



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  
Documentación Técnica**



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

Tutor: Carlos López Nodal



---

# Índice general

---

<b>Índice general</b>	<b>I</b>
<b>Índice de figuras</b>	<b>III</b>
<b>Índice de tablas</b>	<b>VI</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	2
<b>Apéndice B Especificación de Requisitos</b>	<b>7</b>
B.1. Introducción . . . . .	7
B.2. Objetivos generales . . . . .	7
B.3. Catalogo de requisitos . . . . .	7
B.4. Especificación de requisitos . . . . .	8
<b>Apéndice C Especificación de diseño</b>	<b>15</b>
C.1. Introducción . . . . .	15
C.2. Diseño de datos . . . . .	15
C.3. Diseño procedimental . . . . .	17
C.4. Diseño arquitectónico . . . . .	21
<b>Apéndice D Documentación técnica de programación</b>	<b>31</b>
D.1. Introducción . . . . .	31
D.2. Estructura de directorios . . . . .	31
D.3. Manual del programador . . . . .	32

D.4. Compilación, instalación y ejecución del proyecto . . . . .	33
D.5. Pruebas del sistema . . . . .	34
<b>Apéndice E Documentación de usuario</b>	<b>39</b>
E.1. Introducción . . . . .	39
E.2. Requisitos de usuarios . . . . .	39
E.3. Instalación . . . . .	39
E.4. Manual del usuario . . . . .	40
E.5. Manual del usuario dentro de la aplicación . . . . .	64
<b>Bibliografía</b>	<b>73</b>

---

## Índice de figuras

---

B.1. Imagen que muestra el diagrama de casos de uso del proyecto. . .	9
C.1. Imagen que muestra la relación entre las entidades en la base de datos. . . . .	16
C.2. Imagen que muestra la relación entre clases en la aplicación. . .	16
C.3. Imagen que muestra el diagrama de secuencia a la hora de generar un perfil de sonido. . . . .	17
C.4. Imagen que muestra el diagrama de secuencia a la hora de generar un evento GPS. . . . .	17
C.5. Imagen que muestra el diagrama de secuencia a la hora de generar un evento manual. . . . .	18
C.6. Imagen que muestra el diagrama de secuencia a la hora de generar un evento periódico. . . . .	18
C.7. Imagen que muestra el diagrama de secuencia a la hora de generar un evento de calendario. . . . .	19
C.8. Imagen que muestra el diagrama de secuencia a la hora de generar un evento wifi. . . . .	19
C.9. Imagen que muestra el diagrama de secuencia para el borrado de eventos. . . . .	20
C.10. Imagen que muestra el diagrama de secuencia de como funciona el hilo que controla los eventos manuales, periódicos, wifi y calendario. .	20
C.11. Imagen que muestra el funcionamiento del listener GPS. . . . .	21
C.12. Imagen que muestra la arquitectura MVC. . . . .	22
C.13. Imagen que muestra la distribución de los directorios dentro de nuestra aplicación. . . . .	23
C.14. Imagen que muestra la distribución de clases en el directorio activities. . . . .	24

C.15.Imagen que muestra la distribución de clases en el directorio app.	25
C.16.Imagen que muestra la distribución de clases en el directorio observergps. . . . .	26
C.17.Imagen que muestra la distribución de clases en el directorio observadorgenerico. . . . .	26
C.18.Imagen que muestra la distribución de clases en el directorio modificadorsonido. . . . .	27
C.19.Imagen que muestra la distribución de clases en el directorio helpfragments. . . . .	28
C.20.Imagen que muestra la distribución de clases en el directorio fragments. . . . .	29
C.21.Imagen que muestra la distribución de clases en el directorio bd.model. . . . .	30
E.1. Imagen que muestra el menú de la aplicación. . . . .	42
E.2. Imagen que muestra el mensaje en el cual aceptaremos o denega- remos los permisos para permitir que la aplicación pueda acceder a la ubicación del dispositivo. . . . .	44
E.3. Imagen que muestra el mensaje en el cual aceptaremos o denega- remos los permisos para permitir que la aplicación acceda al calendario de nuestro dispositivo. . . . .	45
E.4. Imagen que muestra la pantalla principal de la aplicación. . . .	47
E.5. Imagen que muestra la pantalla para la creación de perfiles de sonido. . . . .	49
E.6. Imagen que muestra la pantalla para la creación de eventos periódicos. . . . .	51
E.7. Imagen que muestra la pantalla para la creación de los eventos manuales. . . . .	53
E.8. Imagen que nos muestra la pantalla para la creación de los eventos de GPS. . . . .	55
E.9. Imagen que nos muestra la pantalla para la creación de eventos wifi. . . . .	57
E.10.Imagen que muestra la pantalla para la creación de los eventos de calendario. . . . .	59
E.11.Imagen en la que veremos la pantalla que nos permite el activar o desactivar un tipo de evento en concreto . . . . .	61
E.12.Imagen que muestra la pantalla en la cual podremos ver los eventos que están creados y eliminarlos de la aplicación. . . . .	63
E.13.Imagen que muestra la pantalla para la creación de eventos manuales. . . . .	65

E.14.Imagen que muestra la pantalla para la creación de eventos periódicos. . . . .	66
E.15.Imagen que muestra la pantalla para la creación de eventos wifi.	67
E.16.Imagen que muestra la pantalla para la creación de eventos de calendario. . . . .	68
E.17.Imagen que muestra la pantalla para la creación de eventos gps.	69
E.18.Imagen que muestra la pantalla para la creación de perfiles de sonido. . . . .	70
E.19.Imagen que muestra la pantalla para el acceso a las diferentes redes sociales y poder establecer contacto para obtener soporte de la aplicación. . . . .	71

---

## Índice de tablas

---

B.1. Crear nuevo perfil de audio . . . . .	10
B.2. Crear nuevo evento periódico . . . . .	11
B.3. Crear nuevo evento manual . . . . .	12
B.4. Crear nuevo evento wifi . . . . .	12
B.5. Crear nuevo evento calendario . . . . .	13
B.6. Crear nuevo evento GPS . . . . .	13
B.7. Configurar app . . . . .	14
B.8. Borrar evento . . . . .	14



## *Apéndice A*

---

# Plan de Proyecto Software

---

### A.1. Introducción

### A.2. Planificación temporal

A continuación se expondrán de manera breve las 8 fases en las que se dividió el proyecto, las cuales se encuentran diferenciadas en las diferentes millstones que encontramos en el repositorio de GitHub del proyecto.

- 1.- Se estudiarán los diferentes componentes de Android. 2.- Contemplar diferentes arquitecturas de Android. 3.- Realizar prototipos con las diferentes arquitecturas. 4.- Documentar la información recogida. <https://github.com/JorgeZamora94/TFG-Gestor-de-sonido-para-dispositivos-Android/milestone/1>
- Documentación sobre la correcta utilización de la plataforma GitHub: Utilización de ramas para gestionar de manera correcta los cambios dentro de nuestro proyecto. Añadir comentarios a las Issues para ver más clara la evolución del proyecto. Generar primera versión del servicio que se encargará de controlar la llegada de eventos GPS a nuestra aplicación, teniéndose que cambiar la tasa de refresco de los datos según la cantidad de cambios que se han generado. Documentar los progresos <https://github.com/JorgeZamora94/TFG-Gestor-de-sonido-para-dispositivos-Android/milestone/2>
- Se creará el patrón observer para los eventos wifi. Se generará documentación: Trabajos relacionados con nuestro proyecto. Se investigarán las diferentes formas de generar pruebas. Se amplió la fecha

de entrega para la realización de pruebas y la documentación de estas. <https://github.com/JorgeZamora94/TFG-Gestor-de-sonido-para-dispositivos-Android/milestone/3>

- Generar las clases necesarias para desarrollar el patrón observer para los eventos calendario, manuales y periódicos. Documentarlo. Búsqueda de información para revisar la calidad de código. <https://github.com/JorgeZamora94/TFG-Gestor-de-sonido-para-dispositivos-Android/milestone/4>
- Añadir serialización de los datos a la aplicación y generar primera versión de la memoria, a su vez que se prepara una demo con todo el contenido desarrollado. <https://github.com/JorgeZamora94/TFG-Gestor-de-sonido-para-dispositivos-Android/milestone/5>
- Se generará el modelo de datos para almacenar las configuraciones de sonido. Se generará el controlador que se encargue de gestionar el cambio de sonido. Se añadirá la interfaz que se encargue de recibir los datos del usuario para crear configuraciones de sonido. Se añadirán una serie de configuraciones de sonido por defecto cuando se instale por primera vez la aplicación. <https://github.com/JorgeZamora94/TFG-Gestor-de-sonido-para-dispositivos-Android/milestone/6>
- Se generará la documentación faltan te, se añadirá una primera versión de internacionalización en la ayuda, se generará una pantalla de configuraciones del usuario y corrección de bugs y mejoras visuales. <https://github.com/JorgeZamora94/TFG-Gestor-de-sonido-para-dispositivos-Android/milestone/7>
- Se mejorará la experiencia de usuario, arreglos menores y añadir mejoras sustanciales a la documentación de la aplicación, cuando se tenga la versión final de la app se añadirán las versiones finales de los diagramas, imágenes etc. <https://github.com/JorgeZamora94/TFG-Gestor-de-sonido-para-dispositivos-Android/milestone/8>

### A.3. Estudio de viabilidad

A continuación se realizará es estudio de viabilidad económica y legal del proyecto realizado.

## Viabilidad económica

En este punto contemplaremos la viabilidad del proyecto desde el punto económico.

### Costes de personal

Contemos con que el proyecto tiene una duración de 5 meses, en los cuales el programador tendrá un sueldo de 1100 euros.

Cálculos para el presupuesto:

- Se calcula un presupuesto de 1100 euros mensuales netos
- Con todos los gastos asociados (SS, formación, desempleo...) que contribuyen 37.4 por ciento del sueldo bruto nos dará un total de 657.19 euros de gastos
- $1100 + 657.19 = 1757.19$  euros a pagar mensualmente la empresa

Por parte de la empresa al trabajador, la empresa pagará  $1757.19 * 5 = 8785.95$  euros

### Costes hardware

El proyecto se podría realizar con cualquier ordenador con una potencia significativa. Podríamos hacer una estipulación de unos 1.200 euros de coste hardware para realizar el proyecto.

$\text{Amortización} = \text{Coste} / \text{Vida útil} \times \text{Tiempo utilizado}$

$\text{Amortización} = 1200 \text{ euros} / 48 \text{ meses} = 25 \text{ euros}$

El total coste hardware es de 1200 euros +  $25 * 5$  euros con un total de 1325 euros.

### Costes software

Las herramientas que han sido utilizadas para el desarrollo de este proyecto han sido gratuitas, así que se podría decir que el coste software es 0, teniendo en cuenta que el S.O. del ordenador con el que se desarrolle ya está incluido dentro del presupuesto hardware.

### Costes derivados

Los costes derivados son costes dentro del proyecto que se pagan indirectamente a la hora de realizar el trabajo, como por ejemplo la luz y el Internet. El gasto de Internet será de unos 35 euros mensuales. El gasto de luz será de unos 5 euros mensuales.

Costes derivados: 40 euros/mes x 5 meses = 200 euros

### Coste total del proyecto

El coste total del proyecto será de:

- 8785.95 euros por parte de pagos de la empresa.
- 1325 euros por parte del hardware.
- 200 euros por parte de gastos adicionales.

**El total de gastos por parte del desarrollo del proyecto serán de 10310.95 euros**

### Viabilidad legal

Mejorar, como enfocar este punto.

### Código

En la actualidad, casi la totalidad del software utilizado es de creación propia, menos una librería importada a través de gradle para su uso, los iconos de la aplicación y un método que contiene el algoritmo para el cálculo de las distancias entre dos puntos sacado de la página de Stack Overflow.

Para el uso de estos contenidos dentro de nuestra aplicación se han tenido que poner sus respectivas referencias dentro de nuestra aplicación (Readme).

También para evitar conflictos de licencias, se ha utilizado la misma licencia que la que utiliza el proveedor de calendarios, la licencia de Apache, en concreto la 2.0.

### **Documentación**

Con respecto a la documentación en principio toda ella entera es de autoría propia así como la documentación, anexos, imágenes, vídeos y presentación Power Point. Toda esta documentación utilizará una licencia Creative Commons.

Para incluir referencias de terceros dentro de nuestra documentación utilizaremos la bibliografía de esta, donde encontraremos la información de dichas fuentes.



## *Apéndice B*

---

# **Especificación de Requisitos**

---

### **B.1. Introducción**

En este apartado hablaremos de cuáles son las capacidades del usuario a la hora de trabajar con nuestra aplicación.

### **B.2. Objetivos generales**

De manera general los objetivos de la aplicación son posibilitar la gestión del volumen del terminal de una manera sencilla programando eventos manuales, periódicos, por localización GPS y Wifi, y mediante la lectura de eventos de calendario.

### **B.3. Catalogo de requisitos**

#### **Requisitos funcionales**

1. El usuario podrá guardar perfiles de audio.
2. El usuario podrá guardar eventos periódicos introduciendo el nombre del evento, seleccionando el día de la semana, seleccionando la hora de inicio y la hora de fin del evento, y el perfil de audio asociada.
3. El usuario podrá establecer eventos manuales introduciendo el nombre del evento, seleccionando el día del evento, la hora de inicio y de fin, y seleccionando el perfil de sonido deseada.

4. El usuario podrá establecer eventos GPS introduciendo el nombre del evento, introduciendo la distancia de reconocimiento, seleccionando el perfil de audio deseada, la localización GPS actual será la que se almacene como punto de control.
5. El usuario podrá establecer eventos Wifi introduciendo el nombre del evento, introduciendo el nombre de la red Wifi deseada, y seleccionando un perfil de audio.
6. El usuario podrá establecer eventos de calendario introduciendo el nombre del evento, introduciendo el título del evento del calendario, y seleccionando el perfil de audio.
7. El usuario podrá borrar cualquier evento.
8. El usuario podrá activar o desactivar los diferentes tipos de eventos.

### **Requisitos no funcionales**

1. La aplicación será de utilización sencilla.
2. Los menús serán intuitivos.
3. Encontraremos información básica de funcionamiento dentro de la misma aplicación.
4. Estará disponible una versión en inglés y otra en castellano.
5. Se podrá contactar con el soporte de la aplicación desde esta.

## **B.4. Especificación de requisitos**

### **Diagrama de casos de uso**



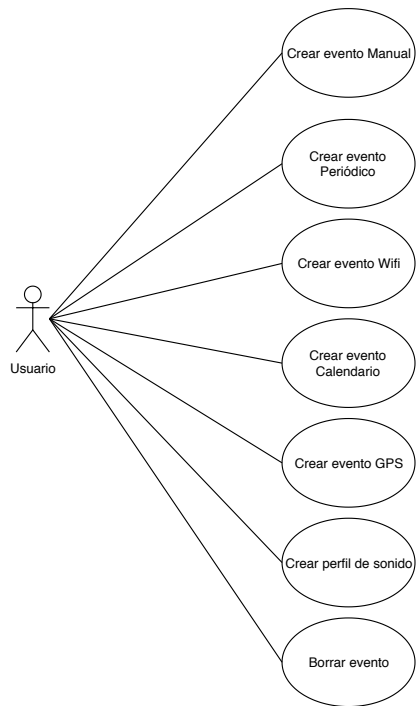


Figura B.1: Imagen que muestra el diagrama de casos de uso del proyecto.

## Casos de uso

A continuación se mostrarán los casos de uso que han sido desarrollados en la aplicación.

