Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicaciones de mensajería para dispositivos móviles basadas en comunicación punto a punto local

Trabajo Fin de Máster

Máster Universitario en Ingeniería Informática

Autor: David Fernández Delegido

Tutor: Enrique Hernández Orallo

Curso 2015/2016

Agradecimientos

Este trabajo ha sido parcialmente financiado por la Generalitat Valenciana, Direcció General d'Universitat, Estudis Superiors i Ciència, bajo la subvención AICO/2015/108, en el proyecto "Análisis de la movilidad y persistencia de la información en redes vehiculares. Aplicación a la gestión de accidentes".

Resumen

El objetivo de este trabajo es realizar un análisis del funcionamiento de la comunicación entre dispositivos móviles locales para el intercambio y difusión de mensajes. Para ello se ha diseñado una aplicación para sistemas Android cuya función básica consiste en el envío de mensajes mediante Bluetooth entre dispositivos cercanos. El sistema desarrollado se basa en el modelo clásico de cliente-servidor, en el que uno de los nodos actúa como cliente y se conecta al segundo nodo servidor para mandarle la información. Este método se ha adaptado para construir el modelo de difusión de mensaje basado en redes oportunísticas, el cual funciona de manera que cuando un usuario envía un mensaje, éste va pasando de un nodo a otro hasta que todos los dispositivos de la red consiguen tener el mensaje en cuestión. El sistema sigue las líneas de diseño de las actuales aplicaciones de mensajería más comunes. Cuenta con un sistema de grupos por el cual se organizan los mensajes de manera que el usuario pueda diferenciar las diferentes conversaciones. En estas conversaciones el usuario puede comunicarse con los otros nodos disponibles en la red en ese instante.

Palabras clave: Android, comunicación, redes oportunísticas, mensajería, Bluetooth.

Abstract

This master thesis presents an analysis of how local communication between mobile devices is made. For this, I have designed an application for Android systems whose primary function is interchanging messages between nearby devices via Bluetooth. The developed system is based on the classic client-server model, in which one node acts as client and it connects to the second server node to send the information. This method has been adapted to build the model broadcast message, which operates so that when a user sends a message, it is passed from one node to another until all network devices receive the message. The system design follows the lines of the current most common messaging applications. It has a group system by which messages are organized so that the user can differentiate between conversations. In these conversations the user can communicate with other available nodes on the network at that moment.

Keywords : Android, communication, opportunistic networks, messaging applications, Bluetooth.

Tabla de contenidos

Índice de ilustraciones

1. Introducción

Hoy en día la comunicación punto a punto local entre dispositivos es un tema que se ha hecho muy popular debido principalmente al aumento de sistemas como Arduinos o Raspberries impulsados principalmente por corrientes como el internet de las cosas. Además, los propios sistemas Android pueden aumentar la funcionalidad de estos sistemas o incluso ofrecer una nueva. La mayor parte de las veces, estos sistemas se comunican de forma local usando algún protocolo clásico como puede ser el Bluetooth o Wi-Fi Direct. En este trabajo de fin de master, para entender la comunicación entre sistemas que usan un canal de comunicación local, se ha realizado una aplicación en Android basada en el envío de mensajes usando la tecnología Bluetooth y redes oportunísticas.

1.1 Presentación del problema

Para poder elaborar una aplicación de mensajería que utilice Bluetooth como medio de difusión de sus mensajes existen diferentes técnicas que son necesarias dominar. Además, también hay que tener en cuenta las herramientas que se necesitan para implementar estas técnicas.

Para el desarrollo de la aplicación se ha seleccionado Android ya que ofrece toda la flexibilidad que necesitamos para la elaboración del proyecto. Este sistema ofrece una herramienta (IDE), llamada Android Studio basado en IntelliJ IDEA, diseñada expresamente para el desarrollo de aplicaciones en su plataforma. Por encima de este sistema se nos ofrecen más funcionalidades específicas como un sistema basado en Gradle. Además del entorno de desarrollo, Android usa Java como lenguaje de programación que nos ofrece una gran ventaja al ser una tecnología ampliamente extendida y con gran soporte.

Para dominar todas estas herramientas hay que comprenderlas primero, por eso es necesario entender que hay que llevar a cabo un estudio previo del funcionamiento de Android. Es necesario comprender cuáles son los métodos correctos para desarrollar en esta plataforma, así como entender los aspectos fundamentales de su API. Todo este proceso conlleva un tiempo, pero es igual de importante que cualquier otra tarea dedicada al proyecto. Además, hay que entender que la lógica que mueve el desarrollo en Android es muy susceptible a cambios debido a que esta plataforma está presente en infinidad de dispositivos nuevos y en constante cambio. Es por ello que se necesita un trabajo extra de seguimiento para estar al tanto de cambios o actualizaciones en su estructura o en la API.

Centrándonos en la aplicación a desarrollar, se presenta el problema como un sistema en el que se intercambia información dentro de una red local. Una posible aplicación real puede ser el despliegue de esta red dentro de casos en los que sea difícil acceder a internet de forma convencional, es decir, en situaciones como conciertos, manifestaciones o catástrofes, donde las redes normales como la telefonía móvil y Wi-Fi estarían inutilizadas. En escenarios como estos, se hace necesario poder difundir la información utilizando un enfoque de comunicación directa entre dispositivos móviles. Este sistema cuenta con una serie de dispositivos que actúan de nodos dentro de la red. El número de nodos es indeterminado por lo que no podemos confiar en un número

determinado mientras construimos la solución. Para que un nodo de la red pueda transmitir los mensajes a otro nodo, el segundo debe encontrarse cercano al dispositivo, esto es debido principalmente a la tecnología Bluetooth que requiere que los dispositivos se encuentren cerca para iniciar la transmisión de datos. Además, estos nodos pueden entrar y salir de la red a placer sin tener que decírselo a ningún otro nodo.

También se cree conveniente incluir un sistema del control de los mensajes de la aplicación. Para esto usaremos un buffer en el que controlaremos la inserción y el borrado de mensajes de forma manual, pudiendo así incluir cualquier modificación conveniente como el tamaño de buffer o el sistema de borrado de los mensajes.

Finalmente, es necesario recalcar que sobre este problema hay que realizar una aplicación en Android, y por ello también es necesario incluir una buena interfaz que soporte y presente cada una de las características comentadas. Esta aplicación debe poder presentar al usuario dos importantes características; el chat en el que se muestran los mensajes y un menú principal en el que poder navegar y visualizar los diferentes mensajes y características de la aplicación.

1.2 Objetivos del proyecto

El objetivo principal del proyecto es el estudio de la implementación de un sistema que permitiese enviar información entre dispositivos locales. Teniendo en cuenta esta idea es necesario definir unos objetivos principales que expliquen claramente la intención del problema y que nos ayuden a no desviarnos de la intención principal. Estos objetivos primarios son:

- Intercambio de información en forma de texto entre dos dispositivos o más que se encuentran físicamente en el mismo medio. Así pues, se debe formar una red local para que se produzca el intercambio de información entre los sistemas. También se debe escoger la tecnología adecuada para este fin, en nuestro caso Bluetooth.
- Difusión de la información entre dispositivos. Es un requisito que la información obtenida por uno de los nodos de la red sea difundida a aquellos dispositivos que sean accesibles por ese mismo nodo, y así sucesivamente. Realizando esta acción se pretende incrementar el número de nodos que reciben los mensajes generados en la red.
- Implementar un sistema simple de manejo de los mensajes. Esto es un buffer en el que se pueda controlar la forma de inserción, así como la forma de extracción o eliminación. De esta forma aseguramos un control exclusivo de todos los mensajes que existen en la aplicación, así como de las características del buffer.

Estos son, a grandes rasgos, los objetivos primarios. Se han llegado a implementar los tres de la mejor forma posible adaptándose a las restricciones que han ido surgiendo durante la elaboración del proyecto.

Tras el análisis del proyecto se desarrollaron los objetivos concretos que la aplicación debía tener. Estos objetivos se clasifican como objetivos secundarios y hacen referencia a la mayor parte de los aspectos del proyecto. Son los siguientes:

- Gestión de los mensajes mediante grupos. Los mensajes que el usuario mande con la aplicación están asociados a un determinado grupo. Esto hace que la gestión de la información sea más fácil, aumentando la comodidad del usuario.
- Mostrar las diferentes características de la aplicación en un menú diferenciado. Este menú presentará un estado de la aplicación y nos servirá principalmente para monitorizar el buffer de la aplicación.
- Poder observar los dispositivos de la red a los que nuestro aparato pueda conectarse. Nos puede resultar de ayuda saber esta información para saber si existen dispositivos con Bluetooth activado en nuestras cercanías.
- Poder obtener y manejar los mensajes entrantes en cualquier parte de la aplicación. Si el usuario está en cualquier parte de la aplicación que no sea la ventana de chat del grupo correspondiente al mismo grupo del mensaje, la aplicación deberá poder manejar el mensaje y clasificarlo según sus características. También deberá poder mostrar la información de dicho mensaje en los menús para que el usuario observe que un nuevo mensaje ha sido recibido.
- La información nueva que haya sido obtenida se deberá mostrar automáticamente en los diferentes menús correspondientes sin que el usuario intervenga en el proceso.
- Implementar características propias de una aplicación de mensajería. Características principalmente enfocadas a la interfaz: como la distribución de los mensajes en el chat de la misma forma que cualquier aplicación de mensajería. Incluir búsqueda de mensajes dentro de un chat. Cambiar el nombre del dispositivo para que sea visible por los demás usuarios. Y todas las posibles funcionalidades extra que se puedan incorporar para ofrecer al usuario una experiencia de aplicación de mensajería profesional.

Como se puede intuir, los objetivos principales están enfocados a la propia funcionalidad de la aplicación, mientras que todo lo referente a la interfaz y a cómo se muestra la información están relegados a objetivos secundarios. Esto es debido a que nos hemos querido centrar en el estudio del funcionamiento de una aplicación de este estilo antes que en el propio desarrollo de una aplicación de mensajería.

Una vez se han establecido los objetivos primarios, se han estudiado los secundarios. Cada uno de los objetivos secundarios descritos ha sido implementado en el proyecto a excepción quizás del último descrito. Este último objetivo engloba una serie de funcionalidades que no están específicamente acotadas ya que existen muchas características que una aplicación similar pueda compartir. Sin embargo, se ha hecho todo lo posible para ofrecer una experiencia similar en la aplicación presentada.

1.3 Resumen de la solución del problema

Para los diferentes tipos de problemas se han propuesto diferentes soluciones ya que no existe un problema común que se pueda resolver con una única solución.

Para el diseño de la aplicación se ha usado la propia API de Android ya que proporciona una base completa y robusta. Nos olvidamos de librerías o añadidos de terceros que compliquen los diseños de los menús. La interfaz está diseñada con elementos propios de Android. Desde hace pocos años, Android ha mejorado sustancialmente en todo lo

referente a la creación de interfaces. Así pues, actualmente se puede diseñar una interfaz muy avanzada sin necesidad de otras librerías como sucedía hace unos años.

Para la comunicación entre dispositivos se usa Bluetooth. Es una tecnología muy extendida que está presente en todos los teléfonos inteligentes de nuestro entorno. El uso en Android está presentado como la clásica conexión mediante sockets cliente-servidor y puede llegar a ofrecer una flexibilidad bastante grande. Además, la documentación presente en la red es muy diversa, ayudando a resolver los diferentes problemas o toma de decisiones que puedan surgir durante el desarrollo del proyecto.

Finalmente se ha decidido adoptar la forma de presentación de la aplicación como un programa clásico de mensajería el cual se divide en diferentes áreas según la funcionalidad y en donde el intercambio de información con los diferentes usuarios se realiza en una ventana donde los mensajes se presentan de diferentes formas según el usuario.

1.4 Ventajas de la solución

Las ventajas de ofrecer un desarrollo de una aplicación de mensajería para redes oportunísticas y ad-hoc son específicas pero importantes. Debido a que este tipo de red se despliega en el momento, son muy propicias para cierto tipo de situaciones. También aportan características diferentes al tipo de móvil centralizada que hacen esta solución una elección interesante frente a las redes tradicionales. A continuación, se exponen tres ventajas importantes de las que dispone la solución propuesta:

- No requiere una infraestructura de red: Debido a esta característica, se hace imposible que la red se llegue a congestionar en algún momento. Esto hace ideal este tipo de red para zonas masificadas como conciertos o manifestaciones, donde la transmisión de datos en una red con infraestructura puede llegar a provocar un mal funcionamiento. Otro posible escenario es uno en el cual directamente la infraestructura de red no exista o no se encuentra disponible en un determinado momento. Estas situaciones se dan en momentos críticos como catástrofes o apagones eléctricos.
- No tiene coste: Al no basarse en redes cuyo uso sea de pago, no hay coste alguno para el usuario. El uso de las redes móviles supone el uso de datos móviles, en cambio, utilizando la aplicación, el usuario dispone de una forma de comunicación totalmente gratuita.
- Privacidad y confidencialidad: Es prácticamente imposible que se intercepten los mensajes al ser una comunicación local y próxima. Los mensajes enviados por el usuario solo están disponibles en aquellos dispositivos que los hayan recibido. Este tipo de comunicación no depende de un nodo central por el cual fluye toda la información generada en la red, posibilitando más fácilmente la interceptación de los mensajes.

Teniendo en cuenta estas tres ventajas propuestas, se concluye que este tipo de solución solo tiene sentido en situaciones muy concretas. Sin embargo, el uso de la misma en las situaciones descritas, proporciona una solución de comunicación para estas circunstancias en las que antes no existía ninguna lo suficientemente estable para proporcionar el servicio deseado.

1.5 Descripción de la estructura de la memoria

En este apartado se pretende dar un breve resumen del contenido de cada uno de los capítulos de la memoria. Este mismo capítulo 1 (Introducción) ayuda a comprender el tema del trabajo introduciendo al lector al tema tratado, dando referencias sobre temas similares y exponiendo la solución escogida en el proyecto.

En el capítulo 2 nos centraremos en los antecedentes del tema del proyecto y describiremos los sistemas más conocidos en el área de aplicación. Este apartado es importante debido a que es necesario conocer cuáles son los problemas a resolver y como tenemos que resolverlos antes de empezar a escribir código. Todo el trabajo de investigación llevado a cabo se expondrá en este capítulo.

Para el capítulo 3 (Análisis) se describe detalladamente la solución, paso a paso. Se evita cualquier referencia a una concreta implementación de la aplicación o a las tecnologías específicas. Se describe el flujo de la información de ciertas áreas de la aplicación y de cómo este avanza a través de ella. También se detallarán las decisiones tomadas más importantes en el proyecto, así como la lista de requisitos recogida para la elaboración de la aplicación.

En el capítulo 4 (Diseño) se exponen las clases principales del programa y la relación entre ellas utilizando UML como lenguaje de descripción. Se describe la metodología de desarrollo, así como el funcionamiento de la aplicación. En este capítulo se hace hincapié en diferentes funcionalidades implementadas que se han considerado importantes y que por ello se explican en dicho apartado. Se expondrán los casos de uso que nos ayudan a entender las interacciones que puede llegar a tener el usuario con la aplicación. También se puntualiza en el diseño de la interfaz de usuario, parte importante en una aplicación para dispositivos móviles y tabletas.

En el capítulo 5 (Resultados) se muestran varios ejemplos completos del uso de la aplicación junto a las interacciones de los usuarios con el programa y sus respectivos resultados. Se evalúa el sistema y se comprueba si se ha alcanzado los objetivos descritos en este primer capítulo.

En el último capítulo tenemos las conclusiones. Se describen las características del desarrollo del proyecto. Se complementa con las conclusiones al trabajo realizado y los futuros trabajos posibles en esta área de trabajo.

Finalmente se adjuntan dos apéndices considerados de interés para complementar el trabajo. El primero de ellos se describe la utilización de la aplicación para que el usuario pueda consultarla en caso de duda. En el segundo, se muestran una lista de las clases con sus métodos más importantes.

2. Antecedentes y estado del arte

Para la realización de los objetivos anteriormente propuestos se han utilizado diferentes tecnologías. Cada una de ellas se ha estudiado correctamente para su correcta utilización en el proyecto. Se ha minimizado el uso de tecnologías de terceros o complejas y nos hemos centrado en aquellas de uso estándar y comúnmente utilizadas. En los siguientes apartados se exponen los sistemas utilizados para el desarrollo del proyecto, así como el estado actual de aplicaciones con objetivos similares y cómo nuestra propuesta puede suponer un paso hacia adelante en el uso propuestas similares.

2.1 Tecnologías utilizadas

En los siguientes apartados se listan las tecnologías más importantes utilizadas para el desarrollo del proyecto.

2.1.1 Android Studio

Android Studio es el entorno integrado de desarrollo (IDE) oficial para el desarrollo de aplicaciones en Android. Fue anunciado hace relativamente poco tiempo (16 de mayo de 2013) y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. Este sistema está basado en IntelliJ IDEA y diseñado expresamente para el desarrollo de aplicaciones en su plataforma. Algunas de las características más destacables que se han usado en este proyecto son: renderización en tiempo real para las interfaces, consola de desarrollador y la refactorización específica de Android.

2.1.2 Android

Android es un sistema operativo diseñado para dispositivos táctiles y una de sus grandes características es que está basado en Linux, sistema operativo libre y multiplataforma. El sistema utiliza una variación del lenguaje de programación Java llamado Dalvik, aunque a partir de la versión de Android Lollipop, Dalvik fue sustituida por ART. Hay que tener en cuenta que Android es un sistema que ha ido sufriendo variaciones a lo largo de los años por lo que es posible que algunas funciones descritas en este trabajo no se mantengan conforme avance el tiempo.

2.1.3 Bluetooth

Bluetooth es una especificación industrial para WPAN y está pensada para la transmisión de datos entre diferentes dispositivos. Bluetooth utiliza una tecnología de enlace por radiofrecuencia en la banda ISM de los 2,4GHz. El propósito principal de la existencia de esta tecnología es facilitar las comunicaciones entre equipos móviles. Además, al ser inalámbrica, nos da la oportunidad de crear pequeñas redes inalámbricas con facilidad y así poder facilitar la sincronización de datos entre diferentes dispositivos.

2.1.4 Smartphone

Para la implementación de la aplicación se ha hecho uso de un teléfono inteligente. Más concretamente, el modelo utilizado es un teléfono BQ Aquaris A4.5 con las características mostradas en la ilustración 1.

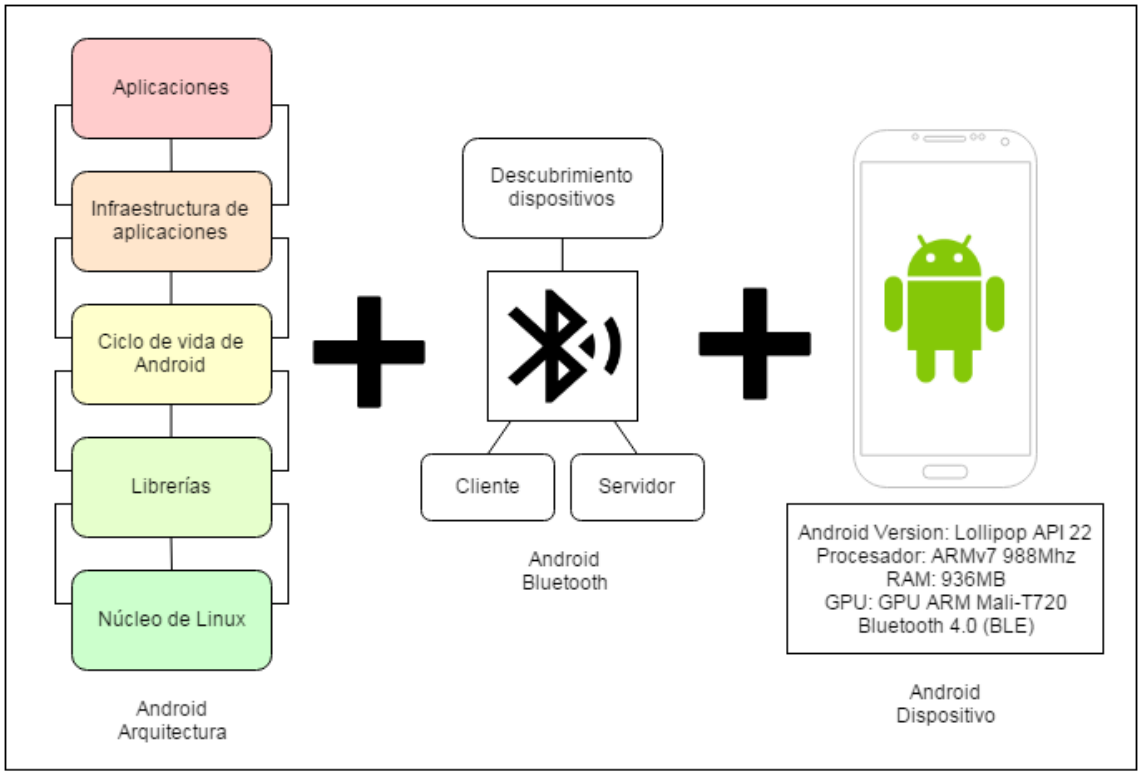

Ilustración 1 - Tecnologías usadas

Todas y cada una de las tecnologías comentadas se unen para formar la propuesta que se ha desarrollado.

