



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



# **CrowdPi: Plataforma basada en dispositivos Raspberry Pi para la implementación de aplicaciones de crowdsourcing**

**TRABAJO FIN DE MÁSTER**

**Autor: Danilo Alfonso Vargas Valenciano**

**Director: Dr. Pietro Manzoni**

**Diciembre de 2015**

**Departamento de Informática de Sistemas y Computadores**

**Grupo de Redes de Computadores**

# Agradecimientos

---

A la Universitat Politècnica de València por permitirme cursar este Máster y en especial al Grupo de Redes y Computadores, por todo el apoyo brindado y la experiencia adquirida. Además deseo agradecer enormemente a mi tutor, Pietro Manzoni, por su guía en la elaboración de esta investigación.

Al Rector de la Universidad Técnica Nacional, Lic. Marcelo Pietro Jiménez, al Decano de la Sede de San Carlos, Msc. Luis Restrepo Gutiérrez y demás autoridades universitarias que me apoyaron en este proyecto educativo.

Dar gracias a mi familia, y amigos, los cuales siempre han estado pendientes de mi avance y sin sus consejos y compañía no hubiera podido concluir este trabajo satisfactoriamente.

## Resumen

---

Algunas técnicas de crowdsourcing se basan en ofrecer servicios utilizando la información del público; existen varios ejemplos de este tipo de aplicaciones como Tripadvisor, Foursquare. Todas estas aplicaciones presuponen que el usuario siempre esté conectado a internet vía WIFI o 3G|4G. Para facilitar el acceso a dicha información y limitar la carga introducida en la red resulta más eficiente posicionar la información físicamente en el sitio de interés y ofrecerla directamente a los usuarios que se encuentren en esa zona. Para favorecer la difusión de este tipo de aplicaciones, se quiere diseñar CrowdPi, una plataforma para aplicaciones de Crowdsourcing basadas en dispositivos Raspberry Pi. Esta plataforma tendrá que poder almacenar la información a distribuir, y se encargará de recibir peticiones y ofrecer información a usuarios cercanos. La estructura de las peticiones y de las respuestas tendrá que seguir el modelo de las redes ICN.

Esta plataforma permite la divulgación de promociones comerciales de manera efectiva y rápida a dispositivos móviles pertenecientes a la tecnología Android que tengan instalado la aplicación CrowdPi. A su vez los usuarios de estos dispositivos inteligentes pueden compartir sus comentarios acerca de los comercios de manera pública o a grupos de usuarios. Además al utilizar la aplicación los usuarios consumen poco de su batería.

# Abstract

---

Some crowdsourcing techniques are based on providing services using public information; there are several examples of such applications as TripAdvisor, Foursquare. All these applications must assume that the user is always connected to the internet via WIFI or 3G | 4G. To facilitate access to such information and limit the network throughput is more efficient positioning information physically at the site of interest and offer it directly to users who are in that area. To promote the dissemination of these applications, CrowdPi app is a perfect solution, is a platform for applications based Crowdsourcing Raspberry Pi devices. This platform will be able to store the information, distributed and be responsible for receiving requests and provide information to nearby users. The structure of the requests and responses will have to follow the model of ICN networks.

This platform allows publicize promotions effectively and quickly to mobile Android devices who have installed the application CrowdPi. In turn, the users of these smart devices can share feedback to everyone or to a private group of friends. In addition, using this application will consume a little percent of the cellphone battery.

## TABLA DE CONTENIDO

### Contenido

|  |           |
|--|-----------|
| <b>CAPÍTULO I: INTRODUCCIÓN.....</b>   | <b>9</b>  |
| 1.1 Objetivo general y objetivos específicos .....                                   | 11        |
| 1.2 Motivación.....  | 12        |
| 1.3 Estructura del documento .....   | 13        |
| <b>CAPÍTULO II: ANTECEDENTES.....</b>  | <b>14</b> |
| 2.1 Estudios realizados .....  | 14        |
| 2.2 Aplicaciones móviles utilizadas en crowdsourcing: .....                          | 16        |
| 2.3 Aplicaciones móviles similares: .....  | 18        |
| 2.4 Hardware utilizado en el experimento .....                                       | 20        |
| 2.5 Dispositivo de comunicación entre dispositivos utilizado en el experimento ..... | 25        |
| 2.6 Software utilizado en el experimento .....                                       | 26        |
| <b>CAPÍTULO III: ANÁLISIS DEL SISTEMA.....</b>                                       | <b>28</b> |
| 3.1 Descripción del sistema .....  | 30        |
| 3.2 Requisitos del Sistema.....  | 31        |
| 3.2.1 Requisitos Funcionales .....   | 31        |
| 3.2.1.1. Aplicación móvil .....  | 31        |
| 3.2.1.2. Aplicación Web .....  | 32        |
| 3.2.1.3. Punto de Acceso .....   | 32        |
| 3.2.2 Requisitos no Funcionales.....   | 32        |
| 3.2.2.1 Aplicación móvil.....  | 32        |
| 3.2.2.2 Aplicación Web.....  | 33        |
| 3.2.2.3 Punto de Acceso.....   | 33        |
| 3.3 Diagramas y Casos de Uso .....   | 33        |
| 3.3.1 Caso de uso de la Aplicación Móvil Nativa .....                                | 34        |
| 3.3.2 Caso de uso del Sistema Web .....  | 34        |
| 3.3.3 Caso de uso del Punto de Acceso.....   | 35        |
| <b>CAPÍTULO IV: DISEÑO DEL SISTEMA.....</b>  | <b>36</b> |
| 4.1 Componentes utilizados .....   | 36        |

|  |           |
|--|-----------|
| 4.2 Aplicación Web CrowdPi.....                        | 40        |
| 4.2.1 Arquitectura.....                                | 40        |
| 4.2.2 Interfaz Sistema Web .....                       | 43        |
| 4.3 Aplicación Móvil CrowdPi.....                      | 46        |
| 4.3.1 Arquitectura.....                                | 47        |
| 4.3.2 Interfaz Sistema Móvil .....                     | 50        |
| 4.4 Store-Scripts .....                                | 53        |
| <b>CAPÍTULO V: PRUEBAS .....</b>                       | <b>54</b> |
| 5. 1. Descripción de las pruebas .....                 | 54        |
| <b>CAPÍTULO VI: CONCLUSIONES Y TRABAJO FUTURO.....</b> | <b>65</b> |
| Conclusiones .....                                     | 65        |
| Trabajo futuro.....                                    | 65        |
| <b>BIBLIOGRAFÍA .....</b>                              | <b>66</b> |
| <b>ANEXO A.....</b>                                    | <b>68</b> |

## Contenido de Figuras

|   |    |
|---|----|
| Figura 2. 1. OpenStreetMap permite generar mapas vehiculares.....                   | 16 |
| Figura 2. 2. Finally genera un mapa de cobertura de redes WiFi.....                 | 17 |
| Figura 2. 3. Weddar genera un reporte sobre el estado del tiempo. ....              | 17 |
| Figura 2. 4. Duolingo permite aprender idiomas.....                                 | 18 |
| Figura 2. 5. Herramienta MeshMe para mensajería. ....                               | 19 |
| Figura 2. 6. Herramienta FireChat para mensajería.....                              | 19 |
| Figura 2. 7. Herramienta Liberouter utilizada para la mensajería. ....              | 20 |
| Figura 2. 8. Diferentes modelos de Raspberry Pi 2.....                              | 21 |
| Figura 2. 9. Modelo de Capas del Sistema Operativo Android.....                     | 23 |
| Figura 2. 10. Logo de WiFi. ....  | 25 |
| Figura 3. 1. Fases que componen la metodología Cascada. ....                        | 28 |
| Figura 3. 2. Diagrama de casos de uso de la aplicación móvil nativa. ....           | 34 |
| Figura 3. 3. Casos de uso utilizados en la Aplicación Web.....                      | 34 |
| Figura 3. 4. Casos de uso utilizados en el Punto de Acceso.....                     | 35 |
| Figura 4. 1. Esquema de componentes de hardware utilizados. ....                    | 36 |
| Figura 4. 2. Transmisión de datos entre Cliente Android y el Servidor Web. ....     | 38 |
| Figura 4. 3. Ejemplo del formato de mensaje JSON. ....                              | 38 |
| Figura 4. 4. Dispositivo WiFi Realtek RTL8188CUS USB.....                           | 39 |
| Figura 4. 5. Tecnologías software utilizadas en el sistema propuesto.....           | 40 |
| Figura 4. 6. Diferentes opciones del menú de la aplicación web. ....                | 40 |
| Figura 4. 7. Diagrama de clases de la aplicación web.....                           | 41 |
| Figura 4. 8. Función Backup de información del local. ....                          | 42 |
| Figura 4. 9. Modelo Entidad-Relación de la aplicación web.....                      | 43 |
| Figura 4. 10. Pantalla de Promociones del sistema web.....                          | 44 |
| Figura 4. 11. Pantalla para guardar nuevas promociones.....                         | 45 |
| Figura 4. 12. Listado de Comentarios realizados por clientes.....                   | 45 |
| Figura 4. 13. Datos del local y respaldo de la aplicación web. ....                 | 46 |
| Figura 4. 14. Diagrama de clases del sistema móvil. ....                            | 47 |
| Figura 4. 15. Modelo Entidad Relación de la base de datos del sistema móvil. ....   | 49 |
| Figura 4. 16. Búsqueda de SSID Mora_AP.....   | 50 |
| Figura 4. 17. Promociones listadas en la aplicación CrowdPi.....                    | 50 |
| Figura 4. 18. Información del local comercial en la aplicación CrowdPi. ....        | 51 |
| Figura 4. 19. Generar un comentario en la aplicación CrowdPi.....                   | 52 |
| Figura 4. 20. Crear Grupos de usuarios en la aplicación CrowdPi.....                | 53 |
| Figura 4. 21. Ingresar a un grupo de Usuarios en la aplicación CrowdPi. ....        | 53 |
| Figura 5. 1. Tasa de Transferencia usando el Raspberry Pi como Punto de Acceso..... | 56 |
| Figura 5. 2. Ancho de banda con varias configuraciones del Punto de Acceso.....     | 57 |
| Figura 5. 3. Jitter obtenido en la red utilizando varias configuraciones.....       | 58 |
| Figura 5. 4. Mapa de Intensidad de señal del punto de acceso Mora_AP.....           | 59 |
| Figura 5. 5. Pruebas realizadas al punto de acceso con aplicación Ekahau. ....      | 59 |
| Figura 5. 6. Retardo de Mensajes enviados al Servidor. ....                         | 61 |
| Figura 5. 7. Tasa de Refrescamiento de la WiFi Mora_AP.....                         | 62 |
| Figura 5. 8. Mili-watts hora usados por la aplicación. ....                         | 63 |
| Figura 5. 9. Duración de horas de la aplicación en las diferentes Etapas.....       | 64 |

## Contenido de Tablas

|  |    |
|--|----|
| Tabla 1. Relación casos de uso de la Aplicación Móvil - Requisitos Funcionales. .... | 34 |
| Tabla 2. Relación casos de uso de la Aplicación Web - Requisitos Funcionales.....    | 35 |
| Tabla 3. Relación casos de uso del Punto de Acceso – Requisitos funcionales.....     | 35 |
| Tabla 4. Tamaño de los datos utilizada para medir el rendimiento de la red. ....     | 55 |
| Tabla 5. Comandos utilizados para evaluar el desempeño de la Red.....                | 55 |
| Tabla 6. Configuración utilizada en los experimentos de Ancho de Banda.....          | 56 |
| Tabla 7. Configuración utilizada en los experimentos de Jitter. ....                 | 58 |
| Tabla 8. Resultados obtenidos con la aplicación Ekahau.....                          | 60 |
| Tabla 9. Consumo energético clasificado por etapas. ....                             | 63 |
| Tabla 10. Consumo energético en horas.....   | 64 |



## Capítulo I: Introducción

Con la entrada en escena de las redes sociales como Twitter, facebook y las aplicaciones de crowdsourcing como Foursquare o TripAdvisor entre otras, los clientes potenciales han cambiado la manera en la que compraba algún servicio o producto. Cada vez son más las personas que consultan información o prestan atención a comentarios otorgados por clientes anteriores antes de decidirse a adquirir algún servicio. Todas estas opiniones ayudan al proceso de mejoramiento de los productos que ofrece cierto comercio.

En relación con el tema discutido anteriormente, se estima que la población genere críticas y comentarios que puedan validar el trabajo y los productos que se ofrecen: (youngmarketing.co, 2014). Al empoderar al usuario y brindarles las herramientas para que aporten a la construcción de su entorno, es una forma de aumentar el sentido de pertinencia y de generar lugares incluyentes, amigables e inteligentes. Lo anterior implica que se genera fidelidad por parte de los clientes o mejoras sustanciales a los procesos que desarrolla la empresa, al ser validados constantemente por sus usuarios.

Por lo anteriormente citado, se considera sumamente importante que las empresas utilicen el crowdsourcing y las herramientas tecnológicas para conectarse con sus clientes y generar discusiones o comentarios por parte de estos.

Se considera crowdsourcing según ( Cambridge University Press, 2015) como el acto de dar tareas a un grupo grande de personas o para el público en general, por ejemplo, al pedir ayuda en Internet, en lugar de haber hecho las tareas dentro de la empresa por empleados. De esta manera, los locales comerciales pueden ofrecer diferentes promociones a sus posibles clientes que se encuentren cerca de su establecimiento y que ellos generen retroalimentación a manera de comentarios, lo que brinda la posibilidad de conocer las necesidades reales de los clientes o las deficiencias que tienen los productos que se proporcionan al público en general. Esto permite que los problemas puedan ser resueltos rápidamente. Además otro punto a ser tomado en consideración es el coste de equipos que implica el ofrecer estos servicios, los cuales son muy asequibles por su bajo coste, permitiendo a cualquier local sin importar su tamaño el poder competir en un mundo globalizado.

En relación con lo anterior, existen soluciones tecnológicas que ofrecen información acerca de comentarios de locales comerciales, pero estas soluciones utilizan las redes 3G o 4G para su funcionamiento, lo que hace necesario tener internet para su funcionamiento. En cambio la

plataforma CrowdPi, facilita el acceso a la información de manera local, sin generar más tráfico a la red o ser víctimas de un ataque informático, todo esto utilizando un dispositivo de bajo coste para la retroalimentación entre clientes y el local comercial.

Basado en el planteamiento anterior, la propuesta por desarrollar en este Trabajo Fin de Máster es el desarrollo de una plataforma de comunicación que almacene y comunique anuncios a los clientes utilizando el WiFi como medio de transporte. Los clientes deben tener instalado una aplicación nativa en Android para poder leer las promociones y generar comentarios entre diversos grupos o de manera pública acerca de los productos o promociones adquiridas.

## 1.1 Objetivo general y objetivos específicos

### Objetivo General

El objetivo general es desarrollar una plataforma para aplicaciones Crowdsourcing (CrowdPi) que facilite el acceso a los datos de manera local, sin generar más tráfico a la red. Utilizando dispositivos Raspberry Pi y la tecnología Android.

Para alcanzar el objetivo general se definen los siguientes objetivos específicos:

### Objetivos Específicos:

1. Definir las diferentes tecnologías por utilizar para la elaboración del proyecto.
2. Construir una aplicación web que ingrese los datos del local comercial y las diferentes promociones que desean comunicar.
3. Elaborar una aplicación móvil en Android que permita la divulgación de mensajes entre el Punto de Acceso del local comercial y el dispositivo móvil.
4. Diseñar varios servicios locales en el Punto de Acceso que permitan la sincronización de mensajes con los dispositivos Android conectados.

## 1.2 Motivación

Las aplicaciones para dispositivos móviles se encuentran liderando el mercado de desarrollo informático, la amplia gama de posibilidades a desarrollar hacen que la elaboración de un proyecto de este tipo sea un reto sumamente llamativo. Además, luego de cursar el plan de estudios del Máster en Ingeniería de Computadores y Redes, y especialmente las asignaturas del último periodo “Sistemas Distribuidos y Empotrados” y “Sistemas basado en Redes Móviles”, gracias a su dinamismo e innovación, ampliaron mi motivación y curiosidad por realizar una investigación en el área de los dispositivos móviles.

Por otra parte, Costa Rica se encuentra posicionada como uno de los países referentes en Latinoamérica en cuanto a la exportación de Software se trata. Por lo que al escoger este tema de fin de máster, también se previó que este era en gran medida una posibilidad comercial. Por lo que el prototipo por desarrollar podría luego optimizarse para utilizarlo en gran variedad de comercios a nivel mundial.