Información	Descripción
Versión	V1.0
Autores	Jorge Zamora Marqués
Requisitos asociados	
Descripción	Se creará un nuevo perfil de audio.
Pre condición	
Secuencia normal	Selecciona en el menú la opción de perfiles de sonido.
	Introduce nombre del perfil.
	Selecciona volumen del sistema.
	Selecciona volumen de la alarma.
	Selecciona volumen de la música.
	Pincha botón de guardado.
	Se guardan los datos en la base de datos.
Pos condición	Perfil de audio guardado.
Frecuencia esperada	Media
Importancia	Alta

Tabla B.1: Crear nuevo perfil de audio

Información	Descripción
Versión	V1.0
Autores	Jorge Zamora Marqués
Requisitos asociados	
Descripción	Se creará un nuevo evento periódico.
Pre condición	Haber perfiles de audio previamente guardados
Secuencia normal	Selecciona en el menú la opción de eventos periódicos.
	Introduce nombre del evento
	Selecciona hora de inicio
	Selecciona hora de finalización.
	Selecciona perfil de audio.
	Pincha botón de guardado.
	Se guardan los datos en la base de datos.
Pos condición	Evento periódico guardado.
Frecuencia esperada	Media
Importancia	Alta

Tabla B.2: Crear nuevo evento periódico

Información	Descripción
Versión	V1.0
Autores	Jorge Zamora Marqués
Requisitos asociados	
Descripción	Se creará un nuevo evento manual.
Pre condición	Haber perfiles de audio previamente guardados
Secuencia normal	Selecciona en el menú la opción de eventos manuales.
	Introduce nombre del evento
	Selecciona día
	Selecciona hora de inicio
	Selecciona hora de finalización.
	Selecciona perfil de audio.
	Pincha botón de guardado.
	Se guardan los datos en la base de datos.
Pos condición	Evento manual guardado.
Frecuencia esperada	Media
Importancia	Alta

Tabla B.3: Crear nuevo evento manual

Información	Descripción
Versión	V1.0
Autores	Jorge Zamora Marqués
Requisitos asociados	
Descripción	Se creará un nuevo evento wifi.
Pre condición	Haber perfiles de audio previamente guardados
Secuencia normal	Selecciona en el menú la opción de eventos wifi.
	Introduce nombre del evento
	Introducimos nombre del wifi.
	Selecciona perfil de audio.
	Pincha botón de guardado.
	Se guardan los datos en la base de datos.
Pos condición	Evento wifi guardado.
Frecuencia esperada	Media
Importancia	Alta

Tabla B.4: Crear nuevo evento wifi

Información	Descripción
Versión	V1.0
Autores	Jorge Zamora Marqués
Requisitos asociados	
Descripción	Se creará un nuevo evento calendario.
Pre condición	Haber perfiles de audio previamente guardados
Secuencia normal	Selecciona en el menú la opción de eventos de calendario.
	Introduce nombre del evento
	Introducimos nombre del evento del calendario.
	Selecciona perfil de audio.
	Pincha botón de guardado.
	Se guardan los datos en la base de datos.
Pos condición	Evento calendario guardado.
Frecuencia esperada	Media
Importancia	Alta

Tabla B.5: Crear nuevo evento calendario

Información	Descripción
Versión	V1.0
Autores	Jorge Zamora Marqués
Requisitos asociados	
Descripción	Se creará un nuevo evento GPS.
Pre condición	Haber perfiles de audio previamente guardados
Secuencia normal	Selecciona en el menú la opción de eventos GPS.
	Introduce nombre del evento
	Introduce la distancia de reconocimiento.
	Selecciona perfil de audio.
	Pincha botón de guardado.
	Se guardan los datos en la base de datos.
Pos condición	Evento GPS guardado.
Frecuencia esperada	Media
Importancia	Alta

Tabla B.6: Crear nuevo evento GPS

Información	Descripción
Versión	V1.0
Autores	Jorge Zamora Marqués
Requisitos asociados	
Descripción	Se cambiará la configuración de los eventos que tratará la app.
Pre condición	
Secuencia normal	Selecciona en el menú la opción de configuración de la app. Seleccionamos si queremos tener o no activo cada evento Pincha botón de guardado. Se guardan los datos en la base de datos.
Pos condición	Configuración guardada.
Frecuencia esperada	Baja
Importancia	Alta

Tabla B.7: Configurar app

Información	Descripción
Versión	V1.0
Autores	Jorge Zamora Marqués
Requisitos asociados	
Descripción	Se podrá borrar cualquier evento creado en la aplicación.
Pre condición	Tendrá que haber eventos creados.
Secuencia normal	Selecciona en el menú la opción del listado de eventos. Seleccionamos el evento que queramos borrar. Pincha botón de borrado. Se borrará el evento de la base de datos.
Pos condición	Evento borrado.
Frecuencia esperada	Media
Importancia	Alta

Tabla B.8: Borrar evento

## *Apéndice C*

---

# Especificación de diseño

---

### C.1. Introducción

En este apartado veremos cómo hemos estructurado el código y a su vez las estructuras utilizadas para el funcionamiento del proyecto.

### C.2. Diseño de datos

Los datos con los que hemos estado trabajando han sido objetos, con lo cual los principales objetos estructurales con los que hemos estado trabajando han sido:

- Configuración de sonido.
- Evento manual.
- Evento periódico.
- Evento GPS.
- Evento de calendario.
- Evento Wifi.
- Configuración de la app

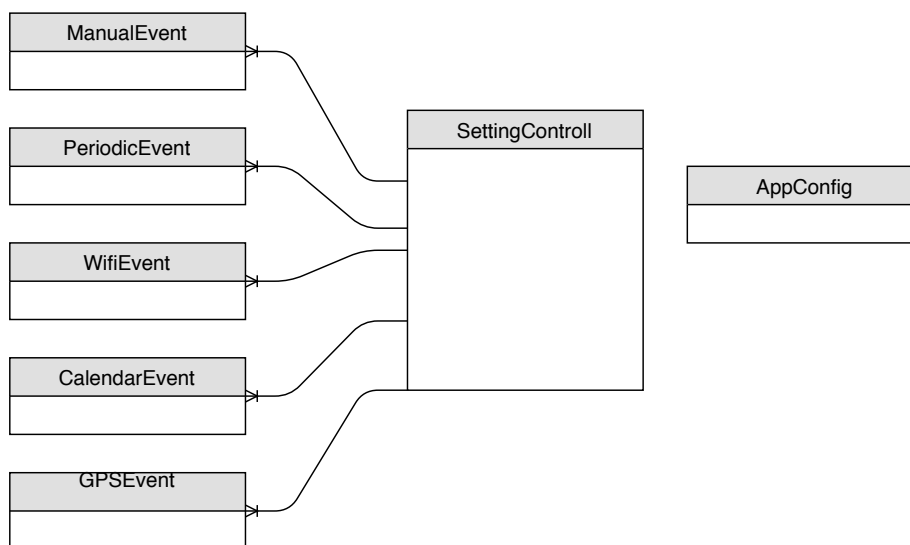


Figura C.1: Imagen que muestra la relación entre las entidades en la base de datos.

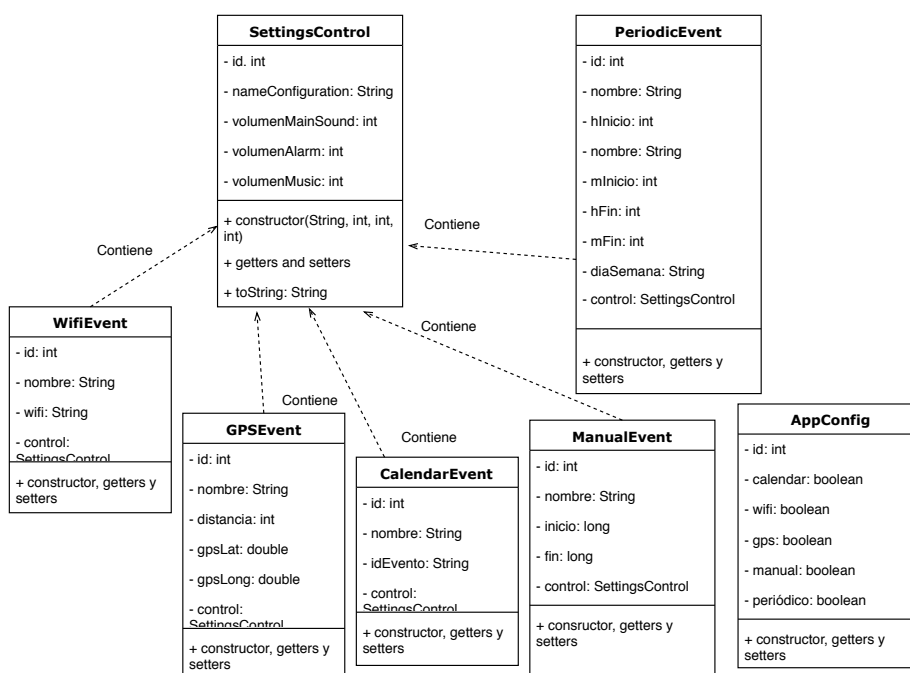


Figura C.2: Imagen que muestra la relación entre clases en la aplicación.



## C.3. Diseño procedimental

A continuación se añadirán los diferentes diagramas de secuencia, representando las diferentes acciones que se dan dentro de la aplicación.

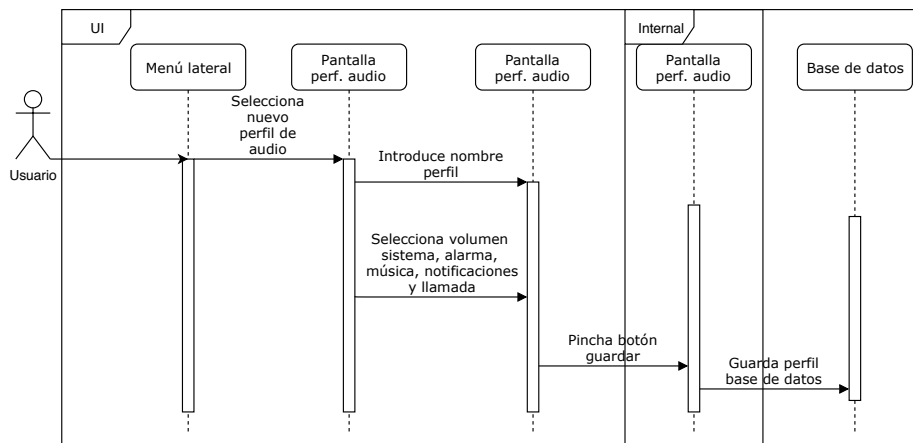


Figura C.3: Imagen que muestra el diagrama de secuencia a la hora de generar un perfil de sonido.

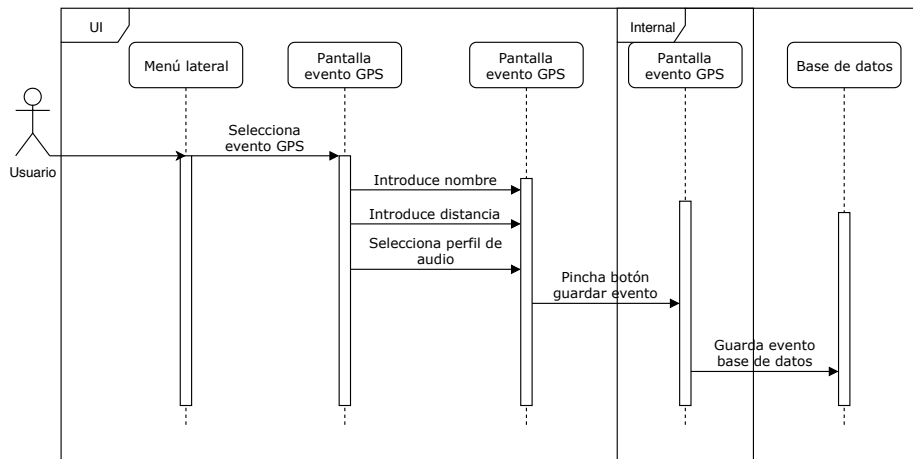


Figura C.4: Imagen que muestra el diagrama de secuencia a la hora de generar un evento GPS.

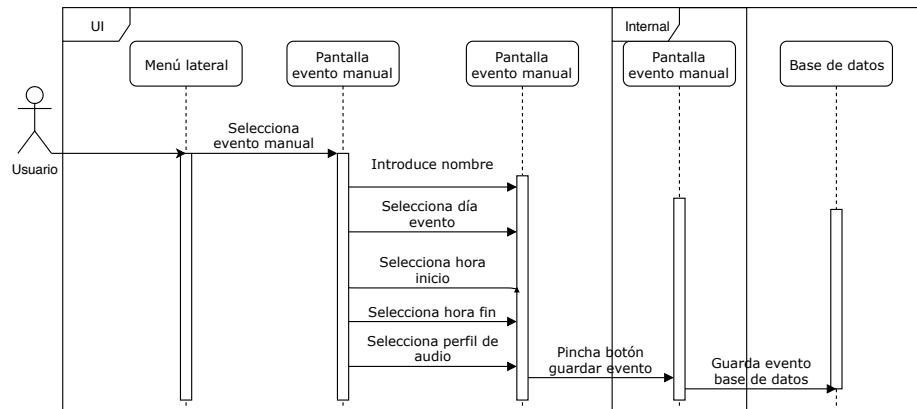


Figura C.5: Imagen que muestra el diagrama de secuencia a la hora de generar un evento manual.

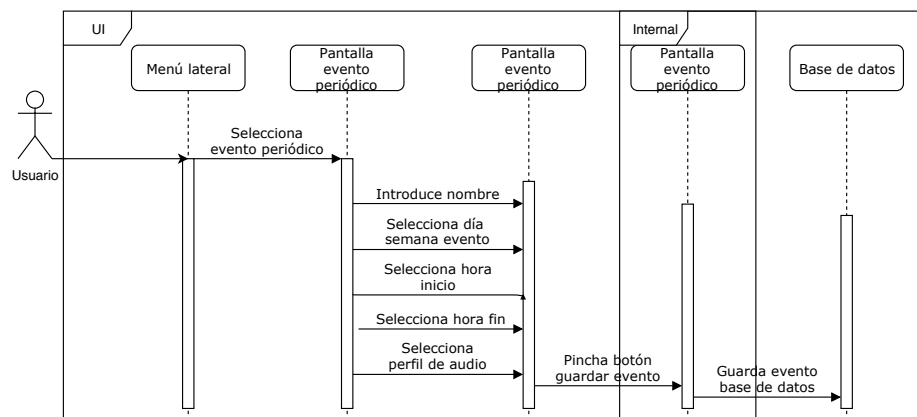


Figura C.6: Imagen que muestra el diagrama de secuencia a la hora de generar un evento periódico.

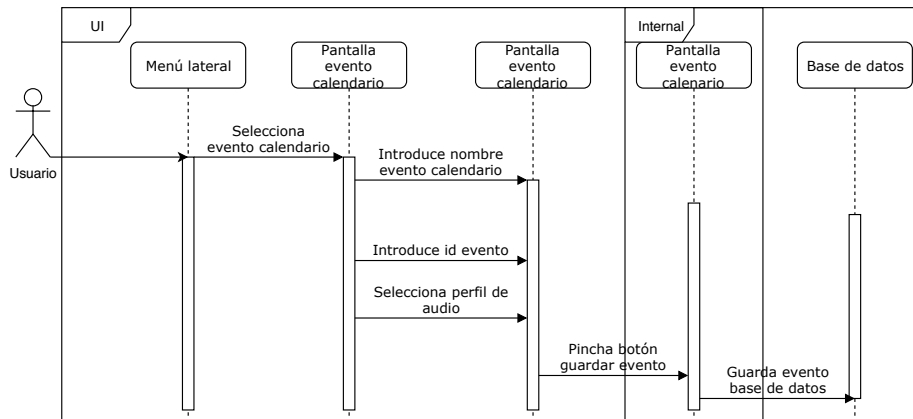


Figura C.7: Imagen que muestra el diagrama de secuencia a la hora de generar un evento de calendario.

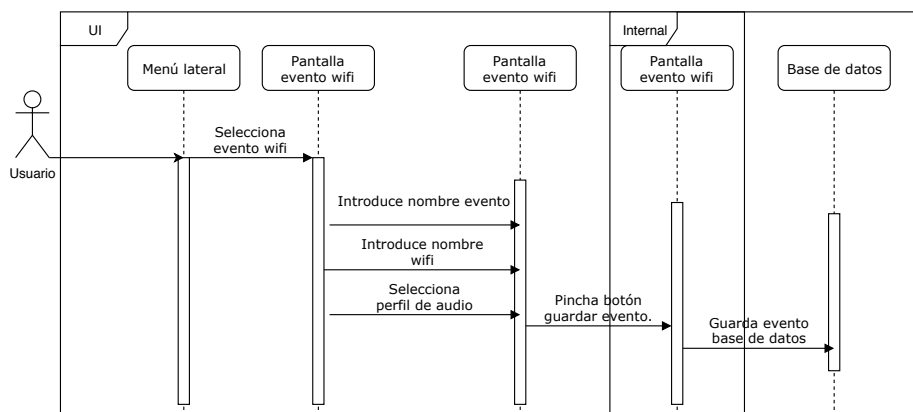


Figura C.8: Imagen que muestra el diagrama de secuencia a la hora de generar un evento wifi.

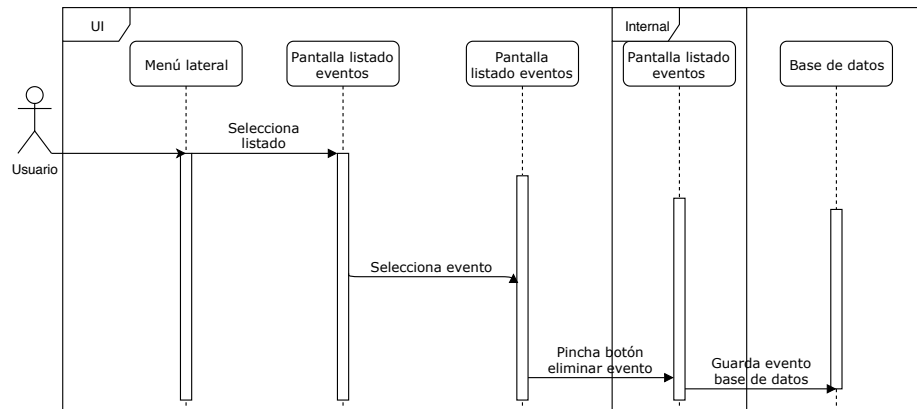


Figura C.9: Imagen que muestra el diagrama de secuencia para el borrado de eventos.

