P2P SHARING 2.0: APLICACIÓN ANDROID P2P PARA COMPARTICIÓN DE ARCHIVOS

TRABAJO FIN DE GRADO
CURSO 2018-2019

AUTOR
JULIO GALILEA MORENO

DIRECTOR
PABLO RABANAL BASALO

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

DEDICATORIA

No importa cuán estrecho sea el camino,
cuán cargado de castigos el viaje…
Soy el amo de mí destino,
soy el capitán de mí alma.

William Ernest Henley,
Invictus.

AGRADECIMIENTOS

A mi familia, por todo el apoyo y confianza que me han dado durante este viaje,

a mis amigos cercanos, por saber aguantarme en los momentos de estrés durante la carrera,

gracias.

RESUMEN

El proyecto P2P Sharing 2.0 ha consistido en la elaboración de nuevas funcionalidades y mejoras de la aplicación P2P Sharing, versión anterior de nuestra aplicación que hemos tomado como base para el desarrollo del proyecto.

Las principales funcionalidades que se han desarrollado en el proyecto son el bloqueo de usuarios con los cuales no se quiere compartir archivos, la compartición de carpetas además de archivos únicos, la previsualización de archivos antes de descargar, la mejora del sistema de envíos y descargas, y la mejora en el consumo de memoria.

Además de estas nuevas características implementadas en la aplicación, se han realizado varias mejoras de la aplicación base, como pueden ser de rendimiento de tareas específicas, de usabilidad y visualización, e implementación de nuevos componentes en lugar de otros que ya están en desuso o anticuados.

Palabras clave

Android, P2P, aplicación, compartir, archivos, programación, java.

ABSTRACT

P2P Sharing 2.0: Android P2P app for file sharing

The P2P Sharing 2.0 project has consisted in the elaboration of new functionalities and improvements of P2P Sharing app, which is the old version of our app and that has been taken as a basis for the development of the project.

The main functionalities that have been developed into the project are groups creation to be able to share files with friends that belong to it, block users you don't want to share files with, folder sharing in addition to single file sharing, archives preview prior to downloading, an uploads and downloads system upgrade, and a memory usage improvement.

Besides these new features implemented in the app, there have been made some more enhancements like performance of specific tasks, usability and display, and implementation of new components replacing others that were obsolete or outdated.

Keywords

Android, P2P, app, share, files, programming, java.

ÍNDICE

ÍNDICE DE FIGURAS

Capítulo 1 - Introducción

En este proyecto he trabajado con uno de los puntos de la informática más importantes, con mayor crecimiento y más populares de las comunidades de desarrolladores y usuarios que es el desarrollo de aplicaciones móviles.

Han pasado ya algunos años desde el lanzamiento del primer "smartphone", pero las aplicaciones en estos dispositivos eran sólo una evolución. De hecho, es difícil hallar cual fue la primera aplicación diseñada para funcionar en un teléfono móvil, pues antes las "apps" eran más bien conocidas como características, como por ejemplo las alarmas, la calculadora o la agenda de contactos. Se quería integrar cuantas utilidades fuera posible siguiendo las últimas tendencias. Hoy un teléfono móvil, si es que se le puede seguir llamando solo teléfono, es una potente herramienta gracias a los avances en las tecnologías.

No tan conocidas, pero no por ello menos importantes, la aplicación se basa en dos tecnologías para la compartición de datos construidas de forma diferente a la tecnología usada en la mayoría de las aplicaciones P2P: Hablamos de WebRTC y PubNub. Ambas serán comentadas con más detalle en el capítulo 2.

Se ha hecho una pequeña incursión en el paradigma de las redes P2P y los principales tipos que han sido formulados a lo largo de la historia reciente, si tenían estructura o si estaban centralizados, observando qué topologías se habían planteado. Todo ello basado en una jerarquía lo más plana posible entre las máquinas de la red. Todas ellas eran a su vez cliente y servidor lo cual rompía con la concepción clásica de las comunicaciones informáticas.

El trabajo[1] sobre el que se ha implementado este solucionaba la gran dificultad que se presenta a la hora de comunicar dos dispositivos de una red P2P sin intermediarios al menos en la transmisión de grandes volúmenes de datos. La

[1] P2P Sharing: https://github.com/jpradas/P2P-Android-App

solución combinada de WebRTC y PubNub para comunicación entre pares ha resultado sin duda una gran opción.

Basándome en una app desarrollada anteriormente por otros compañeros, se han implementado nuevas funcionalidades que dan valor añadido a la aplicación, así como también se han hecho mejoras en varios aspectos de ella en cuanto a rendimiento, apartado visual o uso de nuevos elementos a la hora de programar.

1.1 Introduction

In this project I have worked in mobile apps, which is one of the most important points in computer science, with significant growth and high popularity between the developers and users communities.

Some years have passed since the release of the first "smartphone", but the applications for these devices were only an evolution, In fact, it is hard to find which was the first application designed to work in a mobile phone, since "apps" were better known as features, like alarms, the calculator or the contacts list. It was wanted to include as many utilities as possible following last trends. Nowadays a mobile phone, if it could be called only a phone, is a powerful tool thanks to technological progress.

Not so well known but no less important, there are two technologies on what is based the application for data sharing built in a different way than technology used in the majority of P2P apps: We are talking about WebRTC and PubNub. Both will be discussed in detail in chapter 2.

There has been done a little incursion into P2P networks paradigm and their main kinds that have been formulated all recent history long, if they had any structure or if they were centralized, analyzing which topologies were set out. All of this based on a hierarchy between machines in the network as flat as possible. All

of them were client and server at the same time, that is a thing that broke the classic computer communications idea.

The work on which it has been implemented my code solved the main trouble that stood when two devices of a P2P network communicate without intermediaries at least in transmissions with huge amounts of data. The WebRTC and PubNub combined solution for communication between peers has proved to be a great option certainly.

Being this app based on another one[2] made by other partners, there has been implemented some new functionalities which provide added value, as well as improvements in some aspects regarding performance, visual part or use of new items at the moment of programming.

1.2 Antecedentes y motivación

Las tecnologías móviles llevan varias décadas en constante evolución. A finales del siglo XX experimentaron un crecimiento tecnológico, tal que a día de hoy gran parte de la población joven y de mediana edad dispone al menos de un teléfono móvil por persona en los países desarrollados, y poco a poco también se incrementa su uso en los países en vías de desarrollo. Desde principios de la década del 2000, los teléfonos móviles y las tabletas electrónicas han ido aglutinando funciones que ya estaban cubiertas por otros dispositivos, haciendo que estos quedaran obsoletos por ser menos práctico tener varios dispositivos en lugar de uno y por estrategias de publicidad de los fabricantes entre otros motivos [1].

[2] P2P Sharing: https://github.com/jpradas/P2P-Android-App

Uno de los principales pilares de esta evolución ha sido la posibilidad de comunicarse desde cualquier lugar. Con ello se fueron incluyendo otras funcionalidades como el reproductor de archivos de música, la cámara de fotos y vídeo o el GPS. Sin embargo, el intercambio de archivos tardó algo más en llegar con respecto a su popularidad en el PC. Programas como eMule [2] o Bittorrent [3] empezaron a dominar las redes P2P rápidamente.

La idea de ampliar una aplicación hecha para intercambiar archivos solo entre amigos me pareció muy interesante, principalmente por dos motivos: la ampliación de mis conocimientos de programación para Android y el reto de extender un trabajo ya comenzado. Se debía comprender mediante qué tecnología funciona y realizar una serie de modificaciones que añadieran nuevas funciones -previstas o no por los desarrolladores anteriores-, mejorar funcionalidades actuales y mejorar la experiencia de usuario.

Antes de comenzar a trabajar en la aplicación decidí hacer un estudio de trabajos relacionados, donde seleccioné una serie de aplicaciones existentes en el mercado que poseen algún aspecto en común con mi idea del proyecto. Este estudio fue importante no solo para conocer el interés que tiene la realización de este proyecto teniendo en cuenta la escena actual, sino también a la hora de tomar decisiones posteriores acerca de los puntos que debían tener más peso en la aplicación y qué funciones eran prescindibles o estaban ausentes.

1.3 Background and motivations

Mobile technologies are in constant evolution since some decades. At the end of the 20th century they experienced a technological growth such that today a large part of the young and middle-aged population has at least one mobile phone, and its use is increasing little by little in developing countries. At the early 2000s, mobile phones and tablets have been grouping functions that were already covered

by other devices, making these obsolete because it wasn't handy to have some devices instead of only one and because there were marketing strategies made by companies, among other reasons.

One of the main pillars of this evolution has been the possibility of communicating from anywhere. Besides, other functionalities were being included like the music player, the photo and video camera or the GPS. However, file exchange applications took some more time to arrive in relation to their popularity in PC. Apps like eMule [2] or Bittorrent [3] quickly began to dominate P2P networks.

The idea of broaden an Android app made to share files only between friends was found interesting mainly for two reasons: expansion of my knowledge in Android and the challenge of extending a work that was started before. I should understand the technology with which it is working by, and carry out a number of modifications that adds new functions -planned or not by previous developers-, improve current functionalities and improve user experience.

Before I started to work in the application, I decided to do a study about related works where I selected some of those existing apps on the market that have any common aspect with my idea of the project. This study was important not only to know the interest that the development of this project has taking into account the current framework, but also to make late decisions about the topics that should carry more weight in the app and what functions were dispensable or missing.

1.4 Objetivos

El objetivo de este proyecto es el análisis y la mejora del trabajo anterior. Esta mejora ha consistido principalmente en la inclusión de nuevas funciones que pudieran ser útiles, examinar el rendimiento general de la aplicación y el rendimiento de tareas específicas para poder incrementarlo en aquellos casos que

he creído y hayan sido necesarios, y comprobar las características de la interfaz gráfica en cuanto a usabilidad también con la predisposición de embellecer o afinar elementos.

En cuanto a los objetivos personales, el más importante es el aprendizaje y la mejora de los conocimientos previos que tenía con respecto a la programación (sobre todo del lenguaje Java), pero especialmente relacionado con la programación móvil, en este caso de Android.

Al ser un proyecto construido sobre uno realizado por otros programadores ha supuesto un esfuerzo extra por varios motivos. Hasta ahora en la carrera casi siempre habíamos realizado nuestros códigos en proyectos desde cero. Ahora me he tenido que adaptar a ello estudiando no sólo la implementación de mis compañeros, sino también las tecnologías en las cuales residía la principal meta del pasado trabajo, que era el correcto establecimiento de las comunicaciones entre dos dispositivos situados cada uno tras un cortafuegos y con la dificultad añadida de no estar permitido iniciar las conexiones desde el exterior de una red local por razones de seguridad.

1.5 Objectives

The objective of this project is the analysis and improvement of the previous work. This improvement has consisted mainly in the inclusion of new functions that could be useful, study general performance of the app and specific tasks performance in order to increase it in those cases I have thought and those that had been necessary, and check graphic interface features regarding usability also with predisposition to decorate or tune items.

With regard to personal objectives, the most important one is learning and progress of my previous knowledge I had about programming (Java language above all), but specially related to mobile development, taking Android in this case.

As it is a project built over another made by other programmers it has supposed to me an extra effort for a number of reasons. Until now in the degree we had carried out our codes in projects from zero almost always. Now I have had to adapt to that studying not only our mates implementation, but also technologies in which lived the main goal of the past work, that it was the correct establishment of communications between two devices located each one behind a firewall and the added difficulty of not being allowed to start connections from outside of a local network for security reasons.

1.6 Plan de trabajo

Para ponerme en marcha con las tareas he seguido un plan de trabajo propio en el que he tratado de cubrir de manera práctica las necesidades más inmediatas para desarrollar correctamente el trabajo, y siendo en parte pulido durante el desarrollo:

1. Estudio de aplicaciones similares disponibles en el mercado para tener más ideas de qué prestaciones podía intentar implementar en esta.
2. Estudio del código desarrollado previamente por nuestros compañeros.
3. Análisis de las funciones que están hechas teniendo en cuenta los siguientes aspectos:
    - Efectividad, o precisión con la que el usuario satisface el objetivo de su tarea.
    - Facilidad de aprendizaje de las acciones e interfaz gráfica.

- Eficiencia, o tiempo empleado en realizar una acción navegando por la GUI.

- Rendimiento de las funciones existentes.

4. Evaluación de nuevas funciones interesantes que se echaron en falta en el programa original.

5. Análisis del apartado "Trabajo Futuro" del TFG anterior para estudiar la utilidad e idoneidad de las ideas propuestas comparadas con las que se valoraron en el punto anterior.

6. Desarrollo de las nuevas funciones elegidas y realización de pruebas.

7. Aplicación de los principios de usabilidad expuestos en el apartado 3 en los nuevos casos de uso.

8. Una vez completada la funcionalidad, realización de nuevas pruebas y corrección de los últimos errores.

1.7 Workplan

To get going with the work I have followed my own workplan in which I have tried to cover in a practical way the most immediate needs in order to develop the job correctly, and being partly polished during development.