2.2 Estado actual

No existe una gran cantidad de sistemas que realicen tareas parecidas a las que está enfocado este proyecto. En el artículo de [1] se realiza un análisis sobre aproximaciones de sistemas parecidos al que se ha desarrollado. En este artículo se abordan diferentes sistemas relacionados con la creación de redes de proximidad. También se realiza un estudio en el que se estudian las características de estos sistemas, se identifican los problemas y se ofrecen potenciales soluciones este tipo de aplicaciones. Nos encontramos ante un artículo muy útil que nos ofrece una serie de recomendaciones para la creación de futuros trabajos similares.

Si investigamos entre las aplicaciones que ofrece el mercado de Android, descubriremos que existen bastantes que simplemente se limitan a la creación de un chat mediante Bluetooth. Sin embargo, muy pocas de ellas se centran en la difusión entre grupos y solo se enfocan a chats entre usuarios individuales, es decir una conexión uno a uno. La mayoría de ellas las podemos encontrar buscando bajo el nombre de “Bluetooth Chat”.

Si enfocamos la búsqueda en aplicaciones mucho más profesionales encontramos dos grandes pilares:

- Firechat [2]: Utilizando conexiones punto a punto, establece una red mallada inalámbrica ad-hoc capaz de conectar múltiples usuarios sin necesidad de una conexión exterior.

- Meshme [3]: Aplicación con las mismas características que Firechat. En este caso sólo está disponible para dispositivos con iOS.

Para la estructuración de la aplicación se ha estudiado también aplicaciones muy extendidas actualmente como Whatsapp o Telegram. Estas aplicaciones a pesar de no ofrecer la misma funcionalidad que nuestro trabajo, nos pueden ayudar con ideas de estructuración y diseño, ya que ofrecen unas bases de aplicación típica de mensajería.

2.3 Innovación

Durante el desarrollo del proyecto se han usado herramientas ya existentes que están proporcionadas por la propia API de Android. En todas las tareas se han usado de diferentes formas, pero siempre usando los elementos proporcionados. No es necesario invertir tiempo en crear nuevas herramientas que realicen un determinado trabajo en tu aplicación si ya existen herramientas creadas para este fin, y que probablemente sean mucho mejores que las que uno mimo cree.

En el presente proyecto se elabora la estrategia de difusión de los mensajes, así como la forma de gestión de los mismos, de manera que se puedan adaptar en un futuro próximo para las incorporaciones de mejoras en estos sistemas. Esto quiere decir que el diseño del proyecto ha sido realizado pensando en esta característica. En un futuro se estima que se podrá añadir o modificar las diferentes partes de la aplicación para adaptarla a las diferentes pruebas que se quieran realizar.

Un punto en contra de innovar en un trabajo de estas características es el tiempo acotado en el que se realiza. Normalmente la innovación se consigue tras mucho esfuerzo y tiempo por lo que en este trabajo se presentan estas modificaciones a los sistemas de mensajería locales como algo básico.

2.4 Estado del arte en redes oportunísticas y ad-hoc

Las redes oportunísticas según [4], permiten el establecimiento de comunicación entre dispositivos móviles en lugares donde no hay disponibilidad de una infraestructura de transmisión de datos. El envío y recepción de información solo depende de la movilidad y oportunidad que tengan de establecer contacto con otros dispositivos, siempre y cuando estén dispuestos a colaborar para este efecto.

Debido al característico funcionamiento de las redes inalámbricas oportunísticas, se les suele denominar también como una subclase de las Redes Tolerantes a Retardo o DTN. Según [5], las DTN se pueden definir como redes formadas por otras más pequeñas. Esta tecnología soporta la interoperabilidad con otras redes al acomodar largas interrupciones y retrasos entre las mismas, además de tener la capacidad de operar con distintos protocolos. Con toda esta funcionalidad, las DTN pueden acomodarse a la movilidad y a la limitación energética de los dispositivos de comunicación inalámbricos que nos rodean.

El proceso de comunicación en las redes oportunísticas es caracterizado por la intermitencia y corta duración de los contactos entre pares de dispositivos móviles, modelo perfecto para entornos donde no exista infraestructura de transmisión de datos. Por esta causa, el diseño e implementación de protocolos y aplicaciones para este tipo de redes se ha convertido en objeto de estudio de importancia.

El método de compartir información de las redes oportunísticas está basado en la diseminación de los mensajes de manera epidémica. En [6] se refieren a este tipo de redes como Redes Parcialmente Conectadas debido a la intermitencia de los contactos. Además, detallan en profundidad una posible solución al encaminamiento de los mensajes para este tipo de redes inalámbricas.

Los protocolos en este tipo de comunicación oportunística, están pensados para almacenar y reenviar la información entre dispositivos. Estos definen el modo en que los mensajes puedan ser distribuidos entre los nodos.

Tanto el método de compartir información como el protocolo de almacenar y reenviar la información son la inspiración por la que se ha seguido el desarrollo de la aplicación creada en el proyecto.

Centrándonos en el análisis de los diferentes protocolos existentes, podemos encontrar trabajos como [9], en el cual se propone una taxonomía basada en el reenvío de la información. En cambio, en [10] se plantea una clasificación de acuerdo a la utilización o no de infraestructura de comunicaciones. Exponen estudios reales en los que se analizan los modelos de movimiento y rendimiento de la transmisión de datos de estas redes intermitentes.

Por otra parte, también se encuentran investigaciones con modelos analíticos en los que, en combinación con información de trazas reales, se plantea mejorar el rendimiento de los protocolos a través de la extensión del tiempo de vida de los mensajes (TTL). Además, incluyen variaciones del tamaño de los mensajes, con diferentes patrones de propagación de la información en un entorno de comunicación intermitente. En otros trabajos como [11], el estudio analítico realizado se centra en proponer y evaluar dos nuevos métodos de caracterización del tiempo entre contactos de los nodos de una red oportunística de comportamiento epidémico, desde una perspectiva del tiempo y espacio.

Cada uno de los trabajos citados anteriormente remarcan la importancia del estudio de este tipo de tecnología inalámbrica. Una de las principales ideas con las que se inicia el desarrollo de la aplicación es ofrecer una herramienta para entender mejor el comportamiento de estas redes en términos de rendimiento.

3. Análisis

En el apartado de análisis se pretende dar respuesta a diferentes aspectos de la aplicación planificados previamente a su implementación. En análisis se contesta a la pregunta "¿qué hace la aplicación?". Contestar adecuadamente a la pregunta se convierte en un punto clave del proyecto ya que un buen análisis y planificación ahora mucho trabajo y esfuerzo futuro.

3.1 Descripción de la solución

Explicaremos que es lo que realizan las diferentes tareas del programa. Tenemos la aplicación dividida en 5 grandes apartados. Las cuatro pestañas del menú y la ventana del chat.

En la pestaña de los mensajes recientes (así como en la de grupos y dispositivos), la interfaz se presenta como una lista de elementos de información. En estos elementos se diferencia el contenido de los últimos mensajes, así como el autor de dicho contenido. El usuario podrá iniciar un chat en el grupo asociado al mensaje si pulsa sobre el elemento de la lista mostrada.

Para la pestaña de grupos se listan aquellos que el dispositivo ha descubierto en la red en la que participa. Además de poder iniciar el chat asociado pulsando sobre cualquier elemento de la lista, el usuario puede crear nuevos grupos pulsando sobre el botón específico que se muestra en esta pestaña.

En la pestaña de dispositivos se le muestra al usuario otra lista, pero esta vez con los nombres de los dispositivos cercanos que participan en la red. El usuario en esta ocasión, puede actualizar la lista de dispositivos que el aparato puede alcanzar pulsando sobre el botón mostrado.

Para la última pestaña, se presenta toda la información que ha sido considerada de utilidad para el usuario. Actualmente el usuario puede observar información relevante a su dispositivo y se le da la oportunidad de cambiar el nombre con el que será representado en la red Bluetooth. Además, se puede observar información relevante al buffer de la aplicación: su capacidad, el número de mensajes actuales o el tamaño ocupado hasta el momento.

Finalmente nos centramos en la ventana del chat. Esta parte es la más importante. En esta ventana se presenta la funcionalidad de interacción con los usuarios de la aplicación. A través de un grupo en el que están asociados diversos usuarios, el usuario puede enviar y recibir mensajes mediante la caja de texto y el botón de envío que se mostrarán en la parte inferior de la pantalla, bajo el recuadro donde se mostrarán los mensajes.

La funcionalidad implementada para la recepción y envío de mensajes está oculta del usuario. Se ha creado un sistema que monitoriza la llegada de mensajes y que funciona mientras la aplicación se encuentre funcionando, es decir en cualquier parte del programa. Todo esto está implementado con un manejador que hace uso de los métodos propios de Android para Bluetooth. Se ha creado diversos tipos de mensajes dependiendo del tipo de información que la aplicación necesite recibir o enviar.

3.2 Descripción del flujo de la aplicación

Antes de entrar en detalle en los esquemas que muestran como la información viaja a través del programa, es necesario aclarar la estructura de la aplicación. A continuación, se presenta un esquema donde aparece el menú principal junto a cuatro estructuras de donde podemos acceder a toda la funcionalidad de la aplicación. El menú principal de la aplicación está divido en cuatro pestañas que hacen referencia a las cuatro estructuras comentadas. Estas estructuras están presentadas en cuatro colores diferentes para poder diferenciarlas con mayor facilidad. Hay que prestar especial atención a la tarea Chat cuya funcionalidad recoge el núcleo del proyecto y que se representa con un color diferente al de los demás.

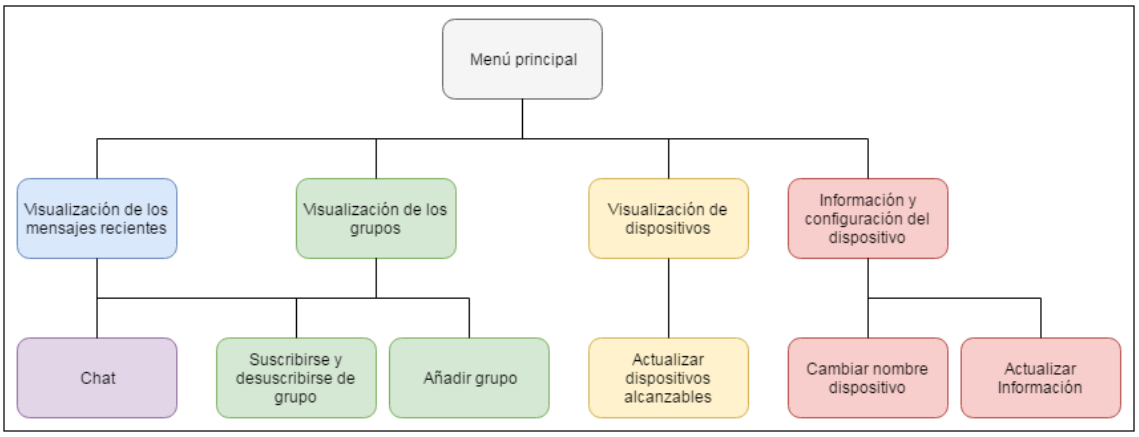

Ilustración 2 - Esquema aplicación

La primera pestaña representada por el color azul es donde se muestran los mensajes recientes. En esta tarea se recoge toda la información, la cual se ordena por grupos con el último mensaje asociado y donde el usuario puede acceder al chat correspondiente pulsando sobre la fila que desee.

La segunda pestaña es la de grupos y viene representada por el color verde. En este menú se ordenan los grupos que la aplicación ha recolectado. En él, el usuario puede suscribirse o cancelar la suscripción simplemente pulsando sobre un icono. También puede acceder a su correspondiente actividad chat pulsando sobre cualquiera de ellos.

La tercera pestaña de la aplicación representa la visualización de los dispositivos con los que la aplicación puede comunicarse en ese instante. En este menú el usuario puede actualizar los dispositivos para ver que aparatos están disponibles cuando lo desee.

La cuarta y última pestaña representa toda la información útil del programa y en ella se le ofrece al usuario la posibilidad de actualizar toda esta información y de cambiar el nombre con el que se visualiza a los demás nodos de la red.

3.2.1 Actividad Menú

En la parte de mensajes recientes se basa en una monitorización constante de posibles mensajes recibidos. Una vez esta tarea obtiene un nuevo mensaje, éste se añade al buffer de la aplicación y se actualiza toda la información referente al mismo; mostrando el nuevo mensaje y añadiendo, si no existe, el grupo asociado.

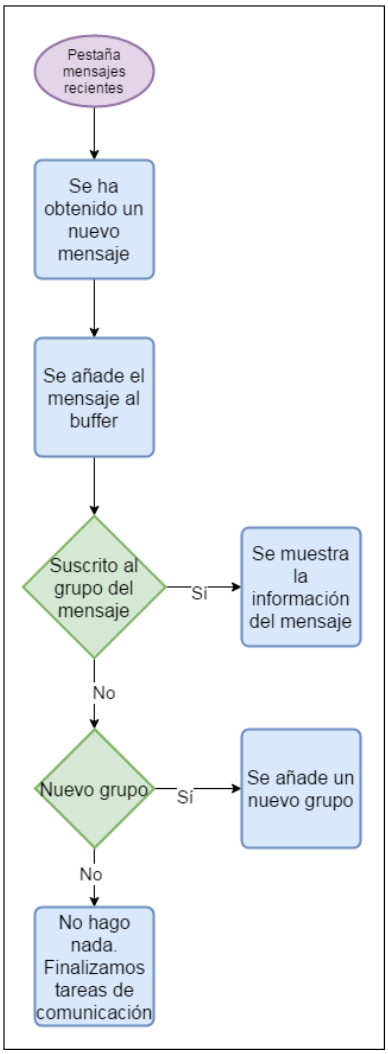

Ilustración 3 - Flujo mensajes recientes

La parte del flujo de gestión de grupos se resume en la siguiente imagen:

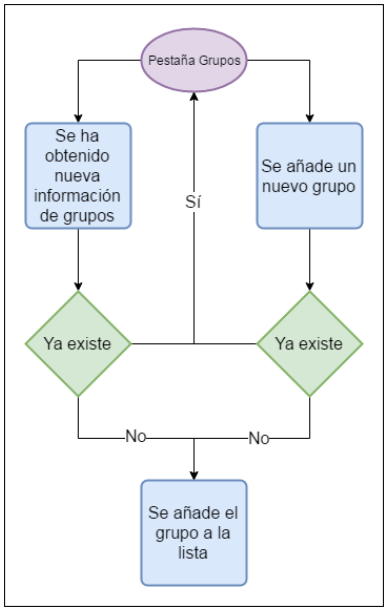

Ilustración 4 - Flujo gestión grupos

Para la gestión de grupos, la parte de trabajo central se encarga de observar si, tras recibir nuevos mensajes, éstos pertenecen a un grupo que el usuario de la aplicación no tenga registrado. Si no se tiene el grupo, se registrará y se le dará la oportunidad al usuario de suscribirse a él o no. En el caso de que el usuario quiera crear un nuevo grupo, se comprobará si ya existe antes de añadirlo.

Las tareas correspondientes a las de actualización de dispositivos y configuración e información no se han representado con un diagrama de flujo ya que son tareas simples que no requieren de grandes decisiones por lo que su representación mediante un esquema se ha descartado.

3.2.2 Actividad Chat

En este apartado se describe la parte de la aplicación más importante.

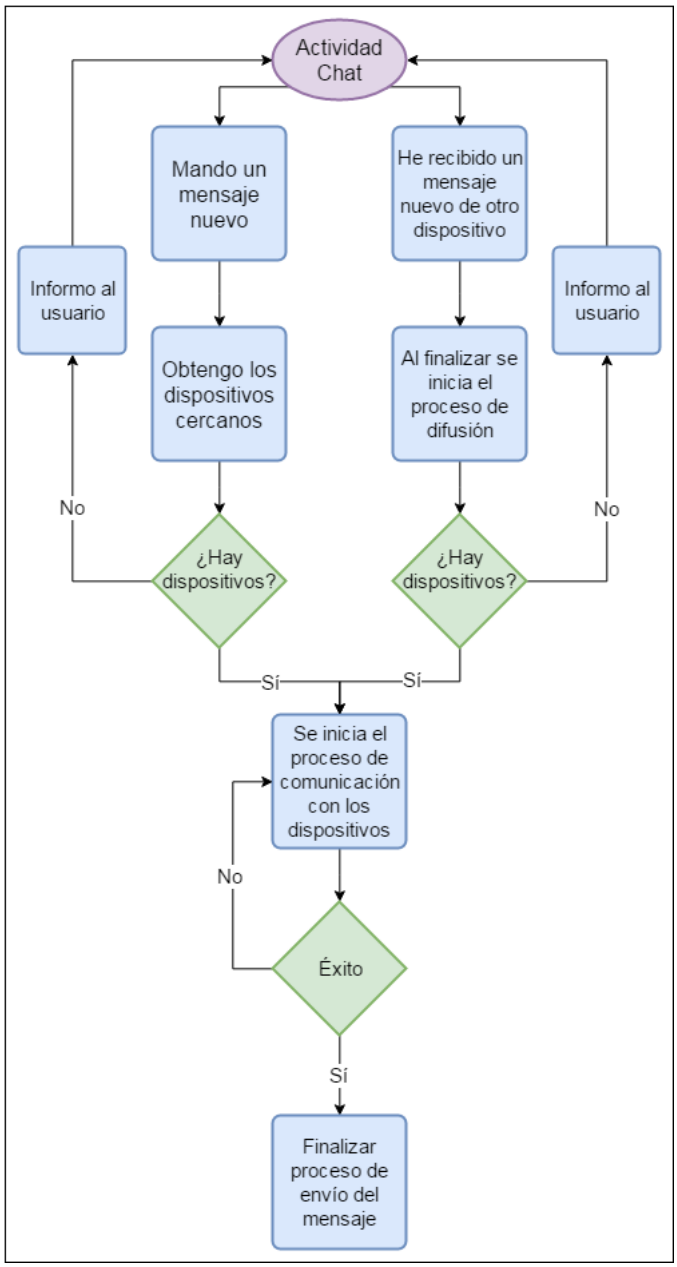

Ilustración 5 - Flujo actividad chat

En la anterior ilustración se observa el flujo de la actividad chat. Se diferencian dos ramas iniciales, en el caso de que se envíe un mensaje se debe de realizar un descubrimiento de los dispositivos cercanos para iniciar el procedimiento de comunicación. En el caso de que no se encuentren apartaos en la red, simplemente se le comunica al usuario. Por otra parte, si recibimos un mensaje entrante, procedemos también al descubrimiento de dispositivos cercanos. En los dos casos, una vez tengamos claros los nodos descubiertos, enviaremos el mensaje (entrante o creado por el usuario) a esos dispositivos.

3.3 Decisiones tomadas

A continuación, se habla de las decisiones tomadas sobre diferentes temas importantes del proyecto. Una decisión acertada puede llevar al éxito de un trabajo y una fallida puede demorar el desarrollo o incluso a tener que volver a empezar su realización. Por ello es por lo que se explican las razones a las que se ha llegado en cada uno de los temas siguientes.

Referente al uso exclusivo de Android y sus características, es importante saber que herramientas es necesario usar en la creación de una aplicación. En el análisis inicial del proyecto se identifica la funcionalidad que debe de tener la aplicación, por consecuencia se deducen las herramientas o utilidades necesarias para su elaboración, siendo Android la única tecnología necesaria para esta primera versión de la aplicación. Durante la realización del proyecto se estimó que no era necesario el uso de librerías o utilidades de terceros para lograr el funcionamiento deseado de la aplicación por lo que el uso exclusivo de Android y su funcionalidad es necesario.

La transmisión de información entre los nodos de la red se realiza mediante Bluetooth. La decisión de escoger Bluetooth y no otra tecnología estuvo sometida a estudio. La causa principal de la decisión final es que Bluetooth se encuentra en prácticamente cualquier teléfono inteligente hoy en día y permite una conexión punto a punto de corto alcance. Además, es una tecnología asequible que aporta toda la funcionalidad necesaria para nuestro proyecto y es fácilmente adaptable para otras ampliaciones futuras.

