

Grado en Ingeniería Informática

Aplicación Android para Fomentar y Facilitar la Ayuda entre Personas

Especialidad en Ingeniería del Software

Autor:

Aarón Jiménez García

Directora:

Maria Jose Casañ Guerrero

Departamento de la directora:

Enginyeria de Serveis i Sistemes d'Informació (ESSI)



25/06/2019

Facultad Informática de Barcelona (FIB)

Universitat Politècnica de Catalunya – BarcelonaTech

Resumen

La tecnología con el paso de los años se ha ido expandiendo por diversas áreas de la vida humana, ocupando un papel principal en ella. Esta gran expansión se debe, sin duda alguna, a dos principales factores: la disponibilidad de Internet en cualquier lugar y al gran auge de los teléfonos inteligentes y el mercado de aplicaciones para teléfonos móviles. Los ámbitos de la colaboración, la ayuda y la forma de compartir han sido de los más afectados y los que más pueden evolucionar de entre todos. Este proyecto propone utilizar dicha expansión para fomentar el consumo colaborativo a través de la creación de una aplicación móvil para dispositivos Android, la cual, fomente y facilite la colaboración entre diferentes personas que requieran ayuda que estén cerca entre sí.

Resum

La tecnologia amb el pas dels anys s'ha anat expandint per diverses àrees de la vida humana, ocupant un paper principal en ella. Aquesta gran expansió es deu, sense cap dubte, a dos principals factors: la disponibilitat d'Internet en qualsevol lloc i al gran augment dels telèfons intel·ligents i el mercat d'aplicacions per telèfons mòbils. Els àmbits de col·laboració, l'ajuda i la forma de compartir han estat dels més afectats i els que més poden evolucionar d'entre tots ells. Aquest projecte proposa utilitzar aquesta gran expansió per fomentar el consum col·laboratiu mitjançant la creació d'una aplicació mòbil per dispositius Android, la qual, fomenti i faciliti la col·laboració entre diferents persones que necessitin ajuda i estiguin a prop.

Abstract

Over the years, technology has expanded through different areas of human life and has acquired an important role in it. This large expansion is undoubtedly due to two main factors: the possibility of using the Internet anywhere and the big expansion of smartphones and the mobile application market. The areas of collaboration, help and sharing have been the most affected and they are the ones that can evolve the most. This project has the purpose of using the expansion in order to promote collaborative consumption through the creation of a mobile application for Android devices which will encourage and facilitate collaboration between different people who require help.

Abreviaturas y Siglas

ACID: Atomicity, Consistency, Isolation, Durability.

API: Application Programming Interface.

CSS: Cascading Style Sheets.

DTO: Data Transfer Object.

GDPR: General Data Protection Regulation.

GPS: Global Positioning System.

HTML: HyperText Markup Language.

HTTP: HyperText Transfer Protocol.

JSON: JavaScript Object Notation.

LOPD: Ley Orgánica de Protección de Datos.

LOPDGDD: Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales.

MVC: Model View Controller.

MVP: Model View Presenter.

ONG: Organización No Gubernamental.

REST: Representational State Transfer.

SQL: Structured Query Language.

UML: Unified Modeling Language.

URL: Uniform Resource Locator.

Glosario

Android: Sistema operativo diseñado para dispositivos portátiles como teléfonos inteligentes, tabletas y relojes.

API Rest: Una API es una serie de funciones, objetos y métodos que permiten a los desarrolladores interactuar con una tecnología específica. Rest es un estándar que describe como deben comunicarse entre sí los sistemas a través de la red.

Asíncrono: Que no tiene lugar en completa correspondencia temporal con otro proceso o con la causa que lo produce.

Backend: Parte de una aplicación destinada a realizar toda la parte lógica, es decir, toda la programación desde las funciones hasta la base de datos que hace que la aplicación funcione.

Backlog: En Scrum, lista de tareas que se quieren realizar a lo largo del desarrollo de un proyecto.

Brainstorming: Brainstorming o lluvia de ideas es una técnica donde varias personas que forman un grupo ponen en común ideas para una finalidad como por ejemplo un proyecto.

Bytecode: Conjunto de instrucciones en formato binario de la Java Virtual Machine (JVM).

Callback: Función de programación que se da a otra para que cuando finalice la ejecución de la segunda la primera se llame.

Encriptar: En criptografía, procedimiento mediante un algoritmo con una clave que permite transformar un mensaje en otro que sea incomprensible o que al menos sea difícil de comprender.

Endpoint: Dirección web de un *backend* que responde a una petición de un cliente.

Entidad: En ciencias de la computación, en base de datos, una entidad es la representación de un objeto o concepto del mundo real.

Feedback: Retroalimentación, reacción, respuesta u opinión que da un interlocutor sobre un tema.

Framework: Conjunto estandarizado de conceptos, prácticas y conceptos que permiten enfocar un tipo de problema y permiten solucionar otros similares.

Frontend: Parte de una aplicación destinada a interactuar con el usuario, es decir, todas las tecnologías de diseño y desarrollo de interfaz.

Hardware: Conjunto de partes físicas o materiales que forman parte de una computadora o un sistema informático.

Hashtag: Cadena de caracteres formada por una o varias palabras seguidas por una almohadilla.

Inception: Etapa donde los participantes de un proyecto planean que es lo que se quiere llevar a cabo en un proyecto a partir de reuniones.

Paginación: Técnica donde la información es dividida en partes del mismo tamaño.

Parsear: En ciencias de la computación, proceso de analizar la sintaxis de un documento o sentencia escrita en un lenguaje particular.

Renderizar: Proceso de generar una imagen visible para el entendimiento humano a partir de información digital.

Rollback: Operación que devuelve un sistema a un estado previo normalmente utilizado cuando se trata con bases de datos.

Scroll: Desplazar o deslizar la ventana que se muestra en una aplicación informática.

Deserialización: En ciencias de la computación, consiste en un proceso de decodificación de un objeto transmitido a través de la red como una serie de bytes u otro formato más humanamente legible.

Servidor: Unidad informática que proporciona servicios a otros sistemas conectados con ella a través de la red.

Síncrono: Que tiene lugar en completa correspondencia temporal con otro proceso o con la causa que lo produce.

Smartphone: Dispositivo táctil con un sistema operativo complejo capaz de ampliar sus funcionalidades mediante la instalación de aplicaciones y hacer uso de Internet.

Software: Conjunto de programas, instrucciones y reglas para ejecutar ciertas tareas en una computadora.

Sprint: Etapa de la metodología Scrum destinada a la programación de algunas funcionalidades elegidas por el equipo de desarrollo que suele durar entre dos a cuatro semanas.

Token: En ciencias de la computación un token de seguridad es una cadena de caracteres que tiene un significado coherente y contiene información que permite acceder a un usuarios a recursos protegidos de por ejemplo un servidor.

Socket: Tecnología que proporciona un canal de comunicación bidireccional entre dos programas a través de la red.

Tabla de Contenidos

1. Introducción.....	8
2. Contextualización.....	9
3. Formulación del Problema.....	10
3.1. Motivación.....	10
3.2. Objetivos.....	10
3.2.1. Objetivos Tecnológicos.....	10
3.2.2. Objetivos Temáticos.....	11
4. Alcance del Proyecto.....	12
5. Obstáculos.....	13
5.1. Tiempo de Realización.....	13
5.2. Acceso a Internet.....	13
5.3. Aparición/Cambios de Funcionalidades.....	13
5.4. Experiencia en Programación Android.....	13
6. Actores Implicados.....	14
6.1. Directora del Proyecto.....	14
6.2. Equipo del Proyecto, Desarrollo y Pruebas.....	14
6.3. Equipo de Mantenimiento del Software.....	14
6.4. Usuarios Estándar.....	15
6.5. Organizaciones Sin Ánimo de Lucro.....	15
6.6. Pequeños Comercios.....	15
7. Estado del Arte.....	16
7.1. Estudio de Mercado.....	16
7.1.1. Impossible People.....	16
7.1.2. FavForMe.....	17
7.1.3. Helpy.....	17
7.1.4. Obsso.....	18
7.1.5. ¿Tienes Sal?.....	18
7.2. Tabla Resumen y Comparativa del Estudio de Mercado.....	19
7.3. Conclusiones.....	21
8. Metodología y Rigor.....	22
8.1. Métodos de Trabajo.....	22
8.2. Herramientas de Seguimiento.....	22
8.2.1. Bitbucket.....	23
8.2.2. Trello.....	23
8.2.3. Google Drive/Google Docs.....	24
8.3. Método de Validación.....	24
9. Planificación de Recursos.....	25
9.1. Humanos.....	25
9.2. Hardware.....	25
9.3. Software y Servicios.....	25
9.4. Tabla Resumen de Recursos.....	27
10. Planificación Temporal.....	29
10.1. Fase de Planificación y Especificación de Requisitos.....	29
10.2. Fase de Implementación.....	30
10.3. Fase de Cierre.....	31
10.4. Tabla Resumen de Tareas.....	31
10.5. Diagrama de Gantt.....	34
10.6. Valoración de Alternativas y Plan de Acción.....	35

11. Gestión Económica.....	36
11.1. Costes Directos.....	36
11.1.1. Recursos Humanos.....	36
11.1.2. Recursos <i>Hardware</i>	39
11.1.3. Recursos y Servicios <i>Software</i>	39
11.2. Costes Indirectos.....	41
11.3. Costes de Imprevistos y Contingencias.....	41
11.4. Resumen del Presupuesto Inicial.....	42
11.5. Control de Gestión.....	43
12. Sostenibilidad y Compromiso Social.....	44
12.1. Auto-evaluación de la Competencia de Sostenibilidad.....	44
12.2. Dimensión Ambiental.....	45
12.3. Dimensión Económica.....	46
12.4. Dimensión Social.....	47
13. Especificación de Requisitos.....	48
13.1. Obtención de Requisitos.....	48
13.2. Requisitos Funcionales: Historias de Usuario.....	48
13.2.1. Autentificación y Cuenta.....	48
13.2.2. Perfiles.....	55
13.2.3. Gestión de Anuncios.....	57
13.2.4. Listas de Anuncios.....	61
13.2.5. Valoraciones.....	67
13.2.6. Seguimiento de Actividad.....	69
13.2.7. Conversaciones.....	73
13.2.8. Notificaciones.....	78
13.3. Requisitos No Funcionales.....	82
13.3.1. Diseño y Apariencia.....	82
13.3.2. Usabilidad y Humanidad.....	82
13.3.3. Rendimiento.....	83
13.3.4. Seguridad y Privacidad.....	83
14. Diseño del sistema.....	84
14.1. Visión General.....	84
14.2. Patrones de Arquitectura y Diseño.....	85
14.2.1. Modelo-Vista-Controlador (MVC).....	85
14.2.2. Modelo-Vista-Presentador (MVP).....	86
14.2.3. Singleton.....	87
14.2.4. Adaptador.....	88
14.2.5. Repositorio.....	88
14.3. Diseño de la Interfaz Gráfica.....	89
14.3.1. Paleta de Colores.....	89
14.3.2. Capturas de Pantalla.....	89
14.4. Diseño del Cliente (<i>Frontend</i>).....	95
14.5. Diseño del Servidor (<i>Backend</i>).....	96
14.6. Diseño de la Base de Datos.....	99
15. Tecnologías Utilizadas.....	101
15.1. Entornos de Desarrollo.....	101
15.1.1. Android Studio.....	101
15.1.2. Eclipse.....	101
15.2. Tecnologías para Desarrollo.....	102
15.2.1. Android Software Development Kit.....	102
15.2.2. Retrofit y GSON.....	102

15.2.3. RxJava.....	103
15.2.4. Picasso.....	103
15.2.5. Spring y Spring Boot.....	103
15.2.6. Java Persistence API (JPA) y Hibernate.....	104
15.2.7. API de Cloudinary.....	104
15.2.8. API de Firebase.....	104
15.2.9. API de Google Maps.....	105
15.2.10. Base de Datos PostgreSQL.....	105
15.3. Tecnologías para Pruebas.....	105
15.3.1. JUnit.....	105
15.3.2. Mockito.....	106
15.3.3. Base de Datos H2.....	106
15.4. Lenguajes de Programación.....	106
15.4.1. Kotlin.....	106
15.4.2. Java.....	107
16. Implementación de Funcionalidades.....	108
16.1. Inicio de Sesión, Sesión Continua y Acceso a Recursos.....	108
16.1.1. Planteamiento.....	108
16.1.2. Solución Implementada.....	109
16.2. Recuperación de Contraseña.....	109
16.2.1. Planteamiento.....	109
16.2.2. Solución Implementada.....	110
16.3. Subida y Descarga de Imágenes.....	110
16.3.1. Planteamiento.....	110
16.3.2. Solución Implementada.....	111
16.4. Listas de Anuncios y Usuarios.....	111
16.4.1. Planteamiento.....	111
16.4.2. Solución Implementada.....	112
16.5. Conversación entre Usuarios.....	113
16.5.1. Planteamiento.....	113
16.5.2. Solución Implementada.....	113
16.6. Conversaciones del Usuario.....	114
16.6.1. Planteamiento.....	114
16.6.2. Solución Implementada.....	114
16.7. Notificaciones.....	115
16.7.1. Planteamiento.....	115
16.7.2. Solución Implementada.....	115
17. Pruebas del sistema.....	117
17.1. Pruebas del Cliente (<i>Frontend</i>).....	117
17.1.1. Monkey Testing.....	117
17.1.2. Pruebas con Usuarios.....	117
17.2. Pruebas del Servidor (<i>Backend</i>).....	118
17.2.1. Pruebas del Modelo.....	118
17.2.2. Pruebas del Controlador/Servicio y de Integración.....	119
18. Identificación de Leyes y Regulaciones.....	120
18.1. General Data Protection Regulation (GDPR).....	120
18.2. Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD).....	121
18.3. Impuesto Sobre Sucesiones y Donaciones.....	121
19. Planificación y Presupuesto Final.....	122
19.1. Desviaciones Temporales.....	122
19.2. Tabla Resumen de Horas por Tarea.....	122

19.3. Diagrama de Gantt Final.....	124
19.4. Presupuesto Final.....	125
20. Conclusiones.....	126
20.1. Valoración Personal.....	126
20.2. Trabajo Futuro.....	126
21. Bibliografía.....	127

Índice de Tablas

Tabla 1: Tabla resumen y comparativa del estudio de mercado.....	19
Tabla 2: Resumen de los servicios necesarios para el proyecto.....	27
Tabla 3: Resumen de las tareas a realizar durante el proyecto.....	32
Tabla 4: Costes de los recursos humanos por rol en el proyecto.....	36
Tabla 5: Costes de los recursos humanos por actividad.....	37
Tabla 6: Coste de los recursos <i>hardware</i> en el proyecto.....	39
Tabla 7: Coste de los recursos <i>software</i> y servicios en el proyecto.....	40
Tabla 8: Costes indirectos.....	41
Tabla 9: Costes de imprevistos.....	41
Tabla 10: Costes de contingencia.....	42
Tabla 11: Presupuesto inicial del proyecto.....	42
Tabla 12: Matriz de sostenibilidad.....	44
Tabla 13: Iniciar sesión.....	48
Tabla 14: Cerrar sesión.....	49
Tabla 15: Registrar usuario.....	50
Tabla 16: Recuperar contraseña.....	51
Tabla 17: Cambiar contraseña.....	53
Tabla 18: Eliminar cuenta.....	54
Tabla 19: Ver mi perfil.....	55
Tabla 20: Editar mi perfil.....	55
Tabla 21: Ver perfil de un usuario.....	56
Tabla 22: Crear anuncio.....	57
Tabla 23: Editar anuncio.....	58
Tabla 24: Cerrar anuncio.....	59
Tabla 25: Guardar anuncio.....	59
Tabla 26: Eliminar anuncio guardado.....	60
Tabla 27: Ver lista de anuncios donde se pide ayuda.....	61
Tabla 28: Ver lista de anuncios de se ofrece ayuda.....	61
Tabla 29: Ver lista de anuncios donde pido ayuda.....	62
Tabla 30: Ver lista anuncios donde ofrezco mi ayuda.....	63
Tabla 31: Ver lista de anuncios de personas que sigo.....	64
Tabla 32: Ver lista de anuncios guardados.....	65
Tabla 33: Buscar anuncios.....	66
Tabla 34: Valorar a un usuario.....	67
Tabla 35: Ver valoraciones dirigidas a mi.....	67
Tabla 36: Ver valoraciones dirigidas a un usuario.....	68
Tabla 37: Seguir a un usuario.....	69
Tabla 38: Dejar de seguir a un usuario.....	70
Tabla 39: Ver los usuarios que me siguen.....	70
Tabla 40: Ver los usuarios que sigo.....	71
Tabla 41: Buscar a un usuario que sigo.....	72
Tabla 42: Buscar a un usuario que me sigue.....	73
Tabla 43: Abrir conversación.....	73
Tabla 44: Enviar mensajes.....	74
Tabla 45: Cerrar conversación.....	75
Tabla 46: Ver mis conversaciones.....	76
Tabla 47: Ver conversación.....	76
Tabla 48: Buscar conversación.....	77