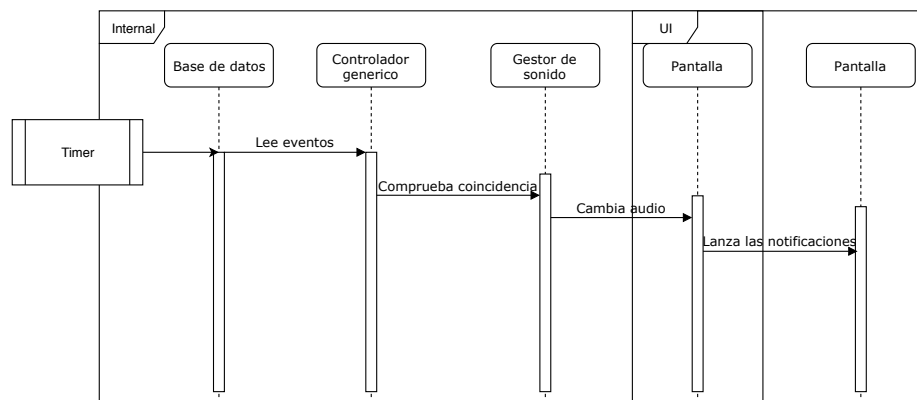


Figura C.10: Imagen que muestra el diagrama de secuencia de como funciona el hilo que controla los eventos manuales, periódicos, wifi y calendario.

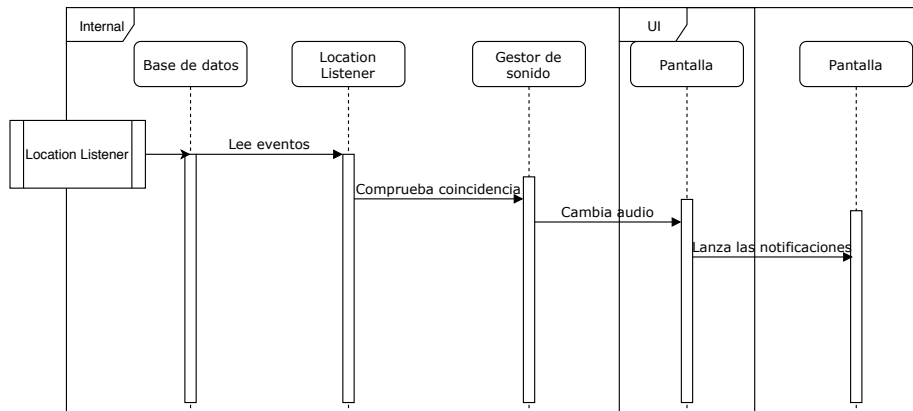


Figura C.11: Imagen que muestra el funcionamiento del listener GPS.

## C.4. Diseño arquitectónico

Como diseño arquitectónico podríamos decir que es una aplicación que se basa en el modelo vista controlador, ya que un controlador irá sacando los datos que se requieran a la vista, de un modelo de datos que no está en contacto directo con la vista, siendo el controlador el que haga el paso de la parte front-end a la back-end.

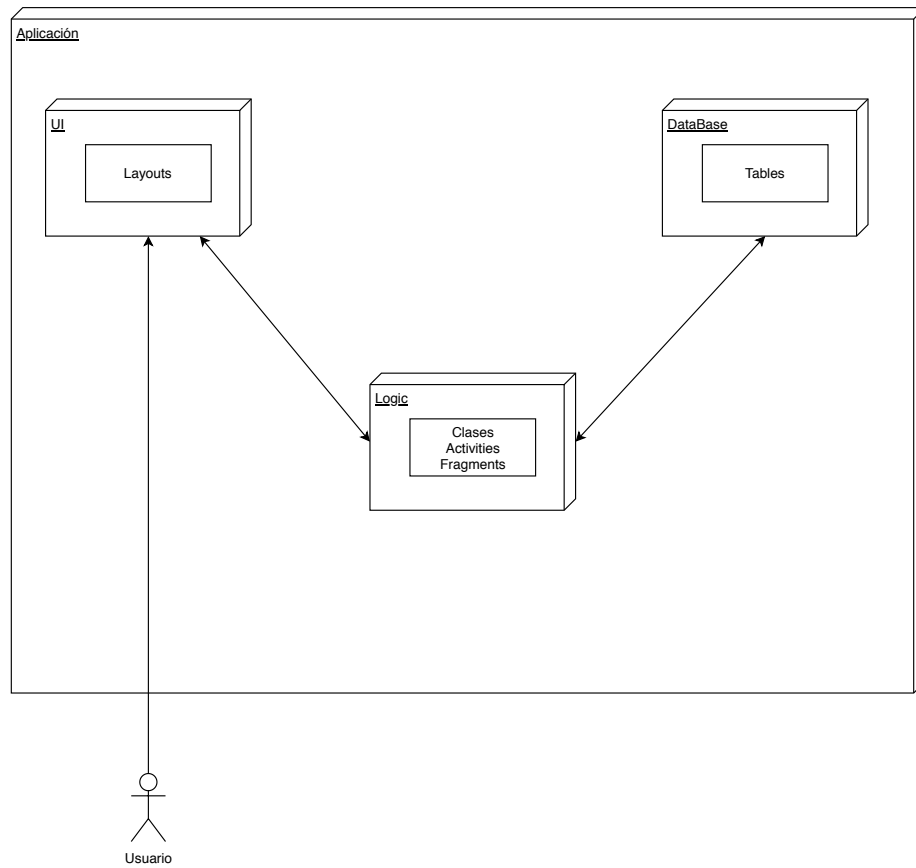


Figura C.12: Imagen que muestra la arquitectura MVC.

## Directorios

A continuación se añadirán los esquemas de cómo está montada la aplicación a nivel de directorios.

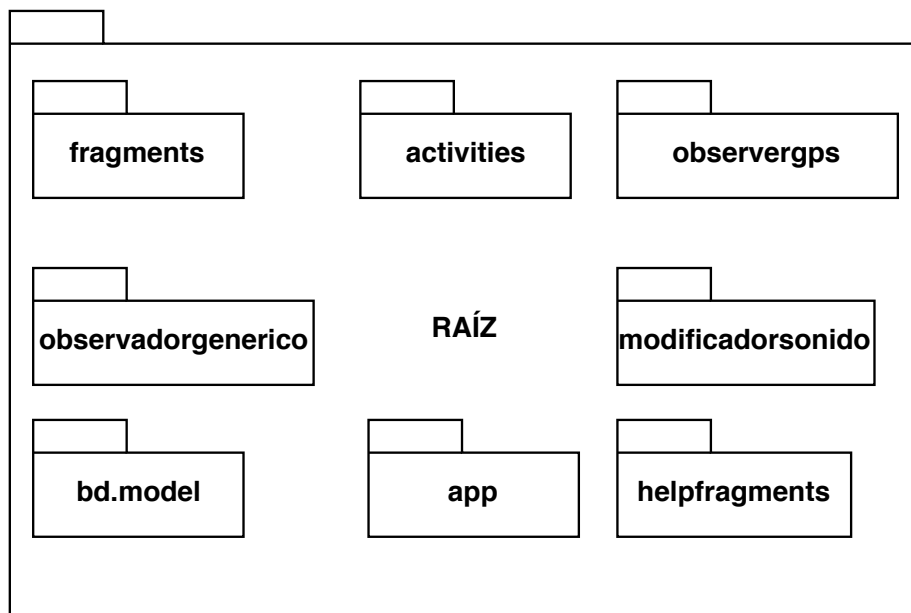


Figura C.13: Imagen que muestra la distribución de los directorios dentro de nuestra aplicación.

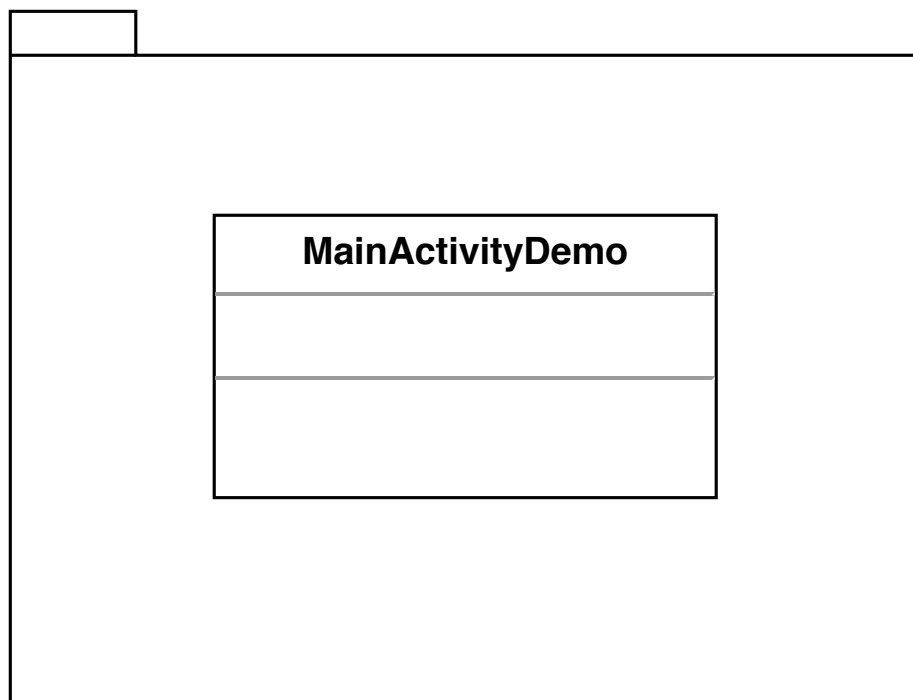


Figura C.14: Imagen que muestra la distribución de clases en el directorio activities.



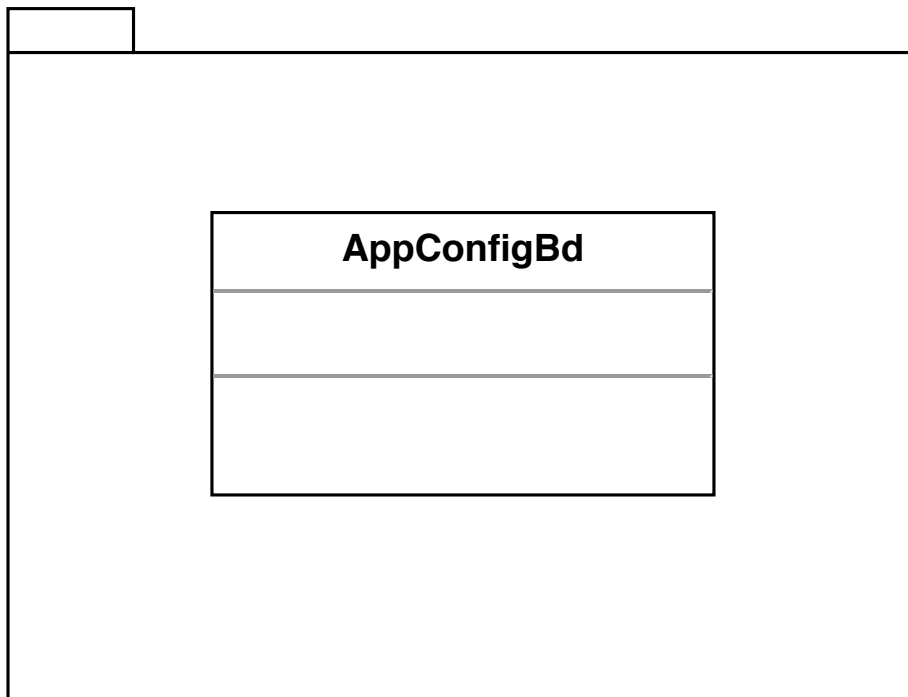


Figura C.15: Imagen que muestra la distribución de clases en el directorio app.

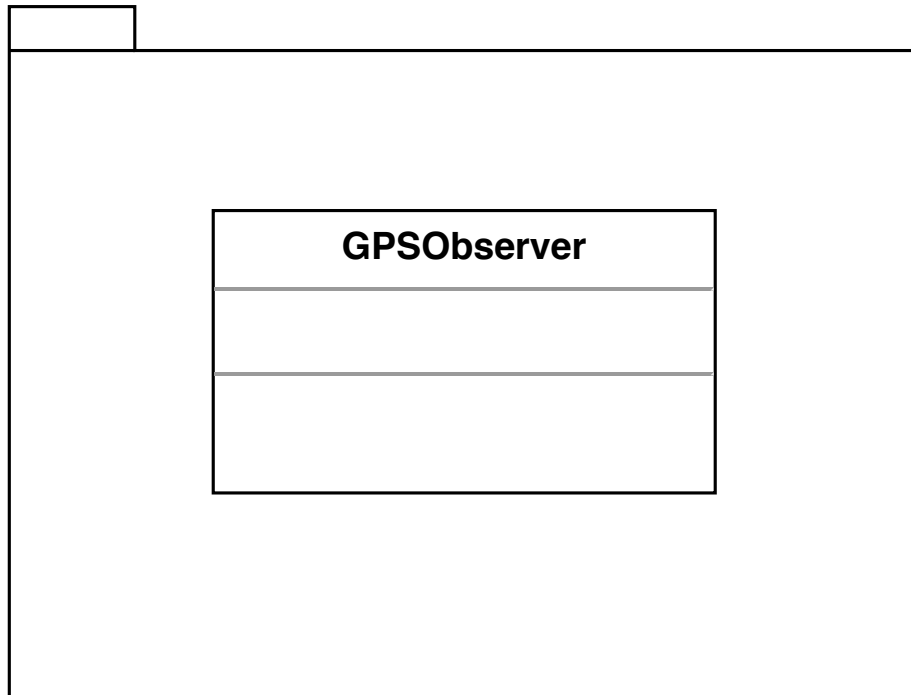


Figura C.16: Imagen que muestra la distribución de clases en el directorio observergps.

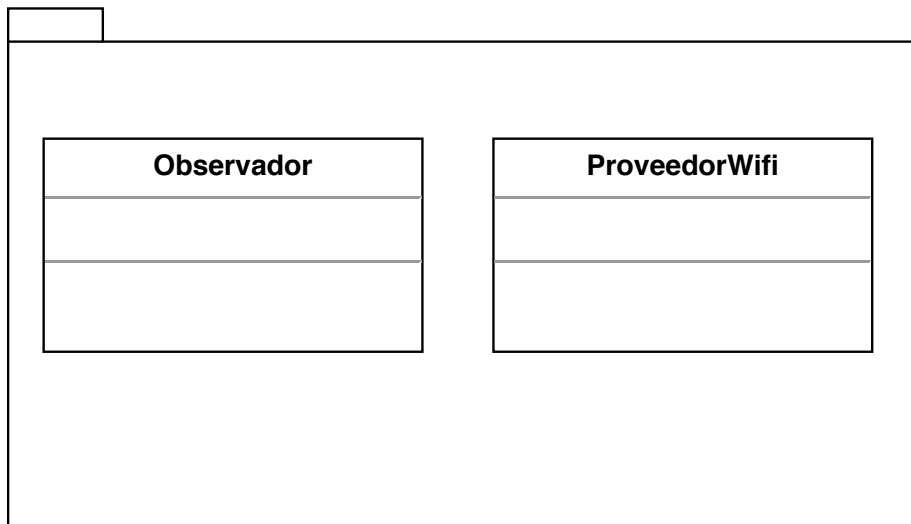


Figura C.17: Imagen que muestra la distribución de clases en el directorio observadorgenerico.