Bluetooth es una tecnología que ha sido renovada en repetidas ocasiones llegando a obtener lo que conocemos hoy en día como Bluetooth Low Energy (BLE). Esta tecnología nos permite hacer uso de Bluetooth con un coste mínimo en el impacto energético del sistema. La aplicación no hace uso del BLE, ya que solo los dispositivos más nuevos la soportan; sin embargo, es posible realizar una actualización a esta tecnología en nuestra aplicación, en caso de que se quiera utilizar, de una forma rápida y limpia.

Para hacer uso de Bluetooth en una aplicación Android, generalmente es necesario por parte del desarrollador, preguntar si el usuario de la aplicación está dispuesto a habilitar el uso de Bluetooth en su dispositivo. Además, cada vez que se realiza una conexión, se tiene que dar permisos por parte de los dos nodos para iniciarla. Debido a la naturaleza del proyecto propuesto, es decir, al gran número de conexiones que se realizan, es inviable realizar estas comprobaciones cada vez que se crea una conexión, por lo que en este proyecto se ha anulado tal proceso. En el caso de que se pregunte al

usuario constantemente por las conexiones Bluetooth, este podría sentirse molesto rápidamente.

La solución tomada para anular este tipo de comprobaciones ha sido evitar el paso recomendado de habilitar el descubrimiento de dispositivos. Introduciendo una comprobación en el código para habilitar el Bluetooth del dispositivo en caso de que este no esté habilitado es suficiente. De esta manera solo se avisará al usuario en caso de que el Bluetooth de su dispositivo no este habilitado y se le pedirá que lo habilite si lo desea. De la misma manera, se evitan los mensajes de peticiones de conexión utilizando una conexión de los dos tipos de conexión existentes en Android para Bluetooth (el normal, en el que se pide una verificación y el inseguro en el que no existe tal comprobación).

Si nos fijamos en la interfaz, la decisión que más impacto puede tener en una aplicación es la forma de distribución y su estilo. La idea de establecer un sistema de pestañas surgió previamente al estudio de aplicaciones similares. Los programas más populares actualmente que están enfocados a la mensajería, usan un sistema de pestañas para clasificar su funcionalidad. Así pues, se ha usado el mismo sistema para clasificar cuatro grandes grupos de información. Por otro lado, tenemos la interfaz del chat que no está enlazada directamente con el sistema de pestañas y que se crea como una nueva ventana cuando el usuario quiere entrar en alguna conversación. Esta técnica también la usan las aplicaciones anteriormente comentadas y supone una ventaja ya que además de proporcionar al usuario una diferenciación visual, también ayuda al desarrollador a controlar mejor el flujo de información para cada chat diferente.

3.4 Lista de requisitos

Los requisitos definidos se obtienen principalmente de los objetivos del proyecto. Estos se desglosan y se ordenan dándoles una especificación clara y concisa. Los requisitos son una parte esencial de un proyecto ya que ayudan al desarrollador a ordenar las tareas para completar el funcionamiento del sistema. Los requisitos de la aplicación se han dividido en dos tipos: funcionales y no funcionales.

3.4.1 Requisitos funcionales

Los requisitos funcionales son los que el usuario necesite que efectúe el software. Normalmente este tipo de requisitos explican el qué realizan ciertas actividades de nuestra aplicación. El usuario debe esperar una respuesta del sistema una vez se inicien este tipo de acciones. Seguidamente se listan los requisitos funcionales de nuestro proyecto:

Identificador:	**F01**
Título:	Inicio aplicación
Descripción:	La aplicación se ejecuta con su correspondiente icono en el menú de Android.

Identificador:	**F02**
Título:	Menú
Descripción:	El menú de la aplicación está dividido en cuatro áreas distintas: mensajes recientes, grupos, dispositivos alcanzables e

	información/configuración. Esto hace que se pueda diferenciar claramente las funcionalidades del programa.

Identificador:	**F02-1**
Título:	Menú - Representación
Descripción:	Para la representación de las áreas descritas en el requisito F02 se utiliza un sistema de pestañas en el cual el usuario puede navegar fácilmente realizando unos simples gestos o pulsando sobre los iconos correspondientes. Esté menú debe ser simple y limpio en consonancia con el resto de la aplicación.

Identificador:	**F03**
Título:	Chat
Descripción:	La aplicación tiene como objetivo el envío y recepción de mensajes. Por ello es necesaria la presencia de una interfaz a modo de chat en el que poder visualizar y diferenciar claramente los mensajes enviados y los recibidos.

Identificador:	**F03-1**
Título:	Chat – Envío
Descripción:	Para enviar los mensajes el usuario debe tener claro el procedimiento. Se debe incorporar una caja en la que el usuario introduzca el texto junto a un botón que se pulsa para enviar el texto introducido.

Identificador:	**F03-2**
Título:	Chat – Visualización
Descripción:	En el chat, la visualización de la información es importante. Por ello se debe idear una forma en la que el usuario diferencie claramente los mensajes de los diferentes dispositivos de la red, así como los suyos propios. Un posible ejemplo sería mostrar los mensajes ajenos en la parte izquierda de la pantalla con diferentes colores según el usuario y los mensajes propios mostrarlos en la parte derecha de la pantalla con un color característico.

Identificador:	**F03-3**
Título:	Chat – Actualización
Descripción:	Como cualquier aplicación chat, los mensajes recibidos deben mostrarse automáticamente sin intervención del usuario. De la misma forma, al enviar un mensaje, aparecerá en la pantalla.

Identificador:	F04
Título:	Recientes
Descripción:	Debe existir un modo en el que el usuario pueda ver de un vistazo los últimos mensajes que han sido recibidos por la aplicación.

Identificador:	F04-1
Título:	Recientes - Visualización
Descripción:	La visualización de los últimos mensajes recibidos debe estar bien diseñada ya que implica la visualización de mucha información en un espacio reducido. Para ello se debe mostrar una lista con el máximo número de mensajes recibidos. En cada mensaje debe mostrarse la información del texto del mensaje, así como el autor del mensaje y en qué grupo se ha mandado.

Identificador:	F04-2
Título:	Recientes - Actualización
Descripción:	Por el mismo método que en el requisito F03-3, la actualización se debe realizar automáticamente. Ya sea cuando el usuario esté visualizando la lista de mensajes recientes o cuando no lo esté. Si el usuario cancela la suscripción a un grupo, los mensajes asociados no se mostrarán en esta parte.

Identificador:	F05
Título:	Grupos
Descripción:	Para organizar los diferentes mensajes se representarán asociados a grupos. Estos grupos organizarán los mensajes de manera que el usuario pueda clasificarlos. Todos los grupos serán listados en esta interfaz.

Identificador:	F05-1
Título:	Grupos - Visualización
Descripción:	Siguiendo con la misma línea visual. Los grupos se listarán en la pestaña correspondiente. Cada elemento de la lista debe contener el nombre del grupo, una pequeña descripción y una manera de suscribirse al grupo en cuestión. Esta ventana también debe de disponer de algún mecanismo para añadir nuevos grupos.

Identificador:	F05-2
Título:	Grupos - Actualización
Descripción:	Siguiendo como en F03-3 y F04-2, la actualización es automática. Los nuevos grupos incorporados se mostrarán en la lista. El usuario podrá observar al instante si ha sido introducido un nuevo grupo. Los grupos se añadirán por defecto sin la

	suscripción activada.

Identificador:	**F06**
Título:	Dispositivos
Descripción:	La pestaña dispositivos simplemente muestra aquellos nodos de la red que están disponibles para realizar una conexión. Estos nodos serán a los que les llegue los mensajes que enviamos o de los que recibamos los mensajes.

Identificador:	**F06-1**
Título:	Dispositivos - Visualización
Descripción:	Los dispositivos se mostrarán mediante una lista. En esta lista cada uno de los elementos simplemente será un texto con el nombre del nodo. Se contará con un botón para actualizar.

Identificador:	**F06-2**
Título:	Dispositivos - Actualización
Descripción:	La actualización se realizará mediante un botón mostrado en la interfaz. Al pulsarlo, se iniciará la búsqueda de nodos cercanos, que se irán mostrando.

Identificador:	**F07**
Título:	Información/Configuración
Descripción:	La pestaña de información mostrará información al menos del propio dispositivo, del buffer y de los grupos. Para el dispositivo: nombre, dirección y estado del Bluetooth. Para el buffer: capacidad actual, numero de mensajes y capacidad máxima. Para los grupos: número de grupos y último grupo agregado.

Identificador:	**F07-1**
Título:	Información/Configuración - Visualización
Descripción:	La visualización de la información listada en F07 se debe agrupar para una mejor interpretación del usuario. Debe existir un botón por el cual el usuario pueda actualizar la información mostrada.

Identificador:	**F07-2**
Título:	Información/Configuración - Actualización
Descripción:	Para actualizar esta pestaña se hará uso del botón correspondiente. Al pulsarse, se comprobará el buffer, realizando

	la pertinente actualización y cálculos para el porcentaje de capacidad. También se debe verificar que el usuario haya cambiado el nombre del dispositivo para cambiarlo adecuadamente.

Identificador:	**F08**
Título:	Funcionamiento
Descripción:	Se debe poder recibir los mensajes de la red en cualquier momento en que la aplicación esté activa. El proceso de envío y difusión también debe seguir ejecutándose después de que el usuario envíe el mensaje. Estos procesos se harán invisibles al usuario sin que moleste su actividad en la aplicación.

Identificador:	**F08-1**
Título:	Funcionamiento - Visualización
Descripción:	Una vez la aplicación reciba un nuevo mensaje y en el caso de que el usuario no esté en el chat de grupo correspondiente a ese mensaje o en otra parte de la aplicación, la aplicación debe de mostrar al usuario la existencia de ese mensaje recibido en la pestaña reciente.

3.4.2 Requisitos no funcionales

Este tipo de requisitos definen las características de funcionamiento o los que especifican la manera de trabajar del sistema. Son los recursos que necesita el sistema para realizar las actividades correspondientes, como que la base de datos debe ser de un tipo en concreto.

Identificador:	**NF01**
Título:	Sistema básico y robusto
Descripción:	Para el uso del envío y recepción de los mensajes se hará uso de la funcionalidad propia de Android. Esto nos asegura que nos basamos en un sistema probado y robusto que nos ofrece una cierta seguridad.

Identificador:	**NF02**
Título:	Interfaz consistente
Descripción:	La interfaz de toda la aplicación debe mantener una coherencia visual y consistencia para no confundir al usuario.

Identificador:	**NF03**
Título:	Velocidad de envío y recepción
Descripción:	El tiempo de envío de los mensajes debe ser el menor posible. De

	la misma manera también se debe reducir el tiempo en el que un dispositivo esté recibiendo los mensajes.

Identificador:	**NF04**
Título:	Consistencia
Descripción:	La información guardada por la aplicación debe ser consistente y no variar de un tiempo a otro. En una primera versión se realiza mediante un archivo encapsulado.

Identificador:	**NF05**
Título:	Instalación aplicación
Descripción:	Si se quiere instalar la aplicación se hará mediante el archivo .apk generado por el framework de desarrollo.

Identificador:	**NF06**
Título:	Accesos
Descripción:	Al usar la aplicación se aceptan los permisos de escritura y lectura del dispositivo. Además, también se dará permisos de Bluetooth.

Identificador:	**NF07**
Título:	Escalabilidad
Descripción:	El diseño de la aplicación debe estar planificado de manera que sea posible añadir nueva funcionalidad futura.

4. Diseño

El objetivo del apartado de diseño es centrarse en cómo la aplicación realiza las actividades descritas. Se evita introducir cualquier extracto de código a no ser que sea estrictamente necesario ya que en ocasiones puede llevar a un entendimiento de la lectura erróneo. En diseño, se contesta la pregunta "¿cómo lo hace la aplicación?", describiendo los diferentes métodos utilizados para este fin.

4.1 Funcionamiento de la aplicación

Esta porción de capítulo es una de las más importantes de la memoria. En los siguientes apartados se explica con detalle los puntos más interesantes del proyecto, ya que en ellos se sustenta en mayor o menor medida las bases del proyecto. En cada uno de ellos se explica cómo funcionan estas bases y cómo se han utilizado para crear la parte del proyecto en la que se basa.

4.1.1 Bluetooth en Android

La plataforma Android ofrece un soporte para Bluetooth muy diverso, permitiendo a los dispositivos realizar diferentes acciones punto a punto con funcionalidad inalámbrica. Algunas de estas acciones es poder detectar los diferentes dispositivos con Bluetooth en la red, así como enviar y recibir información de estos diferentes dispositivos.

El funcionamiento de Bluetooth en Android se basa en un mecanismo cliente-servidor. Un dispositivo debe abrir un socket servidor y el otro dispositivo debe inicializar la conexión usando la dirección MAC del servidor. Una vez que los dos lados se han conectado, cada uno de ellos debe tener un socket de tipo Bluetooth en el mismo canal RFCOMM (*Radio Frequency Communication*, conjunto de protocolos de transporte).

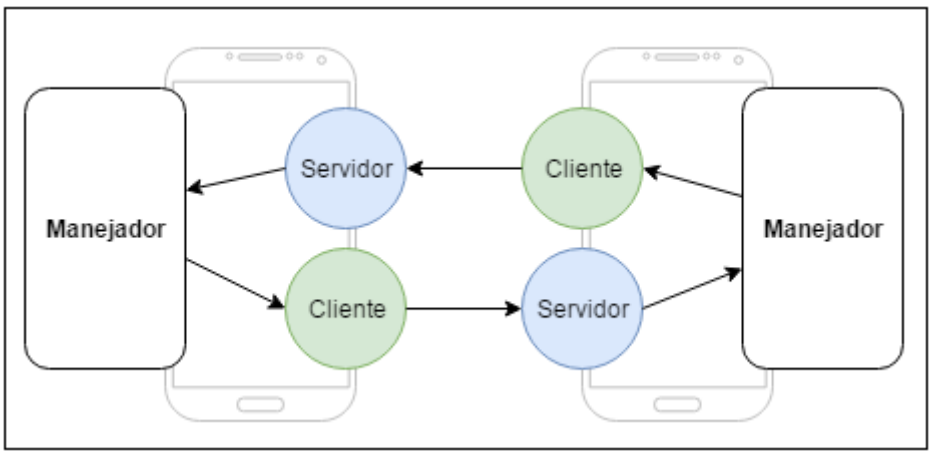

Ilustración 6 - Bluetooth en Android

En el proyecto se utiliza esta mecánica para implementar una Piconet Bluetooth que se explica más detenidamente en uno de los apartados siguientes. Una vez creado todo el sistema de comunicación, se tiene que crear otro sistema que nos permita manejar el tipo de información que obtenemos o enviamos en cada caso. Cuando se realiza el envío de un mensaje, la parte que recibe los datos necesita identificar de qué tipo son para actuar en consecuencia. Por ejemplo, si enviamos sólo la información de los

identificadores de los mensajes, el receptor necesita saber qué es ese tipo de información y que no está recibiendo la información correspondiente a un mensaje.

Para solucionar este problema se ha creado un manejador, que según el tipo de información que contenga el mensaje, la aplicación realiza unas acciones u otras. Este manejador es un elemento muy importante dentro del proyecto, ya que es la piedra angular que distribuye la información y mantiene la coherencia entre el flujo de información en la conexión de los dispositivos.

4.1.1 Difusión dirigida

Cuando hablamos de difusión nos referimos a la dispersión del mensaje dentro de la red local por el mayor número de nodos posibles. Si un nodo inicia una difusión creando un mensaje, éste será enviado a aquellos dispositivos que sólo él pueda encontrar. Sin embargo, pueden existir más nodos en la red que este primer nodo no haya podido alcanzar, por lo que los receptores del mensaje del primer nodo tienen que realizar el mismo procedimiento y difundir este nuevo mensaje.

Cuando hablamos de la especificación de difusión dirigida nos referimos a que se ha incluido una pequeña restricción la cual consiste en verificar si un cierto mensaje ha sido enviado a un dispositivo con anterioridad, si es así el nodo emisor se salta el envío del mensaje a ese dispositivo concreto. Con esta pequeña restricción nos podemos ahorrar ciertos envíos innecesarios.

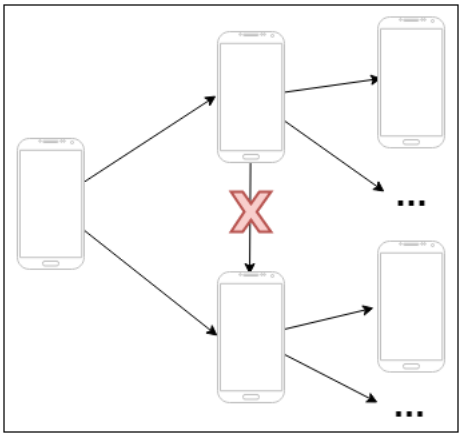

Ilustración 7 - Difusión dirigida

Es cierto que existen diferentes maneras de mejorar la implementación propuesta, sin embargo, actualmente esta aproximación nos sirve para abordar el problema. En un futuro es posible centrase en la creación de una difusión mucho más elaborada.

4.1.2 Intercambio de mensajes exclusivos

El objetivo de esta funcionalidad es que cada nodo de la red tenga todos los mensajes posibles que hayan sido generados por los usuarios que la forman. Para ello se ha establecido un protocolo que se ejecuta cada vez que dos nodos se conectan entre sí.

Este protocolo se puede resumir en tres pasos diferentes. Una descripción más detallada se encuentra en el apartado 4.4, en el que se presenta un diagrama de secuencia de los pasos de la conexión. Aquí se ofrece una explicación del concepto ideado para el protocolo.

A continuación, se muestran una serie de imágenes en las que se ilustran los tres pasos:

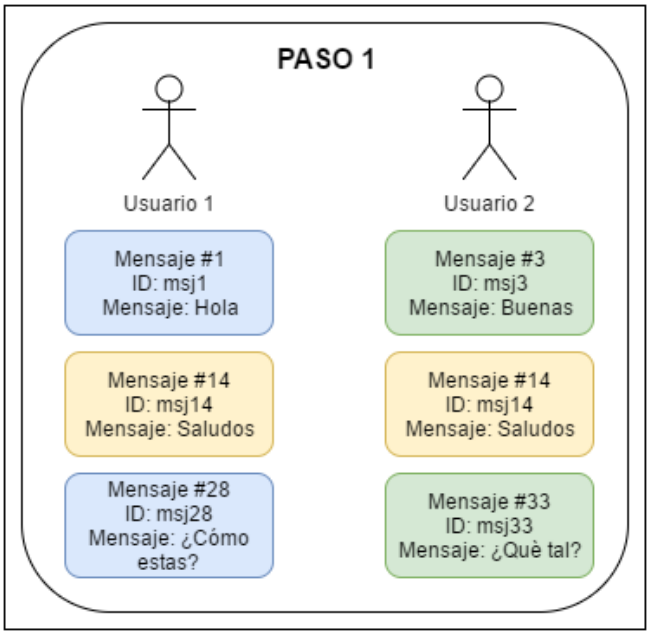

Ilustración 8 - Mensajes exclusivos paso 1

En el primer paso se observan los mensajes que tienen cada uno de los dispositivos que se han conectado. Cada uno de los mensajes tiene un identificador único (además de otra información que se verá en el apartado siguiente). En este nivel se listan todos los mensajes del propio dispositivo y se extraen los identificadores (ID) de cada uno de ellos para formar una lista.

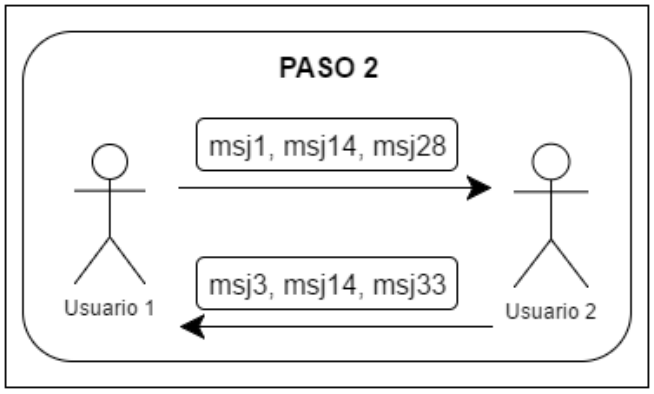

Ilustración 9 - Mensajes exclusivos paso 2

En el segundo paso cada aparato envía a su respectivo receptor la lista generada con todos sus identificadores de mensajes. Con esta lista, la aplicación extrae aquellos mensajes que tiene el dispositivo propio pero que no tiene el dispositivo remoto.

