



FACULTAD DE CIENCIA Y TECNOLOGÍA
ESCUELA DE INGENIERÍA ELECTRÓNICA

**Diseño e implementación de software para la gestión de
calidad de los servicios de telecomunicaciones en
dispositivos móviles**

**Trabajo de graduación previo a la obtención del título de:
INGENIERO ELECTRÓNICO**

Autor:

PATIÑO JARA OSWALDO RICARDO

Director:

CÓRDOVA OCHOA JUAN PATRICIO

CUENCA, ECUADOR

2016

DEDICATORIA

Este trabajo lo dedico a mi familia por apoyarme y guiarme en todo momento, ante cualquier circunstancia de la vida, en donde he aprendido a caminar y crecer con verdaderos valores.

A mi esposa: por ser mi fortaleza; por su amor, su comprensión y su maravilloso empeño que han sido las bases para cumplir este proyecto.

A mi hija: por ser el motor y la inspiración de todos mis esfuerzos y por enseñarme a valorar el tiempo de ser padre.

ÍNDICE DE CONTENIDOS

DEDICATORIA	ii
ÍNDICE DE CONTENIDOS	iii
ÍNDICE DE FIGURAS	vi
ÍNDICE DE TABLAS	viii
ÍNDICE DE ANEXOS	ix
RESUMEN.....	x
ABSTRACT	xi
INTRODUCCIÓN	1
CAPÍTULO I: MARCO TEORICO	2
1.1. Introducción.....	2
1.2. Objetivos	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	2
1.3. Plataforma	3
1.4. Entorno y Herramientas.....	3
1.5. Metodología.....	6
1.6. Conclusiones	7
CAPÍTULO II: DISEÑO DEL SOFTWARE.....	8
1.7. Introducción.....	8

1.8.	Análisis del software	8
1.8.1.	Entorno de desarrollo	8
1.8.2.	Establecimiento de los parámetros QoS que se utilizarán para la gestión	9
1.9.	Diseño y arquitectura.....	12
1.9.1.	Diagramación y estructuración del código	12
1.9.2.	Diseño de la interfaz gráfica.....	14
1.9.3.	Diseño de las clases, capas y patrones de funcionamiento para la comunicación y adquisición de la información de los servicios.....	15
1.10.	Conclusiones	19
 CAPÍTULO III: DESARROLLO DEL SOFTWARE		20
1.11.	Introducción.....	20
1.12.	Desarrollo funcional de la aplicación. Prototipo	20
1.13.	Desarrollo de la interfaz gráfica	22
1.14.	Depuración, simulaciones y pruebas en terminales Android.	25
1.14.1.	Internet	26
1.14.2.	Telefonía móvil.....	26
1.15.	Conclusiones	27
 CAPÍTULO IV: PUBLICACIÓN DE LA APLICACIÓN ANDROID E IMPLEMENTACIÓN		28
1.16.	Introducción.....	28
1.17.	Preparación para el despliegue del software.	28
1.17.1.	Firma Digital.....	28
1.17.2.	Limpieza de código y de archivos del proyecto	29

1.17.3.	Configuración de construcción de la versión.....	29
1.18.	Instalación del software.....	29
1.19.	Publicación en Google Play Store.....	29
1.19.1.	Registro de una cuenta en Google Play para desarrolladores.....	30
1.19.2.	Consola para Desarrolladores de Google Play.....	30
1.19.3.	Publicaciones Alfa y Beta.....	31
1.20.	Lanzamiento del software y sondeo en tiempo real.....	37
1.20.1.	Datos de conexión:.....	37
1.20.2.	Drive-test.....	40
1.21.	Conclusiones.....	42
 CAPÍTULO V: MANUAL DE USUARIO		43
1.22.	Introducción.....	43
1.23.	Módulo de Pruebas.....	45
1.24.	Módulo de Localización.....	47
1.25.	Módulo de Información de Red.....	49
1.26.	Módulo de Reporte de Fallos.....	50
1.27.	Conclusiones.....	51
CONCLUSIONES		52
BIBLIOGRAFÍA		54
ANEXOS		55

ÍNDICE DE FIGURAS

Figura 2.2.1 Diagramación de paquetes. Entidad-Vistas-Servicio	14
Figuras 2.2, 2.3 y 2.4 Diseños de vistas Principal, Internet y Listado de llamadas/sms	14
Figuras 2.5 y 2.6 Diseño de vistas para Pruebas Generales y Mapa de Localización	15
Figuras 3.3.1 y 3.3.2 Servicio Iniciado y Llamada Entrante	21
Figuras 3.3.3 y 3.3.4 Llamada en curso y Llamada finalizada.....	21
Figuras 3.3.5 y 3.3.6 Interacción de funciones y Resultados de conectividad.....	22
Figura 3.3.7 Interfaz principal.....	23
Figura 3.3.8 Interfaz de Información de Red	23
Figura 3.3.9 Interfaz de Localización	24
Figura 3.3.10 Interfaz de Información	25
Figura 4.4.1 Consola de Desarrolladores de Google Play	31
Figura 4.4.2 Menú de selección para versión de APK.....	32
Figura 4.4.3 Cuadro de diálogo para la carga de APK.....	32
Figura 4.4.4 Estado de la versión del APK	33
Figura 4.4.5 Formulario de información de la versión.....	34
Figura 4.4.6 Formulario para la carga de capturas de pantalla de la aplicación	34
Figura 4.4.7 Formulario de información para establecer la calificación de la versión	35
Figura 4.4.8 Información del estado de la aplicación publicada.....	36
Figura 4.4.9 Visualización de la aplicación subida al Google Play Store.....	36
Figura 4.4.10 Estado de la versión de prueba y acciones a ejecutar	37
Figura 4.4.11 Habilidad del seguimiento de localización para el drive-test	40
Figura 5.1. Interfaz Principal. Botones de navegación	44

Figura 5.2. Botones de acción y barra de herramientas	44
Figura 5.3. Pantalla de información de la aplicación	45
Figura 5.4. Módulo de Pruebas. Botones de navegación y pruebas.....	46
Figuras 5.5. y 5.6. Resultados de la conectividad a la red WiFi y Resultados de la conectividad a Internet	46
Figuras 5.7. y 5.8. Resultados de los registros de llamadas y Resultados de los registros de SMS	47
Figura 5.9. Interfaz de Geolocalización. Navegación y despliegue de información..	48
Figura 5.10. Activación del drive-test.....	48
Figura 5.11. Ejemplo de drive-test usando la aplicación	49
Figura 5.12. Interfaz de la Información de Red. Desplazamiento para el ajuste de la pantalla	49
Figura 5.13. Detalle de la información del dispositivo y de la red a la cual está conectado.....	50

ÍNDICE DE TABLAS

Tabla 1.1.1 Porcentaje de número de dispositivos por versión de Android.....	5
Tabla 2.1. Valor Agregado de Internet.....	9
Tabla 2.2. Telefonía móvil.....	10
Tabla 4.4.1 Datos obtenidos de dispositivo Samsung Galaxy S3	38
Tabla 4.4.2 Datos obtenidos de dispositivo Samsung Galaxy S3 mini.....	38
Tabla 4.4.3 Datos obtenidos de dispositivo Sony Xperia M.....	39
Tabla 4.4.4 Datos obtenidos de dispositivo Samsung Galaxy J7.....	39
Tabla 5.1 Ejemplo de Reporte de Fallos	51

ÍNDICE DE ANEXOS

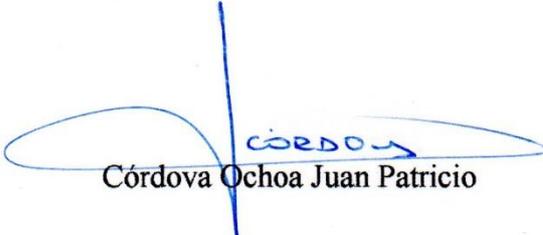
Anexo 1: Drive-Test, Ruta N° 1.....	55
Anexo 2: Drive-Test, Ruta N° 2.....	56
Anexo 3: Drive-Test, Ruta N° 3.....	57
Anexo 4: Drive-Test, Ruta N° 4.....	57

DISEÑO E IMPLEMENTACIÓN DE SOFTWARE PARA LA GESTIÓN DE CALIDAD DE LOS SERVICIOS DE TELECOMUNICACIONES EN DISPOSITIVOS MÓVILES

RESUMEN

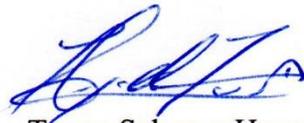
El proyecto consiste en diseñar un software para dispositivos móviles, capaz de realizar una medición de los servicios de telecomunicaciones, en base a las normas que definen los parámetros de calidad de servicio, incluyendo el lugar geográfico en donde se realiza el sondeo. Con este objetivo, se ha aplicado una ingeniería de software para estructurar el diseño y desarrollarlo en el sistema operativo Android, previo estudio del uso de los módulos de comunicación y gestión de los servicios disponibles. Posteriormente se publicará el software para implementarlo abiertamente, obteniendo así la información directamente desde los usuarios.

Palabras Clave. Software, QoS, telecomunicaciones, aplicaciones Android.



Córdova Ochoa Juan Patricio

Director de Trabajo de Titulación



Torres Salamea Hugo Marcelo

Director de Escuela



Patíño Jara Oswaldo Ricardo

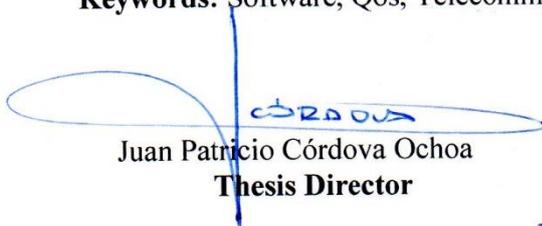
Autor

**DESIGN AND IMPLEMENTATION OF SOFTWARE FOR THE QUALITY
MANAGEMENT OF TELECOMMUNICATION SERVICES IN MOBILE
DEVICES**

ABSTRACT

This project deals with the design of software for mobile devices, capable of performing measurement of telecommunication services, based on standards that define quality service parameters, including the geographic location where the survey is conducted. Based on this objective, software engineering has been implemented so as to structure the design and develop it in the Android operating system. Prior to this, a study of the use of communication and management modules of available services was carried out. Then, the software will be released in order to implement it openly; thus obtaining information directly from users.

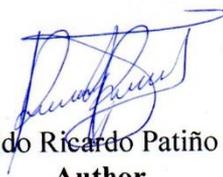
Keywords: Software, Qos, Telecommunications, Android Applications.



Juan Patricio Córdova Ochoa
Thesis Director



Hugo Marcelo Torres Salamea
School Director



Oswaldo Ricardo Patiño Jara
Author



UNIVERSIDAD DEL
AZUAY
Dpto. Idiomas



Translated by,
Lic. Lourdes Crespo

Patiño Jara Oswaldo Ricardo

Trabajo de Titulación

Ing. Córdova Ochoa Juan Patricio. M.Sc.

Noviembre, 2016

DISEÑO E IMPLEMENTACIÓN DE SOFTWARE PARA LA GESTIÓN DE CALIDAD DE LOS SERVICIOS DE TELECOMUNICACIONES EN DISPOSITIVOS MÓVILES

INTRODUCCIÓN

En la actualidad existen varias formas de realizar la medición de los parámetros que definen los estándares de calidad de los servicios de telecomunicaciones, pero en la mayoría se requiere hacer un sondeo de campo con equipos propios de cada empresa, lo que se conoce como el *drive-test*. Lo que se pretende con este proyecto es facilitar esa medición de forma interactiva y objetiva mediante una aplicación para dispositivos móviles, obteniendo así los datos directamente desde los usuarios. Debido a las limitaciones y la variación de la gama tecnológica de los dispositivos no es posible realizar la evaluación de todos los parámetros necesarios para la gestión de calidad, por lo tanto, se estudiarán los parámetros clave que se pueden obtener en la mayoría de usuarios con dispositivos Android, plataforma que se ha escogido por tener el mayor número de usuarios de dispositivos móviles. Android es una plataforma de software de código abierto desarrollada para la abstracción de una gran variedad de dispositivos independientemente del hardware, pero guiados por un estándar de compatibilidad establecido, de esta forma, permitir la innovación del desarrollo para brindar una mejor experiencia a los usuarios de dispositivos móviles. Por lo tanto, se consideró viable y propicio el desarrollo del proyecto en base a esta plataforma.

CAPÍTULO I

MARCO TEORICO

1.1 Introducción

El propósito de este capítulo es el de cubrir todos los conceptos básicos y esenciales que serán usados en materia del presente proyecto además de los objetivos.

Principalmente se explicarán: plataformas, herramientas, metodologías y definiciones. Esto nos servirá como base para el desarrollo y el establecimiento del alcance del proyecto.

1.2 Objetivos

1.2.1 Objetivo General

Integrar un software en los dispositivos móviles para la gestión de calidad de los servicios de telecomunicaciones a nivel nacional.