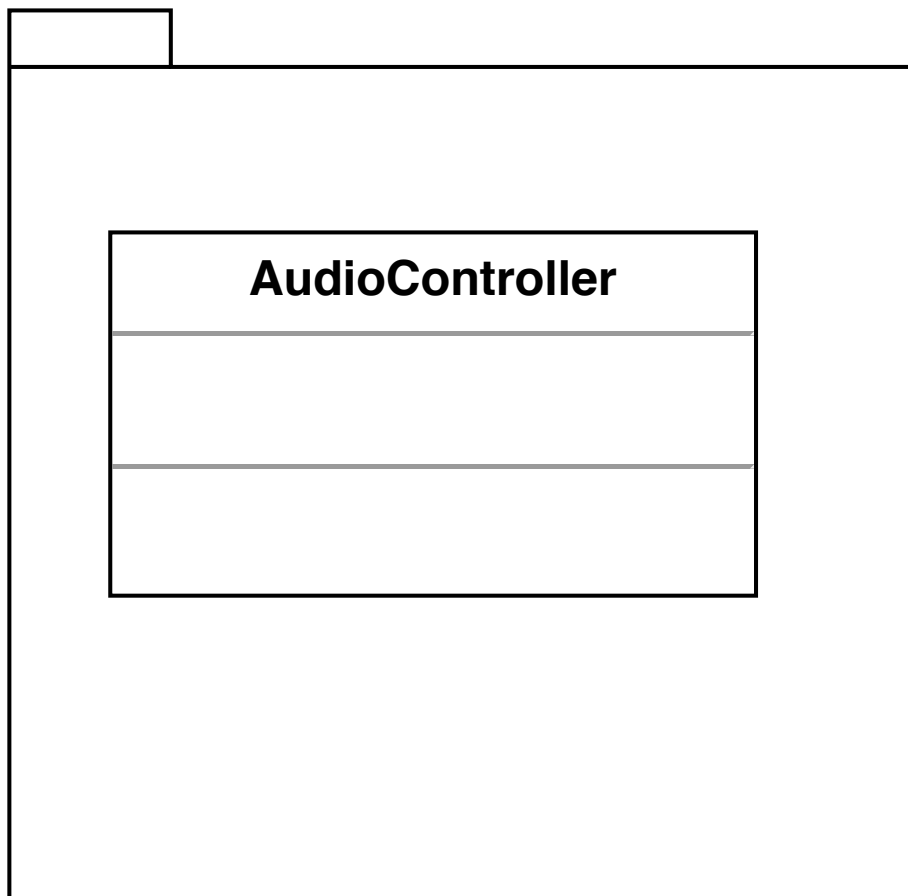


Figura C.18: Imagen que muestra la distribución de clases en el directorio modificadorsonido.

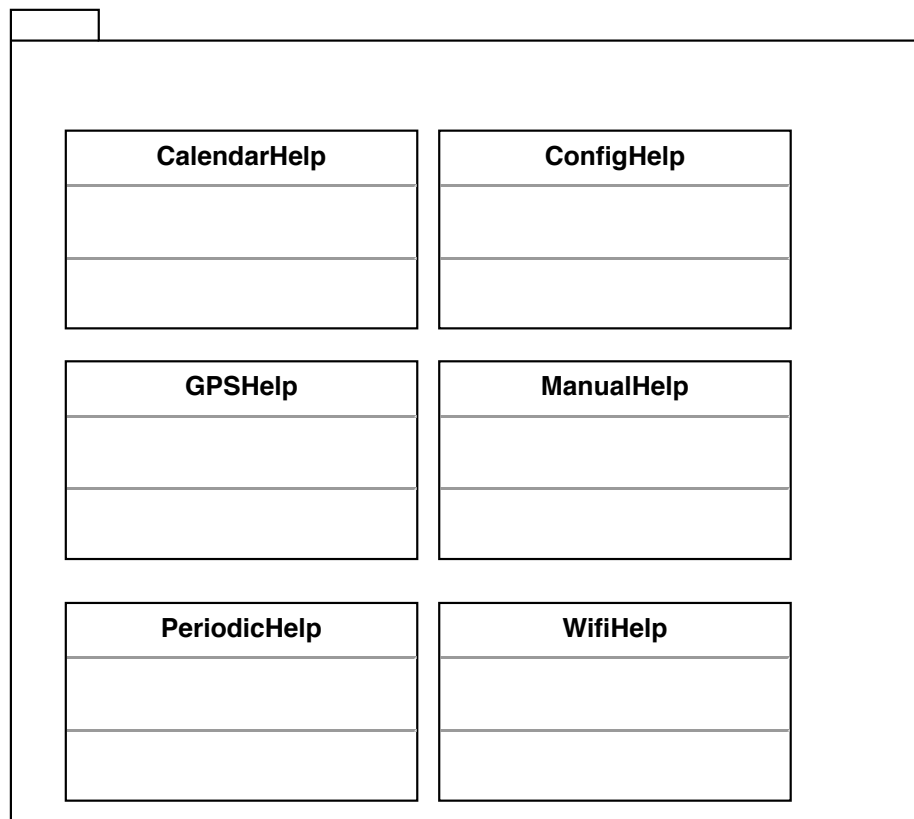


Figura C.19: Imagen que muestra la distribución de clases en el directorio `helpfragments`.

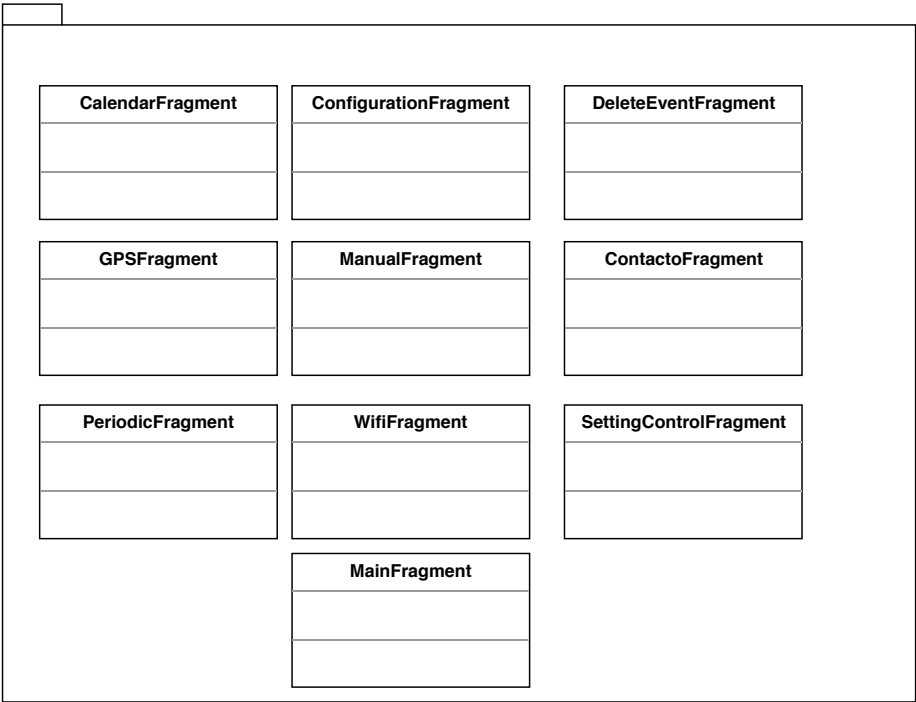


Figura C.20: Imagen que muestra la distribución de clases en el directorio fragments.

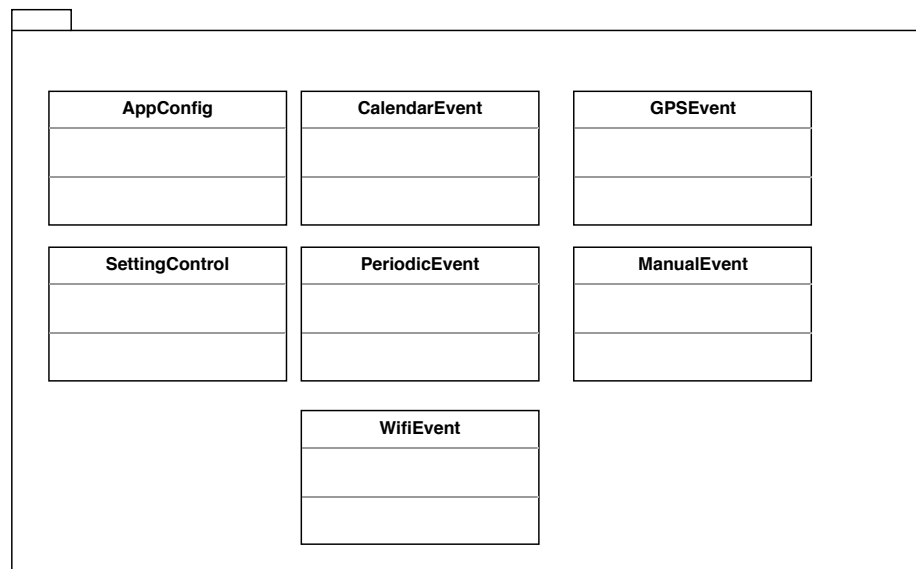


Figura C.21: Imagen que muestra la distribución de clases en el directorio `bd.model`.

## *Apéndice D*

---

# Documentación técnica de programación

---

## D.1. Introducción

A continuación veremos como se ha estado trabajando para generar la aplicación y algunos conceptos necesarios.

## D.2. Estructura de directorios

- Directorio raíz aplicación: /Codigo/
- Directorio de la aplicación: /Codigo/app/
- Directorio del código fuente: /Codigo/app/src/
- Directorio principal de la app: /Codigo/app/src/main/
- Directorio test: /Codigo/app/src/test/
- Directorio test Android: /Codigo/app/src/androidTest
- Directorio código de la aplicación: /Codigo/app/src/java/
- Directorio de los recursos: /Codigo/app/src/res/
- Directorio de la documentación: /Documentacion/
- Directorio de los textos de la documentación: /Documentacion/text/
- Directorio de las imágenes de la documentacion: /Documentacion/img/

## **D.3. Manual del programador**

### **Android Studio**

Android Studio[2] será el IDE de desarrollo que se utilizará para programar en java, el cual se podrá instalar de una manera rápida y sencilla, permitiéndonos el desarrollo de forma sencilla gracias a las ayudas que este tiene para programar, a sí como las ventajas de poder disponer de un emulador para probar al instante el código que se va generando. Otra ventaja de utilizar Android Studio, es que cuando se instala este ya nos indica que se instalará la versión de java correspondiente para el desarrollo de las aplicaciones.

### **Ficheros de configuración**

En nuestro proyecto encontraremos tres ficheros de configuración, dos ficheros que utiliza Gradle, para añadir dependencias a nuestro proyecto y líneas de ejecución de este, y un archivo manifest donde indicaremos todo lo relacionado con la ejecución de librerías de android, versionado de la aplicación y diferentes componentes android.

### **Añadir permisos a la aplicación**

Para añadir nuevos permisos a la aplicación, estos tendrán que ser añadidos al archivo AndroidManifest.xml. Aun con todo eso, es posible que dichos permisos también tengan que ser controlados por código.

### **Añadir nuevas librerías/dependencias**

Como se realizó para añadir el Easy Content Provider[4], para poder añadir una nueva librería, esta se tendrá que añadir al fichero build.gradle(Module:app) y asegurarnos de que se han actualizado las dependencias del proyecto para poder usar la librería con tranquilidad.

### **Cómo saber que version de Android utiliza un proyecto**

Para poder saber qué version de Android se está contemplando, se tendrá que ir al archivo build.gradle(Module:app) y fijarnos cual es la version de compilación, la versión mínima permitida y cual es la máxima.



## Importar un nuevo proyecto

Para importar un nuevo proyecto desde Android Studio simplemente tendremos que ir a /File/Open... y seleccionar el directorio donde se encuentra nuestro proyecto.

## Comandos básicos de Git

A continuación se expondrán unos comandos básicos para la utilización de Git [3].

- git push origin rama: todo lo comiteado se sube al repositorio en la nube en la rama establecida.
- git commit -m : realizamos el commit con el mensaje que introduzcamos dentro de las comillas.
- git add fichero: añadimos a lo traqueado el fichero.
- git add -A .: todo lo que haya en el directorio y sus sub directorios con cambios será añadido.
- git clone url: clonamos en la carpeta actual el repositorio señalado en al url.

## Comandos básicos de Git

Para obtener una revisión de la calidad de nuestro código, podremos utilizar la herramienta Codacy[1], la cual nos sacará una serie de informes para estar al tanto de la calidad de nuestro código, sugiriendo para cada caso cual sería la correcta manera de solventar el problema.

## D.4. Compilación, instalación y ejecución del proyecto

Una vez ya realizados los pasos anteriores vistos en el manual del programador podremos realizar los siguiente pasos.

### Compilación de la aplicación

Para ello tendremos que tener el proyecto abierto con el Android Studio y seleccionar en la barra de herramientas el botón del desplegable de "Build",

## **APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN**

y ahí seleccionaremos "Build Project."º Rebuild Project"si ya lo habíamos buildeado anterior mente.

### **Instalación de la APP y ejecución del proyecto**

Para instalar la aplicación dentro de un simulador Android de IDE Android Studio lo que haremos será crear un dispositivo virtual con la versión de Android que queramos simular. Una vez tengamos creada la máquina virtual de Android lo que haremos será pinchar en el botón de Run.<sup>en</sup> la barra de herramientas de Android Studio, o abriendo el desplegable Runz seleccionando la opción Run 'app'. Una vez hecho esto, tendremos que seleccionar un dispositivo sobre el que instalar y ejecutar la aplicación, por consiguiente se nos abrirá una ventana donde seleccionar el dispositivo simulado sobre el que correr la aplicación, permitiéndonos también crear nuevos dispositivos simulados. Una vez seleccionado el dispositivo pincharemos en el botón .ºKz esperaremos a que el proyecto se buildee, y que la máquina virtual se arranque. Una vez arrancada la máquina virtual automáticamente se instalará y se ejecutará.

## **D.5. Pruebas del sistema**

A continuación se expondrán las pruebas que se le han realizado al sistema.

### **Creación de pruebas Espresso**

Para generar pruebas automáticas, estas se podrán generar con el IDE Android Studio, para ello tendremos que ir a /Run/RecordEspressoTest. Una vez seleccionado, se nos abrirá el emulador de Android, en donde cada acción que se realice en la pantalla, esta será guardada, para posteriormente generar un fichero con las rutinas del test.

### **Pruebas unitarias de la base de datos**

#### **Creación de objetos en la base de datos**

**Premisas** Se tenga instanciada la base de datos y creado el modelo de datos. Se tenga definida la clase que controle los índices de la base de datos, y esta sea arrancada cuando se inicie la aplicación.

**Acciones** Se generarán todos los tipos de eventos en la base de datos. Cada vez que se genere una nueva entrada en la base de datos se comprobará que se introduce comprobando que el numero máximo del id generado es mayor al id máximo que había anteriormente.

**Resultados esperados** Se comprueba que se insertan correctamente todos los tipos de eventos en la base de datos.

**Resultado** Prueba superada.

### Obtención de datos de la base de datos

**Premisas** Se tenga instanciada la base de datos y creado el modelo de datos. Se tenga definida la clase que controle los índices de la base de datos, y esta sea arrancada cuando se inicie la aplicación. Haya datos guardados dentro de la base de datos.

**Acciones** Se hará una búsqueda de cada tipo de datos instanciados. Se comprobará que se puede iterar sobre dichos datos.

**Resultados esperados** Afirmamos que se puede iterar sobre objetos creados de la base de datos y su contenido corresponde con el introducido.

**Resultado** Prueba superada.

## Pruebas unitarias del observador GPS

### Obtención de la localización GPS en cualquier momento

**Premisas** Se tienen que tener aceptados los permisos de localización GPS en la aplicación.

**Acciones** Se generarán todos los tipos de eventos en la base de datos. Cada vez que se genere una nueva entrada en la base de datos se comprobará que se introduce comprobando que el numero máximo del id generado es mayor al id máximo que había anteriormente.

**Resultados esperados** Se comprueba que se insertan correctamente todos los tipos de eventos en la base de datos.

## ~~36~~ **APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN**

**Resultado** Prueba superada.

### **Acceder a los eventos de calendario**

**Premisas** Se tendrán los permisos de lectura y escritura de calendarios en la aplicación.

**Acciones** Se obtendrán los calendarios y sus eventos para iterar sobre ellos.

**Resultados esperados** Se esperará un listado de eventos sobre el que poder iterar.

**Resultado** Prueba superada.

### **Obtener SSID del wifi**

**Premisas** No se requieren premisas.

**Acciones** Se obtendrá la ssid del wifi actual.

**Resultados esperados** Se obtendrá unknown ssid ya que no se puede conectar el emulador a redes wifi.

**Resultado** Prueba superada.

### **Observador genérico**

**Premisas** No se requieren premisas.

**Acciones** Se tendrán que crear eventos para ver si la aplicación los detecta.

**Resultados esperados** Si se encuentran eventos que coincidan con las condiciones, se tendrá que informar de ellos.

**Resultado** Prueba superada.

### **Borrado de eventos**

**Premisas** Tendrá que haber al menos un evento creado.

**Acciones** Se tendrá que seleccionar un evento de la lista, y seleccionar el botón de borrado de eventos.

**Resultados esperados** El evento seleccionado será borrado.

**Resultado** Prueba superada.

### **Bloqueo de eventos**

**Premisas** No hay premisas

**Acciones** Se tendrá se seleccionar la configuración de la app del menú desplegable y guardar la configuración.

**Resultados esperados** Se guarda la configuración de la app, y los eventos no se saltan las restricciones que esta les pone.

**Resultado** Prueba superada.

### **Navegación por la ayuda**

**Premisas** No hay premisas.

**Acciones** Navegaremos por las diferentes pestañas de ayuda y por los enlaces de contacto.

**Resultados esperados** Se puede navegar de manera correcta a través de nuestra aplicación y a su vez podremos acceder a las diferentes páginas / aplicaciones de contacto.

