



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y  
TELECOMUNICACIONES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERÍA EN ELECTRÓNICA Y  
TELECOMUNICACIONES**

**TEMA: “IDENTIFICACIÓN DE LA COBERTURA DE LAS  
REDES INALÁMBRICAS DE LA UNIVERSIDAD DE LAS  
FUERZAS ARMADAS – ESPE A TRAVÉS DE UNA  
APLICACIÓN MÓVIL BASADA EN MOBILE  
CROWDSOURCING”**

**AUTOR: NÚÑEZ HERNÁNDEZ ALONSO ALEJANDRO**

**DIRECTOR: MSC. TRIVIÑO DANIEL**

**SANGOLQUÍ  
2017**

## CERTIFICACIÓN



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y  
TELECOMUNICACIONES**

### CERTIFICACIÓN

Certifico que el trabajo de titulación, "IDENTIFICACIÓN DE LA COBERTURA DE LAS REDES INALÁMBRICAS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE A TRAVÉS DE UNA APLICACIÓN MÓVIL BASADA EN MOBILE CROWDSOURCING" realizada por el señor ALONSO ALEJANDRO NÚÑEZ HERNÁNDEZ, ha sido revisado en su totalidad y analizado por el software antiplagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas – ESPE, por lo tanto me permito acreditarlo y autorizar al señor ALONSO ALEJANDRO NÚÑEZ HERNÁNDEZ para que lo sustente públicamente.

Sangolquí, 14 de febrero del 2017

Ing. Daniel Triviño, MSc.

**DIRECTOR**

## AUTORÍA DE RESPONSABILIDAD



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y**  
**TELECOMUNICACIONES**

### AUTORÍA DE RESPONSABILIDAD

Yo, **ALONSO ALEJANDRO NÚÑEZ HERNÁNDEZ**, con cédula de identidad N° 1718871393 declaro que este trabajo de titulación “**IDENTIFICACIÓN DE LA COBERTURA DE LAS REDES INALÁMBRICAS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE A TRAVÉS DE UNA APLICACIÓN MÓVIL BASADA EN MOBILE CROWDSOURCING**” ha sido desarrollada considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que es trabajo de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

**Sangolquí, 14 de febrero del 2017**

ALONSO ALEJANDRO NÚÑEZ HERNÁNDEZ

C.C.: 1718871393

## AUTORIZACIÓN



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y**  
**TELECOMUNICACIONES**

### AUTORIZACIÓN

Yo, **ALONSO ALEJANDRO NÚÑEZ HERNÁNDEZ**, autorizo a la Universidad de las Fuerzas Armadas – ESPE publicar en la biblioteca virtual de la institución el presente trabajo de titulación “**IDENTIFICACIÓN DE LA COBERTURA DE LAS REDES INALÁMBRICAS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE A TRAVÉS DE UNA APLICACIÓN MÓVIL BASADA EN MOBILE CROWDSOURCING**” cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

**Sangolquí, 14 de febrero del 2017**

ALONSO ALEJANDRO NÚÑEZ HERNÁNDEZ

C.C.: 1718871393

## DEDICATORIA

*A Dios por su amor puro, infinito e incondicional; a mis padres Laly y Alonso que siempre me brindaron su apoyo y me alentaron a alcanzar todos mis sueños, a mis hermanos Eduardo y Pablo que siempre estuvieron cuando los necesite y me ayudaron a salir adelante, a toda mi familia porque con su ejemplo me han enseñado que todo es posible para el que persevera.*

*Alonso Alejandro Núñez Hernández*

## AGRADECIMIENTO

Agradezco de todo corazón a mis padres Alonso y Laly que siempre me apoyaron incondicionalmente y supieron sacar adelante nuestro hogar, a mis hermanos Eduardo y Pablo con quienes compartí todo desde que éramos pequeños, a mi tía Susana quién me demostró que a pesar de cualquier circunstancia es posible seguir adelante en la vida y a toda mi familia que de una u otra manera me brindaron todo su apoyo cuando lo necesité.

Agradezco a todos mis profesores que durante la etapa universitaria me han brindado desinteresadamente sus conocimientos y habilidades profesionales en pos de un mejor futuro para nuestra sociedad.

Agradezco a todas las personas que contribuyeron a la realización de este proyecto que sin duda son muchas y no hubiera podido alcanzar todos los objetivos sin su esfuerzo.

Agradezco a todos y cada uno de mis amigos con quienes compartimos metas, anhelos, sueños, triunfos en esta etapa maravillosa de la vida llamada Universidad y que así mismo estuvieron presentes cuando las cosas se ponían difíciles.

Le agradezco a Dios que sin duda me ha demostrado su infinito amor en todos los momentos de mi vida.

Alonso Alejandro Núñez Hernández

## ÍNDICE GENERAL

CERTIFICACIÓN .....	ii
AUTORÍA DE RESPONSABILIDAD .....	iii
AUTORIZACIÓN .....	iv
DEDICATORIA .....	v
AGRADECIMIENTO .....	vi
ÍNDICE GENERAL .....	vii
ÍNDICE DE TABLAS .....	x
ÍNDICE DE FIGURAS.....	xi
RESUMEN.....	xiii
ABSTRACT .....	xiv
CAPITULO 1 .....	1
INTRODUCCIÓN .....	1
1.1 Antecedentes.....	1
1.2 Justificación e Importancia.....	2
1.3 Alcance del Proyecto .....	4
1.4 Objetivos .....	6
1.4.1 General .....	6
1.4.2 Específicos.....	6
CAPITULO 2 .....	7
MARCO TEÓRICO .....	7
2.1 Crowdsourcing.....	7
2.1.1 Conceptos y Características de Crowdsensing .....	8
2.1.2 Aplicaciones para Mobile Crowdsensing .....	8
2.1.3 Ventajas y Desventajas de Crowdsensing .....	9
2.1.4 Arquitectura Crowdsensing .....	10
2.2 Dispositivos Móviles.....	11

2.2.1 Tipos de Dispositivos Móviles .....	11
2.2.2 Tipos de Terminales en los Dispositivos Móviles .....	12
2.2.3 Mercado de los Sistemas Operativos .....	13
2.2.4 Tipos de Sensores en Dispositivos Móviles .....	14
2.3 Sistema Operativo Android .....	14
2.3.1 Programación en Android.....	15
2.3.2 Interfaz de Programación de Aplicaciones (API) .....	16
2.3.3 Arquitectura del Sistema Operativo Android .....	17
2.3.4 Componentes de una Aplicación Android.....	19
2.3.5 Aplicaciones Móviles .....	21
2.4 Servicio Web .....	22
2.4.1 Arquitectura Cliente – Servidor .....	23
2.5 Redes Inalámbricas .....	26
2.5.1 Arquitectura Wlan IEEE 802.11 .....	27
2.5.2 Parámetro de Intensidad de Señal .....	30
2.6 Métodos de Posicionamiento.....	30
2.6.1 GPS (Global Positioning System).....	31
2.6.2 Posicionamiento mediante GPS Asistido .....	32
CAPÍTULO 3 .....	34
DISEÑO Y DESCRIPCIÓN DEL SISTEMA .....	34
3.1 Descripción de Hardware .....	34
3.1.1 Definición de Requerimientos .....	34
3.2 Descripción y Diseño de Software.....	36
3.2.1 Requerimientos de Diseño.....	36
3.2.2 Aplicación Móvil .....	37
3.2.3 Mapas .....	47
3.2.4 Plataforma Web .....	49
Fuente: (W3C, Servicios Web, s.f.).....	58
3.3 Selección del Alojamiento Web .....	58
CAPÍTULO 4 .....	60
IMPLEMENTACIÓN DEL SISTEMA.....	60
4.1 Arquitectura del Sistema .....	60



4.1.1 Escenario para el Sistema Crowdsensing .....	60
4.2 Desarrollo de la Plataforma Crowdsensing .....	61
4.3 Proceso de Despliegue e Instalación de la aplicación.....	69
4.4 Respaldo Automático de la Base de Datos.....	69
4.5 Definición de Protocolo de pruebas .....	72
CAPÍTULO 5 .....	77
PRUEBAS DE FUNCIONAMIENTO .....	77
5.1 Recolección de Datos por Crowdsensing.....	77
5.1.1 Determinación del número de participantes .....	77
5.1.2 Determinar el tiempo requerido para enviar datos desde el dispositivo móvil	79
5.1.2 Exactitud del método de posicionamiento.....	80
5.1.3 Variación de RSSI con respecto a la distancia .....	84
5.1.4 Uso de la herramienta de análisis Wireshark.....	86
5.1.5 Utilizar las estadísticas del Servidor web.....	89
5.2 Sistema de Monitorización .....	91
5.2.1 Resultados Obtenidos .....	91
CAPÍTULO 6 .....	96
CONCLUSIONES Y LINEAS FUTURAS DE INVESTIGACIÓN .....	96
6.1 Conclusiones.....	96
6.2 Recomendaciones .....	98
6.3 Trabajos Futuros .....	98
BIBLIOGRAFÍA .....	99

## ÍNDICE DE TABLAS

Tabla 1: Dispositivos móviles más conocidos en la actualidad .....	11
Tabla 2: Tipos de gamas .....	12
Tabla 3: Tipos de Sensores en Dispositivos Móviles .....	14
Tabla 4: Dispositivos móviles más conocidos en la actualidad .....	15
Tabla 5: Principales APIs Android .....	16
Tabla 6: Librerías Nativas Android .....	18
Tabla 7: Tipos de Relaciones .....	26
Tabla 8: Ventajas y Desventajas de las redes inalámbricas .....	26
Tabla 9: Modificaciones del estándar 802.11.....	27
Tabla 10: Requerimientos de Hardware .....	34
Tabla 11: Variaciones de BSSID en cada piso (Bloque A) .....	39
Tabla 12: Métodos de AsyncTask .....	43
Tabla 13: Tipos de mapas .....	47
Tabla 14: Determinación de número de decimales a utilizar .....	49
Tabla 15: Opciones incluidas en la página web de la aplicación .....	53
Tabla 16: Rango de RSSI para graficar en el mapa.....	55
Tabla 17: Tipos de datos de los atributos.....	58
Tabla 18: Diferencias entre métodos GET y POST .....	58
Tabla 19: Características del hosting miarroba .....	59
Tabla 20: Componentes de la aplicación desarrollada.....	62
Tabla 21: Tareas llevadas a cabo por el programador de tareas.....	72
Tabla 22: Identificadores únicos de los participantes.....	78
Tabla 23: Tiempo requerido para el envío de datos .....	79
Tabla 24: Medidas de latitud y longitud obtenidas.....	81
Tabla 25: Valores Obtenidos de RSSI .....	84
Tabla 26: Colores para rango de valores de RSSI.....	85
Tabla 27: Determinación de propietarios web .....	86

## ÍNDICE DE FIGURAS

Figura 1: Ejemplo de mapa de cobertura .....	4
Figura 2: Ejemplo de Aplicación Móvil en un Teléfono Celular .....	5
Figura 3: Diagrama de Funcionamiento.....	5
Figura 4: Arquitectura Crowdsensing.....	10
Figura 5: Sistemas Operativos más utilizados.....	13
Figura 6: Arquitectura de Android .....	17
Figura 7: Diálogo ANR.....	20
Figura 8: Línea de tiempo de las distintas tareas realizadas .....	20
Figura 9: Realización de tareas en segundo plano.....	20
Figura 10: Flujo General de una Aplicación .....	22
Figura 11: Dispositivos configurados en modo Ad-Hoc .....	28
Figura 12: Dispositivos configurados en modo Infraestructura .....	28
Figura 13: Proceso de asociación .....	29
Figura 14: Conjunto de Servicios Extendidos.....	29
Figura 15: Triangulación de los satélites .....	31
Figura 16: Visión General de GPS - Asistido .....	33
Figura 17: Requerimientos de Hardware .....	35
Figura 18: Diagrama de modelo de uso de la aplicación móvil .....	38
Figura 19: Ciclo de vida de una activity en Android.....	41
Figura 20: Boceto de la aplicación .....	45
Figura 21: Panel de Navegación de la aplicación (fragments asociados).....	46
Figura 22: Front - End y Back - End .....	49
Figura 23: Portada de la página web desarrollada.....	53
Figura 24: Selección de opción exteriores .....	54
Figura 25: Selección de opción interiores.....	55
Figura 26: Selección del piso en la red correspondiente.....	55
Figura 27: Modelo Entidad – Relación para la Base de Datos.....	57
Figura 28: Escenario para el Sistema Crowdsensing.....	60
Figura 29: Esquema General de Tecnología .....	61
Figura 30: Diagrama de flujo de la actividad principal .....	63
Figura 31: Diagrama de flujo fragment exteriores e interiores .....	64
Figura 32: Diagrama de flujo fragment redes .....	65
Figura 33: Escenario para el envío y adquisición de datos desarrollado .....	66
Figura 34: Tablas pertenecientes a la base de datos creada .....	67
Figura 35: Diagrama de secuencias envío y recepción de datos .....	67
Figura 36: Diagrama de flujo para agregar marcadores en el mapa web.....	68
Figura 37: Algoritmo utilizado para realizar copias de seguridad .....	71
Figura 38: Escenario para adquisición del parámetro RSSI.....	75
Figura 39: Resultados obtenidos de la consulta SQL .....	77
Figura 40: Número de datos enviados por los participantes .....	78
Figura 41: Comparación Tiempo de envío de datos GPS vs A-GPS .....	79
Figura 42: Ubicación de referencia en el lugar de pruebas.....	80
Figura 43: Comparación de Latitud (GPS – GPS-Asistido) .....	82

Figura 44: Comparación de Longitud (GPS – GPS-Asistido) .....	82
Figura 45: Marcadores en el mapa utilizando GPS .....	83
Figura 46: Marcadores en el mapa utilizando GPS - Asistido .....	83
Figura 47: Variación de RSSI respecto a la distancia.....	85
Figura 48: Protocolo HTTP utilizado para la comunicación .....	86
Figura 49: Paquetes / segundo enviados (Exteriores).....	87
Figura 50: Protocolo HTTP en pruebas (exteriores).....	87
Figura 51: Paquetes / segundo enviados (Interiores).....	88
Figura 52: Flujo de datos .....	88
Figura 53: Estadísticas de tráfico mes de enero 2017.....	89
Figura 54: Estadísticas de tráfico mes de febrero 2017 hasta el 12-02-2017.....	89
Figura 55: Estadísticas de Transferencias en el mes de enero .....	90
Figura 56: Estadísticas de transferencias en el mes de febrero hasta 12-02-2017 .....	90
Figura 57: Página de inicio de la Aplicación Web .....	92
Figura 58: Cobertura de la Red Zona Libre a) 13-12-2016 y b) 12-02-2017 .....	92
Figura 59: Cobertura red Zona Libre desde diciembre 2016 hasta febrero 2017.....	93
Figura 60: Elementos para visualizar cobertura en interiores .....	94
Figura 61: Cobertura red Espe piso 2 a) 13-12-2016 y b) 12-02-2017.....	94
Figura 62: Cobertura red Zona Libre en dispositivo móvil (Exteriores) .....	95
Figura 63: Cobertura red Espe piso 2 en dispositivo móvil (Interiores).....	95

## RESUMEN

El presente proyecto consiste en el diseño e implementación de una aplicación que permite visualizar en el mapa, el área de cobertura inalámbrica de las redes Wi-Fi en la Universidad de las Fuerzas Armadas - ESPE tanto en interiores como exteriores a través de un trabajo colaborativo (*Crowdsensing*). Mediante *Crowdsensing*, es posible utilizar los sensores que están disponibles en los dispositivos móviles para la adquisición de información; de esta manera los usuarios pueden obtener datos para enviarlos a un servidor web que permita graficar y visualizar los mapas de cobertura inalámbrica en tiempo real. Para cumplir con este propósito, se desarrolló una aplicación móvil para el Sistema Operativo Android que permite capturar los datos necesarios: latitud, longitud, RSSI (parámetro de intensidad de señal), identificador del dispositivo, fecha y enviarlos al servidor web donde se procesa la información y permite graficar marcadores de distintos colores en el mapa obteniendo así un gráfico de cobertura de las redes disponibles en la Universidad. Los mapas de cobertura obtenidos están disponibles en la web para cualquier usuario ya sea a través de la aplicación móvil o mediante un dispositivo con un navegador web ingresando a la página: [alonsoalejandro.webcindario.com](http://alonsoalejandro.webcindario.com). Con todos los elementos mencionados anteriormente se logró obtener los mapas de cobertura de la Universidad de las Fuerzas Armadas – ESPE permitiendo conocer a todos los usuarios las áreas donde hay una mejor cobertura de las redes inalámbricas.

### **PALABRAS CLAVE:**

- **CROWDSENSING**
- **ANDROID**
- **RSSI**
- **WI-FI**
- **UBICACIÓN**

## ABSTRACT

The aim of this project is to design and implement an application that allows to visualize the wireless coverage area of the Wi-Fi networks at the Universidad de las Fuerzas Armadas - ESPE both indoors and outdoors on the map by a collaborative work (*Crowdsensing*).

By *Crowdsensing*, it is possible to use the sensors that are available in the mobile devices for the acquisition of information; in this way users can obtain data to send them to a web server that allows to graph and visualize the wireless coverage maps in real time.

To achieve this goal, a mobile application was developed for the Android Operating System that allows to capture the necessary data: latitude, longitude, RSSI (signal strength parameter), device identifier, date and send them to the web server where the information is processed and allows to graph markers of different colors in the map obtaining a graph of coverage of the networks available in the University.

The coverage maps obtained are available on the web for any user either through the mobile application or through a device with a web browser entering the page: [alonsoalejandro.webcindario.com](http://alonsoalejandro.webcindario.com)

With all the elements mentioned before, it was possible to obtain coverage maps of the University of the Armed Forces - ESPE, allowing all users to know the areas where there is better coverage of wireless networks.

### KEY WORDS:

- CROWDSENSING
- ANDROID
- RSSI
- WI-FI
- UBICATION

## CAPITULO 1

### INTRODUCCIÓN

#### 1.1 Antecedentes

En la actualidad en un mundo que avanza muy rápido tecnológicamente, es necesario permanecer conectados a la Internet por muchas razones ya sea por educación, trabajo, ocio, para obtener nuevos conocimientos o simplemente para permanecer en contacto con el mundo exterior. De igual manera se busca mejorar las condiciones de vida de las personas brindando un mayor grado de confort a los usuarios finales supliendo sus necesidades relacionadas con el acceso a la red.

Como en cualquier parte del mundo, en nuestro país existe la necesidad para el usuario de encontrar un lugar donde exista la mejor zona de cobertura de la red inalámbrica para satisfacer sus necesidades propias. A través de *Crowdsourcing* (colaboración masiva), y mediante los dispositivos móviles es posible optimizar tiempo y esfuerzo encontrando un lugar apropiado para permanecer conectados a la red.

*Crowdsourcing* utiliza distintos dispositivos (smartphones, sensores vehiculares, etc) o herramientas (redes sociales) para obtener información y contribuir con una tarea social (Bustos Jiménez, 2015). La práctica de una colaboración masiva no es reciente y ésta se remonta mucho antes en la historia de la humanidad como nos indica la antropología donde podríamos encontrar numerosos ejemplos de este tipo de prácticas colaborativas a lo largo del tiempo (Lara, 2014).

En la actualidad con el uso de herramientas tecnológicas de información y comunicación como la Internet, las redes de datos y dispositivos de telecomunicaciones móviles, es posible el acceso a la información en prácticamente cualquier lugar del planeta, originando la formación de comunidades de interés y un trabajo colaborativo de las tareas a través de todos los miembros existentes.

Puesto que los teléfonos inteligentes cuentan con varios sensores como: cámara, micrófono, acelerómetros digitales, sensor de luz, proximidad y otros que serán incluidos en el futuro para la monitorización de contaminación, salud, etc; se amplía la posibilidad de utilizar estos teléfonos en aplicaciones *Crowdsourcing* y así obtener datos en tiempo real, que en conjunto puedan beneficiar brindando información útil al mayor número de usuarios posibles.

## **1.2 Justificación e Importancia**

Con el avance de la tecnología, se busca mejorar las condiciones de vida, ofreciendo mayor seguridad, confiabilidad y brindando un mayor grado de confort a los usuarios finales; los dispositivos móviles han tenido un gran incremento en los últimos años y se han convertido en dispositivos centrales de cómputo y comunicación en la vida de las personas, que al disponer de varios elementos de sensado, han creado la oportunidad de desarrollar aplicaciones para aumentar las capacidades de recolección de la información.

Es posible utilizar los dispositivos móviles para la monitorización de un individuo, por ejemplo en el ámbito de la salud, existen sensores para medir el nivel de estrés y ritmo cardíaco (sensor cardíaco), número de pasos (podómetro), llevar un registro de los alimentos y agua consumidos cada día o medir parámetros de salud como peso, sueño, número de calorías que son utilizados únicamente por la propia persona; también es posible usar los dispositivos móviles para monitorizar un fenómeno a una escala mayor que no puede ser medida usando información de un solo individuo ya sea por tiempo, costo o dificultad, en cuyo caso se habla de sensado grupal o *Mobile Crowdsensing* (Arreondo, 2015).



El propósito es recolectar una mayor cantidad de información proveniente de un amplio grupo de personas a través de sus dispositivos móviles para analizar y usar la misma en beneficio del grupo; en la actualidad se utiliza *Crowdsensing* para obtener datos en tiempo real y utilizarlos en aplicaciones relacionadas con los sistemas de transporte inteligente, monitorización de tráfico y contaminación, calidad de aire, cubrimiento de redes celulares, etc.

En el caso de la red inalámbrica de la Universidad de las Fuerzas Armadas – ESPE, esta consta de varios puntos de acceso y zonas de cobertura a lo largo del campus universitario, sin embargo, se ha evidenciado la necesidad de la comunidad universitaria de identificar aquellas zonas que presentan la mejor cobertura de red inalámbrica para brindar un servicio de calidad a la misma.

Es por ese motivo, que el tema de investigación, se centra en la identificación de las zonas de cobertura y en el análisis de calidad de la señal Wi-Fi en la Universidad de las Fuerzas Armadas – ESPE a través de una aplicación móvil *Crowdsensing* con la finalidad de recolectar datos de latitud, longitud, parámetro RSSI (parámetro de intensidad de señal), identificador del equipo y fecha en los dispositivos móviles de un amplio grupo de estudiantes, analizarlos y obtener las gráficas de cobertura correspondientes.

El proyecto a desarrollar es un proyecto de innovación puesto que los estudiantes son los principales colaboradores cuando se trata de recolectar datos ya que sólo se necesita instalar la aplicación en los dispositivos móviles y automáticamente se van enviando datos útiles a un servidor para su respectivo análisis; así mismo los estudiantes son los principales beneficiarios puesto que tienen a su alcance los mapas de cobertura inalámbrica de la Universidad en tiempo real.

Este proyecto es importante porque busca identificar en el mapa, la cobertura de las redes inalámbricas Wi-Fi en la Universidad de las Fuerzas Armadas – ESPE puesto que en el campus universitario no se cuenta con un zona clara donde exista una mejor cobertura, es por este motivo que se desarrolla el proyecto de investigación para proporcionar datos útiles a todas las personas que requieran una mejor calidad de señal, para ello, se desarrolla una aplicación móvil que permita enviar los datos del estado de

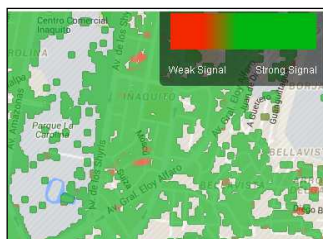
las redes inalámbricas a una base de datos para posteriormente poder visualizar la cobertura en tiempo real en la Universidad.

Cabe mencionar que no existen otras aplicaciones que indiquen en el mapa la cobertura de las redes Wi-Fi por lo que este proyecto puede ser utilizado como referencia para proyectos futuros que requieran identificar dicha cobertura en el mapa, además al tratarse de una aplicación gratuita, no se requiere de ningún pago para utilizar todas sus funcionalidades.

Con el desarrollo de este proyecto y la participación de la comunidad universitaria, será posible obtener resultados sobre el estado de las redes inalámbricas en la Universidad de las Fuerzas Armadas - ESPE, además permitirá al personal de la Dirección de Tecnologías de Información conocer zonas negras o de baja cobertura, y crear planes de mejora de la red, lo cual beneficia directamente a la comunidad universitaria al brindar un mejor servicio.

### 1.3 Alcance del Proyecto

Este proyecto busca solucionar el problema de los estudiantes de la Universidad de las Fuerzas Armadas - ESPE en torno a la identificación de zonas de cobertura inalámbrica Wi-Fi permitiendo visualizar las zonas con acceso a la red y mostrando gráficamente los niveles de señal en el mapa del campus universitario vía web como se muestra en la Figura 1; para ello se plantea el uso de herramientas de hardware y software.



**Figura 1: Ejemplo de mapa de cobertura**  
**Fuente:** (Opensignal, 2016)

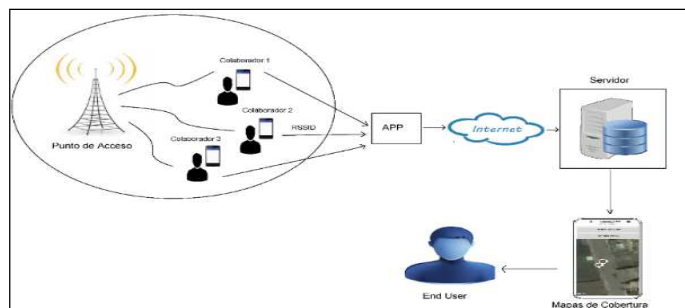
Desde la parte de software se pretende diseñar una aplicación móvil basada en Android como se muestra en la Figura 2 que permita obtener información de la potencia de la señal inalámbrica en el dispositivo y además permita validar el envío y recepción de datos a través de la conexión de red, representando esta información en un mapa de cobertura.

Dicha aplicación será entregada e instalada en los dispositivos móviles de al menos 20 personas, para que en base a *Crowdsensing* se pueda obtener la información necesaria y graficar los mapas de cobertura con múltiples muestras.



**Figura 2: Ejemplo de Aplicación Móvil en un Teléfono Celular**

En cuanto al hardware, se propone trabajar usando los sensores incorporados en los dispositivos móviles y junto con la aplicación desarrollada enviar los datos recolectados a un servidor para el análisis e interpretación de los mismos y de esta manera lograr la correspondiente visualización de zonas de cobertura y niveles de la señal de potencia que beneficiará a todos los usuarios que necesiten cobertura inalámbrica.



**Figura 3: Diagrama de Funcionamiento**

En la Figura 3, se muestra el Diagrama de Funcionamiento; los colaboradores se conectan a una red que existe en la Universidad de las Fuerzas Armadas – ESPE (Zona Libre, Estudiantes) con lo cual enviarán los datos de latitud, longitud y potencia de la señal a un servidor y se obtiene como resultado final el mapa de cobertura de red.

Cabe indicar que la Unidad de Tecnología de la Información y Comunicación de la Universidad, no dispone de mapas de nivel de cobertura de las redes inalámbricas, pero si posee diagramas de ubicación de los puntos de acceso, los cuales se solicitan para levantar la información y crear los mapas de cobertura propuestos.

## **1.4 Objetivos**

### **1.4.1 General**

Identificar el área de cobertura de las redes inalámbricas Wi-Fi de la Universidad de las Fuerzas Armadas –ESPE en mapas en tiempo real, a través de una aplicación móvil en Android basada en *Crowdsensing*.