1.2.2 Objetivos Específicos

El desarrollo del proyecto se realizó usando la metodología de la ingeniería de software: análisis, diseño, codificación y pruebas, usando un desarrollo incremental el cual será llevado a cabo en los objetivos específicos:

- a) Diseño de un software capaz de explotar las herramientas de los dispositivos móviles para la evaluación del estado de los servicios de telecomunicaciones.
- b) Implementación de la aplicación en diferentes dispositivos Android y obtención de los datos cualitativos de los servicios prestados.

1.3 Plataforma

La plataforma de desarrollo es el sistema base sobre el cual se va a asentar el software. Este tema es muy importante debido a que teniendo una plataforma definida se puede establecer que herramientas y que limitaciones se van a presentar en el desarrollo del software.

La plataforma seleccionada para este proyecto es Android. Android es una plataforma de software de código abierto creado para una amplia gama de dispositivos con diferentes factores de forma. Los propósitos principales de Android son la creación de una plataforma de software abierto, disponible para los desarrolladores, para hacer que sus ideas innovadoras se conviertan en realidad y para introducir un producto de éxito, en el mundo real que mejora la experiencia móvil para los usuarios.

1.4 Entorno y Herramientas

En la actualidad existen varios lenguajes en los cuales se puede desarrollar las aplicaciones Android, pero se ha escogido Java, que es el lenguaje con el que inicialmente fue estructurada la interfaz de programación (API), además de ser el más amigable en interacción con la pila estructural de la plataforma Android. Para la facilidad en la construcción del proyecto se ha decidido, en primera instancia, trabajar con el software de programación Eclipse, que ha tenido bastante experiencia en el desarrollo de aplicaciones en base a Java y con el complemento provisto por Google, ADT, permite la fácil elaboración de proyectos para aplicaciones Android.

Sin embargo, ha surgido con fuerza otra herramienta para la construcción de aplicaciones Android que, además es sugerido por el mismo Google y oficializado como el *Android IDE*, debido a que tiene mayores facilidades, funcionalidades y adaptabilidad para la creación, diseño y construcción de aplicaciones Android. Este software es el Android Studio y es de libre descarga y uso, por lo que se realizó una migración del proyecto inicialmente creado en Eclipse para poder usarlo con Android Studio. (Android, 2015)

Posterior a la selección del ambiente de desarrollo, es necesario conocer los paquetes de la API de Android a ser usados para la codificación. De acuerdo a las necesidades del proyecto se utilizarán los siguientes paquetes:

- Localización (android.location)
- Redes (android.net)
- Contenido (android.content y android.provider)
- Servicios (android.app.Service)
- Telefonía (android.telecom y android.telephony)
- Interfaz (android.os, android.view y android.support).

Es importante señalar estos paquetes ya que después los relacionaremos con los KPI a usar para la medición de la calidad de los servicios.

Otro aspecto importante que definir es la versión del sistema operativo a la cual va a apuntar nuestro software. Como resultado de la constante evolución de la plataforma Android, la interfaz de programación a disposición va cambiando, por esta razón y de acuerdo al contexto actual es importante construir la aplicación para que sea compatible con la mayoría de dispositivos móviles Android, que según la información provista en la página de *Android Developers* la mayoría de dispositivos en la actualidad tienen desde la versión *Jelly Bean* en adelante. Según esta información y en base a un estudio superficial de las necesidades del proyecto en cuanto a codificación se ha decidido tomar como versión mínima la de *4.1.X, API 16*. Como se puede apreciar en las siguientes ilustraciones:

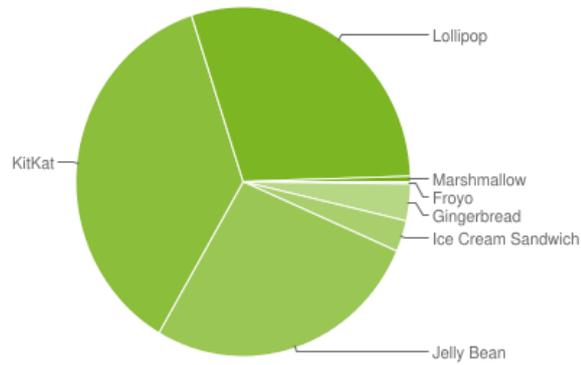


Figura 1.1 Relación de número de dispositivos por versión de Android

Fuente: (Android, 2015).

Tabla 1.1.1 Porcentaje de número de dispositivos por versión de Android

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.9%
4.1.x		16	10.0%
4.2.x	Jelly Bean	17	13.0%
4.3		18	3.9%
4.4	KitKat	19	36.6%
5.0		21	16.3%
5.1	Lollipop	22	13.2%
6.0	Marshmallow	23	0.5%

Fuente: (Android, 2015).

1.5 Metodología

La metodología a usar para este tipo de proyectos es la Ingeniería de Software, que se refiere al conjunto de técnicas para el correcto estudio, desarrollo y mantenimiento de sistemas de software.

La Ingeniería de Software consta de las siguientes etapas: Análisis, Diseño y Arquitectura, Desarrollo, Pruebas y Mantenimiento.

- Análisis. - Consiste en extraer la información necesaria para definir los requisitos que va a cubrir la aplicación. Es una etapa fundamental, ya que permite establecer la estructura del software.
- Diseño y Arquitectura. - Se refiere a la determinación de la funcionalidad del sistema a ser desarrollado, transformando el análisis previo en modelos, entidades, casos de uso, etc., que son de gran importancia para facilitar la construcción del software.
- Desarrollo (Programación). - Esta etapa es la más extensa, consiste en transformar el diseño y la arquitectura en código de programación; está directamente relacionada con las etapas de diseño y de pruebas. Una arquitectura estrictamente elaborada facilitará el desarrollo del sistema de software.
- Pruebas. - Se refiere a la verificación del funcionamiento del software por medio de tareas especificadas, generalmente llamados módulos, esto agiliza la resolución de errores puesto que la división del sistema permite un enfoque de cada funcionalidad. Una vez corregidos todos los errores se procede a integrar todos los módulos en el sistema completo. Esta etapa conjuntamente con la de Desarrollo se suele repetir varias veces hasta conseguir el software deseado.
- Mantenimiento. - Es una etapa de la ingeniería de software que usualmente se usa para el perfeccionamiento del software mediante la corrección de errores no previstos, así como el tratamiento de nuevos requisitos. Para esto es importante brindar al cliente un medio de comunicación accesible, de fácil uso y de respuesta inmediata.

1.6 Conclusiones

Como conclusiones de este capítulo se puede acotar que el tener claramente definidos el entorno, las definiciones y una buena metodología de desarrollo se llevar a cabo de la mejor manera el proyecto previsto.

La ingeniería de software es una metodología bastante útil para el desarrollo de aplicaciones, una vez establecido claramente el contexto sobre el cual trabajar, el desarrollo del sistema se lo puede realizar sin dificultades.

CAPÍTULO II

DISEÑO DEL SOFTWARE

2.1 Introducción

En el presente capítulo se explicará la metodología y el ciclo de vida que se utiliza en la ingeniería de software para el desarrollo de aplicaciones. Además, se abarcará las variables involucradas, así como las herramientas y recursos a utilizar para el desarrollo. Considerando las facilidades que presta la plataforma Android, sistema operativo escogido por la gran cantidad de usuarios y dispositivos, es factible la realización del trabajo en este entorno para conseguir el cumplimiento de los objetivos planteados. De tal forma que se realizará el respectivo análisis y diseño en base a esta plataforma considerando las limitaciones que esto implica.

2.2 Análisis del software

Es el primer paso del ciclo de la ingeniería de software. Consiste en realizar una evaluación de las necesidades que requiere la aplicación, de acuerdo a las características que presenten mayor adaptabilidad, eficacia y rendimiento del software, delimitando así, el alcance del funcionamiento de la aplicación para la meta propuesta.

2.2.1 Entorno de desarrollo

El entorno de desarrollo escogido es: como sistema operativo se ha definido Android; lenguaje de programación para el desarrollo se escogió Java, ya que es base de la plataforma; como herramienta de desarrollo se ha seleccionado *Android Studio*, debido a que facilita el desarrollo de interfaces y tiene esquemas básicos desarrollados como guía.

2.2.2 Establecimiento de los parámetros QoS que se utilizarán para la gestión

En la actualidad, el Ecuador cuenta con organismos responsables para la gestión de la calidad de los servicios de telecomunicaciones: MINTEL, organismo coordinador; CONATEL, organismo regulador; y SUPERTEL, organismo de control; cuya obligación es mantener los servicios que prestan los proveedores en la mejor calidad posible para los clientes, además de establecer las normativas sobre las cuales se regirá la evaluación de dicha información.

El CONATEL es el ente encargado de establecer los parámetros para la calidad del servicio (QoS), dentro de un sistema de regulación adecuado para las necesidades de los usuarios que, en este caso, para los dispositivos móviles aplican el de Valor Agregado de Internet mediante WiFi y el de Telefonía Móvil Avanzado (SMA). Los KPI actualmente vigentes son:

Tabla 2.1. Valor Agregado de Internet

Relación con el cliente
Porcentaje de reclamos generales
Tiempo promedio de resolución de reclamos generales
Porcentaje de reclamos de facturación
Tiempo promedio de reparación de averías
Porcentaje de módems utilizados
Porcentaje de reclamos por la capacidad del canal de acceso contratado

Fuente: (CONATEL, 2009)

Tabla 2.2. Telefonía móvil

Relación con el cliente
Porcentaje de reclamos generales
Tiempo promedio de resolución de reclamos generales
Tiempo promedio de espera de respuesta de operador humano
Porcentaje de reclamos de facturación y débito
Porcentaje de llamadas establecidas
Tiempo de establecimiento de llamadas
Porcentaje de llamadas caídas
Zona de cobertura
Calidad de conversación
Porcentaje de mensajes cortos con éxito
Tiempo promedio de entrega de mensajes cortos

Fuente: (MINTEL, 2014)

Todos estos KPI son seleccionados de acuerdo a las normas ETSI que rige la UIT (Unión Internacional de Telecomunicaciones), que es un organismo internacional encargado de permitir el desarrollo de las redes de telecomunicaciones y de la información facilitando la equidad de acceso a nivel mundial.

En base a estos parámetros vigentes, vamos a definir los que la aplicación va a ser capaz de medir. La API de Android dispone de los paquetes necesarios para acceder al contenido de los dispositivos móviles (llamadas, mensajería, conectividad, etc.) lo que facilita obtener la información necesaria para medir los parámetros QoS antes mencionados. Debido a que Android es una plataforma abierta al desarrollo, Google determinó ciertas limitantes para el desarrollo de las aplicaciones en cuanto a contenido se refiere, esto es lógico ya que se trata de la información personal de cada usuario. Es por esto que no se puede gestionar todos los KPI definidos, entonces consideramos para el proyecto los siguientes:

Valor Agregado de Internet

- Porcentaje de reclamos generales

$$\%Rg = \frac{\text{Número de reclamos}}{\text{Líneas activas}} \times 100 \leq 2\%$$

- Porcentaje de reclamos de facturación

$$\%Rf = \frac{\text{Número de reclamos}}{\text{Líneas activas}} \times 100 \leq 2\%$$

- Porcentaje de reclamos por la capacidad del canal de acceso contratado

$$\%Rc = \frac{\text{Número de reclamos}}{\text{Líneas por canal}} \times 100 \leq 2\%$$

Telefonía móvil

- Porcentaje de reclamos generales

$$\%Rg = \frac{\text{Número de reclamos}}{\text{Líneas activas}} \times 100 \leq 1.5\%$$

- Porcentaje de reclamos de facturación y débito

$$\%Rf = \frac{\text{Número de reclamos}}{\text{Líneas activas}} \times 100 \leq 0.5\%$$

- Porcentaje de llamadas establecidas

$$\%Lle = \frac{\text{Llamadas exitosas}}{\text{Número total de intentos}} \times 100 \geq 95\%$$

- Tiempo de establecimiento de llamadas

$$\%Ce = \frac{\text{Llamadas establecidas} < 12 \text{ seg}}{\text{Número total de llamadas establecidas}} \times 100 \geq 95\%$$

- Porcentaje de llamadas caídas

$$\%Llc = \frac{\text{Llamadas caídas}}{\text{Número total de llamadas establecidas}} \times 100 :$$

$$A \leq 2\%$$

$$B \leq 5\%$$

$$C \leq 7\%$$

en donde A, B y C son los tipos de celda.

- Zona de cobertura

$$\%C = \frac{\text{Número de muestras dentro de los niveles (Ns)}}{\text{Número total muestras válidas}} \times 100 :$$

Urbano $\geq 95\%$, para:

Ns (GSM) ≥ -85 dBm

Ns (CDMA/UMTS) ≥ -14 dB

Rural $\geq 90\%$, para:

Ns (GSM) ≥ -98 dBm

Ns (CDMA/UMTS) ≥ -17 dB

- Porcentaje de mensajes cortos con éxito

$\%SMS \leq 95\%$

La aplicación va a poner a disposición toda la información requerida para los parámetros QoS definidos y será cada organización reguladora y de control que se encargue del tratamiento de los datos para la evaluación de los KPI.