Por consiguiente, la posibilidad de crear una aplicación en la cual los comercios y los clientes puedan interactuar constantemente, al promocionar ofertas de productos y listar comentarios de los consumidores, todo de manera local, sin uso del internet y usando un dispositivo Raspberry Pi como punto de acceso, podría beneficiar a varios comercios de pequeña envergadura, los cuales necesitarían muy poca inversión para poder dar a conocer sus productos y obtener una retroalimentación de los clientes y de esta manera mejorar sus ingresos, consolidar su negocio o posicionar sus productos en el mercado.

Es importante recalcar que desde el inicio del máster, se dieron a conocer los diferentes Grupos de investigación para desarrollar el trabajo de Fin de Máster y luego de analizar las posibilidades, se seleccionó el Grupo de Investigación en Redes de Computadores para elaborar el proyecto final de Máster, ya que poseen gran cantidad de recursos tecnológicos y profesionales dedicados a realizar investigación, y se les podría consultar en caso de tener alguna duda o idea de trabajo.

## 1.3 Estructura del documento

### Capítulo I. INTRODUCCIÓN.

Este capítulo describe la problemática en los que se desarrolla el Trabajo Fin de Máster. Además se lista el objetivo general a realizar junto con los objetivos específicos que lo conforman. Al finalizar este capítulo, se presenta la motivación que posee el autor para realizar dicha investigación.

### Capítulo II. ANTECEDENTES.

En este capítulo se analiza como ha sido tratado el problema de estudio con anterioridad por otros entes investigativos. Además se presenta un resumen actual de las herramientas tecnológicas similares existentes en el mercado.

### Capítulo III. ANÁLISIS DEL SISTEMA.

En este capítulo se describen los requisitos funcionales y no funcionales del proyecto. Utilizando diagramas de casos de uso para la definición de los procesos realizados en el sistema.

### Capítulo IV. DISEÑO DEL SISTEMA.

En este capítulo se describe la arquitectura e interfaz de los componentes de la plataforma desarrollada. Posteriormente se demostrará la funcionalidad de la aplicación web y la aplicación móvil. Además de las interfaces y las bases de datos que utilizan

### Capítulo V. PRUEBAS.

Este capítulo trata de las diversas pruebas realizadas al sistema, con el fin de determinar los resultados obtenidos en el envío de los mensajes entre los diferentes dispositivos.

### Capítulo VI. CONCLUSIONES.

Para finalizar este documento, se describen los principales logros alcanzados y los trabajos futuros a realizar.

## Capítulo II: Antecedentes

Este capítulo centra su atención en la revisión bibliográfica de trabajos académicos y bibliográficos. Los cuales dan pie para conocer la investigación realizada por otros investigadores acerca del tema en estudio. Finalmente, se describirán algunas herramientas tecnológicas existentes en la actualidad, similares al proyecto que se va a elaborar.

### 2.1 Estudios realizados

La gran variedad de sensores que incluyen hoy en día los teléfonos inteligentes, entre los cuales tenemos: geolocalización, luz, movimiento, audio y visión, ofrecen una eficiente manera de recolectar datos y con esto establecer nuevas aplicaciones crowdsourcing en el mercado. Lo anterior llevó a los investigadores: (Crowdsourcing with smartphones, 2012) A realizar experimentos utilizando dispositivos Android con aplicaciones crowdsourcing instaladas. Según los resultados obtenidos, se espera que el crowdsourcing evolucione rápidamente y que el cómputo realizado entre varios dispositivos facilite la resolución de problemas aún más difíciles que los que existen al día de hoy.

Otro punto a tener en consideración es la movilidad de un lugar a otro, realizada por los usuarios de diversas redes, lo cual tiene un gran rol en el diseño de las redes móviles (celular, WIFI, Ad-Hoc) y el desarrollo de sus protocolos, enrutamiento y la gestión de la ubicación de las mismas.

En este sentido, en el documento: (Measurement and Modeling of User Transitioning Among Networks, 2014) los investigadores realizaron un estudio donde se mide la movilidad de los usuarios entre redes, ya que éste puede cambiar de una red inalámbrica a otra en cuestión de segundos, por lo que la movilidad es un factor por tomar en consideración a la hora de diseñar cualquier tipo de red inalámbrica, ya sea pública o privada. Esto con el fin de dar el mejor servicio y cobertura posible a los clientes de la misma.

Con respecto al tipo de tráfico IP utilizado, en el trabajo (Information-centric Networking: The evolution from circuit to packets to content, 2014) el investigador indica que el número de suscriptores móviles activos es superior al número de usuarios cableados por un factor de tres, se espera que para el 2016 apenas el 39% del tráfico IP sea de los usuarios que utilicen servicios cableados, contra un 61% de usuarios que utilizarían WiFi o servicios móviles. Por lo anterior, se espera que el soporte de los servicios móviles sea un componente crítico en el desarrollo futuro de la arquitectura de internet.

Por otra parte, en el trabajo: (Haggle: a Networking Architecture Designed Around Mobile Users, 2006). El investigador indica que muchos usuarios están forzados a estar pendientes acerca de no perder su conexión a internet, ya que muchas de sus aplicaciones solamente funcionan cuando la infraestructura de red está disponible. El simple hecho de enviar un correo a un compañero de trabajo sentado junto a nosotros no es posible si no se invierte en infraestructura necesaria y esta se encuentra trabajando correctamente, lo cual, podría generar grandes costes de creación o mantenimiento de la red a cualquier compañía, los cuales serán graduales dependiendo de la cantidad de colaboradores que posea.

Con respecto a lo citado anteriormente, los mismos autores: (Haggle: a Networking Architecture Designed Around Mobile Users, 2006). Describen a las redes PSN (Pocket Switched Network), como las redes utilizadas por los usuarios móviles. En la actualidad la mayoría de los usuarios posee uno o varios dispositivos de comunicación portable, por ejemplo un teléfono inteligente y/o tableta, los cuales llevan consigo en todo momento. Si el usuario se conecta a un punto de acceso, también llamado isla de conectividad y pasa algún tiempo en ese lugar, puede comunicarse con otros usuarios vía internet, además de la posibilidad de intercambiar datos con todos los dispositivos en el mismo rango de manera local, utilizando medios inalámbricos, por ejemplo el 802.11. Así pues, el intercambio local de datos entre dispositivos es posible utilizando uno que se encargue de la conexión y no necesariamente debe estar conectado a internet para realizar ese proceso, con lo cual podemos proteger a los clientes de posibles ataques del exterior.

En lo que atañe a las redes ICN, según el trabajo: (Toward an Information-centric Delay-Tolerant Network, 2012). Los investigadores indican que las redes ICN centran sus esfuerzos en la entrega efectiva de contenido en lugar de la entrega de paquetes entre dispositivos, para lo cual utilizan algoritmos como el LRU (Last Recent Used), y cada ítem de contenido recibe un único identificador, para que posteriormente sea utilizado como medio de recuperación de la información que posee. De esta manera al utilizar las redes ICN, verificamos la entrega de los mensajes, se pueden descartar mensajes dependiendo de su uso o antigüedad y al utilizar un identificador único, se hace más sencillo su búsqueda y la manipulación del contenido.

En el tema de las redes oportunísticas se encuentran muchas perspectivas de comunicación para aplicaciones locales o de redes sociales, éstas ofrecen muchos mecanismos para soportar la comunicación en zonas remotas. Según los investigadores del siguiente trabajo: (WiFi-Opp: Ad-Hoc-Less Opportunistic Networking, 2011). Los dispositivos inteligentes pueden periódicamente escanear en búsqueda de puntos de acceso inalámbrico en el rango, si son

encontrados se conectan, sino pasan a funcionar como punto de acceso por un determinado lapso de tiempo. En relación con lo anterior, se denota el uso de redes oportunísticas para el descubrimiento y comunicación de dispositivos inalámbricos que se encuentren en el mismo rango o vecindario, lo cual puede ser un medio válido por utilizar en zonas remotas o de difícil acceso a internet.

Por último, en el documento: (Liberouter: Towards Autonomous Neighborhood Networking, 2014). Los investigadores diseñaron un sistema completo de comunicaciones, que permite a dispositivos vecinos comunicarse sin la necesidad de internet. Para lo cual usaron un enrutador de bajo costo como punto de acceso inalámbrico. Además este dispositivo ofrece almacenamiento y retransmisión de datos con una aplicación distribuida. Por otra parte, la administración de los datos se realiza con una aplicación web.

## 2.2 Aplicaciones móviles utilizadas en crowdsourcing:

En este apartado se pueden señalar como primer punto algunas las herramientas utilizadas para hacer crowdsourcing y posteriormente las utilizadas específicamente como medio de mensajería entre usuarios.

A continuación se presentan algunas aplicaciones móviles que se desarrollaron para colaborar con el crowdsourcing: (colaborar-en-proyectos-de-crowsourcing, 2013).

### Colaborar con OpenStreetMap

OpenStreetMap es un proyecto colaborativo que tiene como objetivo desarrollar un sistema de mapas totalmente libre y abierto.

Gracias a OSMTracker para Android y al GPS del teléfono podemos localizar las coordenadas de la ruta que hemos seguido y realizar anotaciones de puntos de interés, dirección de la circulación de las calles por las que pasamos, adjuntar fotos o notas escritas o de voz.

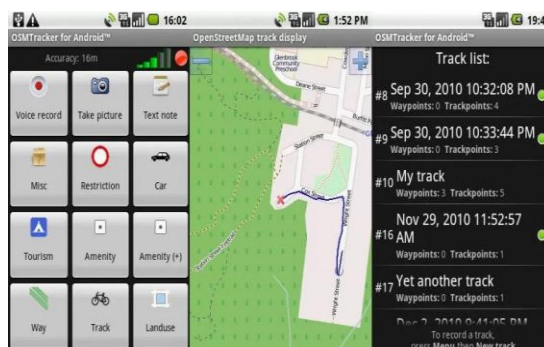


Figura 2. 1. OpenStreetMap permite generar mapas vehiculares.



### Crear un mapa con la cobertura móvil de los operadores

Con la idea de crear un gran mapa de cobertura móvil y Wi-Fi nació el proyecto OpenSignal. Usamos crowdsourcing para convertir nuestros dispositivos móviles en sondas de medida que sirven para realizar mediciones de cobertura y de QoS que percibimos, por ejemplo, al realizar una pequeña descarga de prueba. Todos estos datos, sumados a los de redes WiFi abiertas, forman una gran base de datos mundial.



Figura 2. 2. Finally genera un mapa de cobertura de redes WiFi.

### Trasformar nuestro teléfono inteligente en una estación meteorológica

Mediante el crowdsourcing, Weddar plantea a los usuarios una serie de descripciones sencillas que definen el tiempo de su localidad; un dato que pueden subir al servicio y así aportar cómo está el tiempo en su zona de manera sencilla y, sobre todo, muy visual.



Figura 2. 3. Weddar genera un reporte sobre el estado del tiempo.

### Traducir textos y aprender idiomas

Duolingo es un servicio que facilita a los usuarios una plataforma en la que pueden aprender idiomas de manera gratuita y sin introducir publicidad, solamente tendremos que aportar nuestro trabajo.

Con un modelo muy singular, Duolingo nos propone textos en inglés que debemos traducir al español, al francés, al italiano, al alemán o al portugués o viceversa; unos textos que no son casuales puesto que las frases por traducir proceden de páginas webs que requieren ser traducidas para que los contenidos estén accesibles a mucha más gente.



Figura 2. 4. Duolingo permite aprender idiomas.

### 2.3 Aplicaciones móviles similares:

Ahora bien, en el ámbito de la mensajería tenemos varias herramientas que se han diseñado con el fin de permitir el intercambio de datos entre usuarios. A continuación se presentará una pequeña descripción de las herramientas mayormente utilizadas en este apartado.

#### MeshMe

Nueva aplicación para iOS (aún están trabajando en la de Android) que te permite chatear con otros usuarios sin conexión a Internet utilizando la conectividad Bluetooth o Wi-Fi del teléfono.

Según el periodista de revista: (Zamorano, 2014). La idea de las redes inalámbricas malladas en teléfonos inteligentes es que cada teléfono que tenga la aplicación actúa como si fuera el nodo de una red, traspasando los datos de un teléfono al siguiente hasta que llegue a su destino final. Evidentemente, solo el destinatario final puede abrir el mensaje, pese a que éste pudo viajar por muchos otros teléfonos más para llegar a su destinatario.

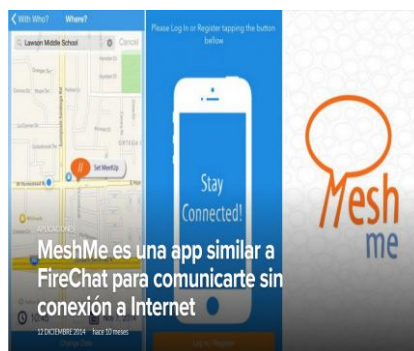


Figura 2. 5. Herramienta MeshMe para mensajería.

### FireChat

Creará una red de malla usando Bluetooth y WiFi P2P y transmite mensajes sin conexión entre dispositivos que estén ubicados a unos 70 metros de distancia uno del otro.

Según sus creadores: (OperGarden, 2015). Gracias a la tecnología innovadora de multisalto y de almacenamiento y reenvío, los usuarios de FireChat forman una red que aumenta de tamaño con el número de dispositivos. Permite que los mensajes salten de un dispositivo al siguiente para llegar al (los) destinatario(s). Por este motivo, cuantas más personas usan FireChat, mejor y más rápida se vuelve la red (al contrario que con las redes móviles).

En la red de malla, los mensajes se transmiten de manera ininterrumpida de un dispositivo al siguiente para llegar al (los) destinatario(s). Cuando una conexión a Internet o red móvil está disponible, FireChat las utiliza para llegar a los destinatarios remotos.

Los mensajes públicos son visibles para todos. Los mensajes privados están encriptados y solo el remitente y el destinatario pueden verlos y leerlos.

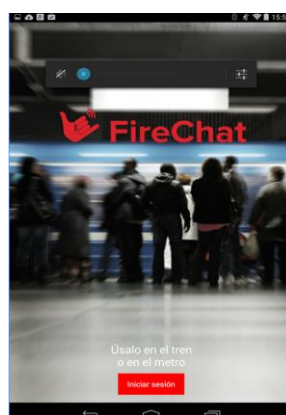


Figura 2. 6. Herramienta FireChat para mensajería.

## Liberouter

Red de comunicación oportunista que permite comunicaciones directamente entre los usuarios móviles cercanos sin necesidad de conexión a Internet. Según sus creadores: (Aalto University, 2012-2015). Cualquiera puede contribuir mediante la colocación de un dispositivo enrutador barato o la instalación de un enrutador de software Android en sus teléfonos. La red resultante es altamente resistente a la censura y la vigilancia centralizada. Liberouter actúa como buzones públicos donde los teléfonos Android cercanos pueden dejar y recoger las copias de los mensajes, si el buzón se llena, se descartan los mensajes más antiguos.



Figura 2. 7. Herramienta Liberouter utilizada para la mensajería.

## 2.4 Hardware utilizado en el experimento

### Raspberry Pi

El Raspberry Pi fue creado por la Fundación Raspberry Pi, una organización benéfica de Reino Unido creada para promover fines educativos.

El Raspberry es un computador de tamaño miniatura, pero muy potente, ya que posee un procesador ARM de última generación y 1 GB de memoria RAM. Además al ser tan pequeño es muy portable y ligero. A partir de febrero del 2015, se utiliza el Raspberry Pi 2. En este nuevo modelo se incluyen las siguientes características de hardware:

- CPU quad-core ARM Cortex-A7 con velocidad de 900MHz.
- 1GB RAM.
- 4 Puertos USB.
- 40 pines GPIO.
- Puerto Full HDMI.
- Puerto Ethernet.
- audio y composite video de 3.5mm.
- Camera interface (CSI)
- Display interface (DSI)
- Slot Micro SD card.
- VideoCore IV 3D graphics core

Desde su creación, Raspberry Pi ha tenido diferentes modelos de hardware y software. (ver figura 2.8). El sistema operativo que utiliza es el Raspbian basado en Linux Debian.



Figura 2. 8. Diferentes modelos de Raspberry Pi 2.

### Servidor Web

El Servidor Web combina varias tecnologías software con el fin de que la información sea mostrada al usuario mediante una interfaz sencilla y simple de utilizar.

Según Wikipedia un servidor web tiene la siguiente definición: (Wikipedia, 2015)

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.

El cliente puede enviar información al servidor web mediante los siguientes métodos:

**Método GET:** Sirve para obtener información del servidor como respuesta. Para esto se tiene que solicitar (request) algún dato, y el medio de envío es la URL. Este método es visible para el usuario.

**Método POST:** Consiste en enviar información desde el cliente al servidor para que sea procesada. Dichos datos son procesados dando como respuesta alguna página con información; los datos enviados aparecen en el cuerpo del mensaje http, por lo que este método es invisible para el usuario.