### **1.4.2 Específicos**

- Analizar el estado del arte y la arquitectura *Crowdsensing*.
- Elegir un Sistema Operativo apropiado para desarrollar la aplicación móvil utilizando todas sus características y funcionalidades para una correcta visualización de los resultados.
- Conocer los parámetros necesarios que serán enviados al servidor para su posterior análisis.
- Comunicar la base de datos del servidor web con la aplicación móvil, para crear el sistema de sensado y muestra de la información.
- Generar los reportes de cobertura en base a los datos recolectados.

## CAPITULO 2

### MARCO TEÓRICO

#### 2.1 Crowdsourcing

El término *Crowdsourcing* fue inicialmente propuesto por Jeff Howe y Mark Robinson (Guo, Yu, Zhou, & Zhang, 2014). Etimológicamente está compuesto por “*Crowd*” que hace referencia a multitud y “*Sourcing*” que se refiere a aprovisionamiento u obtención de materia prima, es por esto que se podría traducir al español como colaboración abierta distribuida (Lara, 2014).

Entonces, *Crowdsourcing* hace referencia a un trabajo colaborativo de muchas personas para resolver problemas y contribuir a encontrar determinadas soluciones a los mismos.

Dentro de *Crowdsourcing*, es posible adquirir datos útiles mediante sensores incorporados en los dispositivos móviles, en este caso se habla de *Crowdsensing* (Sun & Ma, 2014).

Debido al gran número de dispositivos móviles existentes en la actualidad como teléfonos inteligentes o tablets y gracias a su movilidad, es posible adquirir datos mediante los sensores incorporados y además compartir la información con el entorno social (Tanas & Herrera, 2012). La información recopilada puede ser enviada a través de la Internet permitiendo a *Mobile Crowdsensing* adaptarse a las distintas necesidades de los usuarios.

Los desafíos que surgen con respecto a *Mobile Crowdsensing* se basan en los estilos de recopilación de datos, mecanismos de incentivos, la calidad de la información aportada por los usuarios, entre otros.

### **2.1.1 Conceptos y Características de Crowdsensing**

Para que un sistema pueda ser considerado para aplicaciones *Crowdsensing*, se requieren de tres elementos fundamentales (Angelopoulos, Nikolettseas, Raptis, & Rolim, 2014):

1.- La Multitud: Se refiere a todo el conjunto de dispositivos dentro de un área de interés, equipados con sensores para la recolección de datos y que son transportados por personas.

2.- El Servidor: Es un elemento que recopila la información obtenida y utiliza la lógica computacional para realizar una tarea.

3.-Un conjunto de reglas que rigen la interacción entre la multitud y el servidor, por ejemplo, los mecanismos de incentivos, las políticas de trabajo, los resultados esperados.

### **2.1.2 Aplicaciones para Mobile Crowdsensing**

Las aplicaciones existentes para *Mobile Crowdsensing* son tan numerosas y variables como el número de sensores incorporados en los dispositivos móviles, así, éstas aplicaciones se pueden clasificar en distintas áreas como: videojuegos, aplicaciones de salud y bienestar, infraestructura y monitorización ambiental, vigilancia, notificaciones y respuestas de emergencia en redes sociales, aplicaciones de transporte, aplicaciones de marketing y publicidad entre otras, donde ahora las personas son las protagonistas principales al ser las portadoras de los dispositivos para

sensado que también pueden acceder a sus recursos para obtener la información requerida (Mahmud & Aris, 2015).

Uno de los proyectos más conocidos de *Mobile Crowdsensing*, por ejemplo, es: **Waze**. Según el sitio oficial, *Waze* funciona como una aplicación de navegación basada en el GPS (Sistema de Posicionamiento Global) que muestra en la pantalla los detalles de la ruta, tiempos de viaje, reportes de tráfico, alertas que incluyen accidentes, peligros, controles policiales, cierres viales, entre otras características que requieren de la colaboración de los miembros que forman parte de esta comunidad (Waze, 2016).

Otra aplicación *Crowdsensing* es **WeatherSignal** la misma que permite monitorizar las condiciones climáticas en una ciudad para luego mostrarlas en un mapa climático actualizado en tiempo real; para que esta aplicación funcione correctamente, es necesario que existan suficientes usuarios que aporten datos reales (WeatherSignal, 2016).

Otro ejemplo es Google **Crowdsourcing**, que permite a los usuarios realizar tareas relacionadas con traducción de contenido, buscando mejorar los resultados del traductor de Google para obtener una mejor interpretación, así como una traducción más específica (Crowdsourcing, 2016).

### **2.1.3 Ventajas y Desventajas de Crowdsensing**

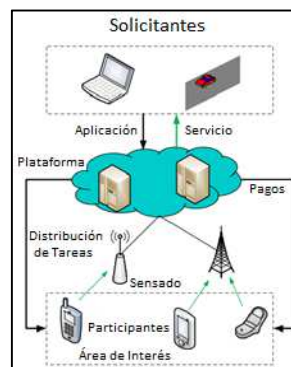
Los beneficios de *Crowdsensing* son: disminución de costos, puesto que son los participantes los encargados de recolectar los datos sin necesidad de una remuneración económica, otra ventaja es el conjunto enorme de participantes permitiendo mejorar la fiabilidad (Vukovic, Laredo, & Rajagopal, 2010). Así también se obtiene una gran ventaja en cuanto a calidad entendiéndola como el número de tareas que se realizan y puesto que se incrementa el número de participantes, también se incrementa el número de posibles soluciones a un determinado problema (Schenk & Guittard, 2009).

Así mismo existen desventajas que conlleva *Crowdsensing*, como el tiempo requerido que implica la creación de una plataforma colaborativa, la poca relevancia

que puede tener la información, los problemas legales relacionados con la propiedad intelectual de ideas utilizadas para iniciativas *Crowdsourcing*, entre otras (Whitla, 2009).

### 2.1.4 Arquitectura Crowdsensing

Los elementos que interactúan en una arquitectura *Crowdsensing* se pueden observar en la Figura 4: *Requestes* (solicitantes), *Participants* (participantes) y finalmente la Plataforma de *Crowdsensing* encargada de procesar los datos (Sun & Ma, 2014).



**Figura 4: Arquitectura Crowdsensing**

**Fuente:** (Sun & Ma, 2014)

**Participantes:** Son todos los dispositivos que disponen de un sensor capaz de recopilar datos y enviarlos a la plataforma para su procesamiento.

**Solicitantes:** Son todos los usuarios que utilizan la aplicación. El objetivo fundamental en esta parte de la arquitectura, es dotar un sistema fiable para la representación de los datos obtenidos que se ajuste a las necesidades de cada usuario.

**Plataforma de Crowdsensing:** Procesa los datos obtenidos de los participantes para enviarlos a los solicitantes según sus requerimientos (Bustos Jiménez, 2015).



## 2.2 Dispositivos Móviles

Un dispositivo móvil también llamado computadora de mano (handheld computer), es un dispositivo que ha sido diseñado para ser extremadamente portátil y que generalmente puede caber en la mano (GCF, 2016).



### 2.2.1 Tipos de Dispositivos Móviles

Existe una gran cantidad de dispositivos electrónicos que pueden ser clasificados como dispositivos móviles (Pozo, 2011). Entre los más conocidos están:

- *Smartphones* (Teléfonos Inteligentes)
- *Tablets*
- Agendas Digitales
- Computadores Portátiles (Laptop)

Sin embargo, para el proyecto de investigación solo se enfocará en Teléfonos Inteligentes y Tablets como se muestra en la Tabla 1.

**Tabla 1**  
**Dispositivos móviles más conocidos en la actualidad**

Tipo	Características	Imagen
Smartphones (Teléfono Inteligente)	Permiten la comunicación a través de Bluetooth, Wifi, conexión a la Internet, Usb. Se rigen por un Sistema Operativo. Permiten guardar información e instalar nuevos programas (aplicaciones). Diseñados para acceder de manera continua a la Internet.	
Tablets	Generalmente son de mayor tamaño que un Smartphone. Posee un hardware más poderoso de procesamiento, capacidades de reproducción multimedia, pantalla multitoque, y gran duración de la batería.	

**Fuente:** (Pozo, 2011)

### 2.2.2 Tipos de Terminales en los Dispositivos Móviles

Para orientar a los usuarios en la valoración de los dispositivos móviles, se ha establecido tres grupos definidos de dispositivos móviles: gama alta, media y baja, pese a que no existen identificadores para ubicar a cada dispositivo dentro de una gama, hay especificaciones mínimas que deben cumplir para ser considerado dentro de alguna (Conde, 2013). Así se establece la Tabla 2:

**Tabla 2**  
**Tipos de gamas**

<b>Gamas</b>	<b>Características</b>
Gama Baja	Precio muy económico. Por lo general son de tamaño pequeño. Tienen funciones básicas y características limitadas.
Gama Media	Precio moderado. Suelen ser muy equilibrados en sus características. Fabricados con materiales de calidad aceptable. Poseen algunas funciones de dispositivos de gama alta. Tienen soporte limitado.
Gama Alta	Precio elevado. Destaca la eficiencia energética. Pueden conectarse con una gran variedad de dispositivos. Generalmente son delgados, poseen cámaras de alta resolución y gran espacio de almacenamiento interno. Diseñados para redes de última generación.

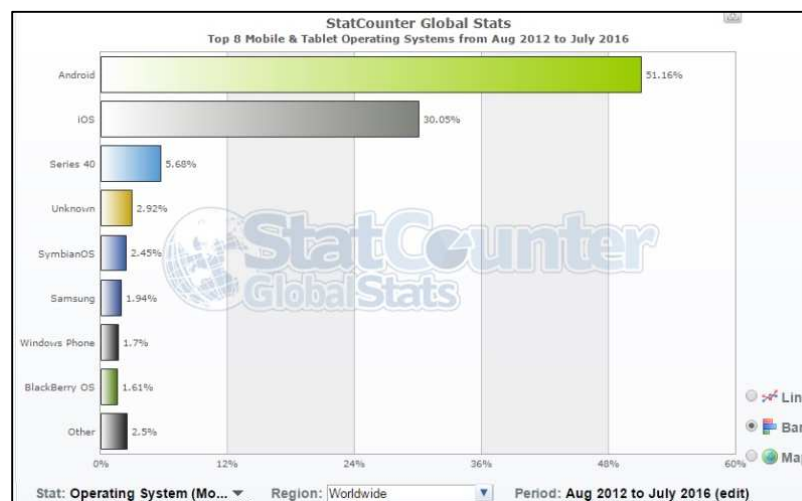
**Fuente:** (SmartFix, s.f.)

Considerando las características de los dispositivos móviles, a la hora de recolectar datos, es necesario que todos dispongan de ciertas especificaciones que incluyen:

- Portabilidad, fácil manejo.
- Disponer de memoria (Interna, Externa) con el fin de gestionar las aplicaciones para que funcionen con normalidad.
- Capacidad de conectarse a redes inalámbricas y celulares.
- Capacidad de interacción con el usuario mediante la pantalla, teclado, voz, etc.
- Disponer de sensores incorporados para la recolección de datos.

### 2.2.3 Mercado de los Sistemas Operativos

En un segmento tan competitivo como es el de los dispositivos móviles inteligentes y dada la gran variedad de los mismos, es fundamental elegir un Sistema Operativo acorde con las capacidades del dispositivo móvil que permita realizar las tareas con mayor eficacia siempre buscando brindar una mejor experiencia al usuario.



**Figura 5: Sistemas Operativos más utilizados**

**Fuente:** (StatCounter, 2016)

Como se observa en la Figura 5, los Sistemas Operativos que dominan el mercado son Android e IOS existiendo otros con menor participación en el mercado (StatCounter, 2016).

Es por este motivo que se ha elegido utilizar el Sistema Operativo Android para el desarrollo del proyecto puesto que posee características únicas como: ser *open-source*, estar disponible para una enorme cantidad de dispositivos, menores costos al momento de publicar la aplicación y tener un gran soporte en la web.

### 2.2.4 Tipos de Sensores en Dispositivos Móviles

Dependiendo de la gama o tipo de dispositivos móviles, éstos pueden tener varios tipos de sensores, los más comunes se muestran en la Tabla 3:

**Tabla 3**  
**Tipos de Sensores en Dispositivos Móviles**

Tipos de Sensores	Características
Sensores de Pantalla	Utilizados para el manejo de pantallas táctiles
Acelerómetro	Detecta la posición del dispositivo respecto a tres ejes.
Giroscopio	Determina el giro de un dispositivo de manera exacta.
Magnetómetro	Permiten detectar campos magnéticos.
Sensor de Luz	Su uso permite regular la luminosidad de la pantalla.
Podómetro	Sirve para detectar los pasos que da el usuario al caminar.
Barómetro	Permite determinar la presión atmosférica en un punto.

**Fuente:** (Calzado, 2017)

### 2.3 Sistema Operativo Android

Android es un Sistema Operativo que está basado en el Kernel de Linux, un núcleo de Sistema Operativo libre, gratuito y multiplataforma; este sistema operativo fue diseñado fundamentalmente para dispositivos móviles con pantalla táctil (tablets, teléfonos inteligentes) aunque en la actualidad su uso se ha extendido a televisores, reproductores mp3, computadores, relojes inteligentes, automóviles, etc.

El Sistema Operativo Android fue desarrollado inicialmente por Android Inc., empresa respaldada por Google, que posteriormente la adquirió en 2005. Para el 2007 Android es presentado junto al “Open Handset Alliance”, una alianza comercial de compañías que desarrollan estándares abiertos para dispositivos móviles y que en la actualidad cuentan con 84 miembros entre los que destacan Toshiba, Dell, Sony, LG, Samsung Electronics, Motorola Mobility, Huawei, Acer, Intel Corporation, Nvidia Corporation, Texas Instruments, Google, eBay, Acces(Compañías de Software), Telefónica, Vodafone, T-Mobile (Operadores) (alliance, 2007).

Desde su lanzamiento inicial, Android ha presentado numerosas actualizaciones del Sistema Operativo con la finalidad de resolver cualquier problema en el código, para mejorar la apariencia o agregar nuevas funcionalidades; los nombres de las actualizaciones tienen nombres de postres en inglés y están en orden alfabético como se puede ver en la Tabla 4:

**Tabla 4**  
**Dispositivos móviles más conocidos en la actualidad**

Número de Versión	Nombre	Fecha de Lanzamiento	Nivel de API
1.0	Apple Pie	23 de septiembre de 2008	1
1.1	Banana Bread	9 de febrero de 2009	2
1.5	Cupcake	27 de abril de 2009	3
1.6	Donut	15 de septiembre de 2009	4
2.0-2.1.X	Eclair	26 de octubre de 2009	5-7
2.2.X	Froyo	20 de mayo de 2010	8
2.3-2.3.4	Gingerbread	6 de diciembre de 2010	9-10
3.0.X-3.2	Honeycomb	22 de febrero de 2011	11-13
4.0-4.0.4	Ice Cream Sandwich	18 de octubre de 2011	14-15
4.1-4.3	Jelly Bean	9 de julio de 2012	16-18
4.4-4.4.W	KitKat	31 de octubre de 2013	19-20
5.0-5.1	Lollipop	12 de noviembre de 2014	21-22
6.0	Marshmallow	5 de octubre de 2015	23
7.0	Nougat	22 de agosto de 2016	24-25

**Fuente:** (Developers A. , 2016)

### 2.3.1 Programación en Android

Todas las aplicaciones que se realizan para el Sistema Operativo Android se programan en lenguaje Java y se utiliza la máquina virtual (Dalvik) para interpretar y ejecutar el código escrito.

La programación para el Sistema Operativo Android busca recopilar en una misma plataforma los recursos que permitan al desarrollador aprovechar al máximo las características de los dispositivos móviles existentes en el mercado (Wi-Fi, Bluetooth, GPS, cámara, sensores, llamadas, mensajes) y su comunicación con la Web, para de esta manera crear aplicaciones portables, reutilizables y útiles para todos los usuarios.

### 2.3.2 Interfaz de Programación de Aplicaciones (API)

Una API (Interfaz de Programación de Aplicaciones) es un conjunto de comandos, funciones y programas informáticos que permiten crear programas específicos para ciertos sistemas operativos. La utilización de las API facilita en gran medida el trabajo de los desarrolladores ya que permiten usar funciones predefinidas para interactuar con el sistema operativo o con otro programa sin tener que escribir nuevamente el código desde cero (Álvarez E. , 2015).

Existen distintos niveles de API cada uno de los cuales ofrece nuevas funcionalidades y capacidades, las principales API se describen en la Tabla 5:

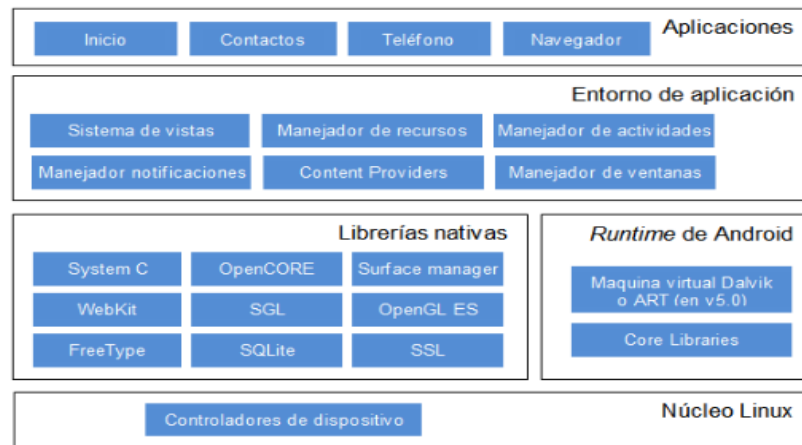
**Tabla 5**  
**Principales APIs Android**

<b>Nombre</b>	<b>Características</b>
<b>Activity Manager</b>	Gestionan el ciclo de vida de las aplicaciones en Android.
<b>Window Manager</b>	Gestiona las ventanas de las aplicaciones.
<b>Telephony Manager</b>	Proporciona acceso a la información sobre los servicios de telefonía en el dispositivo. (Llamadas, mensajes, etc).
<b>Location Manager</b>	Proporciona acceso a los servicios de localización del sistema. Estos servicios permiten a las aplicaciones para obtener actualizaciones periódicas de la situación geográfica del dispositivo, o para disparar un Intento especificado por la aplicación cuando el dispositivo entra en la proximidad de una zona geográfica determinada.
<b>Wifi Manager</b>	Esta clase proporciona la API principal para administrar todos los aspectos de conectividad Wi - Fi
<b>Notification Manager</b>	Notifica al usuario de los acontecimientos que suceden (una llamada entrante, un mensaje recibido, conexión Wi-Fi disponible, etc). También notifica si algo ha sucedido en segundo plano.

**Fuente:** (Developers A. , 2016)

### 2.3.3 Arquitectura del Sistema Operativo Android

La arquitectura del Sistema Operativo Android se puede dividir en capas puesto que es una de las técnicas más comunes para construir sistemas complejos, como se puede observar en la Figura 6, Android se ha dividido en 5 capas.



**Figura 6: Arquitectura de Android**

**Fuente:** (Tomás, 2015)

**Núcleo:** Es la primera capa que tiene el Sistema Operativo, Android utiliza el núcleo de Linux 2.6.

El núcleo o Kernel es la parte más importante del Sistema Operativo y se encarga de manejar toda la compatibilidad entre el hardware, es decir contiene todos los drivers necesarios para que cualquier componente de hardware del teléfono (display, cámara, bluetooth, etc) pueda ser utilizado.

Linux fue elegido principalmente por su característica de ser de código abierto y libre; además porque el núcleo de Linux de por sí tiene numerosas ventajas como manejo eficiente de la memoria, manejo de seguridad, implementación completa de los diferentes protocolos y estándares de red, manejo de drivers, manejo de procesos entre otras.

**Capa de Librerías:** Fueron construidas con C/C++ y son utilizadas en varios componentes de Android, en la Tabla 6, se muestran las principales librerías.

**Tabla 6**  
**Librerías Nativas Android**

<b>Librería</b>	<b>Características</b>
System C library	Es una librería adaptada para dispositivos embebidos basados en Linux.
SQLite	Indispensable para la creación y manejo de bases de datos relacionales permitiendo a las aplicaciones manejar estructuras de datos.
Surface Manager	Encargada de componer los elementos de navegación de pantalla.
Media Framework	Librería encargada del soporte de códecs para reproducción y grabación de audio, video e imágenes.
WebKit/Chromium	Proporciona herramientas para navegar por la web.
Open GL/SL y SGL	Son librerías que permiten el manejo de gráficas en 2D, 3D así como aplicaciones que combinen gráficos en 2D y 3D.
FreeType	Utilizada para generar imágenes en mapas de bits.
SSL	Permite establecer conexiones seguras a través de servicios de encriptación (Secure Socket Layer).

**Fuente:** (Tomás, 2015)

**Entorno de Ejecución:** Se sitúa en el mismo nivel que se sitúan las librerías de Android y está compuesto de Core Libraries (librerías con multitud de clases Java) y la Máquina Virtual.

La máquina virtual se encarga de proporcionar un entorno de ejecución independiente de la plataforma de hardware y Sistema Operativo, es decir permite que un programa siempre se ejecute de la misma forma sobre cualquier plataforma.

Debido a las limitaciones de los primeros dispositivos móviles (baja memoria, menor capacidad de procesamiento, etc), no es posible utilizar la máquina virtual Java estándar en los mismos, es por este motivo que Google creó una nueva máquina virtual (Dalvik) que se adapta a las necesidades de estos dispositivos (Tomás, 2015).

La Máquina Virtual Dalvik transforma el código en otro más eficiente pensado en la capacidad de procesamiento de los dispositivos móviles, es así que los ficheros de Java se compilan en ficheros DEX; después se comprime este resultado en un único archivo de extensión APK (Android Package) que se enviará al dispositivo móvil.



**Capa Entorno de Aplicación (Application Framework):** Se puede definir como el conjunto de herramientas para desarrollar cualquier aplicación.

Es posible acceder al conjunto de funciones del Sistema Operativo Android mediante las API escritas en lenguaje Java.

**Capa de Aplicaciones (Application Layer):** Es la última capa dentro de la arquitectura del Sistema Operativo y es la capa donde interactúa el usuario final teniendo el acceso a todas las aplicaciones. Todas estas aplicaciones utilizan servicios, API y librerías de los niveles anteriores para su funcionamiento.

Los dos últimos niveles de la Arquitectura Android (capa Entorno de Aplicación y Capa de Aplicaciones) están escritos enteramente en lenguaje Java.

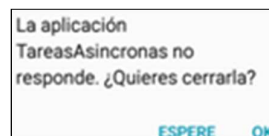
#### 2.3.4 Componentes de una Aplicación Android

Toda aplicación Android está formada por cuatro componentes principales: **Activity, Broadcast Receiver, Service, Content Provider** los que deberán ser declarados de forma explícita en un fichero con formato XML denominado “AndroidManifest.xml” que es donde se definen los componentes de la aplicación y contiene otros datos necesarios como permisos, valores globales, clases que implementa, así como *intent-filters*, que permiten definir el ámbito (clase) que se va a ejecutar en primer lugar.

Dentro de la *activity*, es posible utilizar **fragments**, los mismos que interactúan entre sí permitiendo combinar múltiples *fragments* en una sola actividad para crear una interfaz de usuario multipanel.

También es necesario tener en cuenta el concepto de **hilo** que se define como una secuencia de tareas ligadas que pueden ser ejecutadas por un Sistema Operativo ya que cuando se realiza la programación en Android todos los componentes se ejecutan en el hilo principal (UI Thread, Main Thread) que corresponde a las tareas llevadas a cabo en la interfaz de usuario de la aplicación.

En un proyecto, cualquier instrucción larga o que requiera mucho tiempo puede llegar a bloquear la interfaz de usuario, causar lentitud en el dispositivo o generar un malfuncionamiento del mismo con la respectiva terminación abrupta de la aplicación o la aparición del Diálogo ANR (Application not respond) como se ve en la Figura 7:



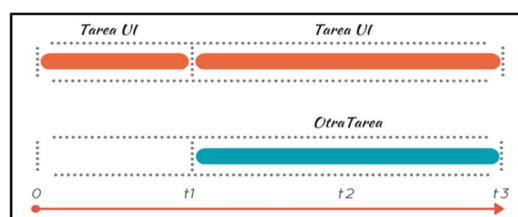
**Figura 7: Diálogo ANR**

Cuando la tarea principal es seguida por otra tarea larga como se aprecia en la Figura 8, ésta puede llegar a colgar la aplicación puesto que mientras no finalice esta tarea, no se podrá pasar a la siguiente.



**Figura 8: Línea de tiempo de las distintas tareas realizadas**  
Fuente: (Revelo, 2014)

La solución a este problema es ejecutar la tarea larga en segundo plano evitando que se cuelgue la aplicación como se muestra en la Figura 9.



**Figura 9: Realización de tareas en segundo plano**  
Fuente: (Revelo, 2014)

Para ejecutar tareas en segundo plano, se tiene como opciones la creación de un nuevo **hilo** o utilizar la clase **AsyncTask** proporcionada por Android.

### 2.3.5 Aplicaciones Móviles

Una aplicación móvil es una aplicación informática que ha sido diseñada por desarrolladores para ser ejecutada en dispositivos móviles como teléfonos inteligentes, tablets entre otros y permite a los usuarios efectuar una tarea concreta de tipo educativa, profesional, entretenimiento, acceso a servicios, acceso a la información, etc.

Estas aplicaciones están escritas en un lenguaje de programación compilado y permiten utilizar las características y funcionalidades de los dispositivos móviles.

Para el desarrollo de las aplicaciones móviles en Android, existen en el mercado varios entornos de desarrollo como Android Studio, Eclipse, App Inventor, Basic4Android, etc. De todas ellas, Android Studio fue desarrollada por Google como principal herramienta de programación y es un entorno de desarrollo integrado (IDE) basado en IntelliJ IDEA que cuenta con una licencia de software libre Apache 2.0.

Android al ser una plataforma abierta, posee varias maneras de distribuir una aplicación hasta el usuario final como:

- Google Play: Plataforma de distribución oficial de Google.
- Descarga Directa: A través de correo electrónico o link de descarga donde se ubica la aplicación.

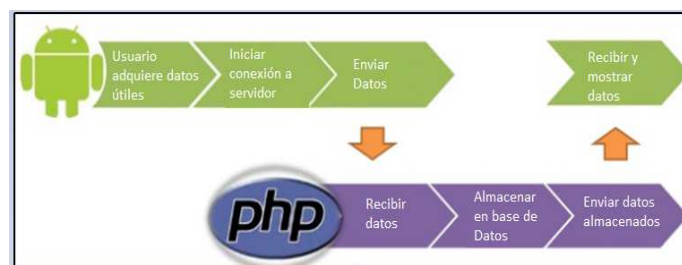
Sin embargo, al utilizar este último método, el sistema identifica la aplicación descargada como no recomendada por lo que hay que permitir en el dispositivo móvil la instalación desde fuentes desconocidas con el fin de permitir la instalación de apps de terceros o en desarrollo (Developers A. , 2016).

El proceso de instalación inicia con la descarga del archivo APK que es el formato de archivo utilizado por el Sistema Operativo Android para luego proceder a instalar la aplicación.

Muchas de las aplicaciones disponibles son gratuitas para el usuario, sin embargo, aplicaciones especializadas o versiones premium tienen un costo. Desde el lado del desarrollador, el costo de publicar una aplicación en Google Play a la fecha es de \$ 25 por abrir una cuenta, con esta cuenta es posible subir la aplicación a Google Play y escoger si la aplicación es gratuita o de pago (Google obtiene el 30% de la facturación total en la venta de la aplicación).