Tabla 49: Recibir notificación de anuncio cercano.....	78
Tabla 50: Recibir notificación de valoración.....	78
Tabla 51: Recibir notificación de nuevo mensaje de conversación.....	79
Tabla 52: Recibir notificación de nuevo seguidor.....	79
Tabla 53: Bloquear notificación de anuncio cercano.....	80
Tabla 54: Bloquear notificación de valoración.....	81
Tabla 55: Bloquear notificación de nuevo mensaje de conversación.....	81
Tabla 56: Bloquear notificación de nuevo seguidor.....	81
Tabla 57: Paleta principal de colores.....	89
Tabla 58: Tabla resumen de horas por tarea.....	123
Tabla 59: Tabla presupuesto final.....	125

Índice de Figuras

Figura 1: Diagrama de Gantt.....	34
Figura 2: Visión general del sistema.....	84
Figura 3: Patrón Modelo-Vista-Controlador.....	86
Figura 4: Patrón Modelo-Vista-Presentador.....	87
Figura 5: Patrón Singleton.....	88
Figura 6: Patrón adaptador.....	88
Figura 7: Patrón repositorio.....	89
Figura 8: Pantalla de carga.....	90
Figura 9: Pantalla de inicio de sesión.....	90
Figura 10: Pantalla de registro.....	90
Figura 11: Pantalla de recuperación de contraseña (I).....	90
Figura 12: Pantalla de recuperación de contraseña (II).....	90
Figura 13: Menú desplegable.....	90
Figura 14: Pantalla de perfil del usuario.....	91
Figura 15: Pantalla de información personal.....	91
Figura 16: Pantalla de valoraciones recibidas.....	91
Figura 17: Pantalla de editar perfil.....	91
Figura 18: Pantalla de seguidores y gente que sigue al usuario.....	91
Figura 19: Pantalla de perfil de otro usuario.....	91
Figura 20: Pantalla de inicio.....	92
Figura 21: Pantalla de mis anuncios.....	92
Figura 22: Pantalla de búsqueda de anuncios.....	92
Figura 23: Pantalla de crear anuncio.....	92
Figura 24: Pantalla de selección de localización.....	92
Figura 25: Pantalla de anuncio de otro usuario donde se pide o busca ayuda.....	92
Figura 26: Pantalla de anuncio del usuario donde pide o busca ayuda.....	93
Figura 27: Pantalla de anuncio de otro usuario donde se ofrece ayuda.....	93
Figura 28: Pantalla de anuncio del usuario donde se ofrece ayuda.....	93
Figura 29: Pantalla de solución de anuncio.....	93
Figura 30: Pantalla de valoración de un usuario.....	93
Figura 31: Pantalla de conversaciones.....	93
Figura 32: Pantalla de conversación.....	94
Figura 33: Pantalla de cambiar contraseña.....	94
Figura 34: Pantalla de eliminar cuenta.....	94
Figura 35: Pantalla de ajustes de notificaciones.....	94
Figura 36: Pantalla de cerrar sesión.....	94
Figura 37: Diagrama de componentes del <i>frontend</i>	95
Figura 38: Diagrama de componentes del <i>frontend</i> de inicio de sesión.....	96
Figura 39: Diagrama de componentes del <i>backend</i>	97
Figura 40: Diagrama de componentes del <i>backend</i> de inicio de sesión.....	98
Figura 41: Diagrama UML de la base de datos.....	99
Figura 42: Diagrama de Gantt final.....	124

1. Introducción

Este proyecto es un Trabajo Final de Grado centrado en la especialidad de Ingeniería del Software impartida en la Facultad Informática de Barcelona (FIB) de la Universitat Politècnica de Barcelona (UPC). El proyecto tiene como directora a María José Casañ Guerrero y de autor a Aarón Jiménez García y es de modalidad tipo A, es decir, ha sido propuesto por iniciativa del alumno. En dicho proyecto se ponen en práctica los conocimientos estudiados a lo largo del grado, especialmente los derivados a la especialidad de Ingeniería del Software.

Este documento pretende ser la memoria del proyecto en cuestión. A lo largo del documento se describen y explican diferentes apartados que han sido considerados de importancia para el desarrollo del trabajo y para el conocimiento del lector.

El propósito de este proyecto es la creación de una aplicación móvil para dispositivos Android que fomente y facilite la colaboración entre diferentes personas que requieran ayuda y que estén cerca entre sí. Dicha aplicación parte de la interacción entre los diferentes usuarios que los pone en contacto para poder solucionar los problemas que describen a través de anuncios, llegando así a un acuerdo entre las dos partes de cómo lograrlo. Su creación se lleva a cabo mediante la utilización de las herramientas más oportunas para facilitar el mantenimiento y la mejora. Además, intenta ser atractiva para los usuarios de forma que, la aplicación pueda llegar a la mayor cantidad de usuarios posibles.

2. Contextualización

Como la mayoría del mundo sabe, una aplicación móvil es un *software* diseñado para ejecutarse en dispositivos móviles como *smartphones* o tabletas. Un *smartphone* [1] no es más que un dispositivo similar a un ordenador, capaz de ampliar sus funcionalidades mediante la instalación de aplicaciones y hacer uso de Internet. Actualmente todos utilizamos estos dispositivos en nuestro día a día junto con todas esas aplicaciones que vienen incorporadas o nos descargamos desde las tiendas en línea. No obstante, las aplicaciones no son un invento reciente.

En 1994, la compañía IBM lanzó al mercado Simon, que fue el primer teléfono en tener aplicaciones como una lista de contactos o calculadora. Posteriormente le siguió Nokia con el mítico juego Snake y en 2002 apareció Blackberry con la incorporación de servicio de correo electrónico. En junio de 2007, Apple dio el salto hacia la tecnología móvil y anunció el Iphone [2]. Este teléfono incluyó aplicaciones donde podías ver mapas, el tiempo o fotografías. Con el anuncio del Iphone 3G en 2008 se anunció la posibilidad de que terceros pudieran diseñar sus propias aplicaciones y publicarlas en línea, dando así lugar a las tiendas que disponemos en nuestros dispositivos hoy en día. A partir de ahí, Google decidió tomar partida y comprar Android y así entrar al mercado contra Apple.

Desde ese momento, cada día surgen tanto en la plataforma de Apple como en la de Google miles de aplicaciones con multitud de usos. Sin duda es un hecho que las aplicaciones llegaron para quedarse y son el futuro de los teléfonos móviles por su facilidad de uso y de creación por parte de los desarrolladores.

3. Formulación del Problema

Antes de comentar otros temas relacionados con el proyecto es importante empezar describiendo aquello que conduce a la realización de este y los objetivos que tiene. En este apartado se describe exactamente esto, la motivación que lleva al autor a la creación de este proyecto y los objetivos principales, de forma breve, que se quieren conseguir mediante la realización del proyecto.

3.1. Motivación

La tecnología con el paso de los años se ha ido expandiendo por diversas áreas de la vida humana, ocupando un papel principal en ella. Esta gran expansión se debe, sin duda alguna, a dos principales factores: la disponibilidad de Internet en cualquier lugar y al gran auge de los teléfonos inteligentes. Los ámbitos de la colaboración, la ayuda y la forma de compartir han sido de los más afectados y los que más pueden evolucionar de entre todos.

Según un informe de We Are Social junto con Hootsuite [3], se calcula que en lo que llevamos de 2019 el número de usuarios de teléfonos móviles ha ascendido a 5 mil millones, lo que supone un 66% de la población mundial. En cuanto a los usuarios con acceso a Internet han llegado a ser 4.3 mil millones que es un 57% respecto la población mundial y específicamente, un 52% del mundo lo consume a través del teléfono móvil.

Estos datos suponen una gran cantidad de la población mundial. Si lo reducimos a un ámbito más cercano, por ejemplo una comunidad o una ciudad, podría ser interesante y una buena oportunidad de fomentar y facilitar la ayuda y colaboración entre personas en cualquier momento y lugar a través de una aplicación.

3.2. Objetivos

Los objetivos de este proyecto se pueden diferenciar en dos vertientes distintas: la tecnológica y la temática. La vertiente tecnológica se centra en los objetivos principales en cuanto a la parte de programación de la aplicación mientras que la temática se centra en el tema de la aplicación y lo que conseguirán los usuarios al utilizarla.

3.2.1. Objetivos Tecnológicos

- Crear una arquitectura lo más robusta posible mediante la aplicación de algún patrón de arquitectura que permitirá tener separación de responsabilidades por capas permitiendo una comprobación mediante pruebas sencilla.
- Diseñar una interfaz clara, sencilla y agradable para los usuarios siguiendo los conceptos de Material Design proporcionados por Google.
- La aplicación debe de estar disponible en al menos tres idiomas: inglés, catalán y castellano
- Crear una API Rest que permita la obtención de los datos de sus usuarios y otros para el correcto funcionamiento de la aplicación. Estos datos se deben de proteger de la mejor manera posible.

3.2.2. Objetivos Temáticos

1. Crear una aplicación que facilite a sus usuarios obtener ayuda en su día a día a través de otros usuarios que la puedan otorgar.
2. Crear una aplicación que facilite a sus usuarios ofrecer su ayuda o sus servicios a otros usuarios que los requieran.
3. Fomentar la colaboración y facilitar la ayuda entre personas poniendo en contacto a usuarios de una comunidad o ciudad o que simplemente estén cerca entre sí.
4. Incrementar la socialización entre los usuarios que utilicen la aplicación.
5. Crear una comunidad de personas que se ayuden entre sí de forma satisfactoria.

4. Alcance del Proyecto

Para poder dar una idea de aquello que se pretende que la aplicación pueda realizar una vez finalizada al lector es necesario describir el alcance del proyecto. En este apartado se ofrece una breve visión de las funcionalidades que debe y no debe tener la aplicación que se pretende desarrollar en este proyecto.

- Se puede y se debe registrar el usuario en la aplicación para su uso donde se pedirán datos como un nombre de usuario, el correo electrónico y una contraseña.
- Los usuarios pueden ingresar en la aplicación una vez registrados con sus credenciales y salir de ella cuando quieran.
- Se puede recuperar la contraseña si se ha olvidado mediante el envío, por parte del servidor, un correo electrónico con un código que el usuario debe ingresar en la aplicación.
- Los usuarios pueden publicar anuncios pidiendo ayuda junto con una serie de parámetros para personalizarlos. Algunos de los parámetros son el tipo de ayuda que se necesita, descripción del problema, localización, título y si se ofrece algo a cambio.
- Los usuarios pueden ofrecer su ayuda mediante anuncios de forma similar al punto anterior. Algunos parámetros son el tipo de ayuda que se ofrece, título, el rango de localización mediante GPS de disponibilidad y si se pide algo a cambio.
- Se puede buscar anuncios a partir de un filtro de búsquedas en la aplicación por coincidencia de nombre, tipo de anuncio, cercanía y fecha de publicación.
- Los usuarios pueden ponerse en contacto con otros usuarios mediante una función de chat implementada en la aplicación para acordar los detalles.
- Los usuarios pueden seguir la actividad de otros usuarios.
- La aplicación, a través de la localización GPS, puede recomendar anuncios que estén cerca de los usuarios.
- La aplicación debe guardar valoraciones de los usuarios que ofrecen ayuda a medida que vayan actuando con la aplicación. Los usuarios deben valorar después de cada ayuda a los usuarios que le ayudaron.
- Cada usuario tiene un perfil que puede personalizar mínimamente como una fotografía de perfil o datos básicos. También puede consultar aquellos usuarios que siguen. En este perfil también se encuentran las valoraciones obtenidas junto con puntuaciones.
- Los usuarios pueden guardar anuncios que son visibles mientras no sean cerrados por el usuario propietario.
- Se puede consultar el perfil de otros usuarios donde se puede ver datos como por ejemplo las valoraciones que ha obtenido, información pública suya o sus seguidores.
- La aplicación debe de poder enviar notificaciones de anuncios creados cerca de usuarios, notificaciones de conversaciones, de nuevos seguimientos de usuarios y de nuevas valoraciones.
- La aplicación debe de estar disponible en tres idiomas desde un inicio, castellano, catalán e inglés.
- La aplicación debe de ser compatible desde la versión 21 hasta la 28 con diferentes tipos de pantallas pero no con tabletas, relojes inteligentes ni televisiones inteligentes..

5. Obstáculos

Para la realización correcta del proyecto hay que anticiparse a los problemas u obstáculos que pueden surgir durante su realización. Algunos de los obstáculos que son susceptibles de encontrar durante la realización del proyecto son los descritos a continuación.

5.1. Tiempo de Realización

Debido a que el proyecto se desarrolla en un espacio de tiempo corto, se debe de realizar una correcta y exhaustiva preparación y planificación. Esto implica estar completamente compenetrado con la directora para la obtención de *feedback* y de este modo modificar convenientemente el proyecto. No obstante, esto a veces no es posible de conseguir y por lo tanto hay que prestar especial atención a que el proyecto esté en las mejores condiciones para la fecha de entrega.

5.2. Acceso a Internet

La aplicación que se propone realizar tiene un requisito indispensable y es que el usuario tenga acceso a Internet a través del dispositivo móvil. Si el usuario no tiene acceso, no podrá acceder a las funcionalidades y por lo tanto la aplicación se volverá inservible. A pesar de que en la época que estamos Internet forma parte de nuestro día a día, es importante tener en cuenta este punto.

5.3. Aparición/Cambios de Funcionalidades

Como se ha comentado en el primer punto de este apartado, el tiempo es corto y se ha de tener especial cuidado con los cambios en las funcionalidades y la aparición de nuevas. Estos sucesos pueden producir que la planificación tenga que ajustarse, de manera que puede perjudicar a la entrega final. Es conveniente que los cambios sean estudiados para ver si es viable el poder aplicarlos o quedarse con la primera idea o ajustarla mínimamente.

5.4. Experiencia en Programación Android

Aunque se dispone de una experiencia previa en la programación para dispositivos Android, este campo es muy amplio y está en continua evolución. Es muy probable que funciones que se hacían de una forma ahora se hagan de otra. Debido a esto, hay que tener especial atención a que el estudio de los nuevos métodos no repercuta de forma drástica al avance del proyecto. Si así fuese, se deben de buscar alternativas y replantear el problema.

6. Actores Implicados

Para plantear de forma correcta este proyecto, o cualquier otro, es fundamental determinar y tener en cuenta todos los actores que están implicados en él ya sea de forma directa o indirecta. En este apartado se describe las personas u organizaciones que tienen o pueden llegar a tener un interés o se pueden ver afectados por el proyecto.

6.1. Directora del Proyecto

Maria Jose Casañ Guerrero es la encargada de guiar y supervisar el proyecto a lo largo de su desarrollo y de esta forma ayudar al autor a que lo complete de la mejor forma posible.

El interés de la directora es principalmente pedagógico y pretende que el proyecto se realice correctamente respondiendo dudas que pueda tener el autor o ayudando en otros aspectos más técnicos.

6.2. Equipo del Proyecto, Desarrollo y Pruebas

El autor Aarón Jiménez García es, en este caso, el único miembro del equipo. Se encarga de documentar la memoria y de llevar a cabo todo el desarrollo tecnológico de la aplicación. Por último, debe de realizar las pruebas pertinentes para la comprobación de la correcta función de la aplicación.

El interés del autor va conducido por el uso de los conocimientos que han sido otorgados a lo largo del grado universitario y el poder obtener nuevos durante la realización del proyecto. Todos estos conocimientos le sirven, en especial, para profundizar en la forma de crear aplicaciones para dispositivos móviles Android correctamente.

6.3. Equipo de Mantenimiento del *Software*

Estos actores corresponden al equipo que se encarga del mantenimiento, el correcto uso y actualización de la aplicación a lo largo de su vida una vez en el mercado. Se ha decidido introducirlo ya que cuando una aplicación u otro producto *software* similar es desarrollado completamente, la mayor parte del equipo original de desarrollo destina sus esfuerzos a otro proyecto. De esta forma, quedan algunos miembros que se encargan de seguir dando soporte a la aplicación o otro equipo de desarrollo se ocupa de ello.

El equipo de mantenimiento del *software* puede verse afectado dependiendo del estado final del *software*. Si este se ha desarrollado utilizando patrones de diseño, una arquitectura de diseño, anotaciones donde sean necesarias y un buen código, el equipo no tendrá dificultad en modificar el *software* para mejorarlo y mantenerlo. De forma contraria, encontrarán muchas dificultades.

6.4. Usuarios Estándar

Los usuarios estándar son los usuarios que no son empresas, trabajadores autónomos o organizaciones sin ánimo de lucro. Ellos son los principales destinatarios del sistema que se quiere desarrollar en este proyecto.

Estos actores tienen interés en las funcionalidades principales de la aplicación ya sea para poder obtener u ofrecer ayuda a través de ella. Debido a que son los actores que van a tener contacto final con el producto, es necesario tener en cuenta conceptos de diseño como tener una interfaz sencilla, clara y atractiva. Una mala interfaz puede conducir a que no se sientan atraídos a utilizar la aplicación y utilizar otros recursos. También es importante otorgarles funcionalidades útiles y que sean de interés para ellos.

6.5. Organizaciones Sin Ánimo de Lucro

Las organizaciones sin ánimo de lucro son aquellas entidades que no buscan un fin lucrativo sino que persiguen una finalidad humanitaria. Cabe destacar estos usuarios ya que las principales funcionalidades de la aplicación van acorde con sus necesidades.

Estos actores tienen interés ya que podrían hacer uso de la aplicación para pedir u ofrecer ayuda y facilitar así su labor.