**Resultado** Prueba superada.

## **Pruebas automáticas de la aplicación**

Se han realizado una serie de pruebas automáticas con la herramienta de Espresso, para el testeo de aplicaciones Android. Las pruebas realizadas son similares a las anteriormente indicadas, pero estas son realizadas de manera automática por Android Studio. Para ejecutar dichas



## *Apéndice E*

---

# Documentación de usuario

---

### **E.1. Introducción**

A continuación describiremos como trabajar con la aplicación desarrollada y los requisitos que se necesiten para poder utilizarla y como poder instalar la aplicación.

### **E.2. Requisitos de usuarios**

El usuario tendrá que tener un dispositivo Android que tenga GPS y conexión Wifi. La aplicación podrá ser arrancada desde la versión 19 de Android.

### **E.3. Instalación**

Una vez tengamos la aplicación, lo que tendremos que hacer será pinchar en la APK, y el dispositivo Android instalará automáticamente la aplicación. Puede ser que nuestro dispositivo rechace la instalación de la APK ya que no tiene "garantías" dicha APK al no provenir dicha aplicación del Play Store de Google, a si que es posible que tengamos que habilitar desde los ajustes de nuestro dispositivo la instalación de aplicaciones de procedencia desconocida o de terceros. Esta opción la podremos activar o desactivar desde Ajustes/Seguridad.

## E.4. Manual del usuario

A continuación se indicará el contenido de la aplicación, como los eventos que se podrán utilizar y los diferentes elementos que nos encontraremos en ella.

### ¿ Qué es y para que sirve esta aplicación?

En este proyecto se desarrollará una aplicación móvil para usuarios de Android. Lo que se intentará sera el automatizar de una manera sencilla el manejo de los volúmenes de sonido de nuestro dispositivo Android, para ello se podrán crear diferentes eventos que se comprobarán de manera automática y periódicamente y que cambiarán estos volúmenes. Los eventos que se podrán programar serán eventos manuales, eventos periódicos, eventos wifi, eventos Gps y eventos de calendario.

### ¿ Pero qué son los eventos?

Los eventos son condiciones que podremos definir para activar las configuraciones de sonido. Podremos encontrar los siguientes eventos:

#### Manuales

En estos eventos tendremos que definir el día y la hora entre los que queramos que se cambie a la configuración deseada.

#### Periódicos

En estos eventos tendremos que definir el día de la semana y las horas en las que queramos que se active la configuración deseada.

#### Wifi

En estos eventos tendremos que definir el nombre de la red Wifi y la configuración a establecerá. Si el dispositivo móvil detecta que hay una red Wifi al alcance con ese nombre se establecerá la configuración deseada.

#### Gps

En estos eventos tendremos que definir la configuración deseada, y cuando guardemos la configuración, el dispositivo móvil detectará la posición GPS



y la guardará. Cuando estemos cerca de esa posición GPS se cambiará de configuración de sonido.

### **Calendario**

En estos eventos se tendrá que definir el Id de una tarea de calendario, y la configuración de sonido a la que queramos cambiar. Si encuentra ese nombre de tarea en uno de los calendarios del móvil, se cambiará la configuración de sonido.

### **¿ Pero qué son las configuraciones de sonido ?**

Las configuraciones de sonido son, los diferentes volúmenes que tiene el móvil, agrupados en una sola entidad. Dentro de una configuración de sonido modificaremos el sonido principal del móvil, llamadas, notificaciones, música y alarma.

### **Aceptar los permisos de la aplicación**

Para poder navegar entre las diferentes pantallas tendremos que utilizar el menú desplegable lateral de la aplicación.

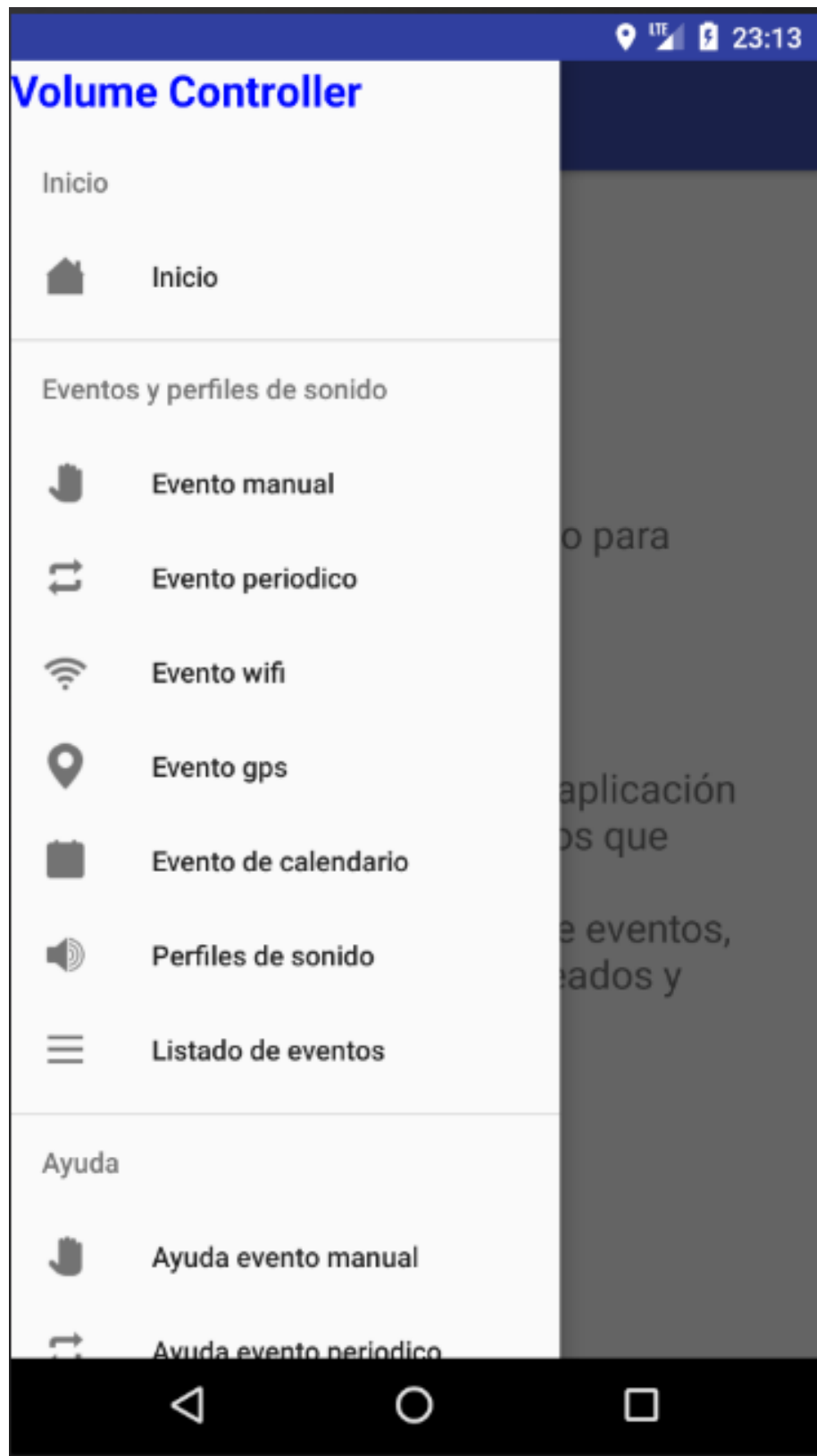


Figura E.1: Imagen que muestra el menú de la aplicación.

**Aceptar los permisos de la aplicación**

Lo primero que tendremos que hacer, para poder utilizar toda la potencia de la aplicación será la aceptación de los permisos, para que la aplicación pueda utilizar el GPS y el calendario, para ello nos saldrán dos ventanas emergentes pidiéndonos dichas confirmaciones.

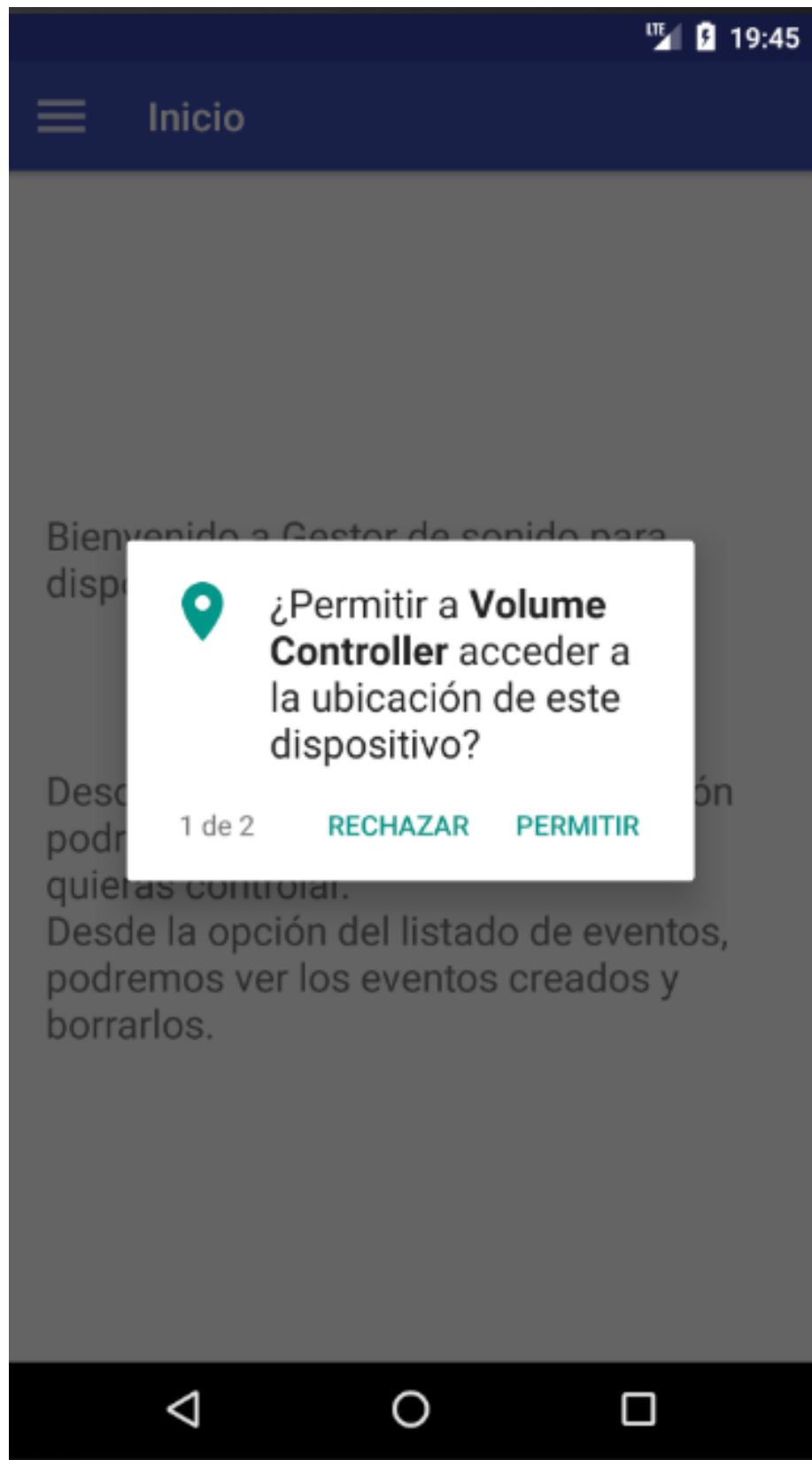


Figura E.2: Imagen que muestra el mensaje en el cual aceptaremos o denegaremos los permisos para permitir que la aplicación pueda acceder a la ubicación del dispositivo.

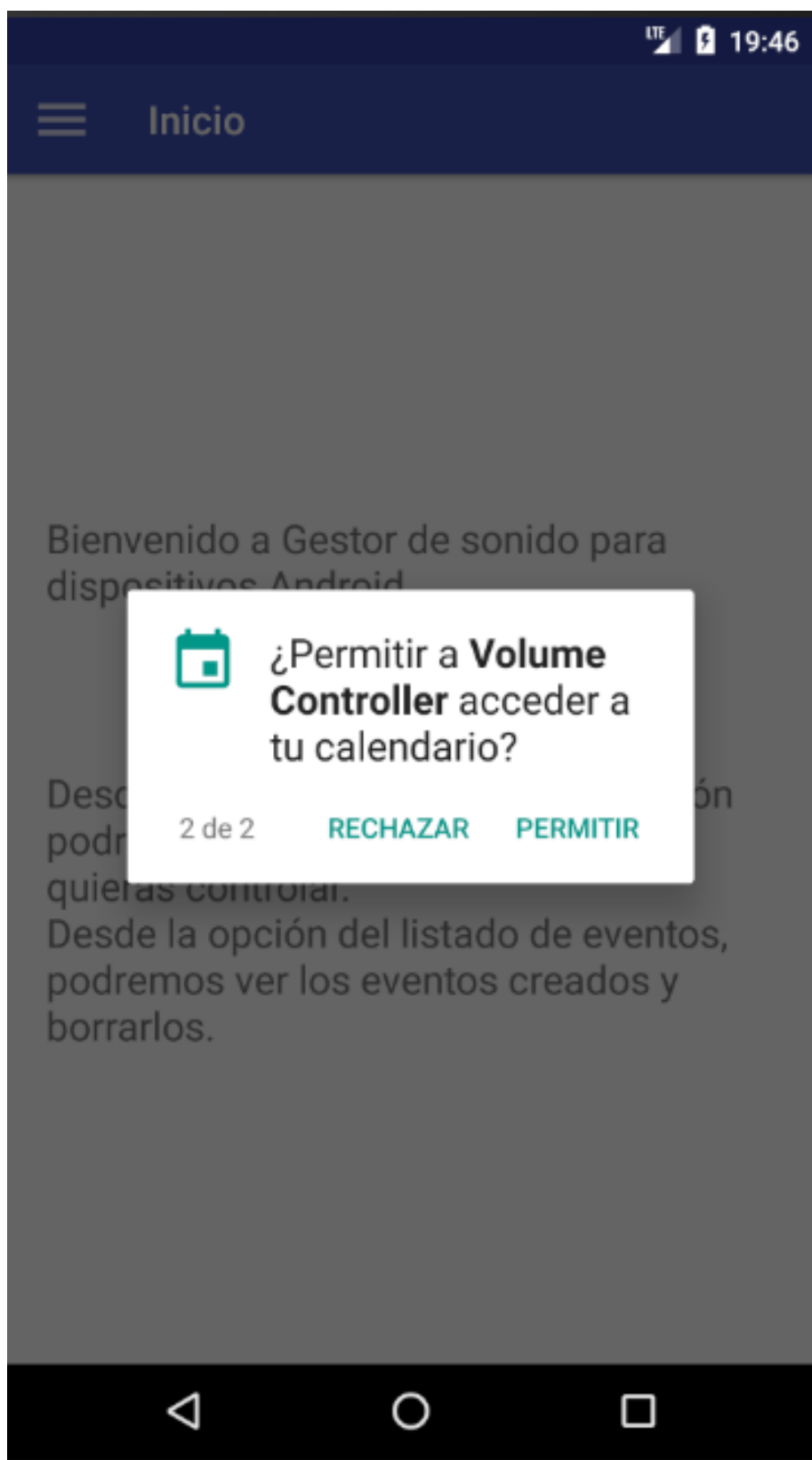


Figura E.3: Imagen que muestra el mensaje en el cual aceptaremos o denegaremos los permisos para permitir que la aplicación acceda al calendario de nuestro dispositivo.

## **Pantalla principal de la aplicación**

La primera pantalla que encontraremos será la pantalla principal de la aplicación en la cual nos encontraremos un mensaje de bienvenida y una pequeña ayuda.

