado también utilizamos Códacy, un herramienta que nos califica el código de nuestra aplicación. Para gestionar las versiones de nuestra aplicación hemos utilizado Git-Hub y para complementarlo y gestionar la planificación del proyecto hemos utilizado Zen-Hub. Para realizar la documentación hemos utilizado T_EXMaker y PowerPoint.

Android Studio

Es un entorno de desarrollo para aplicaciones Android que fue desarrollado por Google en 2014. Utiliza software de IntelliJ IDEA y es gratuito gracias a la licencia de Apache 2.0. Las plataformas que soporta son Windows, macOS y Linux.

T_EXMaker

Es un editor de texto gratuito desarrollado por Pascal Brachet que permite el desarrollo de documentos con LaT_EX.

PowerPoint

Aplicación de Microsoft que nos permite realizar de manera sencilla presentaciones.

Codacy

Es una herramienta en la nube que nos realiza un estudio sobre la calidad de código que está en un repositorio Git, como GitHub. Cada vez que realicemos un commit el aplicativo se encargará de dar un diagnóstico de la calidad de nuestro código así como de indicarnos donde están los problemas de este.

GitHub + ZenHub

GitHub es una plataforma de desarrollo de aplicaciones. Como su propio nombre indica GitHub utiliza Git, que es un software de control de versiones. Complementado a GitHub encontramos a ZenHub, que es una extensión para Google Chrome que nos permite el gestionar proyectos con metodología Ágil.

4.2. Técnicas utilizadas

Las técnicas utilizadas a la hora de desarrollar el proyecto han sido dos, a la hora de planificar el desarrollo del proyecto hemos utilizado la metodología Ágil, y a la hora de realizar la programación hemos utilizado la metodología de programación orientada a objetos.

Metodología Ágil

La metodología ágil es una metodología de desarrollo software que se basa en el desarrollar los proyectos de una forma iterativa e incremental, donde los requisitos y sus soluciones van cambiando según las necesidades de ese momento.

Llamaremos iteración a la toma de decisiones y al trabajo que se realiza durante un periodo estipulado de tiempo. Dentro de cada iteración veremos una fase de planificación, de análisis de requisitos, diseño, codificación, pruebas y documentación.

En la fase de planificación lo que se planificará el trabajo que se tendrá que desarrollar en esa iteración. En la fase del análisis de requisitos se verá desarrollarán las funcionalidades anteriormente planificadas. En la fase de diseño se verá gestionará la forma de abordar dichos requisitos. Las partes de codificación, documentación y pruebas son más parte del equipo de trabajo del proyecto, y serán los encargados del desarrollo dentro del software de las

Herramientas	Aplicación	BD	Memoria
Java	X	X	
Librerías Android Java	X		
Android Studio	X	X	
Realm		X	
Git + Zen-Hub	X	X	X
MikTeX			X
PowerPoint			X
TeXMaker			X
Orientación a objetos	X		
Metodología Ágil del desarrollo	X	X	X

Tabla 4.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

especificaciones anteriores, teniendo que realizar también la documentación del código generado y de las pruebas asociadas a cada parte del código.[11]

Orientación a objetos.

La programación orientada a objetos es una metodología de programación que se basa en el tratamiento de objetos que son entidades que tienen un estado, y los cuales pueden realizar unos determinados comportamientos. Los objetos se descompondrán en atributos y métodos, siendo los atributos los encargados de especificar los estados de los objetos, y los métodos que son los encargados de especificar los comportamientos que pueden realizar dichos objetos.

Las principales características de la programación orientada a objetos son la capacidad para abstraer las características esenciales de los objetos, que todos los elementos pueden considerarse pertenecientes a una misma unidad, permite la herencia entre objetos, es decir que un objeto herede métodos y atributos de otra clase padre, permite la recolección de basura, las propiedades de los objetos están protegidas a modificaciones externas, y se permite la descomposición en partes más pequeñas el código generado.[13]

Aspectos relevantes del desarrollo del proyecto

A continuación se expondrán los aspectos más relevantes del desarrollo del proyecto.

5.1. Inicio del proyecto

Lo que me motivó a la hora de desarrollar este proyecto fue la necesidad de permitirme de una manera rápida y sencilla poder configurar mi terminal móvil para evitar situaciones embarazosas o no deseadas en las que el móvil suena cuando no debía, o por el contrario no sonaba cuando estabas disponible.

Para ello decidí realizar una aplicación que permitiera a un usuario no experto de los dispositivos móviles el poder mantener gestionado en todo momento y de manera sencilla el sonido de su dispositivo.

5.2. Metodologías

Dentro de lo posible se intentó llevar a cabo el desarrollo del proyecto siguiendo una metodología ágil, siendo esta Scrum, intentando ir sacando de manera ágil el contenido del proyecto.