6.6. Pequeños Comercios

Los pequeños comercios son aquellos que disponen de un negocio propio. A pesar de que la aplicación que se quiere desarrollar en este proyecto se centra más en usuarios estándar, no se descarta el uso de la aplicación por parte de estos comercios.

Estos actores como un usuario más, sienten atracción en las funcionalidades principales que presenta la aplicación que les pueda beneficiar, en este caso, poder ofrecer sus servicios como ayuda a otros usuarios.

7. Estado del Arte

Para poder realizar una aplicación que atraiga a usuarios a utilizarla se requiere antes realizar un estudio de aquellas soluciones disponibles ya en el mercado para poder así diferenciarse y otorgar nuevas características o mejorar aquellas que ya existan. En este apartado se muestra un estudio de aplicaciones con funcionalidades o objetivos similares a los comentados en el apartado de Alcance del Proyecto de este documento. Después, se realiza una tabla resumen y comparativa con las distintas funcionalidades de las aplicaciones mostradas junto con la propuesta y se argumenta el por qué de realizar esta aplicación que se propone.

7.1. Estudio de Mercado

En el mercado actual existen múltiples aplicaciones que favorecen el consumo colaborativo. Estas aplicaciones se centran habitualmente en un tema específico como por ejemplo compartir un coche, comprar y vender ropa de segunda mano, compartir apartamento, etc. Esto dificulta encontrar aplicaciones más genéricas como la que se propone realizar que no se centra en un tema específico. En este apartado se muestra una serie de aplicaciones que existen hoy en día y tienen funcionalidades similares a la de la aplicación propuesta. Se ha decidido por la falta de aplicaciones actuales similares escoger aplicaciones que aún no han salido al mercado y aplicaciones que estuvieron disponibles pero ya no o al menos, no se ha podido encontrar una fuente fiable para su descarga.

7.1.1. Impossible People

Impossible People [4] nació en 2013 como una comunidad web para que las personas pudieran ayudarse de manera altruista. En 2016 la aplicación fue renovada y se pasó a las aplicaciones móviles para dispositivos iOS y Android donde estaba disponible en inglés. A pesar de esto, no se ha podido obtener la aplicación por ningún servicio a día de hoy pero su página web sigue en pie con lo que se presupone que el proyecto sigue activo. La aplicación es de código abierto de manera que todo el mundo puede participar para su mejora y se puede encontrar en la página web oficial.

Esta aplicación permite crear anuncios sin condiciones previas para pedir ayuda a los demás usuarios indicando la localización GPS pero no permite anunciar servicios. Estos anuncios se pueden buscar a partir de *hashtags* similares a los que utiliza la red social Twitter y de esta forma encontrar anuncios más concretos a lo que el usuario quiere.

En cuanto a la comunicación entre usuarios implementa un chat en la aplicación que permite concretar detalles. También contiene un mecanismo de amistad similar al de Twitter nuevamente. Este se basa en seguir la actividad de otros usuarios y de esta forma no perder detalle de los anuncios que los usuarios crean.

7.1.2. FavForMe

Los creadores Matías González y Ryan Malone de FavForMe [5] se describen como una comunidad que se centra en la realización de favores para poder crear lazos con otras personas. Esta aplicación está disponible para sistemas Android e iOS y se ha podido comprobar que está disponible en español y en inglés.

Esta aplicación permite la creación de anuncios de ayuda a cambio de monedas virtuales de la propia aplicación. Cuando un usuario se registra por primera vez recibe una serie de monedas que sirven justamente para esto. Sin monedas no es posible publicar ningún anuncio y tan solo hay dos formas de conseguirlas. La primera de ellas es ayudando a otros a través de los anuncios que los usuarios han publicado concretando todo a través de su chat incorporado. Una vez finalizada la ayuda, los dos usuarios concretan que se ha realizado y se le otorgan monedas al usuario que ofreció su ayuda. La segunda es a través de logros disponibles en la aplicación. Estos logros no son infinitos y se consiguen al realizar ciertos objetivos en la aplicación. Una vez conseguido un logro se canjea por las monedas y ya no se puede volver a obtener ninguna recompensa.

El uso de monedas es un incentivo para motivar a sus usuarios a realizar las ayudas pero no es la única motivación. La aplicación dispone de una tabla clasificatoria donde los usuarios van subiendo o bajando escalones dependiendo de si finalizan la ayuda o no. Esto a su vez permite a sus usuarios ver un nivel de valoración de los usuarios a pesar de que no es realmente efectivo pues un usuario puede llegar al puesto más alto y no ofrecer la mejor calidad de ayuda.

Por último, el uso de la tecnología GPS es muy simple y no profundiza mucho en ella. Su uso se centra en tan solo determinar el punto de localización donde se necesita la ayuda.

7.1.3. Helpy

Helpy [6] es una aplicación que según su página web no está disponible pero lo estará próximamente para dispositivos móviles, sin concretar qué plataformas. Debido a que no está disponible aún no se puede obtener mucha información de ella, tan solo la propia que ofrecen en su página web. La aplicación está siendo desarrollada en cooperación con un grupo de desarrolladores franceses y españoles lo que implica que estará disponible en francés y español.

Según comentan los creadores en su página web, la aplicación permite la creación de anuncios para pedir ayuda, sin condiciones previas, en tareas cotidianas como limpiar la casa u otras como ayudar en una mudanza. También posibilita a los usuarios a ofrecer su ayuda o servicios relacionados con los mismos temas. Una vez que los usuarios deciden responder a un anuncio o a contactar con un servicio se ponen en contacto a través de un chat propio donde pueden acordar los detalles.

La organización de la aplicación es precisa, agrupando los anuncios por tipos permitiendo así realizar filtros de búsqueda más exactos a lo que el usuario quiere buscar. También se hace referencia a que la aplicación dispone de un servicio de evaluación que permite saber la fiabilidad del usuario con el que se contacta, no obstante, no se puede saber aún como funcionará debido a su estado de desarrollo.

En cuanto al uso de la tecnología GPS, al contrario que FavForMe si que se añade un poco más de utilidad. A parte de indicar el punto de localización del anuncio también permite notificar a aquellos usuarios que estén cercanos de la publicación de los anuncios.

7.1.4. Obsso

Obsso [7] es una aplicación disponible para dispositivos Android y iOS que se aleja del gasto de dinero y se centra en el trueque, promoviendo la colaboración y la ayuda de sus usuarios.

Los usuarios pueden poner anuncios con objetos que ya no necesitan o no quieren para que otro usuario lo vea y se pueda realizar un trueque. También se posibilita la opción de ofrecer servicios en vez de objetos. Estos anuncios hacen uso de la ubicación del teléfono móvil para determinar donde se producirá el trueque. Una vez que los usuarios están interesados en el trueque y lo han discutido a través del chat que incorpora la aplicación quedan en persona y se produce el intercambio.

Al igual que pasa con Impossible People esta aplicación permite seguir a otros usuarios. De esta forma los usuarios pueden estar atentos de los anuncios que publican los usuarios a los que siguen y seguir haciendo trueques.

7.1.5. ¿Tienes Sal?

La aplicación ¿Tienes Sal? [8] es una aplicación fundada por Sonia Alonso y se trata de una nueva red social que trata de unir a los vecinos haciendo que dejen de ser desconocidos entre ellos. Esta fue exportada de una aplicación ya creada en Alemania. Está disponible para dispositivos iOS y Android y de momento solamente está disponible para la ciudad de Barcelona y Madrid pero se pretende expandir a otras ciudades si continua teniendo el éxito que ha conseguido en estas dos ciudades.

Para iniciarse a la aplicación es necesario registrar un vecindario. Para ello son necesarias entre 20 y 30 personas del vecindario que la aplicación confirmará su dirección para verificar que no sean falsos perfiles. Una vez dentro los usuarios pueden publicar eventos que quieran realizar, pedir ayuda o ofrecerla sin requisitos monetarios. Una vez que los usuarios quieren responder un anuncio lo hacen a partir de comentarios que se pueden añadir en la publicación en vez de un chat.

Al igual que pasa con otras aplicaciones comentadas en los apartados anteriores utiliza la localización GPS para determinar el lugar donde se debe producir la ayuda, evento o servicio.

7.2. Tabla Resumen y Comparativa del Estudio de Mercado

	Impossible People	FavForMe	Helpy	Obsso	¿Tienes Sal?	Proyecto
Perfil	Perfil por cada usuario personalizable.	Perfil por cada usuario personalizable.	Perfil por cada usuario personalizable.	Perfil por cada usuario personalizable.	Perfil por cada usuario personalizable.	Perfil por cada usuario personalizable.
Publicación Anuncios	Posibilidad de crear anuncios de ayuda sin requisitos previos. No se posibilita la función de ofrecer ayuda o anunciar un servicio.	Posibilidad de crear anuncios de ayuda mediante el uso de una moneda virtual de la aplicación. No se posibilita la función de ofrecer ayuda o anunciar un servicio.	Posibilidad de crear anuncios de ayuda y ofrecer ayuda o servicios sin requisitos previos.	Posibilidad de crear anuncios de productos o de ofrecimiento de servicios sin requisitos previos.	Posibilidad de crear anuncios de ayuda, anuncios de ofrecimiento de servicios y eventos sin requisitos previos.	Posibilidad de crear anuncios de ayuda y ofrecer ayuda o servicios sin requisitos previos.
Filtro de Anuncios	Utiliza <i>hashtags</i> similar a la red social Twitter. Estos <i>hashtags</i> permiten encontrar anuncios de temas concretos.	Búsqueda por coincidencia de nombre. Cada anuncio tiene un tipo de anuncio asignado. Gracias a esto se puede realizar una búsqueda por tipo de anuncio.	Cada anuncio tiene un tipo de anuncio asignado. Gracias a esto se puede realizar una búsqueda por tipo de anuncio.	Búsqueda por coincidencia de nombre dependiendo de la sección en la que se esté de la aplicación.	Búsqueda por coincidencia de nombre dependiendo de la sección en la que se esté de la aplicación.	Búsqueda por tipo de anuncio. Búsqueda por coincidencia de nombre. Búsqueda por anuncios de los usuarios que se sigue. Búsqueda por anuncios por cercanía. Búsqueda por fecha de creación.
Mecanismo de Amistad	Mecanismo similar al de Twitter. Los usuarios pueden seguirse entre si teniendo la posibilidad de ver la actividad de aquellos usuarios que siguen.	No se ha encontrado ninguna información que permita saber con certeza si contiene esta funcionalidad.	No se ha encontrado ninguna información que permita saber con certeza si contiene esta funcionalidad.	Mecanismo similar al de Twitter. Los usuarios pueden seguirse entre si teniendo la posibilidad de ver la actividad de aquellos usuarios que siguen.	No utiliza ningún mecanismo. Esta aplicación está restringida al barrio del usuario por lo que no es necesaria.	Mecanismo similar al de Twitter. Los usuarios pueden seguirse entre si teniendo la posibilidad de ver la actividad de aquellos usuarios que siguen.

Comunicación Chat	Se implementa una función de chat para poder concretar los detalles de la ayuda.	Se implementa una función de chat para poder concretar los detalles de la ayuda.	Se implementa una función de chat para poder concretar los detalles de la ayuda.	Se implementa una función de chat para poder concretar los detalles de la ayuda.	No se implementa una función de chat. Funciona a través de comentarios en las publicaciones.	Se implementa una función de chat para poder concretar los detalles de la ayuda.
Localización GPS	Determina la localización de la ayuda.	Determina la localización de la ayuda.	Determina la localización de la ayuda y notifica a usuarios cercanos de la publicación de los anuncios.	Se utiliza para poder determinar la localización del trueque y notificar productos y servicios cercanos.	Se utiliza para poder determinar la localización de la ayuda, evento o servicio.	Determina la localización de la ayuda y notifica a usuarios cercanos de la publicación de los anuncios.
Recomendaciones	No se ha encontrado ninguna información que permita saber con certeza si contiene esta funcionalidad.	No se ha encontrado ninguna información que permita saber con certeza si contiene esta funcionalidad.	No se ha encontrado ninguna información que permita saber con certeza si contiene esta funcionalidad.	Se recomiendan productos y servicios similares a los buscados.	No se ha encontrado ninguna información que permita saber con certeza si contiene esta funcionalidad.	Por el momento esta funcionalidad no se implementa por falta de tiempo.
Valoraciones	No se ha encontrado ninguna información que permita saber con certeza si contiene esta funcionalidad.	<i>Ranking</i> donde los usuarios con mejor valoración están en posiciones más altas. Se genera a partir de las valoraciones de los usuarios.	Sistema de evaluación que permite saber a sus usuarios el nivel de fiabilidad del usuario con el que contactan.	No se ha encontrado ninguna información que permita saber con certeza si contiene esta funcionalidad.	No se ha encontrado ninguna información que permita saber con certeza si contiene esta funcionalidad.	Valoraciones de las ayudas que serán añadidas a los perfiles de cada usuario. Los usuarios pueden verlas para decidir si optar por su ayuda o su servicio.
Multilingue	Disponible en inglés.	Disponible en inglés y español.	Disponible en francés y español.	Disponible en portugués, inglés, francés, italiano y español.	Disponible en español.	Disponible en español, catalán e inglés.

Tabla 1: Tabla resumen y comparativa del estudio de mercado

7.3. Conclusiones

Como se ha comentado anteriormente, actualmente hay pocas aplicaciones similares a lo que se quiere realizar en el proyecto pues muchas han dejado de estar accesibles a través de las tiendas oficiales o de sus páginas web. Además, solo algunas de las que se han escogido son accesibles mientras que las otras o no han salido al mercado aún y no tienen fecha prevista o su futuro es confuso.

Todas las aplicaciones que se han mostrado proponen un consumo colaborativo similar al que quiere conseguir en este proyecto. Todas presentan funcionalidades parecidas como la creación de anuncios, el poder comunicarse a través de la aplicación ya sea mediante un chat integrado o comentarios en la publicación. No obstante, hay funcionalidades en las cuales no acaban de profundizar demasiado en el momento que se ha hecho este estudio y esto supone que estas soluciones no agilicen el proceso del consumo colaborativo que se propone. Esto provoca que las personas dejen de usar las aplicaciones porque piensan que no funcionan correctamente o están mal llevadas.

Si nos fijamos en el filtro que normalmente usan está basado en búsquedas de coincidencia por nombre o por tipo dejando de lado las publicaciones más antiguas o las más cercanas. Esto provoca que dichos anuncios queden olvidados algo que se pretende evitar en la solución que se propone. También el uso de la localización GPS está muy simplificada y podría llevarse a más permitiendo notificar a usuarios con la aplicación instalada cercanos a la ubicación donde se ha creado un anuncio. En cuanto a las recomendaciones solo una aplicación las realiza. Por último las valoraciones solo las implementa una aplicación. Creando un sistema de valoraciones de usuarios permitiría a la aplicación ser más segura para sus usuarios y poder determinar mejor con quien quieren contactar.

En conclusión, en este proyecto se toma la idea de consumo colaborativo ya creada en otros proyectos pero se intenta llevar a más. Es necesario la creación de una nueva aplicación de cero ya que el mercado actual no dispone de gran cantidad de soluciones pues muchas han desaparecido o todavía no han salido al mercado. En este proyecto se crean las funcionalidades necesarias desde cero mejorando las ya existentes para agilizar la finalidad que se quiere conseguir. La idea es crear una comunidad de personas que con su teléfono móvil puedan ayudar a otras de forma sencilla y rápida favoreciendo así el consumo colaborativo.

8. Metodología y Rigor

Para el correcto desarrollo del proyecto es necesario escoger una metodología consolidada que se ajuste a los requisitos del proyecto. En este apartado se describe la metodología que se sigue durante el proyecto. También, las herramientas de seguimiento que se emplean para realizar un correcto progreso y por último los métodos de validación que permiten saber cuando parte del proyecto está finalizado.

8.1. Métodos de Trabajo

Para este proyecto se utiliza una metodología de desarrollo ágil llamada Scrum. Esta metodología al estar basada en equipos se adapta, en este caso, para un único miembro.

Las metodologías ágiles, como se describe en el artículo *Software Development Life Cycle AGILE vs Traditional Approaches* [9, p. 163], están centradas en un desarrollo iterativo e incremental donde las fases del desarrollo se van visitando una y otra vez. Esto proporciona rapidez y flexibilidad para adaptarse a las necesidades del cliente. Estas son ideales para proyectos que cambian a lo largo del ciclo de vida y permiten rectificar y modificar los requerimientos y las prioridades a medida que avanza el proyecto. Además, el cliente, al estar en contacto directo con el desarrollo del proyecto mejora su experiencia.

En un proyecto Scrum [10] hay una lista de historias de usuario que se determinan en la fase de Inception llamado Backlog. Estas historias de usuario son las funcionalidades que se quieren plasmar en el producto final. Las historias son escogidas al inicio de los Sprint que tienen una duración de entre 2 a 4 semanas y son desarrolladas. Una vez pasado ese tiempo el producto desarrollado es entregado al cliente. También se realizan reuniones al inicio para planificar el Sprint y otras al final para comentar cómo ha ido.

En este proyecto de forma similar, se adapta el mecanismo que emplea Scrum de las diferentes etapas. Al final de cada Sprint, se entrega el producto a una selección de usuarios para que den *feedback* para posibles correcciones. Además, se intenta realizar al menos una reunión con la directora al final de cada Sprint, para ver como se ha progresado y determinar las tareas del siguiente a excepción del primer Sprint, que también tiene al inicio.