1. Study of similar applications available on the market to get some more ideas about which features could try to implement into our work.

2. Study of the developed code by my partners previously.

3. Analysis of the functions that are present taking into account the following aspects:

a. Effectiveness, or accuracy with what user satisfies their task objective.

b. Ease of learning of actions and graphic interface.

c. Efficiency, or time spent doing one action navigating through the graphic interface.

d. Performance of existing functions.

4. Test of new interesting functions that were missing in the original program.

5. Analysis of the section "Future work" showed in the previous work to study the usefulness and suitability of suggested ideas with the ones I have assessed in the earlier point.

6. Develop of the new functions chosen and tests execution.

7. Apply the usability principles shown in section 3 into the new use cases.

8. Execution of new tests and final corrections once functionality is completed.

1.8 Contenido de la memoria

Este trabajo está dividido en 4 capítulos, los cuales se explican a continuación.

El capítulo 1 es de introducción al proyecto, en el cual está este apartado junto con un breve resumen de las motivaciones que han llevado a realizarlo y los objetivos fijados.

En el capítulo 2 se explican las principales partes de la informática que se abarcan en el proyecto para ponerlo en contexto. Se explica qué es, en qué se basa la tecnología P2P, y también qué es Android y sus principales características.

A lo largo del capítulo 3 se explican los diferentes desarrollos realizados para la mejora de la aplicación. Todos ellos se separan por funcionalidades nuevas, o bloques de mejora de un aspecto concreto, para así poder entender mejor el desarrollo. Estos apartados son:

- Sistema de bloqueo de usuarios
- Compartición de carpetas
- Previsualización de archivos e iconos
- Sistema de descargas y subidas
- Reducción del consumo de memoria
- Bloqueo para envíos usando la red del operador
- Revisión del borrado de amigos

En el capítulo 4 están las conclusiones tras la finalización del proyecto junto con los resultados obtenidos y posibles mejoras o nuevos desarrollos para llevar a cabo en un futuro, que no se han podido realizar durante este año.

Para terminar, los últimos apartados corresponden a la bibliografía consultada para realizar las diferentes partes del trabajo y una pequeña sección de créditos.

1.9 Memory content

This work is divided in four chapters, which are explained next.

Chapter 1 is an introduction of the project, in which it is this section beside a brief summary of motivations that have taken me to do it and the objectives set.

In chapter 2 main computer parts that are covered in the project are explained to put it into context. It is described what it is, on what is P2P technology based and also what is Android and its main features.

Throughout chapter 3 I explain the different developments made for the application improvement. All of them are separated by new functionalities or upgrade blocks about some concrete aspect so that the work could be understood better. These parts are:

- Blocking users system
- Folder sharing
- File preview and icons
- Uploads and downloads system
- Memory usage reduction
- Block for sending files through mobile network
- Friends deletion review

In chapter 4 are the conclusions after project completion beside results got and possible improvements or new developments to carry out in the future, which they couldn’t be done during this year.

Finally, last parts correspond to the bibliography consulted to do the different parts of the work, and a little credits section.

Capítulo 2 - Estado del arte

2.1 Redes P2P

En esta sección se expone brevemente la tecnología P2P y los tipos más comunes según su centralización y estructura.

Las conexiones que son realizadas en la aplicación están fundamentadas en el protocolo P2P. Las redes P2P surgieron como una de las mejores opciones para la compartición de ficheros entre máquinas, cada una de ellas formando un nodo de comunicación que puede enviar ficheros propios, recibirlos y retransmitir fragmentos de dichos ficheros a otras máquinas colaborando de manera eficiente en el uso de la red.

"Una red peer-to-peer, red de pares, red entre iguales o red entre pares (P2P, por sus siglas en inglés) es una red de ordenadores en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red. Las redes P2P permiten el intercambio directo de información, en cualquier formato, entre los ordenadores interconectados"[3] [4] Además de utilizarse para transmitir ficheros también se emplean en otro tipo de comunicaciones como la tecnología VoIP.

2.1.1 Tipos según centralización

Este tipo de clasificación es el más común, y en él se atiende a las conexiones entre los nodos. Si hay un servidor central por donde pasan todas las conexiones, si la carga está repartida entre varios servidores clave, o si no hay

[3] https://es.wikipedia.org/wiki/Peer-to-peer

ningún punto con mayor carga que el resto. Existen 3 tipos como vemos a continuación:

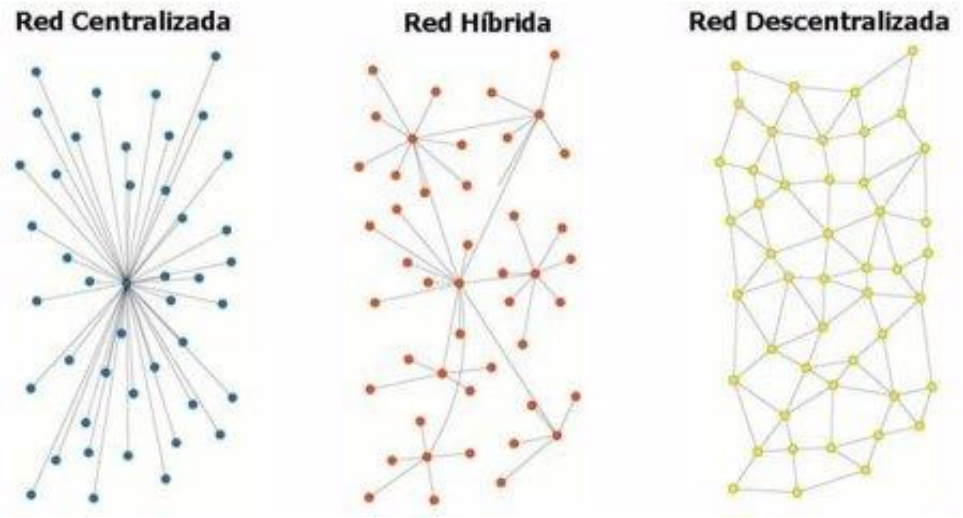

Figura 2-1: Tipos de redes según centralización

Centralizada

Todas las máquinas pertenecientes a la red están conectadas a un servidor central que es el encargado de guardar la información relativa a la estructura. Ejecutar el servidor para el correcto funcionamiento de la red limita directamente el número y la cantidad de datos de las transferencias que pudieran llevarse a cabo ya que dependen de la capacidad de gestión de este, además de imposibilitar cualquier intercambio de información si dicho servidor no está operativo.

Descentralizada

Es el tipo de red P2P conocida como "pura", ya que no dependen de una máquina para las conexiones, sino que los propios pares se encargan de dicha gestión. Todos los pares son capaces de comunicarse entre sí directa o indirectamente con ayuda de otras máquinas que actúen como meros repetidores.

Híbrida

Aquí se tiene una mezcla de las características de los dos tipos de redes anteriores. Se cuenta con uno o varios servidores centrales que gestionan las conexiones entre los nodos sin actuar como retransmisores de datos. Una vez que facilitan la información necesaria a las máquinas que se quieran comunicar entre sí son prescindibles.

2.1.2 Tipos según estructura

En este tipo de clasificación se atiende a la distribución que gestiona la conexión de todos los nodos de la red. En función de si se sigue una estructura determinada, o si es aleatoria la ruta que se toma para las conexiones, existen los siguientes dos tipos:

Estructuradas [5]

Las redes estructuradas se caracterizan por estar construidas en base a una topología determinada. "El tipo más común de redes P2P estructuradas implementan una tabla hash distribuida (DHT)"[4]. Dicha tabla utiliza "hashing" para determinar la propiedad de los ficheros. Cada máquina debe tener información relevante sobre sus nodos vecinos.

No estructuradas [6]

Esta clase de redes no están conformadas según ninguna topología. Los nodos forman conexiones entre ellos arbitrariamente. La obtención de datos cuya ubicación se desconoce se resuelve mediante inundación, esto es, consultas a todos los nodos vecinos para que transmitan los datos buscados a todos los pares que los posean. El inconveniente es que se genera una gran cantidad de tráfico que se traduce en latencia en la red y mayor consumo de recursos.

[4] https://en.wikipedia.org/wiki/Peer-to-peer#Structured_networks

2.1.3 Tecnologías de conexión

A continuación, se van a explicar qué son y cómo funcionan las dos principales tecnologías de conexión que se utilizan para enlazar a los usuarios de la aplicación y así poder compartir la información.

WebRTC [7]

"Es un proyecto libre y abierto que proporciona a navegadores y aplicaciones móviles capacidades de comunicaciones en tiempo real (RTC) mediante APIs sencillas"[5]. Su equipo tiene como objetivo ofrecer mediante esta tecnología una herramienta para que los desarrolladores puedan construir aplicaciones con comunicaciones de alta calidad. Es una iniciativa con el apoyo de Google, Mozilla y Opera.

Entre otras características, los ideólogos de este framework promocionan su calidad haciendo énfasis en la gran utilidad que tendría en programas de transmisión de audio y vídeo en tiempo real. Además de estas razones, resultó especialmente interesante por poder resolver el problema de las comunicaciones entre dos dispositivos detrás de NATs y cortafuegos mediante técnicas para atravesarlos (conocidas como "NAT/firewall traversal").

WebRTC está soportado en los navegadores Chrome, Firefox y Opera, en Android y en iOS. En su página web se puede encontrar todo tipo de información: aplicaciones de demostración, guías de iniciación para su uso, prácticas guiadas, información sobre los códecs de audio y vídeo utilizados, vídeos informativos, etcétera.

[5] https://webrtc.org/

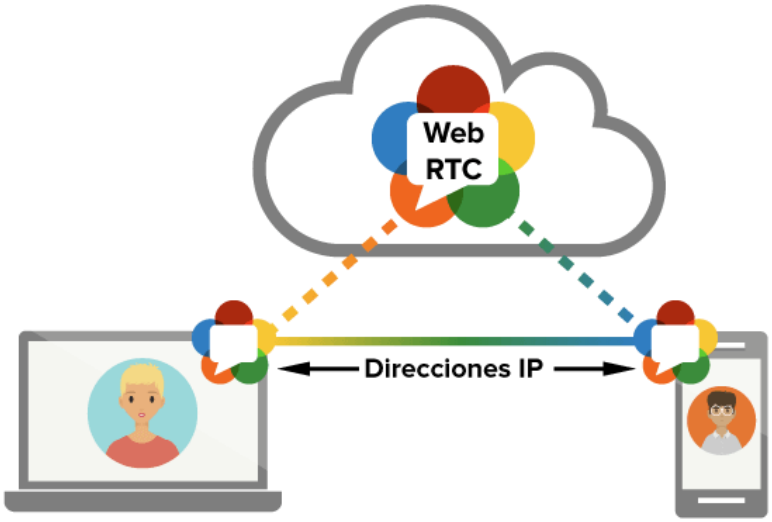

Figura 2-2: WebRTC

PubNub [8]

"PubNub es una red global de transmisión de datos (DSN) y compañía de red-como-servicio en tiempo real"[6]. Aportan una API de tipo publicador-suscriptor basada en mensajes que permite a los programadores realizar aplicaciones con alguna componente que se ejecute en tiempo real. Algunos ejemplos de programas desarrollados son chats, solicitudes de taxis y juegos multijugador. Esta tecnología está pensada para lidiar con corrientes de datos de diversa complejidad, procurando garantizar la menor latencia posible al mismo tiempo que fiabilidad y escalabilidad.

La red replica los mensajes hacia los centros de datos de los que disponen. Así si se produce algún problema en la conexión al centro de datos más cercano a un dispositivo que hubiera de recibir algún mensaje este se conectaría al siguiente centro más cercano y obtendría el mensaje.

---

[6]https://support.pubnub.com/support/solutions/articles/14000046386-what-is-pubnub-

El patrón publicador-suscriptor es una buena opción para el sistema de subidas y descargas. Todos los dispositivos pueden actuar como publicadores y suscriptores a la vez. El que toma el rol publicador es el que sube los mensajes y los suscriptores son aquellos a los que PubNub retransmite estos datos en tiempo real. Los mensajes utilizan canales. Los dispositivos que deseen recibir determinados mensajes han de suscribirse al canal pretendido para ello. Por último, cabe mencionar que los canales empleados tienen topologías de tipo one-to-one(unicast), one-to-many(broadcast), many-to-many(multicast) o many-to-one(consolidation) [10].