Por otro lado, como se observa en la Figura 10, el Flujo General de una aplicación o todo el proceso desde el envío de datos realizado por los participantes, hasta la visualización de mismos, está compuesto por:

- La aplicación Android alojada en el dispositivo móvil, la cual se encarga de recolectar los datos e iniciar la conexión HTTP con el servidor para enviar los mismos.
- El alojamiento en la web con su respectiva base de datos en la cual se van almacenando todos los datos, para su posterior uso y visualización.



**Figura 10: Flujo General de una Aplicación**  
Fuente: (Ovalle, 2013)

## 2.4 Servicio Web

Se puede definir como un conjunto de aplicaciones o tecnologías con capacidad para interoperar en la web, dichas tecnologías intercambian datos para ofrecer unos

servicios (W3C, Consorcio World Wide Web, s.f.). El servicio web como todos los servicios que ofrece Internet se basa en la arquitectura cliente – servidor.

#### **2.4.1 Arquitectura Cliente – Servidor**

Se puede definir como un modelo de sistemas de información que agrupa varios elementos con la finalidad de repartir el tratamiento de información y datos por todo el sistema informático y de esta manera mejorar el rendimiento del sistema global.

**Cliente:** Son equipos utilizados por los usuarios para conectarse a la red y solicitar servicio a los servidores, los clientes se pueden conectar con los servidores a través de redes LANs o WANs para acceder a distintos servicios.

**Servidor:** Se encarga de atender las necesidades de los clientes y devolverles una respuesta, un servidor es una denominación utilizada para el software como es el caso del programa servidor web Apache o para máquina autónoma equipada con el software del servidor (EmprendedorInnovador, 2005).

Los servidores proveen servicios dentro de una red, los más comunes son: servidor de base de datos, servidor de archivos, servidor de correo, servidor web, servidor de aplicaciones, servidor de juegos, entre otros. Los servidores deben tener una gran capacidad de memoria RAM, un amplio espacio en disco duro, así como deben gozar de monitorización permanente.

Para la ejecución de los servicios, es necesario disponer de programas, así, existen tecnologías de programación tanto del lado del cliente como del lado del servidor.

**Programación del lado del Cliente:** Aunque los programas se encuentran junto a la página web en el servidor, son transferidos al cliente para que este los ejecute.

Ejemplos de lenguajes de programación del lado del cliente son: HTML, CSS, Java, JavaScript, VBScript.

**Programación del lado del Servidor:** Estos programas son ejecutados por un servidor, para posteriormente enviar la respuesta de la ejecución al cliente. Ejemplos de lenguajes del lado del servidor son PHP, Perl, entre otros.

### **Servidores Web**

Un servidor web es una entidad que acepta solicitudes a través del protocolo HTTP u otro similar para generar una respuesta. El servidor web tiene un intérprete HTTP que se mantiene a la espera de peticiones y responde al cliente con el contenido solicitado, el mismo que lo interpreta y lo muestra en pantalla (Cuyo, 2015). Los

Los servidores además tienen un intérprete de otros lenguajes de programación como PHP, ASP, Perl y Ajax que ejecutan el código introducido dentro del código HTML. La gran ventaja de utilizar estos lenguajes es que se ejecutan del lado del servidor con la posibilidad de ejecutar tareas más complejas que pasan inadvertidas para el usuario.

### **Alojamiento Web**

Un alojamiento web es un servicio que es ofrecido por diversas compañías que básicamente alquilan servidores de distintos tipos a los clientes para crear un sitio web.

Un proveedor de alojamiento web utiliza dispositivos de red como routers y switches para enviar solicitudes desde y hacia los servidores, además posee una serie de servidores llamados centros de datos que gestionan el hardware y software para los clientes (EmprendedorInnovador, 2005).

## Bases de Datos

Una base de datos se puede definir como una colección de datos interrelacionados que son almacenados en un soporte informático; entre otras cosas una base de datos permite: almacenar grandes volúmenes de información, realizar consultas, administrar los datos con rapidez, exactitud y fiabilidad. Es importante distinguir la información almacenada (datos restricciones y relaciones) y el conjunto de programas para gestionar la información (SGBD) que actúan de intermediarios entre la información y el usuario.

Para diseñar una base de datos, se suele utilizar modelos gráficos con la finalidad de visualizar las entidades que intervienen y las relaciones entre ellas, siendo uno de los más utilizados el Modelo de Entidad – Relación (Cobo, Gómez, Pérez, & Rocha, 2005).

### Modelo Entidad – Relación en Base de Datos

Este modelo se caracteriza por utilizar símbolos y reglas para representar los datos con sus relaciones de una manera gráfica. Los elementos que componen este modelo son las entidades con sus atributos y las relaciones entre entidades (Álvarez S. , 2007).

Por lo tanto:

- **Entidad:** Es un objeto real o abstracto del cual se desea guardar información y está compuesto por atributos que definen el objeto.
- **Atributo:** Corresponde a cada una de las propiedades de una entidad o relación.
- **Relación:** Es la asociación entre dos o más entidades. Existen diversos tipos de relaciones como se muestran en la Tabla 7:

**Tabla 7**  
**Tipos de Relaciones**

<b>Tipos de Relaciones</b>	<b>Características</b>
Uno a uno	Un atributo de una entidad únicamente puede estar relacionado con un único atributo de otra entidad y viceversa.
Uno a muchos	A un atributo de una entidad le pueden corresponder varios atributos de otra entidad.
Muchos a muchos	Varios atributos de una entidad pueden estar relacionados con varios atributos de otra entidad y viceversa.

**Fuente:** (Álvarez S. , 2007)

## 2.5 Redes Inalámbricas

Una red inalámbrica es un término que se utiliza para designar la conexión de nodos a través de ondas electromagnéticas (radio e infrarrojo) sin la necesidad de una conexión física (Kioskea, 2014). Así mismo las redes inalámbricas presentan algunas ventajas y desventajas como se muestra en la Tabla 8:

**Tabla 8**  
**Ventajas y Desventajas de las redes inalámbricas**

<b>Ventajas</b>	<b>Desventajas</b>
Ausencia de cables físicos	Menor Ancho de Banda
Economía	Inseguridad
Alcance a grandes distancias	Obstrucciones
Permite realizar conexiones múltiples	Interferencias
Facilidad de instalación	Peligro por radiaciones ionizantes

**Fuente:** (Juliá, s.f.)

Para que las redes inalámbricas pudieran existir sin problemas de compatibilidad, era necesario establecer estándares, es por ello que IEEE (Institute of Electrical and Electronics Engineers) organización dedicada a la estandarización creó un grupo de trabajo específico para esta tarea llamado 802.11, así pues, se definiría con este estándar el uso del nivel físico y de enlace de datos de la red (subcapa MAC), especificando sus normas de funcionamiento.



En 1999, se crea la asociación Wireless Ethernet Compatibility Alliance (WECA) que en el año 2003 pasó a llamarse Wi-Fi Alliance (Wireless Fidelity), con el objetivo de fomentar la tecnología inalámbrica y asegurar la compatibilidad con los equipos. A continuación, se certifica la norma 802.11b permitiendo que todos los equipos certificados con el sello Wi-Fi puedan trabajar juntos sin problema (Remuñán, 2010).

Posteriormente han surgido nuevos estándares inalámbricos proporcionando cada vez mejores prestaciones en cuanto a velocidad y cobertura, los estándares 802.11 se presentan en la Tabla 9.

**Tabla 9**  
**Modificaciones del estándar 802.11**

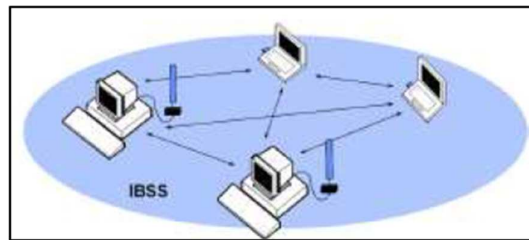
<b>Nombre del Estándar</b>	<b>Velocidad Teórica</b>	<b>Velocidad Práctica</b>	<b>Frecuencia</b>	<b>Ancho de Banda</b>	<b>Año</b>
802.11	2 Mbps	1 Mbps	2,4 GHz	22 MHz	1997
802.11a	54 Mbps	22 Mbps	5,4 GHz	20 MHz	1999
802.11b	11 Mbps	6 Mbps	2,4 GHz	22 MHz	1999
802.11g	54 Mbps	22 Mbps	2,4 GHz	20 MHz	2003
802.11n	600 Mbps	100 Mbps	2,4 GHz; 5,4 GHz	20/40 MHz	2009
802.11 ac	6.93 Gbps	100 Mbps	5,4 GHz	80 o hasta 160 MHz	2013
802.11 ad	7.13 Gbps	Hasta 6 Gbps	60 GHz	2 MHz	2012

**Fuente:** (Carrodegua, 2016)

### 2.5.1 Arquitectura Wlan IEEE 802.11

#### **Independent Basic Service Set (IBSS)**

Los nodos se comunican directamente entre sí, sin la necesidad de un punto de acceso como se puede ver en la Figura 11, cada estación se debe encontrar dentro del rango de señal de otra estación para que exista la comunicación, al no existir dispositivos para su administración (AP), una de las estaciones se encargará de realizar esta tarea, este modo de configuración corresponde a redes Ad-Hoc (Flores Villafuerte & Narváez, 2011).

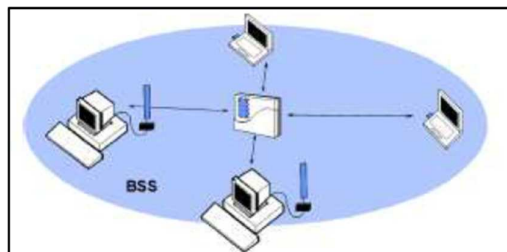


**Figura 11: Dispositivos configurados en modo Ad-Hoc**

**Fuente:** (Flores Villafuerte & Narváez, 2011)

### Basic Service Set (BSS)

Las estaciones se conectan a través de un controlador centralizado, conocido como Punto de Acceso (Access Point - AP), el cual se encarga de realizar las tareas de administración de la red como se observa en la Figura 12.

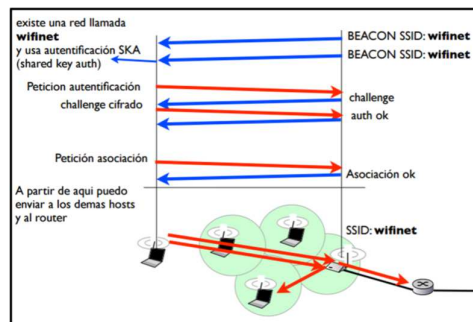


**Figura 12: Dispositivos configurados en modo Infraestructura**

**Fuente:** (Flores Villafuerte & Narváez, 2011)

Para que una estación se conecte con un punto de acceso, son necesarios algunos pasos: escaneo, autenticación, asociación.

En el escaneo la estación busca una trama llamada “Beacon” que es enviada periódicamente por el Access point, una vez elegido el SSID (nombre de la red inalámbrica) se procede con la autenticación y asociación. El proceso de asociación se ve en la Figura 13:



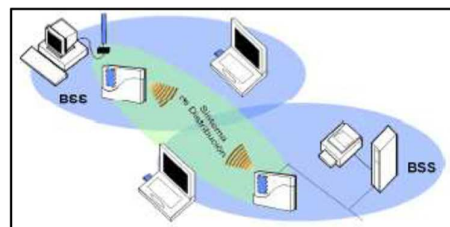
**Figura 13: Proceso de asociación**  
Fuente: (Navarra, 2013)

Cada red BSS tiene asociado un identificador único (**BSSID**) que es publicado por cada punto de acceso y se muestra en el dispositivo del usuario. Este modo de configuración corresponde a redes con infraestructura.

### Extended Service Set (ESS)

Corresponde a varios BSSs interconectados entre sí mediante un sistema de distribución (*Distribution System DS*) la conexión de las redes se puede realizar por medios cableados o inalámbricos, logrando cubrir áreas geográficas mayores.

Las estaciones pueden comunicarse a través de toda la red ESS sin preocuparse de perder la conexión, así mismo un ESS puede proveer acceso a Internet a través de un nodo *gateway*, el cual trabaja como un bridge. Un conjunto de Servicios Extendido se puede observar en la Figura 14.



**Figura 14: Conjunto de Servicios Extendidos**  
Fuente: (Flores Villafuerte & Narváez, 2011)

### **2.5.2 Parámetro de Intensidad de Señal**

El estándar IEEE 802.11 define un mecanismo por el cual se mide la energía RF mediante un circuito en una tarjeta de red inalámbrica. Este parámetro es conocido como RSSI y es un valor numérico entero con un rango admisible de 0 a 255 (1 byte).

Puesto que ningún fabricante ha elegido medir 256 niveles de señal diferentes, cada marca tiene ya especificado su valor máximo de RSSI, por ejemplo, Cisco escoge medir 101 valores separados de energía de RF y su RSSI máximo es de 100; Symbol utiliza un valor máximo de RSSI de 31 mientras que Atheros usa un valor máximo de RSSI de 60 (Bardwell, 2002). Generalmente la escala se expresa dentro de valores negativos, cuanto más negativos, existe una mayor pérdida en la señal.

Puesto que los valores de RSSI son enteros, estos se deben incrementar y disminuir de forma discreta. Por ejemplo, la marca —Symbol provee 32 escalones, mientras que Cisco provee 101, es por este motivo que, si el valor de RSSI cambia en 1, significa que el nivel de potencia ha cambiado por alguna proporción en el rango de potencia medido (Bardwell, 2002).

Pese a que la medida de RSSI fue creada para obtener información acerca del enlace de comunicación también se busca utilizar este parámetro para obtener la posición relativa del receptor respecto a un número determinado de transmisores de los cuales se conoce su posición (Potorti, y otros).

### **2.6 Métodos de Posicionamiento**

Desde tiempo antiguos se han venido desarrollando distintos métodos para conocer la ubicación dentro de la superficie terrestre y en la actualidad existen muchas formas de encontrar la posición de un usuario, siendo el método principal el posicionamiento satelital.

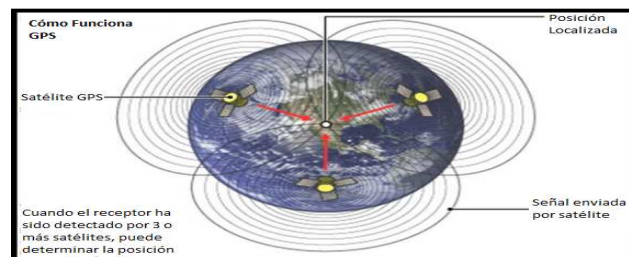
Existen varios sistemas de posicionamiento por satélites como el sistema denominado GPS (Global Positioning System) desarrollado por los Estados Unidos,

Glonass de la Federación Rusa, Galileo de la Unión Europea, Beidou perteneciente a la República Popular China, Qzss de Japón, Irnss de la India; sin embargo, el más extendido es el GPS que también es utilizado por los Smartphones y por el Sistema Operativo Android.

### 2.6.1 GPS (Global Positioning System)

El GPS o Sistema de Posicionamiento Global, es un sistema que permite ubicar la posición de un objeto o persona en el planeta, inicialmente el sistema fue desarrollado con propósitos militares por el Departamento de Defensa de los Estados Unidos para implementar un sistema de navegación en tiempo real, preciso y continuo en tierra, aire o mar independientemente de las condiciones meteorológicas (Chuquitarco, Naranjo, & Ronny, 2012).

El Sistema de Posicionamiento Global funciona mediante una red de 24 satélites en órbita sobre el planeta Tierra, y tiene por objetivo calcular la posición de un punto en un espacio de coordenadas  $(x,y,z)$  partiendo del cálculo de las distancias del punto a un mínimo de tres satélites cuya localización es conocida, una vez identificado cada satélite, se calcula el tiempo que tardan en llegar las señales al equipo permitiendo medir la distancia al satélite y, conociendo esta distancia, se puede determinar la posición relativa con respecto a los satélites para obtener la posición absoluta (coordenadas) del punto de medición como se ve en la Figura 15.



**Figura 15: Triangulación de los satélites**  
Fuente: (Herring, 1996)

Utilizando la información recibida de los tres satélites se puede determinar con precisión la latitud, longitud y altitud de cualquier punto en el planeta Tierra.

La distribución espacial de los satélites permite al usuario disponer de 5 a 8 satélites visibles en cualquier momento. El sistema está diseñado para asegurar que al menos cuatro satélites estarán visibles con una recepción configurada de la señal de 15° sobre el horizonte en un momento dado, en cualquier parte del mundo.

La precisión del posicionamiento GPS puede variar pudiendo llegar a obtener una precisión inferior a 2.5 metros cuando se captan las señales provenientes de entre siete y nueve satélites. El sistema GPS fue diseñado para exteriores puesto que la señal que llega al receptor es extremadamente débil y cuando el receptor se encuentra en interiores, la señal disminuye de 10 a 100 veces en una casa y de 100 a 1000 veces en grandes edificios (Van Diggelen, 2009).

Sin embargo, para mejorar el posicionamiento se hace uso de las redes móviles e inalámbricas utilizando el llamado GPS Asistido.

### **2.6.2 Posicionamiento mediante GPS Asistido**

El GPS Asistido o A-GPS mejora el desempeño del GPS proveyendo información a través de un canal alternativo de comunicación que el receptor GPS lo recibiría desde los mismos satélites (Van Diggelen, 2009).

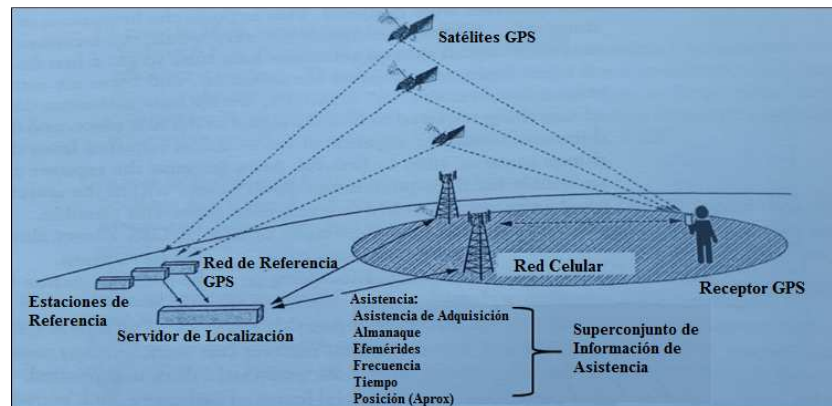
El GPS Asistido no excusa recibir y procesar la señal de los satélites, simplemente hace esta tarea más fácil y reduce la cantidad de tiempo e información requerida de los mismos. De esta manera al utilizar GPS Asistido, un receptor realizará mediciones más rápidas y podrá recibir señales más débiles que un receptor que no utilice GPS Asistido.

Con el desarrollo tecnológico actual, las aplicaciones que utilizan GPS se han incrementado de una manera importante, es por eso que ahora se requiere que GPS trabaje en casi cualquier parte incluso en interiores, que los datos se adquieran al instante y de una manera que añada poco o nada de costo, y que el consumo de la

batería del dispositivo sea bajo. Estos requerimientos son los que impulsaron el desarrollo del A-GPS.

El GPS Asistido trabaja proveyendo la información que permite al receptor GPS conocer previamente las frecuencias y la posición de los satélites. Una vez que se adquieren las señales, se realizan las mediciones (este proceso toma milisegundos) y finalmente el receptor A-GPS puede calcular la posición.

El tiempo que toma este cálculo se reduce del orden de un minuto al orden de un segundo. De la misma manera, al utilizar GPS - Asistido, se incrementa la sensibilidad del receptor que permite adquirir señales de menor intensidad ahorrando el consumo de la batería.



**Figura 16: Visión General de GPS - Asistido**

**Fuente:** (Van Diggelen, 2009)

Como se muestra en la Figura 16, el GPS Asistido brinda asistencia que permite conocer varios parámetros previamente como la frecuencia, tiempo, efemérides (tabla de valores correspondientes a los satélites GPS), almanaque (posición del satélite respecto al resto de satélites).

En resumen, el sistema de GPS Asistido está diseñado para acelerar el proceso del GPS permitiendo reducir notablemente el tiempo de adquisición de datos, para ello utiliza la información de las antenas de telefonía móvil y señales Wi-Fi de manera que se recibe información adicional para calcular la posición.

## CAPÍTULO 3

### DISEÑO Y DESCRIPCIÓN DEL SISTEMA

#### 3.1 Descripción de Hardware

Estos requerimientos, describen las características mínimas necesarias para la implementación del Sistema tanto a nivel de servidores como de dispositivos móviles.

##### 3.1.1 Definición de Requerimientos

El hardware que se requiere para que el sistema funcione consta de algunos elementos mencionados en la Tabla 10:

**Tabla 10**  
**Requerimientos de Hardware**

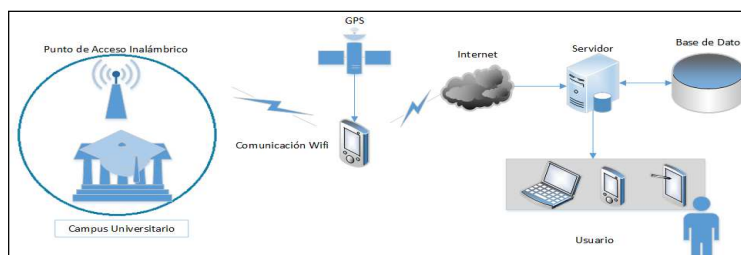
Requerimientos Hardware	Características
Dispositivos Access Point	Conexión a Internet. Compatibilidad con estándar 802.11 b/g/n. Identificación de parámetros requeridos (BSSID). Conocer ubicación dentro del campus de la ESPE.
Tarjetas Inalámbricas	Incluidas en teléfonos de gama alta o media. Compatibles con estándar 802.11 b/g/n. Lectura del parámetro RSSI.
Teléfonos Inteligentes	Gama alta y media. Memoria Ram de 512 MB o superior. Sistema Operativo Android API 15 o superior. Capacidad de obtener ubicación con GPS asistido.

**Continúa →**



Servidor	Memoria RAM de 8GB. Servidor de doble Procesador (Dual Xenon). Discos duros de 400GB. Conexiones de alta velocidad Gigabit Ethernet.
----------	---

Estos elementos se pueden observar en la Figura 17:



**Figura 17: Requerimientos de Hardware**

### Parámetro RSSI

El parámetro RSSI es un valor fundamental que, a más de ser un indicador de la fuerza de la señal recibida, también proporciona una idea de la distancia a la que se encuentra el dispositivo inalámbrico del punto de acceso.

Se puede decir que el valor de RSSI no es un valor 100% fiable puesto que el entorno ocasiona distintos efectos en su funcionamiento, por ejemplo, si el dispositivo móvil se encuentra estático en una posición, se observa que el valor de RSSI puede ir variando constantemente y no siempre resulta un valor fijo, esto se debe a que las ondas electromagnéticas rebotan sobre distintas superficies o son absorbidas por algunos materiales.

Teniendo todo esto en consideración, lo que se puede afirmar con certeza es que cuanto mayor es el valor del RSSI, el dispositivo se encuentra más cercano al punto de acceso (FQ, 2016).

## 3.2 Descripción y Diseño de Software

Los requerimientos de Software, describen los componentes lógicos necesarios para la implementación del Sistema. Existen requerimientos funcionales y no funcionales. Los requerimientos funcionales describen lo que el sistema debe hacer mientras que los no funcionales no son imprescindibles y se enfocan al diseño e implementación.

### 3.2.1 Requerimientos de Diseño

El software del sistema, consta de los siguientes elementos funcionales:

#### a) Aplicación Móvil

##### **Obtener Ubicación**

- Verificar que GPS esté activado.
- Adquirir latitud y longitud con 6 decimales.
- Adquirir datos cuando existan cambios de posición geográfica.

##### **Obtener Red**

- Identificar la red a la que se está conectado (BSSID).
- Adquirir parámetro RSSI de la red a la que está conectada.

##### **Enviar Datos**

- Verificar adquisición de datos.
- Iniciar tarea asíncrona para envío de datos.
- Preparar datos obtenidos (latitud, longitud, RSSI, identificador, fecha).
- Utilizar librerías y levantar puertos para el envío de datos.
- Indicar direcciones IP del destino (Servidor de Base de Datos).
- Enviar datos a la correspondiente Base de Datos.
- Verificación del envío mediante mensaje de confirmación.

##### **Visualizar Páginas Web**

- Poseer un navegador web integrado

#### b) Aplicación Web

##### **Visualizar Datos**

- Definir estructura de la página mediante HTML.
- Definir la presentación de la página a través CSS.
- Utilizar JavaScript para crear páginas web dinámicas.
- Solicitud de Información a la Base de Datos.

#### **Generar Mapas**

- Utilizar API de Google Maps para crear un nuevo mapa.
- Ubicar el mapa en la posición inicial.

#### **Graficar Marcadores**

- Comparar los valores adquiridos de RSSI para determinar el color que se grafica en el mapa.
- Graficar marcadores en el mapa.

#### **c) Alojamiento Web**

- Trabajar con Servidor Web.
- Trabajar con Servidor de Base de Datos.
- Trabajar con código PHP.
- Obtener datos estadísticos.

#### **d) Base de Datos**

- Adquirir datos de conexión para el envío y recepción de datos.
- Definir las tablas para almacenar los datos.

También se pueden mencionar algunos elementos **no funcionales** que pueden ser utilizados para una mejor experiencia del usuario como:

- Utilizar elementos gráficos (botones, selectores) con el fin crear una interfaz intuitiva para el usuario.
- Mantener la seguridad en las claves de acceso.
- Realizar el envío de datos en tiempo real.
- Utilizar software libre para el desarrollo.

### **3.2.2 Aplicación Móvil**

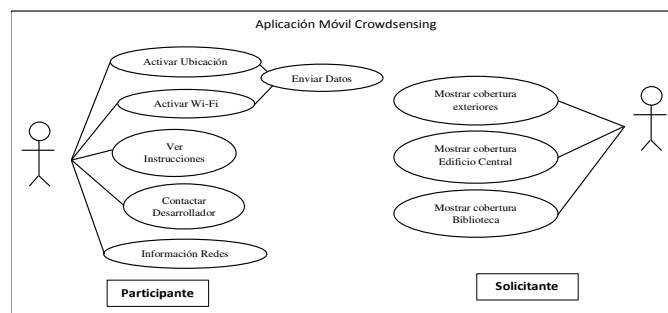
#### **Entorno de Desarrollo IDE**

Para implementar el proyecto, es importante tener en cuenta el Entorno de Desarrollo, pues este proporciona las herramientas necesarias para la creación del

software. Se ha elegido Android Studio debido a su facilidad de uso, por el soporte masivo de comunidades y porque está avalado por compañías como Google.

### Diagrama de Casos de Uso de la Aplicación

Como se observa en la Figura 18, para conocer la interacción entre el usuario y la aplicación a desarrollar, se utiliza el Diagrama de Casos de Uso que representa la forma en como un cliente opera con el sistema en desarrollo.



**Figura 18: Diagrama de modelo de uso de la aplicación móvil**

Un factor importante al momento de mostrar la cobertura en interiores es determinar el piso o área en donde se encuentra el participante que envía el dato.