Existen varios servidores web, algunos necesitan licencias para su utilización y otros son del tipo de software libre y código abierto. Algunos ejemplos son: Apache, Nginx y Microsoft IIS, entre otros.

### **Teléfono Inteligente**

El teléfono inteligente está basado en tecnología Android. Los inicios del sistema operativo Android se remontan al año 2003. Los desarrolladores Andy Rubin, Rich Miner, Nick Sears and Chris White fueron sus creadores, hasta que en el año 2005 Google lo comprara.

También recibió apoyo del consorcio de varias compañías denominado Open Handset Alliance, y entre las que encuentran, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, Intel, LG, Marvell Technology Group, Motorola, y T-Mobile; el 9 de diciembre de 2008, 15 nuevos miembros se unieron al proyecto, incluyendo ARM Holdings, Atheros Communications, Sony Ericsson, Huawei, Toshiba, ZTE, entre otros; con el fin de desarrollar estándares abiertos para dispositivos móviles, y coincidiendo con la formación de la Open Handset Alliance, se presentó el primer producto, Android, construido sobre la versión 2.6 de Linux.

Está basado en GNU Linux y enfocado a dispositivos móviles de todo tipo, ya sean teléfonos móviles, tabletas e incluso mini ordenadores portátiles, que desde hace mucho se pueden encontrar en el mercado.

Su corazón está gobernado por un Kernel totalmente basado en Linux, y es el encargado de acoplar y hacer que todos los componentes de nuestro terminal funcionen correctamente en el sistema operativo. En definitiva, el Kernel es una de las cosas más importantes dentro de Android, y sin él, el sistema operativo no sería capaz de encontrar los componentes básicos de nuestro dispositivo, tales como el tipo de procesador y sus parámetros de uso, la conexión Wifi y sus protocolos de actuación o incluso la cámara de fotos.

La estructura del Sistema Operativo Android se compone de aplicaciones que se ejecutan en un framework Java, permitiendo desarrollar aplicaciones a través de una variante de este lenguaje de programación, para lo cual se necesitan los siguientes componentes de software:

- Un entorno de desarrollo (Eclipse Luna o Android Studio).
- Kit de desarrollo de Android (Android SDK).
- Kit de desarrollo de Java estándar (Java Development Kit ó JDK).

## Arquitectura de Android

Android es un sistema operativo creado para ser independiente de cualquier tipo de arquitectura de hardware en los dispositivos móviles. Esta característica hace que sea tan atractivo ante los fabricantes y desarrolladores.

Adicionalmente su portabilidad, flexibilidad y seguridad le dan ese toque de simpatía a las personas interesadas en los sistemas de código abierto.

### Modelo De Capas En La Arquitectura De Android

Android está construido con una arquitectura de 4 capas o niveles relacionados entre si. A continuación veremos un diagrama ilustrativo extraído del libro Learning Android escrito por Marko Gargenta y Masumi Nakamura:

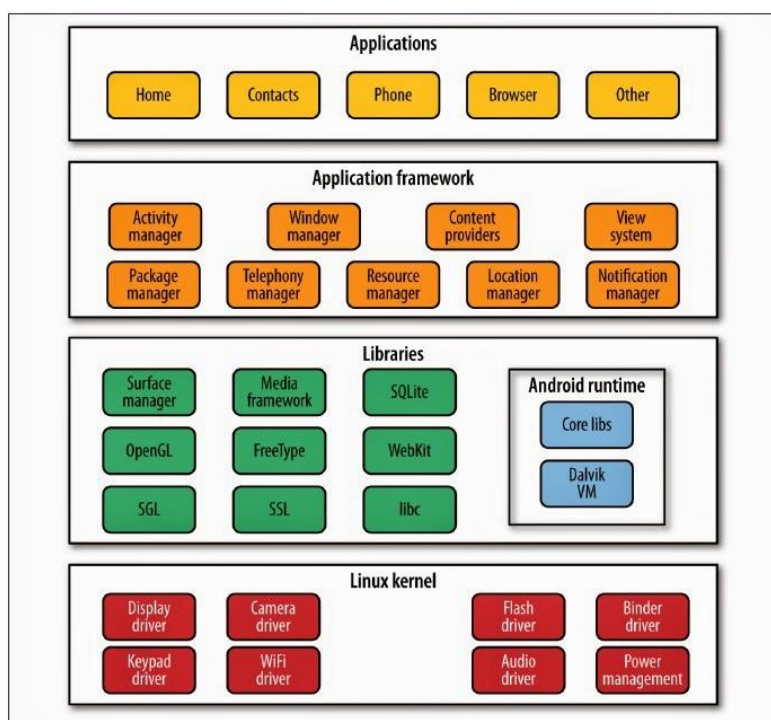


Figura 2. 9. Modelo de Capas del Sistema Operativo Android.

A continuación se presenta un pequeño resumen de las funciones de cada capa, tomado del siguiente artículo: (Revelo, 2014).

### El núcleo Linux

El núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de controladores para dispositivos.

Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del hardware.

### **Runtime de Android**

Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de correr Android (poca memoria y procesador limitado) no fue posible utilizar una máquina virtual Java estándar.

A partir de Android 5.0 se utiliza ART. Esta nueva máquina virtual consigue reducir el tiempo de ejecución del código Java hasta en un 33%.

También se incluye en el Runtime de Android el “core libraries” con la mayoría de las librerías disponibles en el lenguaje Java.

### **Librerías nativas**

Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son:

### **Entorno de aplicación**

Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas (sujetas a las restricciones de seguridad). Los servicios más importantes que incluye son:

**Views:** extenso conjunto de vistas, (parte visual de los componentes).

**Resource Manager:** proporciona acceso a recursos que no son en código.

**Activity Manager:** maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.

**Notification Manager:** permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.

**Content Providers:** mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos).

### **Aplicaciones**

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de correr en la máquina virtual Dalvik o ART para garantizar la seguridad del sistema.



Normalmente las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java podemos utilizar el Android

## 2.5 Dispositivo de comunicación entre dispositivos utilizado en el experimento

El enlace entre el Raspberry Pi y el dispositivo Android se realiza mediante la tecnología WiFi. WiFi, también conocido como Wi-Fi, es una marca comercial de Wi-Fi Alliance. Empresa que se encarga de autenticar y certificar equipos inalámbricos que cumplen con las normas del estándar 802.11.



Figura 2. 10. Logo de WiFi.

El objetivo tras la marca WiFi es fomentar las conexiones inalámbricas y facilitar la compatibilidad de los distintos equipos. Todos los productos con conectividad WiFi tienen certificada su interoperabilidad.

En la actualidad, los estándares certificados por WiFi son muy populares en todo el mundo. Este crecimiento amenaza la disponibilidad del espectro radioeléctrico, sobre todo cuando las conexiones deben concretarse a más de 100 metros de distancia.

Uno de los principales defectos atribuidos a la conectividad WiFi es su poca seguridad. Existen, sin embargo, diversos protocolos de cifrado que permiten codificar la transmisión de los datos y garantizar su confidencialidad.

La infraestructura de una conexión WiFi incluye puntos de acceso (emisores remotos), enrutadores (que reciben la señal que emite el operador de telefonía) y dispositivos de recepción (tarjetas USB, PCI o PCMCIA).

Algunas de las ventajas de usar WiFi son las siguientes: (definicion.de, 2008-2015).

- Las redes WiFi resultan especialmente útiles en los casos que no admiten el uso de cables; por ejemplo, son muy usadas en salas de conferencia y exhibiciones internacionales, y también son ideales para centros comerciales.
- En muchas ciudades, existen puntos de acceso gratuitos en lugares públicos tales como parques, centros comerciales, estaciones de metro y aeropuertos.

La API WiFiManager es encargada de manejar la conexión a nivel del sistema operativo Android. Algunas de sus características son las siguientes:

- La lista de redes configuradas. La lista puede ser vista y actualizada.
- La red WiFi activa actualmente, si la hubiere. La conectividad puede ser establecida y se puede obtener la información sobre el estado de la red.
- Los resultados de las exploraciones de puntos de acceso, y datos que se emiten a cualquier cambio en el estado de WiFi.

## 2.6 Software utilizado en el experimento

### Android Studio

Android Studio es un entorno de desarrollo integrado que reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. Está basado en el software IntelliJ IDEA de JetBrains, y es publicado de forma gratuita mediante la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

### Laravel

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Su filosofía es desarrollar código PHP de forma elegante, expresiva y que permite crear código de forma sencilla. Intenta aprovechar lo mejor de otros frameworks. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

### MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones, es software libre en un esquema de licenciamiento dual. MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google, Facebook, Twitter, Flickr, y YouTube.

### NGINX

NGINX es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3). Es software libre y de código abierto, licenciado bajo la Licencia BSD simplificada. Es multiplataforma, por lo que corre en sistemas tipo Unix (GNU/Linux, BSD, Solaris, Mac OS X, etc.) y Windows. El sistema es usado por una

larga lista de sitios web conocidos, como: WordPress, Netflix, Hulu, GitHub, SourceForge, TorrentReactor y partes de Facebook.

### RabbitMQ RabbitMQ.

RabbitMQ funciona como software de negociación de mensajes de código abierto, y entra dentro de la categoría de middleware de mensajería. Implementa el estándar Advanced Message Queuing Protocol (AMQP). El servidor RabbitMQ está escrito en Erlang y utiliza el framework Open Telecom Platform (OTP) para construir sus capacidades de ejecución distribuida y conmutación ante errores. Rabbit Technologies Ltd, fue adquirida en abril de 2010 por la división SpringSource de VMWare.

### SQLite SQLite

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kiB)<sup>2</sup> biblioteca escrita en C.

A diferencia de los demás sistemas de gestión de bases de datos cliente-servidor existentes, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite mediante llamadas simples a subrutinas y funciones. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

### PHP

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

PHP se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy, lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico, como Facebook, para optar por el mismo como tecnología de servidor.

## Capítulo III: Análisis del sistema.

En el capítulo anterior se describieron todos los componentes tecnológicos empleados para elaborar la plataforma de crowdsourcing planteada. Teniendo en cuenta estas herramientas se debe contemplar la metodología de desarrollo de software por utilizar.

Según Wikipedia: (Wikipedia, 2015) Una metodología de desarrollo se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de creación de un sistema de información.

Ahora bien, no existe una metodología universal para el desarrollo de software y no todas las metodologías son aplicables a todo tipo de proyectos, más bien cada tipo de proyecto tiene una metodología a la que se adapta mejor.

Una metodología de desarrollo consiste en:

- Una filosofía de desarrollo de software con una base de procesos de desarrollo de software.
- Múltiples herramientas, modelos y métodos para asistir al desarrollo del proyecto.
- Suele estar documentada y posee algún tipo de documento formal.

La metodología elegida para el desarrollo es la Cascada, aunque tiene varios inconvenientes que pueden afectar con los tiempos de entrega o el rediseño del producto, sigue siendo de las metodologías más utilizadas a la hora de elaborar software.

### Capas del modelo en cascada

Según (Managing the development of large software system, 1970), estos son los pasos necesarios para el desarrollo de proyectos de software

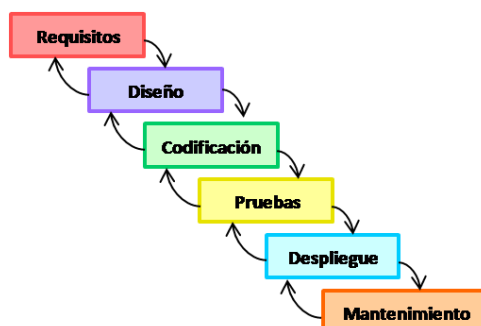


Figura 3. 1. Fases que componen la metodología Cascada.

Además el mismo autor anterior presenta un resumen de las fases que componen dicha metodología.

### **Análisis de requisitos**

En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.

### **Diseño del Sistema**

Descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

### **Diseño del Programa**

Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber qué herramientas usar en la etapa de Codificación

### **Codificación**

Es la fase en donde se implementa el código fuente, haciendo uso de prototipos así como de pruebas y ensayos para corregir errores.

### **Pruebas**

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que cumple con los requisitos, antes de ser entregado al usuario final.

### **Verificación**

Es la fase en donde el usuario final ejecuta el sistema, para ello el o los programadores ya realizaron exhaustivas pruebas para comprobar que el sistema no falle.

## Mantenimiento

Una de las etapas más críticas, ya que se destina un 75 % de los recursos, es el mantenimiento del Software ya que al utilizarlo como usuario final puede ser que no cumpla con todas nuestras expectativas.

Por consiguiente, este capítulo describe las principales características del proyecto realizado, las técnicas aplicadas, los casos de usos y comportamiento del sistema.

### 3.1 Descripción del sistema

Como se ha comentado anteriormente, este proyecto pretende ofrecer al usuario una aplicación con la cual pueda acceder a ofertas generadas por locales comerciales cercanos a su ubicación utilizando puntos de acceso inalámbricos con la tecnología WiFi. Además cada cliente puede leer y crear comentarios sobre los servicios ofrecidos por un local en específico, ya sean para el público en general o exclusivamente para un grupo de usuarios. Esta información va a ser listada de manera que el usuario observe la más reciente primero. El administrador del punto de acceso, mediante una aplicación web, puede crear nuevas promociones o leer los comentarios que hayan generado los usuarios. Además puede realizar un respaldo de la información en un archivo que se guarda en google drive, esto como medida de contingencia por si el dispositivo Raspberry pi sufriera algún daño.

Al conectarse un nuevo cliente al punto de acceso, se ejecutan una serie de eventos tanto del lado del cliente como del lado del backend, cuya finalidad es lograr la completa sincronización de los dispositivos. Si existiera alguna promoción o comentario que exista únicamente en un dispositivo, se solicita la información para lograr la sincronización completa de los mensajes tanto en el lado del cliente como en el punto de acceso.

El dispositivo cliente donde se ejecuta la aplicación móvil está basado en tecnología Android, y proporciona la posibilidad de leer las promociones y generar comentarios acerca de los productos o la atención recibida en el local comercial, luego procede a almacenar la información en su base de datos interna SQLite. Además sincroniza la información con el punto de Acceso. Para realizar lo anterior, se tienen subrutinas que ayudan a establecer la conexión con el punto de acceso y que promueven el intercambio de mensajes entre los dispositivos. Como se indicó en el capítulo III, la tecnología empleada para la comunicación entre dispositivos es el WiFi.

El Servidor Web NGINX, alojado en el Punto de acceso, tiene la función de procesar la aplicación que va a generar nuevas promociones, ver los comentarios de los clientes y guardar la información almacenada en su base de datos MySQL usando un servicio de google drive, de tal forma que se pueda proteger la información en caso de una catástrofe. Para tener acceso a estas opciones se necesitará de un usuario que inicie sesión en el portal web creado.

Por último, el punto de acceso tiene scripts que ayudan a realizar el manejo de mensajes utilizando un servidor de colas RabbitMQ. Esto es necesario para ofrecer el servicio a múltiples usuarios, que deseen obtener información o proporcionar información al punto de acceso. En este mismo dispositivo, se encuentra los procesos para sincronizar las bases de datos que poseen los clientes Android y la del punto de acceso, utilizando la tecnología PHP para este fin.

## 3.2 Requisitos del Sistema

Para detallar la funcionalidad del sistema se ha utilizado la clasificación de requisitos funcionales y no funcionales.

- **Los requisitos funcionales** son declaraciones de los servicios que proveerá el sistema, refiriéndose a las entradas, comportamientos y salidas. Ejemplos de estos son: la creación de promociones, intercambio de datos, otros.
- **Los requisitos no funcionales** describen todos los requisitos que no afectan a un comportamiento específico del sistema. Ejemplo de estos son: herramientas, coste, interfaz, etc.

### 3.2.1 Requisitos Funcionales

#### 3.2.1.1. Aplicación móvil

- RF.1.1.1 El sistema debe iniciar el proceso de búsqueda de redes WiFi usando el nombre del SSID al iniciar la aplicación.
- RF.1.1.2 El sistema debe permitir visualizar los nuevos mensajes o comentarios obtenidos tras la sincronización de las bases de datos.
- RF.1.1.3 La aplicación Android debe permitir a los clientes generar nuevos comentarios, de manera pública o utilizando grupos de usuarios.
- RF.1.1.4 El sistema debe permitir almacenar la información en la base de datos SQLite.

### 3.2.1.2. Aplicación Web

- RF.1.2.1 El sistema debe permitir recibir peticiones de la aplicación móvil.
- RF.1.2.2 El sistema debe permitir iniciar sesión.
- RF.1.2.3 El sistema web debe permitir ingresar nueva información acerca de promociones y guardarlo en la base de datos del punto de acceso.
- RF.1.2.4 El sistema debe permitir modificar las opciones de configuración del local comercial y respaldar la información en un servicio de google drive.