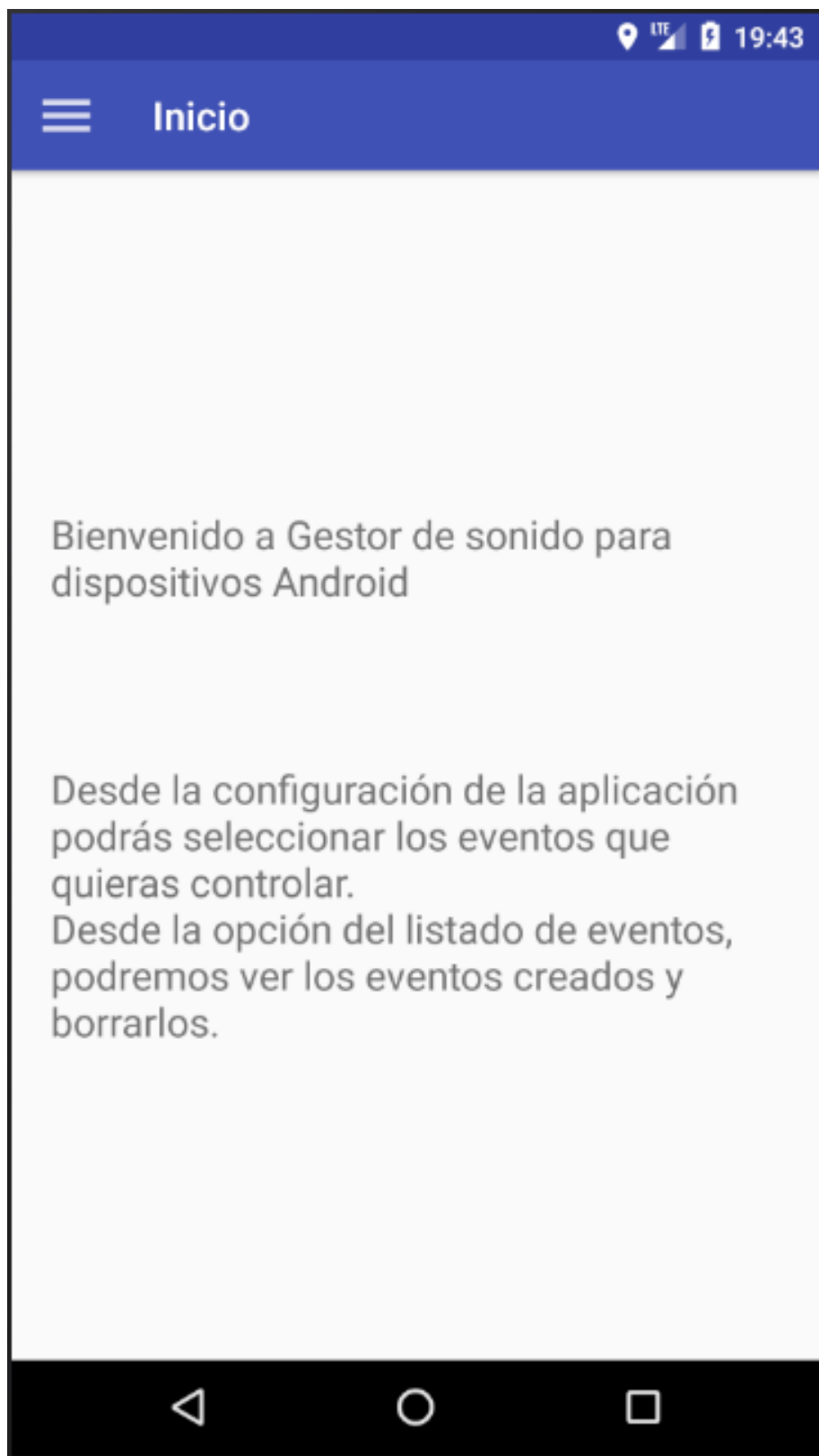


Figura E.4: Imagen que muestra la pantalla principal de la aplicación.

## **Añadir nuevos perfiles de sonido**

Para añadir un nuevo perfil de sonido tendremos que seleccionar en el menú desplegable de la izquierda la casilla de perfiles de sonido. Una vez dentro de esta pantalla seleccionaremos los niveles de volumen e introduciremos el nombre de la configuración. Para guardar la configuración tendremos que pinchar en el botón de guardar configuración. Centrándonos en los elementos de esta pantalla encontraremos:

1. Cuadro de texto para insertar el nombre de la configuración.
2. Selector de volumen principal.
3. Selector del volumen de la música
4. Selector del volumen de la alarma
5. Selector del volumen de conversación.
6. Selector del volumen de las notificaciones



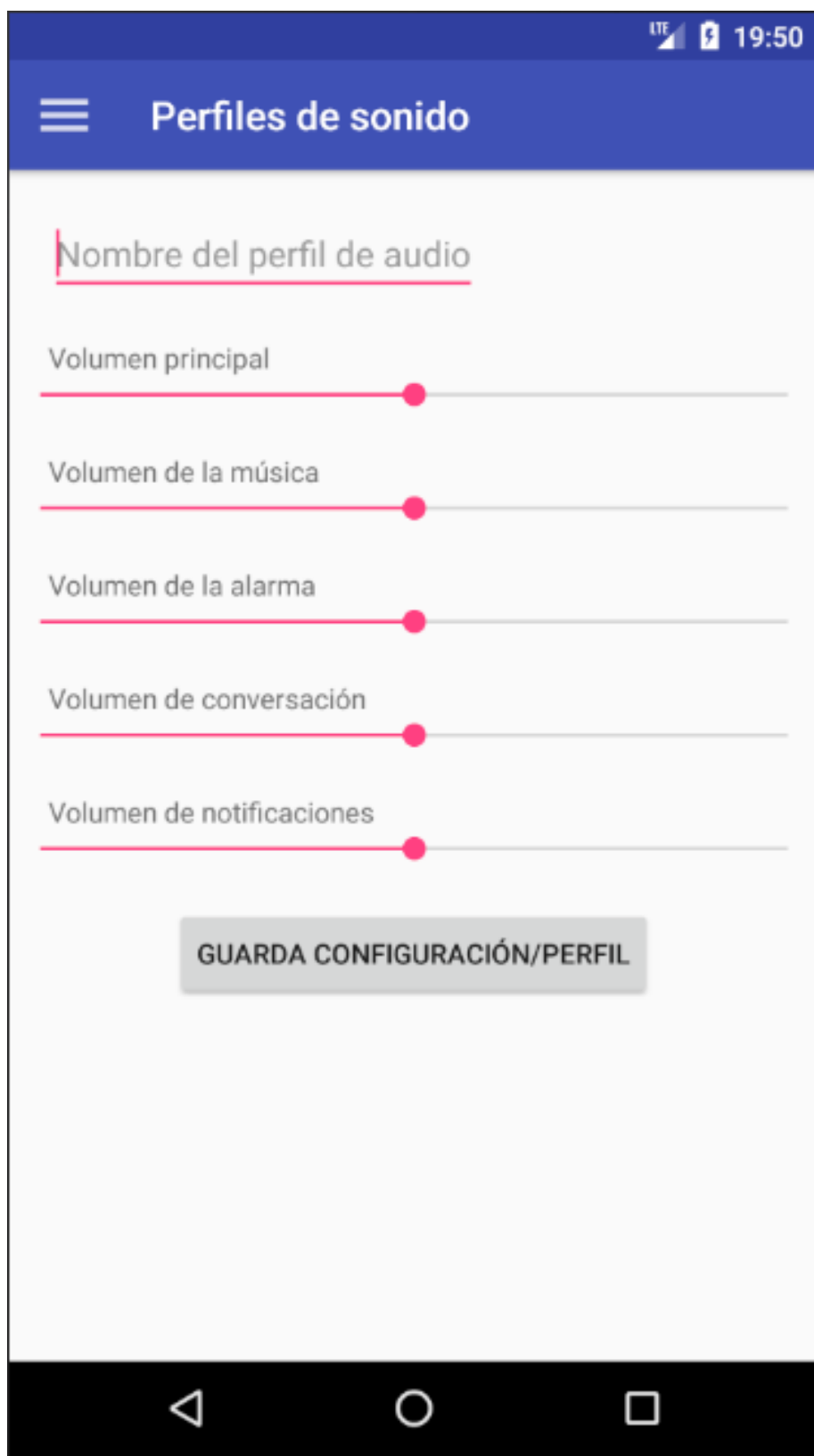


Figura E.5: Imagen que muestra la pantalla para la creación de perfiles de sonido.

## Añadir eventos periódicos

Para crear un evento periódico tendremos que seleccionar en el menú desplegable de la izquierda la casilla de eventos periódicos. Aquí tendremos que seleccionar la hora de inicio y la hora de fin del evento desde los botones respectivos, los cuales nos sacarán un selector de hora, para establecer dicha hora. Una vez seleccionadas las horas de inicio y de fin, tendremos que seleccionar el día de la semana que queremos y la configuración deseada a aplicar. Para ello tenemos dos listas desplegables para seleccionar estas opciones. Una vez que tengamos todos estos atributos seleccionados lo que tendremos que hacer será pinchar el el botón de guardar configuración para guardar los cambios realizados. Centrándonos en los elementos de esta pantalla encontraremos:

1. Cuadro de texto donde tendremos que introducir el nombre del evento a crear.
2. Botones para seleccionar las horas de inicio y fin del evento.
3. Menú desplegable donde seleccionaremos el día de la semana donde queremos crear el evento.
4. Menú desplegable donde seleccionaremos la configuración de sonido anteriormente creada.
5. Botón para guardar el evento.

Introduce el nombre del evento periódico

SELECCIONA LA HORA DE INICIO

SELECCIONA LA HORA DE FIN

Selecciona un día de la semana

Lunes

Selecciona un perfil de audio

Perfil

Figura E.6: Imagen que muestra la pantalla para la creación de eventos periódicos.

## **Añadir evento manual**

Para generar un evento manual tendremos que seleccionar en el menú desplegable de la izquierda la casilla de eventos manuales. En esta pantalla lo primero que tendremos que hacer será seleccionar el día del evento, pinchando en el botón de selección de día, y seleccionándolo en el pop-up del calendario que sale al pinchar en el botón. De la misma manera que antes tendremos que seleccionar la hora de inicio y de finalización del evento en los selectores de hora emergentes de los botones de selección. Tendremos también que seleccionar la configuración deseada desde la lista de objetos emergente. Para guardar este evento manual tendremos que pinchar en el botón de guardar configuración.

Centrándonos en los elementos de esta pantalla encontraremos:

1. Cuadro de texto donde tendremos que introducir el nombre del evento a crear.
2. Botón para seleccionar el día que queremos activar el evento. Este botón nos mostrará un PopUp de un calendario, donde tendremos que seleccionar el día del evento.
3. Botones para seleccionar las horas de inicio y finalización del evento. Estos botones nos sacarán un PopUp de un reloj, donde tendremos que seleccionar las horas de inicio y fin de los eventos, respectivamente.
4. Menú desplegable donde seleccionaremos la configuración de sonido anteriormente creada.
5. Botón para guardar el evento.

Evento manual

Introduce el nombre del evento manual

SELECCIONA EL DIA DEL EVENTO

SELECCIONA LA HORA DE INICIO

SELECCIONA LA HORA DE FIN

Selecciona un perfil de audio

Perfil

Figura E.7: Imagen que muestra la pantalla para la creación de los eventos manuales.

## **Añadir evento GPS**

Para guardar un evento GPS lo que tendremos es que en la lista de configuraciones desplegable seleccionar la configuración deseada. Una vez hecho esto pincharemos en el botón de guardar configuración y automáticamente se nos guardará la posición GPS actual del terminal y la configuración deseada.

Centrándonos en los elementos de esta pantalla encontraremos:

1. Cuadro de texto donde tendremos que introducir el nombre del evento a crear.
2. Cuadro de texto donde introduciremos la distancia de localización.
3. Menú desplegable donde seleccionaremos el perfil de sonido anteriormente creada.
4. Botón para guardar el evento.

The screenshot shows a mobile application interface titled "Evento gps". At the top, there is a status bar with "LTE", a battery icon, and the time "19:48". Below the title bar, there is a menu icon (three horizontal lines) and the title "Evento gps". The main content area has a light gray background and contains three input fields. The first field is labeled "Introduce el nombre del evento GPS" and is empty. The second field is labeled "Inserta la distancia" and contains the number "100". The third field is labeled "Selecciona un perfil de audio" and has a dropdown menu with the text "Perfil" and a downward arrow. At the bottom of the form, there is a large gray button labeled "GUARDA EVENTO". The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Figura E.8: Imagen que nos muestra la pantalla para la creación de los eventos de GPS.

## **Añadir evento Wifi**

Para guardar un evento Wifi tendremos que seleccionar en el menú lateral desplegable y seleccionar la casilla de eventos Wifi. Introduciremos el nombre del evento. Introduciremos en el cuadro de texto el nombre de la red Wifi que deseemos detectar. En la lista desplegable de configuraciones seleccionaremos la configuración deseada. Para guardar el evento lo que tendremos que hacer será pinchar en el botón de guardar el evento Wifi.

Centrándonos en los elementos de esta pantalla encontraremos:

1. Cuadro de texto donde tendremos que introducir el nombre del evento a crear.
2. Cuadro de texto donde tendremos que introducir el nombre de la red Wifi sobre la que queramos hacer la comprobación.
3. Menú desplegable donde seleccionaremos la configuración de sonido anteriormente creada.
4. Botón para guardar el evento.



The screenshot displays a mobile application interface titled "Evento wifi". At the top, there is a blue header bar with a hamburger menu icon on the left and the title "Evento wifi" in white. The status bar at the very top shows "LTE", a battery icon, and the time "19:49". The main content area is white and contains three input fields: "Introduce el nombre del evento wifi" (highlighted with a red underline), "Introduce el nombre de la red wifi" (highlighted with a grey underline), and "Selecciona un perfil de audio". Below the last label is a dropdown menu with the text "Perfil" and a downward arrow. At the bottom of the form is a large grey button labeled "GUARDA EVENTO". The Android navigation bar is visible at the bottom with back, home, and recent apps icons.

Figura E.9: Imagen que nos muestra la pantalla para la creación de eventos wifi.

## Añadir evento Calendario

Para guardar un evento de calendario lo que tendremos que hacer será seleccionar en el menú lateral la casilla de eventos de calendario. Una vez en la pantalla de eventos de calendario lo que tendremos que hacer será introducir el título del evento sobre el que queremos que se active la configuración en el cuadro de texto. Seleccionaremos en la lista desplegable la configuración deseada a aplicar. Guardaremos el evento pinchando en el botón de guardar evento.

Centrándonos en los elementos de esta pantalla encontraremos:

1. Cuadro de texto donde tendremos que introducir el nombre del evento a crear.
2. Cuadro de texto donde tendremos que introducir el nombre del evento de calendario sobre el que queramos hacer la comprobación.
3. Menú desplegable donde seleccionaremos la configuración de sonido anteriormente creada.
4. Botón para guardar el evento.

The screenshot displays a mobile application interface for creating a calendar event. At the top, a blue header bar contains a hamburger menu icon and the title 'Evento de calendario'. The status bar above the header shows 'LTE', a battery icon, and the time '19:49'. The main content area is white and contains three input fields: 'Introduce el nombre del evento de calendario' (highlighted with a red underline), 'Introduce la palabra clave del evento' (underlined in grey), and 'Selecciona un perfil de audio' (with a dropdown menu currently showing 'Perfil'). At the bottom of the form is a grey button labeled 'GUARDA EVENTO'. The Android navigation bar is visible at the very bottom.

Figura E.10: Imagen que muestra la pantalla para la creación de los eventos de calendario.

## **Comprobar eventos y establecer configuraciones asociadas**

Automáticamente la aplicación buscará si hay eventos activos, revisando cada tipo de evento, y si hay alguna condición que active a estos, se realizará el cambio de sonido. Para saber el evento que está activo en ese preciso momento sólo tendremos que mirar la barra de navegación del dispositivo, y si hay activo algún evento, aparecerá la notificación de la aplicación con el nombre del perfil de sonido que está activo en ese preciso momento. La importancia de los eventos sobre otros son: Calendario >Wifi >Periódico >Manual. El evento de GPS es independiente y se irá activando según la localización.

## **Como activar o desactivar eventos**

Para que el usuario pueda seleccionar que eventos quiere que se activen, este podrá ir a la configuración de la aplicación para activar o desactivar los distintos eventos. Para ello iremos a la parte de configuración en el menú lateral y seleccionaremos la opción de configuración.

Encontraremos los siguientes elementos

1. 5 switches para activar o desactivar los eventos
2. Botón para guardar la configuración.