### Determinación del lugar geográfico (interiores)

Para encontrar la solución a este problema se plantearon algunas soluciones:

- Obtener la altura mediante GPS.
- Pedir al participante que manualmente elija el piso en el que se encuentra
- Conocer la ubicación mediante la información que proporcionan los distintos puntos de acceso ubicados en cada piso.

El primer método resulta inviable puesto que los valores de la altura obtenidos a través de GPS tienen un gran margen de error y no permiten conocer con certeza el piso donde se encuentra el participante, además el GPS presenta muchos problemas al trabajar en interiores.

El segundo método no resulta factible puesto que una condición de diseño solicita que el participante únicamente se mueva por el campus sin realizar ninguna acción específica.

El tercer método es viable puesto que en los edificios de la Universidad de las Fuerzas Armadas – ESPE existen varios puntos de acceso ubicados en cada piso, si se identifica a qué punto de acceso está conectado el participante, se puede determinar el piso en el que se encuentra.

Para este fin se utiliza el parámetro BSSID que es el identificador único publicado por cada punto de acceso. Una primera etapa consiste en ir anotando el valor de BSSID de los distintos puntos de acceso ubicados en cada piso para luego comparar este valor con el proporcionado por el participante y de esta manera, determinar el piso en el que se encuentra.

Por ejemplo, en la tabla 11 se observan los identificadores únicos BSSID en cada lugar, con lo cual se puede determinar el piso exacto en donde se encuentra el participante y de esta manera enviar la información a la tabla correspondiente de la base de datos en el servidor.

**Tabla 11**  
**Variaciones de BSSID en cada piso (Bloque A)**

<b>Lugar</b>	<b>BSSID</b>
Bloque A Planta Baja y Subsuelo	4e:d9:e7:09:4a: <b>75</b>
Bloque A Piso 1	4e:d9:e7:09:4a: <b>92</b>
Bloque A Piso 2	4e:d9:e7:09:4a: <b>9a</b>
Bloque A Piso 3	4e:d9:e7:09:4b: <b>e1</b>

### 3.2.2.1 Interfaz de Visualización

Para crear la interfaz de visualización en Android Studio, es necesario conocer los componentes fundamentales que van a interactuar en la aplicación Android:

#### Activity

Una *Activity* se puede definir como cada una de las pantallas que se crean en una aplicación Android. Existen dos partes bien diferenciadas de una *Activity*, la parte lógica y la parte gráfica (Rodríguez J. , 2014).

La parte lógica corresponde a la parte de la programación que es una clase Java encargada de implementar las funcionalidades de la aplicación mientras que la parte gráfica corresponde a un archivo XML, en el cual se definen los componentes gráficos que formarán la interfaz del usuario.

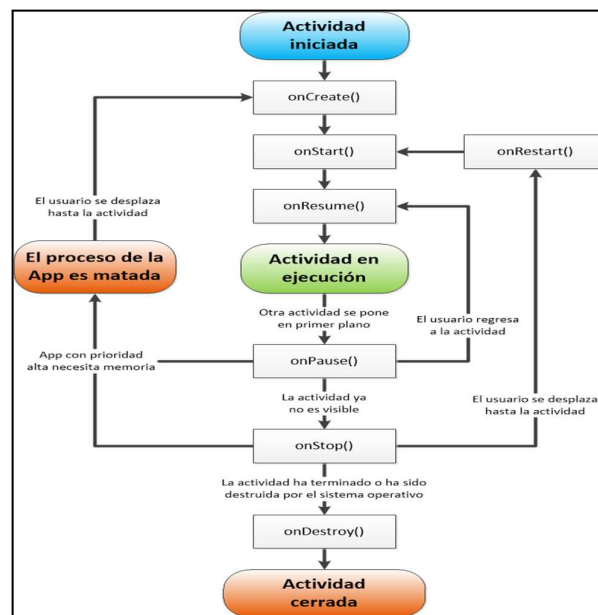
Una aplicación por lo general se compone de varias actividades que están ligadas entre ellas, existe una actividad que se especifica como la actividad "principal", la cual se presenta al usuario donde inicia la aplicación por primera vez. Cada actividad puede iniciar otra actividad con el fin de realizar diferentes acciones.

#### Ciclo de Vida de una Activity

Una actividad se caracteriza por tener un ciclo de vida que se puede ver en la Figura 19, el cual se ve afectada directamente por su asociación con otras actividades, su cometido, etc, algunos métodos importantes son:

- **onCreate(), onDestroy():** Corresponden al inicio y fin de la actividad y contienen todo el ciclo de vida útil de una actividad.
- **onStart(), onStop():** Estos métodos conforman la parte visible para el usuario, el mismo que puede interactuar con la aplicación. Desde el método *onStart()* hasta *onStop()*, la actividad será visible.

- **onResume(), onPause():** Estos métodos delimitan la parte útil del ciclo de vida. Desde *onResume()* hasta *onPause()*, la actividad está delante de todas las demás actividades y tiene el foco de la acción, el usuario puede interactuar con dicha actividad (Developers A. , 2016).



**Figura 19: Ciclo de vida de una activity en Android**

Fuente: (Invarato, 2012)

## Broadcast Receiver

Permite responder a los anuncios de mensajes en todo el sistema. También se utiliza para escuchar *intents*. Los *intents* son mensajes que pasan entre los componentes del sistema, su principal uso es el lanzamiento de actividades.

## Service

Un servicio es un componente de aplicación que puede realizar operaciones de larga ejecución en segundo plano y no proporciona una interfaz de usuario.

## Content Provider

Los proveedores de contenidos gestionan el acceso a un conjunto estructurado de datos. Ellos encapsulan los datos, y proporcionan mecanismos para la definición de seguridad de los mismos. A través de éstos elementos, una aplicación en Android puede almacenar datos en un fichero, en una base de datos, etc (Chhabra, 2013).

En el proyecto, dentro de la actividad principal, se pretende obtener algunos parámetros (latitud, longitud, identificador del dispositivo, BSSID, fecha) e iniciar una tarea asíncrona para enviar los mismos a la base de datos del servidor.

Con el objetivo de crear una interfaz amigable para el usuario, también se decide utilizar *fragments* dentro de la *activity*.

## Fragments en Android

Los *fragments* fueron introducidos a partir de la versión de Android 3.0 (Api level 11) y son componentes que funcionan dentro de una actividad, su finalidad es facilitar la navegación entre pantallas o actividades, al estar embebido dentro de una *activity*, el ciclo de vida de un *fragment* está ligado al de la actividad en donde se ha definido. Existen varias ventajas al utilizar *fragments* ya que cada uno es independiente del resto y por lo tanto reutilizable, además permiten crear aplicaciones que funcionen en dispositivos móviles o en tablets proporcionando diseños dinámicos y flexibles para pantallas de distintos tamaños (Rodríguez J. , 2014).

Se propone desarrollar varios *fragments* para el proyecto:

- **Fragment Exteriores:** Permitirá visualizar la cobertura en los exteriores.
- **Fragment Central:** Permitirá visualizar la cobertura en el Edificio Central.
- **Fragment Biblioteca:** Permitirá visualizar la cobertura en la Biblioteca.
- **Fragment Redes:** Permitirá conocer información adicional de las redes.
- **Fragment Contacto:** Permitirá contactarse con el desarrollador.



- **Fragment Instrucciones:** Permitirá conocer las instrucciones de funcionamiento

Para la comunicación con el servidor web, es necesario que la aplicación trabaje realizando el envío de datos en segundo plano con la finalidad de funcionar de manera eficiente sin causar lentitud en el dispositivo, para este fin es posible utilizar hilos o tareas asíncronas.

### Hilos y tareas asíncronas

Las dos opciones que ofrece Android para realizar trabajos en segundo plano es utilizar hilos o tareas asíncronas. Para la aplicación a desarrollar, se elige trabajar con **tareas asíncronas** puesto que el código a utilizar es mucho menor, mejor entendible y menos propenso a errores, entonces se utiliza la clase *AsyncTask* y se crea una nueva tarea que extienda de ella para proceder a sobrescribir algunos métodos que se describen en la Tabla 12:

**Tabla 12**  
**Métodos de AsyncTask**

Métodos	Descripción
<i>onPreExecute()</i>	Este método ejecuta en el hilo principal de la aplicación, es decir antes de empezar la tarea en segundo plano; generalmente se suele utilizar para preparar la ejecución de la tarea, inicializar variables, preparar objetos, etc.
<i>doInBackground()</i>	Corresponde a la tarea que se ejecuta en segundo plano, permite llamar al método <i>publishProgress()</i> el cual se comunica con el hilo principal.
<i>onProgressUpdate()</i>	Se ejecutará cada vez que llamemos al método <i>publishProgress()</i> desde el método <i>doInBackground()</i> , este método se ejecuta hasta que la tarea en segundo plano finalice.
<i>onPostExecute()</i>	Se ejecutará cuando finalice nuestra tarea en segundo plano, es decir cuando se termina de ejecutar el método <i>doInBackground()</i> .
<i>onCancelled()</i>	Se ejecutará cuando se cancele la ejecución de la tarea antes de su finalización normal.

**Fuente:** (Sgoliver.net, 2012)

Se debe tener en cuenta que solo el método `doInBackground()` se ejecuta en segundo plano (hilo secundario), todos los demás métodos se ejecutan en el hilo principal.

Para la aplicación se pretende utilizar una tarea asíncrona para convertir todos los parámetros (latitud, longitud, identificador del dispositivo, BSSID, fecha) en cadenas, iniciar la comunicación HTTP y enviarlos a la correspondiente base de datos en el servidor.

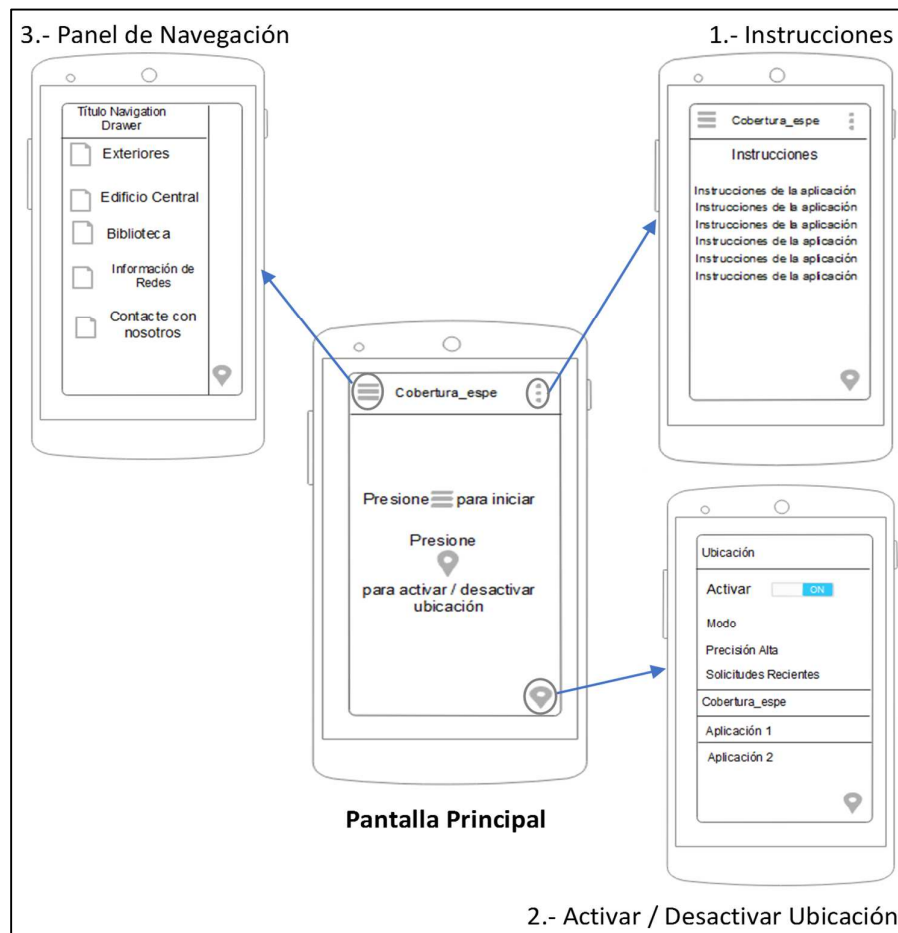
### **Visualización de la Aplicación**

En la aplicación a desarrollar, la actividad principal se va a encargar del envío de datos al servidor ejecutando una tarea en segundo plano para que no afecte la visualización de los elementos gráficos. En la actividad principal también se pretende mostrar un panel de navegación en donde el usuario puede seleccionar entre las distintas opciones y cada una de éstas estará asociada a un fragmento el cual permitirá realizar una actividad específica.

Para poder visualizar la interfaz de la aplicación, es necesario crear un *wireframe* o boceto de las funcionalidades básicas.

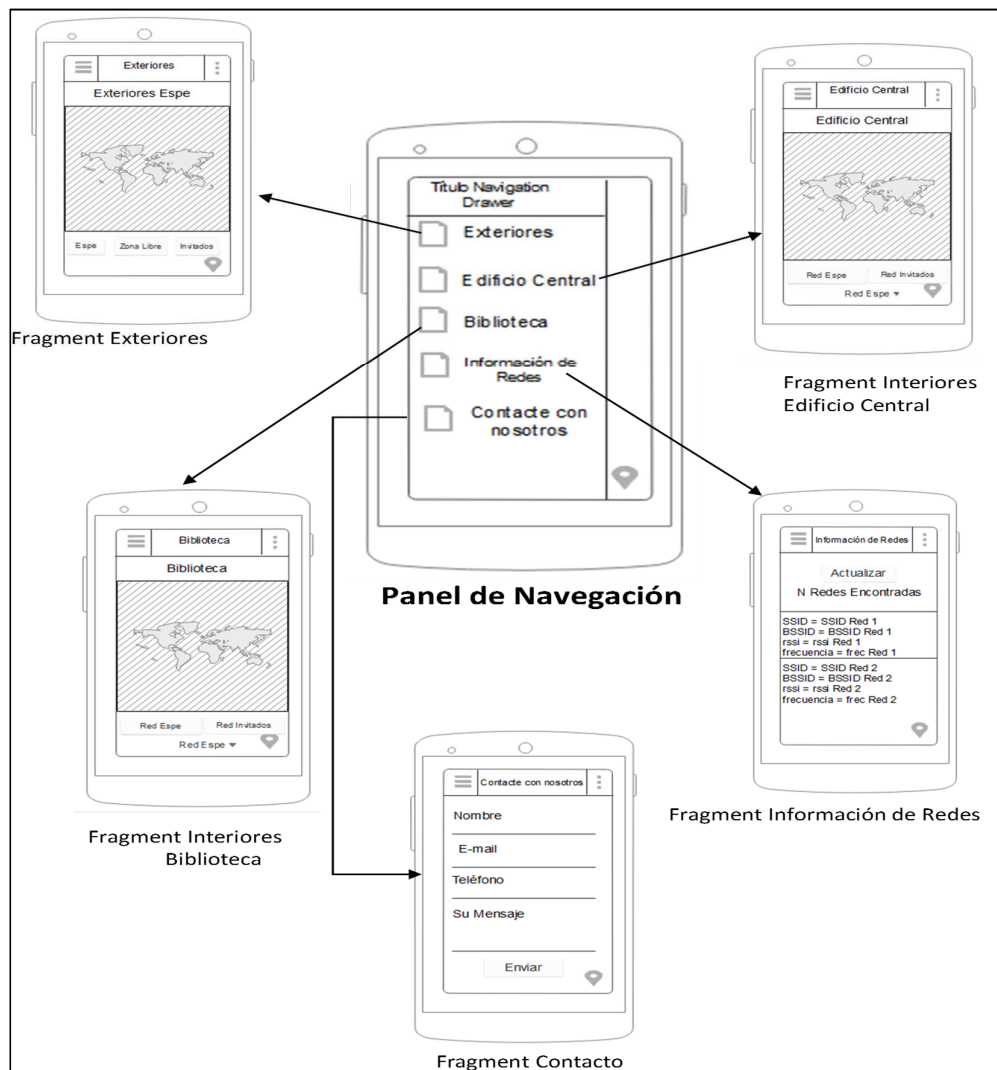
El boceto representa el esqueleto de la aplicación, es decir solamente se representan líneas y vértices sin incluir gráficos, estilos o colores. El boceto permite visualizar la distribución del espacio y conocer las funcionalidades disponibles, además una correcta planificación del boceto permite ahorrar tiempo y dinero en costos de diseño y desarrollo (Lancetalent, 2014).

Para la aplicación se propone la siguiente interfaz de usuario:



**Figura 20: Boceto de la aplicación**

En la Figura 20 se observa en el medio la pantalla principal que se obtendrá al abrir la aplicación instalada en el dispositivo, se indican los elementos para interactuar con el usuario como instrucciones, activar / desactivar ubicación y abrir el panel de navegación.



**Figura 21: Panel de Navegación de la aplicación (fragments asociados)**

A través del panel de navegación mostrado en la Figura 21, el usuario podrá escoger entre las distintas opciones que ofrecen los *fragments* como mostrar el mapa de cobertura en exteriores e interiores (Edificio Central y Biblioteca), conocer la información de todas las redes disponibles y ponerse en contacto con el desarrollador.

### 3.2.3 Mapas

Puesto que un requerimiento es observar gráficamente el área de cobertura en la Universidad de las Fuerzas Armadas - ESPE, es necesario disponer de un servicio donde se muestren gráficamente los resultados obtenidos; es por este motivo que se utiliza Google Maps pues ofrece una tecnología cartográfica potente, fácil de usar y figura como el servicio más utilizado para buscar localizaciones (Medialex, 2016).

#### API Google Maps

Google Maps permite visualizar mapas de todo el mundo haciendo uso de tecnología HTML, CSS y JavaScript trabajando conjuntamente. Los mapas corresponden a imágenes preestablecidas que se cargan a través de peticiones ejecutadas por la tecnología de AJAX y que son insertados en la página HTML (Rodríguez A. , 2011).

El api de Google Maps permite a los desarrolladores visualizar los mapas dentro de sus propias páginas web con JavaScript. Para acceder a esta tecnología, es necesario poseer una clave (Maps API Key) que será proporcionada por Google.

#### Tipos de Mapas

Existen cuatro tipos de mapas disponibles como se observa a continuación:

**Tabla 13**

#### Tipos de mapas

Tipos de Mapas	Características
ROADMAP	Muestra una vista de carreteras.
SATELLITE	Se muestran imágenes satelitales de Google Earth.
HYBRID	Es una combinación del mapa de ruta (Roadmap) con imágenes satelitales.
TERRAIN	Muestra un mapa físico.

**Fuente:** (Developers G. , 2016)

En función de los tipos de mapas descritos en la Tabla 13, para el desarrollo de la aplicación se plantea utilizar el mapa tipo *Satellite* para poder obtener los mapas de cobertura sobre imágenes satelitales del área de la Universidad de las Fuerzas Armadas - ESPE.

### **Ubicación Geográfica**

Para el desarrollo del proyecto, se requiere obtener la ubicación geográfica. Cualquier coordenada del planeta se puede expresar utilizando números decimales separados por coma: (latitud, longitud). La latitud es la distancia que existe entre la línea ecuatorial y un punto de la tierra mientras que la longitud es la distancia al punto tomado como referencia (meridiano de Greenwich).

En los mapas, las coordenadas están expresadas en grados, minutos y segundos, en el caso de la Universidad de las Fuerzas Armadas – ESPE, está dada por:

- Latitud: S 0° 18' 52.329''
- Longitud: W 78° 26' 41.621''

Para convertir éstos datos a decimales, la operación a realizar es:

$$(S\ 0^\circ\ 18'\ 52.329'') = -(0 + (18/60) + (52.329/3600)) = -0.314535$$

$$(W\ 78^\circ\ 26'\ 37.29'') = -(78 + (26/60) + (41.621/3600)) = -78.44489$$

Una vez que se ha obtenido la latitud y longitud, es necesario tener en cuenta el número de decimales a utilizar en el proyecto.

### **Número de Decimales**

Las latitudes y longitudes con más de seis decimales se ignoran por Google Maps, así que el número de decimales establecido viene como se muestra en la Tabla 14:

**Tabla 14****Determinación de número de decimales a utilizar**

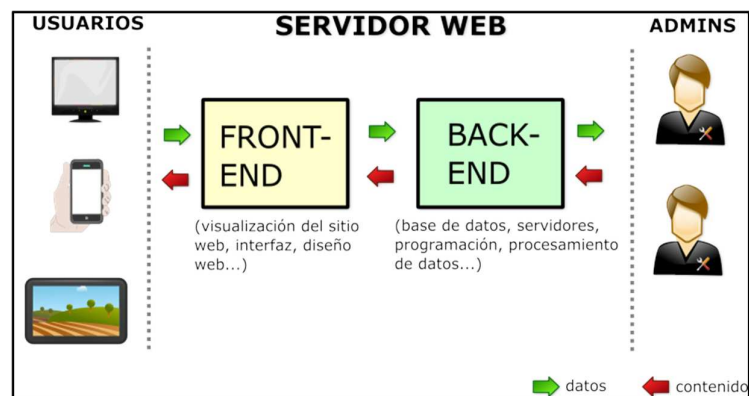
Número de Decimales	Determinación de Lugares
2	Para ubicar países o estados
3	Para centrar ciudades
4	Para algún detalle en el mapa
5 a 6	Máximo número de decimales para ser específicos

**Fuente:** (Rodríguez A. , 2011)

Por lo tanto, para obtener datos lo más cercanos a la realidad posible, se plantea utilizar 6 decimales para determinar la ubicación de un usuario dentro de la ESPE.

### 3.2.4 Plataforma Web

Para el desarrollo web se pretende utilizar la Arquitectura Cliente – Servidor que consta generalmente de dos componentes: *front – end* (cliente) y *back – end* (servidor) e interactúan entre sí para satisfacer todas las necesidades de los usuarios. Además, para que exista la comunicación se requiere de estándares establecidos siendo el principal el protocolo HTTP (Aguilar & Dávila, 2013). La Arquitectura Cliente – Servidor se puede ver en la Figura 22.



**Figura 22: Front - End y Back - End**

**Fuente:** (Alegsa, 2016)

### 3.2.4.1 Front End

Cuando se refiere a *front-end*, se habla de todas aquellas herramientas tecnológicas que corren del lado del cliente (navegador web) y que son visibles para el usuario (CampusMVP, 2015).

Para cumplir todos los requerimientos de diseño, en el *front-end* se definen tres lenguajes principales a utilizar: HTML, CSS y JavaScript.

#### 3.2.4.1.1 HTML

HTML es un lenguaje de descripción de hipertexto compuesto por una serie de comandos, marcas, o etiquetas denominadas *Tags* que permiten definir la estructura lógica de un documento web y establecer los atributos del mismo (color del texto, contenidos multimedia, hipervínculos, etc.) (Cobo, Gómez, Pérez, & Rocha, 2005).

Para crear un documento HTML se puede utilizar cualquier editor de texto o programas específicos creados para editar HTML como Notepad++, TextMate, etc.

Se necesita un navegador web para leer los archivos HTML tal como Google Chrome, Mozilla Firefox, Opera, Microsoft Edge, etc., por lo que la presentación de la página es dependiente del navegador web, es decir que visualizar la misma presentación en navegadores distintos, no produce el mismo resultado.

Las herramientas esenciales asociadas al lenguaje HTML son los editores y los navegadores, aunque también existen otras útiles como los conversores (permiten pasar textos de otros formatos a HTML como latex2html, Total HTML converter, etc.) y validadores (servicios que se encargan de revisar la página web indicando errores en el código HTML como el caso del consorcio de validación W3C, validator.nu, etc.) (Lamarca, 2013).



### 3.2.4.1.2 CSS

CSS corresponde a las iniciales de *Cascade Style Sheets* (hojas de estilo en cascada) y es un lenguaje que indica la manera de cómo se van a mostrar cada uno de los elementos. Este lenguaje ha sido desarrollado por el consorcio *World Wide Web* (W3C). Al referirse al término en cascada, significa que diferentes hojas de estilo se pueden aplicar sobre una misma página (Luján-Mora, 2002).

CSS es utilizado para indicar la presentación de documentos definidos con HTML; haciendo una analogía, HTML tiene que ver con el contenido mientras que CSS se encarga de la presentación o el aspecto.

El lenguaje CSS utiliza reglas para su sintaxis. Las reglas permiten definir el aspecto de los elementos.

Una hoja de estilo está compuesta por una o varias reglas; cada regla consiste en un selector y en una declaración y a su vez la declaración está formada por una propiedad y el valor que se le asigne (W3C, Servicios Web, s.f.).

Gracias a la flexibilidad que ofrece el lenguaje CSS, existen diversas formas de dar estilo a un documento HTML:

- Incluir el lenguaje CSS en la cabecera del mismo documento.
- Definir CSS directamente en los elementos HTML
- Crear una hoja de estilo externa que se enlaza con la página HTML.

### 3.2.4.1.3 JavaScript

Este lenguaje de programación tiene como principal objetivo la creación de páginas web dinámicas (incorporar efectos, animaciones, acciones, abrir nuevas ventanas, etc.) y permite crear *scripts* que se insertan en las páginas HTML para ser ejecutado en un navegador (Google Chrome, Mozilla Firefox, etc.). Los *scripts* por lo general son

funciones que son llamadas desde HTML cuando ocurre algún evento (Navarrete, 2006).

Existen por lo menos tres formas para integrar código JavaScript en una página web:

- Insertar directamente el código JavaScript junto con los elementos HTML.
- Incluir el código en cualquier parte del archivo dentro de la etiqueta `<script>`.
- Crear un archivo externo con el código que será enlazado indicando su ubicación.

#### **3.2.4.1.4 Diseño Front End y Visualización**

A más de poder visualizar la cobertura en el dispositivo móvil, se requiere una aplicación web con el mismo propósito, por este motivo se desarrolló una página web donde es posible ver la cobertura inalámbrica de la Universidad de las Fuerzas Armadas – ESPE tanto en interiores como en exteriores.

Antes de crear una página web, es necesario hacer una planificación para conocer todas las características de la misma. Es por esto que previamente se debe desarrollar un *wireframe* o boceto que muestra de una forma sencilla y esquemática la estructura de la página web a diseñar.

El objetivo de diseñar previamente un boceto es precisar el contenido y la posición de los diversos elementos que serán incluidos en la página web (títulos, párrafos, enlaces, botones, listas, gráficos, etc.) y conocer cómo interactúan estos elementos entre sí.

Un boceto adecuado debe centrarse en la funcionalidad del sitio y la experiencia del usuario. En definitiva, lo importante es tener claro la estructura de la página web antes de comenzar a desarrollar con los lenguajes HTML y CSS (Centelles, 2015).

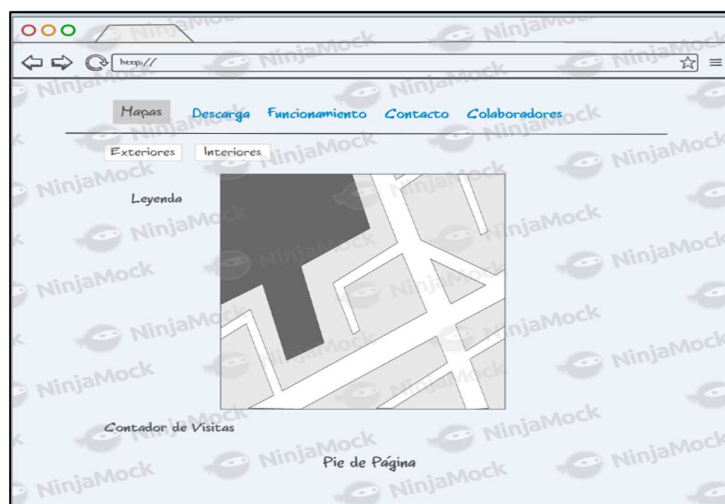
Para diseñar la página web de la aplicación, se pretende hacer uso de los lenguajes del lado del cliente (HTML, CSS, JavaScript) para que ésta cumpla con todos los objetivos del proyecto.