Con mayor o menor fluidez, se han estado generando una serie de Sprints para el desarrollo del proyecto, más concreta mente 8, hasta la fecha.

De estos 8 Sprints, se puede llegar a ver la evolución del proyecto, ya que en los primeros Sprints se buscaba la adquisición de conocimientos, luego la

resolución de los casos de uso, para finalmente en los últimos Sprints pulir el contenido de la App así como añadir mejoras a esta.

5.3. Herramientas para la construcción y versiones de Android

Nada más empezar el desarrollo de la aplicación se especificó que esta sería una aplicación Android, desarrollada en el lenguaje Java, siendo Android 19 la versión mínima y Android 26 la máxima, aunque las pruebas realizadas hasta la fecha con con Android 24. Para el desarrollo de la aplicación se utilizaría Android Studio, para la documentación T_EXMaker para la edición de la documentación en L^AT_EX, y como repositorio utilizaríamos GitHub, que a su vez con Zen-Hub, nos serviría como herramienta de seguimiento del proyecto.

5.4. Aprendizaje e investigación

Para la realización del proyecto, tuvimos que adquirir nuevos conocimientos, ya que aunque en la carrera se trabaje con Java, no con Android y sus librerías, con lo cual tuvimos que tomar como referencia tres puntos de los que agarrarnos para la obtención de información de como funciona Android, y cuáles eran las mejores prácticas para realizar el trabajo.

Stack Overflow

Es un foro donde se encuentra gran cantidad de información sobre diversos lenguajes de programación.[8]

Udemy

Es una página web de cursos online, en donde yo realicé un curso para aprender a programar Android. En este caso realicé el curso titulado Programación de Android desde cero. [6]

Android Developers

Android Developers es la página de referencia para cualquier programador Android, ya que encontramos de una manera rápida y sencilla la información requerida.[1]

5.5. Decisiones propias durante el desarrollo de la aplicación

A continuación se verán las decisiones técnicas que se tuvieron que ir tomando para poder sacar el mejor producto posible.

Persistencia de los datos

Una decisión importante que se tuvo que tomar una vez ya arrancado el proyecto y avanzado, fue el modo de guardar los datos que introducía el usuario, es decir, cada vez que abriéramos la aplicación tendría que poder obtener los datos que habían sido guardados previamente, ya fueran eventos o configuraciones de sonido.

Las opciones que contemplamos fueron SQLite, una base de datos con un tamaño reducido, que se ejecuta con la propia aplicación y Realm, que es un sistema de gestión de bases de datos diseñado inicialmente para Android y IOS de código abierto. Al final se decidió la utilización de la base de datos Realm ya que su utilización era bastante sencilla.

La única complicación, es que hay que generar el id por el que va cada clase, para ello generaremos una clase que se ejecute al principio de cada ejecución de la aplicación, y que se encargara de obtener el ID máximo de cada clase, para evitar problemas de conflictos de PrimaryKey.

TimerTask

TimerTask fue la solución encontrada al problema de gestionar hilos asíncronos en nuestra aplicación, para que el hilo principal de ejecución no se bloqueara cada vez que se tuviera que comprobar el estado de los eventos. TimerTask es un hilo asíncrono el cual se ejecuta cada x tiempo, este tiempo es definido cuando se construye el hilo.

LocationManager

Otra decisión importante fue la forma de acceder a la posición GPS, pero finalmente se optó por Location Manager, ya que de manera sencilla nos permitía el poder obtener un servicio que nos controlara los cambios en la localización GPS. Location Manager es una clase propia de Android que nos permite utilizar los servicios de localización del SO. Con esta clase lo que podremos hacer será obtener la localización por varios métodos, en este caso la localización será la que obtengamos del GPS del móvil.

Easy Content Providers

Otra barrera importante fue el acceso a los calendarios de nuestro dispositivo Android, para lo cual se optó por la utilización del proyecto Easy Content Provider, el cual permite obtener de manera sencilla diferentes datos del dispositivo, como por ejemplo los calendarios, contactos, log de llamadas. . .

Con esta herramienta hemos conseguido iterar sobre los diferentes calendarios que se encuentran en nuestro dispositivo android, para luego obtener los eventos, y saber si hay alguno activo en ese momento o no.[12]

WifiManager

Para el acceso a las configuraciones Wifi, se optó por la utilización de WifiManager, que es la clase Java de Android, que nos permite obtener la información del wifi de nuestro terminal. Se estuvo pensando si utilizar toda la lista de redes wifi disponibles, la red con la intensidad mayor o la red wifi a la que estaba conectada el dispositivo, y al final me decanté por la red wifi a la que está conectado el dispositivo, ya que puede dar la casualidad que haya alguna otra red disponible con el mismo nombre que alguna de las que estuvieran configuradas, y de esta manera nos aseguramos que la red wifi con la que trataremos será la que utilice el usuario del terminal.[10]

Navigation Drawer

Una vez fuimos creando diversas pantallas, se planteó como poder acceder a ellas de una forma sencilla y rápida, con lo que se pensó que la utilización de un menú lateral desplegable sería la forma mas sencilla, rápida y limpia de poder acceder las diferentes pantallas sin demasiado esfuerzo.