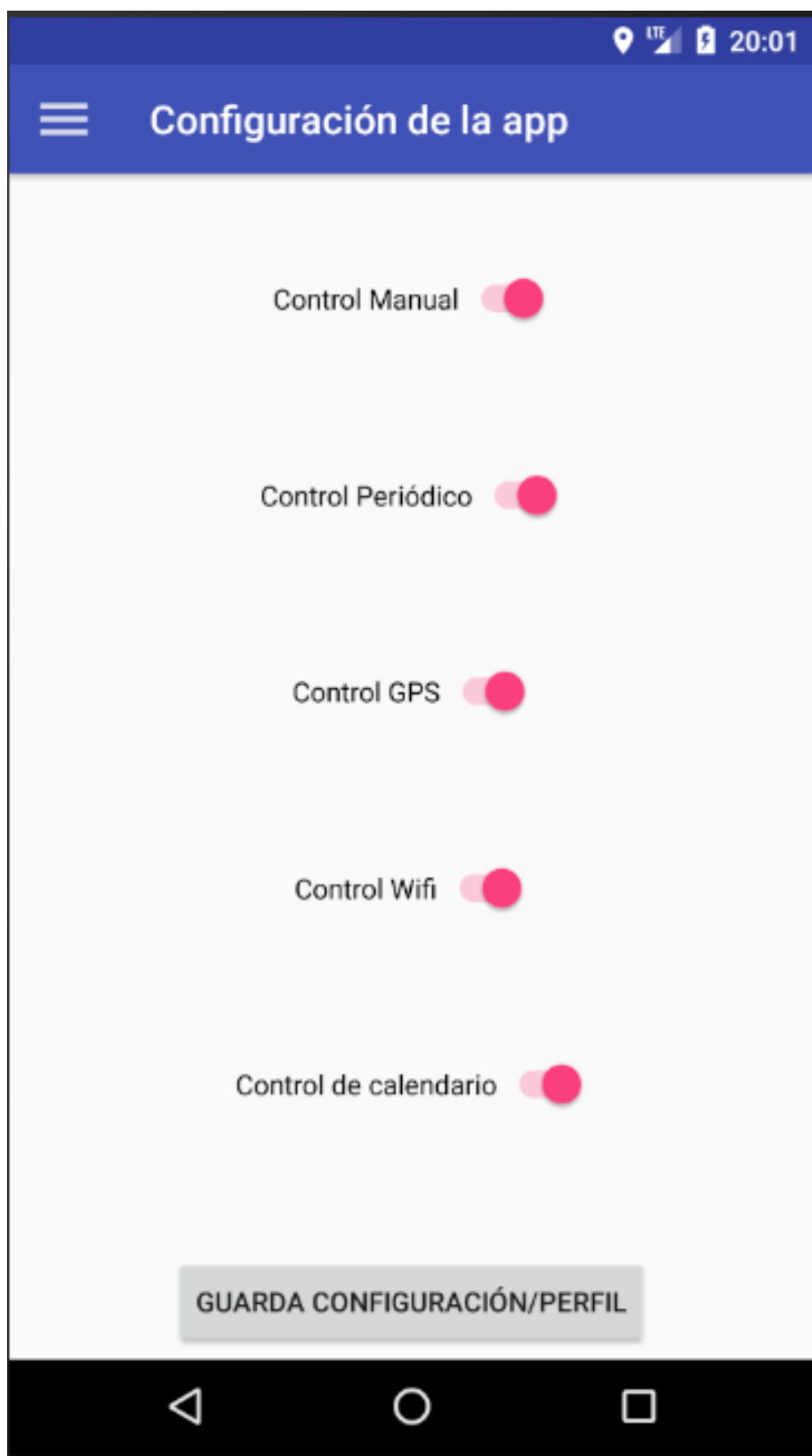


Figura E.11: Imagen en la que veremos la pantalla que nos permite el activar o desactivar un tipo de evento en concreto

## **Eliminar eventos**

Si queremos eliminar algún evento, tendremos que tener previamente creado alguno. Nos iremos a la opción del menú desplegable de listado de eventos, en el menú desplegable elegiremos el evento a borrar y cuando le demos al boton de eliminar, este evento se eliminará de manera automática.

Encontraremos los siguientes elementos:

1. Selector de los eventos.
2. Información sobre el evento seleccionado.
3. Botón que borrará el evento seleccionado.

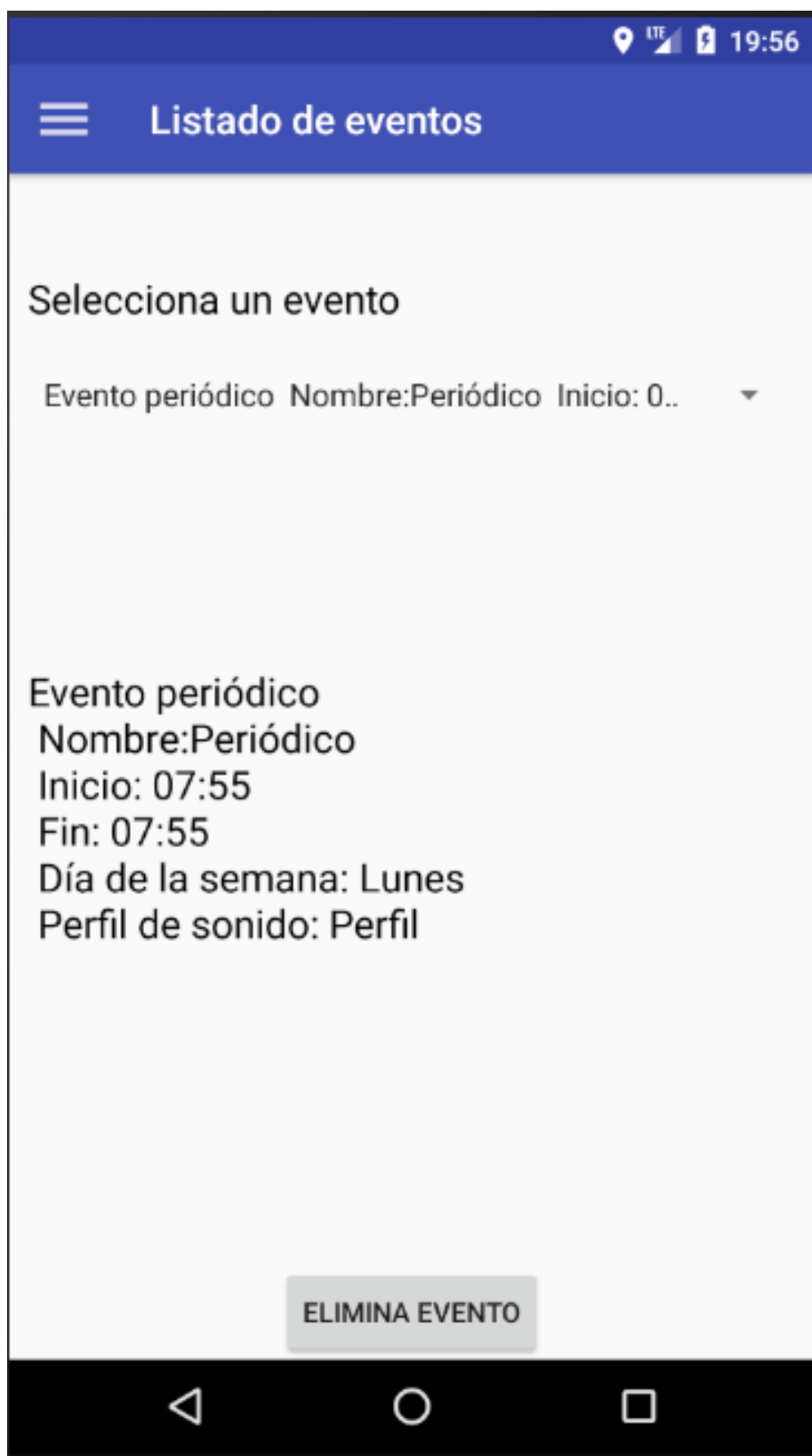


Figura E.12: Imagen que muestra la pantalla en la cual podremos ver los eventos que están creados y eliminarlos de la aplicación.

## **E.5. Manual del usuario dentro de la aplicación**

Para una mayor facilidad se ha introducido una pequeña guía de uso de la aplicación dentro de esta, a sí como una serie de enlaces a las diferentes formas de contacto para el soporte de la aplicación.



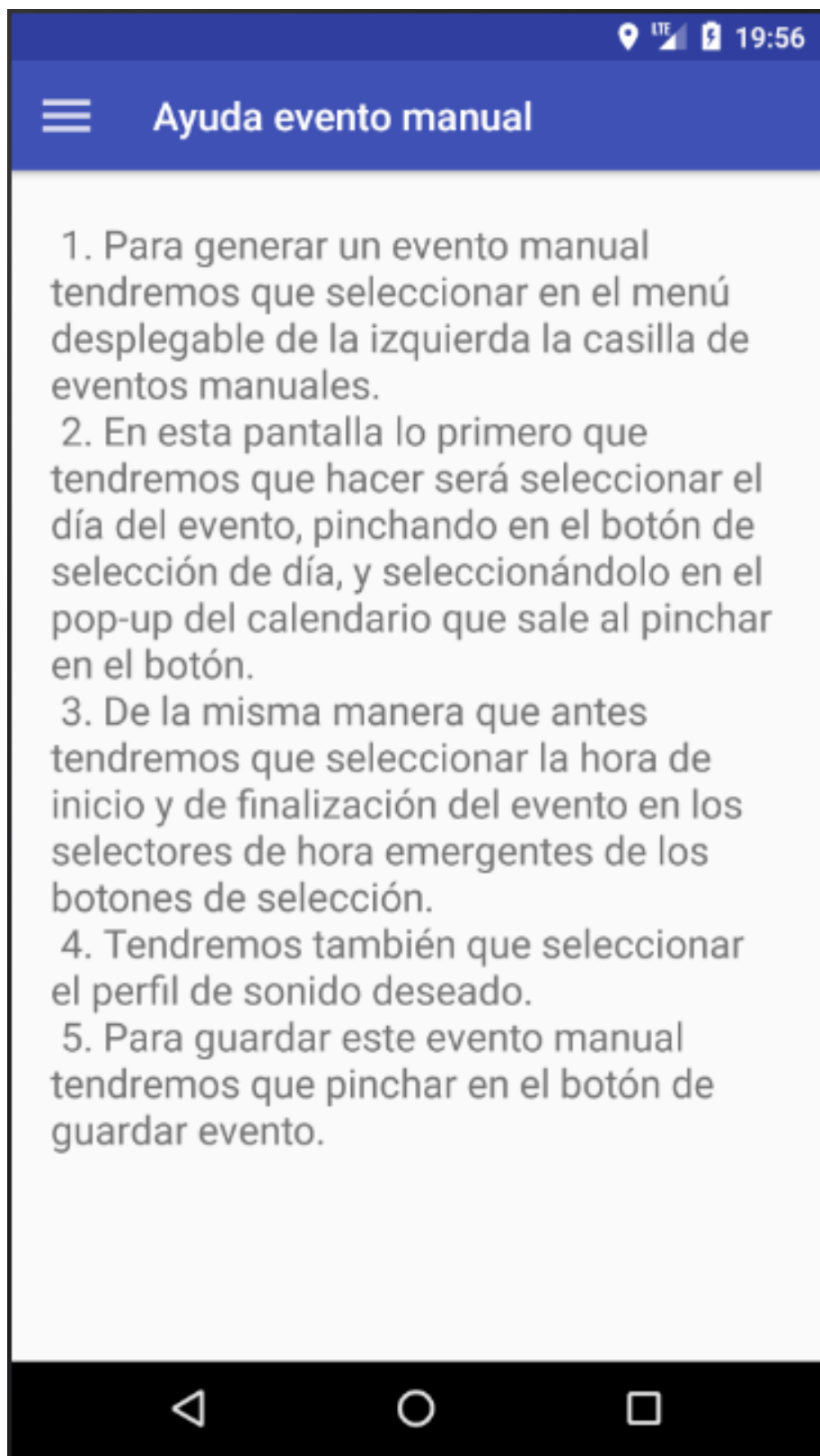


Figura E.13: Imagen que muestra la pantalla para la creación de eventos manuales.

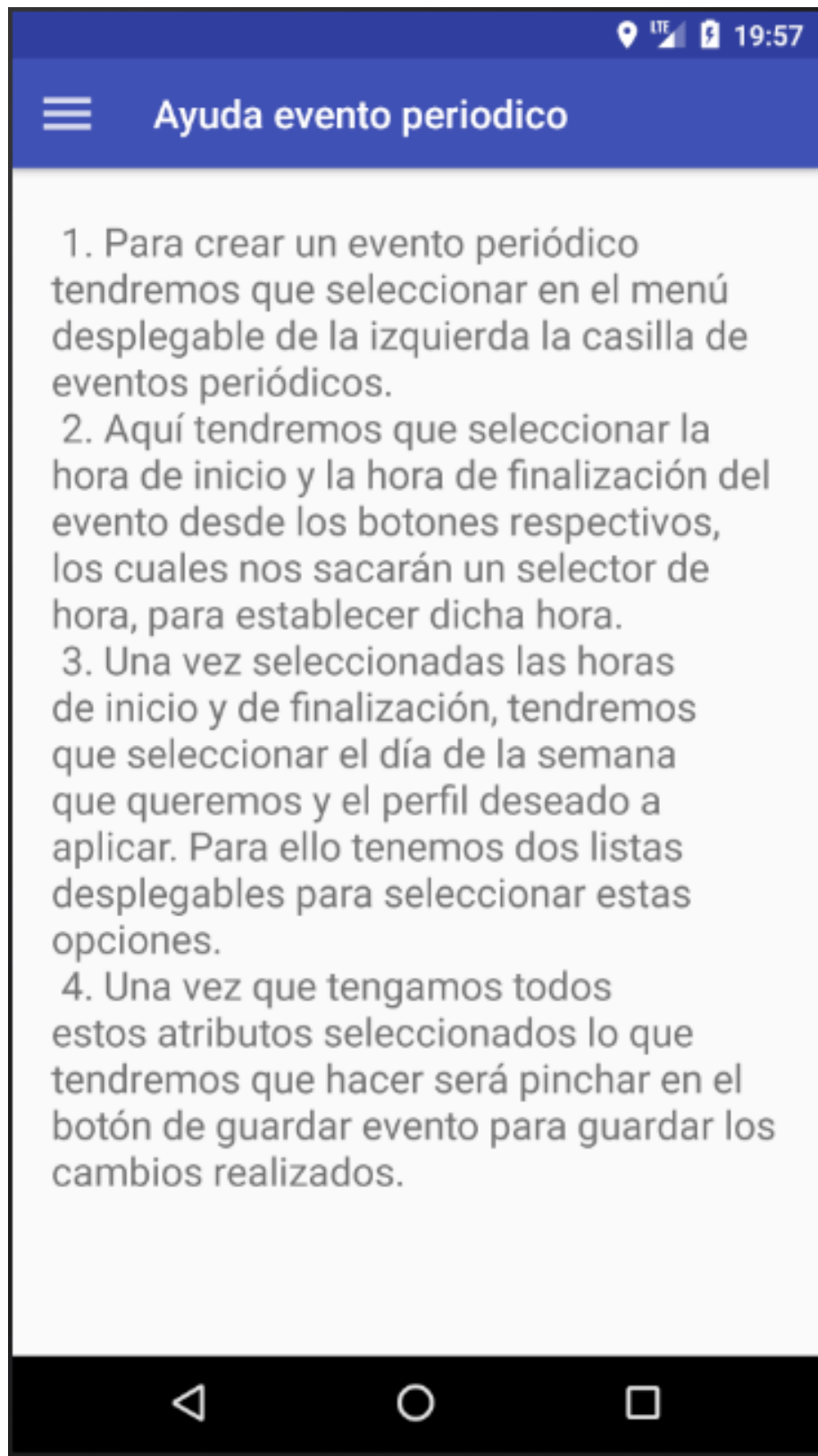


Figura E.14: Imagen que muestra la pantalla para la creación de eventos periódicos.

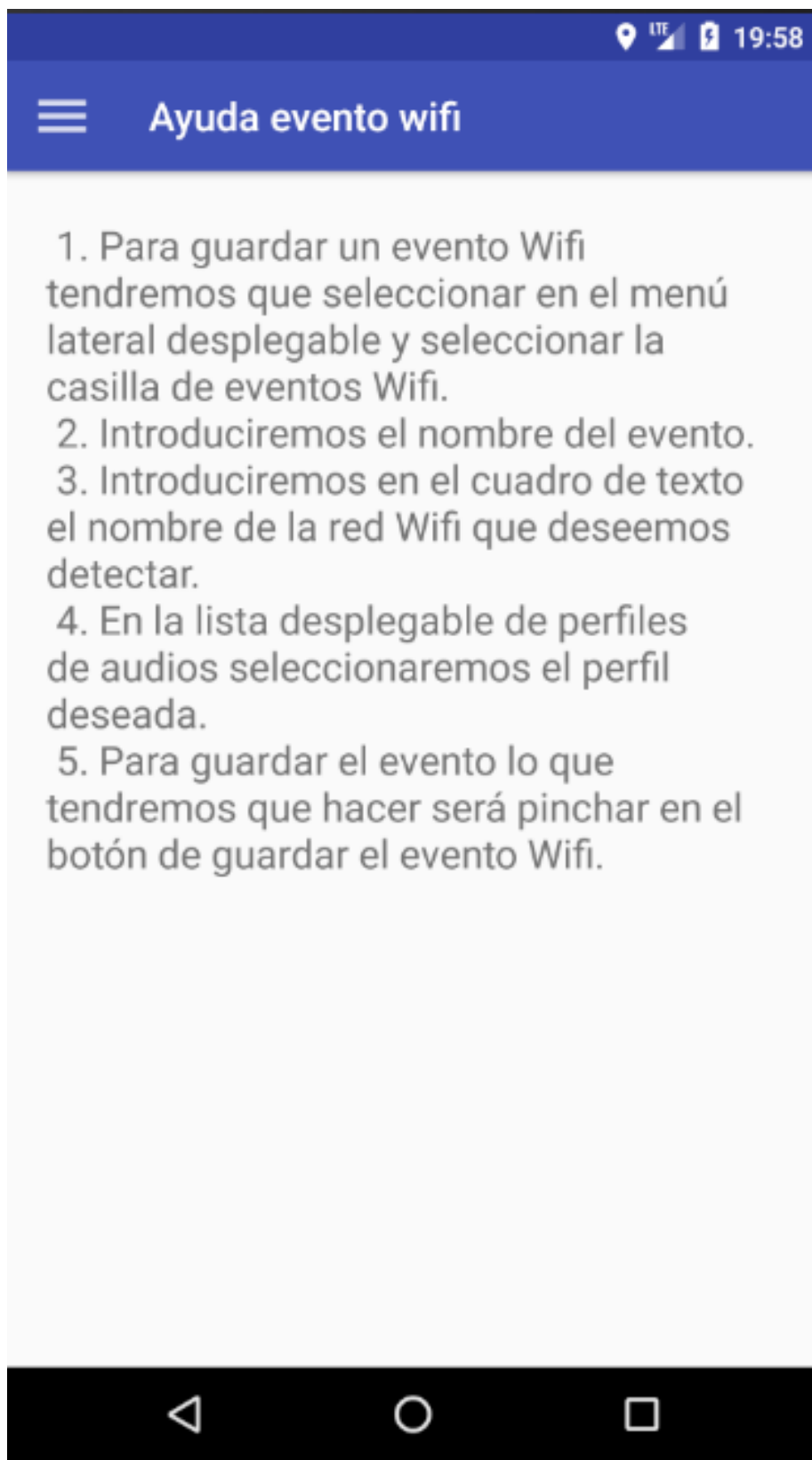


Figura E.15: Imagen que muestra la pantalla para la creación de eventos wifi.