La página web para la aplicación cuenta con varias pestañas disponibles para el usuario y cada una de ellas ha sido pensada para cumplir con una función específica como se muestra en la Tabla 15:

**Tabla 15**  
**Opciones incluidas en la página web de la aplicación**

Opciones	Características
Mapas	Corresponde a la página principal de la aplicación y permite conocer la cobertura inalámbrica de las redes de la ESPE tanto en interiores como en exteriores.
Descarga	Permite descargar la aplicación Android para el dispositivo móvil.
Funcionamiento	Es un pequeño manual de usuario donde se puede encontrar una explicación de cada pestaña disponible en el menú.
Contacto	Permite llenar un formulario a través del cual, el usuario puede ponerse en contacto con el desarrollador.
Colaboradores	En esta opción se puede conocer los colaboradores que hicieron posible la creación de este proyecto.

Teniendo en cuenta los requerimientos funcionales como no funcionales, el diseño general de la página web se muestra a continuación:

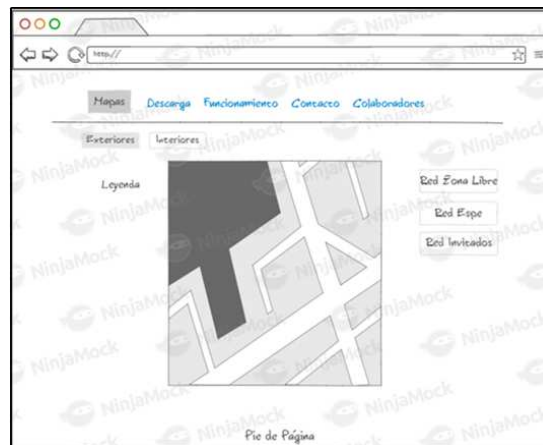


**Figura 23: Portada de la página web desarrollada**

Como se observa en la Figura 23, la página de inicio en su parte superior contará con un menú donde están todas las opciones disponibles para el usuario y en la parte inferior se mostrará el mapa en donde se procederá a visualizar la cobertura inalámbrica de las redes.

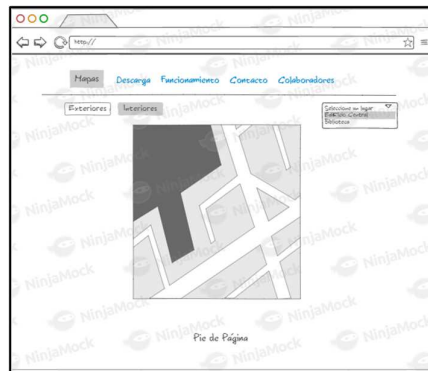
### Opción Mapas

Al seleccionar la opción mapas que se muestra en la Figura 24, habrá dos botones para elegir (Exteriores o Interiores), si el usuario elige exteriores, puede seleccionar en la parte derecha la red (Espe, Zona Libre o Invitados) con lo cual se mostrará en el mapa la cobertura de la red seleccionada



**Figura 24: Selección de opción exteriores**

Si el usuario elige la opción interiores como se ve en la Figura 25, primero tendrá que seleccionar un lugar (Edificio Central o Biblioteca) para después elegir el piso correspondiente a la red deseada como se ve en la Figura 26.



**Figura 25: Selección de opción interiores**



**Figura 26: Selección del piso en la red correspondiente**

También se pretende que en la parte izquierda se muestre un gráfico informativo donde se puedan ver los colores utilizados para los marcadores que dependen de las potencias de RSSI obtenidas, estos rangos de valores se seleccionarán luego de realizar varias mediciones analizando los valores obtenidos y se pueden ver en la Tabla 16.

**Tabla 16**  
**Rango de RSSI para graficar en el mapa**

Color	Intensidad
Verde	Excelente
Amarillo	Muy Buena
Tomate	Buena
Rojo	Regular

Si se selecciona la opción en interiores, en la parte izquierda también aparecerá una imagen referente al piso del cual se está observando la cobertura.

### **3.2.4.2 Back End**

Corresponde a todos los componentes que procesan la entrada de datos efectuada desde el *front end*, es decir, el *back end* hace referencia a todos los procesos utilizados por el administrador del sitio con sus respectivos sistemas con la finalidad de satisfacer las peticiones de los usuarios. Los procesos *back end* se realizan en el lado del servidor. Para la aplicación a desarrollar, en el *back end* se utiliza el servidor HTTP Apache, el lenguaje de programación PHP y MySQL para gestionar la base de datos.

#### **3.2.4.2.1 Servidor Apache**

Es un servidor web HTTP (*Hyper Text Transfer Protocol*) de código abierto, Apache está presente en varios Sistemas Operativos (Microsoft, Unix, Windows, Macintosh) que implementan el protocolo HTTP.

#### **3.2.4.2.2 Programación PHP**

PHP corresponde a las siglas de *pre Hypertext -processor*, el cual es un lenguaje de uso libre y gratuito que se puede combinar con código HTML.

El lenguaje PHP se ejecuta en servidores Web, el cual recibe la petición, procesa la información y devuelve una respuesta, por lo tanto, es necesario que PHP esté instalado en el servidor.

PHP permite realizar distintas operaciones en el servidor y luego muestra la salida por lo general en lenguaje HTML, es decir el usuario no verá el código PHP

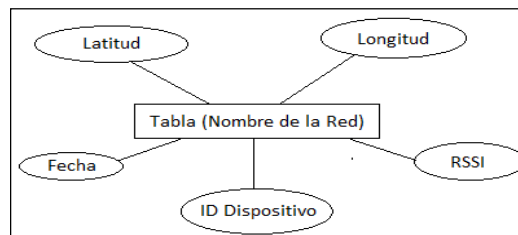
implementado en el servidor, sólo podrá ver páginas HTML con los resultados generados (Cabrera, 2005).

### 3.2.4.2.3 MySQL

Es un software de código abierto que permite gestionar bases de datos relacionales y fue desarrollado por Oracle Corporation. Este software está basado en el lenguaje SQL (Lenguaje de Consulta Estructurado) y permite trabajar en distintos Sistemas Operativos.

Entre las características principales de MySQL se pueden mencionar las siguientes: permite trabajar con volúmenes grandes como pequeños de información, realizar múltiples y rápidas consultas, manipular, crear y seleccionar datos, así mismo facilita la administración remota al disponer de un monitor y un sistema de ayuda en línea que posibilita realizar cualquier operación sin la necesidad de alguna interfaz gráfica (Cobo, Gómez, Pérez, & Rocha, 2005).

El modelo Entidad – Relación es un aspecto importante de diseño, y para la aplicación se propone uno como se muestra en la Figura 27:



**Figura 27: Modelo Entidad – Relación para la Base de Datos**

Se pretende que el nombre de la tabla contenga información útil como la red, el edificio y el piso donde se adquiere el dato. El tipo de datos para los atributos, se muestra en la Tabla 17:

**Tabla 17**  
**Tipos de datos de los atributos**

Nombre de la Columna	Tipo de Datos
Latitud	double NOT NULL
Longitud	double NOT NULL
Rssi	int NOT NULL
Id Dispositivo	varchar(64) NOT NULL
Fecha	varchar(64) NOT NULL

### 3.2.4.3 Protocolo HTTP

El protocolo de transferencia de hipertexto (*Hypertext Transfer Protocol*) HTTP es un estándar para transferencia en la web que se encarga de la comunicación de hipertextos en lenguaje HTML, además utiliza operaciones de solicitud/respuesta, es decir un cliente envía una solicitud (*request*) a un servidor y este responde con los posibles resultados (*response*).

Las operaciones adjuntan un objeto o recurso conocido como URL (Localizador Uniforme de Recursos) que es una secuencia de caracteres que permite localizar un recurso disponible en la web (Pérez & Merino, 2012). Para realizar operaciones de solicitud/respuesta, hay algunos métodos utilizados que se detallan en la Tabla 18:

**Tabla 18**  
**Diferencias entre métodos GET y POST**

Método	Concepto
Get	Solicita datos de un recurso especificado. Los datos son llevados de forma visible al cliente, el medio de envío es la url.
Post	Envía los datos a ser procesados. El envío se realiza de manera oculta (no son visibles para el cliente).

Fuente: (W3C, Servicios Web, s.f.)

### 3.3 Selección del Alojamiento Web

Un paso fundamental para la realización del proyecto consiste en seleccionar un **alojamiento web** con su respectiva base de datos en la cual se irán almacenando los



valores de latitud, longitud, RSSI, ID del dispositivo y fecha que son enviados por los participantes desde un dispositivo móvil para su almacenamiento y posterior visualización en el mapa. Se decidió utilizar un alojamiento web para que la información utilizada esté disponible para todos los usuarios de la red en cualquier parte del mundo.

Para seleccionar el alojamiento web, se tomaron en consideración a una gran variedad de opciones: 000WebHost, HostGator, miarrobaES, zPlanet, 2FreeHosting, Hostinger, etc. Teniendo en cuenta las características y funcionalidades de cada uno, se decidió utilizar el alojamiento **miarrobaES** ya que es completo y gratuito, sus principales características se observan en la Tabla 19.

**Tabla 19**  
**Características del hosting miarroba**

<b>Características</b>	<b>Observaciones</b>
Capacidad de Almacenamiento	100 MB por cada espacio web
Número máximo de espacios	5 espacios web
Límite de transferencia	Ilimitada
Versión de PHP	5.3.16
Versión de MySQL	MySQL 5.6.34
Tamaño máximo permitido para cada archivo	5 MB
Métodos de acceso a ficheros	Cliente FTP o directamente utilizando el administrador WebFTP.
Capacidad de Ampliación del almacenamiento	Hasta 500 MB según las visitas y contenido de la página web.
Página de informe de errores	Se muestra el error 404.
Herramienta gestionar las bases de datos	phpMyAdmin en servidor seguro SSL.
Panel de control disponible	Sí.
Estadísticas por días de tráfico, transferencia de datos	Disponible.
Características administrador web	Permite subir, actualizar, renombrar y borrar archivos y directorios desde cualquier ordenador que tenga acceso a internet.
Características de la url	Url corta y atractiva para acceder a tu WEB, del tipo: http://nombre.webcindario.com

**Fuente:** (miarroba, 2016)

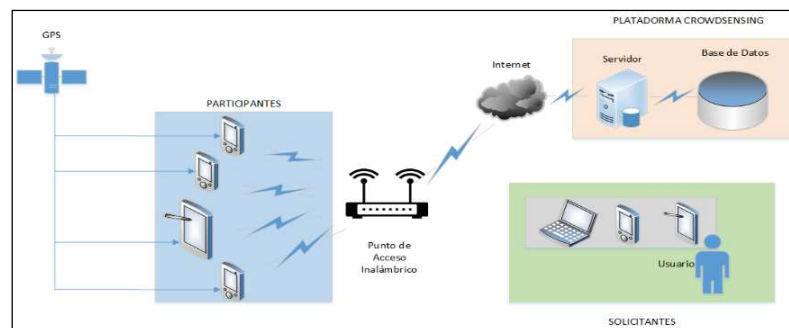
## CAPÍTULO 4

### IMPLEMENTACIÓN DEL SISTEMA

#### 4.1 Arquitectura del Sistema

##### 4.1.1 Escenario para el Sistema Crowdsensing

Para la implementación del sistema *Crowdsensing* es necesario contemplar el escenario de red donde se llevarán a cabo las comunicaciones entre los múltiples participantes, la plataforma *Crowdsensing* y los solicitantes.

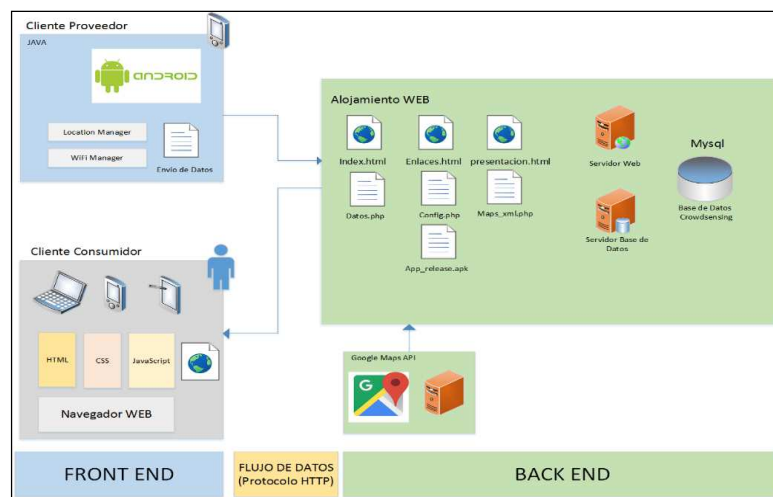


**Figura 28: Escenario para el Sistema Crowdsensing**

En la Figura 28 se observan todos los elementos que componen el Sistema *Crowdsensing* (Dispositivos Móviles, Puntos de Acceso, Servidor, etc.) que interactúan entre sí y con otros elementos externos (GPS). Todos y cada uno de los elementos del escenario planteado son importantes ya que, si alguno de ellos llega a fallar, todo el Sistema de *Crowdsensing* también lo hará.

## 4.2 Desarrollo de la Plataforma Crowdsensing

La arquitectura general de la Plataforma *Crowdsensing* se muestra en la Figura 29 donde se pueden observar tres bloques bien definidos que corresponden a la Arquitectura *Crowdsensing* (Participantes, Solicitantes y Plataforma).



**Figura 29: Esquema General de Tecnología**

Los participantes (cliente proveedor) son las personas que tienen un dispositivo móvil con el Sistema Operativo Android el cual se encarga de la recolección de los datos (latitud, longitud, RSSI, identificador del dispositivo, fecha) para enviarlos a la plataforma y almacenarlos en la respectiva base de datos del servidor.

Los solicitantes (cliente consumidor) utilizan los datos guardados en la correspondiente base de datos de la plataforma y a través de las tecnologías del lado del cliente (HTML, CSS y JavaScript) muestran los mapas de cobertura de las redes inalámbricas de la ESPE tanto en interiores como en exteriores.

La plataforma se encarga de guardar y procesar todos los datos obtenidos de los participantes a través de las tecnologías del lado del servidor (PHP, MySQL) para enviarlos a los solicitantes según sus requerimientos.

Al utilizar este esquema para el desarrollo de la aplicación, se hace uso de dos conceptos fundamentales: por un lado, se utiliza la Arquitectura *Crowdsensing* para envío y recolección de datos y también se utiliza la Arquitectura Cliente – Servidor para un servicio Web.

### Programa Realizado en Android Studio

La aplicación realizada en Android Studio consiste en una única actividad que tiene asociada varios *fragments* con el objetivo de poder visualizar todas las opciones y funcionalidades disponibles como se ve en la Tabla 20. A través de la actividad principal es posible enviar los datos en segundo plano a una base de datos previamente creada en el alojamiento web sin necesitar que el usuario realice ninguna actividad extra y, a través del uso de *fragments* se puede navegar a través del menú haciendo uso de todas las funcionalidades creadas sin afectar el envío de datos.

**Tabla 20**  
**Componentes de la aplicación desarrollada**

Tipo	Nombre	Funcionalidad
Actividad	MainActivity	Actividad principal donde se obtienen los parámetros principales (latitud, longitud, RSSI, BSSID, identificador del dispositivo, fecha) e inicia una tarea en segundo plano para enviar los datos al servidor; en esta actividad también se crea un menú para llamar a cada uno de los <i>fragments</i> .
<i>Fragment</i>	Exteriores	<i>Fragment</i> que permite visualizar en el mapa el campus universitario junto con algunos lugares de información, también muestra la cobertura en el mapa de las redes exteriores.
	Central	Permite visualizar la cobertura para interiores en el edificio central seleccionado previamente la red y el piso.
	Biblioteca	Permite visualizar la cobertura para interiores en la biblioteca seleccionado previamente la red y el piso.
	Redes	Permite conocer la información de todas las redes disponibles como BSSID, frecuencia, RSSI, etc.
	Contacto	Permite contactarse con el desarrollador.
	Instrucciones	Muestra las instrucciones de uso.

A continuación, se muestra el diagrama de flujo de la actividad principal y de todos los *fragments* que conforma la aplicación:

### Actividad Principal

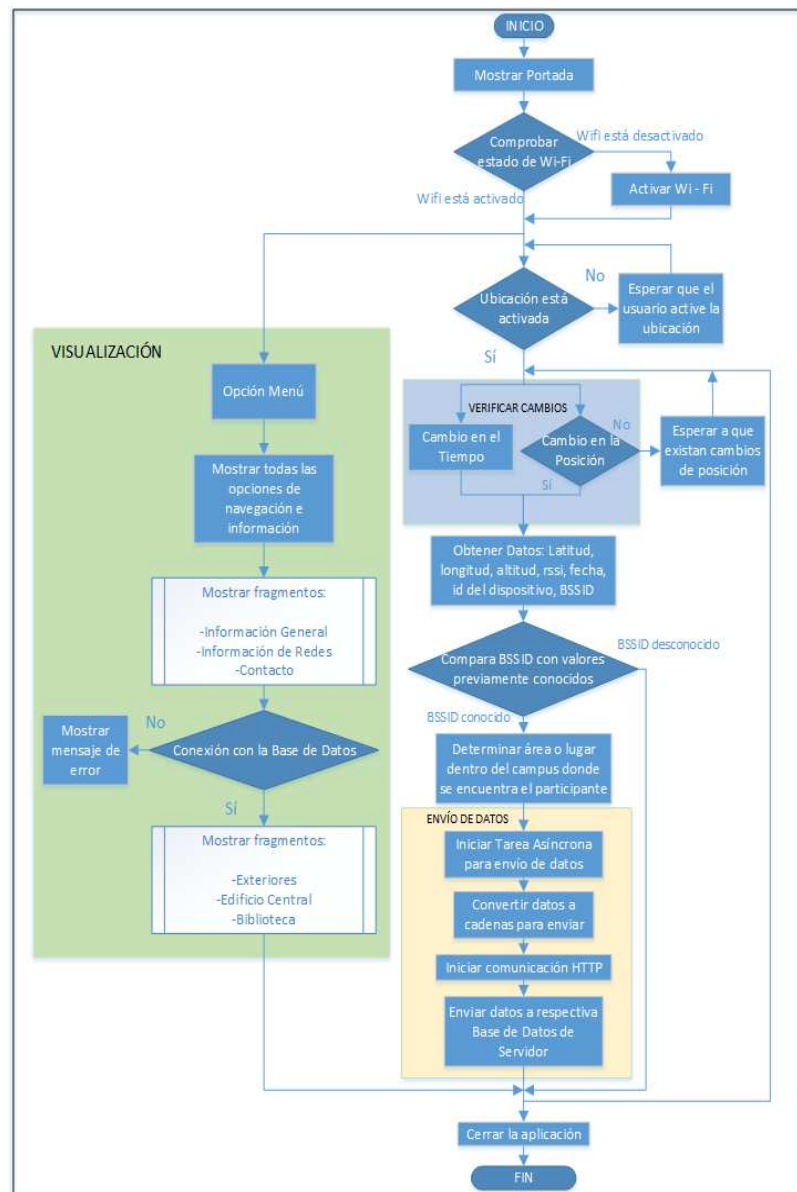
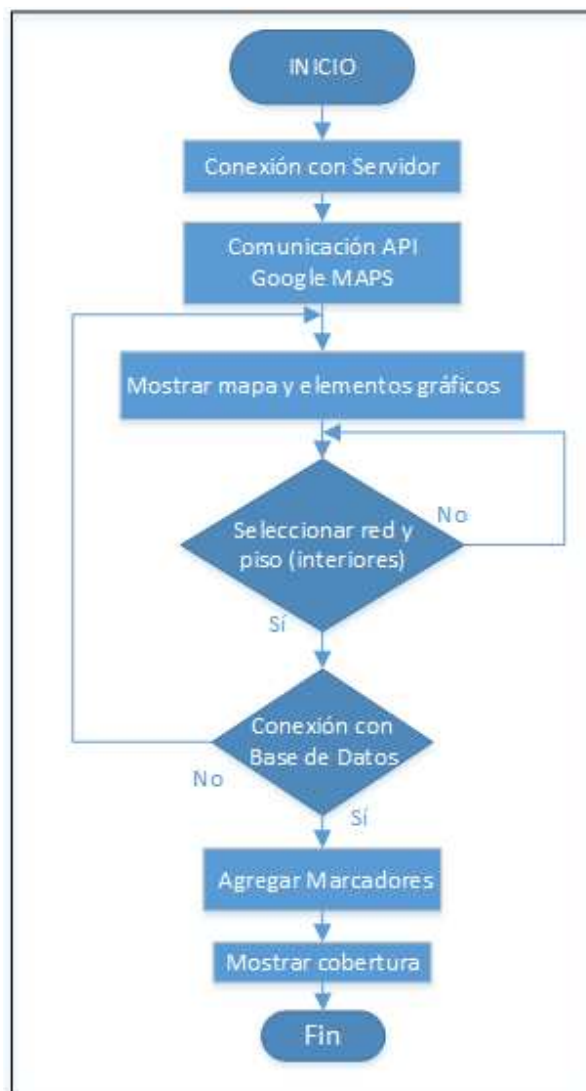


Figura 30: Diagrama de flujo de la actividad principal

En la Figura 30, se observa el diagrama de flujo de la actividad principal, aquí se da a conocer cuál es el algoritmo utilizado para el envío de datos que es realizado en segundo plano, por otro lado, desde esta actividad es posible visualizar todos los fragmentos creados para complementar la funcionalidad de la aplicación.

En la Figura 31, se muestra el diagrama de flujo de los *fragments* implementados (Exteriores, Interiores, Redes) donde se puede conocer su funcionamiento específico.



**Figura 31: Diagrama de flujo fragment exteriores e interiores**

En la Figura 32, se muestra el diagrama de flujo del *fragment* redes con su respectivo algoritmo de funcionamiento.

### Fragment Redes

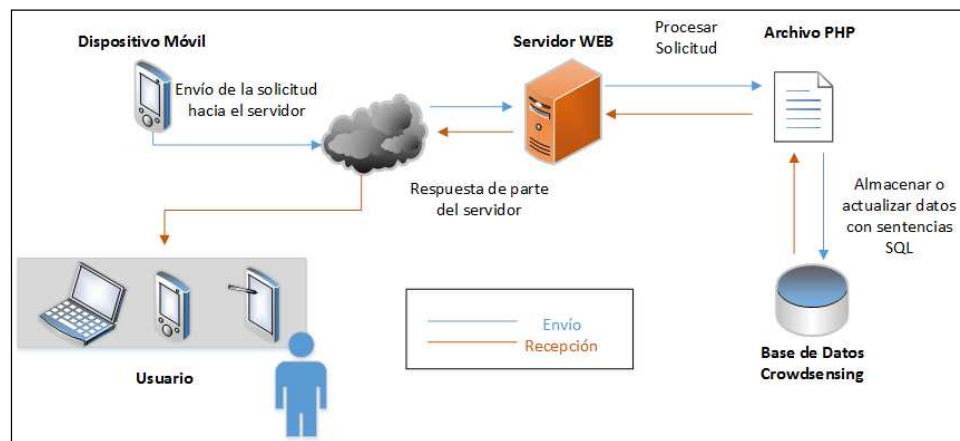


**Figura 32: Diagrama de flujo fragment redes**

### Escenario para el sistema de adquisición de datos

Una vez desarrollado el código del programa en Android Studio, se crea la base de datos para ir almacenando los datos enviados por los participantes y posteriormente visualizarlos en el mapa; esta base de datos se encuentra en el alojamiento web.

Para el proceso de envío de datos, el teléfono inteligente mediante la aplicación Android envía una solicitud HTTP (método post) al servidor, el cual, mediante el lenguaje PHP recibe esta petición y se encarga de almacenar los datos enviados en la correspondiente base de datos para luego mostrar los resultados en dispositivos que tienen un navegador web. El escenario de envío y adquisición de datos, se muestra a continuación:

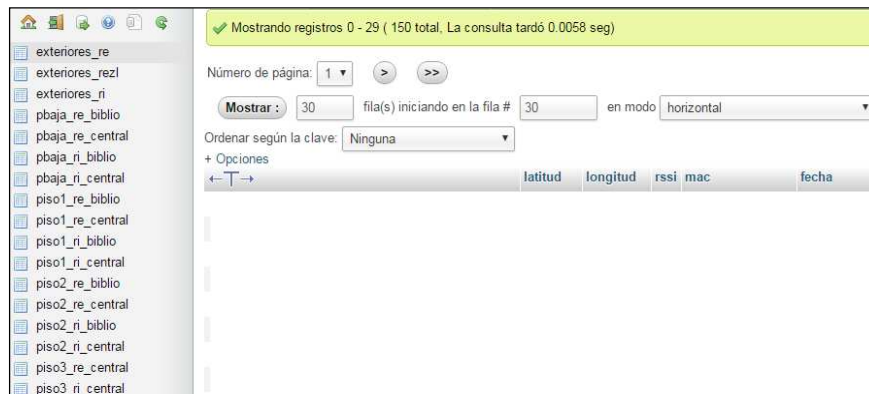


**Figura 33: Escenario para el envío y adquisición de datos desarrollado**

En la Figura 33 se puede ver el proceso de envío de datos, el teléfono inteligente mediante la aplicación Android envía una solicitud HTTP al servidor, el cual, mediante el lenguaje PHP recibe esta petición y se encarga de almacenar los datos enviados en la correspondiente base de datos para luego mostrar los resultados en dispositivos que dispongan de un navegador web.

