UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

título del TFG

Presentado por Nombre del alumno
en Universidad de Burgos — 11 de febrero de 2019
Tutor: nombre tutor

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Nombre del alumno, con DNI dni, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

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, 11 de febrero de 2019

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. nombre tutor	D. nombre co-tutor

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice de figuras

Índice de tablas

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. 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 con los que podremos interactuar serán el volumen general del móvil, el volumen de la música y de la alarma, así como configurar si queremos vibración o no.

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 particualridades de los eventos, como distancia de aproximación en el GPS, lectura de la actual red wifi o poder elegir si queremos los eventos de calendario que su búsqueda sea exacta o que contenga una palabra clave.
- 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.

Conceptos teóricos

3.1. Aplicación móvil

Una aplicación móvil es es una aplicación informática con el objetivo de ser ejecutada dentro de dispositivos móviles como un smartphone o una tablet.

3.2. Android

A continuación hablaremos de como funciona el Sistema Operativo de Android, el cual está basado en el núcleo de Linux y desarrollado para su ejecución en dispositivos móviles como SmartPones, tablets, SmartWatchs... La última versión de este Sistema Operativo es la *9.0 "Pie"*.

Arquitectura de Android

La arquitectura de Andorid está estructurado en cinco niveles, siendo estos las aplicaciones, los marcos de trabajo de las aplicaciones, las bilbiotecas, el runtime de Android y el núcleo de Linux.

Aplicaciones: las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

Marco de trabajo de aplicaciones: los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad

del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario. Bibliotecas: Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.

Runtime de Android: Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecutaba hasta la versión 5.0 archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx". Desde la versión 5.0 utiliza el ART, que compila totalmente al momento de instalación de la aplicación.

Núcleo Linux: Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.[**?**]

3.3. Entorno de desarrollo

El entorno de desarrollo integrado es una aplicación informática que permite el desarrollo de software de una manera más sencilla. Los entornos de desarrollo suelensonstar de un editor de exto, herramientas de construcción automáticas y un debuger. Algunos entornos cuentan con IntelliSense para auto completar texto y un compilador, para poder compilar el código generado con el entorno de desarrollo.

3.4. Java

Java es un lenguaje de programación orientado a objetos, concurrente y de propósito general, y buscaba la independencia de la implementación de las aplicaciones, para conseguir poder ejecutar un mismo código sobre diferentes dispositivos con tecnologías diferentes.

Que sea un lenguaje de programación orientado a objetos quiere decir que cada tipo de dato que tengamos en la aplicación tendrá que llevar sus operaciones específicas para ese dato, es decir, cada objeto tendrá asociado un comportamiento y un estado. Al separar estos comportamientos y estados unos de otros, permite de manera más sencilla el reutilizar código de diferentes aplicaciones.

Para permitir la independencia del código sobre las plataformas Java al compilarse genera un código llamado bytecode que son instrucciones que se le darán a la máquina que corre Java. Es decir, que cuando instalemos Java tendremos que seleccionar el que sea compatible con nuestro dispositivo, pero a la hora de ejecutar código, al este correr por la máquina virtual de Java lo que permite es que no necesite de especificaciones concretas de cada sistema el código de nuestra aplicación.

Otra característica principal de Java es el recolector de basura, que impedirá tener objetos que no se estén utilizando en este momento. Para ello el programador podrá especificar el ciclo de vida de los objetos, para controlar cuando se crean y cuando se borran. Es muy importante el recolector de basura ya que si no podríamos tener problemas con la memoria de nuestro dispositivo, ya que cada instancia de un objeto consume parte de esta.

Los entornos en los que funciona Java son sistemas móviles y embebidos, navegadores web, servidores y aplicaciones de escritorio.[**?**]

3.5. 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 ïniciadoçuando 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 "de enlaceçuando 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.

3.6. 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 procesado, pudiendo capturar datos e información de dichos eventos.

GPS

GPS son las siglas de sistema de posicionamiento global, el cual nos permite establecer la posición de nuestro dispositivo móvil. Se generará un listener para los eventos GPS lo que se nos permitirá saber de manera periódica donde se encuentra nuestro dispositivo móvil.

3.7. 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 nohabía control sobre el bloqueo del hilo principal. Como alternativas se utilizaron los AsincThread, hilis 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 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 TEXMaker.

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.

TEXMaker

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

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 Chroome 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.

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.

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

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 memento.

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 hará será planificar el trabajo que se tendrá que desarrollar en esa iteración. En la fase del análisis de requisitos se verá con más profundidad la funcionalidad que se le quiere dar a lo anteriormente planificado. En la fase de diseño se verá la manera en la que se quiere realizar el trabajo. 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 especificaciones anteriores, teniendo que realizar también la documentación del código generado y de las pruebas asociadas a cada parte del código.

Aspectos 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. Aunque no ha sido conseguido, como forma educativa se ha intentado seguir lo máximo posible esta idea.

Para ello se generaron una serie de hitos a ir cumpliendo divididos en diferentes sprints, que se buscaba que fuera el contenido de trabajo que se podía generar en una semana. Se debería de haber ido quedando con el tutor para la revisión de estos sprints pero por diferentes motivos se fueron fui haciendo revisiones auto críticas para la revisión de estos a la vez que se quedaba con el tutor y se mantenía conversación por correo electrónico.

Cara la planificación de estos sprints se utilizó ZenHub, una herramienta que representa un tablero canvas.

Persistencia de los datos

Otra 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 fuero SQLlite, 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.

LocationManager

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

Es un proyecto de Android que 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.

WifiManager

WifiManager 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.

Navigation Drawer

Para facilitar la experiencia de usuario se pensó en la utilización de este tipo de menú, ya que es muy cómodo de utilizar y no ocupa espacio dentro de la interfaz de trabajo del usuario.

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.

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.

Notificación en la barra de estados del dispositivo

Igualmente que en las notificaciones anteriores, esta notificación será lanzada cunado 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.

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.

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.

6.2. App Volume Control

App Volume Control es una aplicación Android la cual nos permite configurar diferentes configuraciones de audio, para luego utilizarlas según que aplicación hayamos abierto. Es una aplicación bastante útil pero muy tediosa de configurar.

6.3. Go Bees

Go Bees es una aplicación móvil Android creada por el compañero David Miguel de nuestra universidad. La pongo como trabajo relaccionado 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.

6.4. Comparativa

Comparando las funcionalidades vistas del App Volume Control y el no molestar de Android, vemos que una todas ellas trabajan bajo una misma premisa, la configuración del volumen de nuestro dispositivo, pero creo que como idea más concreta nuestra aplicación podría llegar a tener una mayor aceptación, ya que es muy versátil y muy sencilla de utilizar.

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 permitido 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.

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.

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.

Conectarse al servidor de fichado del trabajo

Se podría desarrollar la aplicación para que se conectara al servidor donde se guardan los datos de fichado de los trabajadores, para que cuando entren al trabajo cambien de volumen, y cuando salgan se restablezca la configuración.

Desarrollo para IOS

Se podría desarrollar la aplicación para que se pueda integrar a los sistemas Apple.

Desarrollo con diferente paleta de colores, modo noche, daltónicos etc

Se podría permitir elegir diferentes temas para visualizar la aplicación.

Se realizará una gráfica con relación tiempo de desarrollo y valor aportado a la aplicación de las posibles mejoras y una descripción de cada una de estas, así se podrá ver cuales serían las mejoras a desarrollar con según que previsiones.