2.3 Diseño y arquitectura

En este proceso se determinará el funcionamiento del software tomando en cuenta los alcances y las limitaciones que tendrá de acuerdo al previo análisis realizado. Se definirá las entidades y las clases de diseño para llegar a formar el prototipo que nos servirá de guía para la versión final de la aplicación.

2.3.1 Diagramación y estructuración del código

Para obtener una mejor apreciación de cómo va a funcionar el software es necesario diseñar un diagrama UML en donde se podrá visualizar las funciones repartidas en casos de uso para las respectivas funciones y los alcances de la aplicación.

La estructuración se la va a realizar en paquetes para distinguir cada uno de los diferentes roles que va integrar cada grupo de clases o vistas según la programación orientada a objetos. Los paquetes a utilizar se clasificarán en los siguientes casos de uso:

2.3.1.1 Servicio

Constará de una sola clase encargada de mantener un servicio para la obtención de toda la información del dispositivo, de acuerdo a los parámetros QoS establecidos. Este servicio permanecerá activo aún si la aplicación no se esté utilizando en primer plano por el usuario, permitiendo la disponibilidad de la información para un servidor que, en este caso dependería de la organización regulador y de control.

2.3.1.2 Entidades

Conformarán las clases que incluirán toda la información de los servicios a gestionar divididos en dos, como se indicó en el análisis: Internet (WiFi) y Telefonía (Cell), siendo esta última subdividida en dos: Llamadas (Call) y Mensajería (Sms). Con las entidades definidas se puede personalizar la forma de obtener la información que el dispositivo pone a disposición para cada caso, esto permitirá tener una estructura más ordenada para la visualización de los parámetros necesarios para la evaluación de la calidad.

2.3.1.3 Vistas

Se incluirá todas las clases de la interfaz de usuario, en Android se llaman Actividades, en donde se visualizará los parámetros QoS obtenidos del Servicio y estructurados por las entidades. Es lo que se va a mostrar al usuario.

2.3.1.4 Principal

Es la Actividad que servirá para iniciar la aplicación, en ésta se configurarán todas las bases para el uso de los demás paquetes.

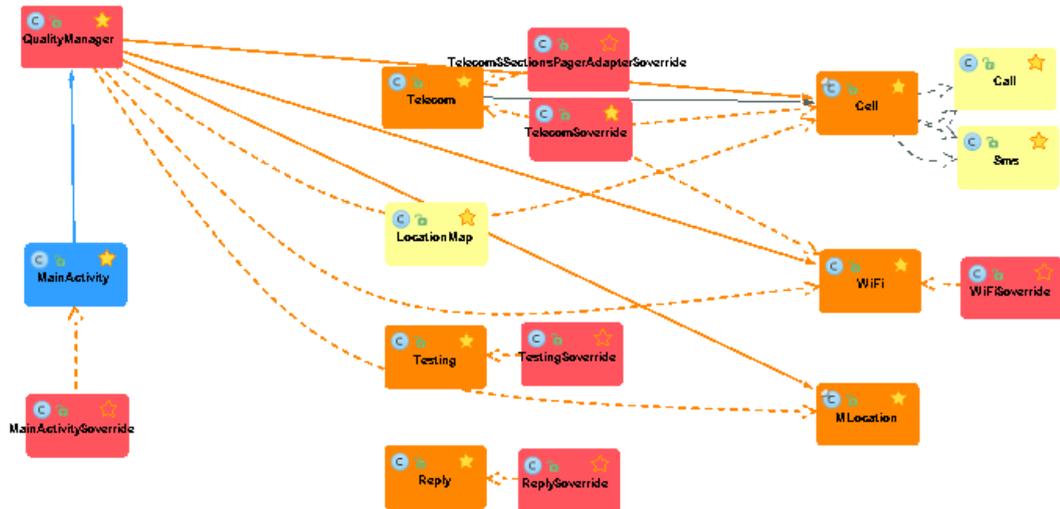


Figura 2.2.1 Diagramación de paquetes. Entidad-Vistas-Servicio

Fuente: Diagrama generado por la herramienta CodeIris de Android Studio.

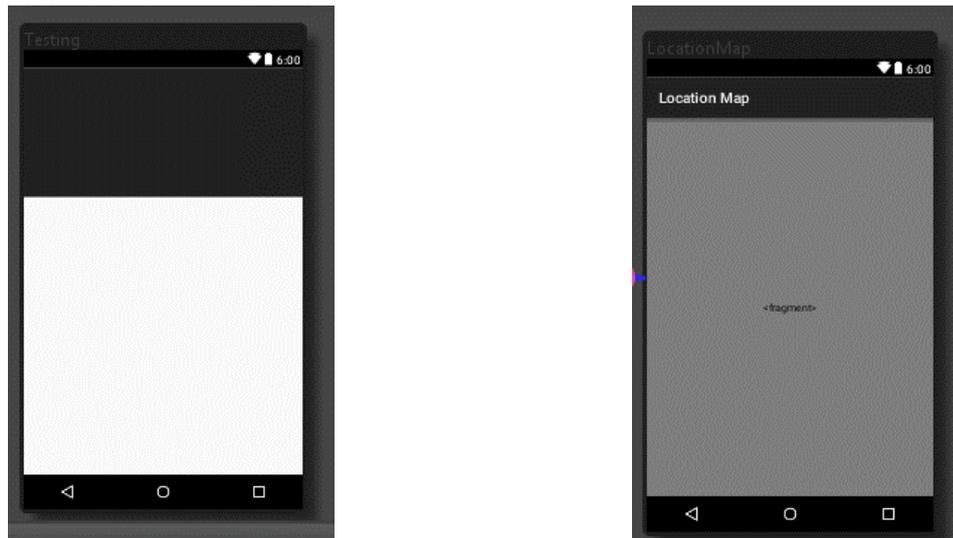
2.3.2 Diseño de la interfaz gráfica

La interfaz gráfica consiste en la representación de las vistas hacia el usuario. Es de gran importancia desarrollar interfaces bastante interactivas, intuitivas y visualmente apreciables para tener una alta probabilidad de adquisición de los usuarios.

Los esquemas tentativos son los siguientes:



Figuras 2.2, 2.3 y 2.4 Diseños de vistas Principal, Internet y Listado de llamadas/sms



Figuras 2.5 y 2.6 Diseño de vistas para Pruebas Generales y Mapa de Localización

2.3.3 Diseño de las clases, capas y patrones de funcionamiento para la comunicación y adquisición de la información de los servicios.

El patrón de diseño que permitirá el mejor desarrollo de la aplicación es el de MVC (Modelo-Vista-Controlador), que nos brindará la mejor interacción de las herramientas disponibles. Este patrón consiste en desarrollar software de tal manera que se disponga de tres capas:

- Modelo, que se refiere a toda la información disponible;
- Vista, que es la forma en la que se va a mostrar la información; y
- Controlador que va a ser el módulo encargado de parametrizar la información para la correcta interacción entre Modelo y Vista.

En el presente proyecto el Modelo se referiría a las Entidades definidas con la información parametrizada, las Vistas serían las actividades que mostrarán dicha información y el Controlador será el Servicio que se encargue de la interacción entre las Entidades y las Actividades.

De esta forma se definirá con más claridad cada una de las clases previstas en el Diagrama 2.1.

2.3.3.1 Servicio (QualityManager)

Este servicio va a ser el gestor y encargado de capturar todo evento de las llamadas, mensajería y conectividad del dispositivo. Android provee una clase para capturar los eventos de acuerdo a lo que la aplicación demande lo que facilita la implementación. Las clases-entidades Cell, Wifi y MLocation, se actualizarán mediante este servicio y se pondrá a disposición la información definida en cada uno, cada vez que la aplicación-usuario lo requiera o cada vez que se envíe a un servidor.

2.3.3.2 Entidad Internet (WiFi)

Es la entidad que contendrá toda la información sobre conectividad de red inalámbrica y de Internet. Los parámetros que se incluirán en esta clase son:

- Datos del proveedor
 - o Nombre
 - o Dirección IP
- Datos del Punto de Acceso
 - o Nombre de la red
 - o Dirección IP de Puerta de enlace.
 - o Dirección MAC de la red
- Datos del Dispositivo
 - o Dirección IP
 - o Dirección MAC
 - o Velocidad de conexión
 - o Fuerza de la señal
- Datos de conectividad
 - o Paquetes enviados
 - o Paquetes perdidos
 - o Desviación estándar de paquetes enviados (*Jitter*)
 - o Tiempo de envío y recepción de paquetes.

2.3.3.3 Entidad Telefonía (*Cell*)

En esta entidad constará la información de la conectividad con la red celular del dispositivo, además del registro de llamadas y mensajería. Los parámetros que se incluirán son:

- Proveedor
 - o Nombre
- Dispositivo
 - o Tipo de red
 - o Código de identificación de la celda
 - o Código de área de la celda
 - o Señal
- Llamadas. Entidad *Call*
 - o Número de recepción
 - o Duración
 - o Fecha y hora
 - o Tipo de llamada
- Mensajería. Entidad *Sms*
 - o Identificación de mensajes
 - o Número de recepción
 - o Fecha y hora
 - o Protocolo
 - o Tipo de mensaje
 - o Estado
 - o Cuerpo codificado

2.3.3.4 Entidad Localización (*MLocation*)

Es la responsable de las funcionalidades para la geolocalización del dispositivo, en donde se tendrá a disposición la latitud y la longitud en la que se encuentra el dispositivo, medidas de acuerdo al entorno actual del usuario, es decir, el estado de la red en ese momento (nivel de señal, movimiento, tipo de red, etc.). Además, se incluye la función de monitoreo constante durante intervalos y por cierto tiempo

establecido, lo que permitirá la realización del drive-test cada vez que lo solicite el usuario.

2.3.3.5 Vista de Estado de Red (*Telecom*)

En esta vista se imprimirá toda la información del estado de la red tanto WiFi como Celular, además de la Localización, obtenidos de las entidades mediante nuestro servicio que mantiene los parámetros actualizados. El usuario podrá conocer cómo se encuentran los servicios prestados y el lugar geoFigura en donde se ubica.

2.3.3.6 Vista de la Pruebas realizadas (*Testing*)

En ella el usuario podrá conocer los resultados de las pruebas realizadas de la red. En base a los parámetros establecidos la aplicación visualizará: la velocidad de conexión, la intensidad de la señal, la demora en el envío y recepción de datos, el promedio de los datos obtenidos y el registro de los puntos en dónde se realizó el sondeo.

2.3.3.7 Vista de la Localización en el Mapa (*LocationMap*)

Mediante el uso de la herramienta de *Google Maps*, nuestra aplicación es capaz de mostrar la localización aproximada del dispositivo en el mapa haciendo más interactiva la aplicación con el usuario. En ella también se podrá visualizar el seguimiento del drive-test realizado.

2.3.3.8 Vista de los registros de telefonía (*Calls y Smss*)

El usuario podrá ver el listado de los últimos registros de las llamadas y mensajes con el motivo de ilustración ya que la verdadera funcionalidad consiste en brindar ésta información para ser procesada en un servidor y así determinar los valores de KPI establecidos.

2.4 Conclusiones

Android a pesar de ser una plataforma de libre desarrollo, tiene varias limitaciones en cuanto a implementación de aplicaciones, esto es debido a la protección de la información personal de los usuarios, sin embargo, provee una gran capacidad de interacción entre hardware y software para el desarrollo de aplicaciones móviles.

Es importante tener en cuenta los parámetros que se rigen en nuestro entorno para la gestión de calidad de los servicios de telecomunicaciones. Así los usuarios sabrán cómo brindan el servicio los diferentes proveedores disponibles.

Para el buen desarrollo de software es primordial la realización previa de un análisis y un diseño de la aplicación, de esta manera se establecen los horizontes y el comportamiento adaptable de la aplicación, lo que se conoce como la Arquitectura de Software que es la base para la correcta construcción del producto final.

CAPÍTULO III

DESARROLLO DEL SOFTWARE

3.1 Introducción

El desarrollo de software consiste en las etapas de programación y pruebas de la ingeniería de software, en donde se construye la aplicación como tal, en algunos casos es necesario recurrir al rediseño de ciertos procesos, que, de acuerdo a los resultados de cada iteración, codificación-prueba, se llegará a definir la aplicación final.

3.2 Desarrollo funcional de la aplicación. Prototipo

Se empezará por desarrollar una primera versión solamente con la parte funcional de la aplicación, lo que se conoce como prototipo de la aplicación. Esto es, construir una aplicación ejecutable para verificar el funcionamiento de los patrones y la interacción de las clases, como se ha establecido en el diseño.