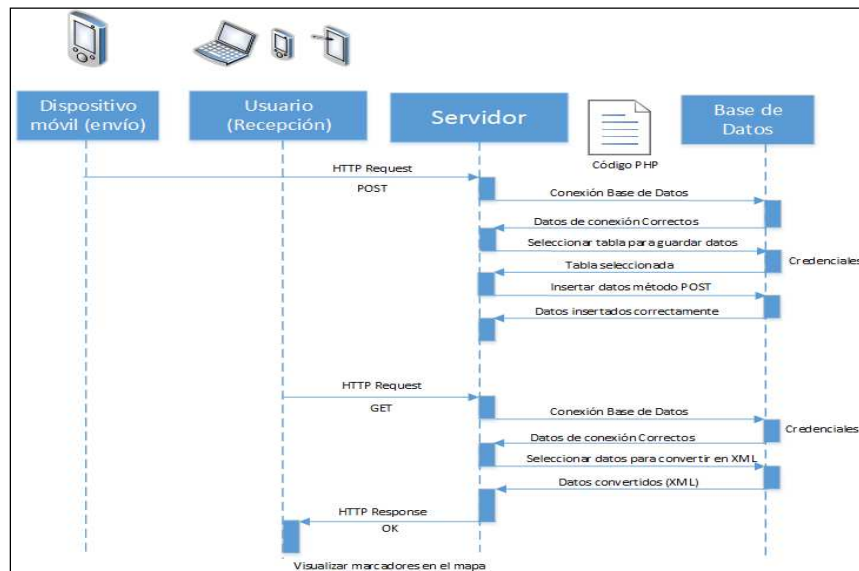
Para la creación de la base de datos, se utiliza el esquema simple entidad – relación cuya implementación se muestra en la Figura 34:





**Figura 34: Tablas pertenecientes a la base de datos creada**

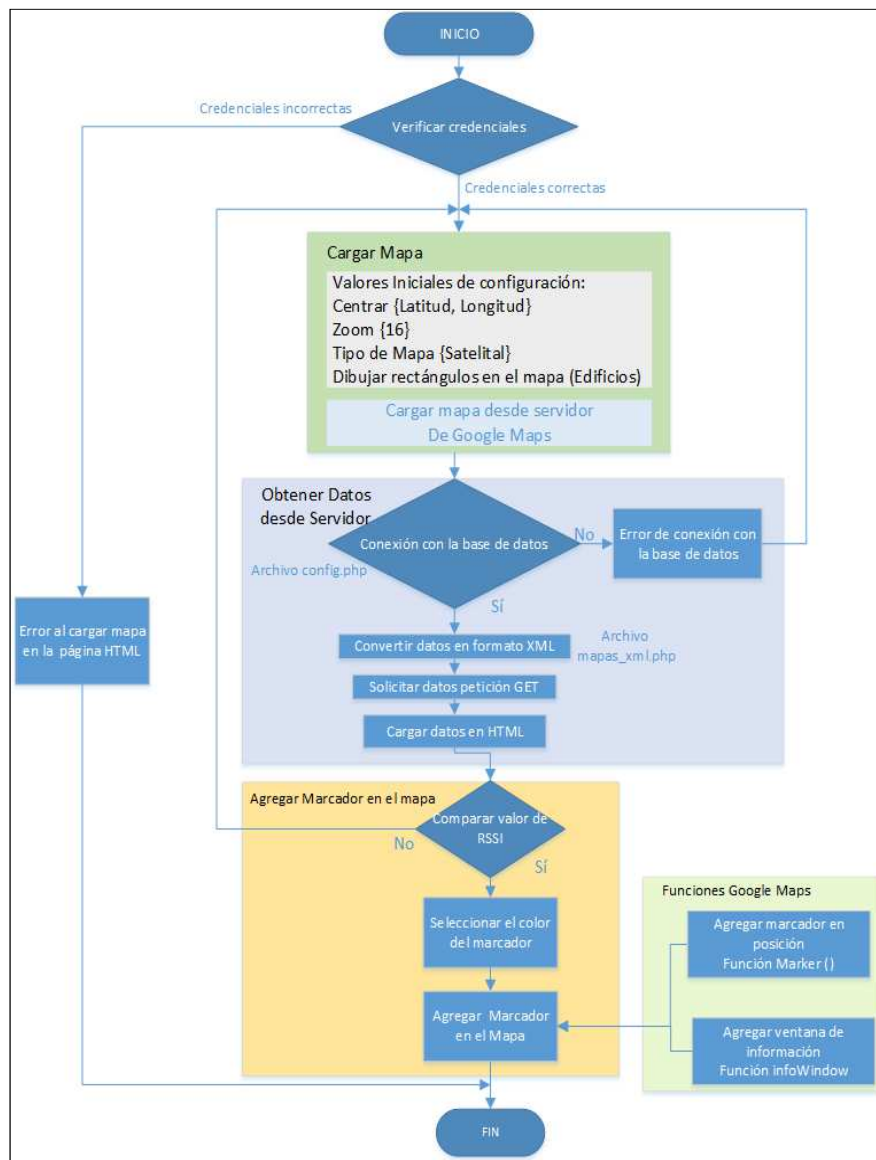
Una vez que se han creado todas las tablas con sus respectivos atributos (latitud, longitud, RSSI, mac o identificador de dispositivo, fecha) para exteriores como para interiores (17 en total), es necesario crear el fichero PHP que permita almacenar los datos enviados por el dispositivo móvil en la tabla correspondiente para su posterior visualización como se muestra en la Figura 35:



**Figura 35: Diagrama de secuencias envío y recepción de datos**

Cuando se dispone de varios datos almacenados en la correspondiente base de datos, éstos son convertidos en formato XML (utilizado para almacenar los datos en forma legible) para agregarlos en forma de marcadores en el mapa.

El algoritmo para para agregar marcadores en el mapa, se muestra en la Figura 36:



**Figura 36: Diagrama de flujo para agregar marcadores en el mapa web**

### **4.3 Proceso de Despliegue e Instalación de la aplicación**

Para exportar una aplicación desde Android Studio, es necesario generar una firma y una clave desde la pestaña Build en el menú de opciones. Terminado el proceso, se obtiene un archivo con extensión APK que puede ser distribuido a los usuarios de varias maneras.

#### **Envío de la aplicación a través de Sitio Web**

Una vez que se ha desarrollado la aplicación, se elige que su distribución sea a través de un sitio web en el cual se encuentre el archivo APK generado previamente. Es por este motivo que se utilizó el alojamiento web que permite subir el archivo APK y mediante un fichero PHP, es posible descargar directamente este archivo al navegador. Para descargar el archivo APK, el usuario simplemente tiene que ingresar desde su dispositivo móvil a un navegador web donde tiene que escribir la siguiente dirección:

**[alonsoalejandro.webcindario.com/descarga.php](http://alonsoalejandro.webcindario.com/descarga.php)**

### **4.4 Respaldo Automático de la Base de Datos**

La base de datos juega un papel fundamental porque contiene la información necesaria para el correcto funcionamiento del sitio web, es por esto que se hace indispensable proteger esta información mediante las copias de seguridad.

La pérdida de información en una base de datos conlleva algunas consecuencias negativas como caídas de la página web, incongruencia de los contenidos, errores en las aplicaciones y en el peor de los casos la pérdida completa de la información de los clientes. A más de todo lo mencionado, se requieren muchas horas extra de trabajo

para tratar de recuperar la información y la confianza de los usuarios (DigitalGuide, 2017).

La principal causa de pérdida de información suele ser debido fallos técnicos internos, por consiguiente, ningún software es capaz de proteger o revertir la pérdida de datos, la única manera en que se puede prevenir la pérdida de información es a través de las copias de seguridad.

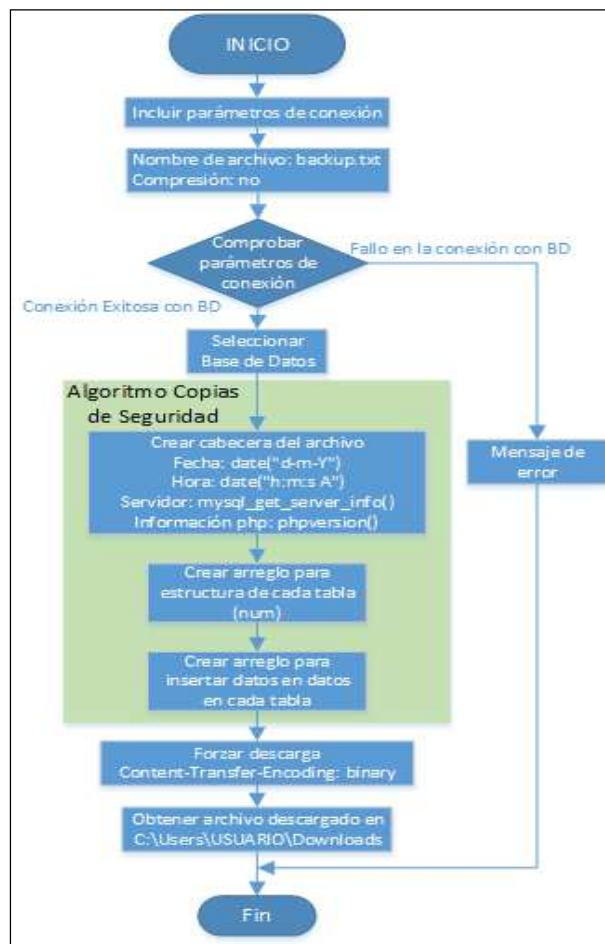
Una copia de seguridad debe proteger los datos ante cualquier posible eventualidad como fallos en el software, fallos en el hardware, software malicioso (virus, gusanos, ataques mal intencionados), catástrofes imprevisibles (robos, incendios), etc. Además, la información guardada tiene que ser recuperable para poder restaurarla (Viteri, 2017).

En definitiva, para llevar a cabo una correcta copia de seguridad, es necesario realizar copias periódicas, tratar de guardar las mismas en distintos equipos físicos o en programas en la red que aseguren su recuperación.

Para realizar el respaldo de la base de datos, se genera un archivo PHP que realice una copia de todas las tablas existentes en la base de datos.

A través de este método, un fichero PHP permite descargar directamente en el navegador la copia de las tablas existentes en la base de datos. El fichero está disponible en el alojamiento web y para su descarga solo es necesario abrir dicho fichero.

En la Figura 37, se muestra el algoritmo utilizado para la descarga del archivo.



**Figura 37: Algoritmo utilizado para realizar copias de seguridad**

El siguiente paso es automatizar el proceso de descarga en el equipo del administrador y para esto se utiliza un archivo *batch* (BAT) junto con el Programador de Tareas de Windows.

El archivo *batch* es un archivo de texto que contiene comandos que pueden ser interpretados sucesivamente por DOS (Disk Operating System). Los archivos *batch* son útiles para automatizar tareas puesto que se ejecutan directamente sin la necesidad de abrir todas las aplicaciones necesarias (Oscar, 2008).

El archivo *batch* únicamente indica la dirección donde se encuentra el fichero PHP que permite descargar la copia de seguridad y para este fin utiliza un navegador web existente en el equipo.

Ahora sólo es necesario programar automáticamente esta descarga mediante el programador de tareas de Windows y para esto se crea una tarea básica donde se indican las siguientes acciones a realizar en la Tabla 21:

**Tabla 21**  
**Tareas llevadas a cabo por el programador de tareas**

<b>Información</b>	<b>Detalle</b>
Nombre	Respaldo Base de Datos
Descripción	Crear un respaldo de la base de datos
Desencadenador	Diariamente; a las 16:20 todos los días
Acción	Iniciar un programa (ruta al archivo <i>batch</i> creado previamente)

Esta tarea básica creada, se va a ejecutar diariamente y de esta manera se respalda la información de la base de datos en el ordenador.

#### **4.5 Definición de Protocolo de pruebas**

##### **Prueba1: Determinación del número de participantes**

Esta prueba tiene como objetivo verificar el número de participantes que han usado el sistema de *Crowdsensing* y validar la obtención de los datos por éstos.

##### **Procedimiento**

**Paso 1:** Ingresar al Alojamiento Web con las credenciales de acceso manual.

**Paso 2:** Ingresar al Servidor de Base de Datos indicando el usuario y clave de acceso.

**Paso 3:** Realizar la siguiente consulta SQL para determinar los identificadores de los participantes que enviaron los datos:

**SELECT mac FROM (tabla de la base de datos)**

Para seleccionar todas las tablas de la base de datos se utiliza la palabra clave UNION que muestra todos los registros que no estén duplicados, por lo tanto, la sentencia queda de la siguiente manera:

```
SELECT mac FROM exteriores_re union SELECT mac FROM exteriores_rezl
union SELECT mac FROM exteriores_ri union SELECT mac FROM
pbaja_re_biblio union SELECT mac FROM pbaja_re_central union SELECT mac
FROM pbaja_ri_biblio union SELECT mac FROM pbaja_ri_central union
SELECT mac FROM piso1_re_biblio union SELECT mac FROM piso1_re_central
union SELECT mac FROM piso1_ri_biblio union SELECT mac FROM
piso1_ri_central union SELECT mac FROM piso2_re_biblio union SELECT mac
FROM piso2_re_central union SELECT mac FROM piso2_ri_biblio union
SELECT mac FROM piso2_ri_central union SELECT mac FROM
piso3_re_central union SELECT mac FROM piso3_ri_central order by mac;
```

**Paso 4:** Realizar un análisis de los datos obtenidos y determinar el número de exacto de participantes.

**Nota:** El atributo mac en la base de datos hace referencia al identificador Android ID que corresponde a un número de 64 bits que se genera al azar cuando el usuario configura al dispositivo por primera vez y debe permanecer constante durante toda la vida del dispositivo (Android Developers, 2016).

## **Prueba 2: Determinar el tiempo de adquisición para enviar el primer dato desde el dispositivo móvil**

Esta prueba consiste en realizar mediciones de tiempo para determinar el tiempo de adquisición requerido por el dispositivo para el envío del primer dato.

### **Procedimiento**

**Paso 1:** Iniciar la Aplicación de *Crowdsensing* en un dispositivo móvil.

**Paso 2:** Medir mediante un cronómetro el tiempo que le toma al dispositivo enviar el primer dato utilizando GPS.

**Paso 3:** Registrar las mediciones en una tabla de datos.

**Paso 4:** Repetir la prueba varias veces (mínimo 10 con el fin de reducir el error cuadrático medio).

**Paso 5:** Medir mediante un cronómetro el tiempo que le toma al dispositivo enviar el primer dato utilizando GPS - Asistido.

**Paso 6:** Repetir paso 3 y 4 utilizando GPS – Asistido.

**Paso 7:** Analizar y comparar los resultados.

**Nota:** De ser el caso, verificar también el tiempo de envío que se requiere para enviar el segundo dato con GPS y GPS - Asistido.

### **Prueba 3: Exactitud del método de posicionamiento**

A través de esta prueba se va a determinar la precisión del método de posicionamiento comparando los datos de posicionamiento de referencia de Google con los obtenidos por GPS y GPS – Asistido.

#### **Procedimiento**

**Paso 1:** Seleccionar lugar de prueba.

**Paso 2:** Obtener datos de referencia de latitud y longitud del lugar de prueba donde el dispositivo permanecerá inmóvil.

**Paso 3:** Obtener datos de latitud y longitud en el dispositivo móvil utilizando GPS.

**Paso 4:** Registrar los datos en una tabla de datos.

**Paso 5:** Repetir la prueba varias veces (mínimo 10 con el fin de reducir el error cuadrático medio).

**Paso 6:** Obtener datos de latitud y longitud en el dispositivo móvil utilizando GPS Asistido.

**Paso 7:** Repetir paso 4 y 5 utilizando GPS – Asistido.

**Paso 8:** Graficar los datos obtenidos tanto con GPS como con GPS - Asistido.

**Paso 9:** Comparar y analizar los resultados obtenidos.



#### Prueba 4: Variación del parámetro RSSI con respecto a la distancia.

El objetivo de esta prueba es comprobar analíticamente en cuánto la distancia al punto de acceso influye en el parámetro RSSI.

##### Procedimiento

**Paso 1:** Identificar y seleccionar un punto de acceso inalámbrico en el campus.

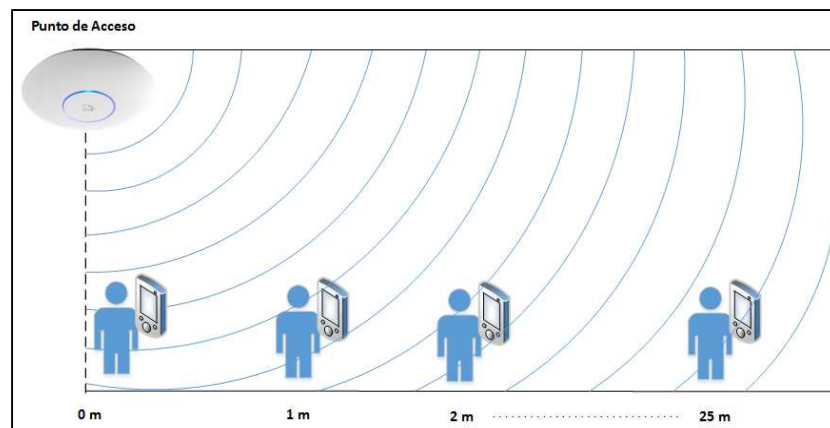
**Paso 2:** Iniciar la aplicación *Crowdsensing* y escoger la opción “Redes”.

**Paso 3:** Medir el parámetro RSSI de la red de interés e ir variando la distancia desde el punto de acceso hasta 25m en intervalos de 1m como se ve en la Figura 38.

**Paso 4:** Repetir el proceso mínimo 10 veces (con el fin de reducir el error cuadrático medio) y registrar los datos en una tabla.

**Paso 5:** Graficar los datos obtenidos.

**Paso 6:** Comprobar analíticamente la variación del parámetro RSSI con respecto a la distancia.



**Figura 38: Escenario para adquisición del parámetro RSSI**

**Prueba 5: Análisis de Envío de Información con Wireshark.**

Esta prueba consiste en utilizar un programa para capturar paquetes (sniffer) como Wireshark para examinar la información y protocolos utilizados durante una consulta web.

**Procedimiento**

**Paso 1:** Abrir Wireshark y elegir la tarjeta de red.

**Paso 2:** Configurar los parámetros y filtros para captura de datos.

**Paso 3:** Iniciar la captura de los datos.

**Paso 4:** Interactuar con la página web mientras se realiza la captura de los datos.

- Visualizar cobertura en exteriores e interactuar con la página web.
- Visualizar cobertura en interiores e interactuar con la página web.

**Paso 5:** Mantener la captura de datos por un minuto.

**Paso 6:** Analizar los datos obtenidos.

**Nota:** Wireshark es un software libre y su descarga se la realiza de manera gratuita en la web (<https://www.wireshark.org/>).

**Prueba 6: Estadísticas del Servidor Web.**

Esta prueba consiste en utilizar las estadísticas del servidor web para conocer el tráfico desde y hacia el servidor durante un tiempo determinado.

**Procedimiento**

**Paso 1:** Ingresar al servidor web con las respectivas credenciales.

**Paso 2:** Seleccionar la opción estadísticas en el panel de control.

**Paso 3:** Seleccionar variables de tiempo para obtener los datos.

**Paso 4:** Analizar los datos obtenidos de tráfico de la red.

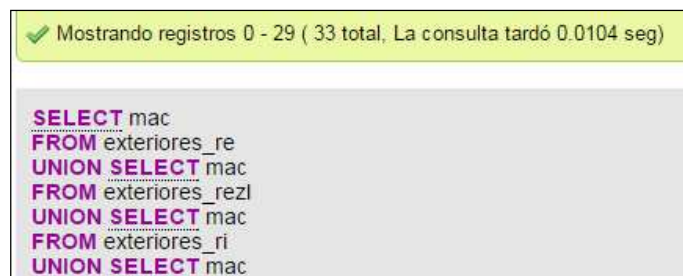
## CAPÍTULO 5

### PRUEBAS DE FUNCIONAMIENTO

#### 5.1 Recolección de Datos por Crowdsensing

##### 5.1.1 Determinación del número de participantes

Siguiendo el procedimiento definido en el protocolo de pruebas descrito en el capítulo anterior, se aplica la sentencia SQL para realizar la consulta y se obtiene el resultado que se muestra en la Figura 39:



```
✓ Mostrando registros 0 - 29 ( 33 total, La consulta tardó 0.0104 seg)

SELECT mac
FROM exteriores_re
UNION SELECT mac
FROM exteriores_rezl
UNION SELECT mac
FROM exteriores_ri
UNION SELECT mac
```

**Figura 39: Resultados obtenidos de la consulta SQL**

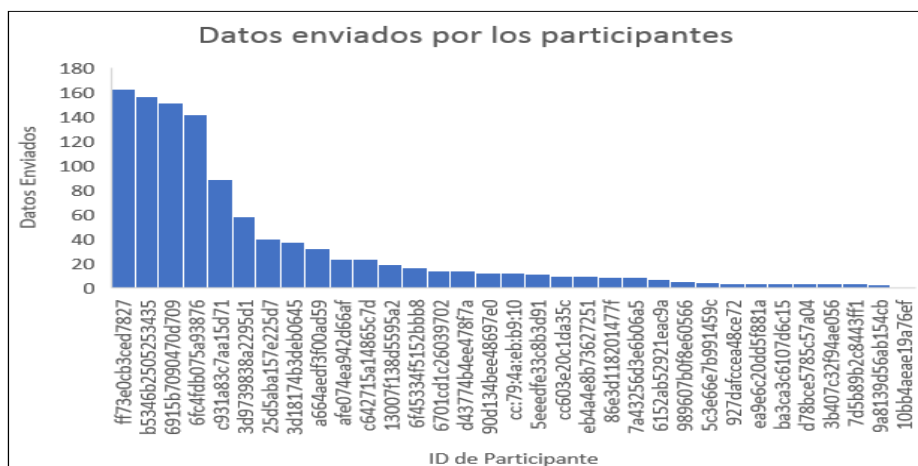
En base a lo descrito en la Figura 39, se puede constatar que existen 33 usuarios que han enviado los datos hacia la plataforma *Crowdsensing* con lo que se ha podido cumplir con el requisito de que existan al menos 20 participantes como mínimo para el envío de datos según el alcance propuesto.

La consulta además devuelve los identificadores únicos (Android ID) de los usuarios que están recolectados en la tabla 22:

**Tabla 22**  
**Identificadores únicos de los participantes**

10bb4aeae19a76ef	6701cd1c26039702	989607b0f8e60566	cc:79:4a:eb:b9:10
13007f138d5595a2	6915b7090470d709	9a8139d56ab154cb	d43774b4ee478f7a
25d5aba157e225d7	6f45334f5152bbb8	a664aedf3f00ad59	d78bce5785c57a04
3b407c32f94ae056	6fc4fdb075a93876	afe074ea942d66af	ea9e6c20dd5f881a
3d18174b3deb0645	7a43256d3e6b06a5	b5346b2505253435	eb4a4e8b73627251
3d9739838a2295d1	7d5b89b2c8443ff1	ba3ca3c6107d6c15	ff73e0cb3ced7827
5c3e66e7b991459c	86e3d118201477f	c642715a14865c7d	
5eedfe33c8b3d91	90d134bee48697e0	c931a83c7aa15d71	
6152ab52921eac9a	927dafccea48ce72	cc603e20c1da35c	

Con esta información, además se puede determinar la participación de cada usuario utilizando la sentencia **select count(mac)**, obteniendo los resultados que son graficados a continuación:



**Figura 40: Número de datos enviados por los participantes**

Analizando la Figura 40, se pueden conocer los identificadores de todos los dispositivos móviles y el número de datos enviados por cada uno, se puede observar que el número máximo de datos enviados por un usuario fue 162 y el número mínimo fue 1, de la misma forma el promedio de envío de datos de los participantes es 33.

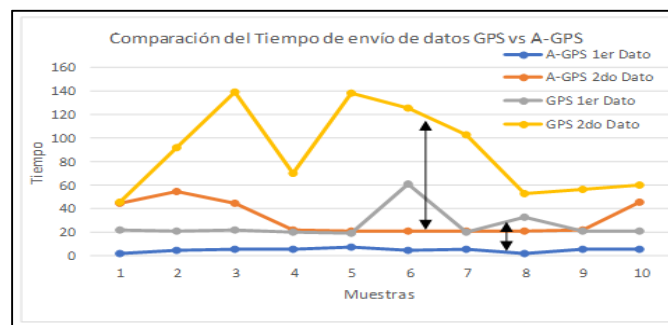
### 5.1.2 Determinar el tiempo requerido para enviar datos desde el dispositivo móvil

Esta prueba se realizó en interiores, se procedió a tomar el tiempo requerido para el envío del primer y segundo dato utilizando GPS y GPS – Asistido, de esta manera, la prueba planteada en el capítulo anterior, devuelve los siguientes valores:

**Tabla 23**  
**Tiempo requerido para el envío de datos**

Muestra	GPS		A – GPS	
	1er Dato (seg)	2do Dato (seg)	1er Dato (seg)	2do Dato (seg)
<b>1</b>	22	46	2	45
<b>2</b>	21	92	5	55
<b>3</b>	22	139	6	45
<b>4</b>	20	70	6	22
<b>5</b>	19	138	7	21
<b>6</b>	61	126	5	21
<b>7</b>	20	103	6	21
<b>8</b>	33	53	2	21
<b>9</b>	21	57	6	22
<b>10</b>	21	60	6	46
<b>Prom</b>	<b>26</b>	<b>88,4</b>	<b>5,1</b>	<b>31,9</b>

Al analizar la Tabla 23, se observa que al utilizar GPS – Asistido, el tiempo requerido es menor comparado con el envío de datos utilizando únicamente GPS.

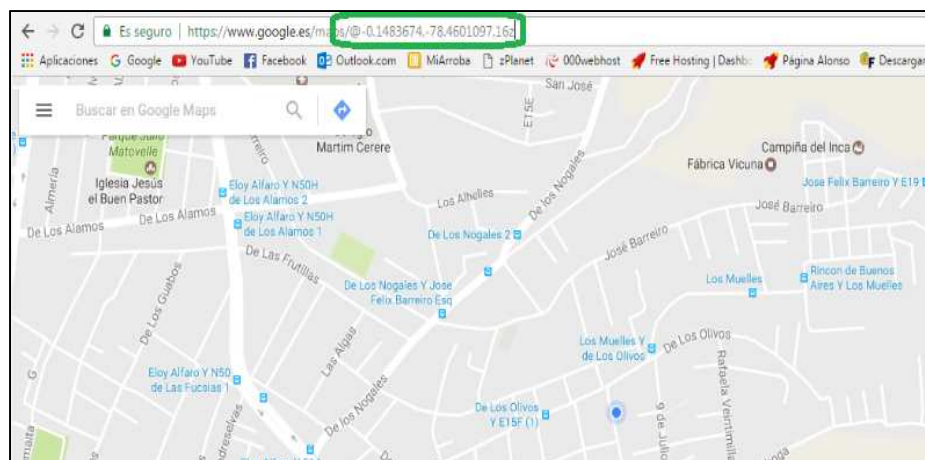


**Figura 41: Comparación Tiempo de envío de datos GPS vs A-GPS**

En la Figura 41, se realiza la comparación gráfica de los tiempos requeridos para el envío de datos, esta gráfica permite corroborar que cuando se utiliza GPS – Asistido el tiempo necesario para el envío del primer dato es en promedio cinco veces menor mientras que el envío del segundo dato es en promedio tres veces menor comparado con GPS.