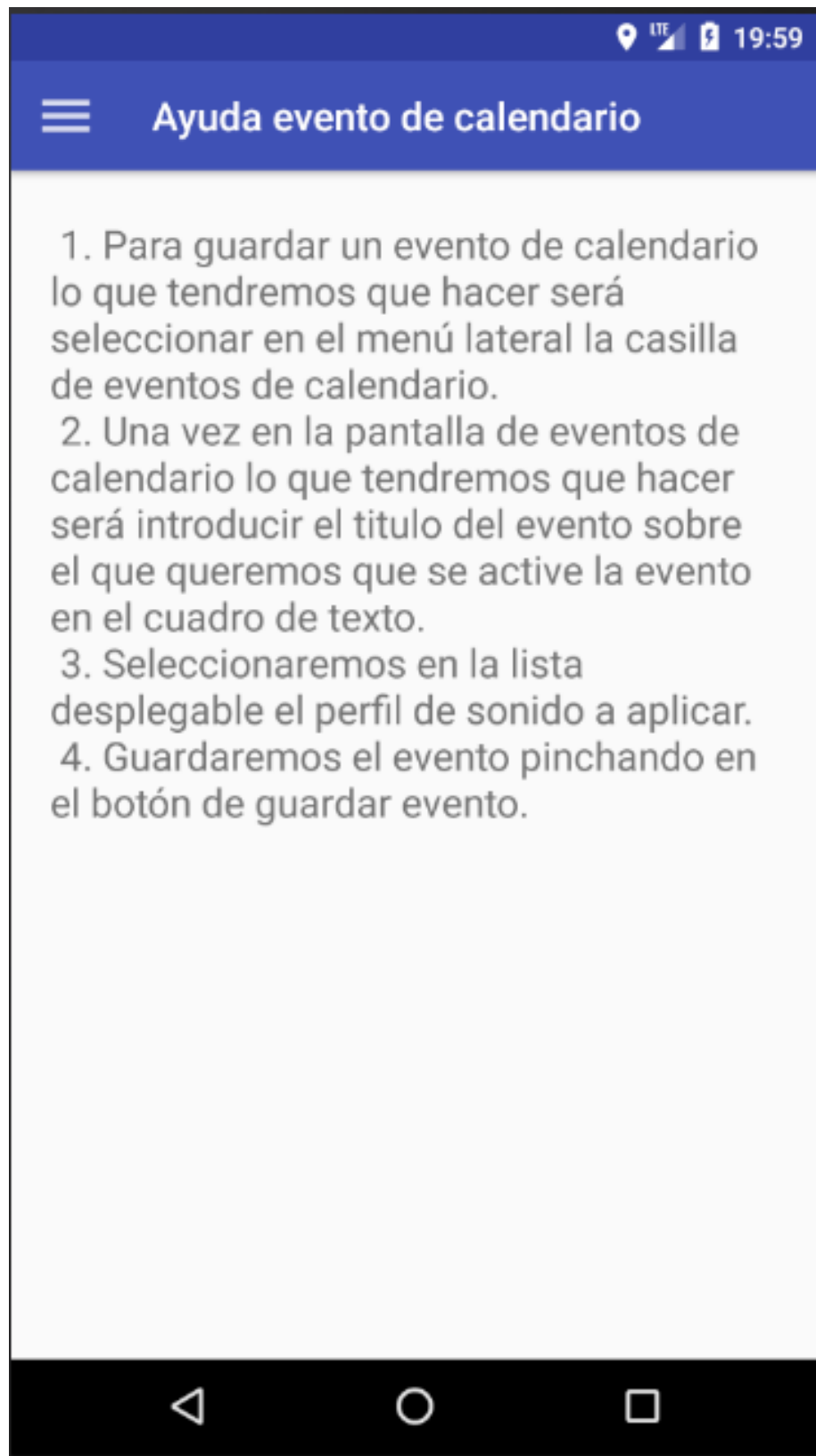


Figura E.16: Imagen que muestra la pantalla para la creación de eventos de calendario.

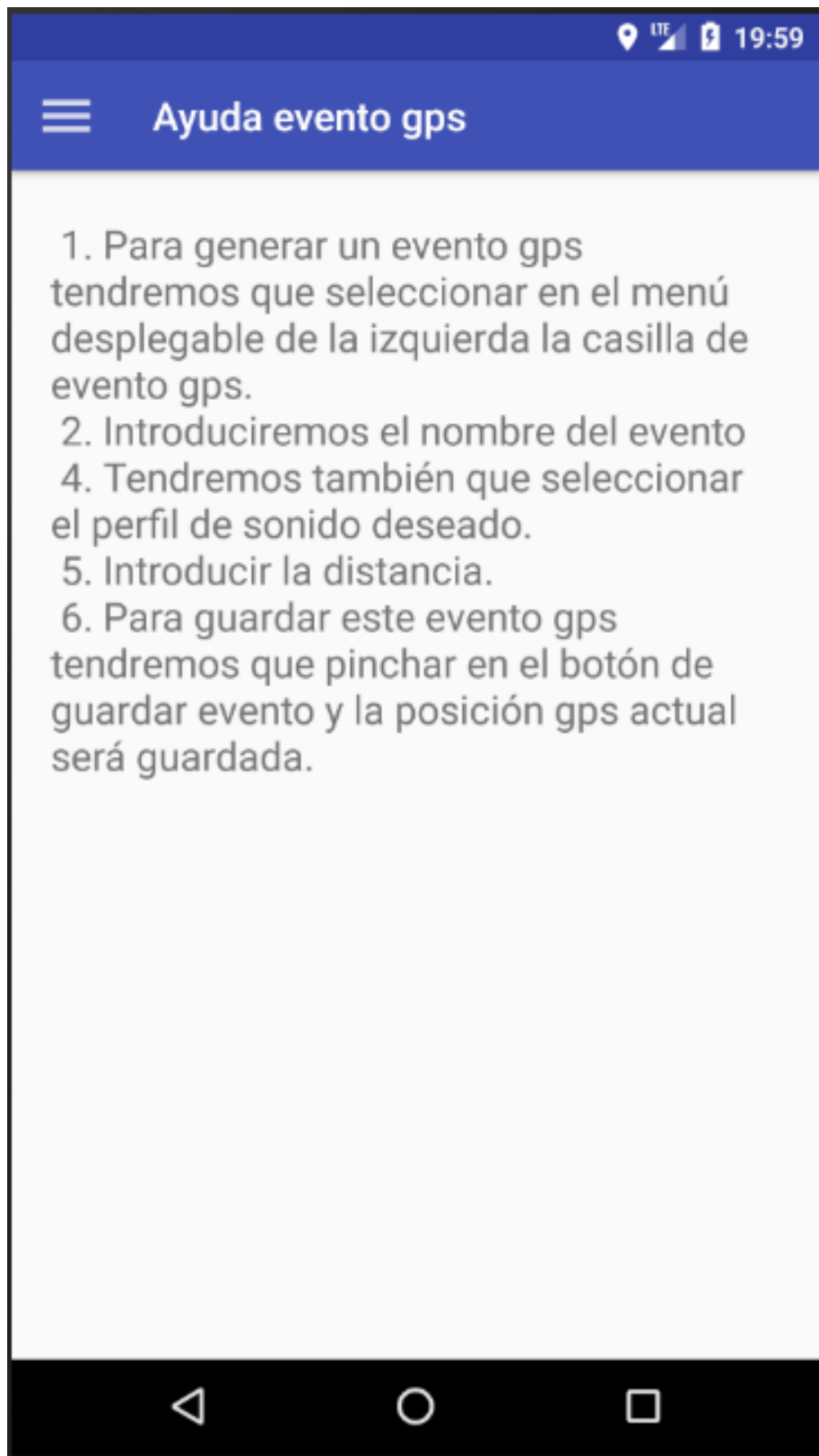


Figura E.17: Imagen que muestra la pantalla para la creación de eventos gps.

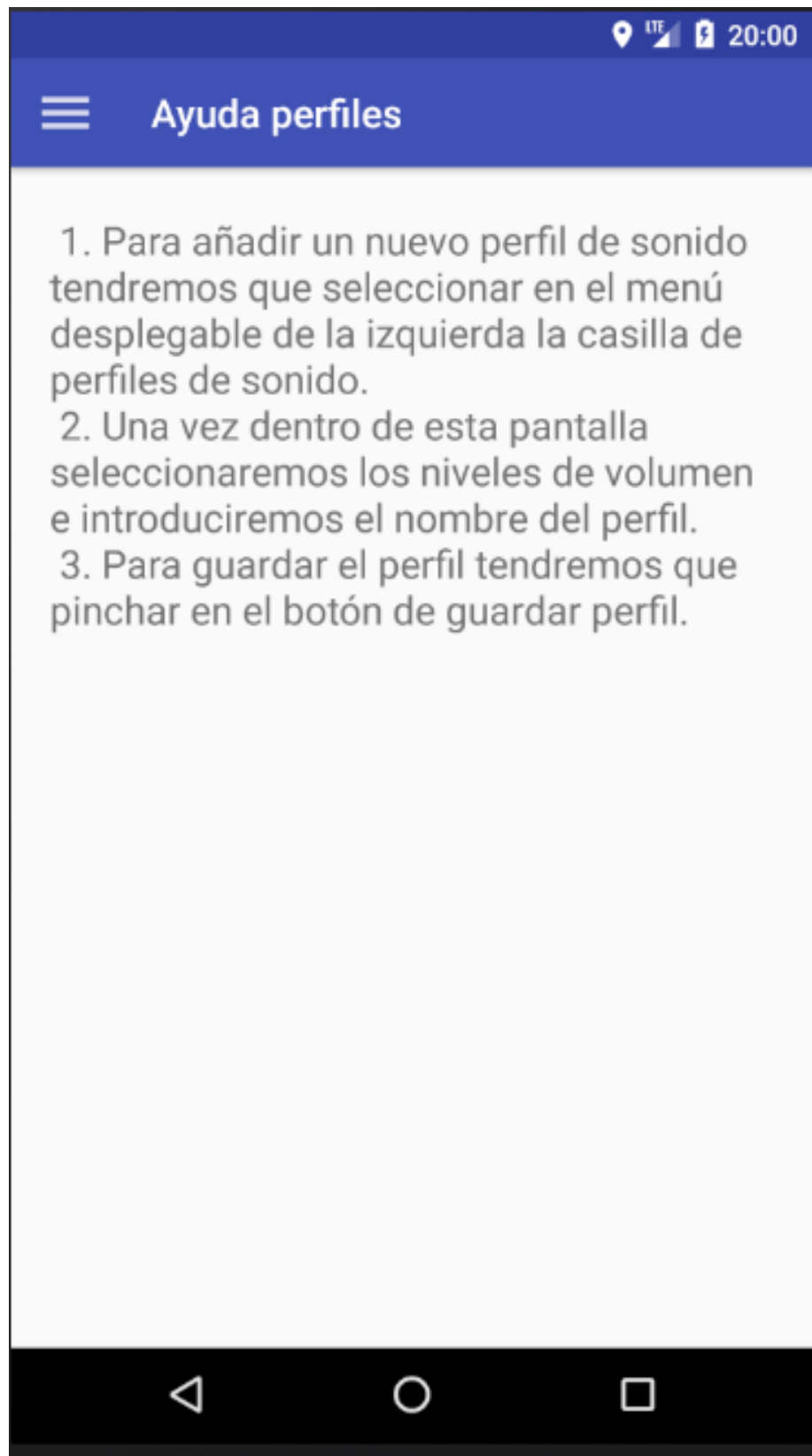


Figura E.18: Imagen que muestra la pantalla para la creación de perfiles de sonido.

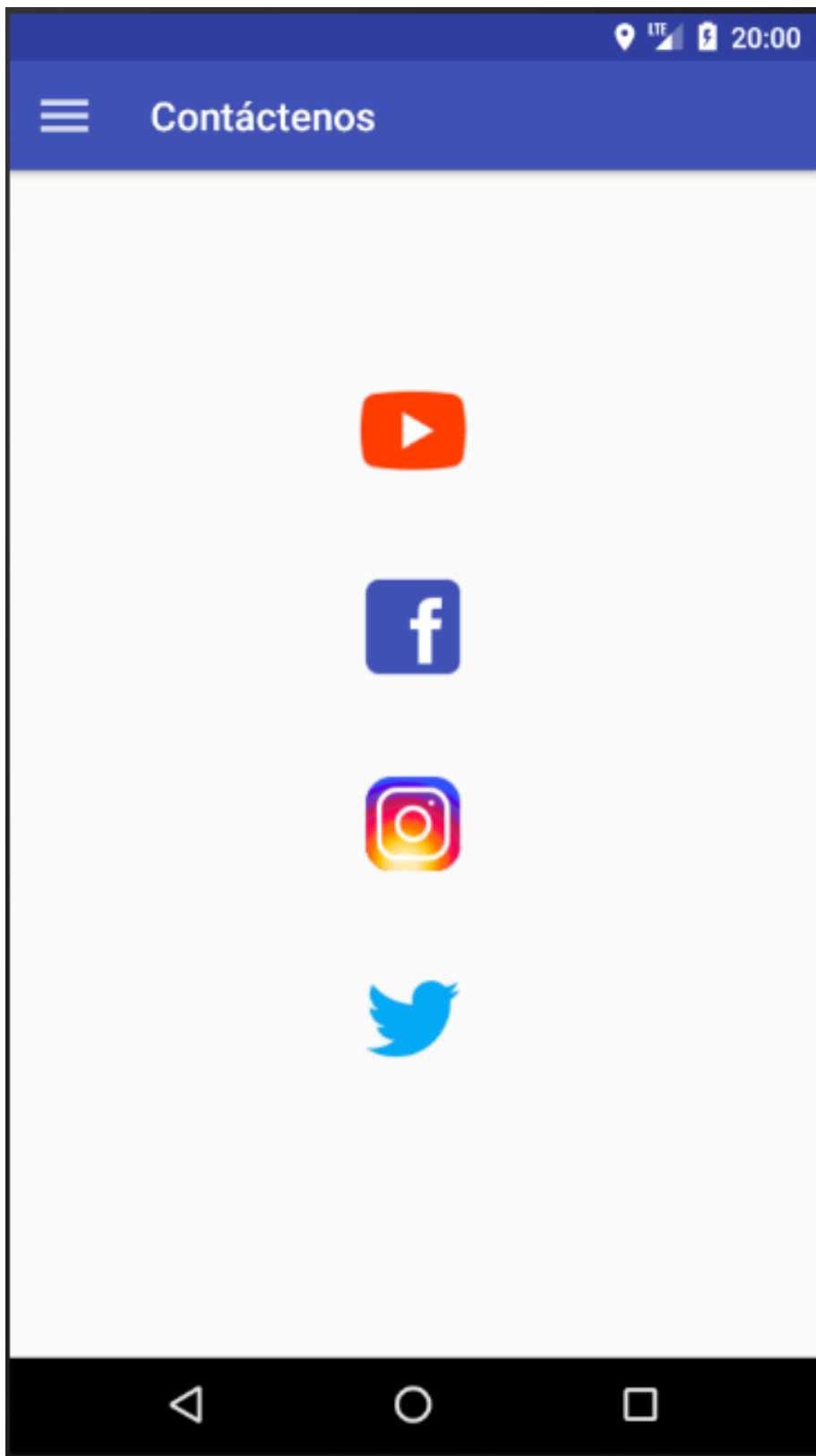


Figura E.19: Imagen que muestra la pantalla para el acceso a las diferentes redes sociales y poder establecer contacto para obtener soporte de la aplicación.





---

## Bibliografía

---

- [1] Codacy.
- [2] Download Android Studio and SDK tools.
- [3] git - the simple guide - no deep shit!
- [4] 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.