Configuración de los eventos

Para permitir al usuario una mayor capacidad de interacción con la aplicación, se permitirá el ajuste de los eventos que se quieran tener activos en cada momento, es decir, se permitirá el borrado de los eventos previamente creados, y a su vez elegir que eventos quieren que se traten.

Permisos

Para poder utilizar ciertos tipos de sensores y servicios de los dispositivos móviles, es necesaria la aceptación de una serie de permisos, para que

nuestra aplicación funcione correctamente. En nuestro caso los permisos que se tendrán que conceder para poder utilizar toda la funcionalidad de la aplicación son los permisos de localización y los permisos de lectura / escritura del calendario del dispositivo.

Para pedir dichos permisos dentro de la aplicación se hace una comprobación de si estos han sido concedidos por el usuario, si estos no han sido concedidos, cuando se arranca la aplicación saldrá un mensaje pidiendo acceso a dichos elementos.

Dentro del desarrollo de la aplicación estos tendrán que ser declarados en el archivo Manifest de la aplicación a si como controlados dentro del código a la hora de ejecutarse.[3]

Notificaciones

Nuestra aplicación utilizará una serie de notificaciones para la comunicación recíproca entre la aplicación y el usuario que la está utilizando. Podremos diferenciar 3 tipos de notificaciones que se darán durante el uso de la aplicación.

Notificación del nivel del volumen

Es una notificación emergente en la parte superior del dispositivo que nos indica que niveles de volumen se han establecido en un momento determinado, cuando un evento configurado es activado.[2]

Notificación en la barra de estados del dispositivo

Igualmente que en las notificaciones anteriores, esta notificación será lanzada cuando un evento configurado sea activado por la aplicación y se realice un cambio de configuración. Esta notificación aparecerá en el menú desplegable superior de nuestro dispositivo, en la cual encontraremos el nombre de la configuración de sonido que ha sido activada.[5]

Notificación por mensaje emergente

Estas notificaciones serán lanzadas cuando vayamos a guardar eventos o configuraciones, estos pequeños mensajes nos darán la confirmación de si se ha guardado de manera correcta la información o no.[9]

5.6. Localización de la App

Una vez tuvo la aplicación una buena estructura y una buena base, se empezó a implementar la localización de la aplicación, eso quiere decir que la aplicación tendrá dos o varios idiomas. En este caso se implementó una primera versión de la aplicación en inglés.

5.7. Pruebas

Durante el desarrollo de la aplicación, se tuvieron que realizar una serie de pruebas para comprobar que el código generado realizaba las acciones esperadas correctamente, para ello se realizaron una serie de pruebas unitarias, que corroboraban el buen o mal funcionamiento de la aplicación. Una vez ya asentada la base de la aplicación se empezaron a desarrollar pruebas automáticas con la herramienta de Espresso.

5.8. Control de la calidad de código

Durante la última mitad de la fase de desarrollo de la aplicación se ha estado utilizando Codacy, una herramienta web que permite el control de la calidad de nuestro código, así como la duplicidad de código, la complejidad de este y que no haya código sin usar o mal estructurado.

Trabajos relacionados

6.1. No molestar (Android)

"No molestar."^{es} una configuración en los ajustes de los terminales Android que permite evitar la llegada de llamadas y notificaciones al dispositivo. Se podrán restringir números de teléfono y las diferentes notificaciones como eventos, llamadas y mensajes.^[4]

6.2. Netroken Volume Control

Es una aplicación que nos permite controlar los diferentes tipos de volúmenes dentro de nuestro dispositivo Android, por defecto ya hay unos creados y nosotros podremos crear más. Con la versión de pago podremos utilizar la localización de nuestro dispositivo para activar un perfil de sonido, a demás de programar cambios de perfil.^[15]

6.3. Go Bees

Go Bees es una aplicación móvil Android creada por el compañero David Miguel de nuestra universidad. La pongo como trabajo relacionado ya que es un buen ejemplo de como generar una buena aplicación Android con un seguimiento correcto del proyecto, documentación etc. La pongo como trabajo relacionado ya que lo utilizo como material de referencia para realizar este proyecto.^[14]

6.4. Comparativa y opinión

Comparando la app Netroken con la del proyecto, vemos que la idea es muy similar en ambas, aunque la aplicación de Netroken gane puntos ya que puede usarse como un gestor de volumen convencional, pierde también parte de su encanto al tener micro pagos para el desbloqueo de todas sus funcionalidades como la de la localización GPS para activar diferentes perfiles. Pensándolo fríamente, las dos aplicaciones se basan en la misma premisa, pero siguen sin ser lo mismo, ya que este TFG se fundamenta en el control automático de los perfiles de sonido, sin darnos la oportunidad de poder cambiar de perfil de sonido manualmente. Esto a su vez le puede quitar valor con respecto a la aplicación Netroken.