El motivo de que únicamente se envíen los identificadores es por el simple hecho de que se tiene que decidir que mensajes se tienen que intercambiar, ya que puede que algunos ya se hayan recibido anteriormente. Una vez se tiene la lista de estos mensajes exclusivos se procede con el siguiente paso.

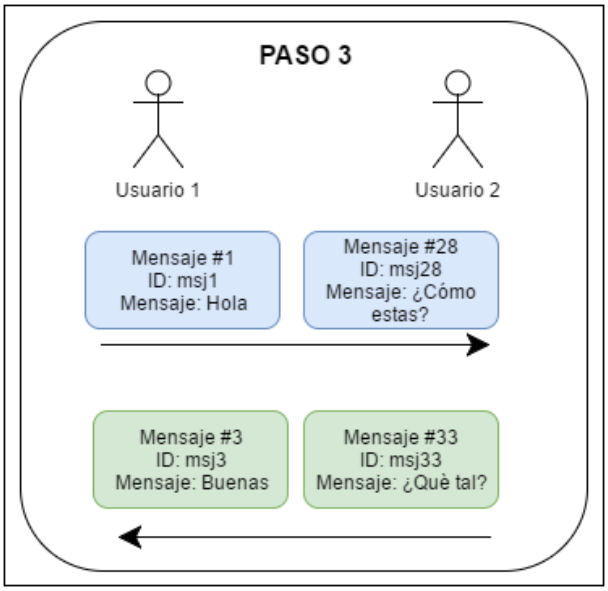

Ilustración 10 - Mensajes exclusivos paso 3

Finalmente, cada uno de los sistemas envía los mensajes exclusivos a sus receptores. Cada uno se encarga entonces de realizar las acciones pertinentes de añadir estos nuevos mensajes al buffer, mostrarlos al usuario si conviene o actualizar nuevos grupos si uno de estos mensajes pertenece a un grupo que el usuario receptor no tenga añadido a la aplicación.

4.1.3 Buffer de mensajes

El buffer de mensajes se pensó inicialmente para un mejor control de los mensajes en la aplicación. Esto ofrece la posibilidad de simular comportamientos en una aplicación de este estilo y estudiar sus resultados. Actualmente los sistemas tienen una gran capacidad de almacenamiento, por lo que, si simplemente almacenásemos los mensajes en una simple lista, podríamos llegar a guardar un número muy grande de mensajes sin problema alguno. Si quisiéramos simular un sistema de poca capacidad de almacenamiento, la forma en que se ha planteado la solución del buffer nos da esta posibilidad.

Ilustración 11 - Estructura mensaje

En la imagen se puede ver la estructura completa de un mensaje con sus campos completados con información de ejemplo. Este tipo de mensajes son los utilizados en el proyecto. A continuación, se define cada campo explicando su uso dentro del proyecto:

- ID: Identificador único de mensaje. Cada mensaje tiene su propio identificador y no se repite bajo ningún concepto. Este inmutable identificador único universal (UUID) representa un valor de 128 bits.
- Tipo: Representa el tipo de mensaje. Este valor ayuda a clasificar los mensajes según quieran representarse de una forma u otra.
- Texto: Cadena de caracteres que el usuario ha escrito en el chat de la aplicación.
- MAC: Dirección de 48 bits que representa el dispositivo o nodo del autor del mensaje en la red.
- Nombre: Cadena de caracteres que identifica al usuario en los grupos de chat.
- Fecha: Fecha de creación del mensaje. De utilidad para administrar el TTL.
- TTL: Tiempo de vida que se le da a un mensaje. Representa el tiempo máximo que el mensaje es válido dentro de la red.
- Grupo: Grupo asociado al mensaje. Cada mensaje se ha debido de generar en el chat de un determinado grupo.
- Dispositivos: Lista de dispositivos que contienen el mensaje en cuestión. Nos ayuda a identificar aquellos dispositivos a los que no tenemos que enviar el mensaje durante el proceso de difusión.

El buffer creado para esta aplicación tiene dos atributos importantes: el tamaño máximo y el tamaño actual. El tamaño máximo es la capacidad (por ejemplo, en bytes) que el buffer puede alcanzar. El tamaño actual es un valor que se actualiza cada vez que se añade o se elimina un mensaje.

Sabiendo la estructura de los mensajes se pueden implementar diversas maneras de administrarlos. En el caso de este proyecto, cada vez que se intentar introducir un mensaje en el buffer, se comprueba si el buffer tiene capacidad suficiente consultando el atributo del tamaño actual. Antes de añadir cada mensaje se realiza una comprobación y si añadiendo el mensaje no se supera la capacidad máxima, el mensaje entra en el buffer.

Por el contrario, al eliminar un mensaje, se ha elegido un método por el cual se hace uso del TTL. Previamente a la inserción de los mensajes, se realiza una comprobación del buffer para eliminar aquellos cuyo TTL ha expirado. Esto es, se comprueba la fecha de creación del mensaje con la actual y si ha superado el TTL asignado, el mensaje se borra del buffer actualizando su tamaño actual.

4.1.4 Gestión de Grupos

Durante el diseño de la aplicación se pensaron diferentes maneras de ordenar los mensajes. Finalmente se llegó a la conclusión de que la más acertada era mediante grupos. Las más famosas aplicaciones de mensajería actuales utilizan este sistema para sus proyectos, además de ofrecer la posibilidad de establecer un chat con una sola persona. Este proyecto está enfocado a la distribución de un mensaje en una red de varios nodos, por lo que la creación de chats con usuarios individuales no tiene sentido.

Junto a la gestión de grupos se ha creado un sistema de suscripción por el cual el usuario podrá decidir que mensajes visualizar según a que grupos este suscrito. Si un

usuario recibe un mensaje que pertenece a un grupo al que esté suscrito, este se mostrará en la interfaz. Por el contrario, si el mensaje pertenece a un grupo que el usuario haya indicado que no está suscrito, el mensaje simplemente se guardará en el buffer para una posible redifusión, pero no se le mostrará en la interfaz.

Si tenemos el caso de que la aplicación ha recibido un mensaje cuyo grupo no exista, el grupo se añade a la lista de grupos. Este nuevo grupo tendrá la suscripción desactivada por defecto ya que puede darse el caso de que se añadan muchos grupos de golpe y el usuario se vea afectado por una gran cantidad de información que puede que no desee. El usuario puede consultar en cualquier momento la lista de grupos y verificar si existen nuevos y si lo desea, poder suscribirse a los recién añadidos.

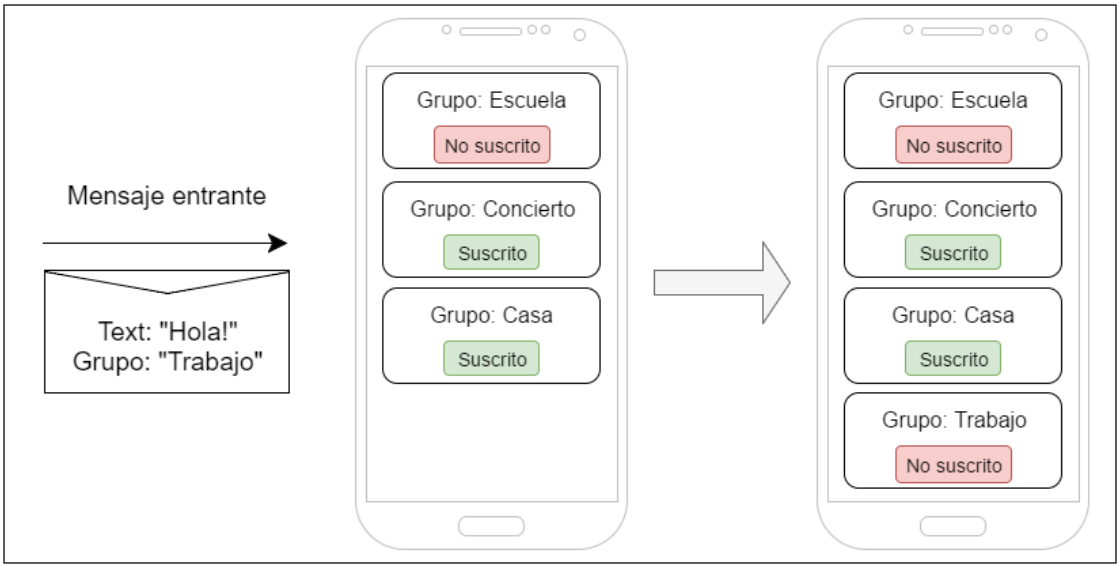

Ilustración 12 - Gestión grupos

Por último, tenemos la creación de grupos por parte de un usuario. La acción de añadir un nuevo grupo es posible realizarla en cualquier momento. El usuario tendrá la posibilidad de introducir el título del grupo junto a una pequeña descripción. Al hacerlo, automáticamente se añadirá el grupo y se podrá empezar a enviar mensajes a través de este nuevo grupo. Los mensajes del nuevo grupo serán recogidos por los demás nodos de la red y este grupo se añadirá automáticamente a la aplicación de estos nodos.

4.1.5 Mensajes recientes

La parte de mensajes recientes está relacionada con la de grupos. Los mensajes que se muestran en esta interfaz serán aquellos cuyo grupo haya confirmado el usuario la suscripción. Si se recibe un mensaje y el grupo de éste no pertenece a la lista de grupos suscritos, la aplicación simplemente guardará el mensaje en el buffer, pero no se mostrará en la parte de mensajes recientes.

Si el usuario activa la suscripción a un grupo determinado, los mensajes asociados a ese grupo se mostrarán de inmediato en la parte de mensajes recientes. De la misma forma si el usuario cancela la suscripción a un grupo, los mensajes asociados a él, se dejarán de mostrar en la ventana. Todo este proceso se verifica cuando la aplicación recibe un nuevo mensaje y se realiza de forma automática.

La ventana de mensajes recientes tiene como finalidad ayudar a el usuario a ver las actualizaciones de los grupos. Por lo que, si se está observando la pestaña de mensajes recientes y se recibe un mensaje nuevo de un grupo suscrito, se actualizará de inmediato la ventana de mensajes recientes. En ese momento el usuario puede pulsar sobre el elemento actualizado y entrar en el chat para contestar el mensaje.

4.1.6 Piconet y broadcast

Una piconet es una red adhoc que une un grupo de usuarios con dispositivos inalámbricos que utilizan Bluetooth. Estas redes constan siempre de un nodo "maestro" dejando a los demás el rol de "esclavos". En esta situación, se ha adaptado esta idea para la implementación de un sistema que emula una difusión de un mensaje a varios nodos a la vez.

El método de piconet se ha usado para cada vez que el aparato necesite enviar datos a los demás dispositivitos de la red, esto es, cada vez que un usuario manda un mensaje o que el dispositivo requiera difundir un mensaje que haya sido recibido. Entonces, el sistema realizara el proceso de descubrimiento para obtener los nodos cercano. A continuación, se iniciará una conexión para cada uno de los nodos. Una vez el sistema emisor haya establecido una conexión con todos los nodos, se realiza el broadcast del mensaje pertinente.

Para manejar las conexiones creadas con diferentes nodos durante el proceso, se han utilizado hilos de trabajo (Threads). Este método nos ayuda a paralelizar todo el proceso que realizaríamos para un solo nodo. De este modo si se quiere enviar un mensaje, el sistema recopila los aparatos disponibles como se ha comentado anteriormente e inicia el mismo proceso para cada uno. Se crea un hilo y en este hilo se establece una conexión con los sockets específicos para el nodo destino. Este hilo se almacena para, una vez acabada la creación de todos los hilos para todos los nodos descubiertos, poder usarlo y mandar el mensaje correspondiente.

La conexión establecida con cada uno de los nodos mediante hilos es bidireccional e independiente de los demás hilos con los demás nodos. Esto quiere decir que se puede recibir y enviar información con cada nodo y que cada uno puede ir a la velocidad que le permita su sistema, aumentando la flexibilidad de la aplicación. Simplemente, se tiene que coordinar que todas las conexiones han terminado para proceder a los siguientes pasos del protocolo.

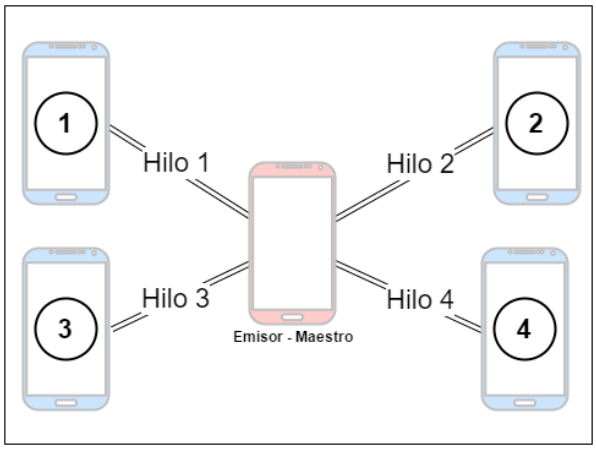

Ilustración 13 - Piconet

Antes de la construcción del sistema piconet, la conexión con los dispositivos para mandar la información se realizaba uno por uno. El motivo por el cual se planteó de esta manera al inicio fue para tener un mayor control en las conexiones que se realizaban con los diferentes dispositivos. Este control nos permitía diferenciar claramente los pasos que se realizaban para un dispositivo en concreto. Sin embargo, este método resultaba muy lento, ya que, si el número de nodos a enviar la información era elevado, el tiempo que se necesitaba para enviar el mensaje a cada uno de ellos crecía sustancialmente.

Las herramientas que ofrece Android sobre Bluetooth están orientadas a la conexión de nodos punto a punto, por lo que no es extraño pensar en construir un sistema como el descrito anteriormente. Sin embargo, por la causa descrita anteriormente, se hizo necesario idear una forma de implementar una piconet para dispositivos Android.

4.1.7 Escucha de mensajes y manejador

Para verificar que se ha enviado un mensaje determinado, que la conexión con un nodo remoto se ha establecido correctamente o que la conexión ha terminado inesperadamente, se ha ideado un sistema el cual permite recibir mensajes y actuar en consecuencia.

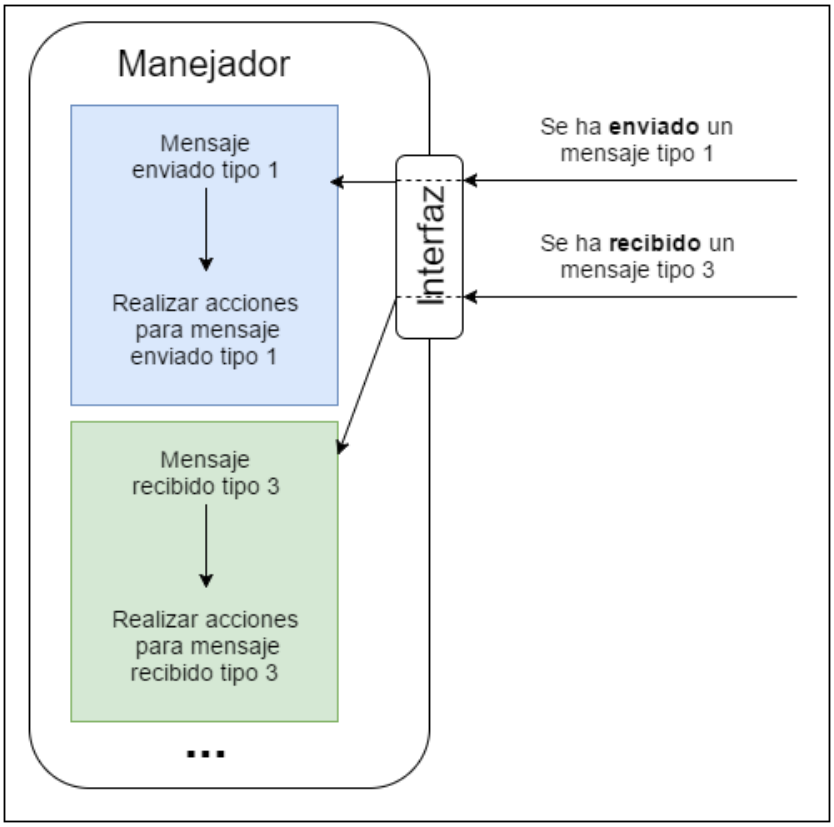

Ilustración 14 - Manejador

Este sistema está pensado inicialmente para poder recibir mensajes en cualquier momento, pero se ha extendido para poder recibir no solo mensajes enviados por otros dispositivos, si no para mensajes producidos por ciertas acciones de nuestra aplicación. Poder estar al corriente de todas las acciones que ocurren en el sistema y actuar en consecuencia es una característica muy importante en sistemas que usan la red y

establecen comunicaciones porque para cualquier inconveniente o acción se puede tener una acción específica que actúe en consecuencia.

Todo este sistema se ha incluido mediante un manejador. Este manejador incluido en el sistema es una rutina que incluye una interfaz para su correcta utilización. Algunos de las posibles acciones que el manejador puede hacerse cargo son:

- Cuando el sistema establece la conexión con los dispositivos.
- Cuando se ha enviado un mensaje, incluidos diferentes tipos.
- Cuando se ha recibido un mensaje, incluidos diferentes tipos.
- Cuando se ha perdido una conexión o cuando no se ha podido establecer una.

Si el sistema se incluyese dentro de alguna clase inferior a la actividad principal, cabe la posibilidad de que esta clase desapareciese debido a ciertas acciones del usuario. Para que todo el sistema funcione correctamente, el manejador es necesario que se sitúe en la clase principal del sistema para que pueda funcionar en cualquier situación. Esta actividad (MainActivity) es la que contiene información importante y es la que se encarga de iniciar y mantener todos los elementos importantes de una aplicación Android. De esta manera el usuario puede estar navegando por cualquier parte de la aplicación y el sistema seguirá teniendo la posibilidad de recibir mensajes o controlar algunas acciones.

4.1.8 Tipos de mensajes

Mientras se mantiene la conexión con un dispositivo se realizan varios intercambios de información mutua y cada uno de estos intercambios contiene información que debe ser tratada de formas diferentes. Por ello se hace necesario identificar cada uno de estos envíos con un identificador para que se procese la información de la manera adecuada.

Como los diferentes tipos de mensaje representan diferentes tipos de datos en las diferentes situaciones se hace necesario encapsular estos datos en un tipo de mensaje estándar. En este mensaje siempre tendremos un atributo que nos indique el tipo de mensaje que transporta además de la información en cuestión. De esta manera el sistema siempre espera el mismo tipo de mensaje, haciendo más fácil la decodificación y la consecuente administración de la información del mensaje.

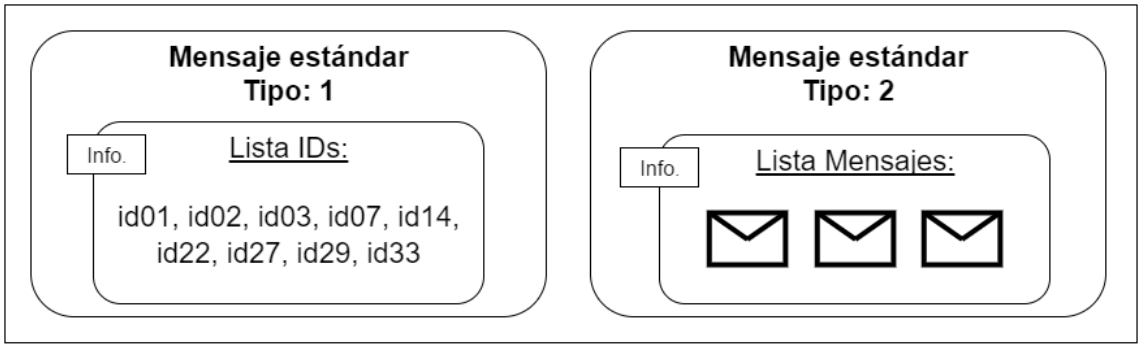

Ilustración 15 - Tipos de mensajes

Antes de entrar a hablar más a fondo con los diferentes tipos de mensajes, es necesario aclarar que, para transportar la información de un sistema a otro, el comentado mensaje estándar se serializa antes. La serialización es un proceso que consiste en un proceso de codificación de un objeto (el mensaje en este caso) en un medio de

almacenamiento con el fin de transmitirlo a través de una conexión como una serie de bytes. Es un proceso estándar que se realiza en las conexiones.