8.2. Herramientas de Seguimiento

Para poder tener un control de como se desarrolla el proyecto y saber donde se está en cada momento es necesario utilizar una serie de herramientas que sean compatibles con la metodología escogida. En los siguientes subapartados se describen las herramientas de seguimiento que se utilizan a lo largo del proyecto.

8.2.1. Bitbucket

Bitbucket [11] es un servicio de almacenamiento de código para proyectos utilizado en la industria del *software*. Permite tener control del código de forma segura y tener versiones de seguridad. También facilita el compartir código con los demás, especialmente con equipos. Este servicio tiene la peculiaridad de que permite crear repositorios privados de forma gratuita al contrario que su competencia, principal motivo de su elección.

A la hora de trabajar con este servicio se aplica el flujo de trabajo llamado GitFlow. GitFlow, tal y como explican los creadores de Bitbucket [12], consiste en trabajar mediante el uso de diferentes ramas, cada cual con su propio objetivo. Podemos diferenciar las siguientes:

- **Master:** Es la rama principal del proyecto que contiene al final del proyecto el producto final.
- **Develop:** Es la rama secundaria que surge de Master y que se encarga de ir recogiendo las historias de usuario que han sido finalizadas y testeadas.
- **Feature-[nombre]:** Estas ramas surgen de Develop y hay una para cada historia de usuario donde se desarrollan.

Al final de cada Sprint, todas las historias de usuario (Feature-[nombre]) acabadas se unen a Develop para poder entregar una versión funcional del producto.

8.2.2. Trello

Trello [13] es una herramienta de organización de proyectos mediante el uso de tableros muy sencilla de utilizar. Mediante esta herramienta se gestionan las historias de usuario para tener un perfecto control de lo que se está haciendo, de lo que se debe cambiar y de lo que está hecho. Una historia de usuario se representa con una tarjeta en el tablero como [#][Título] p.ej. “#1 *Como usuario quiero poder registrarme en la aplicación*”. Dentro de esta hay una lista con los requisitos mínimos que tiene que cumplir la historia y otra con los criterios de aceptación que debe cumplir su código.

Esta gestión se realiza mediante los siguientes tableros:

- **Backlog:** Historias de usuario del proyecto.
- **Sprint[#]-ToDo:** Historias de usuario que se deben hacer durante el Sprint.
- **Sprint[#]-Doing:** Historias de usuario que se están haciendo en el Sprint.
- **Sprint[#]-ToTest:** Historias de usuario que se han desarrollado en el Sprint y están pendientes de testear.
- **Done:** Historias de usuario que ya han sido finalizadas y testeadas.

Al finalizar el Sprint, si hay historias de usuario que no se han finalizado vuelven a la lista para poder recuperarlas en otro Sprint.

8.2.3. Google Drive/Google Docs

El gestor de almacenamiento de Google llamado Google Drive y el gestor de documentos Google Docs son las herramientas utilizadas para el almacenamiento y la creación de documentos para el proyecto. De esta forma, la directora puede acceder al contenido de la documentación de forma más sencilla.

8.3. Método de Validación

En cuanto al método de validación, para verificar si la documentación y código del producto es el esperado, se recurre generalmente a la directora del proyecto. Gracias al uso de Scrum, hay un seguimiento exhaustivo del progreso mediante reuniones durante las diferentes etapas del proyecto. Además se cuenta también con la posibilidad de utilizar los medios tecnológicos como correos electrónicos para cualquier duda.

No obstante, entrando más en concreto a hablar sobre las etapas de Sprint, se utiliza como método de validación lo que se denomina *Definition of Done* [14]. Este concepto consiste en tener una serie de requisitos y criterios de aceptación, como se mencionó en el apartado 8.2.2, para las historias de usuario que sirven para determinar cuándo son válidas. Mientras que los criterios de aceptación varían dependiendo de cada historia de usuario, los requisitos no y son los siguientes:

- Se han especificado y cumplido los requisitos funcionales y no funcionales.
- Se ha implementado una interfaz simple e intuitiva para el usuario.
- Se ha implementado el código cumpliendo los criterios de aceptación.
- Se ha corregido y simplificado el código.
- Se ha testeado de forma conveniente la interfaz de la aplicación.
- Se ha testeado de forma conveniente el código mediante pruebas unitarias.
- La directora ha dado su visto bueno.
- Se han realizado las mejoras correspondientes al *feedback* de los usuarios.

Una vez cumplidos los requisitos mínimos y los criterios de aceptación, la historia de usuario es válida y por lo tanto finaliza.

9. Planificación de Recursos

En este apartado se describen los diferentes tipos de recursos que son necesarios para la realización del proyecto. Podemos encontrar al menos tres tipos de recursos necesarios: recursos humanos, materiales y de *software* y servicios. En los siguientes subapartados se describen dichos recursos.

9.1. Humanos

En lo que respecta a recursos humanos, debido a que este proyecto es la realización de un trabajo de fin de grado, se dispone únicamente de una sola persona. El autor del proyecto, Aarón Jiménez García, es el encargado de realizar todas las fases en las que el proyecto se ha dividido y sus correspondientes tareas, desde documentación hasta programación de funcionalidades y testeo.

9.2. Hardware

Al contrario que con los recursos humanos, son necesarios varios recursos materiales. A continuación se describen los recursos materiales necesarios para poder llevar a cabo este proyecto.

- **Ordenador portátil (Asus Serie F555L, Intel Core i7-5500U hasta 3.0GHz).** Es la herramienta indispensable desde donde se realiza todo el proyecto, desde la memoria hasta el desarrollo de funcionalidades y el testeo.
- **Teléfono móvil (Huawei P9 Lite y Xiaomi MI A2).** Estos dos dispositivos móviles se utilizan para comprobar y testear funcionalidades del *frontend* desde dispositivos físicos.
- **Servidor de pruebas (Heroku).** Heroku [15] es una plataforma en la nube de servidores que permite a sus usuarios utilizar sus servidores para publicar aplicaciones. Este servidor se utiliza para comprobar y testear funcionalidades del *backend* que tendrá el servidor definitivo.
- **Servidor del proyecto (DigitalOcean).** DigitalOcean [16] es una plataforma similar a Heroku que permite a sus usuarios utilizar sus servidores para lo que deseen. Este servidor es el servidor definitivo donde se encuentra la parte *backend* definitiva de la aplicación. Se ha decidido escoger el distribuidor DigitalOcean por su relación calidad/precio.

9.3. Software y Servicios

En cuanto a los recursos centrados en *software* y servicios, son necesarias varias herramientas para el desarrollo del proyecto. Estas herramientas se describen de forma breve a continuación y más adelante en el documento se exponen aquellas relacionadas con la implementación de forma más extendida.

- **Gmail.** Este servicio de correo electrónico de Google es uno de los métodos usados para comunicarse con la directora, especialmente para concretar reuniones.
- **Skype.** Este servicio de Microsoft es uno de los métodos usados para comunicarse con la directora. Se utiliza en el caso de que algunos de los dos, tanto directora como el autor no puedan asistir presencialmente a una reunión.

- **Bitbucket.** Como se comentó en el apartado de Herramientas de Seguimiento de la memoria, se utiliza para almacenar y controlar versiones a modo de copia de seguridad del código que se va desarrollando. Además, la directora puede tener acceso para ver el desarrollo del código.
- **Trello.** También se comentó en el apartado de Herramientas de Seguimiento, se utiliza para organizar el estado actual de las historias de usuario del proyecto mediante el uso de tableros. La directora puede tener acceso a los tableros y ver cómo se están desarrollando las historias de usuario.
- **Google drive/docs.** Como se comentó en el apartado de Herramientas de Seguimiento, se utilizan para el almacenamiento y la creación de documentos para la memoria. Al igual que Trello y Bitbucket, la directora tiene acceso para realizar el seguimiento.
- **Adobe Illustrator.** Adobe Illustrator [17] es un *software* de edición de gráficos vectoriales destinado a la creación de imágenes como logotipos, iconos o formas. Se utiliza para la creación de recursos gráficos para la parte de la interfaz de la aplicación.
- **Android Studio.** Como explica la página oficial de desarrolladores Android [18], Android Studio es el entorno de desarrollo oficial para aplicaciones Android más utilizado en el mercado. Se utiliza para la parte de la aplicación Android del proyecto. Aquí se realiza todo el código, tanto funcionalidades como pruebas de la parte de la aplicación móvil.
- **Eclipse.** Eclipse [19] es un entorno de desarrollo integrado similar a Android Studio para proyectos Java principalmente pero mediante añadidos se puede extender a más ámbitos. Es el entorno de desarrollo integrado que se utiliza para la parte del servidor. Aquí se realiza todo el código, tanto funcionalidades como pruebas de la parte del servidor que utilizará la aplicación móvil.
- **Firebase.** Firebase [20] es una plataforma de desarrollo móvil de Google en la nube con diferentes opciones como base de datos, analíticas o mensajería. En este proyecto se utiliza para la implementación de la funcionalidad del chat y notificaciones comentadas en el alcance del proyecto.
- **Postman.** Postman [21] es un *software* que permite realizar peticiones al servidor para poder comprobar que funcione correctamente a través de Internet.
- **GSON.** GSON es una librería que permite analizar y obtener datos de extensión JSON para poder ser utilizados en el código.
- **Retrofit.** Retrofit es una librería que permite realizar llamadas a través de una API Rest de manera muy simple sin tener que realizar demasiado código.
- **RxJava.** Esta librería permite realizar programas asíncronos.
- **Picasso.** La librería Picasso está disponible para Android Studio y permite la descarga de imágenes a tiempo real en las aplicaciones y se encarga de tratar con ellas sin tener que preocuparnos cuando una imagen ya no es necesaria.
- **Spring Boot.** Spring Boot es un *framework* empleado para la creación de API Rest.
- **Hibernate.** Este *framework* permite la creación de bases de datos simples a través de código de manera que no se requiera ir cambiando de herramientas durante la fase de desarrollo.
- **API Cloudinary.** La API de Cloudinary permite la comunicación con el servicio Cloudinary de almacenamiento de imágenes para guardarlas, recuperarlas y hacer otras operaciones proporcionadas por este servicio.

- **API Google Maps.** Esta API permite la obtención de mapas de los servidores de Google de forma gratuita.
- **PostgreSQL.** PostgreSQL es un tipo de base de datos relacional que se pretende utilizar para la fase de producción del proyecto.
- **JUnit.** Este es un *framework* que permite realizar pruebas unitarias y así comprobar las funcionalidades de los programas.
- **Mockito.** Mockito es un *framework* que permite suplantar una parte del código para poder comprobar a través de JUnit el programa sin tener dependencias.
- **H2.** H2 es un tipo de base de datos relacional que normalmente se utiliza para la comprobación de programas para evitar la alteración y creación de bases de datos destinadas a la producción.

9.4. Tabla Resumen de Recursos

En la siguiente tabla se puede encontrar un resumen de los recursos comentados en el anterior apartado. En la Tabla 2 se muestra el tipo de recurso, el tipo y la finalidad de este en el proyecto.

#Recurso	Recurso	Tipo	Finalidad
R1	Autor	Humano	Desarrollar proyecto.
R2	Ordenador portátil (Asus Serie F555L, Intel Core i7-5500U hasta 3.0GHz)	Material	Desarrollar documentación, funcionalidades y pruebas.
R3	Teléfono móvil (Huawei P9 Lite y Xiaomi MI A2)	Material	Comprobar funcionalidades de la aplicación Android.
R4	Servidor de pruebas (Heroku)	Material	Contener el código del servidor de la fase de desarrollo.
R5	Servidor del proyecto (DigitalOcean)	Material	Contener el código del servidor de la fase de producción.
R6	Lugar de trabajo	Material	Donde se realiza el proyecto.
R7	Gmail	Software y servicios	Comunicación con la directora.
R8	Skype	Software y servicios	Comunicación con la directora.
R9	Bitbucket	Software y servicios	Almacenar y controlar versiones del código.
R10	Trello	Software y servicios	Organizar el estado de las historias de usuario.
R11	Google drive/docs	Software y servicios	Almacenamiento y creación de la memoria.
R12	Adobe Illustrator	Software y servicios	Creación de recursos gráficos.
R13	Android Studio	Software y servicios	Crear aplicación Android.
R14	Eclipse	Software y servicios	Crear código del servidor.
R15	API Firebase	Software y servicios	Envío de mensajes y notificaciones a usuarios.
R16	Postman	Software y servicios	Probar peticiones del servidor.
R17	GSON	Software y servicios	Analizar y obtener datos JSON en

			la aplicación Android.
R18	Retrofit	Software y servicios	Realizar peticiones al servidor.
R19	Rxjava	Software y servicios	Crear código asíncrono.
R20	Picasso	Software y servicios	Descargar y tratar imágenes en Android.
R21	Spring Boot	Software y servicios	Crear Rest API.
R22	Hibernate	Software y servicios	Crear bases de datos a partir de código.
R23	API Cloudinary	Software y servicios	Envío de imágenes al servicio Cloudinary.
R24	API Google Maps	Software y servicios	Obtención de mapas de los servidores de Google
R25	PostgreSQL	Software y servicios	Base de datos de producción.
R26	JUnit	Software y servicios	Realizar pruebas unitarias.
R27	Mokito	Software y servicios	Suplantar código y realizar pruebas.
R28	H2	Software y servicios	Base de datos de la fase de desarrollo y pruebas.

Tabla 2: Resumen de los servicios necesarios para el proyecto.

10. Planificación Temporal

La planificación del proyecto se ha realizado teniendo en cuenta que tiene como fecha de inicio el 18 de Febrero de 2019 momento en que se empieza a impartir la asignatura de GEP, que forma parte del trabajo de fin de grado. Como fecha final para el desarrollo de la aplicación y de la documentación, se ha considerado el 19 de Junio de 2019, una semana antes de la fecha de lectura determinada por la universidad. No obstante, el trabajo de fin de grado finaliza el 28 de Junio de 2019 ya que entre el 1 y el 5 de Julio son las defensas de los proyectos.

En cuanto a las horas necesarias para llevar a cabo el proyecto, se ha tenido en cuenta que el trabajo de fin de grado consta de 18 créditos. A 30 horas cada crédito hace un total de 540 horas de trabajo distribuidas en tres fases diferenciadas que se comentan en los siguientes apartados. Durante la primera fase se realizan aproximadamente 5 horas diarias de trabajo en días laborables mientras que en la segunda 6,5 y en la tercera 5 horas. Por lo tanto, el proyecto debería de constar entre 540 y 600 horas teniendo en cuenta que se trabaja durante días laborables. A pesar de esto, puede ocurrir que en algún momento de alguna etapa del proyecto se produzca un aumento de horas debido a diferentes factores como la aparición de algún obstáculo de los que se mencionaron en puntos anteriores o por motivos personales del autor.

10.1. Fase de Planificación y Especificación de Requisitos

La fase de planificación y diseño se inicia el día 18 de Febrero de 2019 con la primera reunión con la directora y finaliza el día 29 de Marzo de 2019 con la defensa de la fase inicial del proyecto y la segunda reunión. Durante esta primera fase se realiza la planificación del proyecto a través de la asignatura GEP y la especificación de los requisitos de la aplicación de forma paralela. Se trabaja durante aproximadamente 5 horas al día y tiene una duración de 30 días laborables.

Empezando por la planificación, se desarrolla una serie de documentos necesarios para la inicialización y realización del proyecto y se estudia las herramientas necesarias para llevarlo a cabo. Estos documentos se entregan en las fechas previstas por la universidad y forman parte de la documentación final del proyecto. Por último, se prepara y se realiza la defensa oral de la primera fase del trabajo de fin de grado. A continuación se muestran las tareas relacionadas con la planificación ordenadas por orden de ejecución debido a las dependencias que existen entre ellas:

- **Análisis del contexto y alcance.** Esta tarea se centra en la contextualización del proyecto determinando sus objetivos, alcance y metodología de uso.
- **Planificación temporal y recursos.** Consiste en la creación de una planificación temporal del proyecto y de los recursos que se utilizarán.
- **Presupuesto y sostenibilidad.** Esta tarea determina el presupuesto inicial requerido para llevar a cabo el proyecto.
- **Preparación de la presentación y documento final.** Consiste en la creación de una presentación PowerPoint para la defensa de la fase inicial y la entrega del documento final acabado y revisado.

- **Defensa de la fase inicial.** La tarea tiene como objetivo la realización de la defensa de la fase inicial del proyecto de fin de grado.

En lo que respecta a la parte del estudio de requisitos, consiste en realizar gran parte del Backlog para poder seguir con la fase de implementación, que como se comentó en el apartado de metodología son historias de usuario. También se diseñan los diagramas UML de la base de datos, el *backend* y el *frontend* y los diagramas de secuencia que permiten entender el flujo de la aplicación. Al finalizar el estudio se llega al final de la fase y se realiza una segunda reunión con la directora para ver el progreso, corregir aquello que esté mal y dar comienzo a la fase de implementación. A continuación se muestran las tareas ordenadas por orden de ejecución debido a las dependencias que existen entre ellas:

- **Reunión #1.** Esta reunión sirve para planificar el estudio de requisitos con la directora y obtener orientación de cómo realizarlos satisfactoriamente.
- **Definición general del Backlog.** Esta tarea consiste en la definición de las historias de usuario, sin entrar en mucho detalle, que tiene el proyecto.
- **Diseño general de la base de datos.** Consiste en la creación de un diagrama UML que muestre, de forma general, el diseño de tablas de la base de datos.
- **Diseño general del backend.** Consiste en la creación de un diagrama UML que muestre, de forma general, el diseño de clases de la parte *backend* de la aplicación.
- **Diseño general del frontend.** Consiste en la creación de un diagrama UML que muestre, de forma general, el diseño de clases de la parte *frontend* de la aplicación.
- **Diagramas de secuencia.** Esta tarea se centra en la creación de diagramas de secuencia que muestren el comportamiento de las funcionalidades más importantes del código.
- **Reunión #2.** Esta reunión sirve para mostrar el estudio de requisitos realizado a la directora y obtener *feedback* y preparar el primer Sprint para dar paso a la fase de implementación.