En comparación con la aplicación de no molestar de Android, se podría decir que No Molestar es similar en parte a la configuración de los eventos periódicos, pero con el añadido de que nos permite, no solo configurar el volumen, si no la interacción con otros elementos como el filtrado de las llamadas.

Conclusiones y Líneas de trabajo futuras

7.1. Conclusiones

Comentándolo desde la parte del desarrollo de la aplicación, Android me ha parecido una tecnología bastante interesante, y a su vez me ha resultado conocida, ya que al trabajar con xml para el desarrollo de interfaces gráficas y la utilización de Java como lenguaje de programación, me ha resultado más sencillo de aprender.

Por otra parte los objetivos que tenía con este proyecto los he cumplido satisfactoriamente, ya que he conseguido crear una aplicación Android, que automatice el cambio de volumen del móvil con los eventos anteriormente citados.

Como opinión sobre la documentación de Android, creo que está bastante bien, aunque sin duda alguna, lo que me ha ayudado en gran medida a desarrollar el proyecto ha sido un curso de Android que he estado realizando como preparación.

Como pega al desarrollo de aplicaciones en Android, no solo necesitarás que en tu emulador funcione, si no que para cada versión y dispositivo. Con lo cual hay que estar siempre actualizando y mejorando tus aplicaciones, o si no se quedarán obsoletas.

7.2. Nuevas líneas de trabajo

A continuación hablaremos de posibles y propuestas a tener en cuenta para seguir con la evolución de este proyecto y de su coste en tiempo y del

valor añadido que daría dicha acción.

Servidor Web

Para conseguir una mayor eficacia a la hora de gestionar las configuraciones de sonido de una persona, podríamos tener un servidor web, el cual se encargara de almacenar los datos, permitiéndonos conectarnos con una cuenta. Coste: 80 - 100. Valor añadido: 50 -100;

Desarrollo para IOS

Se podría desarrollar la aplicación para que se pueda integrar a los sistemas Apple. Coste: 100 - 100. Valor añadido: 100 - 100;

Testeo de la aplicación en diferentes dispositivos y versiones

Para asegurar un correcto funcionamiento de la aplicación, en un futuro se deberán de realizar más pruebas automáticas para luego estas ser probadas en servicios de virtualización de dispositivos, a si de esta manera podremos encontrar posibles errores de funcionamiento. Coste 50 - 100. Valor añadido 80 - 100.

Publicación de la App

Se podría publicar la App en el Play Store para que cualquier persona pudiera disfrutar de esta aplicación. Coste: 10 - 100. Valor añadido: 100 - 100.

Creación de un nuevo evento de "caminando - corriendo"

Se podría implementar, mediante la utilización del giroscopio del dispositivo, un evento que fuera lanzado cuando el usuario está caminando o corriendo con el dispositivo y habiendo transcurrido un tiempo x desde el inicio del desplazamiento. Coste: 60 - 100. Valor añadido: 40 - 100.

Añadir control sobre la vibración

Normalmente el control del volumen siempre va asociado al control de la vibración, con lo que se podría añadir el control de esta en nuestros perfiles de sonido. Coste: 10 - 100. Valor añadido: 60 - 100.

Bibliografía

- [1] Android Developers.
- [2] AudioManager.
- [3] Cómo solicitar permisos durante el tiempo de ejecución.
- [4] Limitar las interrupciones con el modo No molestar en Android - Ayuda de Android.
- [5] Notifications Overview.
- [6] Programación de Android desde Cero +35 horas Curso COMPLETO.
- [7] Servicios.
- [8] Stack Overflow - Where Developers Learn, Share, & Build Careers.
- [9] Toasts overview.
- [10] WifiManager | Android Developers.
- [11] Desarrollo ágil de software, July 2018. Page Version ID: 109609657.
- [12] Easy integration with Android's built-in and custom content providers data: EverythingMe/easy-content-providers, February 2019. original-date: 2015-08-06T12:26:57Z.
- [13] Programación orientada a objetos, February 2019. Page Version ID: 113748672.
- [14] David Miguel Lozano. Android app for monitoring bee hives activity. Contribute to davidmigloz/go-bees development by creating an account on GitHub, November 2018.

- [15] Uptodown Technologies SL. Netroken Volume Control 4.52 para Android - Descargar.