En el proyecto se han propuesto dos tipos de mensajes importantes:

- Lista de todos los identificadores de los mensajes de la aplicación. Este tipo de mensaje se manda cuando los dispositivos acaban de establecer la conexión. Es el momento entonces de intercambiar todos los identificadores para entonces verificar aquellos que el dispositivo remoto no tenga en su sistema.
- Lista de mensajes exclusivos. Este tipo de mensaje representa el encapsulamiento de una lista de todos aquellos mensajes que el sistema emisor almacena pero que no exista en el nodo remoto.

Además de estos tipos, el sistema se puede extender a otros tipos de mensajes como por ejemplo el envío de imágenes o archivos de sonido. De esta manera se hace mucho más fácil identificar el tipo de información que se envía a través de una conexión.

4.1.9 Descubrimiento de dispositivos

Para poder saber que dispositivos están disponibles en la red, el programa debe iniciar un proceso de descubrimiento el cual mediante la propia API de Android se obtienen los diferentes nodos conforme se van descubriendo. La mecánica utilizada en el proyecto se basa en la inicialización del proceso en el momento que se requiera obtener los dispositivos de la red y durante un tiempo determinado el proceso se cancela para proceder con los siguientes pasos. Durante este tiempo la funcionalidad de Android va arrojando los diferentes dispositivos que encuentra y el programador debe ir guardándolos adecuadamente o realizando las acciones pertinentes.

En nuestra aplicación se ha definido un tiempo determinado por el cual el dispositivo está en modo descubrimiento. Este tiempo es crucial por que debe ser lo suficientemente corto para que los primeros dispositivos encontrados desaparezcan del rango del nodo y debe ser lo suficientemente grande para que al sistema le dé tiempo de encontrar un numero de nodos razonable.

4.1.10 Guardado y carga de la información

En la aplicación también se ha implementado un sistema de guardado y carga. Este sistema es uno muy básico en el que se serializa la información determinante y se guarda en el almacenamiento interno del teléfono móvil en forma de archivo.

Es indispensable saber cuándo guardar la información de la aplicación, ya que un frecuente guardado de información puede producir comportamientos lentos. Por el caso contrario si no guardamos la información en el momento preciso, se pueden producir perdidas de datos. Se ha realizado un pequeño estudio y se ha descubierto que uno de los mejores momentos para realizar estas acciones son en las funciones ejecutadas durante el ciclo de vida de Android. El ciclo de vida de Android son una serie de estados por los cuales tiene que pasar el flujo de información de toda aplicación Android. Durante este ciclo, Android realiza llamadas a funciones establecidas por el sistema. Las dos llamadas que usaremos para guardar y cargar la información son:

- *onStop()*: Esta llamada de función se produce cuando el usuario esconde la aplicación en el sistema Android. Este estado se define como “Parado”, y es uno

por el cual debe pasar de cualquier modo si el usuario cambia y deja la aplicación en segundo plano o la cierra completamente.
- *onCreate()*: Esta llamada se realiza cada vez que la aplicación se inicia. Es la llamada obvia en la que establecer la carga de información de la aplicación.

Otro aspecto a tener en cuenta es la información que necesitamos guardar. Esta decisión es importante ya que es necesario reducir el archivo de guardado, por lo que debemos excluir cierto tiempo de información redundante y escoger aquella que nos haga falta para reestablecer el estado del programa antes de que el usuario lo cerrase. Esta información esta encapsulada en una clase para su posterior serialización y guardado.

La información que se extrae en este caso es:

- Buffer: Buffer de todos los mensajes de la aplicación. La información más importante. Con esta información se puede reestablecer casi la totalidad de la información necesaria de la aplicación.
- Lista de grupos: Grupos que contiene el dispositivo. Para poder saber a qué grupos está suscrito el usuario es necesario guardar este tipo de información.

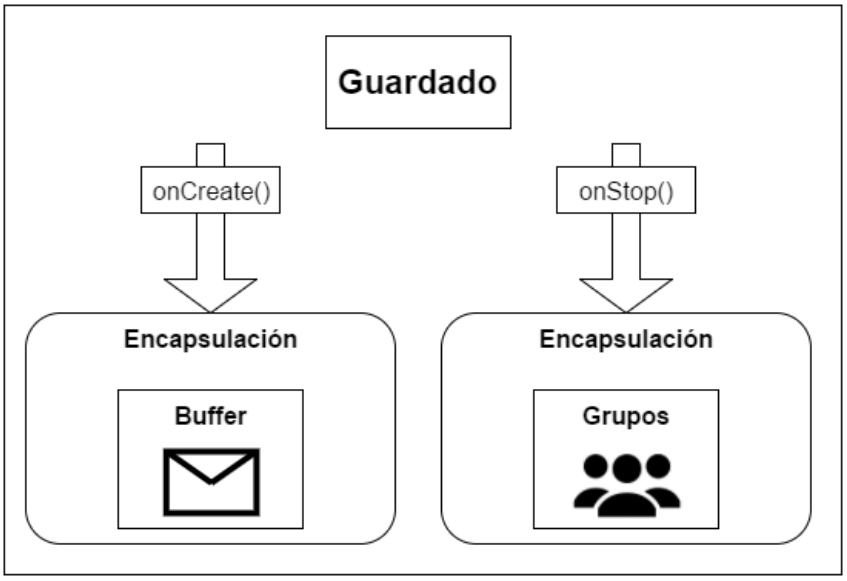

Ilustración 16 - Guardado

4.2 Diagrama de clases

Los diagramas de clases muestran las clases que conforman la aplicación y cómo se relacionan entre sí. Los diagramas de clases son diagramas estáticos que muestran las clases junto a sus métodos y atributos. Con estos tipos de diagramas podemos observar qué clases conocen a qué otras o qué clases forman parte de otras. Sin embargo, este tipo de diagramas no muestran los métodos mediante los que se invocan entre ellas.

En la siguiente ilustración se muestra el diagrama de clases de la aplicación. En Android existen clases denominadas “Activity”, estas clases representa un conjunto de operatividad de una misma parte de la aplicación y que generalmente están asociadas a una parte de la interfaz. Para hacer un símil diríamos que son como las ventanas de Windows. Para la creación de interfaces más complejas dentro de estas actividades se

hacen uso de los “Fragment” que representan partes de la interfaz con una funcionalidad específica asociada. En la ilustración del diagrama observaremos clases que han sido nombradas con estos nombres para una mejor clasificación.

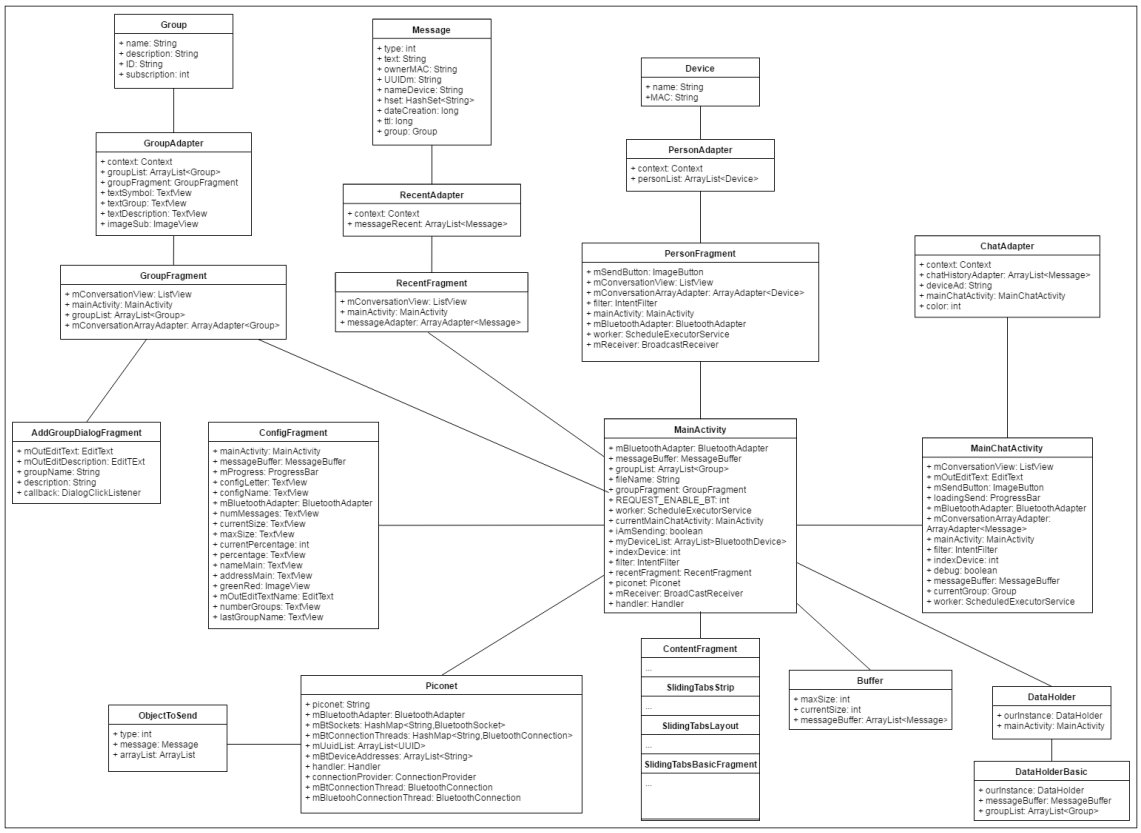

Ilustración 17 - Diagrama de clases

Para cada una de las clases representada por un recuadro, se muestran el título (fila superior) y los atributos (fila inferior). Los métodos de cada una de las clases han sido introducidos como anexo a la memoria debido a que el diagrama se haría ilegible.

Dos de las clases más importantes son “MainActivity” y “MainChatActiviy”. La primera es la clase que controla la emisión y recepción de mensajes, es decir la que aloja el manejador que se ocupa de esto. La segunda clase es la principal del chat de la aplicación. Como es obvio, las dos tienen una fuerte relación. Cada una de las pestañas que forman la aplicación están asociadas a una clase distinta: Recent, Group, Person y ShowConfiguration. A su vez, éstas tienen un adaptador para administrar los elementos que muestran sus interfaces. La mayoría de las clases están interconectadas y el puente que hace esto posible es la actividad principal.

4.3 Diagrama de estados

La funcionalidad principal de los diagramas de estados es conocer el comportamiento del sistema. Describen todos los estados posibles en los que puede aparecer el sistema y la manera en que cambia el estado del sistema. Estos diagramas representan la vida de una aplicación. Se indica mediante flechas, eventos que hacen que un estado cambie a otro y cuáles son las respuestas que éste genera. En nuestro caso, la aplicación es manejada por el usuario y cada uno de los estados es elegido por el mismo. Cada acción

que cambia de estado es iniciada por el usuario ya que el diagrama de estados se utiliza normalmente para describir los estados del dominio del usuario.

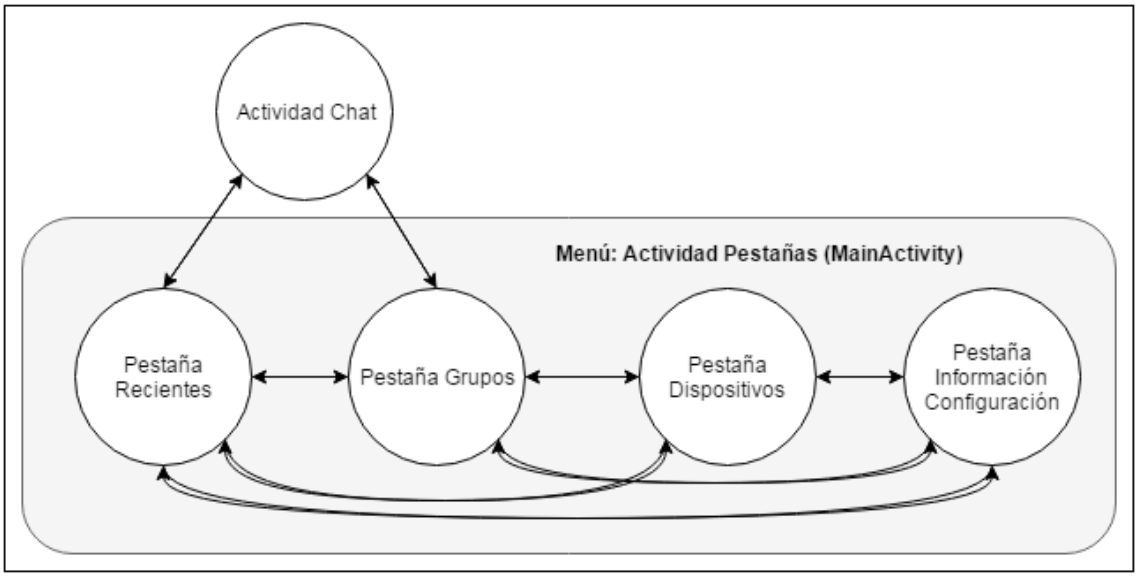

Ilustración 18 - Diagrama de estados

En la imagen se aprecia el diagrama de estados representando la aplicación. Se observan cuatro estados que están englobados por el menú. Estos estados representan las cuatro pestañas del programa que hacen referencia a: mensajes recientes, grupos, dispositivos e información. De cada una de las pestañas se puede acceder a cualquiera de las otras pestañas.

La actividad del chat sólo puede ser accedida por aquellas pestañas que contienen información referente a los grupos (en recientes se muestran los últimos mensajes dentro de los grupos). Se concluye que en general la aplicación no tiene una gran cantidad de estados, esto es debido a que se ha conseguido simplificar la funcionalidad mediante el sistema de ordenación mediante pestañas.

4.4 Diagrama de secuencia

Un diagrama de secuencia nos ayuda a identificar la interacción de diferentes objetos en una aplicación a través del tiempo. Típicamente se usa para representar un caso de uso específico para determinar que objetos son necesarios para la implementación del escenario. Los diagramas de secuencia muestran los objetos que intervienen en el escenario mediante líneas discontinuas verticales, y los mensajes intercambiados entre los roles como flechas horizontales.

Ya que la aplicación incluye comportamiento relacionado con redes, una forma clara de representar la interacción entre los nodos de las conexiones es mediante un diagrama de secuencia. Es recomendable incluir este tipo de diagramas para apreciar más detenidamente los pasos que realiza la aplicación en su comportamiento.

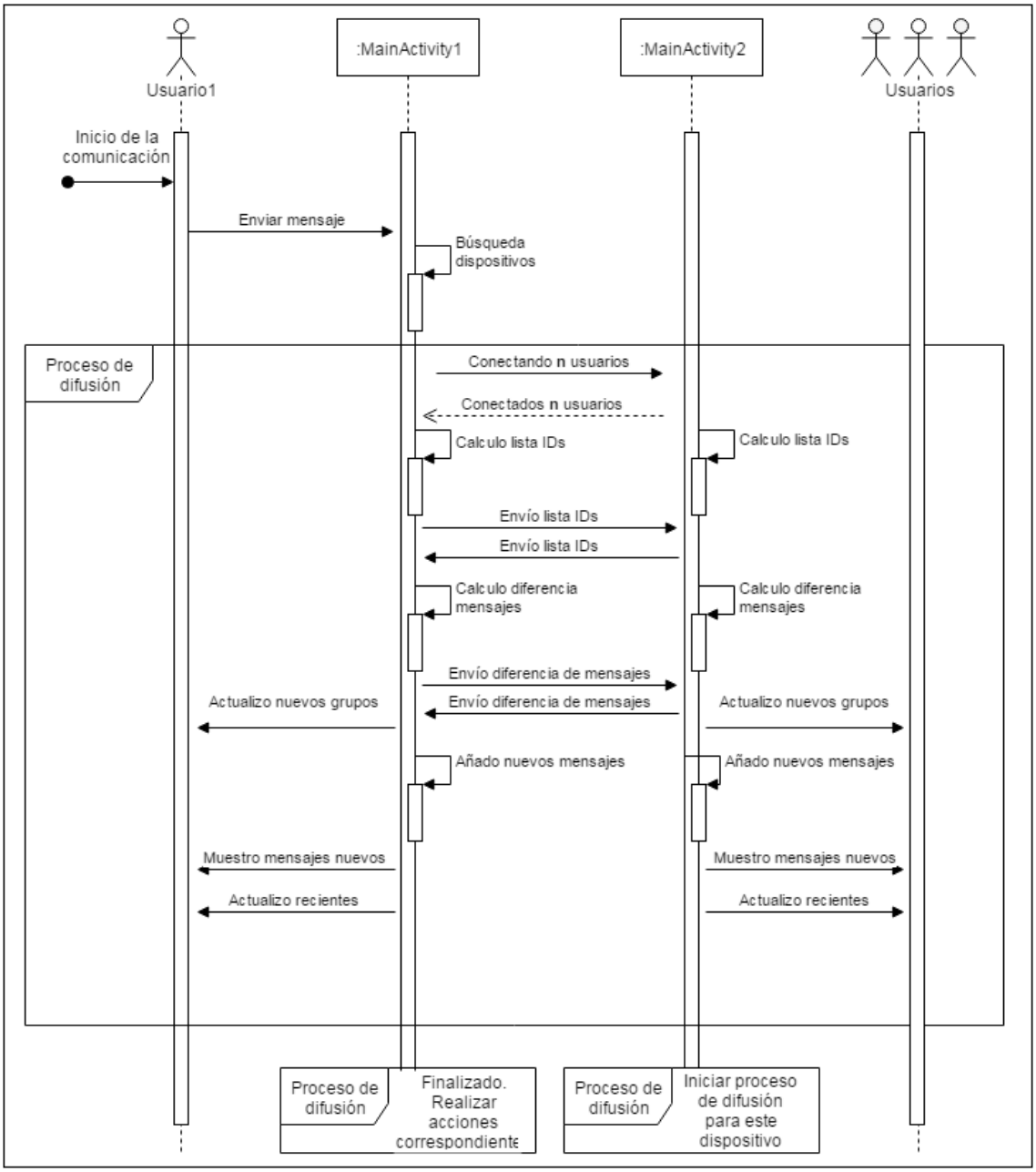

Ilustración 19 - Diagrama de secuencia

En la ilustración se muestra el proceso de conexión con los dispositivos encontrados durante el descubrimiento. Se ha iniciado este ejemplo con el envío de un mensaje en un chat, posteriormente se procede a escanear el entorno para obtener los dispositivos cercanos. En este punto se inicia el llamado "proceso de difusión" señalado en la imagen. A partir de este momento se realizan las acciones pertinentes para la creación de la piconet. De esta manera se crea un hilo de conexión con cada uno de los dispositivos descubiertos. Una vez conectado con todos los dispositivos (marcado en la imagen como una flecha discontinua "Conectados n usuarios"), se realizan los mismos pasos para cada uno de forma paralela. Se inicia el proceso de intercambio de identificadores para posteriormente enviarse mutuamente los mensajes exclusivos. El proceso acaba actualizando las interfaces. Por parte del emisor simplemente se realizan acciones de finalización, pero por parte de los receptores se inicia el mismo proceso realizado por el emisor para difundir el nuevo mensaje en la red.

4.5 Metodología de desarrollo

La metodología de desarrollo seguida para la elaboración del proyecto ha sido una mezcla de varias técnicas muy parecidas. Podemos simplificar todo ello en un llamado modelo en espiral. Este modelo es perfecto para reducir riesgos e introducir mejoras y nuevos requerimientos durante el proyecto.

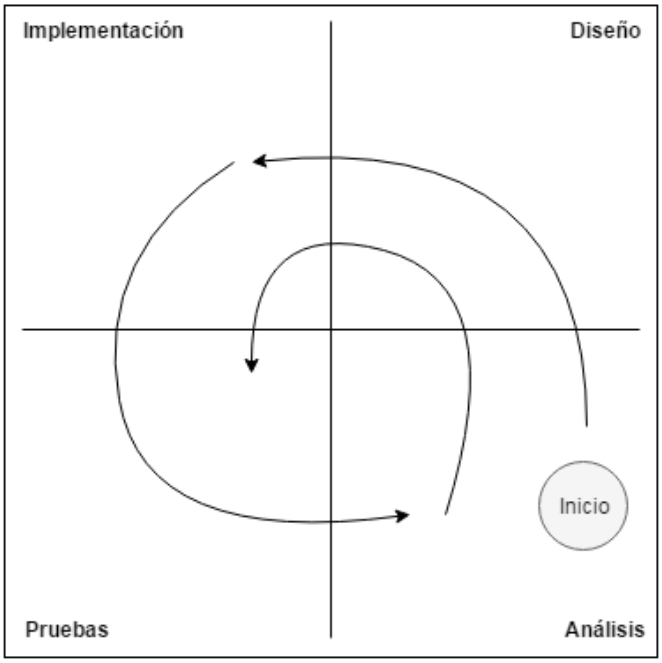

Ilustración 20 - Metodología de desarrollo

El modelo se compone de cuatro actividades por las que se van atravesando durante el desarrollo del trabajo. La forma en la que se dibuja el ciclo de vida del modelo es una espiral ya que hace referencia al orden de las etapas recorridas. Las definiciones de las diferentes actividades y por orden son:

1. Análisis: Estudio de los requerimientos de cada objetivo. Identificación de riesgos y sus alternativas de solución. Definición de los detalles funcionales.
2. Diseño: Con los datos de la etapa anterior, se diseña el sistema.
3. Implementación: Programación del diseño eligiendo un buen modelo de programación.
4. Pruebas: El proyecto hasta entonces es revisado. Se realizan diferentes pruebas a diferentes módulos o a un módulo completo para posteriormente comparar con los requerimientos descritos.