### 3.2.1.3. Punto de Acceso

- RF.1.3.1 La plataforma debe ser capaz de escuchar las peticiones provenientes de los clientes Android.
- RF.1.3.2 La plataforma debe ser capaz de sincronizar las bases de datos de los clientes y el punto de acceso.
- RF.1.3.3 La plataforma debe permitir almacenar la nueva información emitida por los clientes de la aplicación en la base de datos MySQL.

## 3.2.2 Requisitos no Funcionales

### 3.2.2.1 Aplicación móvil

- RN.2.1.1 La aplicación móvil usa el entorno de desarrollo Android Studio de Google, entre sus principales características tenemos:
  - Versión 1.2.1.1.
  - JRE: Java 1.7.0\_75.
- RN.2.1.2 La versión compilada puede instalarse en un dispositivo con sistema operativo Android versión 4.0.x. o superior.
- RN.2.1.3 El idioma de la aplicación móvil es el español.
- RN.2.1.4 La aplicación móvil debe comunicarse mediante un dispositivo WiFi.
- RN.2.1.6 La aplicación móvil necesita como mínimo 20 MB de almacenamiento disponible para su instalación y funcionamiento.
- RN.2.1.7 La aplicación móvil debe mostrar una interfaz conforme a los nuevos estándares de desarrollo de google, que sea amigable, intuitiva y fácil de utilizar.



### 3.2.2.2 Aplicación Web

- RN.2.2.1 La aplicación web usa las siguientes características técnicas en su desarrollo:
  - Ambiente de desarrollo Laravel
  - Servidor NGINX para hospedar las páginas en el Punto de Acceso.
  - PHP versión 5.5.9.
  - JSON encode.
  - MySql versión 5.6.25 Community Server (GPL).
- RN.2.2.2 La aplicación web puede instalarse en servidores con sistema operativo Linux Debian. Se recomienda un espacio de almacenamiento superior a 8 GB para el almacenamiento de las promociones y comentarios y como mínimo 1GB de memoria RAM.

### 3.2.2.3 Punto de Acceso

- RN.2.3.1 La plataforma que se utiliza como punto de acceso posee las siguientes características técnicas:
  - Sistema operativo Raspbian basado en Debian 7.0.
  - Servidor NGINX para hospedar los store script.
  - PHP versión 5.5.9.
  - JSON encode.
  - Java 7 JDK

## 3.3 Diagramas y Casos de Uso

Los casos de uso son una técnica que describe el comportamiento de un sistema y sus especificaciones, involucrando al actor como usuario del sistema. La interacción entre actores no se ve en el diagrama de casos de uso. Si esta interacción es esencial para una descripción coherente del comportamiento deseado. Los actores son una especie de rol, un usuario humano u otra entidad externa pueden jugar varios papeles o roles

El proyecto está conformado por 3 componentes los cuales se citan a continuación:

- La aplicación móvil nativa en Android, donde su actor principal es el usuario móvil.
- El sistema web, que tiene como actores al administrador web.
- El punto de acceso es el último componente, los actores que influyen son los store scripts

### 3.3.1 Caso de uso de la Aplicación Móvil Nativa

En la figura 3.2 se muestra el caso de uso del componente móvil y en la tabla 1 se muestra la relación con los requisitos funcionales antes citados.

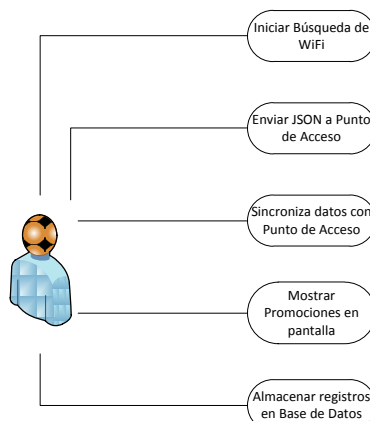


Figura 3. 2. Diagrama de casos de uso de la aplicación móvil nativa.

| Casos de uso - Aplicación Móvil Nativa |                                   |                                  |
|--|-----------------------------------|----------------------------------|
| Identificador                          | Nombre                            | Requisitos Funcionales cubiertos |
| C.1.1                                  | Iniciar Búsqueda de WiFi          | RF.1.1.1                         |
| C.1.2                                  | Enviar JSON a Punto Acceso        | RF.1.1.1                         |
| C.1.3                                  | Sincroniza datos con Punto Acceso | RF.1.1.2                         |
| C. 1.3.1                               | Mostrar Promociones en pantalla   | RF.1.1.2                         |
| C.1.4                                  | Enviar comentarios a Servidor Web | RF.1.1.3                         |
| C.1.5                                  | Almacenar registros en Base Datos | RF.1.1.4                         |

Tabla 1. Relación casos de uso de la Aplicación Móvil - Requisitos Funcionales.

### 3.3.2 Caso de uso del Sistema Web

En la figura 3.3 se muestran los casos de uso del componente web y en la tabla 2 se muestra la relación de éstos con los requisitos funcionales.

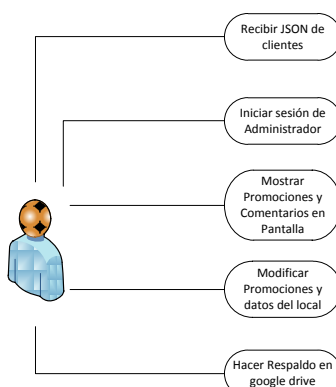


Figura 3. 3. Casos de uso utilizados en la Aplicación Web.

| Casos de uso - Aplicación Web |   |                        |
|-------------------------------|---|------------------------|
| Identificador                 | Nombre  | Requisitos Funcionales |
| C.1.1                         | Recibir JSON de clientes                      | RF.1.2.1               |
| C.1.2                         | Iniciar sesión de administrador               | RF.1.2.2               |
| C.1.3                         | Mostrar Promociones y comentarios en pantalla | RF.1.2.3               |
| C.1.4                         | Modificar promociones y datos del local       | RF.1.2.4               |
| C.1.4.1                       | Almacenar cambios en google drive             | RF.1.2.4               |

Tabla 2. Relación casos de uso de la Aplicación Web - Requisitos Funcionales.

### 3.3.3 Caso de uso del Punto de Acceso

En la figura 3.4 se muestran los casos de uso del Punto de Acceso y en la tabla 3 se muestra la relación de éstos con los requisitos funcionales.

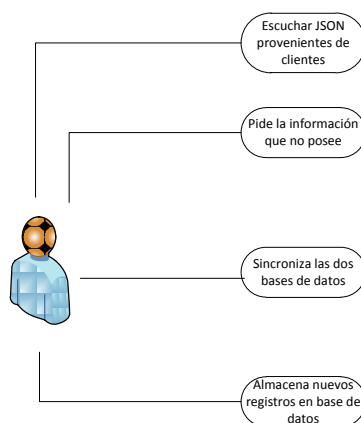


Figura 3. 4. Casos de uso utilizados en el Punto de Acceso.

| Casos de uso – Punto de Acceso |   |                        |
|--------------------------------|---|------------------------|
| Identificador                  | Nombre  | Requisitos Funcionales |
| C.1.1                          | Escuchar JSON provenientes de clientes          | RF.1.3.1               |
| C.1.2                          | Pide la información a los clientes que no posee | RF.1.3.2               |
| C.1.2.1                        | Sincroniza las dos bases de datos               | RF.1.3.2               |
| C.1.3                          | Almacenar nuevos registros en Base Datos        | RF.1.3.3               |

Tabla 3. Relación casos de uso del Punto de Acceso – Requisitos funcionales.

## Capítulo IV: Diseño del Sistema

En este capítulo se describe la arquitectura e interfaz de los componentes de la plataforma desarrollada. Posteriormente se demostrará la funcionalidad de la aplicación web y la aplicación móvil. Además de las interfaces y las bases de datos que utilizan.

Para la realización de este proyecto de investigación se utilizaron varios componentes de hardware y herramientas para el desarrollo de software, los cuales son utilizados para elaborar una plataforma para aplicaciones crowdsourcing que realice el descubrimiento de redes WiFi, la sincronización de mensajes entre dispositivos y el manejo de la información por divulgar.

En la figura 4.1 se puede observar el esquema de componentes de hardware utilizados. Entre los que podemos distinguir tres componentes principales: el servidor web, el Raspberry Pi y el teléfono inteligente utilizando la tecnología Android.

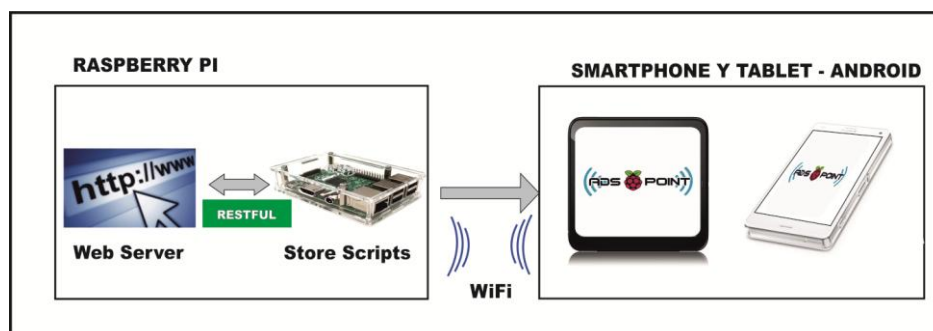


Figura 4. 1. Esquema de componentes de hardware utilizados.

### 4.1 Componentes utilizados

#### Raspberry Pi

El primer componente de hardware por analizar es el Raspberry Pi, el cual funciona como punto de acceso, por lo que el primer paso es la configuración del Raspberry Pi para que sirva como tal servicio. A su vez debe servir como DHCP y asignar una dirección IP a los dispositivos que se conecten a él. Luego de la configuración de los archivos, se configura el servidor web, utilizando un servidor HTTP NGINX, el cual provee alto rendimiento, una configuración simple y bajo consumo de recursos. Además se utiliza otro proceso de NGINX para correr los scripts que escuchan las peticiones que envían los teléfonos Android por el puerto 80, si el programa posee muchos usuarios en un momento dado, es necesario guardar las peticiones, para lo cual se utiliza un servidor de colas usando el protocolo AMQP.

Raspberry Pi utiliza Python como entorno de desarrollo incorporado, además soporta programación en PHP (Pre-Procesador en Hipertexto). Su sistema operativo es conocido como Raspbian, el cual es una distribución basada en Debian. Se utilizan consultas por medio de http para recibir los JSON (JavaScript Object Notation) enviados por los dispositivos móviles y son procesados mediante métodos escritos en PHP, este proceso se realiza cada vez que se conecta a una nueva red WiFi que posea la plataforma CrowdPi instalada. Luego de establecer la comunicación entre los dispositivos, se realizan diferentes procesos de consulta en la base de datos MySQL y se comparan las diferentes cabeceras de mensajes que posee cada dispositivo. Posteriormente el Raspberry Pi, envía un JSON solicitando el envío de la información que no se posee al dispositivo Android. Dicha información sería promociones provenientes de otros locales comerciales y es guardada en la base de datos del punto de acceso. Mediante una serie de arreglos se realiza una sincronización de los datos entre los diferentes dispositivos.

### **Servidor Web**

El servidor web, el cual se encuentra alojado en el Raspberry Pi tiene como función principal la administración de los datos que se va a proporcionar a los clientes Android que estén en su rango, (Promociones). Además, otro punto por destacar es que el usuario del servidor web puede observar los comentarios otorgados por los clientes acerca de las promociones que el local ha publicado. El programa CrowdPi, utiliza el puerto 8000 del servidor, para dar mantenimiento a las promociones o datos del local donde se instaló el punto de acceso.

Por otra parte, el administrador del sitio puede realizar un respaldo de los datos, para lo cual la aplicación web le permite subir un fichero al google drive de toda la información del local y las promociones que se posean en un determinado momento y puede restaurar dicha información descargando el archivo y seleccionándolo en la página de configuración.

Otro punto a destacar es la utilización de los servicios web para la comunicación entre el dispositivo Android y el servidor web. Un servicio web es una solución que usa cualquier tecnología web como PHP o JAVA entre otros. Los servicios tienen la función de facilitar la interoperabilidad entre varios sistemas, independiente del lenguaje de programación o sistema operativo en que fueron desarrollados. Estos servicios web deben cumplir un formato estándar entendible por diferentes plataformas, siendo XML o JSON un buen ejemplo de ello. En este trabajo se utilizó el formato JSON. El cual es un formato de intercambio de datos liviano y sencillo, por lo que ha dado lugar a la ampliación de su uso.

Para la elaboración de este Trabajo de Fin de Máster se utilizó el servidor web Nginx, por su rendimiento, seguridad y fiabilidad. Además de ser un producto catalogado como software libre.

A continuación se muestra en la figura 4.2 la manera como se comunican los dispositivos Android y el servidor web NGINX. Para el intercambio de los mensajes que poseen ambos dispositivos.

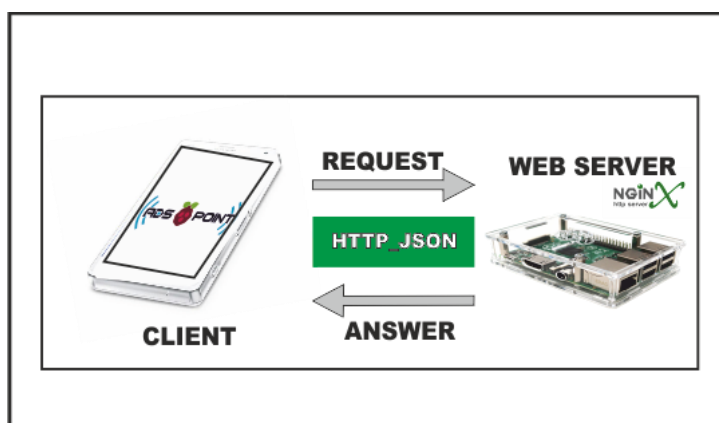


Figura 4. 2. Transmisión de datos entre Cliente Android y el Servidor Web.

En la siguiente figura se observa el formato del JSON que posee la información del local y es transmitido entre el dispositivo Android y el Servidor Web.

```
{"body":{"locales":[{"id":"1","header":"48569857","nombre":"pulperia az de copas","telefono":"24602332","correo":"golozo23@hotmail.com","descripcion":"billar","ubicacion":"sucre"}
```

Figura 4. 3. Ejemplo del formato de mensaje JSON.

### Teléfono Inteligente

El teléfono inteligente Android es otro componente utilizado, en el cual se instala el software desarrollado en Android Studio.

De primera mano, el dispositivo móvil busca realizar la conexión con cualquier punto de acceso WiFi, al ingresar a la red envía un JSON con los encabezados de mensajes que se poseen en la base de datos. Si recibe contestación es que el punto de acceso posee la plataforma de comunicación instalada. Si no, sigue intentando conectarse con otras redes inalámbricas que estén en el rango. Con la aplicación se permite observar las diferentes promociones que posee el local comercial al que nos conectamos, además como es una aplicación crowdsourcing, podemos crear comentarios acerca de los servicios ofrecidos y publicarlos de manera pública o

utilizando algún grupo de usuario al que pertenezca el usuario. Estos grupos se crean en la aplicación Android, para lo cual se envían mensajes SMS a clientes que posean instalado la aplicación y un proceso de escucha que se ejecuta constantemente se encarga de agregar al nuevo miembro a un grupo determinado. Además los usuarios pueden filtrar las promociones para facilitar su búsqueda, utilizando categorías a las que pertenece el local comercial de acuerdo al tipo de actividad que realiza.

La aplicación genera peticiones al servidor web periódicamente. Esto con el fin de sincronizar las bases de datos de ambos dispositivos utilizando los scripts que se ejecutan en el punto de acceso, lo cual permite al cliente tener información actualizada en un período muy corto de tiempo.

Para el proyecto se utilizó un dispositivo WiFi USB (Universal Serial Bus) producido por la compañía Realtek, como el que se muestra en la figura 4.4.

El Realtek RTL8188CUS es una tarjeta de red USB 2.0 inalámbrica, compatible con la especificación 802.11n. Este componente ofrece una solución completa para un alto rendimiento.



**Figura 4. 4. Dispositivo WiFi Realtek RTL8188CUS USB.**

La siguiente ilustración tiene la finalidad de describir las diferentes tecnologías utilizadas para la elaboración de este Trabajo Fin de Máster. En la figura 4.5 se puede apreciar la arquitectura de hardware y software utilizada para el desarrollo del proyecto. Además en esta misma figura, se va a mostrar la manera en que se comunican todos los componentes antes citados.