10.2. Fase de Implementación

La fase de implementación empieza el día 1 de Abril de 2019 y finaliza el día 7 de Junio de 2019. Como se comentó en el apartado de metodología, la metodología usada es Scrum y por lo tanto esta fase se divide en diferentes reuniones y Sprints. Antes de cada Sprint se realiza una reunión con la directora para ver como ha transcurrido el *Sprint* que acaba de finalizar y decidir que se hará en el siguiente. Cada Sprint consta de tres subtareas que son comunes a todos los Sprints las cuales son la implementación en código de las funcionalidades del Sprint, sus tests y la documentación relacionada con ellas. Durante esta fase se trabaja aproximadamente 6,5 horas diarias debido al gran peso que tiene y tiene una duración de 50 días laborables. A continuación se describen los Sprints y las reuniones que tiene el proyecto en orden de ejecución.

- **Sprint #1.** Se realizan las historias de usuario relacionadas con la autentificación del usuario como por ejemplo el registro, recuperación de contraseña, autentificación o cambio de contraseña.
- **Reunión #3.** Sirve para mostrar el resultado del Sprint #1 con la directora y preparar el siguiente.

- **Sprint #2.** Se realizan las historias de usuario relacionadas con la creación de anuncios globales, propios, la visualización de estos, su eliminación y su búsqueda.
- **Reunión #4.** Sirve para mostrar el resultado del Sprint #2 con la directora y preparar el siguiente.
- **Sprint #3.** Se realizan las historias de usuario relacionadas con el perfil del usuario el seguimiento de actividad a usuarios, la visualización de los anuncios de estos y las valoraciones de usuarios.
- **Reunión #5.** Sirve para mostrar el resultado del Sprint #3 con la directora y preparar el siguiente.
- **Sprint #4.** Se completan las historias de usuario que se centran en la comunicación entre usuarios mediante chats.
- **Reunión #6.** Sirve para mostrar el resultado del Sprint #4 con la directora y preparar el siguiente.
- **Sprint #5.** Se lleva a cabo una revisión de las historias de usuario y se realizan las funcionalidades centradas en las notificaciones.
- **Reunión #7.** Sirve para mostrar el resultado final del proyecto en lo que respecta a la aplicación y comentar las siguientes tareas para iniciar la fase de cierre.

10.3. Fase de Cierre

En cuanto a la fase de cierre, se inicia el día 10 de Junio de 2019 y finaliza el día 5 de Julio de 2019. Esta fase consiste en la finalización de la documentación que quede pendiente, su revisión y la preparación y la realización de la defensa del proyecto. Esta fase tiene una duración de 20 días laborables y se trabaja aproximadamente 5 horas diarias ya que la preparación de la defensa y la finalización de la memoria tienen gran peso. Las siguientes tareas son las que corresponden a esta fase en el orden de ejecución teniendo en cuenta las dependencias.

- **Finalización de la memoria.** La tarea consiste en finalizar lo que quede de la memoria o completar apartados que no se han concretado lo suficiente.
- **Revisión de la memoria.** Esta tarea consiste en la revisión de la memoria en cuanto a la gramática, léxico y sintaxis y presentación del documento.
- **Preparación de la presentación y demostración.** Esta tarea se centra en la preparación de la presentación del proyecto y de la demostración del funcionamiento de la aplicación.
- **Defensa del proyecto.** Consiste en la realización de la presentación y la demostración al tribunal del proyecto de fin de grado.

10.4. Tabla Resumen de Tareas

En la siguiente tabla se puede encontrar un resumen de las tareas comentadas en los anteriores apartados que describen las tres fases del proyecto. En la Tabla 3 se muestra la tarea, el número de horas que son necesarias para su realización, las dependencias que tienen, los recursos que son necesarios para su realización y sus correspondientes fechas de inicio y fin.

#Tarea	Nombre	Horas	Dependencias	Recursos	Inicio-Fin (2019)
T1	Análisis, contexto y alcance	17,5	-	R1, R2, R6, R11	18/02 - 26/06
T2	Planificación temporal	10,0	T1	R1, R2, R6, R11	26/02 - 04/03
T3	Presupuesto y sostenibilidad	12,5	T2	R1, R2, R6, R11	05/03 - 11/03
T4	Preparación presentación, documento final	27,5	T3	R1, R2, R6, R11	12/03 - 26/03
T5	Defensa fase inicial	2,0	T4	R1, R11	27/03 - 29/03
T6	Reunión #1	2,0	-	R1, R7, R8	18/02 - 18/02
T7	Definición del Backlog	22,5	T6	R1, R2, R6, R11	19/02 - 01/03
T8	Diseño base de datos	12,5	T7	R1, R2, R6, R11	04/03 - 08/03
T9	Diseño <i>backend</i>	12,5	T8	R1, R2, R6, R11	11/03 - 15/03
T10	Diseño <i>frontend</i>	12,5	T9	R1, R2, R6, R11	18/03 - 22/03
T11	Diagramas de secuencia	10,0	T10	R1, R2, R6, R11	25/03 - 28/03
T12	Reunión 2	2,0	T11	R1, R7, R8	29/03 - 29/03
T13	Sprint #1	63,0	T5, T12	R1, R2, R3, R4, R6, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18, R19, R21, R22, R26, R27, R28	01/04 - 11/04
T14	Reunión #3	2,0	T13	R1, R7, R8	12/04 - 12/04
T15	Sprint #2	63,0	T14	R1, R2, R3, R4, R6, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R24, R26, R27, R28	15/04 - 25/04
T16	Reunión #4	2,0	T15	R1, R7, R8	26/04 - 26/04
T17	Sprint #3	63,0	T16	R1, R2, R3, R4, R6, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R23, R26, R27, R28	29/04 - 09/05
T18	Reunión #5	2,0	T17	R1, R7, R8	10/05 - 10/05
T19	Sprint #4	63,0	T18	R1, R2, R3, R4, R6, R9, R10, R11, R12, R13, R14, R15,	13/05 - 23/05

				R16,R17, R18, R19, R20, R21, R22, R26, R27, R28	
T20	Reunión #6	2,0	T19	R1, R7, R8	24/05 - 24/05
T21	Sprint #5	63,0	T20	R1, R2, R3, R4, R6, R9, R10, R11, R12, R13, R14, R15, R16, ,R17, R18, R19, R20, R21, R22, R25, R26, R27, R28	27/05 - 06/06
T22	Reunión #7	2,0	T21	R1, R7, R8	07/06 - 07/06
T23	Finalización de la memoria	30,0	T22	R1, R11	10/06 - 14/06
T24	Revisión de la memoria	18,0	T23	R1, R11	17/06 - 19/06
T25	Preparación presentación y demostración	42,0	T24	R1, R11	20/06 - 28/06
T26	Defensa del proyecto	2,0	T25	R1, R2, R3, R5, R11	01/07 - 05/07

Leyenda	
	Fase de planificación y especificación de requisitos
	Fase de implementación
	Fase de cierre

Tabla 3: Resumen de las tareas a realizar durante el proyecto.

10.5. Diagrama de Gantt

En la Figura 1 se puede observar el diagrama de Gantt que representa las tareas que se realizan a lo largo del proyecto en la línea temporal de la planificación temporal.

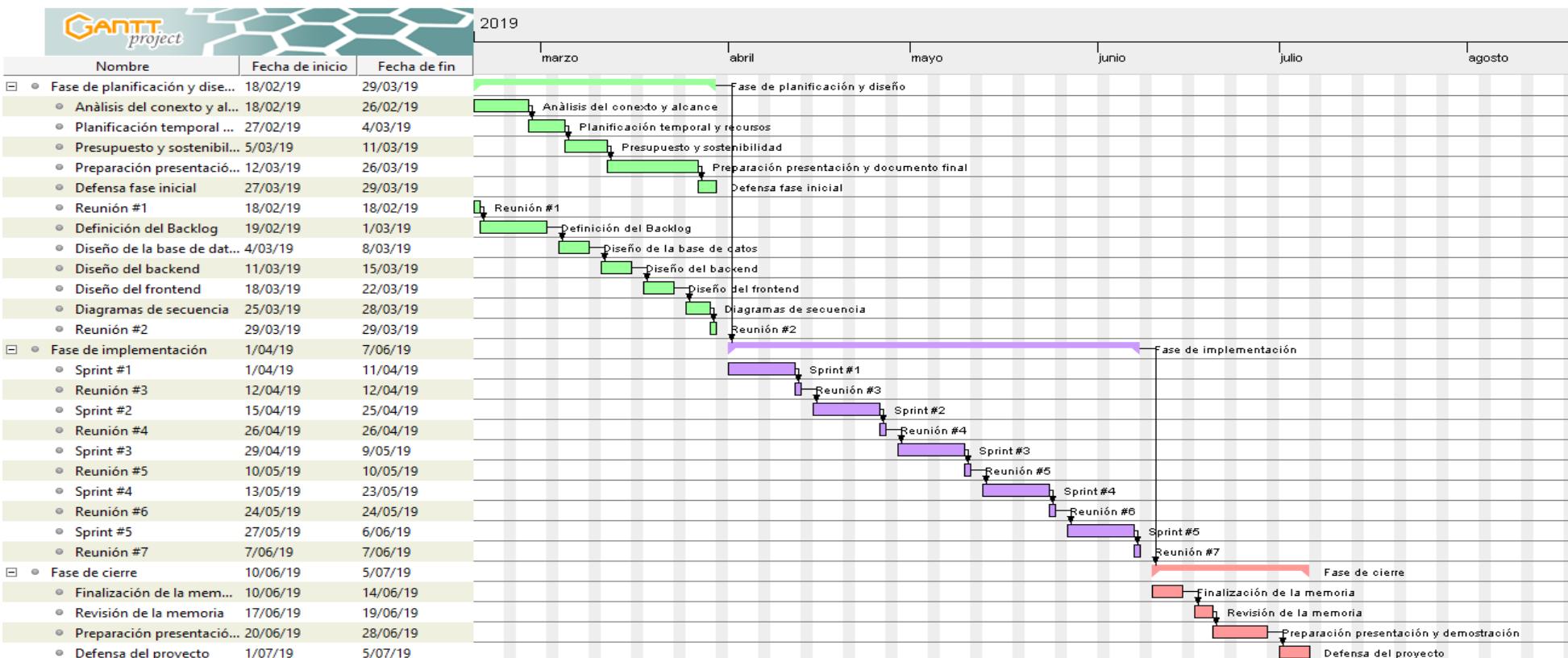


Figura 1: Diagrama de Gantt.

10.6. Valoración de Alternativas y Plan de Acción

A lo largo del desarrollo del proyecto se pueden producir desviaciones en lo que respecta a la planificación temporal. Estas desviaciones son más probables que aparezcan en la fase de implementación que en las demás debido a que es la parte más crítica del proyecto y tiene la mayor cantidad de obstáculos. Aun así no se descarta la posibilidad de que puedan aparecer en las demás. Las correcciones que se proponen a continuación están pensadas para intentar alterar la planificación inicial lo menos posible.

Un posible problema, como se comentó en el apartado de Obstáculos, es el tiempo de realización del proyecto. Al realizar un proyecto de estas dimensiones en un espacio de tiempo reducido, son más notables los retrasos que en un espacio de tiempo más amplio como es obvio y por esto el riesgo de que suceda es alto. Esto hace que la coordinación con la directora tenga que ser exacta y que la planificación sea exhaustiva. La principal idea es que el proyecto no pueda ampliar su duración, por lo tanto, para llegar a solucionar las desviaciones que puedan surgir debido al tiempo se deben añadir horas extras en las tareas que correspondan. De esta forma se intenta cumplir la planificación inicial y completar el proyecto sin dejar ningún cabo suelto.

En lo que respecta a la aparición o a los cambios de funcionalidades, hay que tener especial cuidado ya que pueden alterar la planificación temporal. El riesgo de que aparezcan es medio debido a las diversas reuniones con la directora que pueden conducir a cambiar funcionalidades para hacerlas más amplias o cambiarlas radicalmente. Con la aparición de nuevas funcionalidades o cambios, estas deben de intentar encajar en el calendario y por ello estamos también con un problema relacionado con el tiempo del proyecto. Hay dos soluciones posibles para intentar corregir esta desviación, la primera es quitar horas de otras funcionalidades que puedan realizarse con menos debido a que su estimación en horas era mayor a las reales necesarias y añadirlas a la nueva funcionalidad. Esto indirectamente supondría una reducción de horas de los Sprints que correspondan a dichas funcionalidades y una ampliación de los otros. La segunda consiste en dejar las horas que estaban y realizar extras para completar la nueva funcionalidad como se comentó en el segundo párrafo de este apartado. En cualquier otro caso la funcionalidad extra no podría ser añadida o modificada ya que supondría una inestabilidad para el proyecto.

Por último, encontramos el obstáculo de la experiencia programando en Android. Sin duda este es el obstáculo más crítico ya que la programación Android está en continuo desarrollo y es muy probable que la forma de realizar funciones haya cambiado hasta hoy. Esto puede producir que se tenga que hacer una búsqueda de información de como realizar las cosas afectando a las siguientes tareas y esto no se puede arreglar de ninguna forma salvo haciendo más horas de las planificadas. Puede darse el caso que las horas extras no fuesen suficientes para funcionalidades muy importantes y que requieran una búsqueda de información mayor. En este caso se podría no realizar funcionalidades más pequeñas y readaptar los Sprints para que estén equilibrados en cuanto a carga de trabajo. En todo caso, hay funcionalidades que se consideran el núcleo y que no pueden ser apartadas del proyecto. El riesgo de que aparezca este obstáculo es alto sin duda, no obstante, gracias a la gran comunidad de desarrolladores de Android no debería de ser tan dificultoso encontrar la información necesaria para llevar a cabo lo que se quiere conseguir.

11. Gestión Económica

En este apartado se realiza el estudio de costes que se producen a lo largo del proyecto. Esta gestión tiene en cuenta los costes directos, indirectos y de imprevistos y contingencias. Mediante el estudio del presupuesto se van presentando diversas tablas para presentar los datos de forma más amigable y al finalizar el apartado se muestra el presupuesto final con un resumen del total de cada tipo de coste. Finalmente se explica el control de gestión que se lleva a cabo para controlar las desviaciones que se produzcan en relación al presupuesto.

11.1. Costes Directos

Los costes directos son aquellos que tienen relación directa con el desarrollo de la aplicación pensada en este proyecto. Estos costes están relacionados con los recursos mencionados en la planificación de recursos y son los recursos humanos, los materiales y los de *software* y servicios. En los siguientes apartados se describe cada uno de ellos.

11.1.1. Recursos Humanos

Los costes de recursos humanos son aquellos costes que tienen relación con las personas que participan en el proyecto de manera directa. Este proyecto es realizado por una sola persona ya que es un trabajo de fin de grado y por lo tanto, solo hay un recurso humano y es el propio realizador del proyecto que desempeña todos los roles necesarios. No obstante, para realizar un análisis más ceñido a la realidad, se ha supuesto para este apartado que el proyecto es realizado como un proyecto *software* habitual.

En los proyectos *software* podemos encontrar muchos roles implicados pero se han escogido los principales para hacer el estudio de los costes de los recursos humanos. Estos son el jefe de proyecto, el analista, el programador, el diseñador y el *tester*. Para determinar los salarios de cada rol, se ha utilizado el informe de remuneración de PageGroup de 2018 [22] que proporciona los mínimos y máximos de los salarios brutos anuales de cada rol. Además, se ha supuesto que la jornada ordinaria anual queda establecida en 1760 horas respetando el máximo de 40 horas semanales y 8 diarias. En la Tabla 4 se puede encontrar un resumen de cada rol junto con su coste bruto anual, el coste bruto por hora, el tiempo estimado que deben participar en el proyecto, el coste total bruto de cada uno y el coste total estimado teniendo en cuenta la Seguridad Social u otros costes personales.

Rol	Coste bruto anual (€)	Coste bruto por hora (€/h)	Tiempo estimado (h)	Coste bruto estimado (€)	Coste total estimado (x1,35) (€)
Jefe de proyecto	40.000,0	22,7	175,5	3.983,9	5.378,3
Analista	24.000,0	13,6	72,0	979,2	1.321,9
Programador	18.000,0	10,2	201,0	2.050,2	2.767,8
Diseñador	18.000,0	10,2	43,5	443,7	599,0
<i>Tester</i>	18.000,0	10,2	106,5	1.086,3	1.466,5

Total		8.543,3	11.533,50
--------------	--	----------------	------------------

Tabla 4: Costes de los recursos humanos por rol en el proyecto.