Con cada iteración de la espiral se crean sucesivas versiones de la aplicación, cada vez más completa. Al final de todas las iteraciones el sistema resultante es uno completamente funcional.

Como se ha comentado al principio del apartado el modelo en espiral es el método básico que se ha seguido, pero existen otros métodos semejantes por el que podemos identificar el proceso de creación. Para este otro método se creó un diario de tareas en el que se especificaban la realización de diferentes tareas por día. El nombre de este método es el llamado scrum.

La relación de la forma de trabajar en el proyecto y la metodología scrum se basa en la creación de periodos cortos de trabajo, generalmente de una semana (sprints en scrum), en la que se realizaban diferentes actividades referentes a la aplicación Android. Para la realización de estas actividades se hacía uso de otra herramienta característica de scrum, el tablero de tareas. En él se definen tres columnas diferenciadas llamadas: "Por realizar", "Realizándose" y "Realizadas". Así pues, durante el desarrollo, se establecían una serie de tareas simples para realizar en máximo una semana. Según el estado de estas actividades se clasificaban según las columnas definidas anteriormente.

Resumiendo, podemos definir el conjunto de todo el proceso como un modelo en espiral ayudado para la clasificación de tareas por scrum. De esta forma se ha conseguido una forma de desarrollo adecuada para todo el proceso de creación del proyecto.

4.6 Casos de uso

Los casos de uso muestran las distintas interacciones que el usuario puede llegar a realizar con la aplicación. Para diseñar los casos de uso debemos tener en cuenta una serie de características:

- Siempre son iniciados por el usuario y nunca desde el interior de la aplicación.
- Debe representar una acción completa desde el punto de vista del usuario.
- Debe completarse en un tiempo corto.
- Pueden participar varios usuarios.

Si estas características no llegasen a cumplirse o el desarrollador no pudiera llegar a imponerlas para un caso de uso, es recomendable dividir ese caso en varios independientes. Además de describir una funcionalidad, un caso de uso describe también una interacción con el usuario. Las descripciones de los casos siempre hacen referencia a lo que se espera de la interacción con el usuario y no a cómo realiza la tarea. Además, es recomendable tener en cuenta no solo las tareas básicas si no también tareas relacionadas con el mantenimiento de la aplicación.

A continuación, se muestran los diferentes casos de uso divididos en las cuatro principales actividades que representan las interfaces del menú y la actividad del chat.

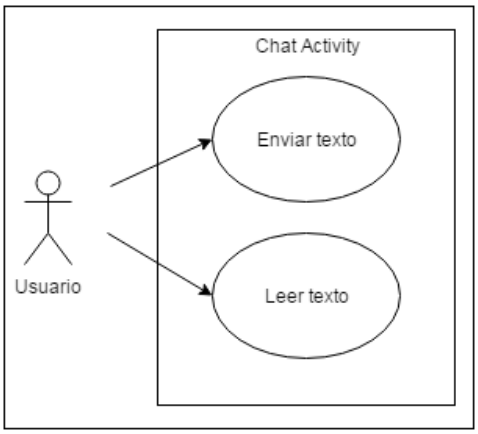

Ilustración 21 - Caso de uso Chat

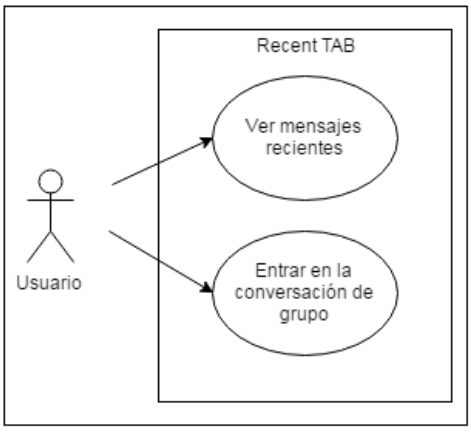

Ilustración 22 - Caso de uso Reciente

En las imágenes anteriores se muestra el chat de la aplicación. De la forma más básica el usuario podrá enviar y leer los mensajes. Ya en el menú, se observa que, en la pestaña recientes, además de visualizar los últimos mensajes, se puede acceder al chat del correspondiente grupo.

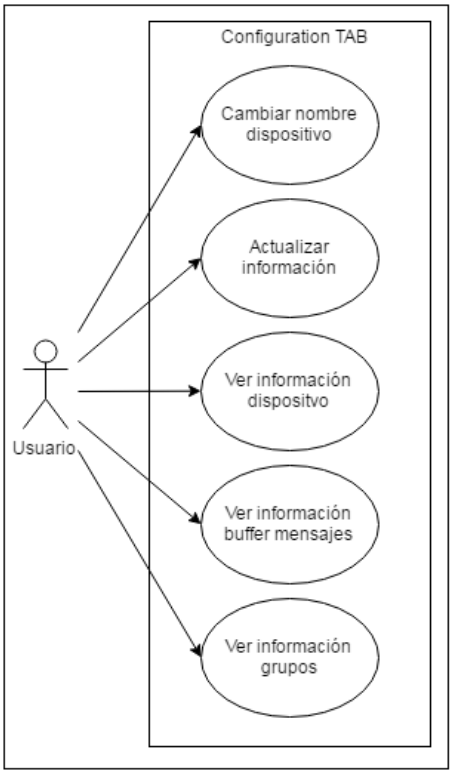

Ilustración 23 - Caso de uso Configuración

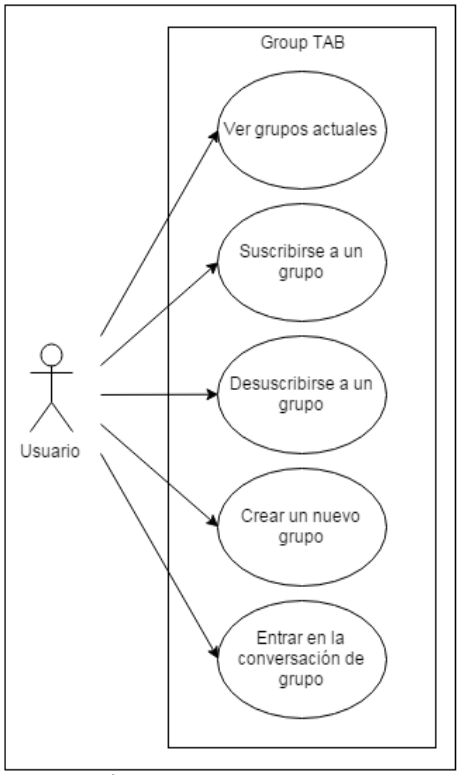

Ilustración 24 - Caso de uso Grupos

Las anteriores imágenes muestran las dos situaciones en las que el usuario puede realizar más acciones. En la parte de configuración el usuario puede ver toda la información referente a la aplicación. También podrá cambiar el nombre del dispositivo y actualizar toda la información mediante el botón correspondiente. Para la parte de grupos el usuario observa todos aquellos que la aplicación haya obtenido. Se le ofrece la posibilidad de suscribirse o cancelar la suscripción de los grupos, además de poder crear nuevos mediante el botón con el signo del mas mostrado en la parte inferior de la pantalla.

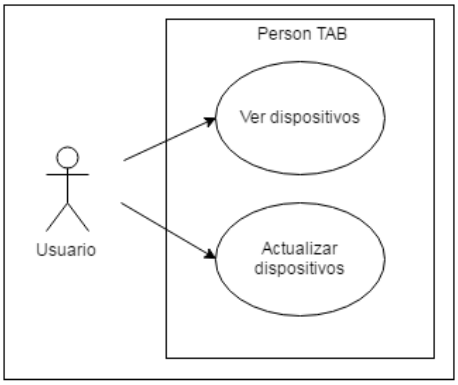

Ilustración 25 - Caso de uso Personas (Dispositivos)

Finalmente, en la pestaña de dispositivos (Person TAB), el usuario podrá ver los usuarios cercanos a él y con los que podrá interactuar en caso de mandar mensajes o recibirlos. Estos nodos se pueden actualizar pulsando también sobre un botón.

4.7 Interfaz de usuario

La interfaz de usuario es la forma con la que el usuario se comunica con la aplicación. Todas las interfaces del programa deben ser fáciles de entender y utilizar. Además, debido a que este proyecto se realiza en una plataforma móvil, es necesario que las interfaces estén bien diseñadas de forma que se adapten a la gran diversidad de dispositivos. Toda buena interfaz debe albergar mínimamente estas cualidades:

- Claridad: interfaz limpia.
- Familiaridad: elementos reutilizables.
- Consistencia: extrapolar el manejo de diferentes áreas de la interfaz.
- Eficiencia: debe ahorrarnos tiempo y esfuerzo.
- Concisión: especificación breve.
- Sensibilidad: rapidez, buen feedback.
- Estética: interfaces agradables.
- Errores: no castigar al usuario por sus errores.

Es recomendable, antes de diseñar las interfaces digitalmente, realizar un boceto a lápiz y papel que intente cubrir los distintos escenarios propuestos. Una vez satisfecho con el resultado, se procede a la realización de los bocetos o maquetas digitalmente. Existen multitud de herramientas para este fin. En este caso se ha escogido una herramienta online para la realización de todo tipo de esquemas e interfaces [12].

Un punto importante a destacar en la realización de interfaces en Android es la gran cantidad de dispositivos existentes. Cada uno de estos, presenta las interfaces de una aplicación de forma diferente por lo que es importante prestar atención en este paso. En el caso que nos ocupa se han seguido los pasos adecuados para poder ejecutar y visualizar adecuadamente la aplicación en cualquier dispositivo Android.

Las maquetas nos permiten visualizar el diseño de la interfaz de una manera muy aproximada a la versión final. En las siguientes ilustraciones se muestran las que se han realizado para el proyecto.

Las dos primeras pestañas son las de Reciente y Grupos. En la primera observamos cómo se ha propuesto el listado de los mensajes. Se ha creado un elemento rectangular en el que se incluye la persona autora del mensaje, el grupo en el que se ha mandado y el texto del mensaje en cuestión. También se ha asignado un icono representativo a cada fila que representa la primera letra del autor del mensaje.

Para la pestaña Grupos, se ha seguido la misma mecánica anterior (también para la pestaña Dispositivos). En esta ocasión se muestra el título del grupo y su descripción además del icono con la primera letra del nombre del grupo. También se aprecia el botón redondo con el símbolo mas (“+”) en la parte inferior derecha cuyo objetivo es el de añadir nuevos grupos al sistema.

Ilustración 26 - Interfaz Reciente

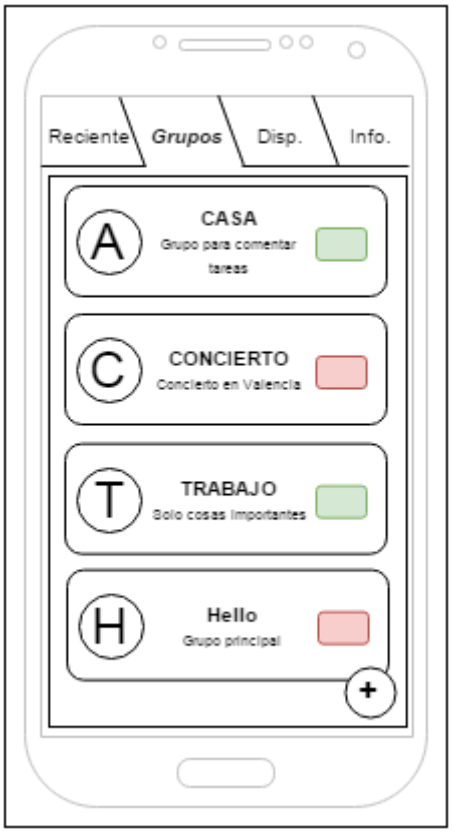

Ilustración 27 - Interfaz Grupos

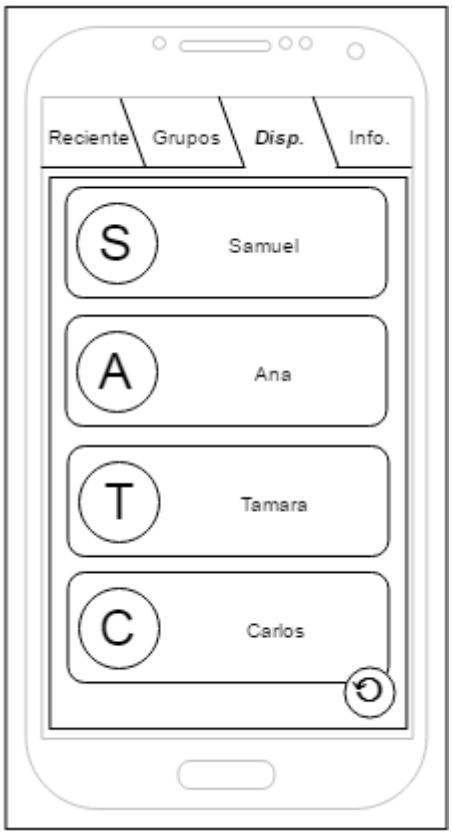

Ilustración 28 - Interfaz Dispositivos

Ilustración 29 - Interfaz Información

Las dos anteriores, muestran Dispositivos e Información respectivamente. En la primera de ellas se sigue el mismo patrón de elementos rectangulares con un símbolo representativo y un texto indicando que nodos han sido encontrados. Se observa en este caso el botón para actualizar la lista. En la imagen de la derecha, se muestra la pestaña de información. La idea es mostrar la información agrupada en diferentes áreas de la aplicación. Por ejemplo, en esta ilustración se muestra el estado del dispositivo, información del buffer y de los grupos. De la misma manera que en la parte de dispositivos, se puede actualizar la información mostrada con el botón inferior.

Ilustración 30 - Intefaz Chat

Finalmente, y la parte de la interfaz más importante es la referente al chat de la aplicación. La idea inicial era crear un chat simple y limpio en el que los usuarios puedan identificar claramente los mensajes de cada uno de los participantes, así como las opciones correspondientes. Se observan los mensajes de otros usuarios situados en la parte izquierda, mientras que los propios están en la parte derecha. En el inferior de la pantalla se sitúa la caja de texto y el botón de enviar.

Tras el diseño de las interfaces es conveniente analizarlas. Existen diferentes métodos para este fin, uno de ellos es pedir a personas ajenas al proyecto los pasos para la realización de tareas determinadas. Hay que tomar constancia de todas las indicaciones y ver en qué pasos se producen más errores o donde pierde más tiempo. Con este trabajo se puede rediseñar la interfaz adaptándola adecuadamente.

5. Resultados

En este punto de la memoria ya se han dado todas las claves de la elaboración del proyecto por lo que a continuación se presenta el funcionamiento de la aplicación creada funcionando en un dispositivo móvil. En este apartado veremos las consecuencias finales del desarrollo de la aplicación y se explica si estos resultados son buenos o no.

5.1 Ejemplos

La finalidad de este apartado es mostrar la aplicación funcionando en un dispositivo. Se muestran algunas capturas del dispositivo en las distintas partes de la aplicación. De esta manera se observa el resultado de la ejecución. Debido a que las entradas de la aplicación las proporciona el usuario, las imágenes mostradas contienen datos ya introducidos con el fin de estudiar las distintas posibilidades que ofrece el programa.

Ilustración 31 - Ejemlo Chat 1

Ilustración 32 - Ejemplo Chat 2

Las capturas anteriores muestran la parte de la aplicación correspondiente al chat. Donde se puede observar en la captura de la izquierda la distribución de los mensajes en una conversación. Los mensajes que el usuario envía se encuentran a la derecha y con un color de fondo oscuro. Los mensajes que se reciben se colocan a la izquierda y vienen con un color claro y diferente en el nombre según el usuario.

La captura de la derecha muestra el teclado con el cual nos comunicaremos con los demás usuarios. Una vez se introduzca el texto deseado en la caja de texto inferior, el usuario tiene que pulsar el botón con el icono del sobre para enviar el mensaje.

Ilustración 33 - Ejemplo Recientes

En la ilustración anterior nos encontramos en el menú principal, más concretamente en la pestaña de mensajes recientes. En la parte superior las pestañas se han representado por cuatro iconos diferentes. El icono se oscurece para representar la pestaña en la que nos encontramos actualmente además de estar subrayado por una línea gruesa. La imagen muestra la parte de los mensajes recientes. En ella se aprecia los diferentes elementos que se comentaron en el diseño de interfaces. También se puede mostrar un menú contextual para cada elemento de la lista realizando una pulsación larga, pero la funcionalidad no se ha implementado aún. Todo esto se implementa con un estilo común y colores básicos.

Ilustración 34 - Ejemplo Grupos

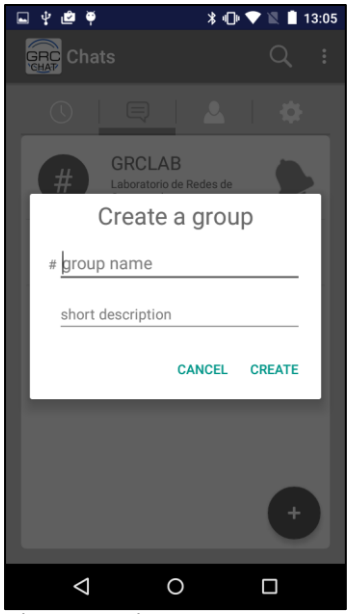

Ilustración 35 - Ejemplo Grupos creación

Para la parte de los grupos tenemos dos capturas. La captura de la izquierda nos muestra la interfaz en la que se encuentran los diferentes grupos en los que el usuario puede participar actualmente. En la imagen se observa que cada una de las filas pertenece a un grupo y que cada una contiene un icono de una campana. La campana representa la suscripción del usuario a ese grupo determinado. El usuario puede tocar la campana para suscribirse o cancelar la suscripción y se le indica mediante un color, siendo el más oscuro la suscripción y el color más claro el caso contrario.

En la captura de la derecha observamos la creación de un grupo que se inicia pulsado el botón inferior con el icono de un más. Al pulsar el botón se muestra un menú modal en el que se le pide al usuario el título y la descripción del grupo en cuestión.

La siguiente captura de la izquierda, hace referencia a la parte de dispositivos alcanzables. En ella se muestran los dispositivos encontrados hasta el momento. El tiempo que está el dispositivo descubriendo nuevos nodos está establecido. Durante ese tiempo se muestran aquellos dispositivos encontrados junto a un mensaje que indica que el proceso está en marcha.

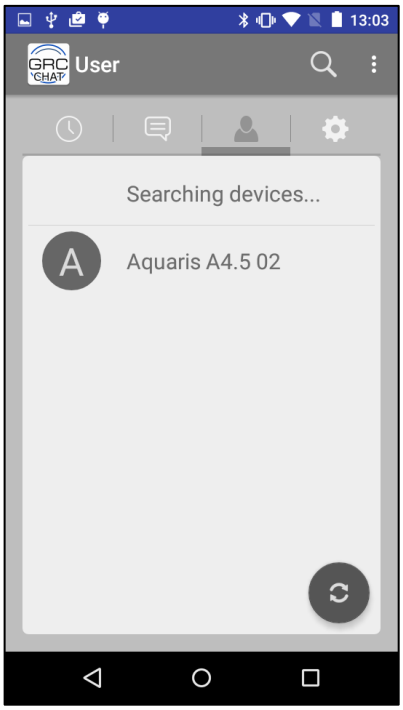

Ilustración 36 - Ejemplo Dispositivos

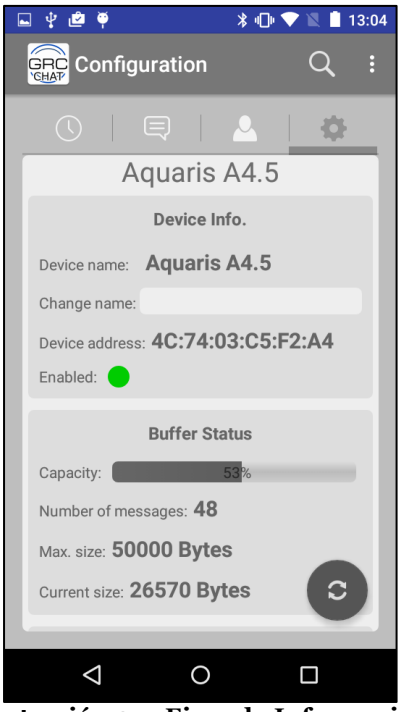

Ilustración 37 - Ejemplo Información

Finalmente, en la última captura se muestra pestaña de configuración e información. En ella no se aprecia toda la información que muestra la aplicación, solo una muestra. Los diferentes grupos de información se agrupan en una caja de color más oscuro que el fondo. En esta caja se listan diferentes datos sobre el grupo en el que están.