### 5.1.2 Exactitud del método de posicionamiento

El primer paso consiste en seleccionar el lugar para usarlo como patrón de posición de referencia, puesto que no es posible utilizar dispositivos GPS comerciales en interiores se utiliza la herramienta de localización de Google Maps la cual permite obtener los siguientes valores de latitud y longitud en el lugar de referencia como se muestra en la Figura 42:



**Figura 42: Ubicación de referencia en el lugar de pruebas**

De esta manera, la ubicación de referencia del participante para esta prueba es:

**Latitud:** -0.1483674

**Longitud:** -78.4601097

En base a la prueba propuesta para evaluar la exactitud en el método de posicionamiento descrito en la sección de protocolo de pruebas, se procede a obtener los valores de latitud y longitud en el mismo lugar utilizando GPS y GPS – Asistido y se calcula el porcentaje de error en los decimales obtenidos mediante la ecuación 1:

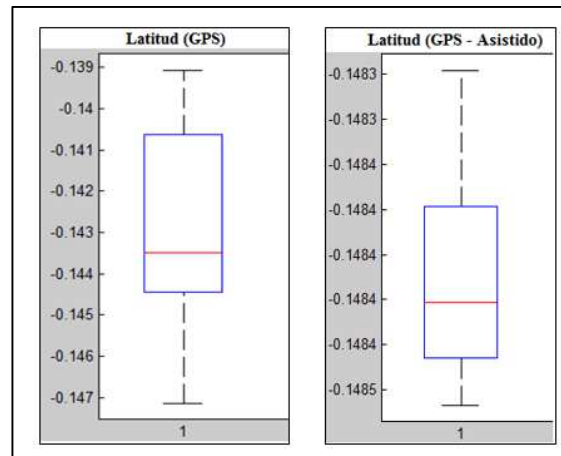
$$(|\text{valor real} - \text{valor medido}|/|\text{valor real}|) * 100\% \quad (1)$$

**Tabla 24**  
**Medidas de latitud y longitud obtenidas**

N	GPS	% Error GPS	A – GPS	% Error A-GPS
1	Latitud: -0.13908	6.25	Latitud: -0.1483841	0.011
	Longitud: -78.45832	0.38	Longitud: -78.4601191	0.002
2	Latitud: -0.1401	5.57	Latitud: -0.1484296	0.042
	Longitud: -78.45664	0.75	Longitud: -78.459965	0.031
3	Latitud: -0.14064	5.2	Latitud: -0.1484668	0.067
	Longitud: -78.45638	-0.8	Longitud: -78.4599829	0.027
4	Latitud: -0.14317	3.5	Latitud: -0.1484474	0.054
	Longitud: -78.45087	2	Longitud: -78.4599649	0.031
5	Latitud: -0.14342	3.33	Latitud: -0.1484247	0.038
	Longitud: -78.4515	1.87	Longitud: -78.4599659	0.031
6	Latitud: -0.14444	2.64	Latitud: -0.1483185	0.033
	Longitud: -78.45341	1.46	Longitud: -78.460162	0.011
7	Latitud: -0.1466	1.19	Latitud: -0.1484458	0.053
	Longitud: -78.45731	0.61	Longitud: -78.4600355	0.016
8	Latitud: -0.14356	3.24	Latitud: -0.1483688	0.0009
	Longitud: -78.4606	0.11	Longitud: -78.4600349	0.016
9	Latitud: -0.14415	0.28	Latitud: -0.1483788	0.0076
	Longitud: -78.46042	0.067	Longitud: -78.46014	0.0065
10	Latitud: -0.14714	0.83	Latitud: -0.1484176	0.034
	Longitud: -78.46001	0.02	Longitud: -78.459965	0.031
<b>Promedio</b>	<b>Latitud: -0.14323</b>	<b>3.46</b>	<b>Latitud: -0,14840821</b>	<b>0.0088</b>
	<b>Longitud: -78,456546</b>	<b>0.77</b>	<b>Longitud: -78,46003352</b>	<b>0.016</b>

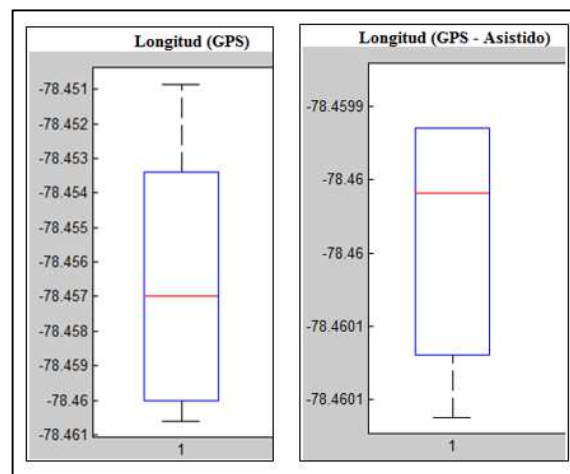
Al analizar los datos obtenidos, tanto en la Tabla 24, se observa que cuando se utiliza únicamente GPS, el promedio de error es mayor, mientras que cuando se utiliza GPS – Asistido, los valores son mucho más cercanos al valor real de referencia.

Teniendo en cuenta los valores obtenidos mediante GPS y GPS - Asistido, se procede a utilizar la herramienta de Matlab para realizar una comparación gráfica de los resultados obtenidos.



**Figura 43:** Comparación de Latitud (GPS – GPS-Asistido)

En la Figura 43 se observan los datos de latitud utilizando GPS y GPS – Asistido, se puede observar que al utilizar GPS – Asistido se obtiene un promedio con un valor más cercano al lugar de referencia dado (-0.1483674).



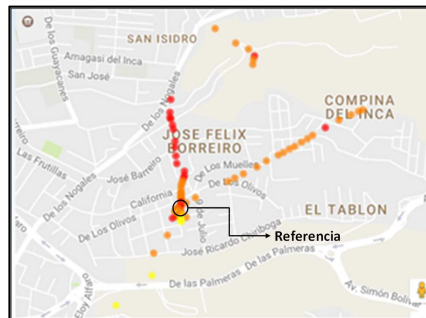
**Figura 44:** Comparación de Longitud (GPS – GPS-Asistido)



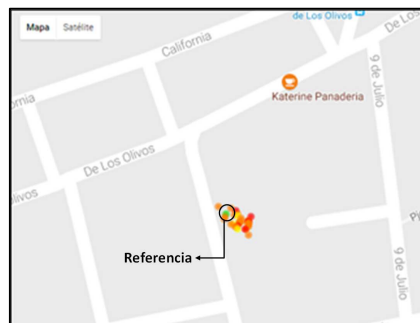
También se procede a comparar los valores obtenidos de longitud en la Figura 44, encontrando que con GPS – Asistido el valor promedio obtenido es más cercano al valor de referencia (-78.4601097).

Cabe indicar que, aunque para el caso de GPS se obtiene un error numérico pequeño, esto implica una gran variación de metros en la posición, en cambio cuando se utiliza GPS – Asistido, se obtiene un error promedio en la distancia de 10m.

A continuación, se procede a graficar los valores obtenidos tanto con GPS como con GPS-Asistido en el mapa, junto con la ubicación de referencia:



**Figura 45: Marcadores en el mapa utilizando GPS**



**Figura 46: Marcadores en el mapa utilizando GPS - Asistido**

Después de analizar las gráficas de las Figuras 45 y 46 se puede corroborar gráficamente que el error obtenido cuando se utiliza GPS en interiores es mucho mayor que cuando se utiliza GPS – Asistido, es por este motivo que la aplicación se desarrolló utilizando GPS – Asistido.

### 5.1.3 Variación de RSSI con respecto a la distancia

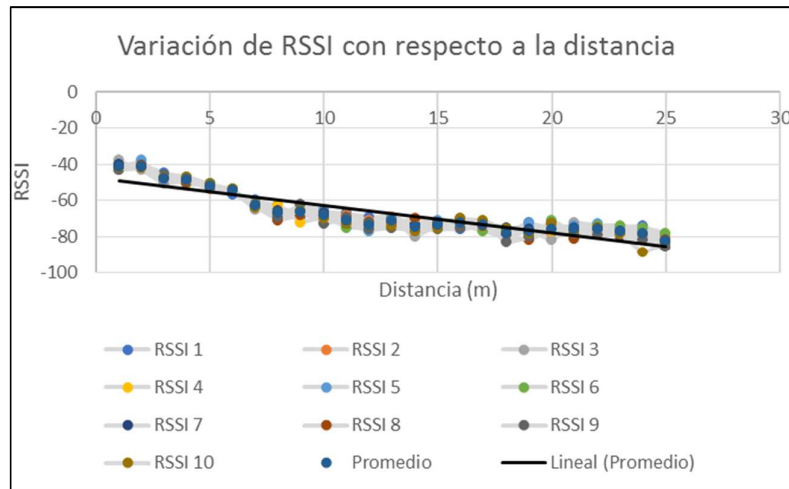
Al usar el parámetro RSSI para determinar la intensidad de la señal, es necesario realizar medidas de este parámetro según la distancia ya que es un parámetro variante.

En esta prueba se realizaron 10 mediciones desde el punto de acceso hasta 25m de distancia. La prueba se realizó en el piso 3 del Bloque A (Edificio Central) de la Universidad de las Fuerzas Armadas - ESPE y se obtuvieron los resultados de la Tabla 25:

**Tabla 25**  
**Valores Obtenidos de RSSI**

Distancia (m)	RSSI 1 (dbm)	RSSI 2 (dbm)	RSSI 3 (dbm)	RSSI 4 (dbm)	RSSI 5 (dbm)	RSSI 6 (dbm)	RSSI 7 (dbm)	RSSI 8 (dbm)	RSSI 9 (dbm)	RSSI 10 (dbm)	Promedio
<b>1</b>	-43	-42	-38	-41	-42	-41	-40	-42	-43	-42	<b>-41</b>
<b>2</b>	-40	-41	-43	-41	-38	-42	-41	-41	-42	-42	<b>-41</b>
<b>3</b>	-45	-50	-51	-49	-49	-47	-47	-48	-46	-47	<b>-48</b>
<b>4</b>	-47	-48	-48	-47	-49	-51	-47	-50	-49	-47	<b>-48</b>
<b>5</b>	-51	-54	-52	-53	-51	-51	-51	-52	-53	-51	<b>-52</b>
<b>6</b>	-57	-55	-54	-55	-54	-55	-54	-54	-55	-54	<b>-55</b>
<b>7</b>	-60	-63	-65	-62	-63	-61	-63	-63	-62	-64	<b>-63</b>
<b>8</b>	-63	-70	-64	-63	-69	-68	-66	-71	-69	-67	<b>-67</b>
<b>9</b>	-69	-67	-68	-72	-63	-64	-67	-68	-62	-66	<b>-67</b>
<b>10</b>	-66	-70	-67	-69	-65	-67	-67	-68	-73	-70	<b>-68</b>
<b>11</b>	-67	-68	-73	-70	-73	-75	-73	-71	-69	-73	<b>-71</b>
<b>12</b>	-69	-71	-72	-73	-77	-75	-72	-74	-76	-73	<b>-73</b>
<b>13</b>	-69	-69	-70	-69	-70	-73	-72	-72	-75	-74	<b>-71</b>
<b>14</b>	-73	-74	-80	-73	-74	-75	-73	-70	-76	-77	<b>-75</b>
<b>15</b>	-75	-74	-73	-72	-71	-75	-72	-73	-76	-75	<b>-74</b>
<b>16</b>	-76	-72	-70	-73	-71	-73	-71	-71	-75	-70	<b>-72</b>
<b>17</b>	-73	-74	-76	-74	-73	-77	-73	-74	-72	-71	<b>-74</b>
<b>18</b>	-78	-76	-80	-75	-82	-80	-75	-79	-83	-76	<b>-78</b>
<b>19</b>	-72	-77	-76	-75	-73	-81	-76	-82	-80	-78	<b>-77</b>
<b>20</b>	-76	-80	-82	-77	-75	-71	-75	-75	-73	-73	<b>-76</b>
<b>21</b>	-76	-75	-72	-75	-74	-78	-75	-81	-78	-77	<b>-76</b>
<b>22</b>	-78	-76	-74	-78	-73	-74	-76	-75	-79	-76	<b>-76</b>
<b>23</b>	-75	-79	-77	-76	-75	-74	-77	-80	-79	-78	<b>-77</b>
<b>24</b>	-74	-77	-75	-77	-76	-75	-80	-79	-82	-89	<b>-78</b>
<b>25</b>	-81	-82	-80	-85	-84	-78	-83	-82	-86	-83	<b>-82</b>

Con los datos obtenidos en esta prueba, se procede a graficar los valores como se muestra a continuación:



**Figura 47: Variación de RSSI respecto a la distancia**

Como se observa en la Figura 47, la distancia es un valor que tiene relación con el parámetro RSSI, ya que, a mayor distancia del punto de acceso, este valor disminuye.

Es por este motivo que en la Tabla 26 se han fijado los colores (rangos de luz visible de menor a mayor longitud de onda) para los diferentes niveles obtenidos de RSSI y de esta manera graficar la potencia en el mapa de cobertura para la aplicación Web.

**Tabla 26**  
**Colores para rango de valores de RSSI**

Rango RSSI	Color	Intensidad
RSSI mayor que -60	Verde	Excelente
RSSI entre -60 y -70	Amarillo	Muy Buena
RSSI entre -70 y -80	Tomate	Buena
RSSI menor que -80	Rojo	Regular

**Nota:** En la aplicación móvil, este rango para RSSI puede ser observado a través de la opción información.

### 5.1.4 Uso de la herramienta de análisis Wireshark

Mediante el uso de Wireshark se puede conocer los protocolos utilizados en la arquitectura cliente-servidor, el primer paso es seleccionar la tarjeta de red y configurar los parámetros, a continuación, se inicia la captura y se ingresa a la página web, obteniendo el siguiente resultado:

Source	Destination	Protocol	Leng	Info
192.168.100.2	5.57.226.202	HTTP	591	GET / HTTP/1.1
5.57.226.202	192.168.100.2	HTTP	471	HTTP/1.1 200 OK (text/html)
192.168.100.2	5.57.226.202	HTTP	619	GET /Presentaci%C3%83n/images/tooplate_wrappe.png HTTP/1.1
192.168.100.2	104.25.207.8	HTTP	623	GET /?_muid=c055704515ef29f1d51fbed762bd10ad0f27fd1&h=1698300&t=1486833981&k=fea2380d02bfe9157895b42aa HTTP/1.1
192.168.100.2	216.58.212.131	HTTP	485	GET /csi?v=2&s=mapsapi3&v3v=27.12&action=apiboot2&e=10_1_0_10_2_0&rt=main.14 HTTP/1.1
5.57.226.202	192.168.100.2	HTTP	325	HTTP/1.1 404 Not Found (text/html)
192.168.100.2	5.57.226.202	HTTP	616	GET /Presentaci%C3%83n/images/tooplate_mai.jpg HTTP/1.1
192.168.100.2	5.196.26.93	HTTP	471	GET /hitv4.php?digit=amini&page=d17c0775872be8c2c0adf38f3fd3151a&t=1481684586 HTTP/1.1
104.25.207.8	192.168.100.2	HTTP	59	HTTP/1.1 200 OK (application/javascript)
216.58.212.131	192.168.100.2	HTTP	366	HTTP/1.1 204 No Content
5.57.226.202	192.168.100.2	HTTP	325	HTTP/1.1 404 Not Found (text/html)
5.196.26.93	192.168.100.2	HTTP	60	HTTP/1.1 200 OK (application/javascript)
192.168.100.2	62.212.87.142	HTTP	534	GET /b/9883108d8c26315993?&1486833973714.7822&nw=false&cm=true&fp=false&pos=bottom&w=320&h=100 HTTP/1.1
192.168.100.2	5.196.26.93	HTTP	502	GET /count.php?ac=0&page=d17c0775872be8c2c0adf38f3fd3151a&digit=amini&temp=792892469 HTTP/1.1
5.196.26.93	192.168.100.2	HTTP	60	HTTP/1.1 200 OK (PNG)
62.212.87.142	192.168.100.2	HTTP	54	HTTP/1.1 204 No Content
192.168.100.2	104.25.207.8	HTTP	936	POST /b05e705195fc4e9cb3d0155b9152841e_checkpriv.php HTTP/1.1
104.25.207.8	192.168.100.2	HTTP	59	HTTP/1.1 200 OK (application/json)

**Figura 48: Protocolo HTTP utilizado para la comunicación**

En la Figura 48 se puede apreciar la comunicación entre cliente y servidor utilizando el protocolo HTTP.

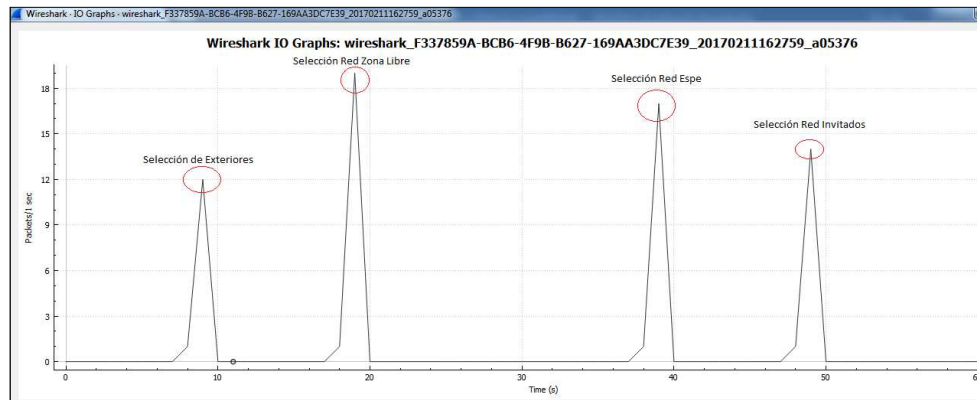
Al analizar las direcciones IP de origen y destino utilizando WHOIS se puede encontrar la información que se presenta en la Tabla 27:

**Tabla 27**  
**Determinación de propietarios web**

Dirección IP	Pertenciente	Uso
5.57.226.202 5.196.26.93	Centro de Coordinación de redes IP europeas (Servidor Miarroba)	Identificador del Servidor utilizado para el desarrollo de la aplicación.
192.168.100.2	IP del usuario	Identificador de la red local.
104.25.207.8	Cloudflare	Empresa proveedora de seguridad web.
216.58.212.131	Google	Cargar mapa. Api Google Maps.

**Nota:** WHOIS es un protocolo que permite conocer la información del dueño de una dirección IP en internet. (<https://who.is/>).

Ahora se procede a interactuar con la página web y a visualizar la cobertura en exteriores mientras se captura los datos en Wireshark durante un minuto:



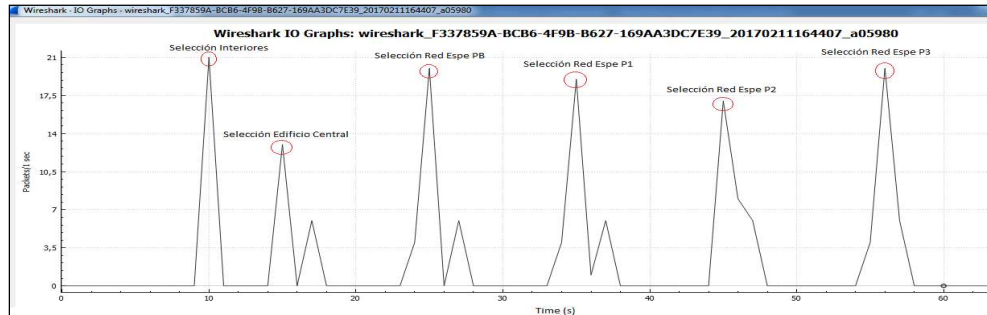
**Figura 49: Paquetes / segundo enviados (Exteriores)**

También es posible verificar esta comunicación mediante el protocolo HTTP utilizando Wireshark y los resultados se muestran en la Figura 50:

Source	Destination	Protocol	Leng	Info
192.168.100.2	5.57.226.202	HTTP	663	GET /Enlaces/exteriores1.html HTTP/1.1
5.57.226.202	192.168.100.2	HTTP	12..	HTTP/1.1 200 OK (text/html)
192.168.100.2	104.25.206.8	HTTP	647	GET /?_muid=51fd10d6c360bc4ec74c862935260fe50e21e08b&h=1698300&t=1486848498&k=936c5f65
192.168.100.2	64.233.162.94	HTTP	509	GET /csi?v=2&s=mapsapi3&v3v=27.12&action=apiboot2&e=10_1_0_10_2_0&rt=main.11 HTTP/1.1
192.168.100.2	5.57.226.202	HTTP	619	GET /Presentaci%C3%83n/images/tooplate_wrappe.png HTTP/1.1
192.168.100.2	5.57.226.202	HTTP	616	GET /Presentaci%C3%83n/images/tooplate_mai.jpg HTTP/1.1
104.25.206.8	192.168.100.2	HTTP	60	HTTP/1.1 200 OK (application/javascript)
5.57.226.202	192.168.100.2	HTTP	326	HTTP/1.1 404 Not Found (text/html)
5.57.226.202	192.168.100.2	HTTP	326	Continuation
64.233.162.94	192.168.100.2	HTTP	366	HTTP/1.1 204 No Content
192.168.100.2	5.57.226.202	HTTP	694	GET /Graficar_Mapa/Exteriores_Red_Espe_Zona_Libre/index.html HTTP/1.1
5.57.226.202	192.168.100.2	HTTP	788	HTTP/1.1 200 OK (text/html)
192.168.100.2	104.25.206.8	HTTP	678	GET /?_muid=51fd10d6c360bc4ec74c862935260fe50e21e08b&h=1698300&t=1486848509&k=8ddfc56
192.168.100.2	64.233.162.94	HTTP	540	GET /csi?v=2&s=mapsapi3&v3v=27.12&action=apiboot2&e=10_1_0_10_2_0&rt=main.11 HTTP/1.1
192.168.100.2	5.57.226.202	HTTP	619	GET /Presentaci%C3%83n/images/tooplate_wrappe.png HTTP/1.1
192.168.100.2	5.57.226.202	HTTP	616	GET /Presentaci%C3%83n/images/tooplate_mai.jpg HTTP/1.1
104.25.206.8	192.168.100.2	HTTP	60	HTTP/1.1 200 OK (application/javascript)

**Figura 50: Protocolo HTTP en pruebas (exteriores)**

Continuando con las pruebas definidas en el capítulo 4, durante un minuto se utiliza Wireshark para capturar datos obtenidos mientras se muestra la cobertura en interiores y se tiene los siguientes resultados:



**Figura 51: Paquetes / segundo enviados (Interiores)**

Al analizar las Figuras 49 y 51, se puede conocer el número de paquetes enviados mientras el solicitante navega por la página web visualizando la cobertura en exteriores e interiores.

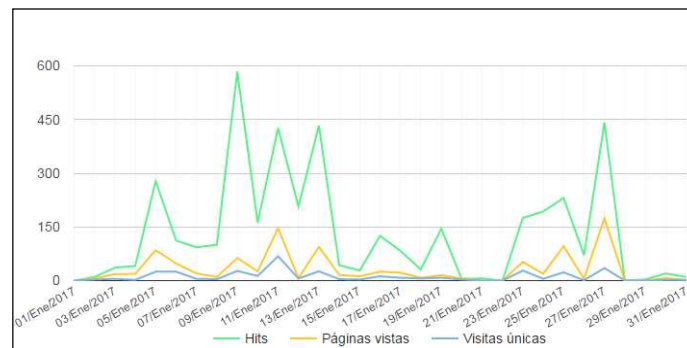
Wireshark también permite conocer la forma de interacción entre el servidor con un usuario en donde se obtienen datos de tiempo, sentido de flujo de los datos, puertos, números de secuencia y acuses de recibo, protocolos, entre otros datos útiles para verificar el funcionamiento como se muestra en la Figura 52.

Time	192.168.100.2	5.57.226.202	Comment
35.264046	49625→80	[SYN] Seq=0 Win=8192 Len=0	TCP: 49625→80 [SYN] Seq=0 Win=8192 Len=0 ...
35.445814	80→49625	[SYN, ACK] Seq=0 Ack=1 Win=0	TCP: 80→49625 [SYN, ACK] Seq=0 Ack=1 Win=...
35.445926	49625→80	[ACK] Seq=1 Ack=1 Win=0	TCP: 49625→80 [ACK] Seq=1 Ack=1 Win=66304...
35.452702	49625	GET /Presentaci%C3%B3n/images/hooplat...	HTTP: GET /Presentaci%C3%B3n/images/hooplat...
35.631444	80→49625	[ACK] Seq=1 Ack=563 Win=158...	TCP: 80→49625 [ACK] Seq=1 Ack=563 Win=158...
35.631959	49625	[TCP segment of a reassembled PDU]	TCP: [TCP segment of a reassembled PDU]
35.632205	49625	HTTP/1.1 404 Not Found (text/html)	HTTP: HTTP/1.1 404 Not Found (text/html)
35.632257	49625→80	[ACK] Seq=563 Ack=1686 Win=...	TCP: 49625→80 [ACK] Seq=563 Ack=1686 Win=...
35.693833	49625	GET /Graficar_Mapa/Interiores_Piso_1_Esp...	HTTP: GET /Graficar_Mapa/Interiores_Piso_1_Esp...
35.881314	49625	HTTP/1.1 200 OK	HTTP/XML: HTTP/1.1 200 OK
36.081688	49625→80	[ACK] Seq=1135 Ack=228 Win=...	TCP: 49625→80 [ACK] Seq=1135 Ack=2281 Win=...
37.210266	49625→80	[FIN, ACK] Seq=1135 Ack=228 Win=...	TCP: 49625→80 [FIN, ACK] Seq=1135 Ack=228...
37.389172	80→49625	[FIN, ACK] Seq=2281 Ack=113...	TCP: 80→49625 [FIN, ACK] Seq=2281 Ack=113...
37.389255	49625→80	[ACK] Seq=1136 Ack=228 Win=...	TCP: 49625→80 [ACK] Seq=1136 Ack=2282 Win=...

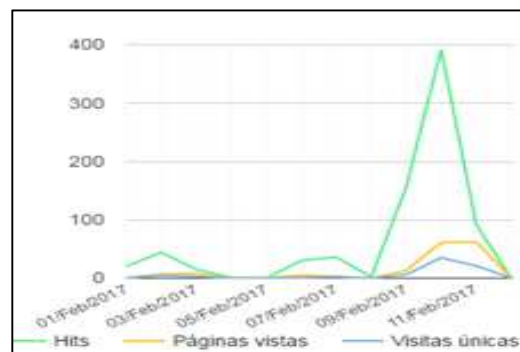
**Figura 52: Flujo de datos**

### 5.1.5 Utilizar las estadísticas del Servidor web

A través del uso de la herramienta estadísticas del alojamiento web utilizado, se pueden conocer algunos datos adicionales que son muy importantes como por ejemplo el tráfico de la red, el número de peticiones, impresiones de página, visitas únicas, además esta opción permite visualizar los datos de una forma gráfica seleccionando el período de tiempo de interés como se ve en las Figuras 53 y 54.

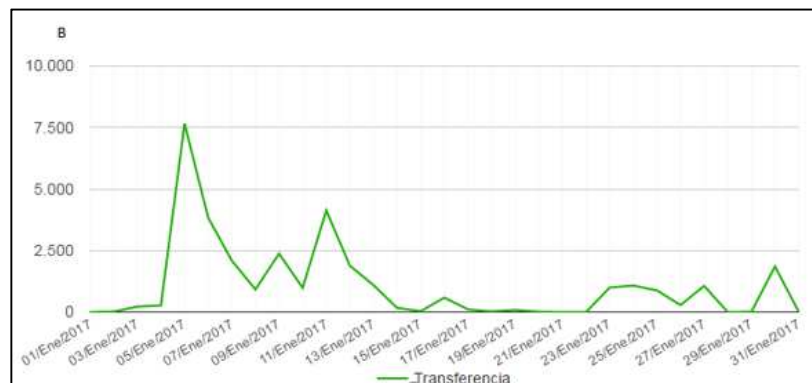


**Figura 53: Estadísticas de tráfico mes de enero 2017**

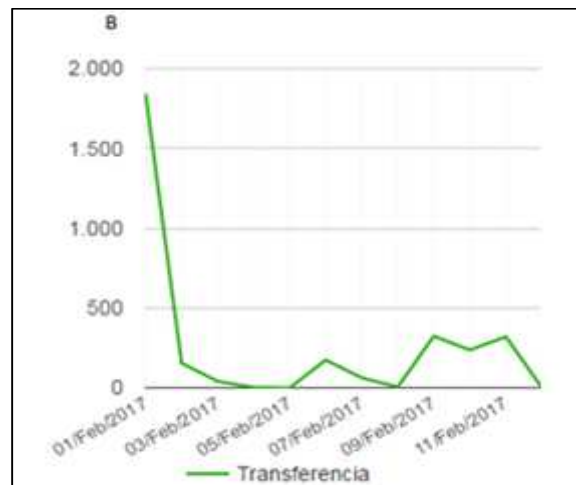


**Figura 54: Estadísticas de tráfico mes de febrero 2017 hasta el 12-02-2017**

También es posible verificar gráficamente la transferencia de datos en la red cuyos resultados se muestran en las Figuras 55 y 56:



**Figura 55: Estadísticas de Transferencias en el mes de enero**



**Figura 56: Estadísticas de transferencias en el mes de febrero hasta 12-02-2017**

Estas herramientas que provee el alojamiento web resultan útiles permitiendo conocer las estadísticas del sitio web creado.