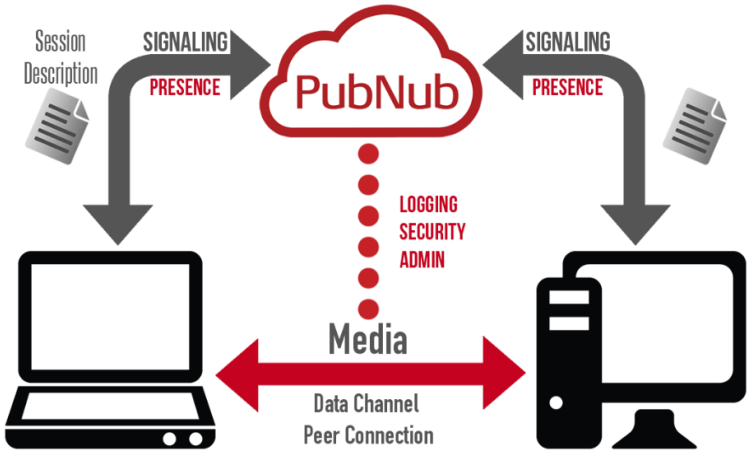

Figura 2-3: PubNub

2.2 Android

Android es un sistema operativo desarrollado para dispositivos móviles con pantalla táctil, que se basa en una versión modificada del kernel de Linux y que es de código abierto. En la actualidad, gracias al desarrollo de la tecnología no solo se usa en teléfonos móviles, sino que además se puede utilizar en tablets, relojes inteligentes, automóviles o incluso televisores [11].

Figura 2-4: Imagen Android

Android fue creado por Android Inc., una empresa que fue comprada por Google en el año 2005. Posteriormente, en el año 2007 Google funda un consorcio de 47 empresas dedicadas a la fabricación y desarrollo de hardware y software, llamado Open Handset Alliance para así promover y avanzar en los estándares abiertos de los dispositivos móviles. A su vez Google también anuncia la primera versión del sistema operativo, Android 1.0 Apple Pie que se lanzaría en 2008 [12].

Desde el lanzamiento de Android 1.0 Apple Pie en 2008, hasta la última versión, a día de hoy, Android 10 lanzada en agosto de 2019, se han sucedido hasta 17 versiones nuevas del sistema operativo. Durante estos años y con el paso de las versiones, se han ido añadiendo nuevas funcionalidades, mejoras gráficas, y de rendimiento, nuevos diseños de interfaz, avances a nivel de uso de hardware y software y mayor importancia y desarrollo a nivel de seguridad. Estas mejoras han surgido según ha evolucionado la tecnología durante estos 11 años de la primera versión a la última [13] [14].

2.2.1 Versiones de Android

A continuación, se van a enumerar las diferentes versiones de Android con sus principales características y mejoras [13] [15]:

- Android 1.0 Apple Pie (Tarta de manzana): aplicaciones de Google -GApps-, Android Market, patrón de desbloqueo, aviso por batería baja.

- Android 1.1 Banana Bread (Pan de plátano): búsqueda por vos -Google Voice Search-, aplicaciones y juegos de pago, compartir ubicación -Google Latitude-.

- Android 1.5 Cupcake (Magdalena): teclado virtual, panel de notificaciones rediseñado con apariencia más limpia y suave, iconos en la pantalla de inicio -Live Folders-, grabación de vídeos con la cámara del móvil, zoom en páginas web, imágenes y mapas, opción de subir vídeos a Youtube desde el móvil.

- Android 1.6 Donut: soporte para diferentes tamaños de pantalla, mejoras de las aplicaciones del sistema, soporte para redes CDMA, rediseño Android Market con un estilo más claro y colorido, capturas de pantalla de aplicaciones, sistemas de texto a voz, informe sobre el estado de la batería.

Figura 2-5: Primeras versiones Android

- Android 2.0 y 2.1 Éclair (Palo de crema): renovada pantalla de bloqueo, barra de búsqueda de Google, navegación GPS en Google Maps, brillo

automático, distintas escenas en la cámara nativa, una aplicación de Facebook preinstalada, fondos de pantalla animados, nuevo cajón de Apps, rediseño de aplicaciones, animaciones en toda la interfaz, ampliar o reducir el tamaño de imágenes, páginas web o mapas (pinch-to-zoom).

- Android 2.2 Froyo (Yogur helado): dock de aplicaciones en la pantalla de inicio con dos iconos laterales y uno central (destinado al cajón de Apps), botón para la actualización de todas las aplicaciones o la habilitación de las actualizaciones automatizadas, Adobe Flash Player, soporte para mover aplicaciones a la tarjeta microSD, control del sistema a través de comandos de voz.

- Android 2.3 Gingerbread (pan de jengibre): diseño más moderno, conectividad NFC, panel de notificaciones con tono más oscuro, acceso de juegos al audio, controles, gráficos y almacenamiento del sistema.

- Android 3.0 Honeycomb (Panal): exclusiva de tablets, diferentes tonos de azul brillante, volvió a cambiar la pantalla de bloqueo, barra de navegación virtual en pantalla, botonera virtual de menú, home y atrás, menú de aplicaciones recientes, panel de ajustes rápidos.

- Android 4.0 Ice Cream Sandwich (Sándwich de helado): botones virtuales en Android, aplicaciones e iconos rediseñados de una manera más futurista, cajón de aplicaciones dividido en dos pestañas -Apps y Widgets-, panel de notificaciones con fondo transparente, añadir carpetas de aplicaciones, redimensionar widgets, conexión NFC, Google Play Store.

- Android 4.1 y 4.2 Jelly Bean (Gominola): Project Butter -proyecto para mejorar la sensación de fluidez-, modificación de panel de notificaciones con reloj digital y notificaciones ampliables, añadir diferentes cuentas de usuario a un mismo dispositivo, Google Now, se implementan los servicios de Google Play, añadir widgets a la pantalla de bloqueo.

- Android 4.4 KitKat (Kit Kat): Project Svelte producía una importante reducción en el uso de la memoria RAM, tonos blancos, Google Now se

integra en el launcher, modo inmersivo, comando de voz "Ok Google", introducción de Google Fotos.

- Android 5 Lollipop (Piruleta): líneas de diseño Material Design creando un lienzo tridimensional, nuevo panel de ajustes rápidos, barra de navegación más minimalista, máquina virtual ART para mejorar la velocidad de apertura de aplicaciones y optimizar el consumo de memoria, Project Volta para mejorar la duración de la batería, notificaciones en pantalla de bloqueo.

- Android 6 Marshmallow (Malvavisco): cajón de aplicaciones con scroll vertical original; Now on Tap con una pulsación larga sobre el botón home; sistema de permisos granular; sistema DOZE que habilitaba un modo de bajo rendimiento cuando el teléfono se encontraba estático en un lugar, desconectado y con la pantalla apagada; Adoptable Storage sd externa como una ampliación de la memoria interna; avisos en formato Peek; cambio formato de control de volumen; actualizaciones de seguridad mensuales; lectores de huellas dactilares -Fingerprint API-.

- Android 7 Nougat (Turrón): ejecución de dos aplicaciones en pantalla partida, contestación de mensajes desde las notificaciones, mejores de DOZE, aparición de accesos directos rápidos, dock de aplicaciones con fondo semitransparente, división del almacenamiento del sistema en dos particiones diferentes -Seamless Updates-.

- Android 8 Oreo: gestión de notificaciones a distintos niveles de prioridad, nuevos formatos de aviso como Notification Dots o Notification Badgest, Picture in Picture que permite mostrar diferentes tipos de contenido en una pequeña pantalla flotante mientras se siguen usando otras aplicaciones, autorrelleno de texto nativo, selección de texto inteligente, nuevo pack de emojis, Project Treble para modularizar el sistema operativo, panel de ajustes rápidos semitransparente, menú de apagado flotante, API para redes neuronales para implementar sistemas de IA y Machine Learning en sus Apps.

- Android 9.0 Pie (Pastel): navegación por gestos, sistema de posicionamiento en interiores a través de la tecnología Wi-Fi RTT, mejoras rendimiento -ART- y ahorro de energía -DOZE-, soporte a sistemas fotográficos formados por dos cámaras, nuevo panel de ajustes rápidos, respuestas rápidas a mensajes desde notificaciones, editor de capturas de pantalla, mejora control volumen, brillo adaptativo.

- Android 10: mejora en privacidad y seguridad, tema oscuro, más herramientas personalización, modo escritorio, burbujas flotantes de notificaciones, generación automática de subtítulos en vídeos. Primera versión sin nombre de dulce.

Figura 2-6: Versiones de Android, de 1.6 Donut a 9.0 Pie

2.2.2 Principales características

A continuación se mencionan cuáles son las características más importantes que hacen tan particular a este sistema operativo y que han hecho que tenga tanto

éxito. Como se ha podido ver en el punto anterior no todas estas características estaban desde el principio, sino que muchas se han ido introduciendo a lo largo del tiempo con las mejoras de las diferentes versiones.

El núcleo de Android está basado en el Kernel de Linux, el cual también está basado en el de Unix.

El código fuente principal es Android Open Source Project(AOSP) [16], y tiene una licencia Apache [17], con la cual el código es abierto y se pueden modificar y distribuir versiones modificadas de él. Esto es utilizado para mejorar el sistema operativo mediante la comunidad de desarrolladores, o crear versiones propias con desarrollos específicos, como pueden hacer las diferentes empresas de móviles.

Android se puede adaptar a una gran cantidad de tipos de pantallas con diferentes resoluciones y tecnologías como VGA, biblioteca de gráficos 2D o 3D y diseño de teléfonos tradicionales.

Uno de sus puntos fuertes es la gran cantidad de tecnologías de diferentes tipos con las que puede trabajar y soportar. A continuación, vamos a ver las principales tecnologías de cada tipo con las que se puede trabajar:

- Conectividad: Bluetooth, WiFi, LTE, HSDPA, NFC, GPRS, USB, etc [18].
- Multimedia: MP4, MPEG-4, AMR,AMR-WB, AAC, MP3, MIDI, WAV, JPEG, PNG, GIF, BMP [19]
- Mensaje: MMS,SMS, FCM
- /RTSP, HTML5, RTMP, Adobe Flash Player

Tiene soporte con Java. Gran parte de las aplicaciones están escritas en Java y además una parte de la arquitectura de Android, como es la Java API Framework, está desarrollada en Java. [20]

Para el almacenamiento de datos utiliza SQLite [21], una biblioteca en lenguaje C que implementa un motor de base de datos SQL pequeño, rápido, autónomo y con todas las funciones.

Android dispone de un entorno de desarrollo integrado (IDE) propio basado en IntelliJ, este IDE es Android Studio [22]. Entre sus características destaca el emulador de dispositivos, un entorno unificado para desarrollar para todos los dispositivos Android, o la ejecución al instante para la aplicación de los cambios. Aun así, existen más IDEs para desarrollar apps, como pueden ser Eclipse, NetBeans, IntelliJ o Aide [23].

Google Play [24]. Un catálogo oficial de Android de donde se puede descargar e instalar aplicaciones, tanto gratuitas como de pago. Una forma de controlar y difundir las aplicaciones desde un único lugar de forma fácil.

2.2.3 Arquitectura

La arquitectura de Android se divide en 6 bloques, como se puede ver en la Figura 2-7 [20]. Cada uno es una pieza de todo ese software de código abierto del que está formado Android, entre ellos el kernel de Linux en el que se basa, más otros desarrollos que lo complementan. A continuación, vamos a explicar cada bloque para poder entenderlo mejor:

- System apps (Aplicaciones del sistema). Compuesta por las principales apps que tiene el sistema operativo y también algunas de las apps que instale el usuario, las cuales pueden usar las de Android para realizar la misma función.

- Java API Framework. Es el conjunto de herramientas de desarrollo para una aplicación. Esta API está escrita en Java y formada por las funciones necesarias para poder usar los componentes y servicios del sistema operativo.

- Native C/C++ Libraries (Librerías C/C++ nativas). En este conjunto se encuentran las librerías utilizadas por Android para algunas de sus componentes y servicios, están escritas en C/C++.

- Android Runtime, ART (Tiempo de ejecución de Android). Está diseñado para que se puedan ejecutar varios procesos por separado, pudiendo lanzarse varias apps de forma independiente con sus propios procesos.

- Hardware Abstraction Layer, HAL (Capa de Abstracción de Hardware). Esta capa es donde se encuentran los interfaces estándares de los diferentes componentes hardware del dispositivo para que cuando se quieran usar desde una aplicación los invoquen mediante la API de Java.

- Linux Kernel. Es el núcleo de Android, del que parte todo el sistema operativo, y donde se encuentran los drivers o controladores.