Algunos ejemplos que se observan en la captura de información son: se aprecia que en el “Buffer Status” la barra de capacidad está al 53% con un numero de mensajes de 48, que existe un tamaño actual de 26570 Bytes, que el dispositivo Bluetooth del aparato está actualmente funcionando y que el nombre del dispositivo es “Aquaris A4.5” (con posibilidad de cambiarlo).

5.2 Evaluación del sistema

Una vez completado el proyecto queda por realizar una parte fundamental. Como se ha comentado en esta memoria, la metodología de trabajo llevada a cabo ha sido un modelo en espiral, por lo que muchos fallos fundamentales se han ido corrigiendo durante la fase correspondiente. En esta metodología se realizan pruebas cada vez que se implementa un bloque de tareas, por lo que de esta forma se evalúa el sistema repetidamente hasta conseguir una versión funcional estable.

A pesar de realizar esta metodología siempre existe un límite, una meta en la que se tiene que finalizar el proyecto. En este apartado se exponen algunas opiniones sobre la evaluación final del proyecto. También se habla sobre los objetivos que se han llegado a cumplir y aquellos que no se han cumplido totalmente o simplemente no se han conseguido.

De la aplicación final podemos decir que tiene una estructura muy bien definida. Una buena estructuración de un sistema ahorra muchas incomodidades al usuario que la usa. La elección de un menú principal que engloba toda la funcionalidad de la aplicación puede no ser la elección más acertada de todas, pero tampoco resulta desconcertante como pueden haber sido otros métodos de mostrar la información.

Siguiendo con una evaluación de la interfaz, se observa que en todas las actividades la interfaz es muy sencilla, el usuario puede intuir perfectamente cuales son las acciones a realizar en cada una de ellas. Por este motivo se ha obviado cualquier tipo de información de ayuda que pueda molestar o distraer al usuario.

La elección de la distribución de la información parece ser acertada. Si se hubiera optado por juntar todos estos datos en una sola ventana, el usuario se sentiría abrumado y le costaría encontrar la acción que desease realizar.

Si nos enfocamos en la parte funcional de la aplicación observamos que existen ciertos aspectos que son candidatos a mejorar, la velocidad es uno de ellos. En una aplicación de este tipo la velocidad con la que se conecta a otros usuarios o la velocidad con la que se transmiten los datos es crucial. Generalmente se suele gastar mucho tiempo en mejorar este aspecto en proyectos de este tipo. A pesar de todo el sistema final presenta una velocidad aceptable, debido principalmente a la incorporación de la piconet.

Otro aspecto funcional es el rendimiento. En general el rendimiento de la aplicación es bueno. La transición entre las distintas partes de la aplicación se produce de manera suave y rápida. La visualización de textos, carga y guardado y envío de la información dan buenos resultados cuando se realiza de manera exitosa.

La estabilidad de la aplicación es quizás el aspecto en el que mayor tiempo habría que dedicarle a mejorar. En proyectos en los que se utiliza la red formando conexiones con otros sistemas, se producen muchos errores. Todo esto es debido a que las actividades realizadas relacionadas con la red son muy inestables en comparación con otros conceptos. Aun en esta última versión de la aplicación se producen cierto tipo de errores y la mayoría están relacionados con problemas derivados de la red. Es necesario elaborar un sistema más completo y complejo para poder administrar todos estos errores y poder darles solución sin que ello tenga repercusión en la experiencia final del usuario.

5.2.1 Cumplir objetivos

Si se observan los objetivos principales del proyecto redactados en los primeros apartados, observamos que se han cumplido los tres principales objetivos propuestos. Esto es un éxito total sobre las expectativas iniciales.

Los tres objetivos hacen referencia simplemente al intercambio de mensajes entre dos dispositivos mediante la tecnología Bluetooth. Si nos centramos exclusivamente en la funcionalidad descrita se cumplen los objetivos, sin embargo, si se empieza a matizar algunos aspectos es indudable que existan mejoras que implementar.

A continuación, observamos los objetivos secundarios, redactados como extensión a los principales. El primer objetivo secundario propuesto es la gestión de grupos. Esta idea se ha fusionado de tal manera en el proyecto que ha podido pasar en cualquier momento como objetivo primario.

Los objetivos secundarios referentes al menú también se han conseguido implementar en su totalidad. Si bien es cierto que la interfaz resultante no es perfecta, los objetivos referentes a la visualización de los dispositivos de la red y la información del sistema han sido cumplidos.

Otro objetivo que quizás se puede dar por hecho en una aplicación de este tipo es el manejo de los mensajes en cualquier situación. En las primeras versiones de la aplicación, esta solo recibía y enviaba mensajes cuando el usuario se encontraba en una ventana de chat. Sin embargo, debido a la necesidad de actualizar las diferentes interfaces, se trasladó toda esta funcionalidad a la actividad central del programa. Este objetivo está sujeto a una mejora futura que consiste en la obtención de los mensajes incluso cuando el usuario haya cerrado la aplicación. Esto es posible delegando la funcionalidad necesaria para este fin a una tarea del sistema Android, ejecutada constantemente.

Finalmente, el objetivo de implementar las mayores características propias de aplicaciones similares ha sido cumplido en parte. Existe una funcionalidad básica, sin embargo, hay muchas otras características que son útiles y que en la versión actual de la aplicación no se han implementado. Este tipo de características como puede ser la búsqueda de mensajes, la difusión a diferentes grupos de un mensaje o personalización de la interfaz, se pueden programar en futuras versiones de la aplicación para ofrecer al usuario una mejor experiencia.

5.2.2 Pruebas de rendimiento de comunicación

Para evaluar el rendimiento del tipo de aplicaciones basadas en contactos utilizando tecnología Bluetooth, se han realizado una serie de medidas en una situación en la cual se establece una conexión entre dos dispositivos transmitiendo diferentes tipos de datos. Estas medidas se realizaron para la elaboración del artículo [13], estudio previo al desarrollo de la aplicación de este proyecto. Para este caso, la versión de la aplicación consistía en una interfaz básica de comunicación utilizando un modelo básico de cliente-servidor sin ningún añadido más.

El objetivo de estas mediciones es obtener el tiempo de entrega de los mensajes dependiendo del tamaño del mensaje en diferentes distancias. Para ello se enviaron 500 mensajes de un dispositivo a otro. Cada medición comprende los tres pasos básicos

del envío: la conexión con el dispositivo remoto, el envío del mensaje y el cierre de conexión. La nomenclatura utilizada para las distancias se resume en:

- Dispositivos juntos: Distancia aproximada 10cm.
- Dispositivos distancia media: Distancia aproximada 5m.
- Dispositivos alejados: Distancia aproximada 10m.

A continuación, se mostrarán las gráficas obtenidas para cada una de las tres distancias propuestas en cada uno de los diferentes tamaños: 375Bytes representando un mensaje corto, 109KB representando una imagen de baja resolución y 11MB representando un video corto o una imagen de alta resolución.

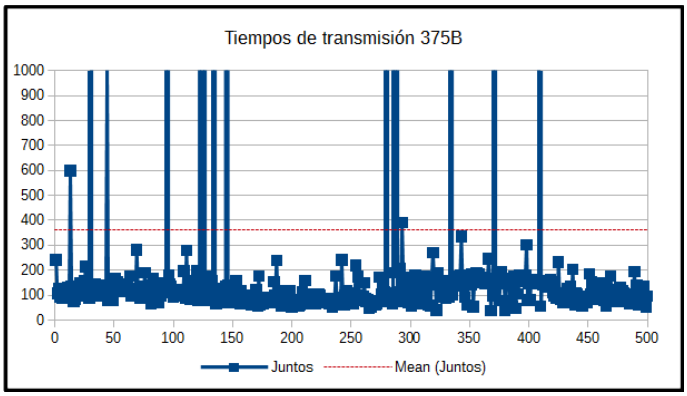

Ilustración 38 - 375Bytes Distancia corta

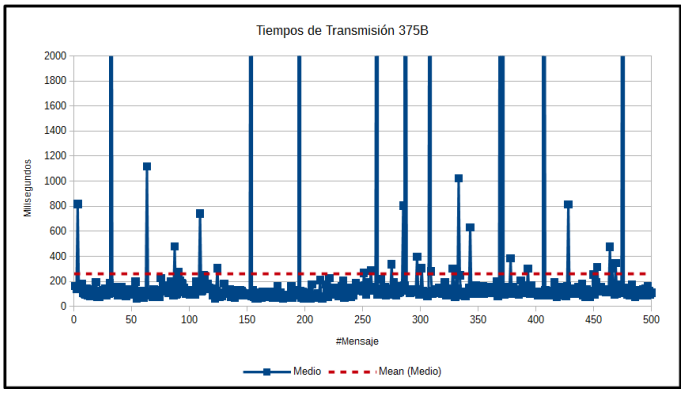

Ilustración 39 - 375Bytes Distancia media

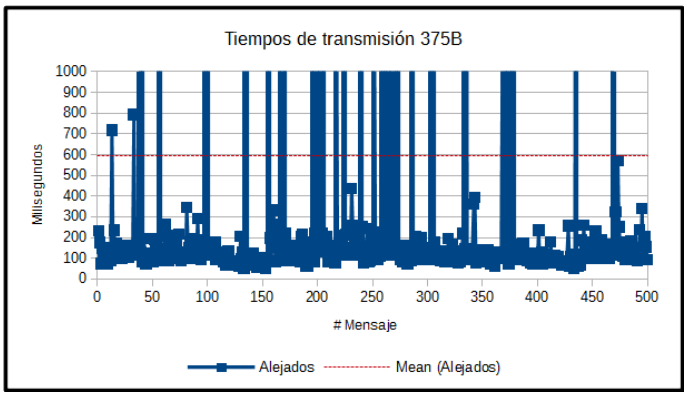

Ilustración 40 - Distancia larga

Las tres primeras gráficas presentadas pertenecen a las pruebas efectuadas para el envío de un mensaje de tamaño 375Bytes. El eje de los milisegundos está limitado a 1000ms o 2000ms, para apreciar mejor los datos más representativos. En este tipo de mensaje de menor tamaño se observa claramente la influencia que tiene la distancia. Se observa la gran variación de tiempos existente entre la situación en la que los dispositivos se encuentran más alejados, produciendo una gran inestabilidad.

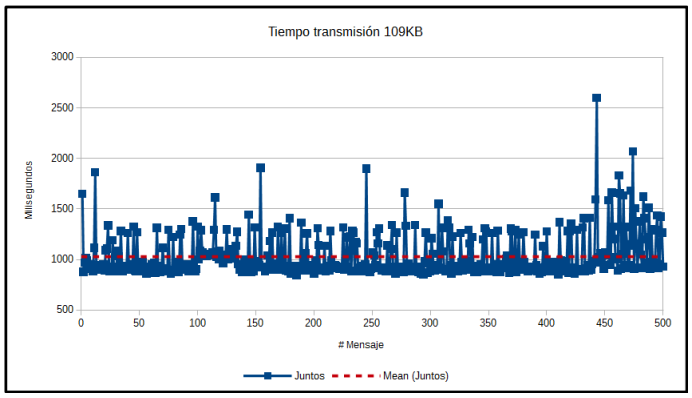

Ilustración 41 - 109KB Distancia corta

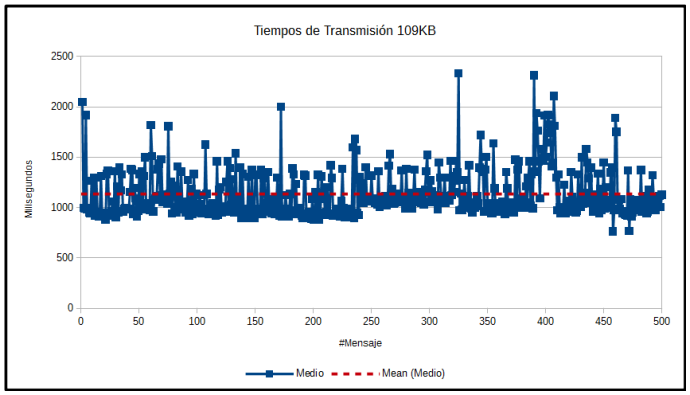

Ilustración 42 - 109KB Distancia media

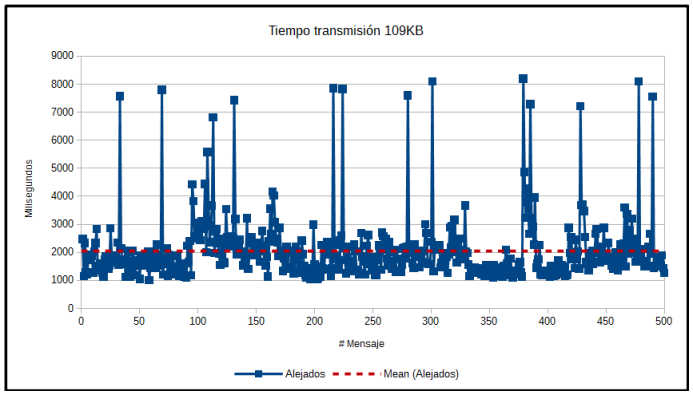

Ilustración 43 - Distancia larga

Para este caso del mensaje de 109KB, se observa que en el caso de la distancia más alejada las variaciones se reducen. Esto es debido a que el tiempo de envío ha aumentado debido a que también lo ha hecho el tamaño del mensaje, enmascarando así los posibles problemas surgidos durante la conexión inicial que se veían con más claridad en el caso del mensaje de 375B.

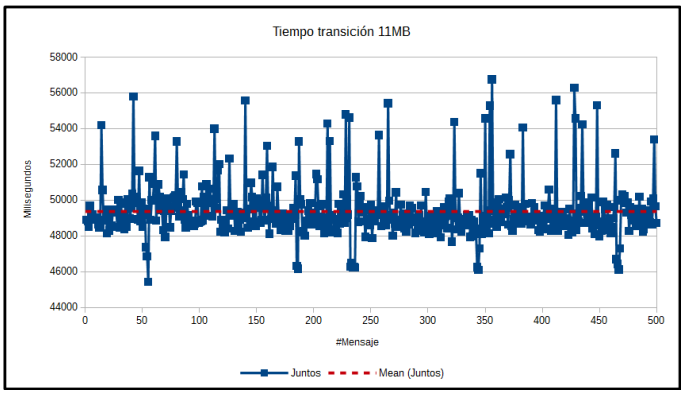

Ilustración 44 - 11MB Distancia corta

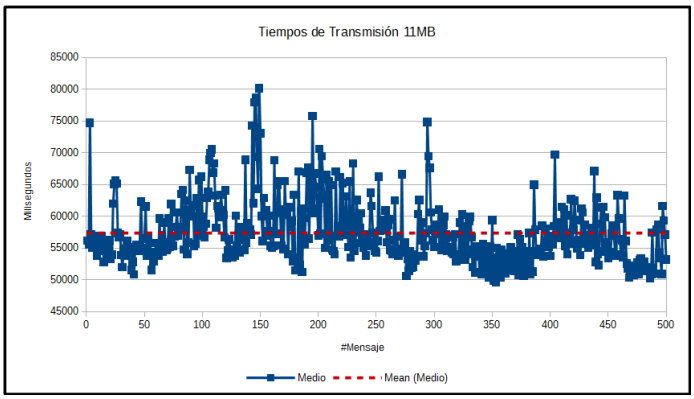

Ilustración 45 - 11MB Distancia media

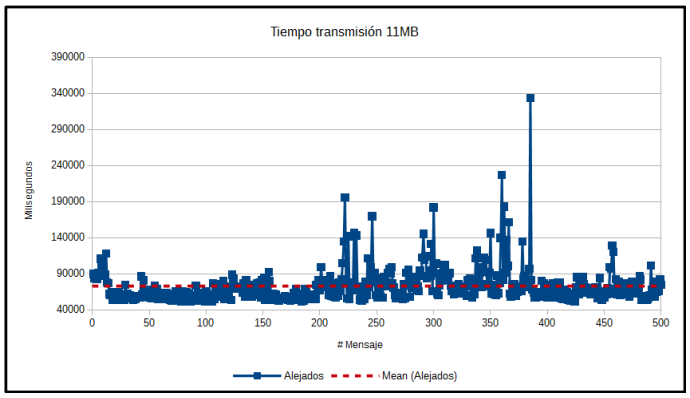

Ilustración 46 - Distancia larga

Finalmente, para el caso del mensaje de mayor tamaño (11MB), se observa una línea de tiempos mucho más estable para cada uno de los casos. Sin embargo, aún se aprecian grandes variaciones llegando a casos extremos como el punto máximo observado en la última gráfica.

En general se observa que el tiempo en los mensajes pequeños tienen una gran variabilidad debido principalmente al tiempo de conexión de los dispositivos, que tiene un impacto menor en los mensajes de gran tamaño. Los resultados para distancias medias y cortas son muy similares. Cuando la distancia es grande (más grande que el rango de operatividad del Bluetooth, que es 7m), el tiempo medio de envío se incrementa especialmente para mensajes cortos, debido a los problemas surgidos durante la conexión y retransmisión, afectando gravemente al rendimiento.

En la tabla siguiente se resumen en gran medida los resultados obtenidos en cada una de las situaciones:

	Media	Mínimo	Máximo	Q1	Q3
375B					
Juntos (10cm)	360.44	38	30125	94	150
Medio (5m)	262.89	60	9356	97	140
Lejos (10m)	596.47	51	62139	95	161
109KB					
Juntos (10cm)	1028.11	848	2598	901.75	1073
Medio (5m)	1134.40	762	2334	970	1275.75
Lejos (10m)	2046.31	1008	8198	1446	2238.25
11MB					
Juntos (10cm)	49372.12	45434	56750	48663.50	49714.50
Medio (5m)	57348.41	49623	80178	54107	59559.25
Lejos (10m)	72679.01	51205	333855	58509.25	77950

Ilustración 47 - Resumen conectividad. Todos los valores en milisegundos

Las columnas con nombres "Q1" y "Q3", representan los cuartiles uno y tres respectivamente para los datos presentados. Estos valores ayudan a hacernos una idea de cómo los valores están distribuidos.

5.2.3 Pruebas de rendimiento de la aplicación

La realización de pruebas de rendimiento en un sistema nos permite determinar lo rápido que se realiza una tarea en ese sistema en condiciones particulares de trabajo. Las pruebas ayudan a verificar otros atributos de la calidad del sistema como la fiabilidad o el uso que se realiza de los recursos.

En el caso que nos ocupa, las pruebas realizadas han sido la toma de tiempos para el funcionamiento normal de la aplicación. Es necesario realizar un estudio previo a la toma de tiempos para verificar que es interesante medir y por lo tanto en qué puntos de la aplicación es necesario introducir los puntos de medición para obtener las medidas deseadas.

En la aplicación del proyecto se han considerado dos medidas que son de interés para un estudio posterior. Una de ellas es el tiempo de conexión de un dispositivo, es decir, el tiempo que tarda un nodo en realizar la conexión con el nodo remoto. La otra medida interesante es el tiempo de envío de un mensaje de un nodo a otro. Hay que recalcar que estas medidas son específicas de la aplicación desarrollada para este proyecto y que no son significativas para todas las aplicaciones de mensajería.