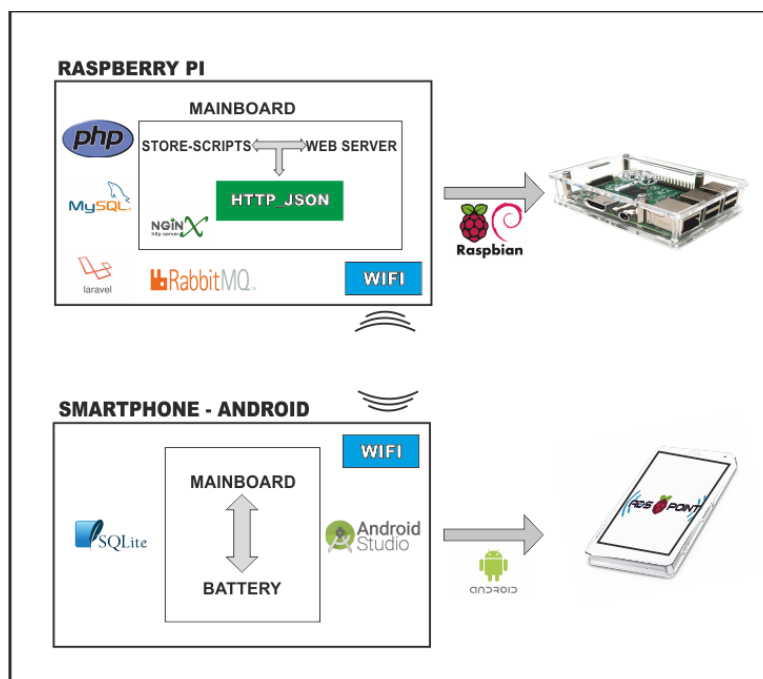


Figura 4. 5. Tecnologías software utilizadas en el sistema propuesto.

## 4.2 Aplicación Web CrowdPi

A continuación se detalla los componentes que integran la aplicación web de la plataforma desarrollada, la cual se encarga de la configuración y mantenimiento de promociones y la información pertinente al local donde se instaló en punto de acceso.

En la figura 4.6 se puede observar el menú que se obtiene luego de haber hecho el ingreso al sistema por parte de un usuario autenticado. Al posesionarse sobre uno de los ítems, este se pone de un color turquesa.



Figura 4. 6. Diferentes opciones del menú de la aplicación web.

### 4.2.1 Arquitectura

El sistema web fue elaborado utilizando el framework Laravel, haciendo uso de la tecnología PHP, utilizando además una base de datos MySQL y objetos JSON para enviar la información al Raspberry Pi. En este dispositivo se encuentra instalado un servidor NGINX que se encarga de



guardar la información sobre las nuevas promociones en la base de datos. En la figura 4.7 se muestra el diagrama de clases sobre el cual se basa su diseño.

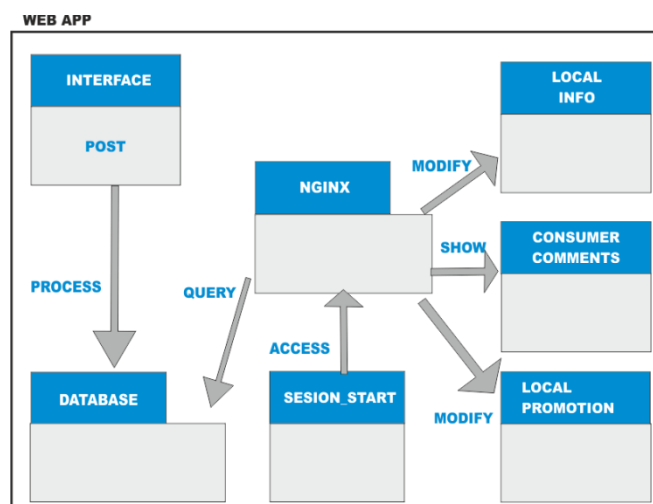


Figura 4. 7. Diagrama de clases de la aplicación web.

Laravel proporciona una serie de ventajas por la cual es una de las mejores opciones para el desarrollo de sistemas web en la actualidad, entre ellas encontramos las siguientes:

- Reducción de costos y tiempos en el desarrollo y mantenimiento.
- Curva de aprendizaje relativamente Baja.
- Flexible y adaptable.
- Abundante y buena documentación en la web. Además de muchas comunidades y foros de ayuda.

Lo primero a desarrollar fue una página de ingreso al sitio, en la cual el usuario administrador se debe autenticar para acceder al menú mostrado anteriormente.

Cada ítem del menú posee una página PHP. Cada página web posee código indispensable para la realización de los requisitos funcionales antes citados. Las siguientes son las páginas utilizadas para la aplicación web.

- login.php
- LocalController.php
- ComentarioController.php
- PromocionController.php

El archivo LocalController.php se encarga de cargar y modificar los datos del local donde se encuentra el punto de acceso. Además tiene la función de guardar los datos y promociones en

google drive, para poder proteger la información en caso de ocurrir alguna catástrofe o malfuncionamiento del dispositivo Raspberry Pi.

Entre las funciones utilizadas para esta página tenemos:

#### **Función Guardar:**

La cual sirve para guardar cambios que ocurrieran en la información de un local.

#### **Función Store:**

Carga la información actual que se posee del local comercial.

#### **Función Backup:**

Sirve para respaldar la información en google drive, por si ocurriera algún problema con el dispositivo o algún siniestro en el comercio donde se instaló en punto de acceso. A continuación se muestra el código utilizado para realizar la función backup:

```
public function backup()
    {
        $this->layout->titulo = 'Backup';
        $client = new Google_Client();
        $client->setApplicationName("CrowdPi");
        $client->setClientId('xxx');
        $client->setClientSecret('xxx');
        $client->setRedirectUri('xxx');
        $client->setScopes(array('https://www.googleapis.com/auth/drive'));
        if (Input::has('code')) {
            $accessToken = $client->authenticate(Input::get('code'));
            $client->setAccessToken($accessToken);
        }

        if ($client->getAccessToken()) {
            $credentialsPath = $this->expandHomeDirectory(BACKUP_PATH);
            if (!file_exists(dirname($credentialsPath))) {
                mkdir(dirname($credentialsPath), 0700, true);
            }
        }
    }
}
```

**Figura 4. 8. Función Backup de información del local.**

En el archivo ComentarioController.php se listan todos los comentarios del local de manera descendente, para que se muestren los más nuevos de primero.

Con respecto al `PromocionController.php` este se encarga de crear las nuevas promociones, modificarlas y borrarlas. Cabe destacar que esta última función no elimina la información de la base de datos.

#### **Función Create:**

Muestra las promociones del json en una lista

#### **Función Store:**

Guarda las promociones en la base de datos

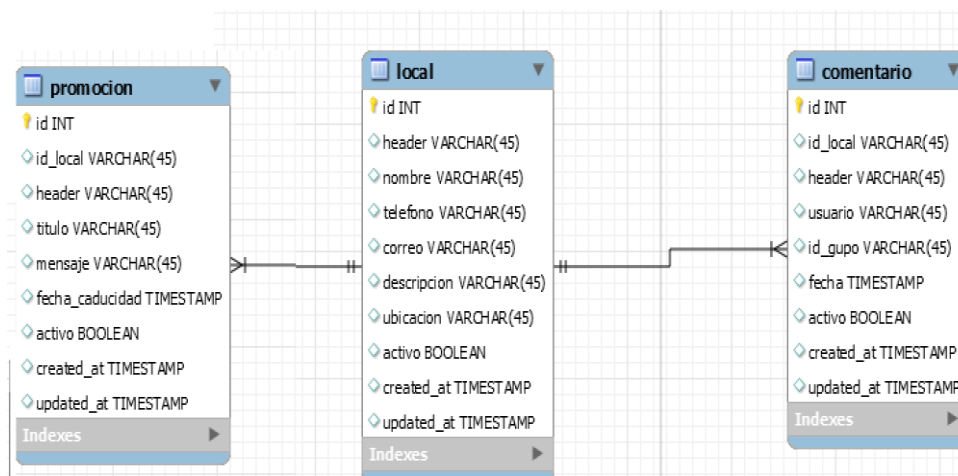
#### **Función Guardar:**

Guarda las modificaciones hechas a las promociones

#### **Función Borrar:**

Se encarga de poner una fecha inválida a la promoción, no la borra de la base de datos por si el administrador desea ver un historial de promociones hechas en el local.

A continuación en la figura 4.9 se muestra el diagrama de Entidad Relación utilizado en la aplicación web.



**Figura 4. 9. Modelo Entidad-Relación de la aplicación web.**

### **4.2.2 Interfaz Sistema Web**

La interfaz se ha diseñado de manera simple e intuitiva, especialmente realizada para usuarios que no posean mucha experiencia en este tipo de aplicaciones. Con un menú y botones llamativos. Por lo que el administrador del sitio web va a tener una aplicación muy sencilla de

utilizar. En este apartado se explica la funcionalidad de las páginas desarrolladas para el sistema web.

Luego de haber ingresado el login y contraseña correctamente en el sistema, la aplicación inicia primeramente con la página de promociones.

A continuación se muestra dicha página, la cual está compuesta por el título de la promoción, el mensaje que va a recibir el cliente, la fecha de expiración de la promoción y los botones de edición y borrar.

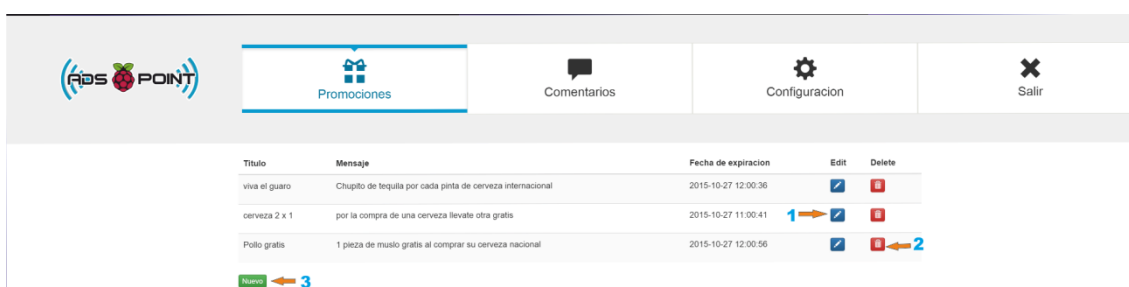


Figura 4. 10. Pantalla de Promociones del sistema web.

A continuación se van a explicar los botones de la pantalla de promociones siguiendo la guía numeral que se encuentra en la figura 4.10.

- (1) Se puede editar una promoción presionando el botón azul "Edit". Estos cambios se hacen utilizando el id de la promoción.
- (2) Al presionar el botón rojo "Delete", se actualiza la fecha de expiración, para que esta promoción pierda vigencia y no se vuelva a ofrecer a los clientes, sin embargo se conserva en la base de datos para que el administrador pueda tenerla como referencia.
- (3) Al dar clic sobre el botón verde, se muestra una ventana emergente similar a la figura 4.11. En la cual se deben de agregar todos los datos que se solicitan.
- (4) Al finalizar se debe catalogar la nueva promoción en una categoría. Esto se realiza para hacer más fácil la búsqueda de las promociones en la aplicación móvil. Una vez escogida la categoría, se da clic en el botón naranja "Guardar".

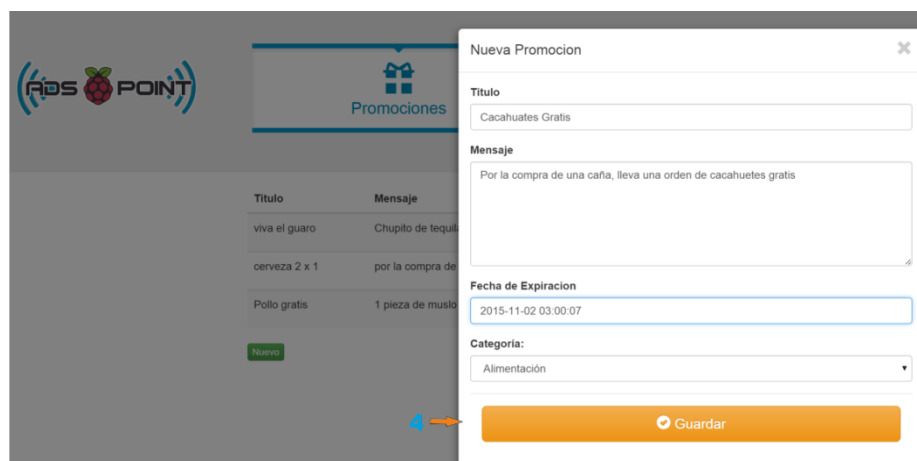


Figura 4. 11. Pantalla para guardar nuevas promociones.

Otra de las páginas que posee el sistema web es la de comentarios, en la cual se listan todos los comentarios referentes al local comercial, por lo que el administrador del sitio web puede leer información acerca de las promociones efectuadas, la atención al público y la opinión en general que tengan los consumidores de sus productos y servicios.

En esta página se observa el nombre del local de quien hace referencia el comentario, el nombre del usuario que realizó la anotación, la fecha y hora en que se efectuó el mismo y el contenido del mensaje.

Cabe destacar que los comentarios siguen un orden descendente, por lo que los más recientes son mostrados de primero.

En la figura 4.12 Se puede observar un ejemplo de la página de comentarios realizada por la aplicación CrowdPi.

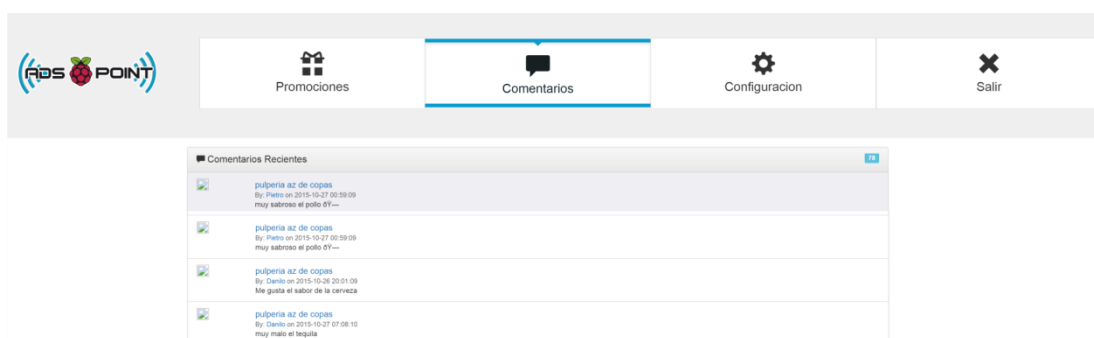


Figura 4. 12. Listado de Comentarios realizados por clientes.

Por otra parte, la página de configuración, ver figura 4.13, permite al administrador del sitio poder modificar la información del local donde se encuentra instalado el punto de acceso, además se puede ejecutar un respaldo de todas las promociones, comentarios y la información del local.

- (1) Se puede editar los datos del local comercial como por ejemplo: el nombre del local, la descripción, la ubicación, el teléfono y el correo. Posteriormente se procede a dar clic sobre el botón “Guardar” para almacenar los datos realizados.
- (2) Al presionar el botón verde “Backup”, se realiza un respaldo completo de toda la información que posee el punto de acceso en google drive. Esto con la finalidad de evitar que por un mal funcionamiento del Raspberry pi o por una catástrofe se pierda la información que almacena la base de datos.

Figura 4. 13. Datos del local y respaldo de la aplicación web.

El administrador puede salir del sitio web al dar clic en el botón salir, el cual se encuentra en el menú horizontal.

### 4.3 Aplicación Móvil CrowdPi

A continuación se detalla los componentes que integran la aplicación nativa desarrollada usando la tecnología Android, la cual se encarga de la conectarse al punto de acceso, mostrar las promociones que ofrece los puntos de accesos cercanos a la posición del dispositivo móvil. Además se pueden generar comentarios ya sea de manera privada, usando algún grupo de usuarios al que pertenezca el cliente o de manera general para que sea visto por todo el público en general.

### 4.3.1 Arquitectura

El sistema móvil ha sido desarrollado usando un patrón derivado del modelo Vista Controlador llamado MVP (Model View Presenter). El cual entre sus principales beneficios permite separar la capa de presentación de la lógica de la misma, de tal forma que todo lo relacionado con cómo funciona la interfaz queda separado del cómo representarlo en pantalla

El desarrollo del aplicativo se ha realizado en Android Studio usando Java JDK 1.7 y SQLiteOpenHelper como el ORM para el manejo de la base de datos. En la figura 4.14 se muestra el diagrama de clases utilizado para el desarrollo.