El servicio Android desarrollado permanecerá siempre activo en segundo plano, esto quiere decir que el usuario no podrá percibir la ejecución de los procesos que el servicio realice a menos que así se disponga en la aplicación. Como prototipo se indicará mediante mensajes de diálogo el momento en el que cada evento del servicio ocurra. Como por ejemplo cuando se realiza o se recibe llamadas como se puede apreciar en los siguientes Figuras.

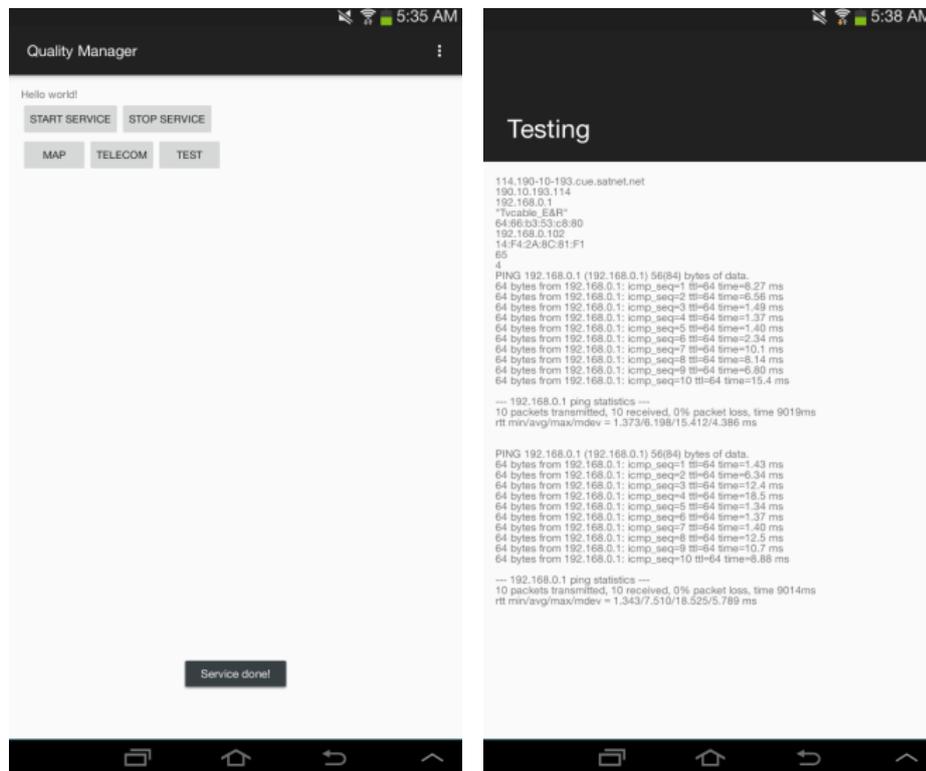


Figuras 3.3.1 y 3.3.2 Servicio Iniciado y Llamada Entrante



Figuras 3.3.3 y 3.3.4 Llamada en curso y Llamada finalizada

Se desarrolló la siguiente interfaz para las diferentes funciones de la aplicación:



Figuras 3.3.5 y 3.3.6 Interacción de funciones y Resultados de conectividad

Las demás actividades servirán para visualizar los parámetros en ese momento que el usuario lo solicite, según se indicó en el diseño.

3.3 Desarrollo de la interfaz gráfica

Las interfaces para el usuario se desarrollaron usando un tema personalizado de colores de acuerdo a la mayoría de los recomendados por la comunidad de desarrolladores de Android. Fue necesario realizar varias pruebas de cada interfaz, para verificar la correcta presentación: fuentes, alineación, formatos, etc. A pesar de disponer de la herramienta de diseño de *Android Studio* la mayoría de las interfaces se construyeron en modo de ejecución, es decir se construían por código cuando se ejecuta la aplicación.



Figura 3.3.7 Interfaz principal

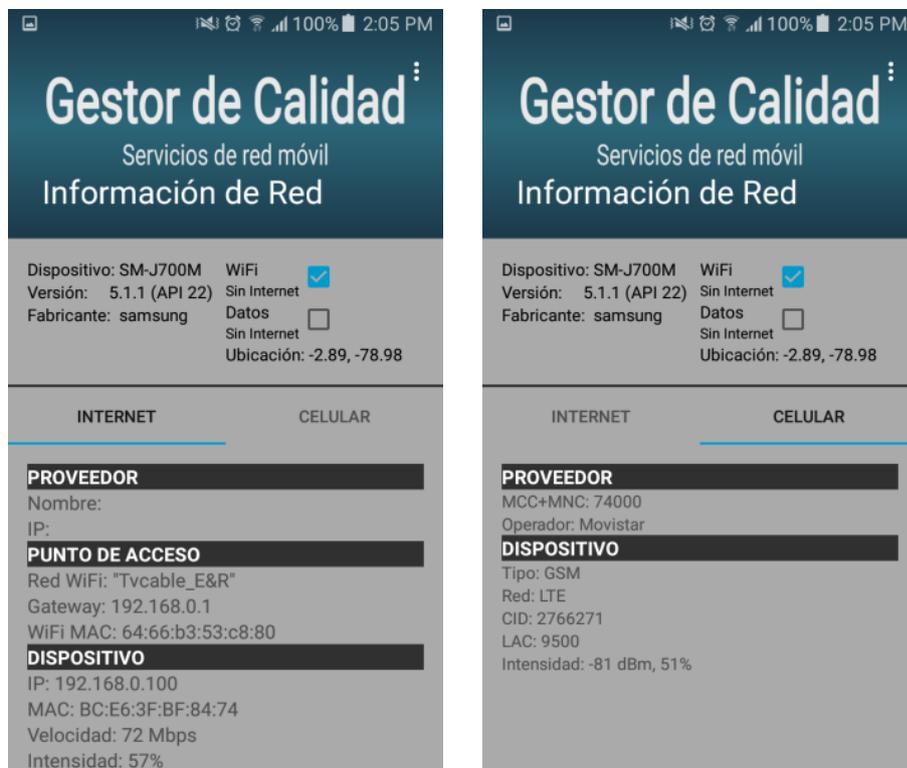


Figura 3.3.8 Interfaz de Información de Red

Para la localización del dispositivo se decidió implementar la herramienta de Google: *Google Maps API v2* para aplicaciones Android, lo que permite tener al cliente una mejor interacción para información de su ubicación.

Para esto fue necesario incluir la librería de *Google Play Services* dentro del proyecto, esto permite usar la existente herramienta de geolocalización en la aplicación.

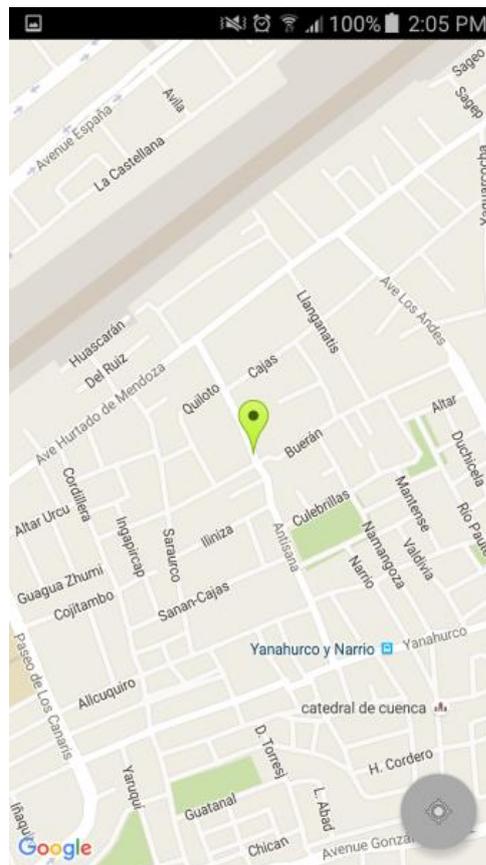


Figura 3.3.9 Interfaz de Localización

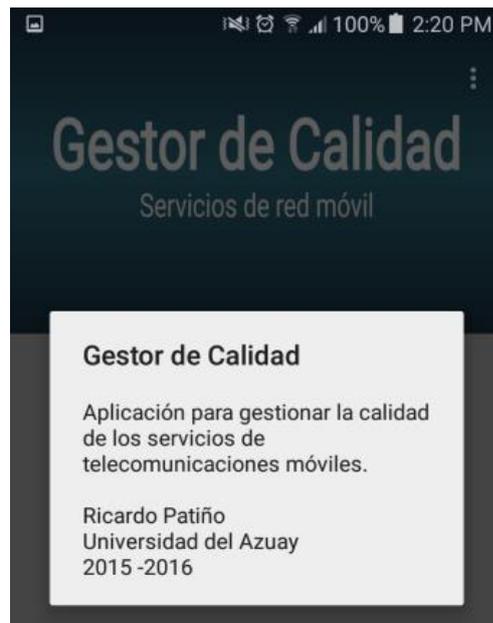


Figura 3.3.10 Interfaz de Información

Las interfaces fueron desarrolladas para los usuarios, las cuales le permiten tener la información de la red en ese momento.

El servicio es el encargado de almacenar la información necesaria para luego ser consumida por algún servidor de la organización reguladora y de control, interesada en la gestión. De tal manera que para esta versión de la aplicación no es posible verificar el estado de los parámetros de calidad, esto es debido a que es necesaria la implementación de un servidor abierto a la red pública para el tratamiento de los datos y la evaluación de los mismos. Es decir, la recepción de los datos de todos los dispositivos, el almacenamiento en una base de datos para su control y mapeo, y definición de tareas programadas para la evaluación de los KPI.

3.4 Depuración, simulaciones y pruebas en terminales Android.

Se realizó todas las pruebas necesarias para la verificación del buen funcionamiento de la aplicación, probando todas las interfaces y analizando los resultados que se obtienen del servicio implementado.

Para comprender el funcionamiento apropiado de la aplicación se va a detallar, a continuación, de donde se obtiene cada uno de los KPI definidos:

3.4.1 Internet

- Porcentaje de reclamos

Registro de reclamos receptados de los clientes y validados de acuerdo a las normas de cada empresa reguladora para su respectivo análisis. En la versión final de este proyecto no existirá un esquema genérico y completo de un formulario de reclamo debido a la falta de información para su elaboración.

3.4.2 Telefonía móvil

- Porcentaje de reclamos

Registro de reclamos receptados de los clientes y validados de acuerdo a las normas de cada empresa reguladora para su respectivo análisis. En la versión final de este proyecto no existirá un esquema genérico y completo de un formulario de reclamo debido a la falta de información para su elaboración.

- Porcentaje de llamadas establecidas

Se obtendrá el número total de intentos de llamadas de cada dispositivo, y se verificará en la parte del servidor las llamadas que se han establecido con éxito, sean éstas: cuándo el terminal receptor contesta, cuando suena el tono de ocupado o cuándo el receptor envía al buzón de voz, es decir que nuestra la debe tener los registros de los dos dispositivos en el servidor, si no existe el registro del receptor, es una llamada fallida.

- Tiempo de establecimiento de llamadas

Es el tiempo medido desde que el usuario presiona el botón de marcar hasta que el receptor recibe el primer tono de llamado. Para esta medición deben existir los dos registros (llamada y recepción), y se calculará la diferencia del tiempo tomado entre

el que llama y el receptor. Las llamadas que cumplan un tiempo $t \leq 12$ segundos serán tomadas sobre las llamadas establecidas para su determinación.

- Zona de cobertura

Para su medición se incluyó una función de *drive-test* en el cual cada usuario será capaz de realizar la medición de los niveles de señal por el tiempo que lo requiera, registrando el nivel de señal de cada punto en el cual se movilice con una actualización de 3 segundos, considerando los niveles mínimos de la señal. Los registros serán almacenados en el dispositivo con información adicional como tipo de red y tecnología para poner a disposición ya sea del usuario como de la entidad reguladora para su respectivo análisis.

- Porcentaje de mensajes cortos con éxito

Se contabilizará el número de mensajes que consten con doble registro (llamada y recepción) sobre el número total de mensajes para el cálculo.

3.5 Conclusiones

De acuerdo al análisis y al diseño de la aplicación se pudo desarrollar de una mejor manera el software propuesto, sin embargo, hubo algunos limitantes a causa de la compatibilidad de las versiones del sistema operativo Android definida para la mayoría de usuarios.

La versión de la aplicación desarrollada aún se encuentra en un estado básico, esto depende de la utilidad que se quiera dar en la organización reguladora y de control para la evaluación de los resultados.

Los resultados de las pruebas en algunos dispositivos fueron los esperados, además se pudo verificar el registro de la información realizado por el servicio de la aplicación funcionando en todo momento.

CAPÍTULO IV

PUBLICACIÓN DE LA APLICACIÓN ANDROID E IMPLEMENTACIÓN

4.1 Introducción

Para la publicación de la aplicación es muy importante conocer todos los requisitos previos que se deben aplicar a la versión final del software, de esta manera la aplicación cumple con todos los estándares para brindar la mejor experiencia a los usuarios. Una versión de lanzamiento hace que nuestro paquete de instalación sea generado como apto para su distribución, es decir, que mientras se tenga una versión de desarrollo (*debug*) el usuario debe habilitar la opción “permitir aplicaciones de origen desconocido” en su dispositivo para hacer posible la instalación del mismo. Es por esta razón, la importancia de la preparación de la aplicación antes de su lanzamiento.