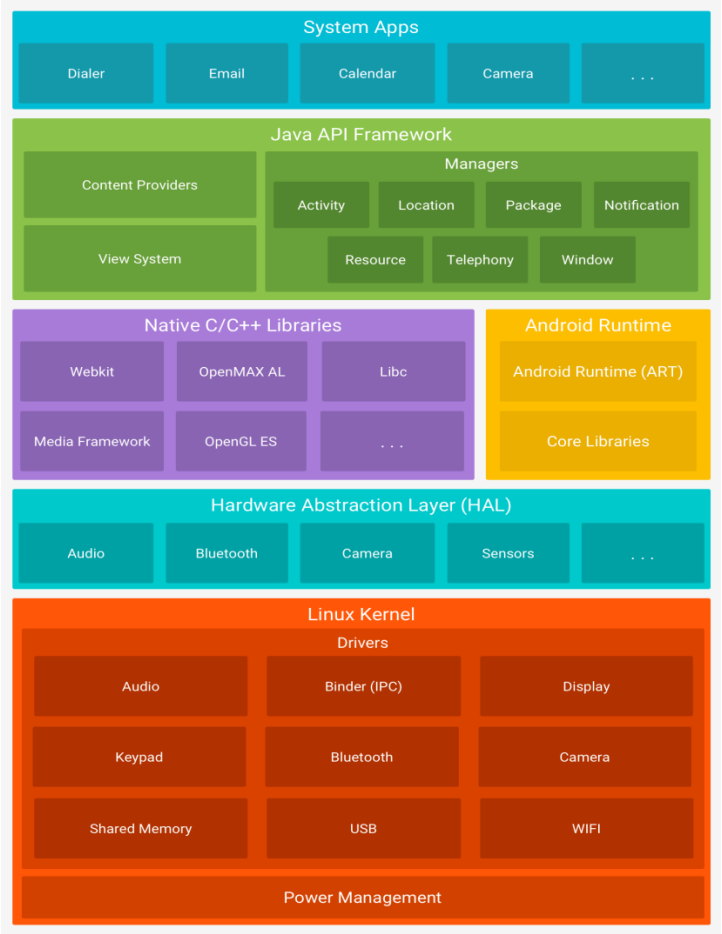

Figura 2-7: Arquitectura Android

2.3 Trabajo relacionado

En este punto se van a explicar cuáles son las principales características de las aplicaciones (Figura 2-8) más similares a la aplicación desarrollada que hay en el mercado. De ellas se ha podido obtener una idea de las funcionalidades básicas que se dan a los usuarios y de las carencias que pueden tener.

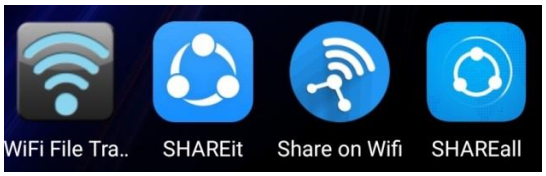

Figura 2-8. Aplicaciones Similares

2.3.1 Wifi File Transfer

Esta aplicación[7] permite compartir toda clase de archivos entre un móvil o tablet y un ordenador. Requiere que ambos dispositivos estén conectados a la misma red wifi para establecer la conexión mediante uno de sus puertos, al que se puede acceder desde cualquier navegador web en el PC.

Una vez introducidas la URL y la contraseña en el navegador, puede empezar la subida y/o descarga de archivos entre los dispositivos. Esta aplicación también puede ser utilizada para compartir estos archivos con amigos siempre y cuando estén conectados a la misma red wifi.

2.3.2 SHAREit

[7] https://play.google.com/store/apps/details?id=com.smarterdroid.wififiletransfer

Es una de las aplicaciones[8] más descargadas de Google Play en varios países. Da la posibilidad de transferir archivos de cualquier formato sin necesidad de utilizar cable USB o conexión a internet a una velocidad de hasta 20 Mb/s.

Ofrece muchas funcionalidades interesantes: transferir contactos, mensajes y fotos de un teléfono antiguo a uno nuevo, ver iconos o miniaturas de los ficheros, o la compartición dentro de un grupo. Además, es una aplicación multiplataforma.

2.3.3 ShareOnWifi

Es una aplicación[9] que utiliza el protocolo P2P para compartir archivos entre distintos dispositivos de forma directa, ya sean Windows, Android o iOS. Los dispositivos deben estar conectados a la misma red wifi. A diferencia de otras aplicaciones de intercambio de archivos, en ShareOnWifi no es necesario realizar un proceso para establecer una conexión cada vez que se quiera compartir un archivo; en su lugar, el dueño de los archivos elige qué archivos compartir mediante la aplicación y con quién compartirlos (no obstante, existe la opción de dejarlos en estado público). Después, las personas conectadas a la misma red wifi que tengan permiso para ver estos archivos pueden buscarlos y descargarlos.

La aplicación también cuenta con un aspecto social mediante el cual cada usuario puede personalizar su perfil y formar grupos con los que compartir ciertos archivos. Este sistema hace que el proceso sea muy cómodo y permite que varios dispositivos accedan a los mismos archivos en cualquier momento sin realizar peticiones al dueño.

Esta aplicación es la más similar a la idea que tenemos para nuestro proyecto. Entre sus similitudes está la posibilidad de tener grupos y el que cada

---

[8] https://play.google.com/store/apps/details?id=com.lenovo.anyshare.gps

[9] https://play.google.com/store/apps/details?id=aaqib.shareonwifi

usuario suba un archivo, y se pueda descargar cuando cada usuario quiera, generando en ese momento la conexión. La principal diferencia con nuestra aplicación es el rango de actuación, ya que en esta app es necesario estar conectado a la misma red wifi, en cambio, nuestra app no es necesario.

2.3.4 SHAREall

Otra aplicación[10] que posibilita la compartición de archivos sin conexión a internet mediante la creación de un hotspot en el receptor o bien utilizando Wifi Direct. Es capaz de no solo compartir archivos, sino también películas o aplicaciones.

No tiene límite de compartición, ya que, al no usar conexión a internet, no hace uso de datos ni internet. También para dar mayor seguridad puedes usar contraseñas en la compartición de archivos.

[10] https://play.google.com/store/apps/details?id=com.pnd.shareall

Capítulo 3 - Desarrollo del proyecto

En este apartado se entrará en profundidad en las funciones que se han decidido desarrollar. Para tomar esta decisión y tras tener una lista suficiente de ideas, se ha acordado con el director aquellas que nos han parecido de mayor utilidad y fueran posibles de llevar a cabo.

Además, algunos de estos desarrollos también han sido realizados sin estar planificados anteriormente en función de las necesidades que se han ido observando durante el avance del proyecto.

Antes de entrar a describir cada función realizada se van a comentar brevemente algunos de los componentes más importantes que se han utilizado y que están integrados en Android:

- **Actividades** [24]

Son el componente principal de casi cualquier aplicación en Android. Albergan la interfaz gráfica que será utilizada por el usuario y tienen la capacidad de iniciar otras actividades, quedándose las anteriores guardadas en una pila y siendo recuperables pulsando el botón atrás. Al cambiar de una a otra se producen llamadas a los métodos de "callback" que permiten controlar su ciclo de vida al programador. Además es posible sobrescribir estos métodos para conseguir el comportamiento deseado. Las actividades iniciadas pueden devolver resultados de su ejecución mediante las intenciones.

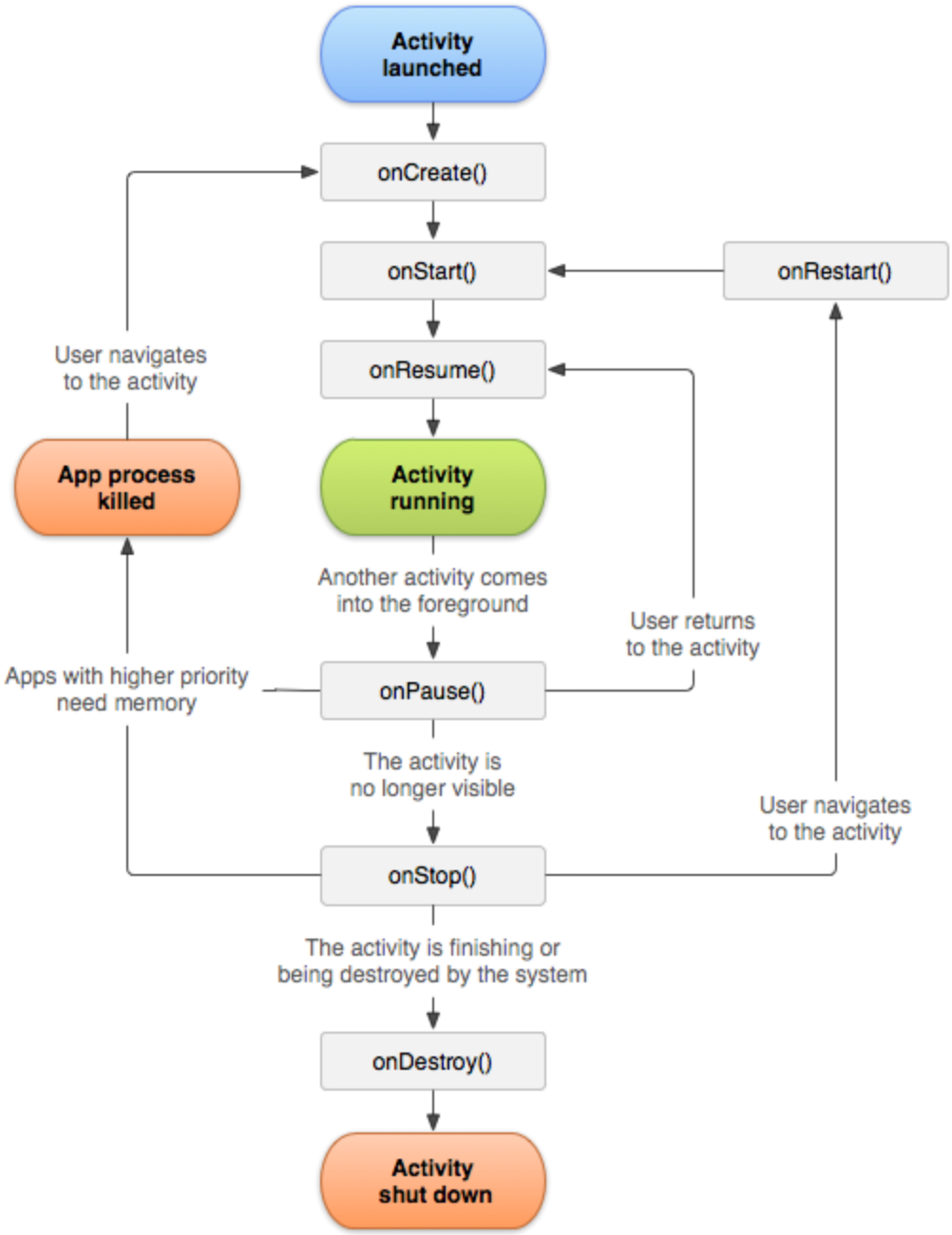

Figura 3-1: Ciclo de vida de una actividad

- **Intenciones** [25]

Una intención (o "Intent") "es un objeto de mensajería que puedes usar para solicitar una acción de otro componente de una app"[11]. Los casos de uso más utilizados son los siguientes:

  - **Iniciar una actividad,** para lo cual se crea una intención con la actividad a iniciar y los datos que sean necesarios. Además como se ha comentado anteriormente también se puede utilizar para devolver algún resultado de otra actividad.
  - **Iniciar un servicio,** para realizar tareas en segundo plano.
  - **Transmitir una emisión,** para avisar a otras aplicaciones sobre determinados eventos.

- **Servicios** [26]

Son componentes pensados para tareas que no necesiten interacción con el usuario, que requieran largos periodos de tiempo y que sea conveniente ejecutarlas en segundo plano, aunque es posible hacer funcionar un servicio en primer plano. Se pueden enlazar componentes a un servicio para controlar su ejecución y mostrar cambios en la interfaz gráfica, por ejemplo para que en el administrador de descargas implementado se pueda ver el progreso de un fichero descargándose.

A continuación se explicará en qué consiste la funcionalidad desarrollada, qué valor aporta a la aplicación o qué se mejora de ella. También se detallará el funcionamiento de esta, qué flujo hay que realizar para poder realizar esa

[11] https://developer.android.com/guide/components/intents-filters.html?hl=es

funcionalidad y los pasos a seguir. El código fuente se encuentra en https://github.com/jotagalilea/P2P-Android-App.

3.1 Sistema bloqueo usuarios

Descripción

Consiste en el descarte de aquellos mensajes procedentes de un usuario con el cual no se desea tener contacto.

A priori, la compartición de archivos entre amigos era uno de los principales atractivos del TFG anterior. Para un usuario que no está entre los amigos del usuario local no se implementó ningún mecanismo de seguridad más allá de la mera exclusión de su lista de amigos, cosa que por supuesto impide la descarga de cualquier archivo que se haya compartido. Sin embargo, esto no tenía en cuenta las peticiones de amistad, era posible enviar innumerables peticiones. Por ello se pensó en el bloqueo de usuarios para fortalecer la exclusión de aquellos que no formaran parte de nuestra red de amigos.

El sistema funciona de una manera sencilla: basta con pulsar un botón y escribir el nombre de usuario de aquel al que se quiera bloquear. Todas las posibles peticiones procedentes de dicho usuario son descartadas, aunque este procesamiento se realiza en el dispositivo destino. Una posible mejora sería implementar este sistema a nivel de PubNub.

Proceso e implementación

Para acceder a la actividad se ha de pulsar en el icono de 3 puntos de la barra de herramientas de la pantalla que aparece al iniciar la aplicación, y seleccionar la opción "Ver usuarios bloqueados". Esta actividad consta de una lista de usuarios bloqueados y un botón. El botón permite bloquear un usuario

cualquiera, y si es un amigo se pedirá confirmación. Al realizar la acción aparecerá en la lista. Además se puede bloquear directamente a un amigo desde la actividad principal tocando sobre él o ella y pulsando en el botón “Bloquear amigo”.