Resulta importante mencionar que las transferencias realizadas no son un problema puesto que el alojamiento miarroba cuenta con un límite de transferencia ilimitado a diferencia de otros alojamientos web gratuitos.



## **5.2 Sistema de Monitorización**

El sistema de monitorización en conjunto brinda información de las redes inalámbricas de la ESPE a través de una medición de múltiples dispositivos *Crowdsensing*.

### **5.2.1 Resultados Obtenidos**

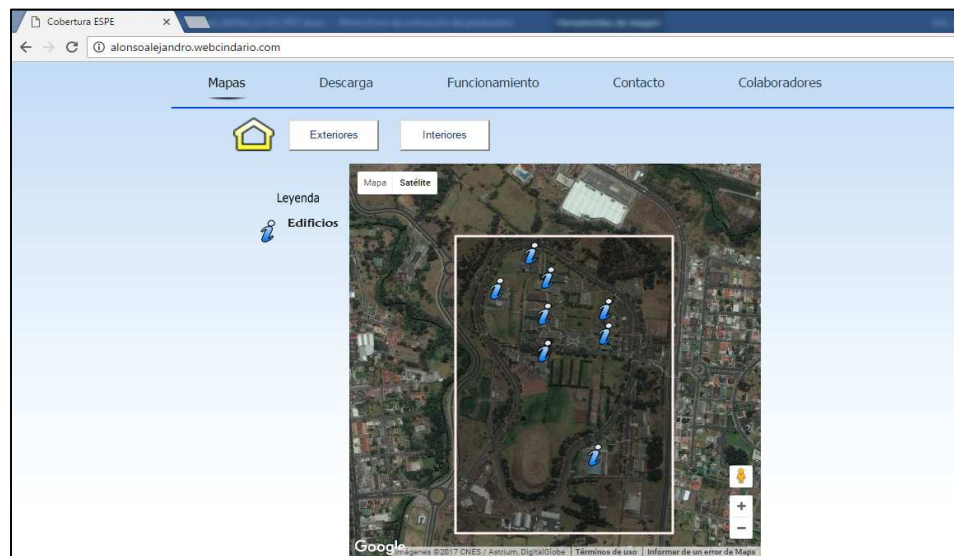
Para realizar la prueba de la aplicación, es necesario tener instalada la misma en el dispositivo móvil y a continuación empezar a utilizar todas las funcionalidades disponibles.

Para visualizar los datos, también se puede hacer uso de un navegador web para ir observando cómo se siguen obteniendo datos con el transcurrir de los días.

Para este ejemplo se toma como referencia la red de exteriores Zona Libre y se va graficando la cobertura en distintas fechas en el campus universitario.

Como se mencionó previamente, para observar el mapa de cobertura a través del navegador web es necesario ingresar a la dirección en Internet:

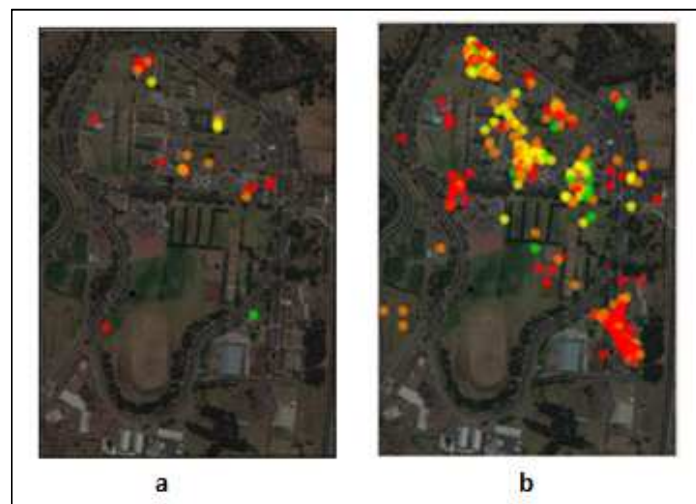
**[alonsoalejandro.webcindario.com](http://alonsoalejandro.webcindario.com)**



**Figura 57: Página de inicio de la Aplicación Web**

Una vez ingresada esta dirección en el navegador, se obtiene la página de inicio que se observa en la Figura 57:

Ahora se selecciona la opción exteriores y se elige la Red Zona Libre para conocer la cobertura inalámbrica de esta red:



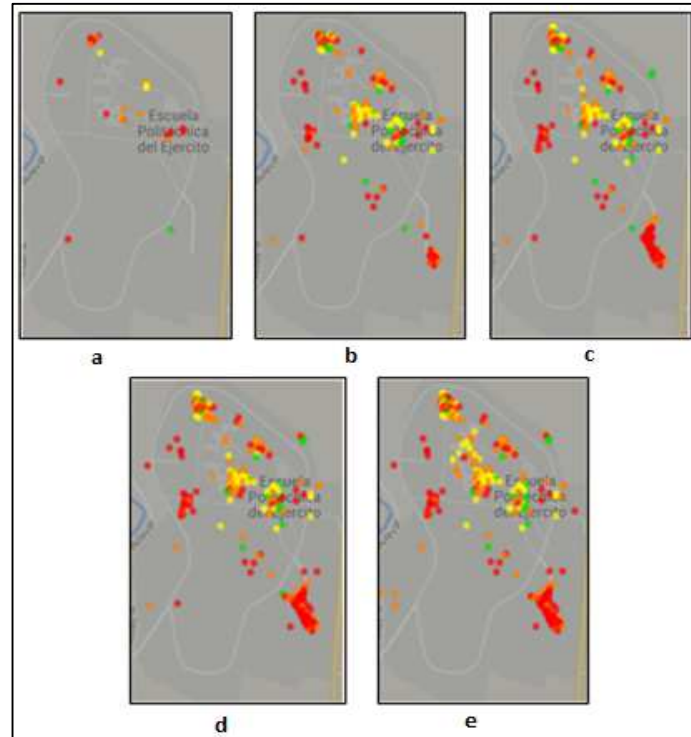
**Figura 58: Cobertura de la Red Zona Libre a) 13-12-2016 y b) 12-02-2017**

En la Figura 58 se puede ver la cobertura de la Red Zona Libre en distintas fechas utilizando un Mapa Satelital.

También es posible graficar la evolución en el envío de datos desde el mes de diciembre del 2016 hasta febrero del 2017, la obtención de datos se realizó progresivamente de acuerdo a las fechas:

- a) 13-12-2016.
- b) 22-12-2016.
- c) 09-01-2017.
- d) 14-01-2017.
- e) 12-02-2017.

En este caso, para la visualización, se utiliza un mapa de carreteras (Roadmap) y los resultados se pueden ver en la Figura 59.



**Figura 59: Cobertura red Zona Libre a) 13-12-2016, b) 22-12-2016  
c) 09-01-2017, d)14-01-2017, e)12-02-2017**

Para observar los resultados de cobertura en interiores se elige la opción “Interiores” y después el lugar y la red como se muestra en la Figura 60:



**Figura 60: Elementos para visualizar cobertura en interiores**

También es posible verificar la evolución del envío de datos en interiores a través de la Figura 61:



**Figura 61: Cobertura red Espej piso 2 a) 13-12-2016 y b) 12-02-2017**

El funcionamiento desde el lado de la aplicación móvil puede observarse en las Figuras 62 y 63:



**Figura 62: Cobertura red Zona Libre en dispositivo móvil (Exteriores)**



**Figura 63: Cobertura red Espe piso 2 en dispositivo móvil (Interiores)**

Es necesario mencionar que no existe ninguna aplicación gratuita para verificar el estado de las redes inalámbricas y por este motivo no se puede realizar una comparación en cuanto a las características, funcionalidades para tener una referencia de comparación.

## CAPÍTULO 6

### CONCLUSIONES Y LINEAS FUTURAS DE INVESTIGACIÓN

#### 6.1 Conclusiones

A través de una colaboración masiva (*Crowdsensing*), se pudo obtener e identificar el área de cobertura de las redes inalámbricas en la Universidad de las Fuerzas Armadas – ESPE utilizando como herramienta de adquisición de datos las tarjetas inalámbricas de los dispositivos móviles de los participantes (estudiantes en el campus).

Una aplicación *Crowdsensing* tiene una arquitectura bien definida conformada por participantes, solicitantes y plataforma web, los cuales interactúan entre sí para solucionar determinados problemas y si alguno de ellos llega a fallar, todo el sistema actuará de la misma manera.

Para el desarrollo del programa que permite enviar y visualizar datos en los dispositivos móviles, se optó por utilizar el Sistema Operativo Android puesto que es el que domina el mercado, de esta manera se pudo llegar a un número mayor de usuarios para que utilicen la aplicación.

El uso de hilos y tareas asíncronas en el desarrollo de la aplicación móvil basada en Android, es de vital importancia puesto que permiten realizar otras tareas en segundo plano evitando que la aplicación llegue a colgarse o funcione con lentitud.

Para visualizar correctamente el contenido en la aplicación, se hizo uso de los *fragments* que permiten realizar una aplicación multipanel facilitando la interacción con la interfaz gráfica para hacer uso de todas las funcionalidades.

Dentro de la *activity* de la aplicación (MainActivity), es posible utilizar *fragments* (Exteriores, Edificio Central, Biblioteca), los mismos que interactúan entre sí permitiendo combinar múltiples *fragments* en una sola actividad para crear una interfaz de usuario multipanel.

Mediante el uso de algunos parámetros como: Latitud, Longitud, RSSI, Identificador del Dispositivo, Fecha, es posible crear una aplicación que permite visualizar y conocer el estado de las redes inalámbricas Wi-Fi en tiempo real.

El parámetro RSSI a pesar de ser variante, es fundamental para el desarrollo de la aplicación puesto que permite medir la energía de radio recibida en cada uno de los dispositivos móviles y de esta manera permite graficar los íconos en el mapa según la intensidad de señal recibida.

Para el desarrollo del servicio web se utilizó la arquitectura cliente – servidor que permite comunicar todas las tecnologías del lado del cliente (HTML, CSS, JavaScript) con las del lado del servidor (PHP, MySQL) a través del protocolo HTTP para cumplir con los requerimientos del sistema.

Al momento de obtener la ubicación en los dispositivos móviles, es importante seleccionar un método adecuado que permita obtener datos reales y al mismo tiempo optimice la energía de la batería del dispositivo móvil, es por este motivo que se trabajó utilizando GPS – Asistido.

Con el desarrollo de la aplicación, es posible identificar las áreas de cobertura inalámbrica así como también los lugares donde existe una mayor concentración de usuarios.

Al aumentar el número de participantes que utilicen la aplicación existirá un mayor número de datos favoreciendo una mejor visualización de las áreas de la Universidad de las Fuerzas Armadas – Espe.

## 6.2 Recomendaciones

Para visualizar los resultados, se recomienda utilizar el navegador Google Chrome puesto que cuenta con todas las funcionalidades para ver los resultados vía Web.

Tanto para el desarrollo móvil como el desarrollo web, se recomienda utilizar algoritmos simples que no comprometan la velocidad, funcionalidad, usabilidad de las aplicaciones y que al mismo tiempo provean de una interfaz amigable para el usuario.

Para poder visualizar los mapas de cobertura en la Universidad de las Fuerzas Armadas -ESPE, es necesario que la aplicación *Crowdsensing* cuente con un gran número de participantes, de esta manera se asegura que exista la cantidad de datos necesarios para mostrar la cobertura en la mayor cantidad de áreas posibles en el campus.

Para desarrollar una aplicación académica es necesario utilizar en lo posible elementos gratuitos y de software libre que cumplan con todos los requerimientos del usuario.

## 6.3 Trabajos Futuros

Como trabajo futuro se puede desarrollar la aplicación móvil para otro Sistema Operativo como por ejemplo IOS puesto que este factor fue un limitante para los participantes que no disponían del Sistema Operativo Android, así mismo se requiere realizar la comparación del desempeño de la aplicación con otras existentes.

Se plantea la posibilidad de ampliar la zona de cobertura a mostrar, que en este caso no esté limitado únicamente a la Universidad de las Fuerzas Armadas – ESPE sino a un sector mayor.

Es posible utilizar los datos recogidos para crear nuevos servicios basados en localización y geocercas como aplicaciones de seguridad, marketing, notificaciones de texto, etc.



## BIBLIOGRAFÍA

- Aguilar, E., & Dávila, D. (2013). *Análisis, diseño e implementación de la aplicación web para el manejo del distributivo de la facultad de ingeniería*. Obtenido de <http://dspace.ucuenca.edu.ec/bitstream/123456789/4303/1/tesis.pdf>
- Alegsa, L. (02 de 08 de 2016). *Definición de Back-end*. Recuperado el 12 de 2016, de Alegsa.com.ar: <http://www.alegsa.com.ar/Dic/back-end.php>
- alliance, o. h. (5 de 11 de 2007). *Open Handset Alliance*. Obtenido de Industry Leaders Announce Open Platform for Mobile Devices: [http://www.openhandsetalliance.com/press\\_110507.html](http://www.openhandsetalliance.com/press_110507.html)
- Álvarez, E. (2015). *Publicación de cartografía en Internet*. Oviedo.
- Álvarez, S. (12 de Septiembre de 2007). *Modelo entidad-relación*. Recuperado el 12 de 2016, de desarrolloweb.com: <https://desarrolloweb.com/articulos/modelo-entidad-relacion.html>
- Angelopoulos, C., Nikoletseas, S., Raptis, T., & Rolim, J. (November de 2014). Characteristic utilities, join policies and efficient incentives in Mobile Crowdsensing Systems. *IFIP Wireless Days (WD)*. *IEEE*, 1-6.
- Antunes, A. (19 de October de 2016). *10 LIBRERÍAS QUE DEBES CONOCER SI ERES DESARROLLADOR ANDROID*. Obtenido de Bemobile: <http://bemobile.es/blog/2016/10/10-librerias-que-debes-conocer-si-eres-desarrollador-android/>
- Arreondo, J. (11 de 05 de 2015). *¿Para que sirve cada uno de los sensores que tiene tu móvil?* Obtenido de SMART LIFE: [http://cincodias.com/cincodias/2015/05/11/lifestyle/1431341623\\_109997](http://cincodias.com/cincodias/2015/05/11/lifestyle/1431341623_109997)
- Bardwell, J. (2002). *Converting Signal Strength Percentage to dBm Values*.
- Bustos Jiménez, S. (2015). *Crowdsensing en Ciudades Inteligentes*.
- Cabrera, J. (2005). *Diseño e Implementación de un Sistema en Web de Biblioteca Digital de Documentos de Literatura Científica*. Obtenido de <http://www.utm.mx/~dtorres/doc/BibliotecaDigitaldeLiteraturaCientifica.pdf>
- Calzado, Á. (2017). *Sensores de dispositivos móviles. La extensión de los sentidos*. Recuperado el 12 de 2016, de Sintaxisweb: <https://www.sintaxisweb.es/sensores-dispositivos-moviles/>

- CampusMVP. (08 de 2015). *Desarrollador web: Front-end, back-end y full stack. ¿Quién es quién?* Recuperado el 12 de 2016, de CampusMVP.com: <http://www.campusmvp.es/recursos/post/Desarrollador-web-Front-end-back-end-y-full-stack-Quien-es-quien.aspx>
- Carrodegua, N. (2016). *Tipos de redes y estándares Wi-Fi, sus diferencias.* Recuperado el 12 de 2016, de norfipc.com: <https://norfipc.com/redes/tipos-redes-estandares-wi-fi-diferencias.php>
- Centelles, S. (2015). *Wireframes: Que son y cómo crearlos.* Recuperado el 01 de 2017, de webdesdecero: <http://webdesdecero.com/wireframes-que-son-y-como-crearlos/>
- Chhabra, P. (2013). *Android Fundamentals: Building Blocks, Intents and User Interface.* Recuperado el 12 de 2016, de optimusinformation: <http://www.optimusinfo.com/android-fundamentals-building-blocks-intents-and-user-interface/>
- Chuquitarcó, M., Naranjo, S., & Ronny, J. (09 de 2012). *Diseño e instalación de un sistema de rastreo satelital mediante GPS y GPRS al vehículo Chevrolet-Aveo de la Escuela de Conducción de ESPE -Latacunga.* Obtenido de Repositorio Dspace: <http://repositorio.espe.edu.ec/handle/21000/5879>
- Cobo, Á., Gómez, P., Pérez, D., & Rocha, R. (2005). *PHP y MySQL Tecnologías para el desarrollo de.* España: Díaz de Santos.
- Conde, R. (2013). *Parámetros reales para identificar gamas de los celulares.* Recuperado el 12 de 2016, de <http://celulares.about.com/od/Smartphones/ss/Parametros-Reales-Para-Identificar-Gamas-De-Los-Celulares.htm>
- Crowdsourc. (2016). *Google Crowdsourc. Google Play.* Recuperado el 12 de 2016
- Cuyo, R. (10 de 2015). *Servidor Web.* Obtenido de <http://camanaa.blogspot.com/2015/10/servidor-web-un-servidorweb-servidor.html?view=flipcard>.
- Developers, A. (2016). *Android Developers. Google, [En línea].* Obtenido de <https://developer.android.com/index.html>
- Developers, G. (2016). *Google Developers.* Obtenido de Google. [En línea]: <https://developers.google.com/maps/?hl=es-419>.
- DigitalGuide. (2017). *Las bases de datos y la importancia de sus copias de seguridad.* Recuperado el 01 de 2017, de <https://www.1and1.es/digitalguide/servidores/seguridad/>

- EmprendedorInnovador. (16 de June de 2005). *¿Cuál es la diferencia entre servidor web y alojamiento web?* Obtenido de Cómo iniciar un negocio en línea: <https://www.emprendedorinnovador.com/cual-es-la-diferencia-entre-servidor-web-y-alojamiento-web/>
- Flores Villafuerte, M., & Narváez, A. M. (2011). *Estudio de la Calidad de Servicio en la Coexistencia entre Nodos Wlan 802.11 b, g y 802.11 e*. Obtenido de <http://dspace.esPOCH.edu.ec/handle/123456789/616#sthash.IGES2hAS.dpuf>
- FQ, I. (12 de 02 de 2016). *¿Qué es el valor RSSI en un sistema RFID?* Obtenido de <http://www.fqingenieria.com/es/conocimiento/que-es-el-valor-rssi-en-un-sistema-rfid-95>
- GCF. (2016). *What is a mobile device?* Recuperado el 12 de 2016, de GCF LearnFree.org: <http://www.gcflernfree.org/computerbasics/mobile-devices/1/>
- Guo, B., Yu, Z., Zhou, X., & Zhang, D. (2014). From participatory sensing to Mobile Crowd Sensing. *From participatory sensing to mobile crowd sensing. In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference IEEE*, 593-598.
- Herring, T. A. (February de 1996). The global positioning system. (cover story). *Scientific American*, 274, 44. Obtenido de <http://ludwig.missouri.edu/2840/GPS.pdf>
- Invarato, R. (04 de 11 de 2012). *Activity – entender y usar una Actividad*. Recuperado el 11 de 2016, de jarroba.com: <https://jarroba.com/activity-entender-y-usar-una-actividad/>
- James, R. (2014). *AsyncTask: Tareas Asíncronas en Android*. Recuperado el 11 de 2016, de Hermosa Programación: <http://www.hermosaprogramacion.com/2014/12/android-async-task-hilos/>
- Jones, H. (2010). *10 Excellent Tools for Creating Web Design Wireframes*. Recuperado el 01 de 2017, de webdesignledger.com: <https://webdesignledger.com/10-excellent-tools-for-creating-web-design-wireframes/>
- Juliá, S. (s.f.). *¿Redes inalámbricas o redes por cable?* Recuperado el 01 de 2017, de Gadae Netweb: <http://www.gadae.com/blog/redes-inalambricas-o-redes-por-cable/>
- Kioskea. (06 de 2014). *Redes inalámbricas*. Recuperado el 12 de 2016, de <http://es.ccm.net/contents/818-redes-inalambricas>

- Lamarca, M. (12 de 2013). *HTML*. Recuperado el 12 de 2016, de [www.hipertexto.info](http://www.hipertexto.info): <http://www.hipertexto.info/documentos/html.htm>
- Lancetalent. (08 de 2014). *Las Mejores Herramientas Para Hacer El Prototipo De Tu App*. Obtenido de [lancetalent.com](https://www.lancetalent.com): <https://www.lancetalent.com/blog/mejores-herramientas-prototipo-app/>
- Lara, T. (2014). *Crowdsourcing. Cultura compartida*. Anuario AC/E .
- Luján-Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante: Club Universitario.
- Mahmud, F., & Aris, H. (2015). State of mobile crowdsourcing applications: A review. In *Software Engineering and Computer Systems (ICSECS). 4th International Conference. IEEE.*, 27-32.
- Medialex. (04 de 2016). *LA IMPORTANCIA DE GOOGLE MAPS*. Obtenido de Mediaelx web design: <http://www.mediaelx.net/blog/201/la-importancia-de-google-maps/>
- miarroba. (2016). *Hosting*. Obtenido de <http://hosting.miarroba.es/features.php>
- Navarra, U. (2013). *Arquitectura de Redes, Sistemas y Servicios*. Obtenido de Redes inalámbricas 802.11 y acceso al medio: [https://www.tlm.unavarra.es/pluginfile.php/8477/mod\\_resource/content/0/slidesPDF/31y32-CSMA-CA.pdf](https://www.tlm.unavarra.es/pluginfile.php/8477/mod_resource/content/0/slidesPDF/31y32-CSMA-CA.pdf)
- Navarrete, T. (2006). *El lenguaje JavaScript*.
- NinjaMock. (2017). *NinjaMock Slice your work in half*.
- Opensignal. (2016). *About OpenSignal: Mission & Vision*. Recuperado el 12 de 2016, de <https://opensignal.com/>.
- Oscar. (2008). *Lenguaje batch tutorial y comandos básicos .bat*. Recuperado el 01 de 2017, de TodoHacker: <https://todohacker.com/tutoriales/lenguaje-batch>
- Ovalle, J. P. (06 de 05 de 2013). *Conexion a servidor desde android*. Recuperado el 07 de 2016, de <http://es.slideshare.net/JosePabloOvalle/conexion-a-servidor-desde-android>
- Pérez, J., & Merino, M. (2012). *Definición de URL*. Obtenido de <http://definicion.de/url/>

- Potorti, F., Corucci, A., Paolo, N., Furfari, F., Barsocchi, P., & Buffi, A. (s.f.). Accuracy limits of in-room localisation using RSSI. *2009 IEEE Antennas and Propagation Society International Symposium*.
- Pozo, J. D. (2011). *Introducción a los dispositivos móviles*.
- Remuñán, M. (01 de 2010). *WiFi: historia, evolución, aplicaciones, desarrollos*. Recuperado el 12 de 2016, de WifiClub: <http://ww1.wificlub.org/featured/wifi-historia-evolucion-aplicaciones-desarrollos/>
- Revelo, J. (23 de 12 de 2014). *AsyncTask: Tareas Asíncronas en Android*. Obtenido de Hermosa Programación: <http://www.hermosaprogramacion.com/2014/12/android-async-task-hilos/>
- Rodríguez, A. (04 de 2011). *Google Maps API V3 introducción y primeros pasos*. Recuperado el 12 de 2016, de Maestros del Web: <http://www.maestrosdelweb.com/google-maps-api-v3-introduccion-y-primeros-pasos/>
- Rodríguez, J. (2014). *Activity y Fragments*. Recuperado el 11 de 2016, de Academia Android: <http://academiaandroid.com/activity-y-fragments/>
- Schenk, E., & Guittard, C. (7 de December de 2009). Crowdsourcing. *In Workshop on Open Source Innovation, Strasbourg*, 72.
- Sgoliver.net. (07 de 2012). *Tareas en segundo plano en Android (I): Thread y AsyncTask*. Recuperado el 11 de 2016, de <http://www.sgoliver.net/blog/tareas-en-segundo-plano-en-android-i-thread-y-async-task/>
- Shailendra123. (08 de 03 de 2013). *WifiDemo*. Obtenido de GitHub,Inc: <https://github.com/shailendra123>
- SmartFix. (s.f.). *¿COMO CLASIFICAR UN SMARTPHONE A LA HORA DE HACER COMPRAS?* Recuperado el 12 de 2016, de <https://smartfixcr.wordpress.com/2015/05/17/que-significa-que-un-celular-sea-gama-alta-media-o-baja>
- StatCounter. (07 de 2016). *Top 8 Mobile & Tablet Operating Systems from Aug 2012 to July 2016*. Recuperado el 11 de 2016, de StatCounter Global Stats - Browser, OS, Search Engine including Mobile Usage Share: <http://gs.statcounter.com/#mobile+tablet-os-ww-monthly-201208-201607-bar>.

- Sun, J., & Ma, H. (2014). A behavior-based incentive mechanism for crowd sensing with budget constraints. *Communications (ICC), IEEE International Conference on*.
- Tabernilla, L., & Díaz, A. (2008). *Sistema de localización en interiores*.
- Tanas, C., & Herrera, J. (July de 2012). Users as Smart Sensors: A Mobile Platform for Sensing Public Transport Incidents. *Citizen in Sensor Networks*, 7685, 81-93. doi:DOI: 10.1007/978-3-642-36074-9\_8
- Tomás, J. (11 de 2015). *Arquitectura de Android*. Obtenido de iploma de Especialización en desarrollo de aplicaciones para Android: <http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/99-arquitectura-de-android>
- Van Diggelen, F. S. (2009). *A-GPS : assisted GPS, GNSS, and SBAS*. Boston: Artech House.
- Vicente. (2013). *Cómo conseguir un buen enlace WiFi*. Obtenido de Redes Zone: <https://www.redeszone.net/redes/conseguir-buen-enlace-wifi/>
- Viteri, S. (2017). *La importancia del backup*. Recuperado el 01 de 2017, de Blog de Sergio Viteri: <https://sergioviteri.com/2009/09/29/la-importancia-del-backup/>
- Vukovic, M., Laredo, J., & Rajagopal, S. (2010). Challenges and Experiences in Deploying Enterprise Crowdsourcing Service. *Web Engineering. ICWE 2010. Lecture Notes in Computer Science*, 6189. doi:10.1007/978-3-642-13911-6\_31
- W3C. (s.f.). *Consortio World Wide Web*. Recuperado el 12 de 2016, de <https://www.w3.org/>
- W3C. (s.f.). *Servicios Web*. Recuperado el 12 de 2016, de Guías Breves de Tecnologías W3C: <http://www.w3c.es/Divulgacion/GuiasBreves/>
- Waze. (2016). <https://www.waze.com/es-419>. Recuperado el 28 de November de 2016
- WeatheSignal. (2016). *WeatherSignal - clima sensors. Google Play*. Recuperado el 12 de 2016
- Whitla, P. (March de 2009). Crowdsourcing and Its Application in Marketing Activities. *Contemporary Management Research*, 5(1), 15-28.