4.2 Preparación para el despliegue del software.

Después de haber aplicado las pruebas pertinentes con la versión de desarrollo, ésta debe ser construida mediante la “*Lista de comprobación para el lanzamiento*” que exige Google, en donde se verificará: las políticas y los acuerdos de privacidad, la calificación del contenido de la aplicación, los países de distribución, tamaño, compatibilidad, precios, localización e información promocional.

El proceso de preparación fue llevado a cabo en base a la guía disponible en la página de *Android Developers* en la sección “*Preparing for release*”.

4.2.1 Firma Digital

Para la creación y colocación de la firma digital se ha usado la herramienta de *Android Studio* que genera un paquete de instalación auto-firmado que incluye la

información personal básica. Se puede implementar también un certificado digital, pero esto requiere costos adicionales dependiendo de la entidad de certificación.

4.2.2 Limpieza de código y de archivos del proyecto

Consiste en eliminar toda línea de código que estaba únicamente como guía personal para la programación: comentarios, lista de pendientes, reporte de errores, etc., y de eliminar los archivos del proyecto, dejando únicamente los necesarios para el funcionamiento de la aplicación.

4.2.3 Configuración de construcción de la versión

Se debe incluir en el proyecto el nombre del paquete, el ícono para la aplicación, la versión de Android compatible y la versión de nuestro software.

4.3 Instalación del software.

La instalación se la pudo realizar de forma manual en algunos dispositivos exitosamente, realizando las pruebas correspondientes. Para la instalación manual del software se debe habilitar la instalación de aplicaciones de fuentes que no sean Play Store. Esto se lo puede realizar en Ajustes → Seguridad → Orígenes desconocidos.

La forma de instalar la aplicación puede ser: por medio de USB con la ejecución de línea de comandos y por medio de un paquete de instalación (*APK*) que puede ser distribuido en diferentes medios dependiendo del tamaño del paquete (correo, bluetooth, etc.). Se usó las dos formas para las pruebas en los diferentes dispositivos.

4.4 Publicación en Google Play Store.

Para la publicación de una aplicación Android en el Google Play Store es necesario realizar algunos pasos que permiten el uso de los servicios de Google para la distribución de software. Se detallará los pasos realizados de acuerdo a las páginas de *Android Developers: "Get Started with Publishing"* y de *Google Support: "How to use the Google Play Developer Console"*.

4.4.1 Registro de una cuenta en Google Play para desarrolladores

- Ingresar con la cuenta Google

En primer lugar, se necesita una cuenta Google que generalmente la mayoría de usuarios la poseen, en caso de no tener una, se la puede crear durante el proceso. El siguiente paso es acceder con la cuenta Google e ingresar datos básicos de información personal como nombre, correo, país, ciudad, página web, etc., que servirán para la identificación hacia los usuarios de las aplicaciones que se publiquen.

- Leer y aceptar el “Acuerdo de Distribución para Desarrolladores”

Es un documento detallado en el que especifica todos los términos a los que se sujeta el desarrollador al momento de publicar aplicaciones.

- Agregar un método de pago para la cuota de registro

En este paso se creará una cuenta de *Google Payments*, en caso de no poseerla, se la puede crear de manera rápida durante el proceso. El monto a abonar es de \$25 con cualquier tarjeta de débito o crédito habilitada para transacciones internacionales.

- Detalles finales de la cuenta de Desarrollador y verificación

Es el último paso, en donde se verifica toda la información mediante una confirmación por correo. Además, se debe ingresar el “Nombre de Desarrollador”, el cuál es el que se va a mostrar en las aplicaciones.

4.4.2 Consola para Desarrolladores de Google Play

Es el lugar en donde se puede publicar nuestras aplicaciones y administrarlas según los comentarios y las calificaciones de los usuarios.

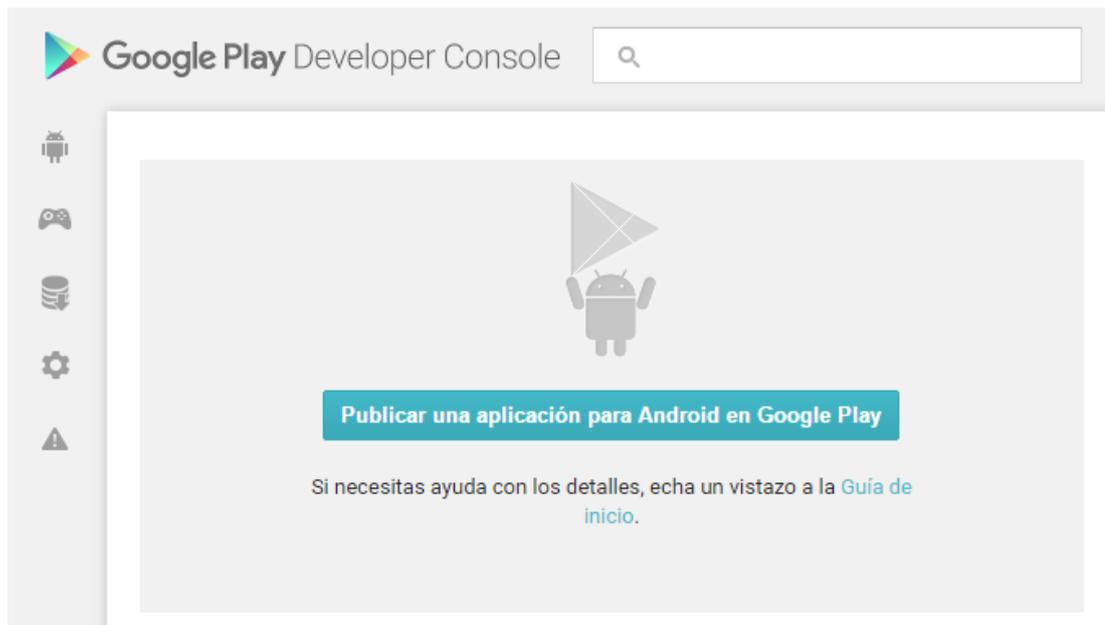


Figura 4.4.1 Consola de Desarrolladores de Google Play

Sin embargo, Google ofrece una herramienta bastante útil para asegurar el correcto lanzamiento de las aplicaciones mediante una etapa de pre-lanzamiento conocido como las versiones Alfa y Beta de la aplicación. En este punto se puede publicar la aplicación a un seleccionado grupo de usuarios, que, para esta etapa, se les llama “probadores” o “evaluadores”. Estos evaluadores darán la información del funcionamiento de la aplicación y serán capaces de reportar fallos mediante correo o por grupos de Google+, logrando así mejorar la calidad del software sin afectar la calificación de la aplicación cuando ésta sea lanzada abiertamente a los usuarios.

4.4.3 Publicaciones Alfa y Beta

Como se mencionó antes consiste en versiones de prueba antes de la versión final de producción. A continuación, se indicarán los pasos a seguir para configurar versiones de prueba Alfa y Beta de la aplicación Android.

En primer lugar, se debe cargar el *APK* en la consola de desarrolladores de *Google Play*.



Figura 4.4.2 Menú de selección para versión de APK

Fuente: Captura de pantalla tomada de Google Play Developer Console.

<https://play.google.com/apps/publish/>

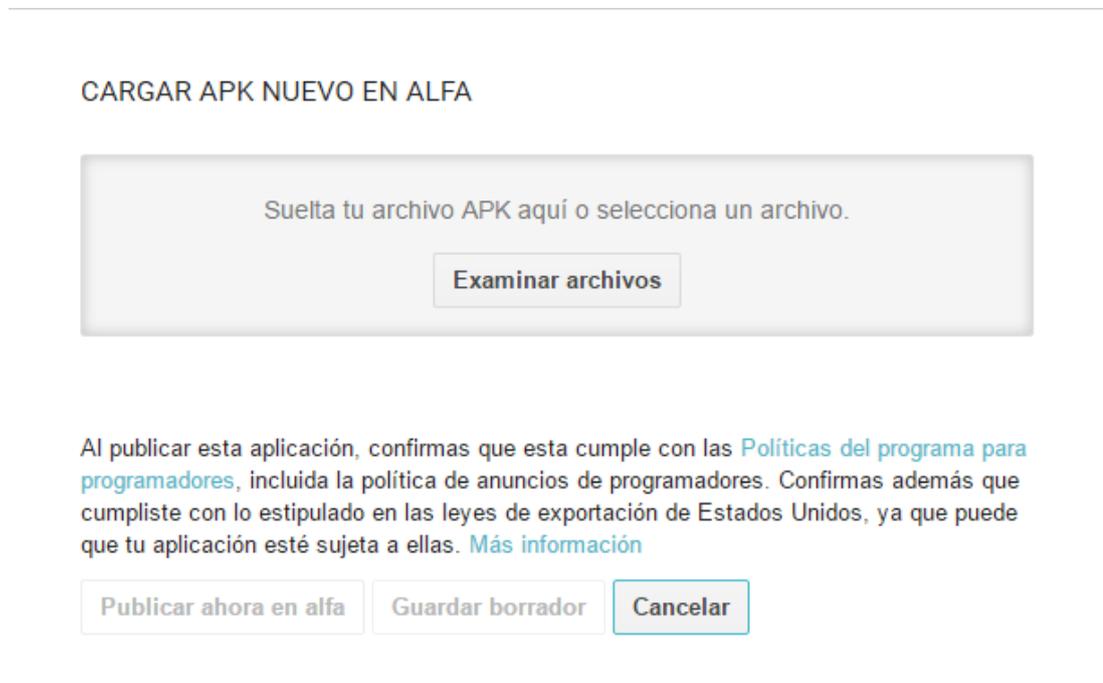


Figura 4.4.3 Cuadro de diálogo para la carga de APK

Fuente: Captura de pantalla tomada de Google Play Developer Console.

<https://play.google.com/apps/publish/>

Después se selecciona el método de prueba, en el cuál se puede elegir como se desea distribuir la versión hacia los verificadores. Existen tres métodos:

- Prueba Alfa cerrada: Los verificadores son agregados por medio de una lista de correos electrónicos, que es administrada por el dueño de la aplicación.

- Prueba Alfa abierta: Se distribuye un vínculo generado en el momento en que se publica la versión, el mismo que puede ser enviado a cualquier usuario.
- Prueba Alfa a través de Grupos de Google: Los usuarios que forman parte de Grupos de Google determinados para la distribución de la versión pueden acceder a la misma.
-

Se ha decidido implementar la prueba abierta para distribuir la versión de manera más simple por medio de un vínculo para la descarga.

APK ACTUAL publicado el 27/4/2016 12:09:49

▼ VERSIÓN	CARGADO EL	ESTADO	ACCIONES
1 (1.0)	22/4/2016	En fase Alfa	Ascender ▼

Figura 4.4.4 Estado de la versión del APK

Fuente: Captura de pantalla tomada de Google Play Developer Console.
<https://play.google.com/apps/publish/>

Una vez que se ha subido el APK satisfactoriamente, Google evalúa la aplicación y muestra el número de dispositivos compatibles, así como ofrece la opción de excluir específicamente los que no se desee implementar como alcance de la aplicación.

Llenar la *Ficha de Play Store*. Es el lugar en dónde se ingresa la información a ser publicada en la tienda de aplicaciones de Google. En dónde se exige:

- Título y Descripción

Español (Latinoamérica) – es-419
Administrar traducciones ▾

Título*
 Español (Latinoamérica) – es-419

 23 de 30 caracteres

Descripción breve*
 Español (Latinoamérica) – es-419

 77 de 80 caracteres

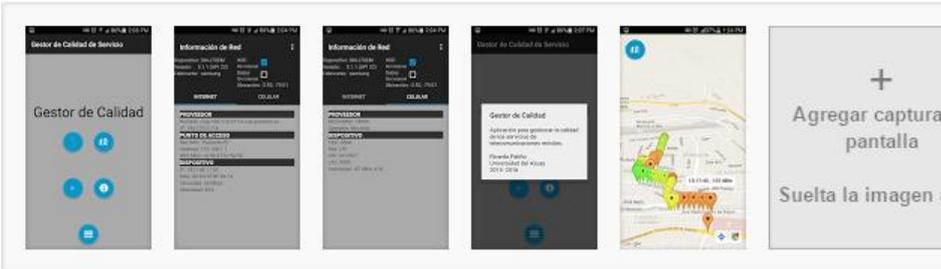
Descripción completa *
 Español (Latinoamérica) – es-419

Figura 4.4.5 Formulario de información de la versión

Fuente: Captura de pantalla tomada de Google Play Developer Console.
<https://play.google.com/apps/publish/>

Capturas de pantalla, ícono y Figura de presentación.