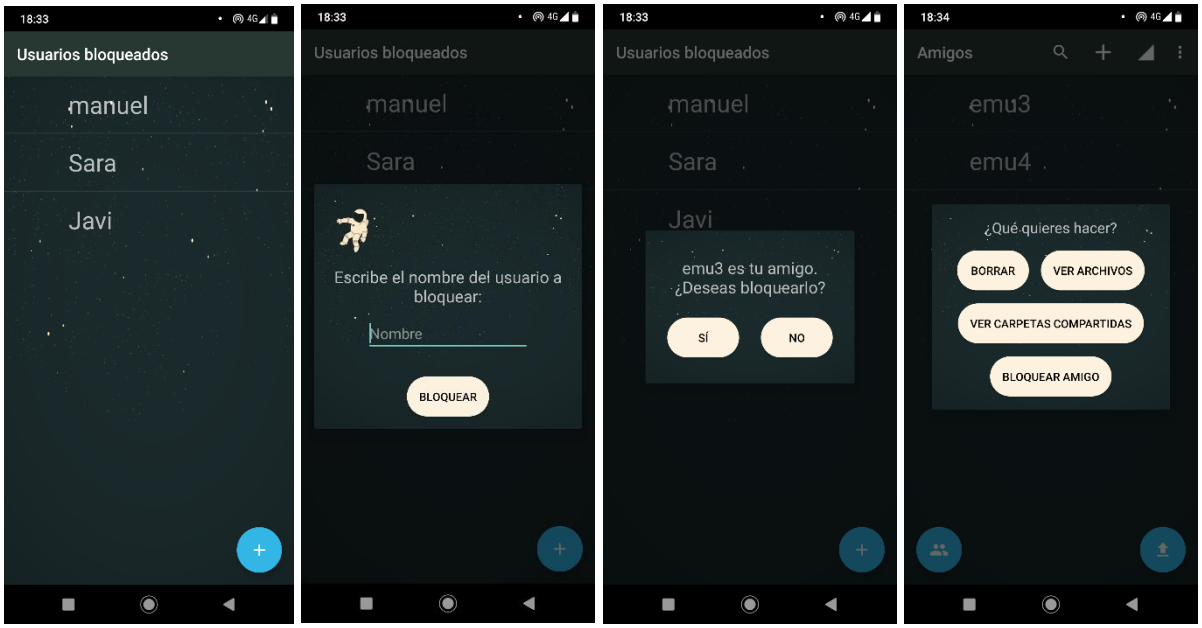

Figura 3-2: Bloquear usuario

Pulsando en uno de los usuarios de la lista se hace que aparezca un diálogo en el que se ofrece la posibilidad de desbloquear a ese usuario. Además, si se realiza una petición de amistad a un usuario bloqueado aparecerá un diálogo explicándolo y dando la opción de desbloquearlo para que las comunicaciones posteriores puedan producirse.

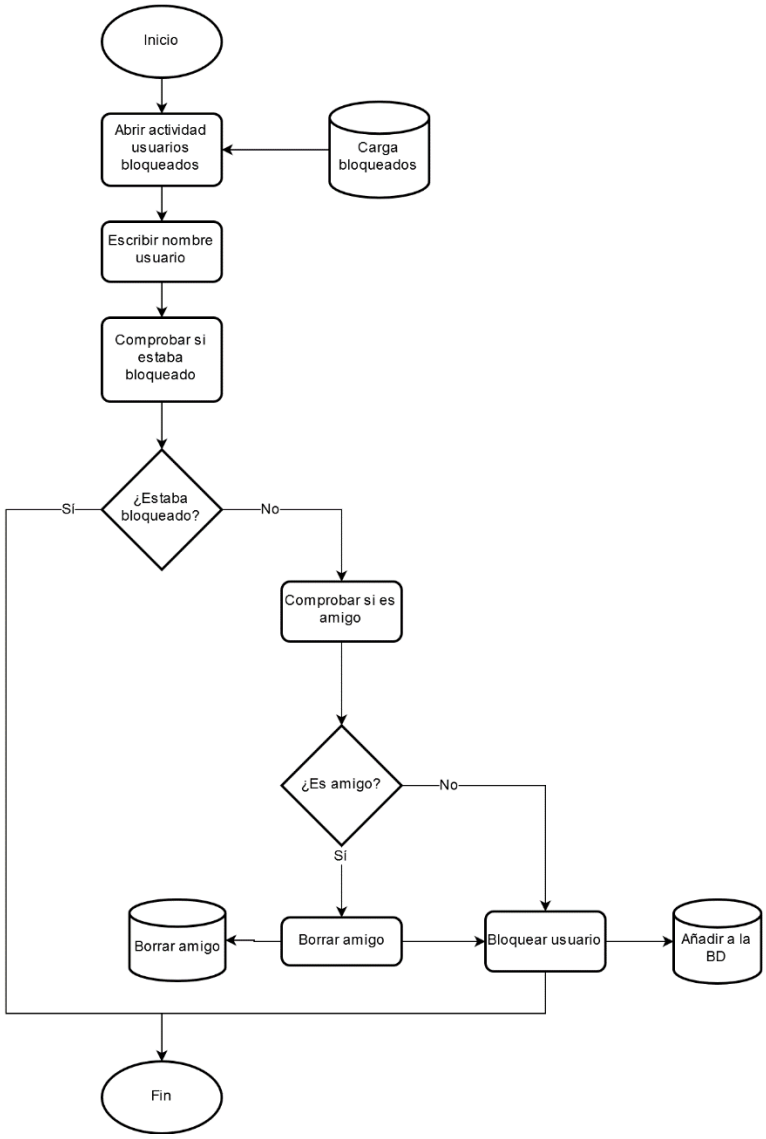

Figura 3-3: Diagrama bloquear usuario

3.2 Compartición de carpetas

Descripción

Para un usuario acostumbrado a utilizar plataformas de compartición de archivos como Dropbox, Google Drive, o las mencionadas en el apartado 2.3, resulta habitual pensar en compartir el contenido de una carpeta entera,

especialmente en el ámbito académico ya que los alumnos solemos organizar los ficheros por asignatura, tema, o año por ejemplo. Sin embargo, en esta aplicación no se implementó esta opción. Por ello desarrollamos la compartición de carpetas.

Estas carpetas compartidas podrían ser visibles para todo el mundo que se tuviera agregado como amigo, pero podría no ser deseable por todos los usuarios. Sería interesante que cada carpeta compartida sólo fuera visible por un subconjunto de amigos, ya que por ejemplo no tendría mucha utilidad compartir archivos académicos con familiares si estos no tienen ninguna relación o interés en ellos. Así se mostraría contenido coherente para cada uno. Por esta razón es buena idea realizar esta tarea como una función separada de la compartición de archivos ya implementada.

Por otra parte, se ha decidido que los ficheros que se comparten de una carpeta seleccionada sean solo los que estén en el nivel que se está explorando en ese momento, y que no se profundice en carpetas anidadas por simplicidad en la implementación y en el guardado en la base de datos.

Proceso e implementación

Para compartir una carpeta primero se ha de acceder al explorador de archivos pulsando sobre el botón flotante situado en la esquina inferior derecha de la primera pantalla; seguido de haber entrado en la carpeta que se desea compartir se ha de pulsar un nuevo botón flotante con un icono en forma de carpeta. Aparecerá un diálogo pidiendo confirmación y avisando de que no se compartirán carpetas anidadas. Si se pulsa "Sí" se mostrará un diálogo en el que se podrá seleccionar los usuarios a los que se les quiera proporcionar acceso. Finalmente la carpeta quedará compartida pulsando en "Añadir seleccionados".

Por otro lado, para ver las carpetas que se han compartido con anterioridad se ha de pulsar en el botón con tres puntos de la barra de la actividad principal y seleccionar "Ver carpetas compartidas". Aparecerán todas ellas en una lista

mostrando la ruta y el número de amigos con acceso. Pulsando sobre una se podrá ver los archivos que contiene o bien ver los usuarios a los que se les ha dado acceso, pudiendo añadir nuevos o eliminar alguno.

Para acceder a una carpeta compartida por alguno de los amigos se tiene que pulsar sobre el amigo deseado desde la actividad principal y después en "Ver carpetas compartidas". Si el amigo ha compartido alguna carpeta con el usuario esta aparecerá. Pulsando sobre una de las que aparezcan se podrá acceder a su contenido, pudiendo descargar o previsualizar uno de los archivos.