Como se puede observar en la Tabla 4 el total del coste bruto de los recursos humanos asciende a 8.543,3 euros. Este solo es el coste bruto y por lo tanto faltaría añadir el coste de la Seguridad Social y otros costes del personal como pueden ser los costes de formación. Debido a que no se disponen de estos datos, dado que es un trabajo de fin de grado, se ha decidido realizar una aproximación mediante un factor del 1,35 del salario bruto de cada rol. De esta forma el coste total estimado de los recursos humanos, que incluye salario bruto más Seguridad Social más otros costes de personal, quedaría en un total de 11.533,5 euros.

De forma más específica, utilizando los recursos proporcionados en el apartado de planificación temporal se ha realizado un estudio del presupuesto por cada tarea que forma este proyecto en la Tabla 5.

Tarea	Horas estimadas tarea	Roles que participan	Horas estimadas roles	Coste estimado bruto (€)	Coste total estimado (x1,35) (€)
Análisis, contexto y alcance	17,5	Jefe de proyecto	17,5	397,3	536,4
Planificación temporal	10,0	Jefe de proyecto	10,0	227,0	306,5
Presupuesto y sostenibilidad	12,5	Jefe de proyecto	12,5	283,8	383,1
Preparación presentación, documento final	27,5	Jefe de proyecto	27,5	624,3	842,8
Defensa fase inicial	2,0	Jefe de proyecto	2,0	45,4	61,3
Reunión #1	2,0	Jefe de proyecto (100%) Analista (100%)	2,0	45,4 27,2	61,3 36,7
Definición del Backlog	22,5	Analista	22,5	306,0	413,1
Diseño base de datos	12,5	Analista	12,5	170,0	229,5
Diseño backend	12,5	Analista	12,5	170,0	229,5
Diseño frontend	12,5	Analista	12,5	170,0	229,5
Diagramas de secuencia	10,0	Analista	10,0	136,0	183,6
Reunión #2	2,0	Diseñador (100%) Programador (100%) <i>Tester</i> (100%) Jefe de proyecto (100%)	2,0	20,4 20,4 20,4 45,4	27,5 27,5 27,5 61,3
Sprint #1	63,0	Diseñador (10%) Programador (60%) <i>Tester</i> (30%)	6,3 37,8 18,9	64,3 385,7 192,8	86,8 520,7 260,3
Reunión #3	2,0	Diseñador (100%) Programador (100%) <i>Tester</i> (100%) Jefe de proyecto (100%)	2,0	20,4 20,4 20,4 45,4	27,5 27,5 27,5 61,3
Sprint #2	63,0	Diseñador (10%)	6,3	64,3	86,8

		Programador (60%) <i>Tester</i> (30%)	37,8 18,9	385,7 192,8	520,7 260,3
Reunión #4	2,0	Diseñador (100%) Programador (100%) <i>Tester</i> (100%) Jefe de proyecto (100%)	2,0	20,4 20,4 20,4 45,4	27,5 27,5 27,5 61,3
Sprint #3	63,0	Diseñador (10%) Programador (60%) <i>Tester</i> (30%)	6,3 37,8 18,9	64,3 385,7 192,8	86,8 520,7 260,3
Reunión #5	2,0	Diseñador (100%) Programador (100%) <i>Tester</i> (100%) Jefe de proyecto (100%)	2,0	20,4 20,4 20,4 45,4	27,5 27,5 27,5 61,3
Sprint #4	63,0	Diseñador (10%) Programador (60%) <i>Tester</i> (30%)	6,3 37,8 18,9	64,3 385,7 192,8	86,8 520,7 260,3
Reunión #6	2,0	Diseñador (100%) Programador (100%) <i>Tester</i> (100%) Jefe de proyecto (100%)	2,0	20,4 20,4 20,4 45,4	27,5 27,5 27,5 61,3
Sprint #5	63,0	Diseñador (10%) Programador (60%) <i>Tester</i> (30%)	6,3 37,8 18,9	64,3 385,7 192,8	86,8 520,7 260,3
Reunión #7	2,0	Diseñador (100%) Programador (100%) <i>Tester</i> (100%) Jefe de proyecto (100%)	2,0	20,4 20,4 20,4 45,4	27,5 27,5 27,5 61,3
Finalización de la memoria	30,0	Jefe de proyecto	30,0	681,0	919,4
Revisión de la memoria	18,0	Jefe de proyecto	18,0	408,6	551,6
Preparación presentación y demostración	42,0	Jefe de proyecto	42,0	953,4	1.287,1
Defensa del proyecto	2,0	Jefe de proyecto	2,0	45,4	61,3
Total	560,5			8.543,3	11.533,50

Leyenda	
	Fase de planificación y especificación de requisitos
	Fase de implementación
	Fase de cierre

Tabla 5: Costes de los recursos humanos por actividad.

La Tabla 5 se muestran las diferentes tareas a realizar en el proyecto. El proyecto como se dijo en el apartado de Planificación Temporal consta de 560,5 horas en total. Debido a que en algunas de las tareas participan más de un rol, como puede ser el caso de los Sprint, se ha otorgado un porcentaje estimado del trabajo de las horas que debe realizar cada rol en dicha tarea y consecuentemente sus

horas estimadas y el coste bruto y total de estas horas. Como dato a tener en cuenta, a pesar de que la parte de documentación durante los Sprints la realiza el programador, el jefe de proyecto debe de asistir a las reuniones ya que de esta forma puede observar como va avanzando el proyecto durante la fase de implementación y modificar la planificación si es necesario. De esta forma se está cumpliendo la metodología Scrum. Finalmente tanto en la Tabla 4 y la Tabla 5 se muestran el coste bruto final de los recursos humanos estimado en 8.543,3 euros y el coste total aproximado teniendo en cuenta Seguridad Social y otros costes de personal en 11.533,5 euros.

11.1.2. Recursos *Hardware*

Los costes de los recursos materiales son aquellos costes que tienen relación con las herramientas físicas que se utilizan de manera directa en el desarrollo del proyecto. En lo que respecta a estos recursos, la Tabla 6 muestra los necesarios para llevar a cabo el proyecto junto con su precio estimado, el número de unidades necesarias y su vida útil. También se proporciona la cantidad amortizada para cada uno siguiendo la fórmula que se muestra a continuación.

$$\text{Amortización (€)} = \frac{\text{Precio compra equipo (€)} * \text{Horas uso recurso (horas)}}{\text{Vida útil equipo (años)} * \text{Jornada ordinaria anual (horas)}}$$

Recurso	Precio estimado (€)	Unidades	Vida útil (años)	Amortización (€)	Coste estimado (€)
Ordenador portátil	850,0	1	4	67,8	850,0
Teléfono móvil Huawei P9 Lite	329,0	1	3	20,6	329,0
Teléfono móvil Xiaomi MI A2	184,5	1	3	11,4	184,5
Total				99,8	1363,5

Tabla 6: Coste de los recursos hardware en el proyecto.

Si se observa la Tabla 6 el coste estimado de los recursos materiales se estima a un total de 1.363,5 euros de los cuales tan solo se amortiza 99,4 euros ya que el proyecto tiene una duración corta respecto a la vida útil de los recursos.

11.1.3. Recursos y Servicios *Software*

Los costes de los recursos *software* y servicios son los costes que tienen relación con las herramientas de tipo *software* y los servicios utilizados para llevar a cabo el proyecto. En la Tabla 7 se proporciona información de dichos recursos de manera similar a los anteriores.

Recurso	Precio (€/mes)	Unidades	# Veces pagado (meses)	Amortización (€)	Coste estimado (€)
Gmail	0,0	1	-	0,0	0,0
Skype	0,0	1	-	0,0	0,0
Bitbucket	0,0	1	-	0,0	0,0
Trello	0,0	1	-	0,0	0,0
Google drive/docs	0,0	1	-	0,0	0,0
Adobe Illustrator	19,7	1	2	39,4	39,4
Android Studio	0,0	1	-	0,0	0,0
Eclipse	0,0	1	-	0,0	0,0
Firebase	0,0	1	-	0,0	0,0
Postman	0,0	1	-	0,0	0,0
JSON	0,0	1	-	0,0	0,0
Retrofit	0,0	1	-	0,0	0,0
RxJava	0,0	1	-	0,0	0,0
Picasso	0,0	1	-	0,0	0,0
Spring Boot	0,0	1	-	0,0	0,0
Hibernate	0,0	1	-	0,0	0,0
API Cloudinary	0,0	1	-	0,0	0,0
API Google Maps	0,0	1	-	0,0	0,0
PostgreSQL	0,0	1	-	0,0	0,0
JUnit	0,0	1	-	0,0	0,0
Mockito	0,0	1	-	0,0	0,0
H2	0,0	1	-	0,0	0,0
Servidor de pruebas (Heroku)	0,0	1	-	0,0	0,0
Servidor del proyecto (DigitalOcean)	5,0	1	3	15,0	15,0
Total				54,4	54,4

Tabla 7: Coste de los recursos software y servicios en el proyecto.

Como se puede observar, a diferencia de los anteriores recursos, estos tienen una columna de precio por mes y veces pagado en vez de precio único ya que estos elementos normalmente, si son de pago, se paga una licencia que dura un determinado tiempo. También cabe destacar que como el pago de estos recursos es por un tiempo, específicamente durante la fase de implementación, la amortización siempre va a ser muy cercana al coste exacto ya que se realiza el pago cuando es necesario su uso.

Entrando de forma más específica, se puede ver que la mayoría de estos recursos son gratuitos a excepción de Adobe Illustrator y el Servidor de DigitalOcean. Para el primero de estos dos se ha considerado la versión de pago que requiere licencia de estudiante como muestra su página web [23] durante dos meses que dura la fase de implementación. Para el segundo se ha optado por la versión

del servidor más barato [24] debido a que este proyecto no demanda necesidad de más durante tres meses que incluye la fase de implementación y la fase de cierre para la demostración de la aplicación. Finalmente el coste estimado de estos recursos ronda en los 54,4 euros tan solo que son amortizados por lo comentado anteriormente.

11.2. Costes Indirectos

Los costes indirectos son aquellos costes que afectan a la realización de la aplicación que se propone para este proyecto pero que no están relacionados directamente. Cabe mencionar que el proyecto se realiza en las instalaciones de la universidad o en el hogar particular. Normalmente gastos como la luz, el alquiler o el teléfono forman parte de los costes indirectos pero debido a que no se puede saber el coste que le supone a la universidad solo se tienen en cuenta los del hogar. A continuación se muestra la Tabla 8 que contiene los costes indirectos.

Producto	Precio estimado	Cantidad	Periodo de tiempo (meses)	Coste estimado (€)
Luz	0,125 €/kWh	-	5	456,3
Internet	45,0 €	-	5	225,0
Línea telefónica	30,0 €	-	5	150,0
Tarjeta T-10 1 zona	10,2 €	2	-	20,4
Total				851,7

Tabla 8: Costes indirectos.

Como se puede observar en la Tabla 8 se consideran elementos como la luz, el internet y la línea telefónica típicos de un lugar de trabajo. Estos recursos solo se atribuyen al hogar como se mencionó antes y se tienen en cuenta durante la realización del proyecto, es decir, 5 meses. Además también hay como recurso indirecto las tarjetas de desplazamiento por Barcelona. Este recurso es para llevar a cabo el desplazamiento mediante transporte público hasta la universidad para las reuniones y las defensas y para volver a casa. Su precio estimado es el proporcionado por el Transport Metropolità de Barcelona (TMB) [25]. El coste total estimado es de los recursos indirectos es de 851,7 euros.

11.3. Costes de Imprevistos y Contingencias

Los costes de imprevistos son aquellos costes que no se han tenido en cuenta ya que se presentan de forma imprevista durante el proyecto. En la Tabla 9 se puede encontrar los diferentes tipos de imprevistos que pueden surgir a lo largo del proyecto.

Imprevisto	Probabilidad	Precio estimado (€)	Coste estimado (€)
Avería ordenador Asus	10	300,0	30,0
Avería teléfono Huawei	5	50,0	2,5
Avería teléfono Xiaomi	5	50,0	2,5
Total			35,0

Tabla 9: Costes de imprevistos.

Como se puede observar en la Tabla 9, todos los imprevistos están relacionados con el *hardware* que se utiliza en el proyecto. Se ha dado una probabilidad más alta a una avería del ordenador debido a que se utiliza durante todas las fases del proyecto y por tanto es más susceptible de fallo que los teléfonos que solo se utilizan en la fase de implementación. El coste estimado de los imprevistos ronda en los 35,0 euros.

En cuanto a los costes de contingencia, estos son los destinados a cubrir los gastos inesperados del proyecto. A continuación se muestra mediante la Tabla 10 el coste de contingencia.

Tipo de coste	Porcentaje	Coste (€)
Costes directos	15	1.942,7
Costes indirectos	15	127,8
Total		2.070,5

Tabla 10: Costes de contingencia.

Se ha reservado un 15% tanto de los costes directos como los indirectos para esta reserva. La contingencia para costes directos es de 1.942,7 euros mientras que para los indirectos 127,8 euros.

11.4. Resumen del Presupuesto Inicial

Una vez realizado el estudio de costes para cada tipo se tiene el presupuesto inicial del proyecto tal y como se muestra en la Tabla 11. Cabe destacar algunas consideraciones respecto a este presupuesto.

- Se presupone que los costes son constantes pues al ser un proyecto de corta duración no se prevé ningún aumento.
- No se añade ningún margen de beneficios al presupuesto debido a que es un trabajo de fin de grado y está realizado sin ánimo de lucro.

Tipo de Coste	Presupuesto (€)
Directos	12.951,4
Indirectos	851,7
Imprevistos	35,0
Contingencias	2.070,5
Total	15.908,6

Tabla 11: Presupuesto inicial del proyecto.

Como se puede observar en la Tabla 11 el coste final estimado del proyecto es la suma de todos los costes mencionados y calculados en los apartados anteriores y es de 15.908,6 euros en total.

11.5. Control de Gestión

Con el fin de controlar los costes que se van realizando durante la ejecución del proyecto en relación al presupuesto inicial creado se ha decidido hacer un control de gestión. Mediante este control de gestión se puede observar desviaciones producidas en el presupuesto y de esta forma encontrar la causa que las ha producido y corregir el presupuesto.

En lo que respecta a los costes producidos por los recursos humanos, debido a que son el mayor coste de recursos de todo el proyecto y es un problema común una desviación en el tiempo de la duración del proyecto es importante tenerlos controlados. Para llevar a cabo este control se utiliza un fichero Excel llamado Project Record Track donde todos los roles que participen en el proyecto deben apuntar cada día las horas que realicen de cada tarea junto con su fecha y hora. De esta forma se pueden observar las desviaciones de los recursos humanos al finalizar cada tarea y ver si se ha desviado utilizando como comparativa la Tabla 5 mostrada anteriormente que contiene las horas estimadas de cada tarea. Los indicadores utilizados son los siguientes.

- **Desvío de recursos humanos en precio** = (Coste estimado - Coste real) * Horas reales.
- **Desvío de mano de obra en consumo** = (Consumo horas estimado - Consumo horas real) * Coste estimado.
- **Desvío total en mano de obra** = Total coste estimado de recursos humanos - Total coste real recursos humanos.

En cuanto a los costes producidos por los recursos materiales, aunque son de pago único, también se deben de tener en cuenta ya que su precio puede variar al estimado y ser superior o inferior a la hora de comprarlos dependiendo del mercado. Por ello se ha optado por realizar una comparativa con los costes estimados y los costes reales a la hora de su compra a través de otro archivo Excel donde se anotan los precios reales a la hora de la compra. Los indicadores utilizados son los siguientes.

- **Desvío total en materiales** = total coste estimado recursos materiales - total coste real recursos materiales.

Comentando ahora los costes de los recursos de *software* y servicios, al ser todos gratuitos a excepción de dos que se pagan por mes y teniendo en cuenta que sus precios si varían lo hacen al iniciar el año, no se realizará ningún control. Esto es debido a que este proyecto se realiza a partir de febrero una vez que los precios de dichos servicios y licencias ya se han establecido y no pueden variar por sus políticas de empresa.

Una vez se han descrito los anteriores que forman parte de los costes directos se puede entrar a tratar a los costes directos en general junto con los indirectos, contingencia e imprevistos. Para todos estos se anota también en un archivo Excel los gastos que se van realizando a medida que avanza el proyecto. Los indicadores utilizados son los siguientes.

- **Desvío total en costes (Directos, Indirectos, etc)** = total coste estimado en costes (Directos, Indirectos, etc) - total coste real en costes (Directos, Indirectos, etc).

12. Sostenibilidad y Compromiso Social

En este apartado se describe la sostenibilidad y el compromiso social que se debe tener en cuenta con la realización del proyecto. Primero de todo se expone en un breve resumen una auto-evaluación de la competencia de sostenibilidad que muestra mis conocimientos y mis defectos en este ámbito. Seguidamente se tiene en cuenta la sostenibilidad y el compromiso social del proyecto desde tres dimensiones distintas, ambiental, económica y social mediante el uso de la matriz de sostenibilidad mostrada en la Tabla 12.

	PPP	Vida Útil	Riesgos
Ambiental	Consumo de diseño	Huella ecológica	Ambientales
Económico	Factura	Plan de viabilidad	Económicos
Social	Impacto personal	Impacto social	Sociales

Tabla 12: Matriz de sostenibilidad.