Teléfono
Tablet
Android TV
Android Wear



Ícono de alta resolución *
 Español (Latinoamérica) – es-419 (predeterminado)
 512 x 512
 PNG de 32 bits (alfa)

Gráfico de la función *
 Español (Latinoamérica) – es-419 (predeterminado)
 1024 x 500
 JPG o PNG de 24 bits (no alfa)




Figura 4.4.6 Formulario para la carga de capturas de pantalla de la aplicación

Fuente: Captura de pantalla tomada de Google Play Developer Console.
<https://play.google.com/apps/publish/>

Indicar la categoría y la calificación de contenido. Además de la calificación de edad, se debe llenar un formulario para especificar el tipo de contenido de la aplicación como violencia, lenguaje, sexualidad y privacidad de información personal, de esta manera Google evalúa y aplica el tipo de calificación según los diferentes estándares que se rigen en el mundo.

Tipo de aplicación *

Categoría *

Clasificación de contenido *

Se inhabilitaron algunas opciones de clasificación en función del contenido de la aplicación. [la calificación del contenido](#)

Nueva calificación de contenido *

 **CALIFICACIÓN APLICADA**
Enviada: 22 de abr. 11:20
[Ver detalles](#) [Más información](#)

Figura 4.4.7 Formulario de información para establecer la calificación de la versión

Fuente: Captura de pantalla tomada de Google Play Developer Console.
<https://play.google.com/apps/publish/>

Finalmente, ingresar datos de contacto, precio, países distribución y política de privacidad.

Una vez finalizados todos los pasos que Google exige en las aplicaciones se puede publicar la misma y el vínculo de Prueba Alfa Abierta es generado para ser distribuido hacia los verificadores. De esa forma se puede recibir informes de fallos y sugerencias, sin afectar nuestra reputación en las calificaciones de la aplicación. Toda la información previamente ingresada servirá para el ambiente de producción y puede ser actualizada en cualquier momento, los cambios serán reflejados en cada actualización del *APK* que sea publicado.



Figura 4.4.8 Información del estado de la aplicación publicada

Fuente: Captura de pantalla tomada de Google Play Developer Console.

<https://play.google.com/apps/publish/>



Figura 4.4.9 Visualización de la aplicación subida al Google Play Store

Fuente: Captura de pantalla tomada de Google Play Apps.

<https://play.google.com/apps/>

Según se vaya obteniendo la información de los usuarios, así como se vaya desarrollando avances y mejoras de la versión de la aplicación se puede subir a la versión Beta o inmediatamente a la versión de producción.

▼ VERSIÓN	CARGADO EL	ESTADO	ACCIONES
1 (1.0)	22/4/2016	En fase Alfa	Ascender ▼ Ascender a beta Ascender a producción

Figura 4.4.10 Estado de la versión de prueba y acciones a ejecutar

Fuente: Captura de pantalla tomada de Google Play Developer Console.

<https://play.google.com/apps/publish/>

4.5 Lanzamiento del software y sondeo en tiempo real.

El lanzamiento del software consiste en subir una versión de producción, la cual estará disponible para la libre descarga de los usuarios en la tienda de *Google Play*. Esto permitirá verificar el funcionamiento de la aplicación y comprobar si los resultados son los esperados de acuerdo a los propósitos de este proyecto. Debido a que el desarrollo de la parte del servidor está fuera de lo estimado tanto en los límites de tiempo como económicamente, se decidió verificar las pruebas realizadas de forma manual en cada dispositivo, es decir, obtener todos los reportes generados directamente en cada dispositivo, en lugar de tener un servidor en donde se podría consultar globalmente todas las pruebas de todos los dispositivos que tengan la aplicación instalada.

Las pruebas obtenidas de los dispositivos registrados para la versión de prueba que se ha subido a la tienda de aplicaciones de Google Play retornaron los resultados esperados tanto de la red inalámbrica como de la red celular, además se pudo observar el registro de la información de llamadas y mensajería como se detalla en el siguiente reporte:

4.5.1 Datos de conexión:

Las pruebas se tomaron en diferentes lugares y con diferentes dispositivos, entre las diferentes pruebas se obtuvieron los siguientes resultados:

Tabla 4.4.1 Datos obtenidos de dispositivo Samsung Galaxy S3

Dispositivo	Modelo – Versión Android:	Samsung – 4.1.2 (API 16)
	Localización:	-2.90, -78.98
	Proveedor de Internet:	Satnet
Internet	Velocidad de conexión:	26 Mbps
	Intensidad de conexión:	30 %
	Proveedor de Red Celular:	Movistar
Telefonía	Red:	GSM, UMTS
	Intensidad de la Señal:	-81 dBm
	Identificación de la Celda:	2766272

Tabla 4.4.2 Datos obtenidos de dispositivo Samsung Galaxy S3 mini

Dispositivo	Modelo – Versión Android:	Samsung – 4.1.2 (API 16)
	Localización:	-2.90, -78.98
	Proveedor de Internet:	Satnet
Internet	Velocidad de conexión:	72 Mbps
	Intensidad de conexión:	99 %
	Proveedor de Red Celular:	Movistar
Telefonía	Red:	GSM, UMTS
	Intensidad de la Señal:	-83 dBm
	Identificación de la Celda:	6793013

Tabla 4.4.3 Datos obtenidos de dispositivo Sony Xperia M

Dispositivo	Modelo – Versión Android:	Sony – 4.4 (API 19)
	Localización:	-2.89, -78.98
	Proveedor de Internet:	Satnet
Internet	Velocidad de conexión:	56 Mbps
	Intensidad de conexión:	49 %
	Proveedor de Red Celular:	Movistar
Telefonía	Red:	GSM, UMTS
	Intensidad de la Señal:	-80 dBm
	Identificación de la Celda:	2766272

Tabla 4.4.4 Datos obtenidos de dispositivo Samsung Galaxy J7

Dispositivo	Modelo – Versión Android:	Samsung – 5.1.1 (API 22)
	Localización:	-2.89, -78.98
	Proveedor de Internet:	Satnet
Internet	Velocidad de conexión:	72 Mbps
	Intensidad de conexión:	88 %
	Proveedor de Red Celular:	Movistar
Telefonía	Red:	GSM, LTE
	Intensidad de la Señal:	-77dBm
	Identificación de la Celda:	2766272

4.5.2 Drive-test

Para realizar el *drive-test* desde la aplicación se usa la interfaz de Localización con el *Google Maps*. Luego se activa el seguimiento de localización mediante el botón de localización que se encuentra en la parte inferior derecha. Como se puede observar en el siguiente Figura.

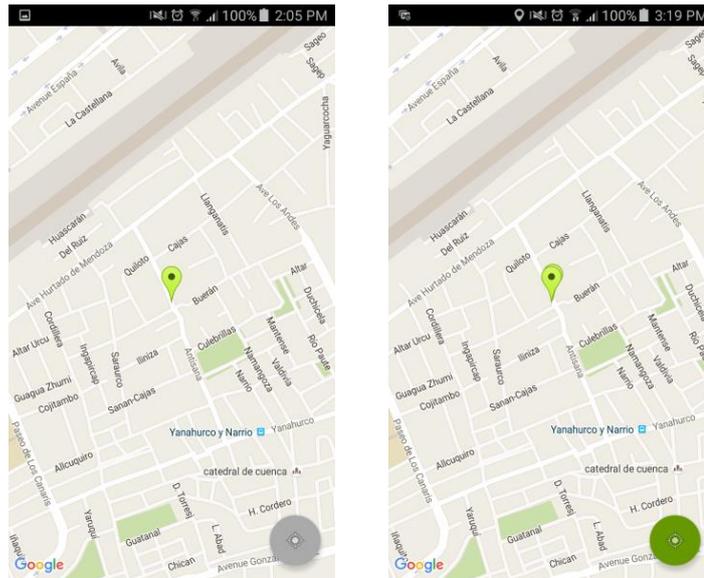


Figura 4.4.11 Habilitación del seguimiento de localización para el drive-test

De esta manera el dispositivo comenzará a imprimir en el mapa la actual localización cada 3 segundos, además de indicar la intensidad de la señal por medio del color del marcador, esto permitirá hacer el seguimiento de la intensidad de la señal celular por ubicación, distancia y tiempo.

Con esta herramienta se obtuvieron los siguientes resultados:

- Ruta 1

Totoracochoa, Parque Curiquingue – Circunvalación Sur, Cardeca B.C.

Distancia: 5.8 Km,
 Media de Intensidad: -99 dBm
 Máximo de Intensidad: -79 dBm
 Mínimo de Intensidad: -109 dBm

Ver Anexo 1

- Ruta 2

Hospital de IESS – Totoracocha, Parque Curiquingue

Distancia: 2.3 Km,
Media de Intensidad: -82 dBm
Máximo de Intensidad: -71 dBm
Mínimo de Intensidad: -101 dBm

Ver Anexo 2

- Ruta 3

Parque El Paraíso – Av. Américas – Totoracocha, Parque Curiquingue

Distancia: 8.5 Km,
Media de Intensidad: -98 dBm
Máximo de Intensidad: -71 dBm
Mínimo de Intensidad: -113 dBm

Ver Anexo 3

- Ruta 4

Estadio Alejandro Serrano – Universidad de Cuenca – “Y” del Cebollar

Distancia: 4.9 Km,
Media de Intensidad: -96 dBm
Máximo de Intensidad: -71 dBm
Mínimo de Intensidad: -110 dBm

Ver Anexo 4

4.6 Conclusiones

Como conclusiones de este capítulo se puede destacar que, para realizar la correcta publicación de una aplicación Android, es necesario realizar una serie de procesos para cumplir con los estándares básicos que requiere una aplicación de software.

Gracias a la ingeniería de software aplicada se pudo obtener una aplicación bastante estructurada e interactiva, lo cual no requirió de bastantes correcciones previo a la publicación.

CAPÍTULO V

MANUAL DE USUARIO

5.1 Introducción

El manual de usuario de una aplicación es un documento en el cual se especifica todas las funcionalidades por medio de pasos detallados para brindar una guía al usuario de cómo usar el software, garantizando el correcto uso del mismo.

La aplicación consta de los siguientes módulos:

- Pruebas
- Localización
- Información de Red
- Reporte de fallos

En la pantalla principal se puede navegar a cualquiera de los módulos mediante botones cuya representación de la funcionalidad de cada uno se muestra por medio de imágenes a intuición del usuario. Además, en la barra de herramientas se encuentra un menú para mostrar una pantalla de información de la aplicación, esta barra fue diseñada con vista al futuro como posible implementación para “Opciones de Configuración” para la personalización de las pruebas del usuario.

Como se había indicado previamente, como pantalla principal tenemos la siguiente:



Figura 5.1. Interfaz Principal. Botones de navegación

En donde la opción 1, despliega las diferentes opciones de la aplicación y la opción 2, despliega el menú de la barra de herramientas en donde se tiene únicamente la información de la aplicación.



Figura 5.2. Botones de acción y barra de herramientas

En base al concepto, la barra de herramientas se encuentra disponible en todas las interfaces, en donde el usuario puede acceder en todo momento a la pantalla de información, que es la siguiente:



Figura 5.3. Pantalla de información de la aplicación

A continuación, detallaremos las funcionalidades por separado de los diferentes módulos de la aplicación.

5.2 Módulo de Pruebas

En este módulo se puede evaluar el estado de la red en ese momento y además de verificar el registro de los eventos realizados por el teléfono celular. Las pruebas que se pueden realizar son:

- Red WiFi.
- Conectividad a Internet.
- Registro de Llamadas
- Registro de Mensajes (SMS)

Como se muestra en la siguiente ilustración:

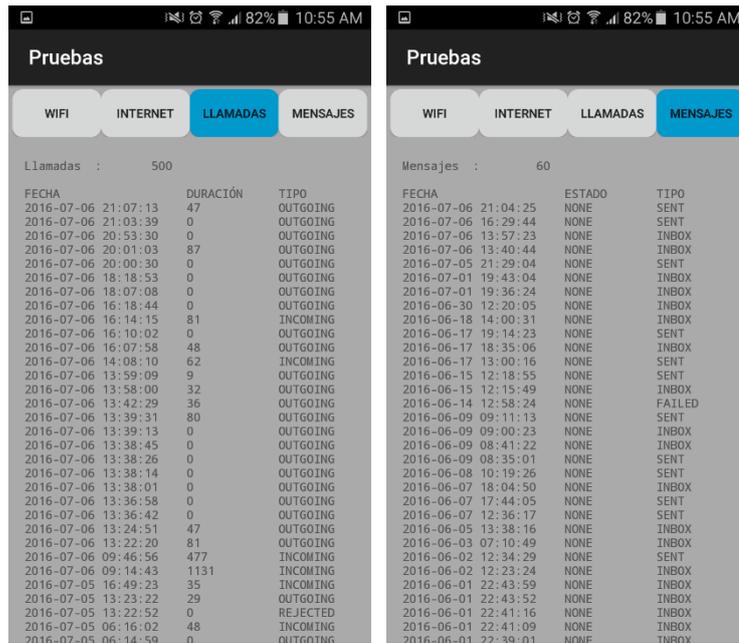


Figura 5.4. Módulo de Pruebas. Botones de navegación y pruebas

Cada botón especifica el tipo de prueba a realizar y los resultados que se muestran son los indicados en capítulos anteriores.