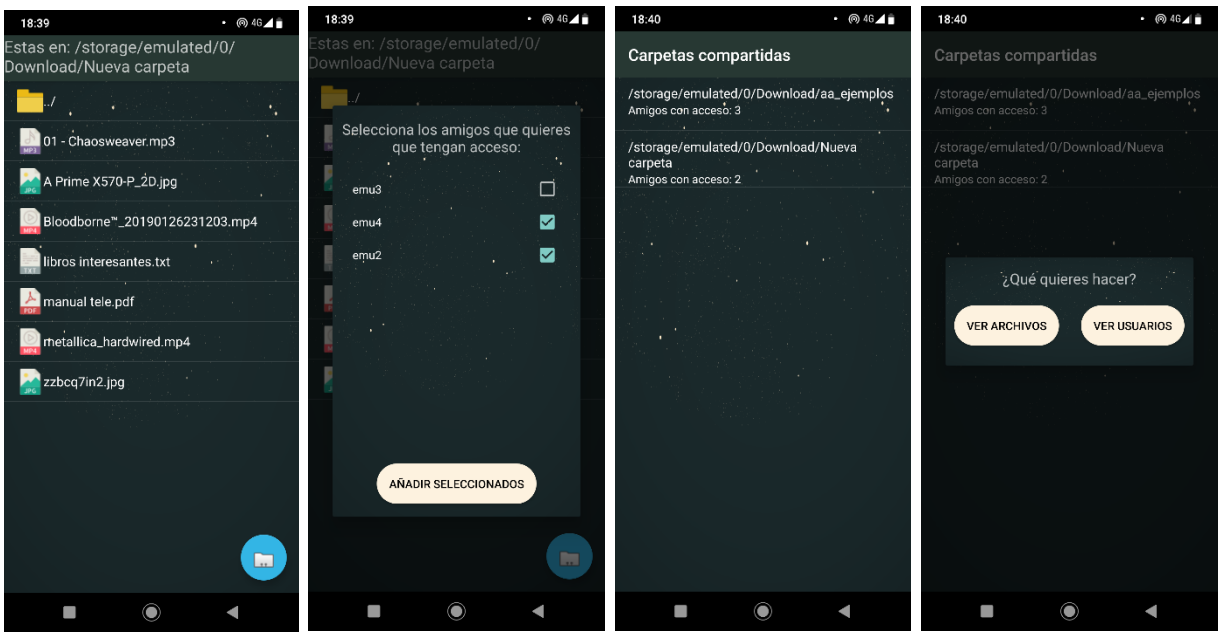

Figura 3-4: Compartición de carpetas 1

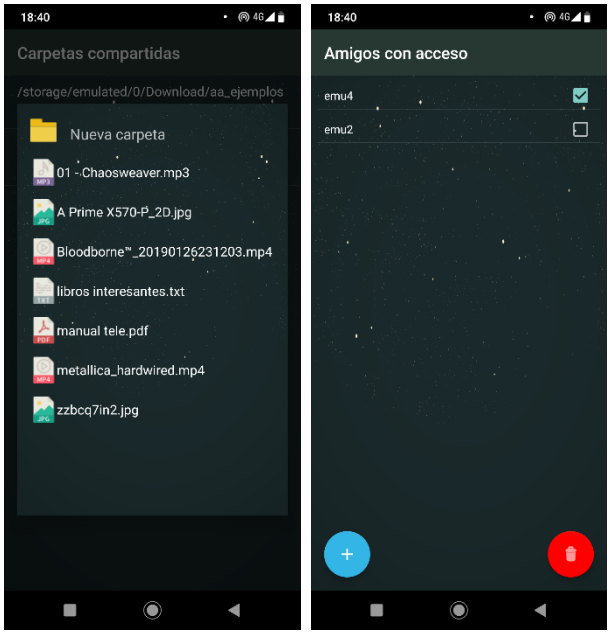

Figura 3-5: Compartición de carpetas 2

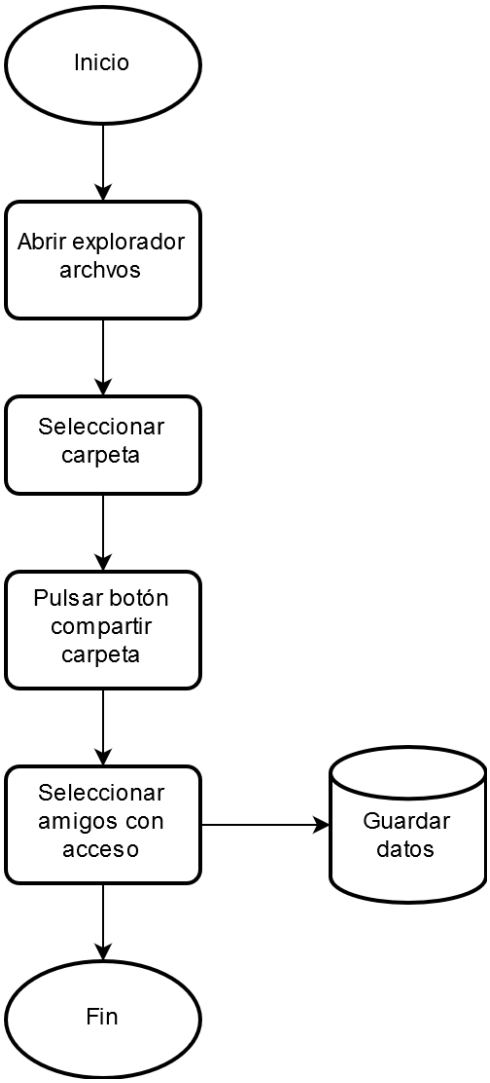

Figura 3-6: Diagrama compartir carpeta

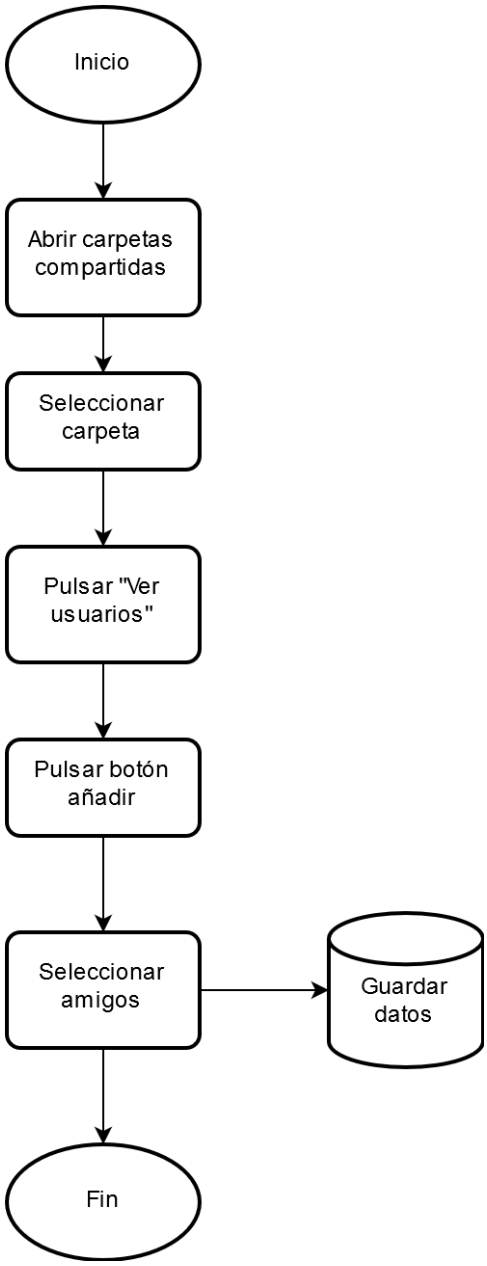

Figura 3-7: Diagrama añadir usuarios a carpeta compartida

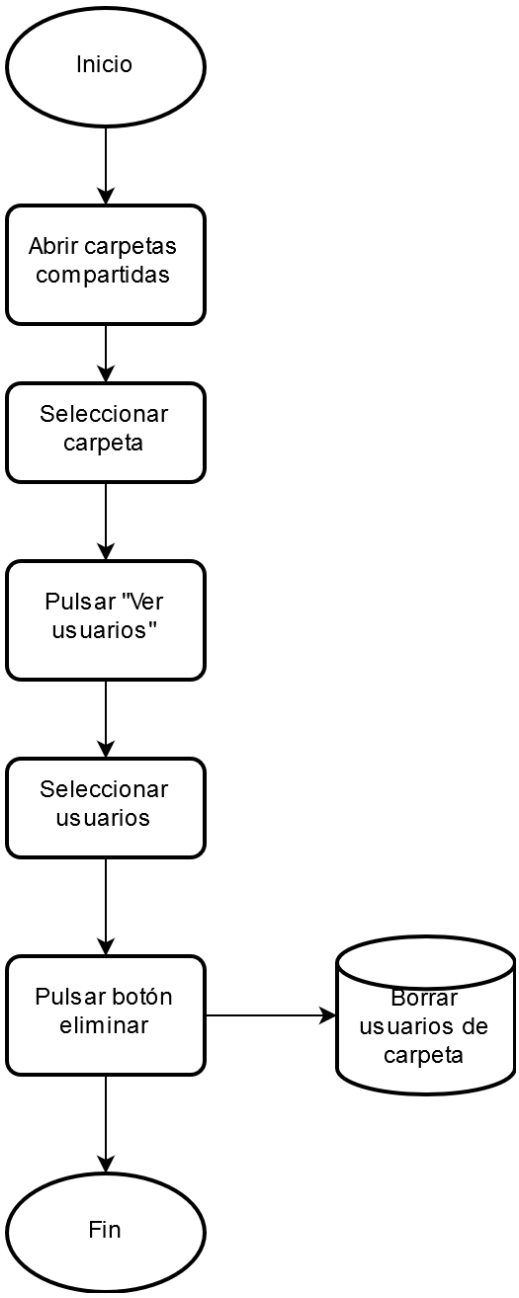

Figura 3-8: Diagrama borrar usuarios con acceso a una carpeta

3.3 Previsualización de archivos e iconos

Descripción

En ocasiones el archivo que se quiere descargar tiene un tamaño lo suficientemente grande como para que la transferencia no se pueda realizar en un período de tiempo razonablemente corto, así que se tomó la idea de previsualizar ficheros del trabajo previo.

Debido a los numerosos tipos de archivo que existen y a la complejidad variable de cada uno de ellos se ha tenido que hacer una selección teniendo en cuenta dos factores:

- ***Rareza:*** Es más útil para los usuarios poder ver una parte de un fichero con un formato común (como txt o pdf) que otros menos corrientes. Está basado en mi experiencia.
- ***Complejidad:*** Cuánta mayor sea, mayor tiempo de investigación es necesario para dar con la forma de fragmentar el fichero en cuestión, también probablemente buscando librerías.

A continuación, se exponen los tipos soportados y tamaños de previsualización relevantes para cada caso: txt 16KB, html 16KB, css 16KB, pdf 3 páginas, mp3 1MB, imágenes jpg y png se comprimen y reducen la calidad al 40%.

Además, se han añadido iconos a los ficheros en el explorador según su tipo para facilitar la identificación de cada uno. En el apartado de créditos se nombra al autor de estos.

Proceso e implementación

Para previsualizar un archivo hay que realizar los mismos pasos que si se fuera a descargar: pulsar sobre uno de los amigos de la lista de la actividad principal, elegir si se quiere ver los archivos o las carpetas compartidas y pulsar

sobre el archivo deseado. Aparecerá un diálogo preguntando si se quiere descargar o previsualizar y si el formato del fichero está dentro de los formatos soportados para la previsualización el botón "Previsualizar" estará habilitado. Pasado un breve período de tiempo tras pulsar el botón se habrá descargado un fragmento del fichero y se abrirá automáticamente con la aplicación predeterminada para este tipo de fichero, o bien aparecerá un menú que permitirá seleccionar una de entre varias aplicaciones para abrirlo.

El diagrama de flujo de este caso está incluido en el diagrama "Descargar/Previsualizar archivo".

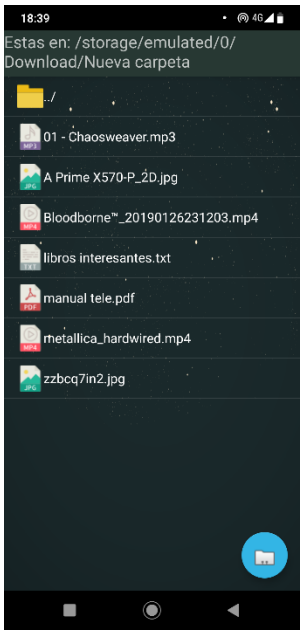

Figura 3-9: Ejemplo de iconos en el explorador

3.4 Sistema de descargas y subidas

Descripción

El sistema de descargas y subidas antiguo era algo rudimentario. Si se abandonaba la actividad "Profile" el comportamiento era impredecible por su dependencia, codificaba todo el archivo de golpe (generando el problema que se

describe en el apartado E), lo enviaba a trozos sin tener en cuenta que de todos modos un archivo muy grande no iba a caber en memoria, mostraba muy poca información sobre el estado de la descarga y no permitía descargas en paralelo ni tenía ningún tipo de cola. Una implementación con muchas mejoras posibles que necesitaba una renovación.

Ahora cuenta con cola de hilos de subida, cola de mensajes de petición en la parte de descargas, administrador tanto de hilos de subida como de bajada, descargas paralelas, mejora del consumo de memoria (explicado en el siguiente apartado), y aumento de información sobre el estado de una descarga.

Proceso e implementación

Las subidas se realizan utilizando un administrador de hilos. Este es un hilo independiente que almacena en una cola hasta 4 hilos para enviar archivos, funcionando sólo uno de ellos a la vez para reducir el uso de CPU y el consumo de batería.

La parte de descargas ha sido desarrollada como un servicio porque se quiso que funcionara en segundo plano y que hubiera varias descargas al mismo tiempo. También se ha limitado el número de descargas paralelas a 2 por motivos de ahorro de batería y uso de CPU.

El servicio de descargas funciona así: Se cuenta por un lado con un hilo ("manager") que se encarga de pasar los mensajes al hilo de descarga que corresponda, por otro lado una lista de objetos de descarga que será la que se muestre en la interfaz gráfica, con hasta 2 hilos de descarga activos (según la implementación actual), y con una cola de mensajes de petición cuyo objetivo es realizar dichas peticiones de ficheros cuando se disponga de hilos libres. Cada uno de ellos tiene un monitor asociado, así se procura el correcto paso de mensajes a cada una de las descargas y si no tienen trabajo se mantienen pausados sin consumir CPU. Los hilos de descarga se encuentran alojados en un HashMap que tiene como clave el nombre del fichero que gestionan y como valor el par <monitor,

hilo de descarga>. Cuando llega un mensaje se puede identificar si pertenece a una descarga o a otra rápidamente, pues este viene con la información del nombre del fichero asociado. Así se obtiene el monitor del hilo que corresponda y se le notifica que ha llegado un mensaje que debe tratar él, tomando también la referencia a ese hilo para pasarle acto seguido dicho mensaje.

Para realizar la sincronización de la mejor forma que se me ocurrió utilicé bloques "synchronized", monitores y variables condicionales. Al estar Android basado en una versión modificada del kernel de Linux [25] sabía que podía utilizar este mecanismo que hemos estudiado en la carrera. La elección de usar "while" en vez de "if" en las partes en las que se comprueba el estado de estas variables fue tomada teniendo en cuenta esta característica del kernel y apoyándonos tanto en el correcto funcionamiento de la implementación como en la recomendación del manual de Linux [26] y en un manual de concurrencia Java [27]. Estas fuentes advierten que algunas implementaciones del kernel pueden despertar múltiples hilos que previamente hayan llamado al método "wait()" por lo que es muy recomendable encerrar esa llamada en un "while" que no deje avanzar la ejecución a menos que el estado de la variable condicional haya cambiado realmente.