12.1. Auto-evaluación de la Competencia de Sostenibilidad

Al realizar la encuesta he reflexionado sobre el nivel de formación que tengo actualmente relacionado con la sostenibilidad. Además, he considerado todo lo que he aprendido mediante el grado universitario y lo que no. A continuación se expone algunos puntos fuertes y débiles que se considera que se tiene una vez realizada la encuesta.

Desde mi punto de vista, creo que soy consciente y capaz de analizar la sostenibilidad de un proyecto desde los ámbitos ambiental, social y económico ya que durante el grado universitario he realizado múltiples análisis para los proyectos que he realizado. A la hora del desarrollo soy participativo en estos temas intentando dar mi opinión y soluciones para que el proyecto sea lo más sostenible posible. También intento que los productos de los proyectos que desarrollo tengan un impacto positivo para vida de las personas y puedan ser accesibles por la mayor cantidad de gente posible. Un ejemplo claro es este proyecto ya que mi propósito es fomentar el consumo colaborativo y que la gente ofrezca su ayuda y obtenga de otros. Por último, comprendo y conozco de forma suficiente aunque no tanto como debería ser el proceso de gestión de un proyecto y se como realizarla durante la vida útil del proyecto. Durante esta primera fase del proyecto se ha intentado mostrar este último punto a través de la gestión económica del proyecto.

En lo que respecta a los puntos débiles, hay temas que se tratan en la encuesta que no era consciente de su existencia y no tengo suficientes conocimientos sobre ello. Un ejemplo de estos son los principios deontológicos que no sabía de su existencia hasta el momento de realizar la encuesta. Tampoco soy conocedor de las tecnologías “sostenibilistas” aplicables a los proyectos TIC y los indicadores ambientales. Por último, quería comentar que al finalizar la encuesta me he informado y ahora tengo una idea más o menos clara de que son aunque considero que sigo sin tener los conocimientos necesarios para considerar una valoración positiva en la encuesta.

12.2. Dimensión Ambiental

Durante este proyecto se utilizan diferentes herramientas que repercuten en el ambiente por su consumo. Primero tenemos los recursos materiales como el ordenador y los dos teléfonos móviles. Durante la fase de implementación se utilizan los dos teléfonos que permiten comprobar el *software* en dispositivos reales pero su consumo es mínimo ya que solo se utilizan para eso. Este consumo supondría 0,005 kWh que en 325 horas que dura la fase de implementación sería 1,625 kW de consumo total. Teniendo en cuenta que la Generalitat de Catalunya estima que las emisiones de CO₂ en 2018 eran de 321 g/kWh [26], supondría que el uso del teléfono móvil produciría un total de 1,605 g de emisión de CO₂.

En cuanto al ordenador, es utilizado a lo largo del proyecto para la realización de la memoria y de la aplicación y su consumo de electricidad es más elevado que el resto pero imposible de minimizar ya que es necesario en todo momento. Teniendo en cuenta que el ordenador es utilizado en todas las horas del proyecto a excepción de las reuniones y las defensas, esto supone un total de 542,5 horas de las 560,5 totales. Este ordenador consume una media de 0,092 kWh que en 542,5 horas supone un consumo de 49,9 kW en total en el proyecto. Como antes, teniendo en cuenta 321 g/kWh de emisión de CO₂ obtendríamos que la producción de CO₂ por parte del ordenador sería de 29,5 g.

Debido a que no se ha podido obtener los datos de consumo y contaminación en la fabricación de los teléfonos y el ordenador, no se ha podido determinar el impacto ambiental que tienen sus fabricaciones. En cuanto a los recursos de *software* y servicios pasaría de forma similar a los anteriores. Cabe destacar de estos últimos el uso del servidor, cuyo consumo depende de las empresas que lleven a cabo el mantenimiento, en este caso, Digital Ocean y Heroku y de las tarifas del país donde se encuentren los servidores los cuales no se posibilita su selección.

También, al tratarse de un proyecto *software* se tiene en mente la reutilización de código en otros proyectos. Esto sin duda mejoraría el impacto ambiental de los futuros proyectos que tengan funcionalidades similares al de este. Esta reutilización provocaría que no solamente el impacto ambiental disminuyese sino que también lo hiciesen el coste del nuevo proyecto y el número de horas ya que gran parte ya estaría hecho.

En lo que respecta a la vida útil y a los riesgos, como este proyecto se trata de un proyecto *software*, se ha creado todo de manera digital y no se ha producido ninguna fabricación ni uso de materias primas o herramientas manufacturadas. Todo eso dejando de lado la necesidad de la compra de recursos como el ordenador que si las ha necesitado y que como se ha mencionado en el anterior párrafo, no se han podido obtener datos. De tal forma, no se produce ninguna huella ecológica salvo el provocado por el consumo de electricidad a lo largo del proyecto y el mantenimiento de este una vez acabado.

Finalmente, comparando con otros proyectos similares no se realiza una mejora sustancial del impacto que produce en cuanto a la dimensión ambiental. Esto es debido a lo comentado anteriormente, al ser un proyecto *software* el uso de electricidad para el ordenador es necesario

para llevarlo a cabo y para mejorar esto la única forma es que las empresas creen sistemas que consuman menos recursos.

12.3. Dimensión Económica

Para determinar si el proyecto es viable económicamente se ha realizado en el apartado de Gestión Económica un estudio de los diferentes costes de los recursos que son necesarios. Principalmente, la mayor parte de los costes del proyecto provienen de los recursos humanos seguidos de los materiales, como por ejemplo un ordenador, que son necesarios para su desarrollo. En cuanto a los recursos de *software* y servicios la mayoría son gratuitos y no suponen un gasto económico. Además, se ha considerado la posibilidad de imprevistos que puedan surgir durante el progreso del proyecto y se ha reservado un fondo monetario para contingencias. También se ha proporcionado mecanismos para determinar las posibles desviaciones que puedan surgir a lo largo del proyecto y así poder realizar ajustes en el presupuesto. Con todo esto se ha obtenido un presupuesto ajustado y asequible para el proyecto.

Comparando con otras soluciones existentes, este proyecto puede ser competitivo con el presupuesto diseñado si se tiene en cuenta que es realizado por una única persona pero en lo que respecta a velocidad de realización no. Se ha otorgado a cada tarea el mínimo de horas posibles para poder realizarlas con el mínimo de presupuesto posible. Al participar una única persona realizando todos los roles, el tiempo es mayor que al tener un equipo a pesar de que se ha dado a cada tarea el mínimo de horas posibles para su realización. Esto condiciona a que las tareas no puedan ser realizadas de forma paralela de forma real, aumentando así el número de horas totales y afectando al presupuesto final. En el caso de las soluciones existentes emplean más dinero en recursos humanos que en este proyecto pero se ven beneficiados con una disminución de horas de trabajo ya que las tareas se realizan con más rapidez justo al contrario que en este proyecto. También emplean más recursos materiales al ser más gente y en este caso solamente se requieren los materiales para una única persona.

En lo que respecta al beneficio de este proyecto, el beneficio es para sus usuarios mediante la facilitación y obtención de ayuda y no para los desarrolladores salvo el beneficio pedagógico que pueda otorgar ya que es un proyecto sin ánimo de lucro. Debido a esto, no está pensado para obtener beneficios monetarios con él. Si en algún futuro se decidiese obtener beneficios con él, la cantidad invertida podría ser recuperada mediante la incorporación de publicidad como muchas otras aplicaciones tienen en el mercado. Otras maneras de obtener beneficios serían poner precio al producto o cobrar una suscripción por uso. No obstante, estas dos últimas opciones se consideran para este proyecto soluciones poco elegantes ya que la intención real es promover y ayudar a las personas.

Por último, en cuanto a los riesgos posibles, dejando de lado los imprevistos que se mencionan en el apartado de Gestión Económica, desde un primer punto de vista parecen pocos. Al ser un proyecto centrado en el uso de recursos *software* y servicios que son en la gran mayoría gratuitos y muy populares facilita que no haya ningún problema. Si se mira desde el peor punto de vista podría pasar que alguno de estos desapareciese y tuviese que ser sustituido por otro pero al haber múltiples

alternativas gratuitas no importaría. En lo que respecta a los de pago si afectarían si una vez realizada la aplicación se quieren añadir nuevas funcionalidades y seguir dando soporte pues el precio podría aumentar. También, al no depender de ningún tercero en cuanto a datos se refiere facilita a no tener problemas de cara en un futuro porque estos datos hayan pasado a ser de pago.

12.4. Dimensión Social

Como se comentó en el apartado de Contextualización, las aplicaciones se han extendido de forma abrumadora por todo el mundo. Esto ha dado paso a nuevas formas de ayudar y a colaborar y aquí es donde la aplicación que se propone realizar en este proyecto entra en escena. Esta aplicación permite que personas que necesiten ayuda puedan obtenerla de otras personas de forma sencilla y de esta forma promover la ayuda y la colaboración.

Este proyecto supone para el autor una forma de demostrar gran parte de los conocimientos que han sido obtenidos a lo largo del grado universitario y en especial la especialidad de *software*. También de profundizar en más detalle conceptos que no se habían tratado con demasiada ímpetu como en la gestión de proyectos. Por último, supone un aprendizaje de la realización de proyectos Android junto con las herramientas relacionadas con este sistema operativo.

En cuanto a lo que puede aportar a los usuarios, el principal punto es el de obtener ayuda de otras personas ya sea a través de un precio o no. Esto facilita la vida de los usuarios ya que con tan solo un teléfono móvil y unos cuantos segundos podría disponer de dicha ayuda. También beneficiaria a anunciantes de comercios ya que podrían ofrecer su ayuda a los usuarios y darse a conocer y ampliar su negocio. La idea es crear una comunidad de usuarios que se ayuden entre sí y así favorecer el consumo colaborativo.

Actualmente no hay casi aplicaciones similares a la que se propone en el proyecto ya que muchas ya no se encuentran disponibles tal y como se comentó en el apartado del Estado del Arte, por esto es necesario la creación de una nueva. Estas aplicaciones requieren de objetos como monedas virtuales para poder hacer uso de diversas funcionalidades. Además estas soluciones no agilizan el proceso de obtener ayuda y esto implica que muchos usuarios no la obtengan en un plazo de tiempo mínimamente corto. Es por esto que se quiere realizar una aplicación desde cero que sea amigable, fácil y atractiva y completamente accesible para el usuario y permita agilizar el proceso de obtención de ayuda mediante mecanismos como recomendaciones o valoraciones.

13. Especificación de Requisitos

A lo largo de este apartado se presenta como se han obtenido los requisitos del proyecto así como cuales son empezando por los funcionales y finalizando con los no funcionales.

13.1. Obtención de Requisitos

Para obtener y determinar los requisitos del proyecto se ha utilizado una técnica muy común en la obtención de requisitos llamada *brainstorming* o lluvia de ideas [27, pp 24]. Esta técnica consiste en una serie de reuniones en grupo que tienen como objetivo la generación de ideas en un ambiente libre de críticas. Estas reuniones suelen estar formadas de cuatro a diez integrantes donde uno de estos es el jefe de la sesión que se encarga de controlarla. La lluvia de ideas tiene diferentes fases:

- **Preparación de la sesión:** En esta fase se prepara la sesión, el lugar y se determinan los diferentes integrantes.
- **Realización de la sesión:** Durante esta fase se proponen ideas a través de la primera idea que ha propuesto el jefe de la sesión. Estas ideas pueden ser básicas o avanzadas o combinadas entre varias ideas de los integrantes. No se permite la crítica.
- **Consolidación de las ideas:** Se revisan las ideas propuestas, se organizan, se descartan aquellas que no son asequibles y se priorizan.

Debido a que el proyecto lo realiza una única persona, se ha realizado una primera lluvia de ideas las cuales se han apuntado en un papel. Posteriormente estas ideas se han compartido con una serie de compañeros del autor para que actuaran como si formaran parte del equipo de desarrollo y diesen sus opiniones e ideas y se ha debatido cuales de estas podían ser más interesantes para una posible aplicación. De esta forma se ha conseguido descartar ideas y se han priorizado algunas según lo discutido entre el autor y los diferentes compañeros.

13.2. Requisitos Funcionales: Historias de Usuario

En los siguientes subapartados se describen las historias de usuario que forman parte de los requisitos funcionales del proyecto. Se han dividido en los siguientes temas: autentificación y cuenta, perfiles, gestión de anuncios, listas de anuncios, valoraciones, conversaciones y notificaciones. Cabe destacar que para las horas reales se ha redondeado a la media hora, por ejemplo si para una historia de usuario se ha tardado 5 horas y 45 minutos se han considerado 6 horas y si se ha tardado 5 horas y 15 minutos en total entonces, 5 horas y 30 minutos.

13.2.1. Autentificación y Cuenta

Iniciar sesión	
Identificador de historia: 1.	Sprint asignado: 1.
Riesgo de desarrollo: Medio.	Valor de la historia: 6/10.
Horas estimadas: 16,0 horas.	Horas reales: 16,5 horas.

Descripción: Como usuario ya registrado, quiero poder iniciar sesión mediante el uso de mi nombre de usuario o correo electrónico y mi contraseña, de forma que pueda entrar y hacer uso de la aplicación.

Criterios de aceptación:

- **Escenario 1: Inicio de sesión válido.**

Dado un nombre de usuario o correo y una contraseña registrados, cuando el usuario pulsa el botón de iniciar sesión, entonces, se compara la información del servidor con las introducidas por el usuario, se devuelven los *tokens* necesarios para el inicio de sesión al usuario y se le redirige a la pantalla principal de la aplicación.

- **Escenario 2: Nombre de usuario o correo vacío.**

Dado un nombre de usuario o correo vacío, cuando el usuario pulsa el botón de iniciar sesión, entonces, se muestra un mensaje de error indicando que se encuentra vacío y se mantiene al usuario en la misma pantalla.

- **Escenario 3: Contraseña vacía.**

Dada una contraseña vacía, cuando el usuario pulsa el botón de iniciar sesión, entonces, se muestra un mensaje de error indicando que se encuentra vacía y se mantiene al usuario en la misma pantalla.

- **Escenario 4: Nombre de usuario o correo erróneo.**

Dado un nombre de usuario o correo no registrado, cuando el usuario pulsa el botón de iniciar sesión, entonces, se muestra un mensaje de error que indica que alguna de las credenciales no son correctas y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 5: Contraseña errónea.**

Dada una contraseña no vinculada con el nombre de usuario o correo del formulario, cuando el usuario pulsa el botón de iniciar sesión, entonces, se muestra un mensaje de error que indica que alguna de las credenciales no son correctas y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 6: Sin acceso a Internet.**

Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de iniciar sesión, entonces, se muestra un mensaje de error indicando que no se ha podido iniciar sesión y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 7: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de iniciar sesión, entonces, se muestra un mensaje de error indicando que no se ha podido iniciar sesión y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

Tabla 13: Iniciar sesión.

		Cerrar sesión
Identificador de historia: 2.	Sprint asignado: 1.	

Riesgo de desarrollo: Medio.	Valor de la historia: 6/10.
Horas estimadas: 5,7 horas.	Horas reales: 5,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder cerrar sesión en la aplicación, de forma que pueda iniciar sesión con otra cuenta o que otras personas no puedan ver mi cuenta desde mi dispositivo.	
Criterios de aceptación:	
<ul style="list-style-type: none"> ● Escenario 1: Cierre de sesión válido. Dado un usuario que ha iniciado sesión, cuando el usuario pulsa el botón de cerrar sesión, entonces, se elimina la sesión del usuario de la aplicación, se eliminan los datos necesarios del servidor y se redirige al usuario a la pantalla de inicio de sesión. ● Escenario 2: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de cerrar sesión, entonces, se muestra un mensaje de error indicando que no se ha podido cerrar sesión y se mantiene al usuario en la misma pantalla. ● Escenario 3: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario pulsa el botón de cerrar sesión, entonces, se muestra un mensaje de error indicando que no se ha podido cerrar sesión y se mantiene al usuario en la misma pantalla. 	

Tabla 14: Cerrar sesión.

Registrar usuario	
Identificador de historia: 3.	Sprint asignado: 1.
Riesgo de desarrollo: Medio.	Valor de la historia: 6/10.
Horas estimadas: 16,0 horas.	Horas reales: 16,5 horas.
Descripción: Como potencial nuevo usuario, quiero poder registrarme con mi correo electrónico, un nombre de usuario que pueda elegir propio y una contraseña propia, de forma que pueda entrar y hacer uso de la aplicación.	
Criterios de aceptación:	
<ul style="list-style-type: none"> ● Escenario 1: Registro válido. Dado un nombre de usuario, un correo electrónico y una contraseña, cuando el usuario pulsa el botón de registrar, entonces, se comprueba que estos datos sean válidos para el servidor, se envían a este, se devuelven los <i>tokens</i> necesarios para el inicio de sesión al usuario y se redirige a este a la pantalla principal de la aplicación. ● Escenario 2: Nombre de usuario vacío. Dado un nombre de usuario vacío, cuando el usuario pulsa el botón de registrar, entonces, se muestra un mensaje de error que se encuentra vacío y se mantiene al usuario en la misma pantalla. ● Escenario 3: Nombre de usuario ya registrado. 	

Dado un nombre de usuario ya registrado, cuando el usuario pulsa el botón de registrar, entonces, se muestra un mensaje de error indicando que el nombre de usuario ya existe y se mantiene al usuario en la misma pantalla.