Figuras 5.5. y 5.6. Resultados de la conectividad a la red WiFi y Resultados de la conectividad a Internet



Figuras 5.7. y 5.8. Resultados de los registros de llamadas y Resultados de los registros de SMS

5.3 Módulo de Localización

En este módulo se puede apreciar el geolocalizador del dispositivo con la interfaz que provee *Google Maps*, en donde además de mostrar la ubicación actual, muestra también la calidad de la señal mediante dos formas: el color del marcador en un rango de colores entre verde y rojo, y una etiqueta que indica el nivel en dBm al presionar el marcador. Para usar esta funcionalidad de geolocalización es necesario tener activada la opción de “Ubicación” del teléfono; si no se encuentra activada, la aplicación le pregunta al usuario si desea activarla.

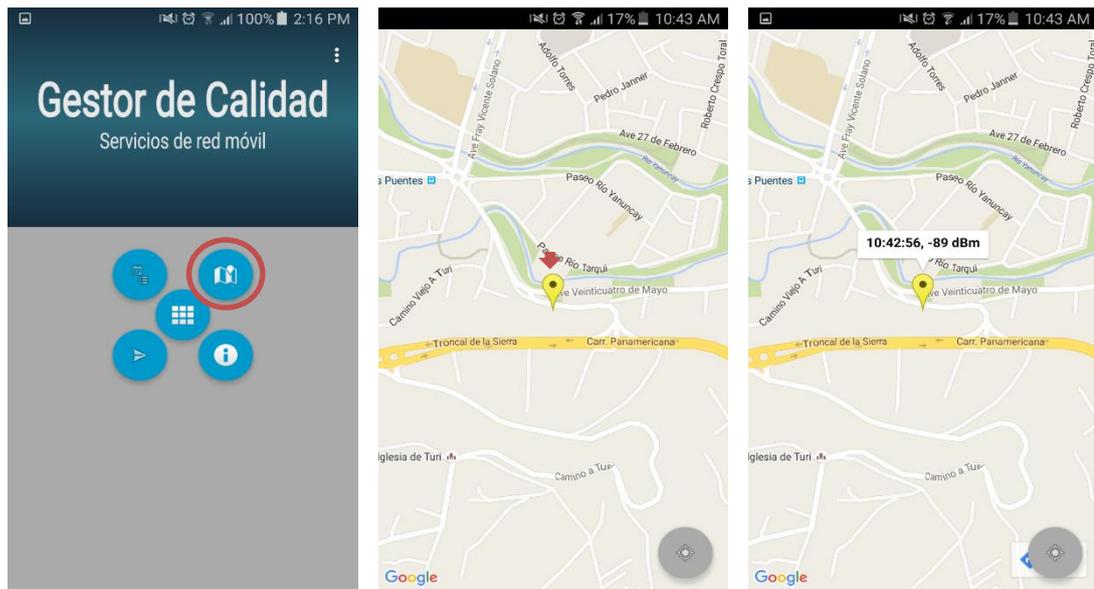


Figura 5.9. Interfaz de Geolocalización. Navegación y despliegue de información

Para realizar el *drive-test* se tiene un botón en la parte inferior derecha que inicialmente se encuentra deshabilitado. Al presionar el botón, éste cambia de color a verde, indicando que el *drive-test* se encuentra activo. En el mapa se dibujará un nuevo marcador con la respectiva ubicación e intensidad de señal cada 3 segundos, obteniendo un sondeo de la zona recorrida con el servicio activo.

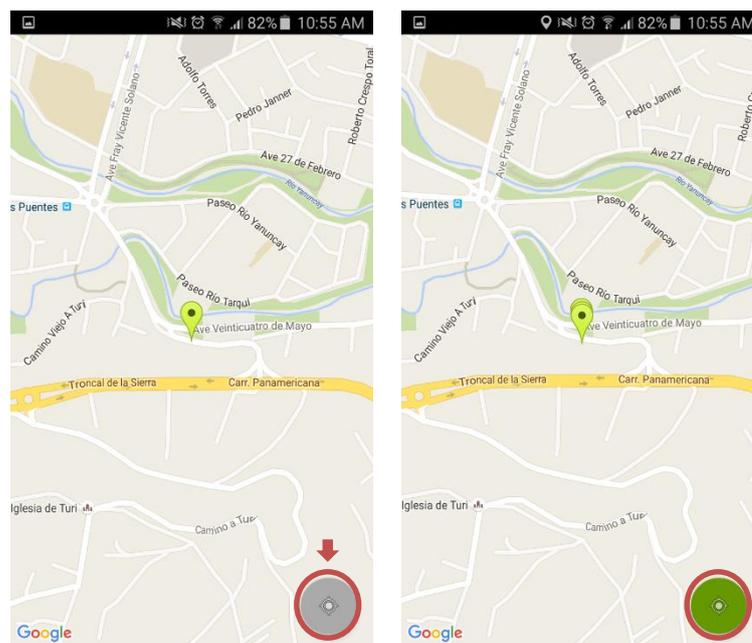


Figura 5.10. Activación del drive-test

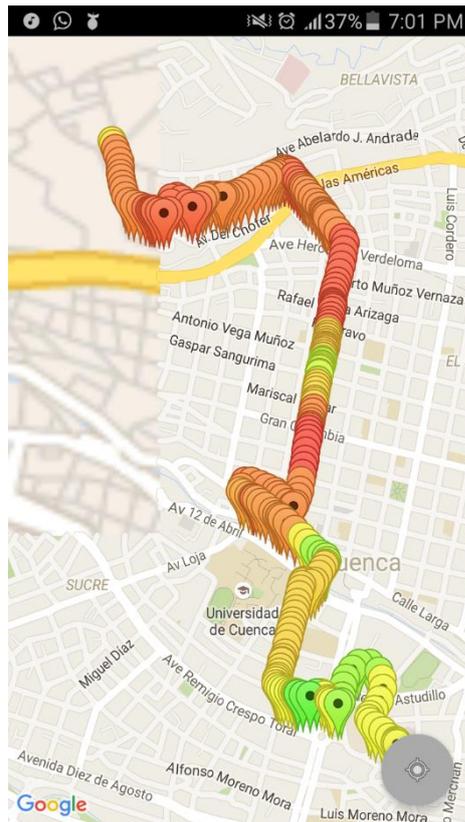


Figura 5.11. Ejemplo de drive-test usando la aplicación

5.4 Módulo de Información de Red

En esta pantalla se muestra toda la información de las redes conectadas actualmente al dispositivo. Inicialmente en la pantalla puede no mostrarse toda la información, pero este módulo cuenta con una interfaz desplazable que brinda al usuario la adaptabilidad de la información de acuerdo a la resolución de cada dispositivo.



Figura 5.12. Interfaz de la Información de Red. Desplazamiento para el ajuste de la pantalla

La visualización de la información está dividida en dos partes: WiFi y Celular, en donde se detallan todos los parámetros que se definieron como objeto de estudio para el proyecto. Además, en la parte superior se puede observar algunos datos del dispositivo como: modelo, versión de Android, WiFi y/o Datos Móviles activados y fabricante, como información general.

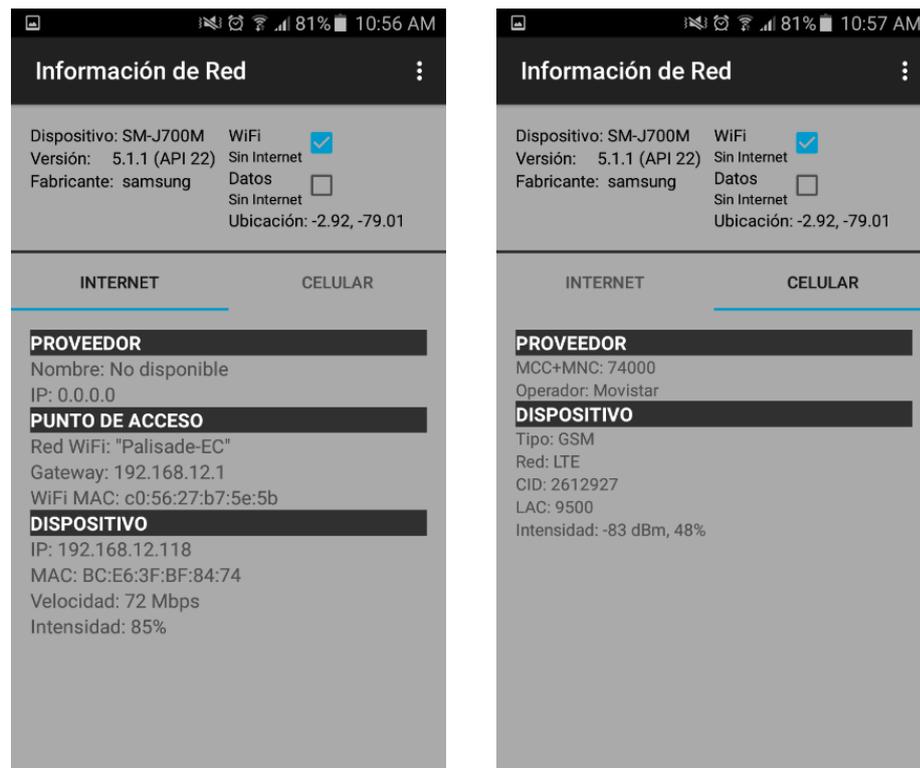


Figura 5.13. Detalle de la información del dispositivo y de la red a la cual está conectado

5.5 Módulo de Reporte de Fallos

Este módulo en particular es el único que no posee interfaz gráfica, puesto que no se definió un formato de reporte de fallos. Como se explicó anteriormente esto depende de cada entidad reguladora que, para el presente proyecto, no se consideró dentro del alcance de la aplicación.

De todas formas, se posee toda la información necesaria para generar un informe, que a manera de ejemplo se presentaría de la siguiente manera:

Tabla 5.1 Ejemplo de Reporte de Fallos

Dispositivo	Modelo – Versión Android:	Samsung – 5.1.1 (API 22)
	Fecha y Hora:	01/07/2016 12:35:46
	Tipo de Fallo:	Sin señal
	Localización:	-2.89, -78.98
	Proveedor de Internet:	Satnet
Internet	Velocidad de conexión:	1 Mbps
	Intensidad de conexión:	5 %
	Proveedor de Red Celular:	Movistar
Telefonía	Red:	GSM, LTE
	Intensidad de la Señal:	-113 dBm
	Identificación de la Celda:	2766272

5.6 Conclusiones

El manual de usuario es una herramienta de gran utilidad puesto que brinda todos los detalles del funcionamiento del software, de esta forma se puede evitar que el usuario final cometa errores que, dependiendo de la magnitud del software, pueden ser manejados como no corregibles para versiones posteriores, obviamente relacionado con la probabilidad de ocurrencia de los mismos.

CONCLUSIONES

Las conclusiones más destacadas que se pudo encontrar fueron las siguientes:

- Desarrollar aplicaciones en la plataforma Android tiene varias ventajas entre las principales está la gran cantidad usuarios que se puede alcanzar. Al ser de código abierto permite el libre desarrollo de cualquier tipo de aplicación, dependiendo de las necesidades y las limitaciones que se encuentren en la compatibilidad de los dispositivos móviles.
- Realizar la gestión de los servicios de telecomunicaciones directamente desde los usuarios permite tener una mejor interacción para la regulación y el control de los diferentes proveedores que prestan los servicios, además de mantener una mejor calidad. Es por esta razón que la aplicación desarrollada está bien enfocada para este tipo de implementación.
- Para el correcto funcionamiento de un software es necesario realizar un buen análisis y un buen diseño del mismo, de esta manera el desarrollo se lo puede realizar de una forma más eficiente, evitando una reestructuración del sistema que implica la inversión de muchos más recursos. Gracias a este procedimiento la aplicación pudo ser desarrollada de la manera correcta y con los resultados esperados.
- La publicación del software en la tienda de aplicaciones de *Google Play* requiere una serie de requisitos y procesos a preparar para cumplir con las políticas que rige *Google*, pero toda la información se encuentra disponible en la página de desarrolladores de Android y en la documentación de *Google Play* en donde se especifica paso a paso la elaboración de la versión para su publicación. Lo más importante a destacar es la útil herramienta proporcionada para medios de prueba Alfa y Beta, lo que permite tener una primera impresión de los usuarios acerca del funcionamiento de la aplicación para mejorar y evitar malas reseñas antes de la versión de producción.