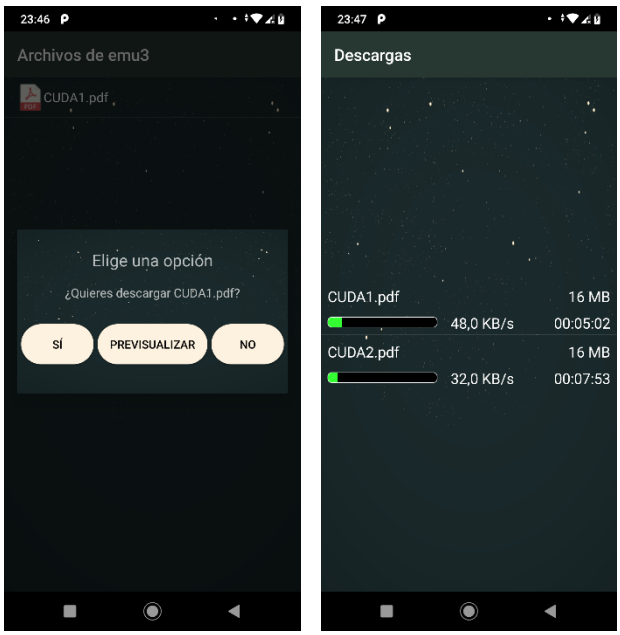

Figura 3-10: Ejemplo del estado de las descargas

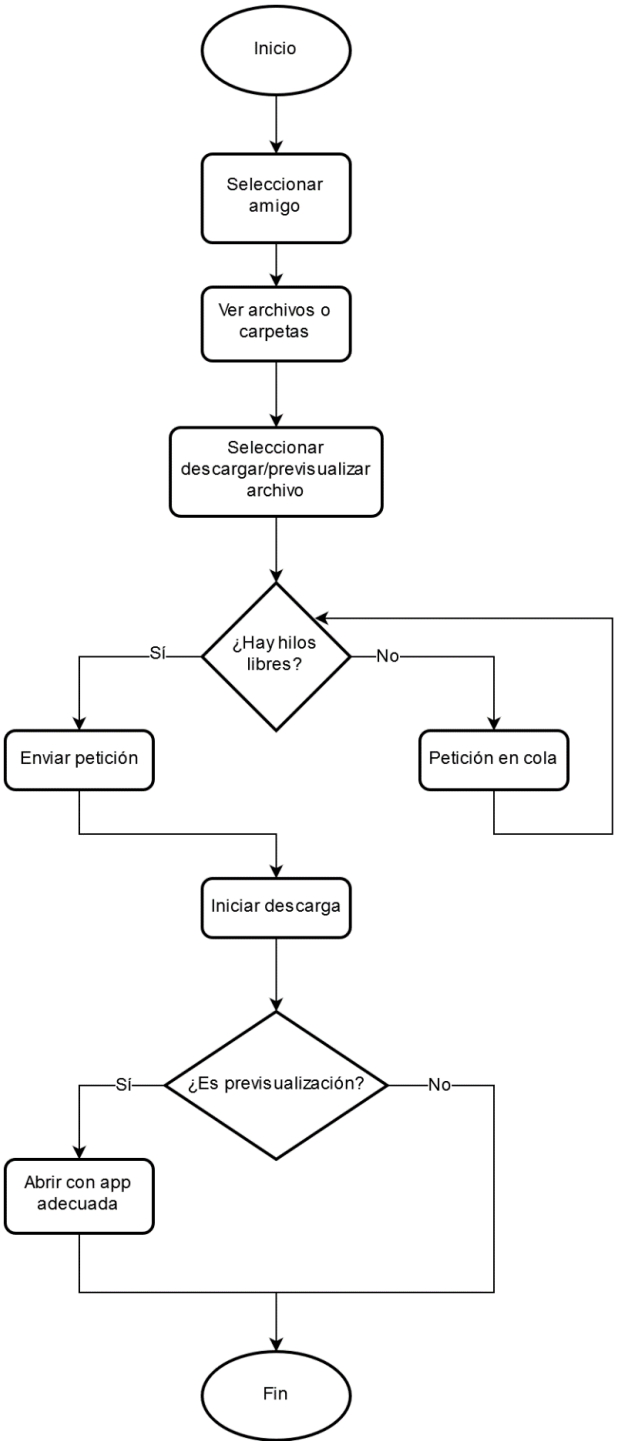

Figura 3-11: Diagrama descargar/previsualizar archivo

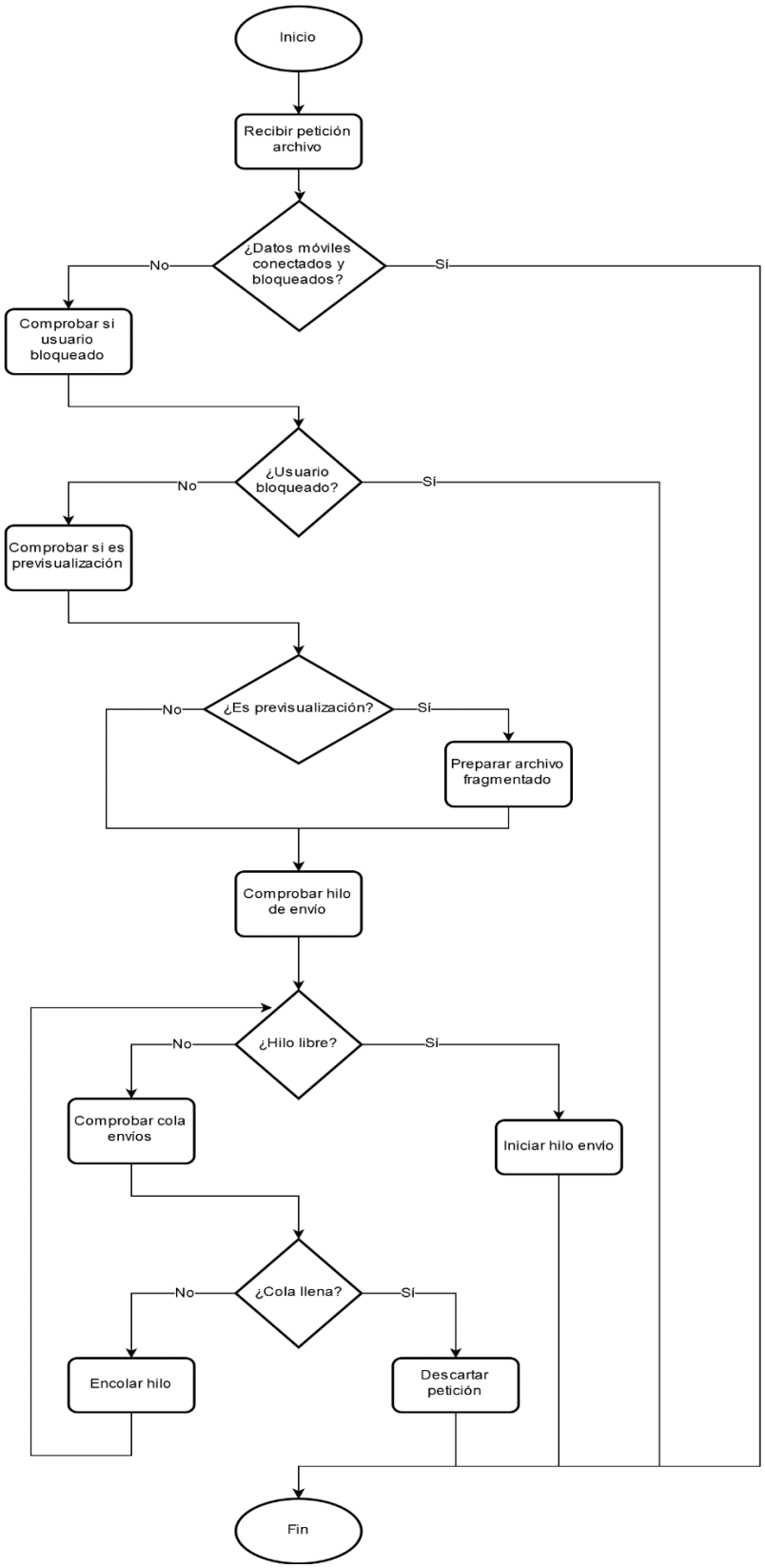

Figura 3-12: Diagrama enviar archivo

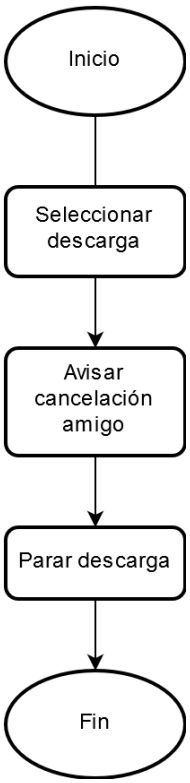

Figura 3-13: Diagrama cancelar descarga

3.5 Reducción del consumo de memoria

Descripción

En el proyecto previo a este TFG existía un problema en la codificación de un fichero que estuviera siendo enviado. El funcionamiento consistía en lo siguiente: se copiaba el contenido del fichero en un array de bytes para más tarde codificarlo íntegramente en un string utilizando Base64, ocupando un 133% del tamaño original del archivo según la memoria del TFG anterior. No obstante, al realizar una prueba con un archivo de 112 MB se pudo comprobar haciendo uso de la herramienta "Profiler" integrada en Android Studio que la aplicación no tardaba en incrementar el uso de memoria hasta cerrarse por desbordamiento, llegando a utilizar más de 600 MB en este caso. Esta forma de proceder supone un problema puesto que para sobrepasar el límite de memoria permitido para una aplicación bastaría con tratar un fichero lo suficientemente grande.

Implementación

Para solucionar esto se pensó en transmitir el fichero a trozos, partiéndolo antes de la codificación. Aunque en la memoria del trabajo anterior explicaban una solución muy similar solo que troceando el string codificado, no reparamos en ello hasta que ya estuvo implementado con lo que, por otro lado, nos dimos cuenta de que la versión que debió ser entregada no corresponde con el código alojado en Github que es del que se parte en el desarrollo.

3.6 Bloqueo para envíos usando la red del operador

Descripción

Actualmente las compañías de telefonía móvil están empezando a ofrecer en España tarifas planas de datos para la actual red móvil 4G y futura 5G. Por ahora la gran mayoría de clientes no tienen contratado este tipo de servicio, por tanto es buena idea dar a los usuarios la opción de permitir o no el uso de esta red para evitar posibles sobrecostes en la factura.

Para reducir estos costes se tenía que identificar en qué casos las comunicaciones pueden transmitir un volumen de datos significativamente grande. Las peticiones de amistad y el envío de información sobre archivos o carpetas compartidas no suponen grandes intercambios de información, por lo que no se han tenido en cuenta. Los dos casos de uso a tener en cuenta en los cuales esto ocurre son el envío y la recepción de ficheros. Sin embargo, no hemos introducido la limitación de uso en ambos casos porque estamos ante acciones sobre las que podemos hacer una distinción clara: La recepción de un fichero responde a una petición de éste previa y consciente por parte del usuario. Podemos asumir que si decide descargar algo es porque se lo puede permitir acorde a la tarifa que tenga contratada. Sin embargo, el envío, tal y como está implementada la aplicación, es una tarea que ocurre de forma imperceptible, de manera que el usuario no es

directamente consciente del uso de datos que está haciendo la aplicación y para conocer este dato tendría que acudir al registro de uso de datos de las aplicaciones dentro de los ajustes de Android, algo poco cómodo. Por esta razón, el impedimento de uso de la red sólo se ha implementado para los envíos.

Proceso e implementación

Se ha introducido un botón con forma triangular en la barra superior de la actividad principal que bloquea y desbloquea el uso de la red con solo una pulsación.

Para llevar a cabo esta sencilla tarea se toma el servicio de conectividad del sistema y simplemente se le pregunta si está conectado a la red o si está conectándose. Si se está utilizando wifi no ocurre nada, y en caso de usarse la red se comprueba si el usuario lo permite o no. Se ha implementado en el método que trata los mensajes de petición de archivos.

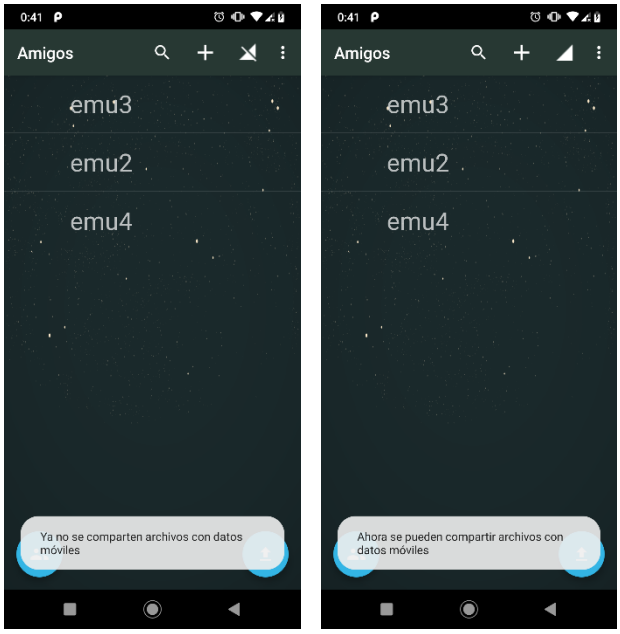

Figura 3-14: Bloqueo de datos móviles

3.7 Revisión del borrado de amigos

Descripción

El borrado de amigos ha cambiado ligeramente con respecto al flujo original. Tras haber incluido las carpetas compartidas ahora es necesario comprobar en este proceso si dicho usuario tiene acceso a alguna carpeta. En caso afirmativo se elimina.