En la imagen se muestran en realidad tres medidas: el tiempo de conexión para el dispositivo emisor además de su tiempo de emisión del mensaje y el tiempo de emisión del mensaje por parte del dispositivo receptor ya que éste también manda información referente a los mensajes que almacena en su sistema. Las medidas han sido realizadas para el envío de 50 mensajes en una situación estándar, es decir, estando los dos dispositivos cerca uno de otro. Cada envío se realiza con un solo mensaje. Los mensajes

tienen una longitud variable y son del tipo que se suelen dar en un tipo de conversación habitual. Pese a que los mensajes enviados en estas situaciones suelen tener una longitud relativamente corta, aunque las pruebas se realizaran con una longitud de mensaje mayor, el tiempo medio de envío no cambiaría mucho al representado en estas pruebas debido a que este proceso incluye otras acciones a tener en cuenta. A continuación, se muestra una gráfica en la que observan las medidas descritas anteriormente:

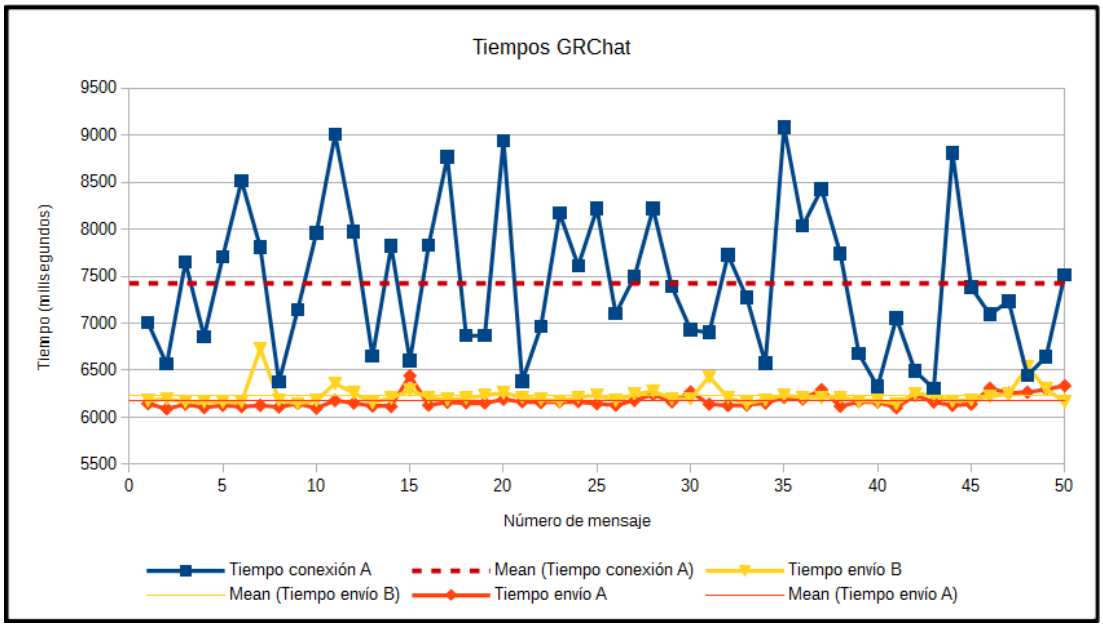

Ilustración 48 - Tiempos aplicación

Analizando la gráfica se pueden llegar a diferentes conclusiones. Lo primero que se observa es que los tiempos de envío de mensajes para uno y otro dispositivo son prácticamente idénticos destacando algunos pequeños picos sin importancia. El tiempo medio de estos envíos se sitúa en torno a los 6,2 segundos, Durante este tiempo se envían todos los identificadores de los mensajes del dispositivo y a continuación los mensajes cuyo dispositivo remoto no tenga en su sistema. Que los tiempos de envíos sean prácticamente iguales nos hace pensar en este proceso, la mayor parte del tiempo se realiza tareas del sistema Android y no trabajo referente al propio envío de la información. Debido a ello existe una posibilidad de mejora de reducir el tiempo de envío.

El tiempo de conexión se ha obtenido para el dispositivo emisor del mensaje. Lo primero que se observa son las grandes variaciones de tiempos que existen entre un mensaje y otro. Esto es consecuencia de la naturaleza propia de Bluetooth y del propio funcionamiento de la piconet. La tecnología Bluetooth trabaja en emisiones de radiofrecuencia por lo que es prácticamente normal que en ocasiones se produzca algún retraso en la conexión con otros dispositivos. En la piconet, el tiempo de conexión varía conforme al número de dispositivos a los cuales el sistema debe conectarse. Si en el entorno se encuentran varios dispositivos, el protocolo realiza el paso de intento de conexión con cada uno de los dispositivos por lo que el tiempo de conexión puede variar. Sumando estas dos principales causas obtenemos la variación de tiempos que se observa en la gráfica para el tiempo de conexión de un dispositivo cuya media se sitúa en torno a los siete segundos y medio.

Finalmente se puede realizar un cálculo básico para obtener el tiempo total que tarda un dispositivo en enviar un mensaje a los nodos disponibles en su entorno. Este cálculo se efectúa sumando el tiempo de conexión con el tiempo de envío del mensaje. Si se realiza, se obtienen 13,6 segundos de envío total de un mensaje.

6. Conclusiones

En esta parte final de la memoria se habla sobre la descripción del desarrollo junto a las conclusiones finales. Finalmente se complementa con un punto muy importante como es los trabajos futuros, donde se exponen las mejoras que nuestra aplicación puede llegar a tener.

6.1 Desarrollo del proyecto

La elaboración del proyecto ha constituido un objetivo principal en los últimos meses. Se ha aprendido mucho sobre diferentes conceptos de Android, principalmente sobre Bluetooth. Además, se ha dado uso a multitud de funcionalidad adicional y desarrollados conceptos de la creación y administración de redes.

El estudio previo a la implementación de las soluciones ha sido arduo, ya que una gran parte del tiempo se ha dedicado a la investigación y el estudio de la funcionalidad de la API de Android. También han sido un gran apoyo los problemas similares solucionados por otras personas que son expuestos en diversas páginas y foros de internet.

6.1.1 Duración

La duración del proyecto ha sido aproximadamente de 5 meses. En este intervalo están también incluido el tiempo dedicado a la elaboración de esta memoria y su correspondiente documentación durante el proceso de desarrollo de la aplicación. Además, hay que tener en cuenta el tiempo dedicado a la investigación y estudio de las características de la tecnología usada.

6.1.2 Problemas encontrados

Los problemas encontrados durante la elaboración del proyecto son diversos. Uno de los problemas que tiene desarrollar con Android es la utilización de elementos que en el mismo momento del desarrollo están desactualizados, por lo que no es recomendable su utilización por lo que pueda pasar. En ese momento se tiene que encontrar una solución que realice la misma actividad con elementos actualizados.

Otro problema que además es muy recurrente, es la infinidad de maneras que existe en Android de realizar una misma actividad. Puede no parecer un problema, pero si desarrollas una forma de realizar una tarea que no funciona con el resto de tu aplicación pierdes el tiempo en buscar información para corregir esos problemas.

El problema más grande al que nos hemos encontrado tiene relación con la principal causa de la aplicación. El sistema de comunicación de Android con Bluetooth, se basa en un sistema básico de cliente y servidor. Dada estas herramientas básicas el gran problema a enfrentarse es la creación de un sistema que maneje esta conexión según nuestras necesidades. Durante la memoria se ha explicado cómo se ha llevado a cabo esta solución, pero no se han expuesto los problemas surgidos durante ese desarrollo. El principal está relacionado en poder manejar cada uno de los problemas surgidos en la conexión. Mediante el manejador comentado en la memoria se han puesto solución a estos errores fortuitos, sin embargo, aún queda mucho trabajo para que la aplicación funcione de la mejor manera posible.

6.2 Conclusiones finales

Con la elaboración del proyecto se han realizado una serie de tareas y métodos que ayudan a entender mejor el proceso adecuado que hay que realizar a la hora de la elaboración de un proyecto de estas características. Los conocimientos obtenidos referente a el funcionamiento de una conexión punto a punto, estudiando todos sus pasos, han sido muy diversos y de gran utilidad.

En este caso se ha observado que debe existir un proceso largo de documentación antes del inicio del desarrollo de software. Hay que convertirse en un experto en el área de la aplicación del proyecto, y esto sólo se consigue estudiando el trabajo de otros expertos. Aplicando todas las técnicas estudiadas se puede llegar a elaborar una idea clara de la aplicación futura.

El área de desarrollo de Android es un terreno difícil ya que se encuentra en constante cambio. Por este motivo es necesario que el desarrollador invierta una cantidad de tiempo considerable en el estudio y seguimiento de esta tecnología. Aunque el proyecto tratase del estudio del funcionamiento de la comunicación entre dispositivos Android, no hay que olvidarse de que para la elaboración de esta meta se ha invertido una gran cantidad de tiempo en el estudio de la plataforma Android. Todo este tiempo invertido no es tiempo desperdiciado ya que se aprende una tecnología que actualmente está muy solicitada.

6.3 Contribuciones

Previa a la elaboración de la aplicación de mensajería se realizó un estudio para evaluar la difusión de los mensajes entre dispositivos móviles basado en contactos oportunísticos:

Enrique Hernández-Orallo, David Fernández-Delegido, Andrés Tomás, Jorge Herrera-Tapia, Juan-Carlos Cano, Carlos T. Calafate, Pietro Manzoni. GRChat: A Contact-based Messaging Application for the Evaluation of Information Diffusion. INFOCOMP 2016.

Se desarrolló una aplicación que usaba el protocolo Bluetooth en Android más básico posible y en la cual se obtenían diferentes resultados de eficiencia de los mensajes enviados en relación a su tamaño y a la distancia entre dispositivos.

Este trabajo previo sirvió como avance para la aplicación final presentada en este proyecto. Se desarrolló la base en la que se sustenta el sistema de comunicación actual y se determinaron ciertas decisiones por las que previamente se encaminaría el trabajo actual.

6.4 Trabajos futuros

En este apartado se proponen algunas mejoras que puedan complementar al proyecto presentado. Se redactan algunos aspectos a perfeccionar del proyecto que por tiempo o conocimiento no se han podido implementar. Las mejoras están divididas en tres grupos.

Referente a la interfaz:

- Mejorar la interfaz de chat, añadiendo un fondo característico.
- Mejorar el menú de inicio, para cada una de las pestañas.
- Mostrar más información de los mensajes que se reciben en el chat.

Referente a la funcionalidad:

- Mejorar la pestaña de información y configuración. Mejorar la manera en la que se presenta la información. Añadir funcionalidad extra como poder cambiar el tamaño del buffer.
- Añadir todas las características descritas de las aplicaciones de este tipo. Difusión a diferentes grupos, silenciar grupos, etc.

Referente a la implementación:

- Mejorar la difusión. Actualmente tal y como esta implementada la difusión se pueden producir conexiones innecesarias.
- Mejorar la conexión con los diferentes dispositivos. Mejorar la velocidad de conexión, así como la velocidad de envío y recepción de información.
- Implementar un mínimo de seguridad en la aplicación.
- Implementación del módulo de obtención de mensajes para que siga funcionando incluso el usuario haya cerrado la aplicación.

A. Manual de usuario

Tras la instalación de la aplicación, aparecerá en el menú de Android el icono de la aplicación para poder iniciarla. Tras pulsar sobre el icono, se iniciará el menú principal de la aplicación. En la siguiente imagen se observa la pantalla que aparece cuando se acaba de instalar limpiamente la aplicación.

Ilustración 49 - Manual icono

Ilustración 50 - Manual menú recientes

Las pestañas están numeradas del 1 al 4, e iremos explicando el uso de cada una de ellas a continuación. La pestaña número 1 corresponde a los mensajes recientes y es la que se presentará por defecto al iniciar la aplicación. El usuario podrá ver este menú al pulsar sobre el icono o deslizarse entre las pestañas. El número 5 es un chat, éste en concreto es el chat por defecto que se crea al instalar la aplicación. Si pulsamos sobre el elemento 5, la aplicación nos dirige a la ventana de chat que hablaremos más adelante. El número 6 corresponde a la búsqueda y configuración de la aplicación, estas características están incluidas, pero no se han llegado a implementar en la aplicación.

Si pulsamos sobre el número 2 nos redirige a la parte de gestión de grupos. En esta ventana se observan tres puntos clave. El primero de ellos (2.1), realiza la misma acción que el punto 5 de la anterior ilustración, inicia una conversación del grupo indicado por el usuario pulsando sobre él. El punto 2.2 muestra un icono de una campana que sirve para indicar la suscripción del grupo, en la imagen se muestra que el usuario está suscrito al grupo. Pulsando el icono, el usuario cancela la suscripción y el icono se mostrará de esta forma:

Ilustración 51 - Manual suscripción

Ilustración 52 - Manual grupos

Finalmente, el punto 2.3 es el botón correspondiente para añadir grupos. Al pulsarlo se muestra una ventana modal para crear el grupo en el que indicaremos el título y una pequeña descripción. En la imagen de la derecha se observa un ejemplo: "Compras" es el título, mientras que "Chat para las compras" es la descripción. Al acabar de introducir los datos pulsaremos sobre "Create". Se observa entonces como el nuevo grupo se ha añadido a la lista de grupos con la suscripción activa por defecto.

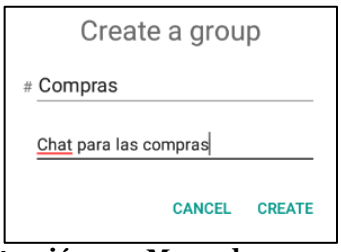

Ilustración 53 - Manual crear grupo

Ilustración 54 - Manual nuevo grupo

Si nos dirigimos al punto 3 de la primera imagen del apartado, nos encontraremos en la ventana de dispositivos. En esta parte, simplemente se nos muestran los dispositivos que se han encontrado en la red. Si pulsamos sobre el botón 3.1, la lista se borra y se nos indica mediante un mensaje "Searching devices…" que se está realizando la búsqueda. Durante el tiempo que dura el descubrimiento de nodos, se van añadiendo dinámicamente a la lista los que se vayan encontrando. Al finalizar el proceso, se ocultará el mensaje de "Searching devices…" y quedarán sólo los nodos listados en elementos por filas.

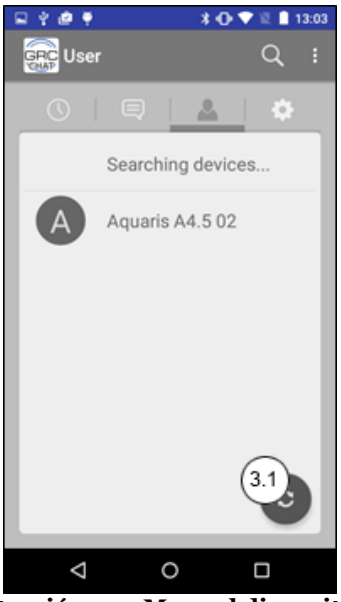

Ilustración 55 - Manual dispositivos

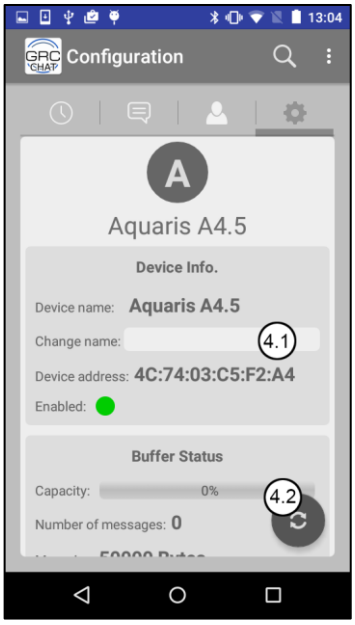

Ilustración 56 - Manual información

La ultima pestaña representada por el número 4 de la primera imagen, muestra la información importante de la aplicación. En esta ventana se puede cambiar el nombre pulsando sobre la caja de texto marcada por el punto 4.1. Al hacerlo se mostrará el teclado en pantalla en el que podremos introducir el nombre. Para actualizar el nombre debemos pulsar sobre el botón de actualización marcado como 4.2. Este botón además de actualizar el nombre, también verifica si la información que se muestra ha cambiado y si es así se actualizará por esta.

Para la ventana del chat tenemos varios elementos importantes. El grupo corresponde al recientemente creado “Compras”. El punto A muestra la barra de texto en la que el usuario tiene que introducir el mensaje. Pulsando sobre ella se muestra el teclado para que se pueda introducir el texto. Una vez acabado de teclear el mensaje, se tiene que pulsar sobre el botón B, mostrado con el icono de un sobre. Se inicia entonces el proceso de envío a los nodos descubiertos en la red. El mensaje introducido se mostrará inmediatamente en pantalla como se indica en C, y se observará un icono animado (D) que indica que la aplicación está realizando algún trabajo. De este modo funciona la parte del chat. Si se recibiese algún mensaje de otro dispositivo del mismo grupo, se mostraría automáticamente en la misma pantalla.

Ilustración 57 - Manual chat

Si a continuación nos dirigimos a la pestaña de mensajes recientes, se observará que el último mensaje reciente para el grupo “Compras” es el que acabamos de mandar. Además, en la pestaña de información, si actualizamos, se mostrará nueva información con el nuevo mensaje incluido en el buffer. En la imagen mostrada, se muestran 2 mensajes en la aplicación, referentes al recién mandado y al mensaje de bienvenida del programa.

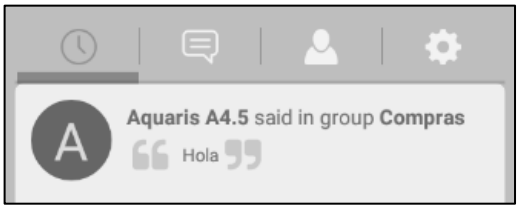

Ilustración 58 - Manual nuevo mensaje

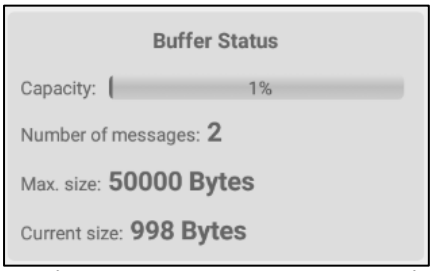

Ilustración 59 - Manual buffer actualizado

B. Lista de clases y métodos

Los métodos más importantes de las clases más importantes están listados a continuación:

Buffer
+ getLastMessageInGroup(Group): Message
+ isMessageInBuffer(Message): boolean
+ addMessage(Message): boolean
+ removeMessagesWithOldTTL():
+ getNumberOfBytesMessage(Message): int

MainActivity
+ writeDifferenceMessagesDevice(ArrayList<String>):
+ updateNewGroups(ArrayList<Message>):
+ getLastMessageNoDebug(): Message
+ getBufferWithoutDebug(): ArrayList<Message>
+ getDifferenceOfIDs(ArrayList<String>): ArrayList<Message>
+ getChatHistoryAdapterToSendID(): ArrayList<String>

MainChatActivity
+ setupChat():
+ loadMessagesThisGroup():
+ associateNameToColor(Message):
+ getColorByName(String): int
+ setTextInterface(Message):
+ printAllNewMessages(ArrayList<Message>):

Piconet
+ startPiconet(ArrayList<BluetoothDevice>):
+ getConnectedSocket(BluetoothDevice, UUID): BluetoothSocket
+ connect(BluetoothDevice): boolean
+ bluetoothBroadcastMessageID(ArrayList<String>):
+ bluetoothBroadcastDifferenceMessages(ArrayList<Message>):
+ sendDifferenceMessages(String, ArrayList<Message>):
+ sendMessageID(String, ArrayList<String>):

Bibliografía

[1] Yufeng Wang, Athanasios V. Vasilakos, Qun Jin, Jianhua Ma. Diciembre 2013. Survey on mobile social networking in proximity (MSNP): approaches, challenges and architecture

[2] http://opengarden.com/firechat/

[3] http://www.meshme.co/

[4] S. Ferretti, Shaping opportunistic networks," Computer Communications, vol. 36, pp. 481{503, 2013.

[5] F. Warthman, Delay-and Disruption-Tolerant Networks (DTNs) A Tutorial, version 2.0," The InterPlaNetary (IPN) Internet Project. InterPlanetary Networking Special Interest Group (IPNSIG), 2012.

[8] A. Vahdat and D. Becker, Epidemic routing for partially connected ad hoc networks,"Technical report number CS-200006, Duke University, pp. 1{14, 2000.

[9] B. Poonguzharselvi and V. Vetriselvi, Survey on routing algorithms in opportunistic networks," 2013 IEEE International Conference on Computer Communication and Informatics, pp. 1{5, 2013.

[10] L. Pelusi, A. Passarella, and M. Conti, Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks," IEEE Communications Magazine, pp. 134{141, 2006.

[19] E. Hernández-Orallo, J.-C. Cano, C. T. Calafate, and P. Manzoni, A representative and accurate characterization of inter-contact times in mobile opportunistic networks," Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems - MSWiM '13, pp. 309{316, 2013.

[12] https://www.draw.io/

[13] Enrique Hernández-Orallo, David Fernández-Delegido, Andrés Tomás, Jorge Herrera-Tapia, Juan-Carlos Cano, Carlos T. Calafate, Pietro Manzoni. GRChat: A Contact-based Messaging Application for the Evaluation of Information Diffusion. INFOCOMP 2016.