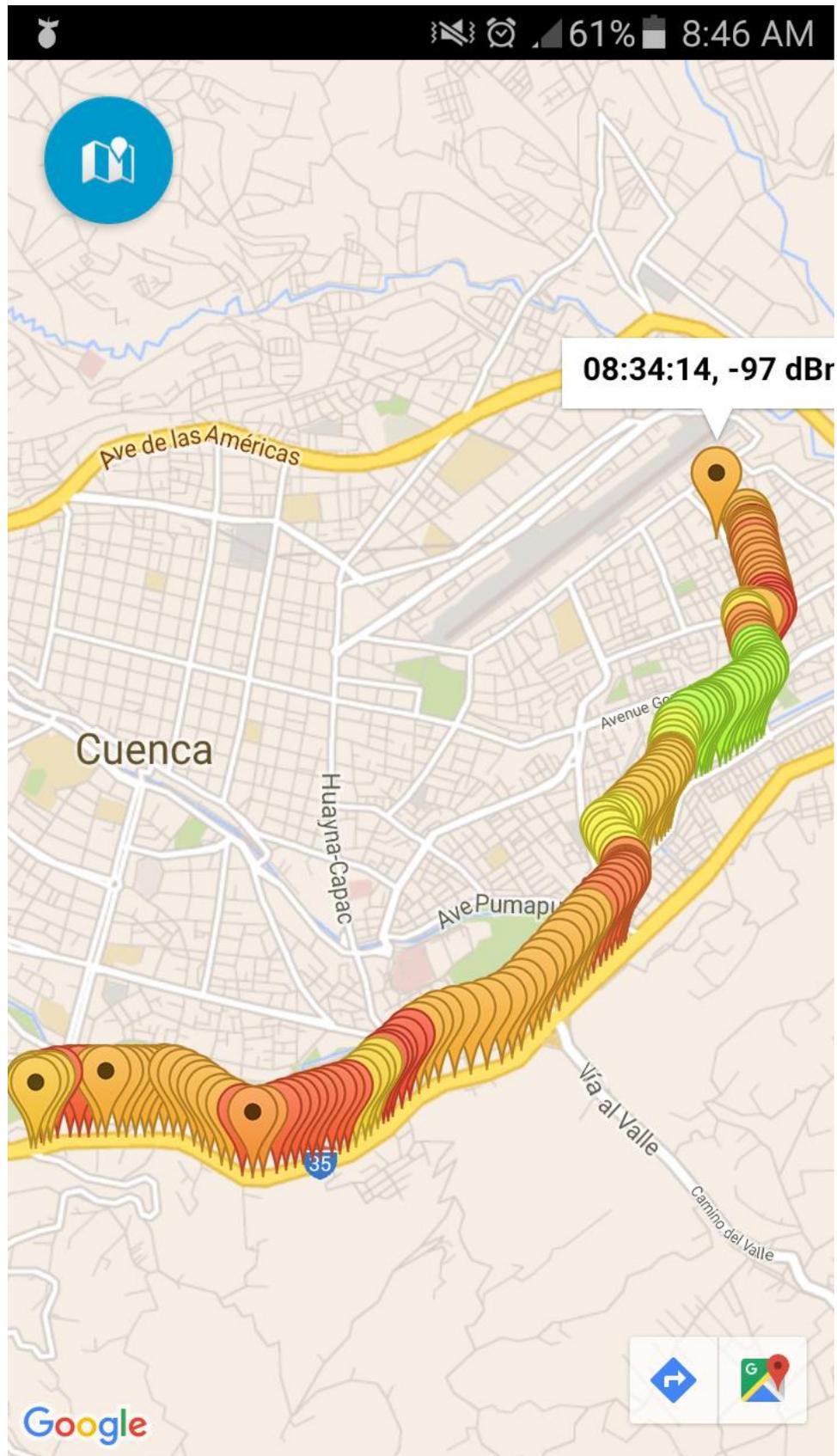
- A pesar de lograr una versión final de la aplicación aún se pueden observar mejoras a realizar, sin embargo, esto depende de la coordinación de las organizaciones que rigen la gestión de los servicios para su completa implementación. Además, cabe especificar que esto depende del desarrollo en la parte del servidor según la empresa reguladora que lo desee. En el presente proyecto, debido a falta de recursos no fue posible realizar este desarrollo del servidor con apertura a la red pública.

BIBLIOGRAFÍA

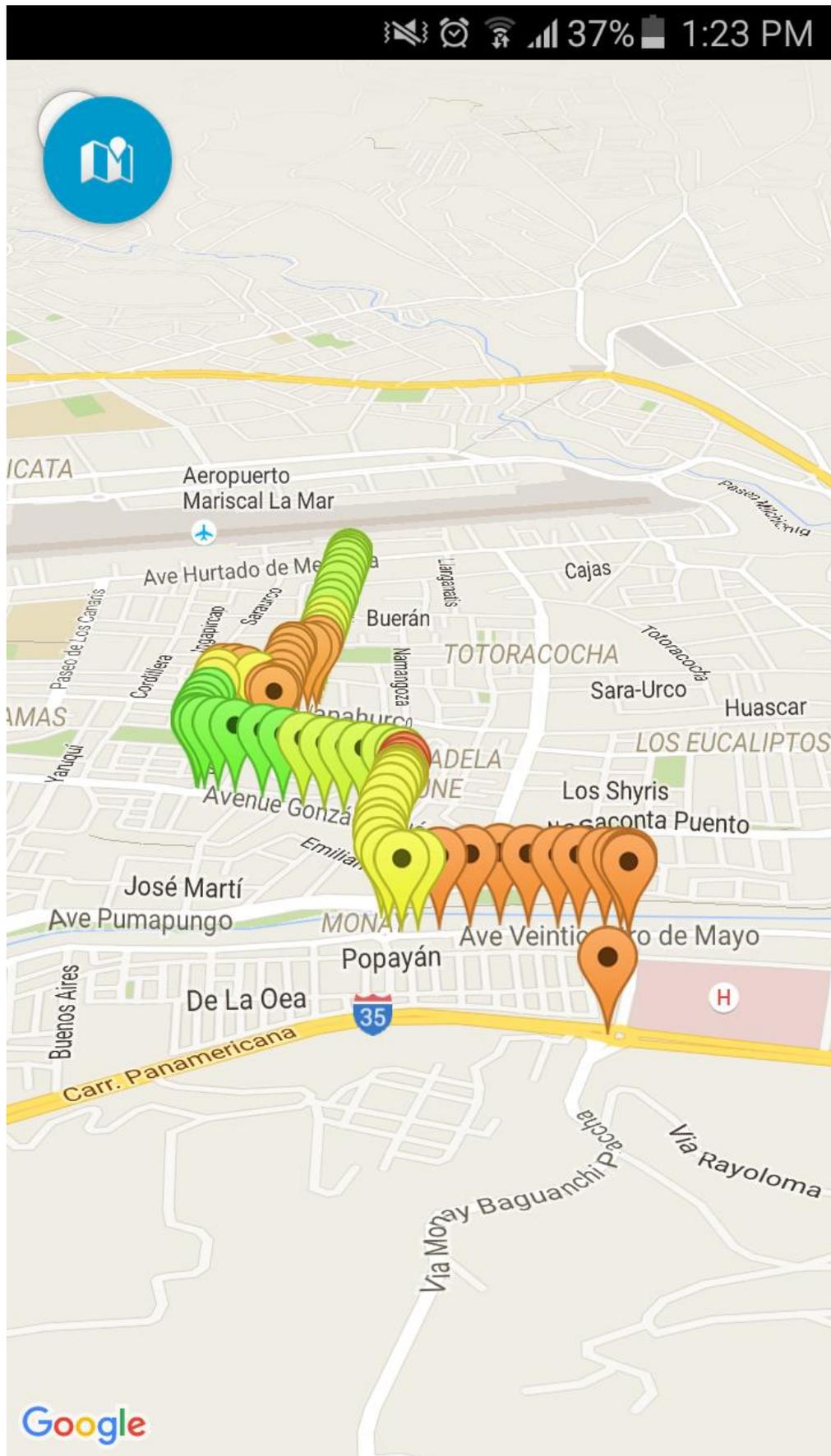
- Android. (Diciembre de 2015). *Dashboards | Android Developers*. Obtenido de <http://developer.android.com/intl/es/about/dashboards/index.html>
- CONATEL. (Septiembre de 2009). *Ministerio de Telecomunicaciones y Sociedad de la Información | Ecuador*. Obtenido de <http://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2012/11/Parametros-de-Calidad-del-Servicio-de-Valor-Agregado-de-Internet.pdf>
- INTECO, Laboratorio Nacional de Calidad del Servicio. (2009). *Ingeniería del Software: Metodologías y Ciclos de vida*.
- MINTEL. (Junio de 2014). *Ministerio de Telecomunicaciones y Sociedad de la Información | Ecuador*. Obtenido de <http://www.telecomunicaciones.gob.ec/parametros-de-calidad-mas-estrictos-para-la-telefonía-movil-seran-controlados-desde-el-primer-de-julio/>
- ORTEGA GARZON, W. D. (2010). *Establecimiento de parámetros de calidad de servicio para comunicaciones de voz en el Servicio Móvil Avanzado (SMA), en la República del Ecuador*. Sangolquí.

ANEXOS

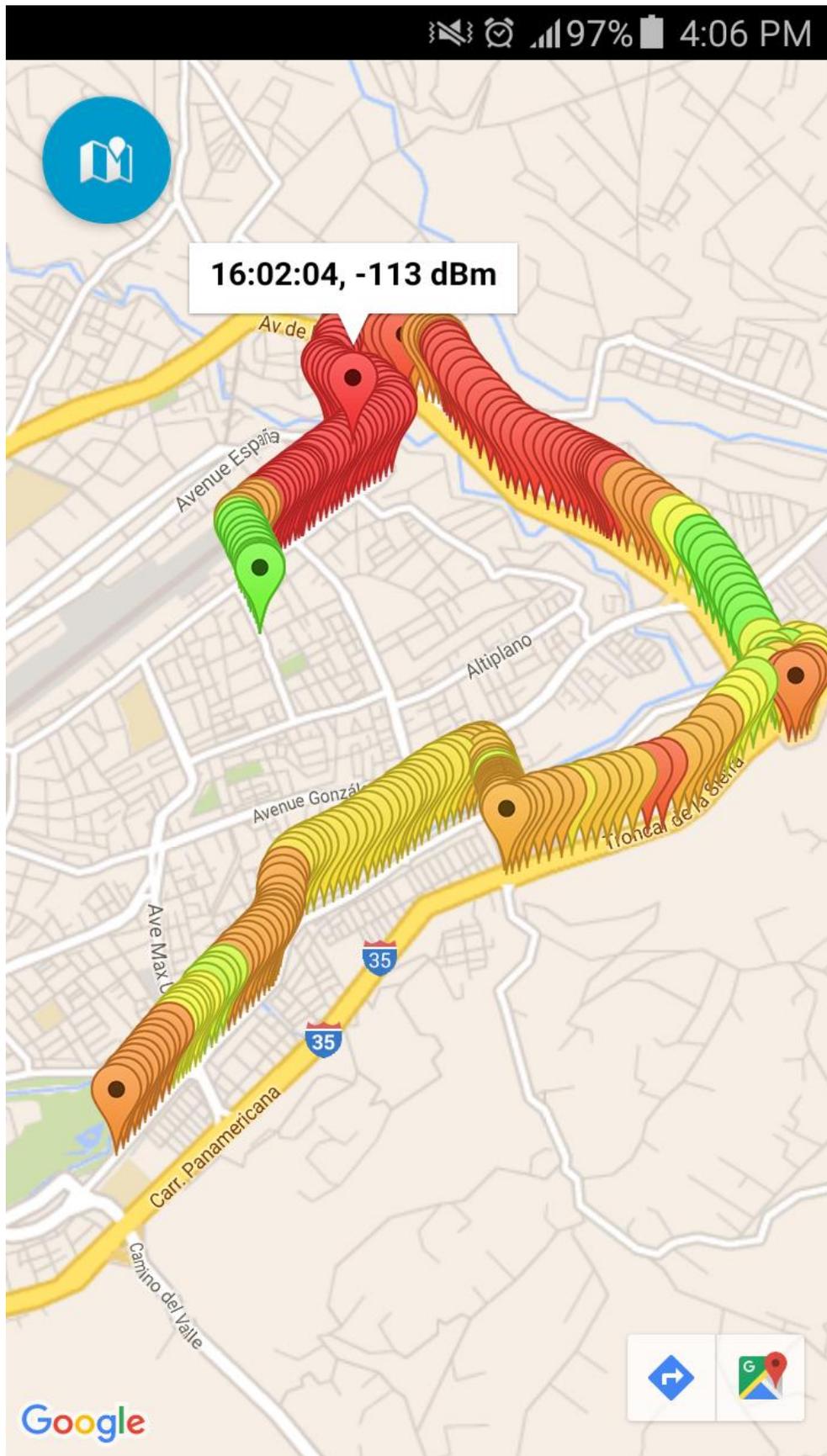
Anexo 1: Drive-Test, Ruta N° 1.



Anexo 2: Drive-Test, Ruta N° 2.



Anexo 3: Drive-Test, Ruta N° 3.



Anexo 4: Drive-Test, Ruta N° 4.