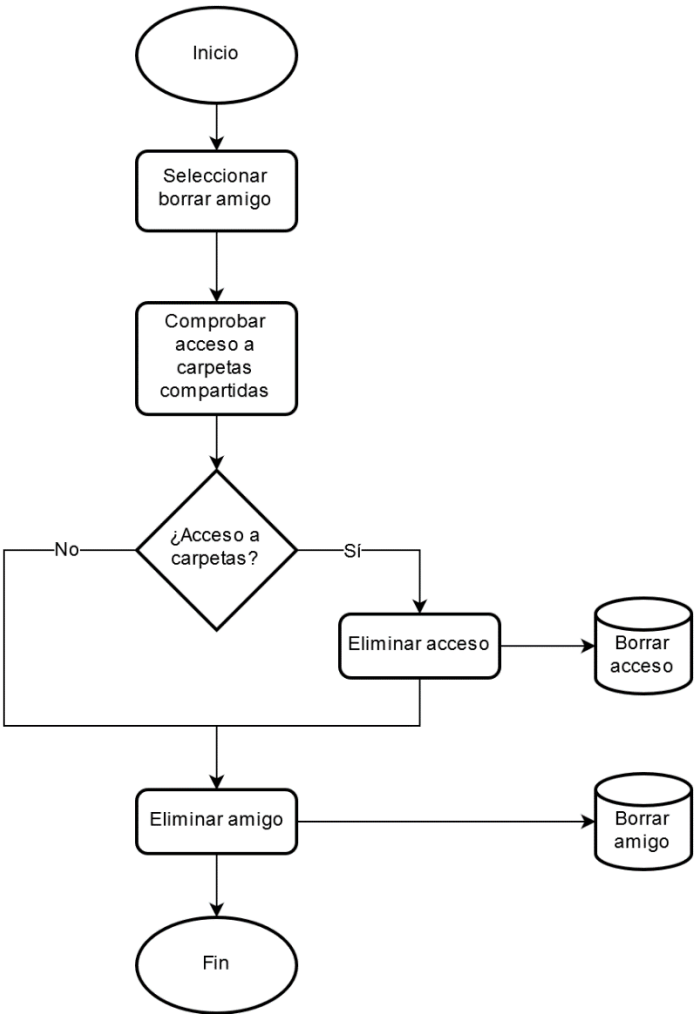

Figura 3-15: Diagrama borrado de amigo

Capítulo 4 - Resultados

En este último capítulo se tratarán tareas que podrían llevarse a cabo para la mejora de este trabajo seguido de unas conclusiones acerca de si se han cumplido los objetivos y qué ha supuesto este proyecto.

4.1 Trabajo futuro

A lo largo del desarrollo han ido surgiendo nuevas ideas y mejoras que podrían ser interesantes para este trabajo pero que han sido guardadas para su implementación en caso de disponer del tiempo necesario para llevarlas a cabo. Algunas sí se han podido ejecutar, pero otras han tenido que ser descartadas. Para comenzar el último capítulo quiero exponer estas ideas de cara a un futuro proyecto que retome el nuestro o como recordatorio para mí mismo si retomara el desarrollo más adelante, ya que se trata de una aplicación muy útil y de la máxima confianza para mí.

Verificación de la integridad de los archivos tras la descarga.

Codificación de los mensajes. Por ahora todos los identificadores de los mensajes JSON y todos los datos se envían como cadenas de caracteres planas. Aunque PubNub ya implementa encriptación TLS punto a punto y AES en el envío de los mensajes se podría añadir otra capa. La encriptación TLS es decodificada en los centros de datos por los que pasan los mensajes para ser en el reenvío recodificada. [28]

Mejoras en la interfaz gráfica. Si bien se han mejorado las actividades y los diálogos, siempre se pueden añadir o modificar cosas que modernicen las vistas.

Dar soporte a más formatos de archivo para previsualizaciones, especialmente aquellos formatos más comunes en los dispositivos a los que no se les ha podido dedicar más tiempo debido a su complejidad para ser fragmentados.

Ellos requerían hacer un estudio profundo de su estructura o bien encontrar alguna librería que pudiera ser utilizada en Android. Entre otros formatos, estaba pensado poder previsualizar mp4, avi, docx, xlsx y pptx.

Envío de archivos o carpetas comprimidos, realizando la compresión en el emisor y descompresión en el receptor en tiempo real.

Mejora del bloqueo de usuarios, si es posible implementándolo a nivel de API en PubNub para que aquellas peticiones no deseadas sean rechazadas antes de la recepción en el destinatario.

Mejora de la cancelación de descargas activas. El sistema no permitía enviar un nuevo mensaje al remitente de un fichero tras haber cancelado su descarga, al menos durante el periodo de tiempo en que estaba previsto que se completara. Esto se debe al modelo que sigue PubNub publicador-suscriptor. Cuando los datos de un fichero son publicados al suscriptor que los ha pedido la red permite que estos sean enviados muy eficientemente. Sin embargo, el reenvío desde el centro de datos por el que pasan los fragmentos del fichero es notablemente más lento, por lo que el suscriptor tarda mucho más tiempo en recibir todos los datos. Esto es, traducido para nuestra aplicación, en que las subidas son mucho más rápidas que las descargas. Si el suscriptor decide cancelar la transmisión es seguro que en ese momento ha recibido menor número de mensajes de datos que los que ha enviado el publicador. Tal y como está implementado, el suscriptor cuando cancela destruye la parte del fichero recibida, envía notificación de ello al remitente y detiene el hilo que recibe los mensajes. No obstante, el centro de datos continúa enviando mensajes por un tiempo y el canal se queda ocupado, imposibilitando nuevos mensajes hasta que el programa que se está ejecutando en el centro de datos decide descartarlos. Esto ha sido mitigado creando un nuevo objeto cliente en la aplicación cuyo único objetivo es poseer otro canal de envío de forma que el cliente principal pueda seguir atendiendo otras peticiones procedentes del mismo dispositivo. Este comportamiento podría inducir (y de hecho lo hizo) a que pensase en crear tantos clientes de envío como fueran necesarios para tener a su vez tantos canales disponibles como clientes, pero se

obtuvo el mismo resultado que con solo un cliente nuevo. Tras investigar y tratar de subsanar esto, decidí conformarme con el comportamiento conseguido hasta el momento. He de dejar esta tarea para el futuro.

Guardado del estado de las descargas añadiendo funciones de pausa y reanudación para poder continuarlas en otro momento.

Funcionamiento en segundo plano desde el arranque, haciendo no necesario el inicio manual de la aplicación.

Control de los ficheros y carpetas compartidos. Actualmente si el usuario borra o modifica un fichero o carpeta de forma externa este cambio no aparece reflejado en la base de datos si estaban insertados.

Implementación de nuevos componentes de Android. En cada versión de actualización del sistema operativo de Android, vemos que se añaden nuevas mejoras, funcionalidades o herramientas que hacen más atractiva la interacción con las aplicaciones. Dado que la app es dentro de lo que cabe simple, se puede intentar modernizar usando nuevos componentes.

4.2 Conclusiones

Este proyecto ha supuesto un reto diferente a lo que estaba acostumbrado en la carrera. Se debía tomar un proyecto desarrollado por otros compañeros, adaptándome a no una, sino dos nuevas tecnologías que desconocía y que proveían una gran alternativa para las comunicaciones "peer-to-peer". Se tenía que entender el concepto para así poder avanzar, dedicando el tiempo necesario para comprender el paradigma sobre el que estaba construida una de dichas tecnologías.

Los principales objetivos se han cumplido, poder proveer cualquier tipo de función que realmente fuera relevante y que estuviera presente en una aplicación móvil moderna. De entre todas las que han emergido se ha tenido que realizar una

selección dedicando tiempo de estudio a su posible complejidad para obtener las idóneas.

Cabe recalcar la especial dificultad que ha desentrañado implementar algo tan cotidiano para mí como un buen sistema de descarga y subida de archivos. Este tipo de funciones están integradas en todo tipo de aplicaciones de nuestro día a día, como por ejemplo la tienda de Android Google Play, BitTorrent, u otros tantos programas. Se ha podido comprobar que diseñarlo y que sea lo suficientemente robusto ha sido todo un reto. La paralelización, las colas, la gestión de los canales de transmisión, la integración del sistema en un servicio, el control sobre el consumo de CPU y otras cosas han añadido un importante valor que estaba ausente en este apartado.

4.3 Conclusions

This project has meant a different challenge from I was used to in the degree. I was supposed to take a project that was developed by other mates, adapting myself to not only one but two new technologies unknown for me and that they provided a great choice for "peer-to-peer" communications. I had to understand the concept in order to move on, spending the time necessary to get the paradigm one of these technologies was built over.

The main objectives have been achieved, to provide any kind of functionality that could be relevant actually and that it was present in a modern mobile application. I have had to select some of them among every that has emerged, spending some time studying their possible complexity in order to get those that were appropriate.

It should be emphasized the special difficulty that such a usual thing as a good upload and download system has been unraveled for me to be implemented. This kind of functionalities are in a lot of our daily applications, like Google Play, BitTorrent, and so many other programs. I have been able to check that design it

and make it strong enough has been a real challenge. Parallelization, queues, transmission channels management, integration as a service, some control over CPU usage and many other things have added an important value that was missing in this topic.

BIBLIOGRAFÍA

[1] El periódico, Articulo Web de El Periódico, 5 10 2017. [En línea]. Available: https://www.elperiodico.com/es/sociedad/20171005/cada-vez-hay-mas-ninos-con-movil-desde-los-10-anos-6333468.

[2] Emule, Página principal de Emule, [En línea]. Available: https://www.emule-project.net/.

[3] BitTorrent, Pagina principal de BitTorrent, [En línea]. Available: https://www.bittorrent.com.

[4] Wikipedia, «Peer to peer,» [En línea]. Available: https://es.wikipedia.org/wiki/Peer-to-peer.

[5] Wikipedia, «Peer to peer Structured networks,» [En línea]. Available: https://en.wikipedia.org/wiki/Peer-to-peer#Structured_networks.

[6] Wikipedia, «Peer to peer Unstructure networks,» [En línea]. Available: https://en.wikipedia.org/wiki/Peer-to-peer#Unstructured_networks.

[7] «WebRTC,» [En línea]. Available: https://webrtc.org/.

[8] «PubNub,» [En línea]. Available: https://www.pubnub.com/.

[9] «PubNub channel topologies,» [En línea]. Available: https://www.pubnub.com/developers/tech/key-concepts/topologies/.

[10] Wikipedia, «Android,» [En línea]. Available: https://es.wikipedia.org/wiki/Android.

[11] Uc3m, «Android-Introduccion,» [En línea]. Available:

https://sites.google.com/site/swcuc3m/home/android/portada.

[12] Andro4all, «Versiones de Android,» [En línea]. Available: https://andro4all.com/2018/08/versiones-android-historia.

[13] Android, «Android History,» [En línea]. Available: https://www.android.com/intl/es_es/history/#/marshmallow.

[14] Android.com, «Historia Android,» [En línea]. Available: https://www.android.com/intl/es_es/history/#/donut.

[15] Source Android, «Android Open Source Proyect-AOSP,» [En línea]. Available: https://source.android.com/.

[16] Apache, «Apache License,» [En línea]. Available: https://www.apache.org/licenses/LICENSE-2.0.

[17] Developers Android, «Connectivity,» [En línea]. Available: https://developer.android.com/guide/topics/connectivity.

[18] Developers Android, «Supported media formats,» [En línea]. Available: https://developer.android.com/guide/topics/media/media-formats.

[19] Developers Android, «Arquitectura de la plataforma,» [En línea]. Available: https://developer.android.com/guide/platform?hl=es.

[20] SQLite Consortium, «SQLite,» [En línea]. Available: https://www.sqlite.org/index.html.

[21] Developers Android, «Android Studio,» [En línea]. Available: https://developer.android.com/studio/intro/?hl=ES.

[22] Academia Android, «IDE: Entornos Integrados de Desarrollo para Android,» [En línea]. Available: https://academiaandroid.com/ide-entornos-integrados-de-

desarrollo-para-android/.

[23] Developers Android, «Google Play Store,» [En línea]. Available: https://developer.android.com/distribute/google-play?hl=es.

[24] «Android Developer - Actividades,» [En línea]. Available: https://developer.android.com/guide/components/activities?hl=es_419.

[25] «Android Developer - Intents,» [En línea]. Available: https://developer.android.com/guide/components/intents-filters.html?hl=es.

[26] «Android Developer - Servicios,» [En línea]. Available: https://developer.android.com/guide/components/services?hl=es_419.

[27] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Android_(operating_system).

[28] «linux.die.net,» [En línea]. Available: https://linux.die.net/man/3/pthread_cond_wait.

[29] «How to do in Java,» [En línea]. Available: https://howtodoinjava.com/java/multi-threading/wait-notify-and-notifyall-methods/.

[30] «PubNub. Point-to-Point Encryption (Transport Layer Security),» [En línea]. Available: https://www.pubnub.com/developers/tech/security/.

[31] «PubNub,» [En línea]. Available: https://www.pubnub.com/.

[32] «PubNub,» [En línea]. Available: https://support.pubnub.com/support/solutions/articles/14000046386-what-is-pubnub-.

[33] Andro4all, «Versiones de Android,» [En línea]. Available:

https://andro4all.com/2018/08/versiones-android-historia.

CRÉDITOS

Iconos del explorador de archivos hechos por https://www.flaticon.com/authors/smashicons, https://www.flaticon.com/

bajo licencia http://creativecommons.org/licenses/by/3.0/ CC 3.0 BY