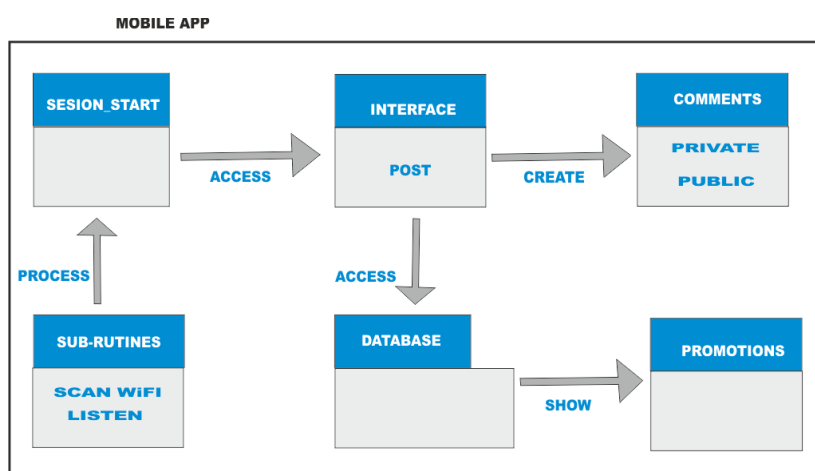


Figura 4. 14. Diagrama de clases del sistema móvil.

Android Studio es un entorno de desarrollo integrado, basado en IntelliJ IDEA de la compañía JetBrains, que proporciona varias mejoras con respecto al plugin ADT (Android Developer Tools) para Eclipse. Android Studio está programado en Java y es multiplataforma.

Ahora bien, es el entorno exclusivo para realizar programas para dispositivos Android, según lo estipuló Google, debido a que proporciona un mayor control sobre el proceso de producción.

Según (James Revelo, 2015). Un proyecto en Android utilizando el MVP o capas tiene la siguiente estructura:

**Conexión:** Se utiliza la librería volley para realizar peticiones HTTP. Volley es una librería desarrollada por Google para optimizar el envío de peticiones Http desde las aplicaciones Android hacia servidores externos. Este componente actúa como una interfaz de alto nivel, liberando al programador de la administración de hilos y procesos tediosos de parsing, para permitir publicar fácilmente resultados en el hilo principal.

**Datos:** SQLite posee acá todos los modelos de cada tabla. Todos son derivados de la clase principal DBHelper, el cual es el encargado de manejar el ORM (Object Relational Mapping) de la base de datos. En esta sección se declaran además variables que pueden ser utilizadas en todo el proyecto.

**Fragment:** En esta sección se maneja la lógica separada para cargar las vistas y paneles. Además el Recicler view es un nuevo componente para cargar los datos en el layout holder.

**Lógica:** Se encarga de formatear los objetos JSON para ser utilizados en la petición HTTP por medio de la librería volley.

**Objetos:** Son todos los modelos GET y SET.

**Servicio:** En esta sección se encuentra gran parte de la magia del proyecto en sí, ya que se encuentra los procesos para conectarse a la red, y el pedir nuevos mensajes. Además del proceso de escucha para agregar a un móvil a un grupo de usuarios utilizando los SMS.

Estos servicios están destinados a recibir y responder ante eventos globales generados por el sistema, como un aviso de batería baja, un SMS recibido, un SMS enviado. Además deben responder a eventos producidos por otras aplicaciones.

**Layout:** En esta sección se definen los layouts o contenedores de la aplicación.

**Menu:** Es el menú que poseen los activities y los botones.

**Mipmap:** Carpeta de imágenes a usar en el proyecto.

**Values:** Contiene el formato de las letras a utilizar en el proyecto.

**Gradle:** Gestor de paquetes, es lo más reciente en el manejo de librerías Android.

La siguiente figura muestra el modelo Entidad Relación utilizado en SQLite.



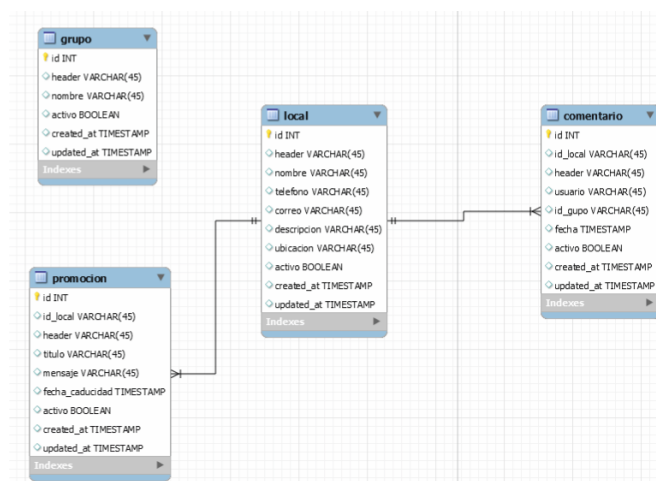


Figura 4. 15. Modelo Entidad Relación de la base de datos del sistema móvil.

Adicionalmente, en el diseño se destacan algunos permisos necesarios en el archivo manifiesto para Android Studio. Éstos son necesarios para poder ejecutar las funciones que posee el proyecto.

El método que se muestra en la figura 4.16 utiliza WiFiManager, una librería de Android para la conexión a puntos de acceso inalámbricos. Este método efectúa una búsqueda entre los SSID que están en el rango de acceso, hasta encontrar a Mora\_AP, la cual es el nombre que posee el SSID del Punto de Acceso. Este servicio se ejecuta automáticamente cada cierto tiempo, lo cual es propicio para el envío de mensajes entre los usuarios conectados.

Además en este evento se realiza las peticiones HTTP al servicio en la Raspberry, si se validó que se encuentra conectado con la WiFi Mora\_AP.

```

@Override
public void onReceive(final Context context, Intent intent) {
    WifiManager wifiManager =
(WifiManager)context.getSystemService(Context.WIFI_SERVICE);
    WifiInfo wifiInfo = wifiManager.getConnectionInfo();

    List<ScanResult> results = wifiManager.getScanResults();
    String networkSSID = "Mora_AP";

    for( ScanResult result : results ) {
        if (result.SSID.equals(networkSSID)) {
            if (!wifiInfo.getSSID().equals("\\" + networkSSID + "\\")) {
                WifiConfiguration conf = new WifiConfiguration();
                conf.SSID = "\\" + networkSSID + "\\";
                conf.allowedKeyManagement.set(WifiConfiguration.KeyMgmt.NONE);
                wifiManager.addNetwork(conf);
                List<WifiConfiguration> list = wifiManager.getConfiguredNetworks();
            }
        }
    }
}
  
```

```
for (WifiConfiguration i : list) {  
    if (i.SSID != null && i.SSID.equals("\"" + networkSSID + "\"")) {  
        wifiManager.disconnect();  
        wifiManager.enableNetwork(i.networkId, true);  
        wifiManager.reconnect();  
    }  
}
```

Figura 4. 16. Búsqueda de SSID Mora\_AP.

### 4.3.2 Interfaz Sistema Móvil

La interfaz se ha diseñado con los últimos patrones de diseño Material Design de Google, generando pantallas bastante atractivas y fáciles de usar.

Luego de instalar el programa e ingresar al mismo, se solicita un nombre de usuario para identificar al cliente cuando realice los comentarios o al pertenecer a algún grupo.

A continuación se muestra diferentes capturas de la aplicación, y se procede a explicar su funcionamiento:

La aplicación envía notificaciones una vez que se ha conectado al WiFi y se haya efectuado el proceso de sincronización de las bases de datos del cliente y el punto de acceso. Una vez que el teléfono recibe la notificación, se muestra en la aplicación todas las promociones. Las más recientes se observan de primero y en otro color. La aplicación no tiene límite de anuncios, simplemente las promociones ya caducadas no se vuelven a mostrar en el aplicativo. La figura 4.17 muestra un ejemplo de la funcionalidad de la aplicación en este sentido.



Figura 4. 17. Promociones listadas en la aplicación CrowdPi.

Al darle clic sobre la promoción que se desea observar se va a mirar con mayor detenimiento la información del local. Esto se puede observar en la siguiente figura.

- (1) En esta sección de la pantalla se observa los datos más relevantes del local comercial que emite la promoción. Lo cual permite al cliente comunicarse con el encargado del negocio por si existiera alguna duda o necesidad.
- (2) Al tocar este botón, se puede generar comentarios acerca de la promoción o servicio brindado. En la figura 4.19 se puede observar con mayor detalle cómo se efectúa un comentario en la aplicación.

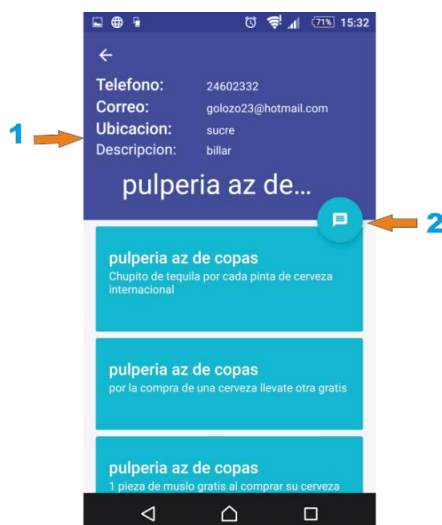


Figura 4. 18. Información del local comercial en la aplicación CrowdPi.

Parte de la importancia que posee la aplicación es la posibilidad de que el cliente pueda emitir su comentario y que la gerencia del local comercial pueda retroalimentarse acerca de la atención que brindan, la calidad del productos que ofrecen, si el precio parece justo entre otros. El cliente tiene la posibilidad de escoger si realiza los comentarios al público en general o enviarlos a algún grupo en específico al que el cliente pertenezca. La siguiente figura expone los componentes de la aplicación para realizar dicha tarea.

- (1) Estos radio buttons permiten al usuario de la aplicación escoger si desea que su comentario se muestra a todos los clientes en general o a un grupo de amigos en específico.
- (2) En caso de que se desee emitir un criterio o comentario de manera privada, la aplicación lista todos los grupos a los que pertenece el usuario. Únicamente se puede escoger un grupo para realizar el comentario.

- (3) En esta parte de la aplicación, se detalla el cuerpo del mensaje. La información que se desea comunicar.
- (4) Al tocar esa flecha, se publica el comentario en el servidor, por lo que en cuestión de segundos, los usuarios autorizados pueden observarlo, luego de que la aplicación ejecute la sincronización con el punto de acceso.



Figura 4. 19. Generar un comentario en la aplicación CrowdPi.

Otro componente importante que posee la aplicación es la posibilidad de crear grupos privados de usuarios. Lo cual permite a los usuarios comunicar sus opiniones o comentarios a un grupo selecto de clientes, con los que posiblemente conozco o posea alguna relación previa.

- (1) El proceso de crear un grupo es muy sencillo, nada más se debe digitar un asunto. Al terminar el cliente que creó el grupo ya pertenece al mismo.
- (2) Lista los diferentes grupos a los que el cliente pertenece. Se puede seleccionar alguno para poder ingresarle usuarios.
- (3) El usuario de la aplicación puede ingresar cualquier usuario al grupo creado. Para lo cual debe tocar el botón que se muestra numerado en esta pantalla.
- (4) Se ingresa el número del móvil que ya tenga instalado la aplicación CrowdPi.
- (5) Agregamos a la persona al grupo recién creado.
- (6) El cliente puede cuando lo desee, salir de un grupo de usuario pulsando el botón rojo.

(7) Este es el mensaje que recibe el móvil una vez que ha sido ingresado el usuario a un grupo en específico.

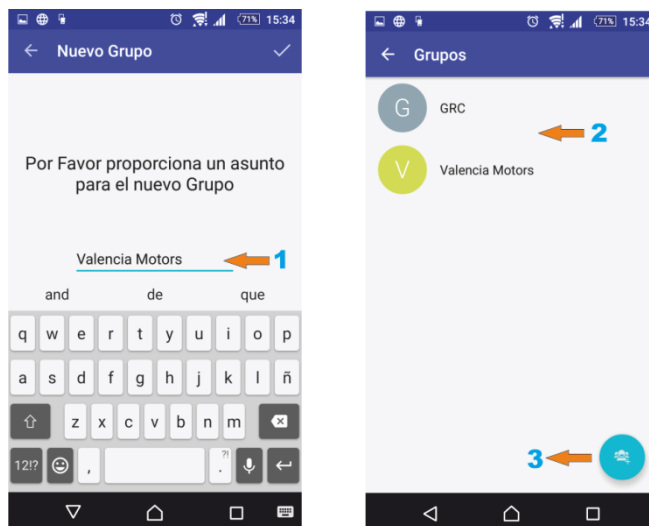


Figura 4. 20. Crear Grupos de usuarios en la aplicación CrowdPi.

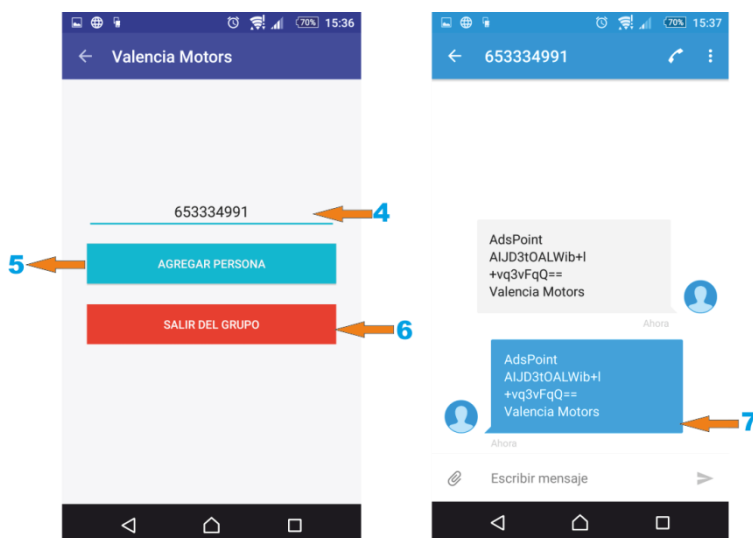


Figura 4. 21. Ingresar a un grupo de Usuarios en la aplicación CrowdPi.

#### 4.4 Store-Scripts

En el dispositivo Raspberry Pi se encuentran varios scripts PHP ejecutándose, los cuales se encargan de la sincronización de las bases de datos de los clientes y del punto de acceso. En el Anexo A, se comentará los principales scripts utilizados en este proyecto.

## Capítulo V: Pruebas

Antes de comenzar las pruebas, se realizaron algunas mejoras a la configuración del punto de acceso (WiFi), con el fin de mejorar sus prestaciones, entre las cuales tenemos:

1. Se cambió el canal utilizado para evitar interferencias con otras señales de WiFi, para lo cual primero se analizó los canales usando el aplicativo WiFi Analyzer, luego se modificó el archivo `hostapd`, utilizando el canal menos utilizado, el cual según los resultados obtenidos era el canal 8.
2. Se modificó la forma que tiene el adaptador de gestionar la energía que recibe de la Raspberry Pi, para asegurar una conexión WiFi estable y de calidad. Al obligar a la tarjeta WiFi a consumir la energía que sea necesaria para tal fin, lo cual no supondría ningún problema al estar conectado al conector AC.
3. Se verificó la compatibilidad del adaptador WiFi para ser utilizado como punto de acceso en la siguiente página: ([http://elinux.org/RPi\\_USB\\_Wi-Fi\\_Adapters](http://elinux.org/RPi_USB_Wi-Fi_Adapters)), para evitar problemas de funcionamiento.

### 5. 1. Descripción de las pruebas

En la presente sección se procede a evaluar el desempeño de tres aspectos que conforman la plataforma desarrollada.

1. Rendimiento de la red WiFi utilizando el Raspberry Pi como punto de acceso.
2. Rendimiento de la plataforma utilizando datos de prueba.
3. Consumo energético de la aplicación.

#### Rendimiento de la red WiFi utilizando el Raspberry Pi como punto de acceso

Para comprobar el rendimiento de la red se realizaron los siguientes pasos:

- Se instala `Iperf` en el dispositivo Raspberry Pi (Punto de Acceso WiFi).
- Se instala `Iperf` para Android en el móvil en que se realizan las pruebas.
- Se procede a colocar el punto de acceso a 5 metros del dispositivo móvil. Luego se procede a ejecutar una serie de comandos para evaluar el desempeño de la red utilizando 10 clientes en paralelo. Las pruebas se ejecutaron en los protocolos TCP y UDP. Luego se agregan 15 metros a la distancia original y se ejecutan nuevamente los comandos para obtener los nuevos valores.
- Los resultados obtenidos son almacenados en ficheros (.log).

La siguiente tabla indica los parámetros por defecto con los cuales se ejecuta el programa Iperf, entre los cuales tenemos el tamaño de los datos prueba (KiloByte), el intervalo de tiempo en que se va a transmitir (segundos) y la distancia del cliente con respecto al punto de acceso (metros). Todos estos aspectos conforman los experimentos que se utilizaron para medir el rendimiento de la red.

| Protocolo | Número Clientes | Tamaño de ventana/Buffer (KB) | Intervalo (segundos) | Distancia |
|-----------|-----------------|-------------------------------|----------------------|-----------|
| TCP       | 10              | 23.8 tamaño de ventana        | 0-10                 | 5 metros  |
| TCP       | 10              | 23.8 tamaño de ventana        | 0-12.5               | 20 metros |
| UDP       | 10              | 160 tamaño buffer             | 0-10                 | 5 metros  |
| UDP       | 10              | 160 tamaño buffer             | 0-12.5               | 20 metros |

Tabla 4. Tamaño de los datos utilizada para medir el rendimiento de la red.

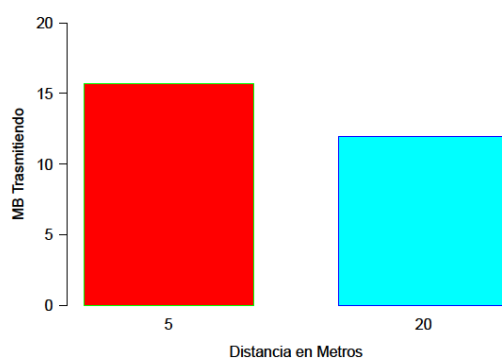
La siguiente tabla muestra los comandos utilizados en el punto de acceso y en el cliente Android para generar tráfico en la red y así poder observar el comportamiento del dispositivo Raspberry Pi al tener varios clientes conectados y generando peticiones.

| Protocolo | Dispositivo Ejecuta | Comando                       | Objetivo   |
|-----------|---------------------|-------------------------------|--|
| TCP       | Punto de Acceso     | Iperf -s                      | Escuchar peticiones TCP de los clientes                        |
| TCP       | Android             | Iperf -c 192.168.42.1 P 10    | 10 clientes en paralelo envían paquetes TCP al punto de acceso |
| UDP       | Punto de Acceso     | Iperf -s -u                   | Escuchar peticiones UDP de los clientes                        |
| UDP       | Android             | Iperf -c 192.168.42.1 -u P 10 | 10 clientes en paralelo envían paquetes UDP al punto de acceso |

Tabla 5. Comandos utilizados para evaluar el desempeño de la Red.

Lo primero que se desea comprobar es la tasa de transferencia que posee el dispositivo Raspberry Pi sin utilizar la aplicación CrowdPi. Para esto se ideó un experimento que ejecute 50 muestras con 10 clientes en paralelo a una distancia de 5 metros y 50 muestras de clientes en paralelo a 20 metros y con obstáculos. Utilizando los valores de la tabla 4. Con los resultados obtenidos se puede comprobar las prestaciones que posee la red al tener varios usuarios conectados. Cabe destacar que los resultados son reportados por el punto de acceso.

Como se puede analizar en la figura 5.1, la tasa de transferencia aumenta en 3.73 MB al realizar la prueba con 5 metros de distancia del punto de acceso. Esto se debe a que la distancia juega un papel muy importante en la atenuación de una señal WiFi. Además en las pruebas realizadas a 20 metros se encontraba un obstáculo (pared). Que aumenta también la atenuación de la señal y empeora la velocidad de transferencia.



**Figura 5. 1. Tasa de Transferencia usando el Raspberry Pi como Punto de Acceso.**

Se realizó un segundo experimento para medir el ancho de banda utilizando Iperf en el cliente y en el punto de acceso, para lo cual se utilizó los datos de la tabla 6. Al igual que en el experimento anterior se coloca el Raspberry Pi a dos distancias diferentes del móvil que hace las solicitudes. Esta vez el punto de acceso es colocado en rango de visión, por lo que no hay obstáculos que generen interferencias.

Los comandos utilizados en el móvil son “Iperf –c 192.168.42.1” para los experimentos con un único cliente y “Iperf –c 192.168.42.1 P 10”, para indicar que se van a enviar 10 solicitudes paralelas al servidor. Todas estas solicitudes son generadas sobre tráfico IP, similar a los JSON utilizados en la aplicación. Es importante aclarar que la dirección IP debe ser la misma del punto de acceso.

| Configuración | Distancia Aproximada (metros) | Cantidad de Usuarios | Intervalo (segundos) |
|---------------|-------------------------------|----------------------|----------------------|
| <b>A</b>      | 20                            | 10                   | 10-12.2              |
| <b>B</b>      | 20                            | 1                    | 10.2                 |
| <b>C</b>      | 5                             | 10                   | 10-12.2              |
| <b>D</b>      | 5                             | 1                    | 10.2                 |

**Tabla 6. Configuración utilizada en los experimentos de Ancho de Banda.**



Los principales aspectos que se analizan luego de ver los resultados obtenidos del programa lperf son:

- Las pruebas con 10 clientes en paralelo poseen un mayor promedio de entrega, que las realizadas con un único cliente. Pero también las pruebas en paralelo poseen un intervalo de tiempo superior para la entrega de los paquetes.
- Además se puede observar que entre menos distancia se posea, la tasa de transferencia aumenta, debido a que disminuye la atenuación.

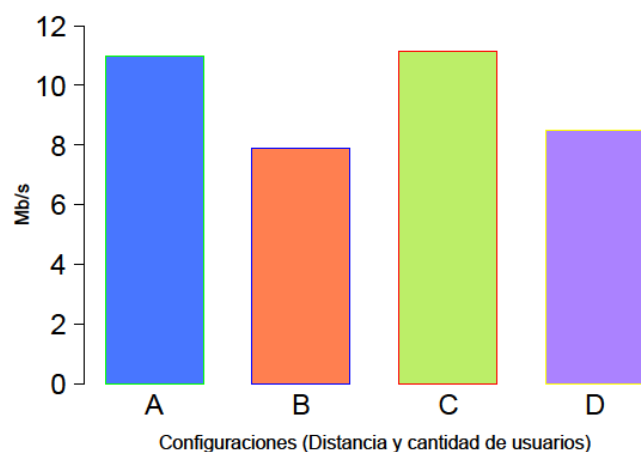


Figura 5. 2. Ancho de banda con varias configuraciones del Punto de Acceso.

Posteriormente en un tercer experimento, se realizaron pruebas utilizando el protocolo UDP. El cual es necesario cuando se manejan colas de solicitudes, las cuales son guardadas en el servidor NGINX utilizando comandos AMQP.

Para obtener los resultados, se realizaron pruebas sobre el retraso (jitter) y la pérdida de paquetes se procedió a realizar los siguientes pasos:

- Se ejecuta el comando para escuchar solicitudes UDP en el servidor.
- Se procede a colocar el punto de acceso a 5 metros del dispositivo móvil. (ver tabla 7). Luego se procede a ejecutar una serie de comandos para evaluar el desempeño de la red utilizando 10 clientes en paralelo.
- Se procede a colocar el móvil la nueva distancia y se ejecuta nuevamente comando lperf para Android en el móvil para solicitar los datos al punto de acceso.
- Los resultados obtenidos son almacenados en ficheros (.log).

Los datos de la configuración del experimento se pueden observar en la tabla 7.

| Configuración | Distancia Aproximada (metros) | Cantidad de Usuarios |
|---------------|-------------------------------|----------------------|
| A             | 20                            | 1                    |
| B             | 5                             | 1                    |
| C             | 20                            | 10                   |
| D             | 5                             | 10                   |

Tabla 7. Configuración utilizada en los experimentos de Jitter.

Los resultados obtenidos demuestran que no existe pérdida de datos en el servidor. (0%).

El segundo aspecto a analizar es la duración del jitter. El estudio realizado demuestra un crecimiento del mismo dependiendo del aumento de la distancia en que se encuentre el cliente con respecto al punto de acceso. Además cuando se envían varios paquetes simultáneos, el Raspberry Pi, debe hacer uso de las colas para almacenar los datos, para evitar los errores y la pérdida de datos, lo cual aumenta el jitter. Dato que se demuestra en las configuraciones "C" y "D" de la figura 5.3.

El presente experimento puede ser observado en las siguientes figuras:

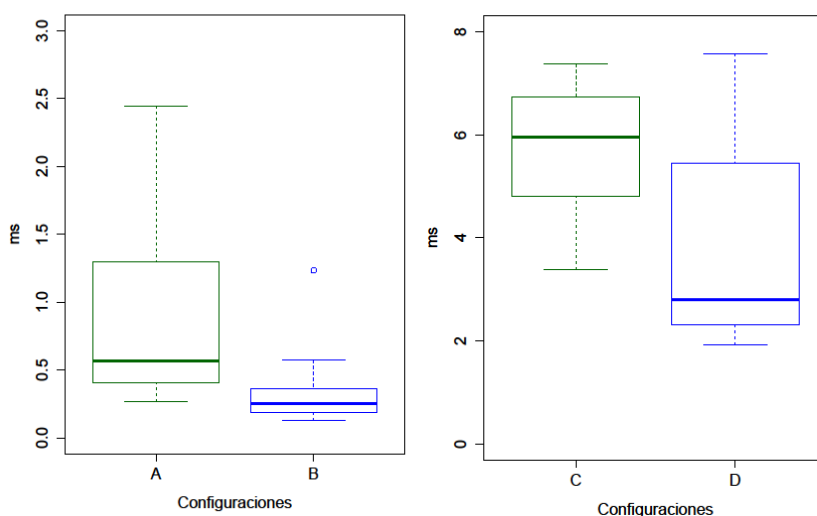


Figura 5. 3. Jitter obtenido en la red utilizando varias configuraciones.

Para realizar el último experimento a la red se instaló en el móvil una aplicación llamada "Ekahau". La cual permite entre otras cosas, localizar todos los puntos de acceso en un rango cercano, la tasa de transferencia, el retardo y la pérdida de paquetes. Todas estas mediciones se realizaron en torno a la distancia que se encontraba el punto de acceso del móvil con la aplicación instalada.

Las mediciones se realizaron en dos lugares diferentes, dentro del edificio 1G del campus de Vera y en los alrededores. El mismo programa permite tomar una captura de pantalla con la intensidad de la señal del punto de acceso que se está evaluando. Entre más oscuro sea el color verde, la señal es más fuerte, según se puede observar en la siguiente figura.

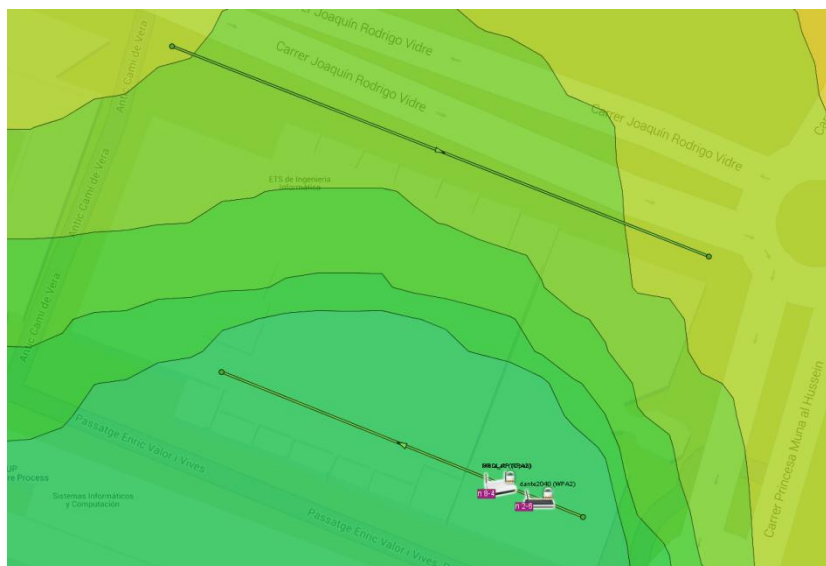


Figura 5. 4. Mapa de Intensidad de señal del punto de acceso Mora\_AP

A continuación se muestran varias capturas de pantalla con los datos obtenidos del programa ekahau durante las mediciones que se hicieron en el sitio.

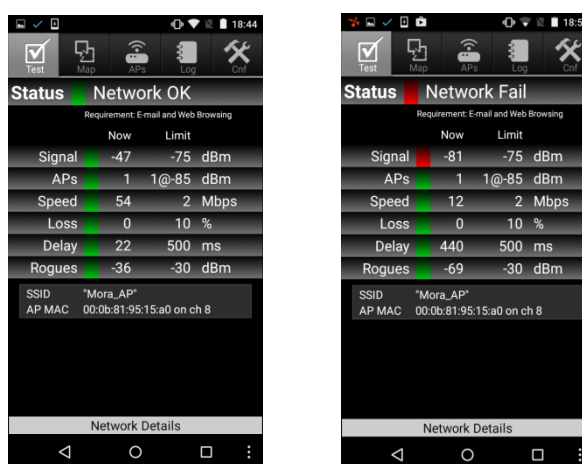


Figura 5. 5. Pruebas realizadas al punto de acceso con aplicación Ekahau.

La tabla 8 muestra los resultados de las diferentes pruebas que se realizaron tanto dentro del edificio como en las afueras del mismo.

|                | Señal (dBm) | Velocidad (Mbps) | Pérdida % | Retraso (ms) | Metros Aproximados |
|----------------|-------------|------------------|-----------|--------------|--------------------|
| <b>Adentro</b> | 47          | 54               | 0         | 22           | 5                  |
|                | 49          | 54               | 0         | 13           | 10                 |
|                | 59          | 54               | 0         | 21           | 15                 |
|                | 66          | 54               | 0         | 54           | 20                 |
|                | 62          | 54               | 0         | 51           | 25                 |
|                | 68          | 54               | 0         | 318          | 30                 |
|                | 81          | 12               | 0         | 440          | 35                 |
|                | 84          | 24               | 0         | 280          | 40                 |
|                | 84          | 18               | 0         | 260          | 45                 |
|                | 87          | 12               | 0         | 839          | 50                 |
| <b>Afuera</b>  | 71          | 54               | 0         | 115          | 5                  |
|                | 68          | 54               | 0         | 77           | 10                 |
|                | 63          | 54               | 0         | 115          | 15                 |
|                | 61          | 54               | 0         | 16           | 20                 |
|                | 45          | 54               | 0         | 13           | 25                 |
|                | 74          | 36               | 0         | 133          | 30                 |
|                | 73          | 2                | 0         | 106          | 45                 |
|                | 83          | 12               | 0         | 77           | 40                 |
|                | 81          | 2                | 0         | 473          | 45                 |
|                | 87          | 2                | 0         | 838          | 50                 |

**Tabla 8. Resultados obtenidos con la aplicación Ekahau.**

Los resultados obtenidos luego de terminar las pruebas nos indican que tanto fuera como dentro del edificio no hubo pérdida de datos. Además la velocidad y el retraso de los paquetes van ligados a la pérdida de la señal, esto debido a la atenuación que se sufre por el aumento de la distancia o por pérdida de línea de visión. (Obstáculos).

Dentro del edificio, el cambio de punto de acceso ocurre luego de que el móvil se aleja por más de 40 metros, con varios obstáculos que imposibilitan obtener la señal adecuadamente.

Los resultados obtenidos en las afueras del edificio sugieren que el cambio de punto de acceso ocurrió luego de tener una potencia de señal inferior a los -81 dBm, punto donde las prestaciones se degradan rápidamente.

### Rendimiento de la plataforma utilizando datos de prueba.

El primer experimento que se llevó a cabo consistía en medir la duración de los mensajes en llegar al punto de acceso. (ms). Se dispuso a crear una bandera a la hora de enviarlos desde el móvil y un ACK en el punto de acceso una vez que habían llegado sin errores.

- La distancia del móvil con respecto al punto de acceso son 20 metros.
- Para establecer la duración se dispuso a medir la duración de 50 mensajes, divididos en segmentos de 5 mensajes, 20 mensajes y 25 mensajes.
- Luego se procedió a realizar la misma medición pero colocando el punto de acceso cerca de otras antenas para generar ruido a la señal.
- Los datos obtenidos se guardan en un fichero .log.

Luego de realizar el análisis de los datos se pudo comprobar una similitud en el tiempo de entrega de ambas muestras. Además el promedio obtenido de sumar los primeros dos segmentos. (5 mensajes y 20 mensajes). Es muy similar al promedio de enviar 25 mensajes.

Además como ocurrió en las pruebas realizadas a la red anteriormente, el ruido disminuye a la medida que el punto de acceso se encuentra alejado de otros puntos de acceso que pueden ocasionar interferencias. Como se puede observar en la figura 5.5.

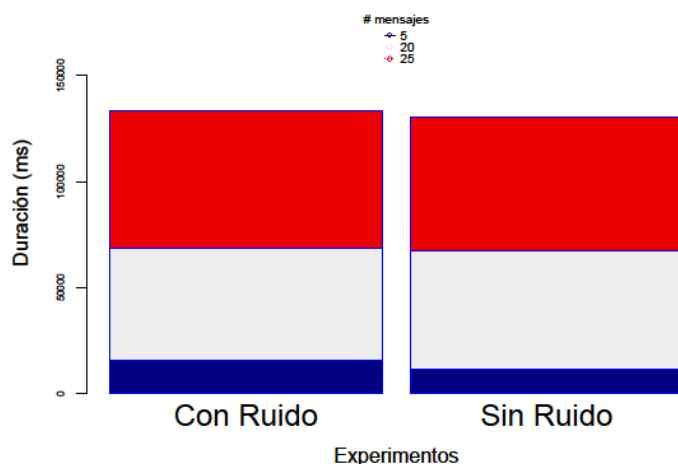


Figura 5. 6. Retardo de Mensajes enviados al Servidor.

El segundo experimento consistía en evaluar el tiempo que le tomaba al cliente Android en volver a refrescar la conexión con el punto de acceso, este proceso es propio del sistema

Android y utiliza la librería WiFiManager. Se procede a colocar una bandera posterior al descubrimiento de la red "Mora\_AP".

- Para el experimento se seleccionó 50 diferentes registros de conexiones con el servidor, guardados en un fichero .log.
- Luego se procedió a realizar un promedio de conexiones de clientes utilizando intervalos de 30 segundos.

Es importante destacar que los resultados obtenidos indican que casi la mitad de los clientes duran de 30 a 60 segundos en refrescar su conexión con el servidor, por lo que la obtención de nuevas promociones o comentarios es sumamente rápido. Esto se puede observar en la figura 5.7.

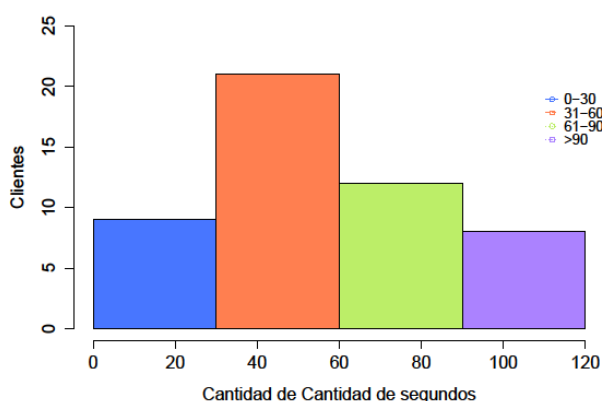


Figura 5. 7. Tasa de Refrescamiento de la WiFi Mora\_AP.

### Consumo energético de la aplicación.

Otro punto que es importante destacar es el consumo energético que posee la aplicación desarrollada, para lo cual se procede a crear un experimento utilizando un dispositivo Android totalmente cargado con una batería de capacidad de 2100 miliamperios (mA). Además se definieron tres etapas de medición, la primera de ellas es la base, donde el dispositivo posee la aplicación instalada pero sin ejecutar, esta demostrará cuanto es el consumo normal del teléfono. La siguiente etapa es Ejecutando, en la cual se procede a abrir la aplicación pero sin conectarse o descargar contenido del punto de acceso. Por último, la etapa de conexión, donde se procede a tomar datos mientras el cliente recibe y envía información al punto de acceso.

El propósito del primer experimento es medir el consumo energético. Para lo cual se procede a tomar muestras durante una hora, (3600 segundos) en cada una de las etapas, para así

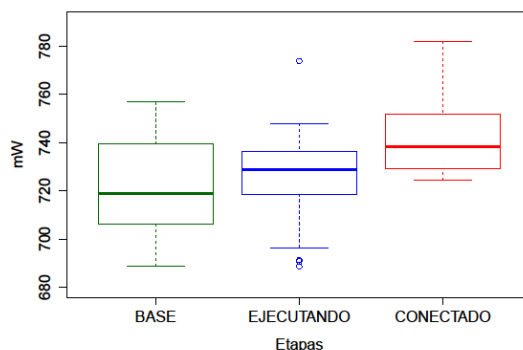
conseguir cuánta corriente (mAh) es consumida por el dispositivo móvil en cada fase. Luego de observar los resultados se puede determinar que el consumo de la aplicación es de 2,66 mAh. Mientras que el total de consumo de miliamperios a la hora de transmitir y recibir datos es de casi los 4,28 mAh.

Según se explica en el documento. (Power Consumption Evaluation in Vehicular Opportunistic Networks, 2014). Para obtener el poder necesario, se multiplica por el valor nominal de voltaje de la batería promedio (3.8 voltios). Los datos obtenidos se encuentran reflejados en la siguiente tabla:

| No | Etapas     | Corriente mA (hora) | Poder mW |
|----|------------|---------------------|----------|
| 1  | BASE       | 188.76              | 717.28   |
| 2  | EJECUTANDO | 191.42              | 727.39   |
| 3  | CONEXION   | 195.70              | 743.66   |

**Tabla 9. Consumo energético clasificado por etapas.**

En la figura 5.8 se puede demostrar los resultados obtenidos. Donde el consumo de mW es más elevado en la etapa de conexión, ya que en esta fase se reciben promociones y mensajes constantemente.



**Figura 5. 8. Mili-watts hora usados por la aplicación**

Para realizar el segundo experimento, se va a utilizar la tabla 9. Con esos datos se procede a determinar la vida útil de la batería del móvil, y la estimación del valor óptimo que el dispositivo debe conectarse al punto de acceso para el mejor rendimiento de la batería. Para este experimento se envían mensajes TCP a un móvil Android con una batería de 2100 mA.

Para obtener las diferentes horas del consumo se procede a dividir el consumo de mAh por el total que posee el móvil con el que se realizó la prueba. El resultado óptimo demuestra que conectándose cada 17 minutos al punto de acceso se puede obtener mejor consumo de la batería. Los resultados se pueden apreciar en la siguiente tabla:

| Etapa      | Horas |
|------------|-------|
| BASE       | 11,12 |
| EJECUTANDO | 10,97 |
| CONEXION   | 10,60 |
| ÓPTIMO     | 10,74 |

Tabla 10. Consumo energético en horas.

La figura 5.9 muestra los datos que se obtuvieron del experimento anterior. Actualmente la aplicación está desarrollada para que se conecte más frecuentemente al punto de acceso, pensando en la satisfacción del cliente al obtener la información en un tiempo prudente y luego proceder a la toma de decisiones o compras a realizar.

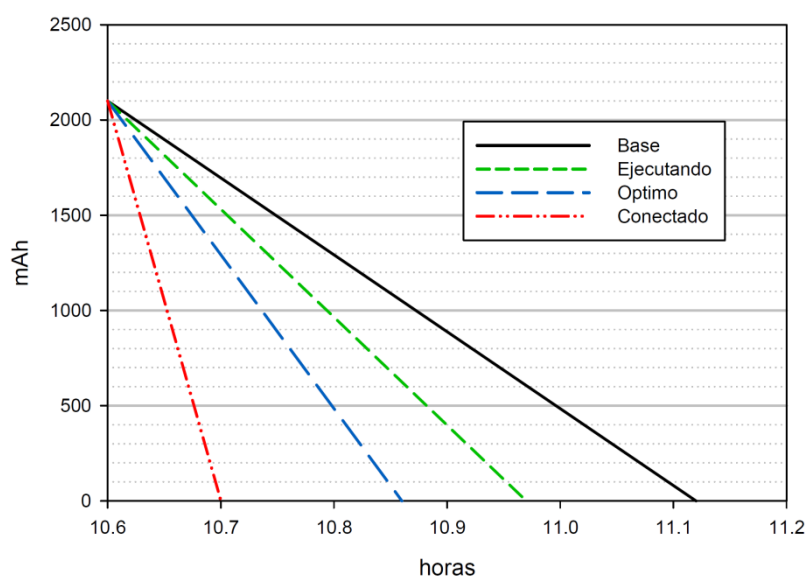


Figura 5. 9. Duración de horas de la aplicación en las diferentes Etapas.



## Capítulo VI: Conclusiones y Trabajo futuro

### Conclusiones

A modo de conclusión de este proyecto se puede indicar que la aplicación utiliza dispositivos de bajo coste, que permiten a locales comerciales difundir promociones de manera ágil e intuitiva, sin generar más costes de transmisión a internet, a los clientes que la tienen la aplicación instalada les permite obtener promociones de los locales comerciales cercanos a su ubicación y difundir comentarios de los servicios que utilizó en muy poco tiempo.

Otro aspecto a resaltar son las tecnologías utilizadas, los cuales por su novedad poseen gran cantidad de soporte y documentación en internet, lo que haría más sencillo una mejora o nueva integración a la plataforma.

Los procedimientos que se ejecutan en el Raspberry Pi, permiten la sincronización de los datos de manera automática, sin que el administrador deba ejecutar algún proceso.

Para finalizar con lo propuesto se quiere indicar que el consumo de batería es reducido, permitiendo al usuario usar por varias horas la aplicación sin que esto signifique un gasto significativo de la batería del móvil.

### Trabajo futuro

Para un trabajo a futuro se desea mejorar la aplicación permitiendo a los usuarios compartir imágenes.

Por otra parte se desea aplicar la geolocalización para descubrir los locales comerciales más cercanos a la ubicación de los clientes de manera automática.

Para terminar, se quiere utilizar redes oportunísticas con el fin de mejorar la comunicación de promociones de locales que se encuentren lejos geográficamente y ofrecerles igualdad de condiciones que sus competidores de las grandes urbes.

## Bibliografía

- Cambridge University Press. 2015.** <http://dictionary.cambridge.org>. [En línea] 2015. [Citado el: 28 de octubre de 2015.] <http://dictionary.cambridge.org/dictionary/english/crowdsourcing>.
- Aalto University, Spacetime Networks. 2012-2015.** <http://www.liberouter.mobi/>. [En línea] 2012-2015. [Citado el: 29 de 10 de 2015.] <http://www.liberouter.mobi/>.
- colaborar-en-proyectos-de-crowsourcing, aplicaciones-para. 2013.** [hipertextual.com](http://hipertextual.com). [En línea] 08 de 2013. [Citado el: 28 de 10 de 2015.] <http://hipertextual.com/archivo/2013/08/aplicaciones-para-colaborar-en-proyectos-de-crowdsourcing/>.
- Crowdsourcing with smartphones.* **Georgios Chatzamilioudis, Andreas Konstantinidis, Christos Lauodias, Demetrios Zeilanipour-Yatzi. 2012.** 1089-7801, Cypros : IEEE Internet Computing, september de 2012, IEEE Computer Society.
- definicion.de. 2008-2015.** Definicion.de. [En línea] 2008-2015. [Citado el: 30 de 10 de 2015.] <http://definicion.de/wifi/#ixzz3q4DjdULq>.
- Droideando. 2011.** Droideando. [En línea] 02 de 2011. [Citado el: 30 de 10 de 2015.] <http://droideando.blogspot.com.es/2011/02/ciclo-de-vida-de-una-aplicacion-android.html>.
- Haggle: a Networking Architecture Designed Around Mobile Users.* **James Scott, Jon Crowcroft, Pan Hui, Christophe Diot. 2006.** Les Menuires : HAL, 2006, págs. 78-86.
- Information-centric Networking: The evolution from circuit to packets to content.*
- Kurose, Jim. 2014.** Massachusetts : Elsevier, 2014, Computer Networks, págs. 112-120.
- James Revelo. 2015.** [hermosaprogramacion](http://www.hermosaprogramacion.com). [En línea] 2015. [Citado el: 9 de 11 de 2015.] <http://www.hermosaprogramacion.com/2015/02/android-volley-peticiones-http/>.
- Liberouter: Towards Autonomous Neighborhood Networking.* **Teemu Karkkainen, Jorg Ott. 2014.** 978-1-4799-4937-4, Espoo : IEEE, 2014.
- Managing the development of large software system.* **Royce, Wiston. 1970.**
- Measurement and Modeling of User Transitioning Among Networks.* **Sookhyun Yang, Jim Kurose, Simon Heinlicher, Arun Venkatarami. 2014.** UM-CS-2014-023, Amherst : University of Massachusetts, 2014.
- Metodología para la Monitorización de Ozono en Valencia mediante Sensores de gama baja.* **Alvear, Óscar, y otros. 2015.** 2015, Jornadas de Computación Empotrada, Vol. VI.
- OperGarden. 2015.** <http://opengarden.com/>. [En línea] 2015. [Citado el: 30 de 10 de 2015.] <http://opengarden.com/>.
- Power Consumption Evaluation in Vehicular Opportunistic Networks.* **Jorge Herrera Tapia, Pietro Manzoni, Enrique Hernandez, Carlos Calafate, Juan Carlos Cano. 2014.** Valencia : UPV, 2014.
- Programador, Báculo del Mono. 2012.** Proyecto Simio. [En línea] 2012. [Citado el: 29 de 10 de 2015.] <http://www.proyectosimio.com/es/programacion-android-ciclo-de-vida-de-una-app/>.
- Revelo, James. 2014.** [Hermosaprogramacion.com](http://www.hermosaprogramacion.com). [En línea] 08 de 2014. [Citado el: 30 de 10 de 2015.] <http://www.hermosaprogramacion.com/2014/08/aprendiendo-la-arquitectura-de-android/>.
- RStudio.** RStudio Products. [En línea] [Citado el: 12 de 07 de 2015.] <https://www.rstudio.com/products/rstudio/>.
- Toward an Information-centric Delay-Tolerant Network.* **Gareth Tyson, Jon Bigham, Eliane Bodanese. 2012.** London : University of London, 2012.

*WiFi-Opp: Ad-Hoc-Less Opportunistic Networking*. **Sasha Trifunovic, Bernhard Distl, Dominic Schatzmann, Franck Legendre**. 2011. 978-1-4503-0722-2, Las Vegas, Nevada : ACM, 2011.

**Wikipedia**. 2015. Wikipedia.com. [En línea] 25 de 10 de 2015. [Citado el: 30 de 10 de 2015.]

[https://es.Wikipedia.org/wiki/Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_software](https://es.Wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software).

**Wikipedia**. 2015. Wikipedia.com. [En línea] 25 de 10 de 2015. [Citado el: 29 de 10 de 2015.] [https://es.Wikipedia.org/wiki/Servidor\\_web](https://es.Wikipedia.org/wiki/Servidor_web).

**youngmarketing.co**. 2014. Beneficios del crowdsourcing para mejorar las ciudades. *Hermida, Camilo Alicia Ortega*. [En línea] 29 de setiembre de 2014. [Citado el: 12 de 08 de 2014.] <http://www.youngmarketing.co/crowdsourcing-y-ciudades-una-mezcla-poderosa-para-el-futuro/#ixzz3ojDyslks>.

**Zamorano, Esteban**. 2014. wayerless.com. [En línea] 2014. [Citado el: 30 de 10 de 2015.] <https://www.wayerless.com/2014/12/meshme-es-una-app-similar-a-firechat-para-comunicarte-sin-conexion-a-internet/>.

## ANEXO A

### STORE-SCRIPTS

Los siguientes archivos se encuentran ejecutándose en memoria en el dispositivo embebido, por lo que el administrador del local comercial no debe preocuparse por su manipulación. Los mismos realizan el trabajo de escuchar peticiones, sincronizar los datos con los clientes, creación de nuevas promociones, lista de comentarios efectuados por los usuarios del local y respaldo de la información en caso de siniestro. A continuación se detallará las funciones principales que realizan cada uno.

#### Routes.php

- Llama la clase y el método para comunicarse con el servidor web.

#### QueueController.php

- Recibe el json del cliente.
- Invoca al cliente de colas.
- Devuelve el JSON con los datos que el Android no tiene luego del proceso intercambiar realizado en el script de la página procesamiento.php.

#### RPCClient.php

- Constructor de hilos.
- Usa AMQP para encolar los datos.
- Devuelve el JSON a la página QueueController para que envíe los datos al móvil.

#### RPCServer.php

- Declaración del canal para recibir solicitudes de los clientes.
- Recibe los datos enviados por los clientes.
- Llama al proceso de sincronización.
- Devuelve los datos al RPCClient

#### Procesamiento.php

- Este proceso se encarga de sincronizar los datos del Android y el broker.
- Convierte el JSON que recibe RPCServer en un arreglo.
- Se sincronizan los datos que posee el servidor y el dispositivo móvil. Eliminando los registros repetidos.
- Guarda los datos sincronizados en la base de datos.
- Se construye el JSON que se va a devolver al cliente.

### **Database.php**

- Crea la conexión a la base de datos.

### **LocalController.php**

- Permite modificar los datos del local como teléfono, facebook, correo electrónico entre otros.
- Guarda los datos de promociones e información del local en google drive.

### **ComentarioController.php**

- Muestra la lista de comentarios realizados a las promociones que se han efectuado en un local comercial.

### **PromocionController.php**

- Permite crear, modificar y anular promociones utilizando una fecha de caducidad para evitar difundir información caducada.