- **Escenario 5: Correo electrónico vacío.**

Dado un correo electrónico vacío, cuando el usuario pulsa el botón de registrar, entonces, se muestra un mensaje de error indicando que se encuentra vacío y se mantiene al usuario en la misma pantalla.

- **Escenario 6: Correo electrónico inválido.**

Dado un correo electrónico inválido, cuando el usuario pulsa el botón de registrar, entonces, se muestra un mensaje de error indicando que el correo no cumple los requisitos y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 7: Correo electrónico ya registrado**

Dado un correo electrónico ya registrado, cuando el usuario pulsa el botón de registrar, entonces, se muestra un mensaje de error indicando que el correo electrónico ya existe y se mantiene al usuario en la misma pantalla.

- **Escenario 8: Contraseña vacía.**

Dada una contraseña nueva vacía, cuando el usuario pulsa el botón de registrar, entonces, se muestra un mensaje de error indicando que la contraseña se encuentra vacía y se mantiene al usuario en la misma pantalla.

- **Escenario 9: Contraseña inválida.**

Dada una contraseña inválida, cuando el usuario pulsa el botón de registrar, entonces, se muestra un mensaje de error indicando que la contraseña no cumple los requisitos y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 10: Sin conexión a Internet.**

Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de registrar, entonces, se muestra un mensaje de error indicando que no se ha podido registrar al usuario y se mantiene al este en la misma pantalla sin alterar los datos introducidos.

- **Escenario 11: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de registro, entonces, se muestra un mensaje de error indicando que no se ha podido registrar al usuario y se mantiene a este en la misma pantalla sin alterar los datos introducidos.

Tabla 15: Registrar usuario.

Recuperar contraseña	
Identificador de historia: 4.	Sprint asignado: 1.
Riesgo de desarrollo: Medio.	Valor de la historia: 6/10.
Horas estimadas: 14,0 horas.	Horas reales: 16,0 horas.
Descripción: Como usuario que no puede iniciar sesión pero está registrado en la aplicación, quiero poder recuperar la contraseña olvidada a través de mi correo electrónico, de forma que	

pueda iniciar sesión y hacer uso de la aplicación con la nueva contraseña.

Criterios de aceptación:

Primera fase: Envío de correo electrónico de recuperación.

• **Escenario 1: Envío de correo electrónico de recuperación válido.**

Dado un correo electrónico vinculado a un usuario registrado, cuando el usuario pulsa el botón de enviar correo, entonces, se envía un código a dicho correo y se muestra al usuario el formulario de recuperación de contraseña de la aplicación.

• **Escenario 2: Correo electrónico vacío.**

Dado un correo electrónico vacío, cuando el usuario pulsa el botón de enviar correo, entonces, se muestra un mensaje de error indicando que se encuentra vacío y se mantiene al usuario en la misma pantalla.

• **Escenario 3: Correo electrónico inválido.**

Dado un correo electrónico inválido, cuando el usuario pulsa el botón de enviar correo, entonces, se muestra un mensaje de error indicando que el correo no cumple los requisitos y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

• **Escenario 4: Correo electrónico erróneo.**

Dado un correo no registrado, cuando el usuario pulsa el botón de enviar correo, entonces, se muestra un mensaje de error indicando que el correo es erróneo y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

• **Escenario 5: Sin acceso a Internet.**

Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de enviar correo, entonces, se muestra un mensaje de error indicando que no se ha podido enviar el correo de recuperación y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

• **Escenario 6: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de enviar correo, entonces, se muestra un mensaje de error indicando que no se ha podido enviar el correo de recuperación y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

Segunda fase: Reinicio de contraseña.

• **Escenario 1: Reinicio de contraseña válido.**

Dado un código de recuperación registrado y una nueva contraseña válida, el usuario pulsa el botón de reiniciar contraseña, entonces, se comprueba la validez del código en el servidor, se almacena la nueva contraseña para el usuario, se muestra un mensaje de éxito en la aplicación y se redirige al usuario a la pantalla de inicio de sesión.

• **Escenario 2: Código vacío.**

Dado un código de recuperación vacío, cuando el usuario pulsa el botón de reiniciar contraseña, entonces, se muestra un mensaje de error indicando que se encuentra vacío y se mantiene al usuario en la misma pantalla.

• **Escenario 3: Contraseña vacía.**

Dada una contraseña nueva vacía, cuando el usuario pulsa el botón de reiniciar contraseña, entonces, se muestra un mensaje de error indicando que la contraseña se encuentra vacía y

se mantiene al usuario en la misma pantalla.

- **Escenario 4: Código inválido.**

Dado un código de recuperación inválido, cuando el usuario pulsa el botón de reiniciar contraseña, entonces, se muestra un mensaje de error indicando que el código no cumple los requisitos y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 5: Contraseña inválida.**

Dada una contraseña nueva inválida, cuando el usuario pulsa el botón de reiniciar contraseña, entonces, se muestra un mensaje de error indicando que la contraseña no cumple los requisitos y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 6: Sin acceso a Internet.**

Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de reiniciar contraseña, entonces, se muestra un mensaje de error indicando que no se ha podido reiniciar la contraseña y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 7: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de reiniciar contraseña, entonces, se muestra un mensaje de error indicando que no se ha podido reiniciar la contraseña y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

Tabla 16: Recuperar contraseña.

Cambiar contraseña	
Identificador de historia: 5.	Sprint asignado: 1.
Riesgo de desarrollo: Medio.	Valor de la historia: 6/10.
Horas estimadas: 5,7 horas.	Horas reales: 5,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder cambiar la contraseña por una nueva propia, de forma que pueda iniciar sesión y hacer uso de la aplicación con la nueva contraseña.	
Criterios de aceptación: <ul style="list-style-type: none"> ● Escenario 1: Cambio de contraseña válido. Dada una contraseña nueva válida, cuando el usuario pulsa el botón de cambiar contraseña, entonces, se muestra un mensaje de éxito, se cambia la contraseña en el servidor comprobando la antigua y se redirige al usuario a la pantalla de inicio de sesión de la aplicación. ● Escenario 2: Contraseña de confirmación vacía. Dada una contraseña de confirmación vacía, cuando el usuario pulsa el botón de cambiar contraseña, entonces, se muestra un mensaje de error indicando que se encuentra vacía y se mantiene al usuario en la misma pantalla. ● Escenario 3: Contraseña nueva vacía. 	

Dada una contraseña nueva vacía, cuando el usuario pulsa el botón de cambiar contraseña, entonces, se muestra un mensaje de error indicando que se encuentra vacía y se mantiene al usuario en la misma pantalla.

- **Escenario 4: Contraseña de confirmación errónea.**

Dada una contraseña diferente a la del usuario que ha iniciado sesión, cuando el usuario pulsa el botón de cambiar contraseña, entonces, se muestra un mensaje de error indicando que la contraseña de confirmación es errónea y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 5: Contraseña nueva inválida.**

Dada una contraseña nueva inválida, cuando el usuario pulsa el botón de cambiar contraseña, entonces, se muestra un mensaje de error indicando que la contraseña nueva no cumple los requisitos y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 6: Sin acceso a Internet.**

Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de cambio contraseña, entonces, se muestra un mensaje de error indicando que no se ha podido cambiar la contraseña y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 7: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de cambiar contraseña, entonces, se muestra un mensaje de error indicando que no se ha podido cambiar la contraseña y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

Tabla 17: Cambiar contraseña.

Eliminar cuenta	
Identificador de historia: 6.	Sprint asignado: 1.
Riesgo de desarrollo: Medio.	Valor de la historia: 6/10.
Horas estimadas: 5,7 horas.	Horas reales: 4,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder eliminar mi cuenta, de forma que la aplicación y el servidor no dispongan mis datos.	
Criterios de aceptación:	
<ul style="list-style-type: none"> ● Escenario 1: Eliminación de cuenta válida. Cuando el usuario pulsa el botón de eliminar cuenta, entonces, se muestra un mensaje de éxito, se elimina la cuenta del servidor y la sesión en la aplicación, y se redirige al usuario a la pantalla de inicio de sesión de la aplicación. 	
<ul style="list-style-type: none"> ● Escenario 2: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de eliminar cuenta, entonces, se muestra un mensaje de error indicando que no se ha podido eliminar la cuenta y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. 	

- **Escenario 3: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de eliminar cuenta, entonces, se muestra un mensaje de error indicando que no se ha podido eliminar la cuenta y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

Tabla 18: Eliminar cuenta.

13.2.2. Perfiles

Ver mi perfil	
Identificador de historia: 7.	Sprint asignado: 3.
Riesgo de desarrollo: Bajo.	Valor de la historia: 7/10.
Horas estimadas: 5,0 horas.	Horas reales: 5,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver mi perfil con mi información personal, de forma que pueda ver mis datos personales privados y públicos.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Escenario 1: Obtención de perfil válido. Cuando el usuario pulsa el botón de ver perfil, entonces, se le redirige a la pantalla de perfil y se obtiene su información personal privada y pública del servidor la cual se muestra. 	
<ul style="list-style-type: none"> • Escenario 2: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de ver perfil, entonces, se le redirige a dicha pantalla y se muestra un mensaje de error indicando que no se ha podido obtener su información y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> • Escenario 3: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario pulsa el botón de perfil, entonces, se le redirige a dicha pantalla donde se muestra un mensaje de error indicando que no se ha podido obtener la información del usuario y se mantiene al usuario en la misma pantalla. 	

Tabla 19: Ver mi perfil.

Editar mi perfil	
Identificador de historia: 8.	Sprint asignado: 3.
Riesgo de desarrollo: Bajo.	Valor de la historia: 6/10.
Horas estimadas: 3,0 horas.	Horas reales: 3,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder modificar mi fotografía de perfil y mi descripción, de forma que pueda corregir o adaptar estos datos a mi gusto.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Escenario 1: Modificación de perfil válido. 	

Dado una fotografía de perfil y una descripción válidas, cuando el usuario pulsa el botón de guardar cambios, entonces, se muestra un mensaje de éxito, se cambian los datos antiguos por los nuevos en el servidor y se redirige al usuario a la pantalla de su perfil con los datos actualizados.

- **Escenario 2: Descripción vacía.**

Dada una descripción vacía, cuando el usuario pulsa el botón de guardar cambios, entonces, se muestra un mensaje de error indicando que se encuentra vacía y se mantiene al usuario en la misma pantalla.

- **Escenario 3: Sin acceso a Internet.**

Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de guardar cambios, entonces, se muestra un mensaje de error indicando que no se han podido guardar los cambios y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

- **Escenario 4: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de guardar cambios, entonces, se muestra un mensaje de error indicando que no se han podido guardar los cambios y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos.

Tabla 20: Editar mi perfil.

Ver perfil de un usuario	
Identificador de historia: 9.	Sprint asignado: 3.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,0 horas.	Horas reales: 5,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver el perfil de un usuario de la aplicación, de forma que pueda ver su información pública.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Escenario 1: Obtención de perfil de usuario válido. Cuando el usuario pulsa el botón de ver el perfil de otro usuario, entonces, se le redirige a la pantalla de perfil de dicho usuario y se obtiene la información del servidor. 	
<ul style="list-style-type: none"> • Escenario 2: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de ver el perfil de otro usuario, entonces, se le redirige a la pantalla del perfil de este y se muestra un mensaje de error indicando que no se ha podido obtener la información manteniéndolo en la misma pantalla. 	
<ul style="list-style-type: none"> • Escenario 3: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario pulsa el botón de ver el perfil de otro usuario, entonces, se le redirige a la pantalla del perfil de este y se muestra un mensaje de error indicando que no se ha podido obtener la información manteniéndolo en la misma pantalla. 	

Tabla 21: Ver perfil de un usuario.

13.2.3. Gestión de Anuncios

Crear anuncio	
Identificador de historia: 10.	Sprint asignado: 2.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 8,0 horas.	Horas reales: 8,0 horas.
<p>Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder crear anuncios de para pedir ayuda o ofrecerla, de forma que puedan ser publicados en la aplicación y otros usuarios puedan verlos.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> ● Escenario 1: Anuncio creado. Dados un título, una descripción, una localización, una recompensa si es un anuncio para pedir ayuda o una distancia si es un anuncio de ofrecerla, cuando el usuario pulsa el botón de crear anuncio, entonces, se comprueban los datos, se envían al servidor y se almacenan mostrando un mensaje de éxito al usuario. ● Escenario 2: Título vacío. Dado un título vacío, cuando el usuario pulsa el botón de crear anuncio, entonces, se muestra un mensaje de error indicando que el título no puede estar vacío y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. ● Escenario 3: Descripción vacía. Dada una descripción vacía, cuando el usuario pulsa el botón de crear anuncio, entonces, se muestra un mensaje de error indicando que la descripción no puede estar vacía y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. ● Escenario 4: Sin recompensa. Dado que el usuario quiere crear un anuncio para pedir ayuda y una recompensa vacía, cuando el usuario pulsa el botón de crear anuncio, entonces, se muestra un mensaje de error indicando que la recompensa no puede estar vacía y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. ● Escenario 5: Sin distancia. Dado que el usuario quiere crear un anuncio ofreciendo su ayuda y una distancia vacía, cuando el usuario pulsa el botón de crear anuncio, entonces, se muestra un mensaje de error indicando que la distancia no puede estar vacía y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. ● Escenario 6: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de crear anuncio, entonces, se muestra un mensaje de error indicando que no se ha podido crear el anuncio y se mantiene al usuario en la misma pantalla. 	

- **Escenario 7: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de crear anuncio, entonces, se muestra un mensaje de error indicando que no se ha podido crear el anuncio y se mantiene al usuario en la misma pantalla.

Tabla 22: Crear anuncio.

Editar anuncio	
Identificador de historia: 11.	Sprint asignado: 2.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,0 horas.	Horas reales: 5,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder editar mis anuncios donde pido ayuda o la ofrezco, de forma que tengan la información actualizada.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Escenario 1: Anuncio editado. Dados un título, una descripción, una localización, una recompensa si es un anuncio para pedir ayuda o una distancia si es un anuncio para ofrecerla, cuando el usuario pulsa el botón de guardar cambios anuncio, entonces, se comprueban los datos, se envían al servidor y se almacenan sustituyendo los anteriores y se muestra un mensaje de éxito al usuario. 	
<ul style="list-style-type: none"> • Escenario 2: Título vacío. Dado un título vacío, cuando el usuario pulsa el botón de guardar cambios, entonces, se muestra un mensaje de error indicando que el título no puede estar vacío y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. 	
<ul style="list-style-type: none"> • Escenario 3: Descripción vacía. Dada una descripción vacía, cuando el usuario pulsa el botón de guardar cambios, entonces, se muestra un mensaje de error indicando que la descripción no puede estar vacía y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. 	
<ul style="list-style-type: none"> • Escenario 4: Sin recompensa. Dado que el usuario quiere editar un anuncio para pedir ayuda y una recompensa vacía, cuando el usuario pulsa el botón de guardar cambios, entonces, se muestra un mensaje de error indicando que la recompensa no puede estar vacía y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. 	
<ul style="list-style-type: none"> • Escenario 5: Sin distancia. Dado que el usuario quiere editar un anuncio donde ofrece su ayuda y una distancia vacía, cuando el usuario pulsa el botón de guardar cambios, entonces, se muestra un mensaje de error indicando que la distancia no puede estar vacía y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. 	
<ul style="list-style-type: none"> • Escenario 6: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de guardar cambios, 	

entonces, se muestra un mensaje de error indicando que no se han podido guardar los cambios y se mantiene al usuario en la misma pantalla.

- **Escenario 7: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de guardar cambios, entonces, se muestra un mensaje de error indicando que no se han podido guardar los cambios y se mantiene al usuario en la misma pantalla.

Tabla 23: Editar anuncio.

Cerrar anuncio	
Identificador de historia: 12.	Sprint asignado: 2.
Riesgo de desarrollo: Medio.	Valor de la historia: 6/10.
Horas estimadas: 4,0 horas.	Horas reales: 5,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder cerrar mis anuncios donde pido ayuda o la ofrezco, de forma que estos ya no sean visibles para el resto de usuarios.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Escenario 1: Anuncio cerrado. Dado un anuncio del usuario abierto, cuando el usuario pulsa el botón de cerrar anuncio, entonces, todos los datos relacionados con el anuncio se eliminan del servidor y se muestra un mensaje de éxito. 	
<ul style="list-style-type: none"> • Escenario 2: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de cerrar anuncio y este es suyo, entonces, se muestra un mensaje de error indicando que no se ha podido cerrar y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> • Escenario 3: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario pulsa el botón de cerrar anuncio y este es suyo, entonces, se muestra un mensaje de error indicando que no se ha podido cerrar y se mantiene al usuario en la misma pantalla. 	

Tabla 24: Cerrar anuncio.

Guardar anuncio	
Identificador de historia: 13.	Sprint asignado: 2.
Riesgo de desarrollo: Medio.	Valor de la historia: 6/10.
Horas estimadas: 5,0 horas.	Horas reales: 5,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder guardar anuncios donde se pide ayuda o se ofrece de otros usuarios de forma que pueda consultarlos más tarde.	

Criterios de aceptación:**• Escenario 1: Anuncio guardado.**

Dado un anuncio que el usuario no ha guardado, cuando el usuario pulsa el botón de guardar anuncio, entonces, se muestra un mensaje de éxito y se almacena en el servidor los datos necesarios.

• Escenario 2: Sin acceso a Internet.

Dado que no hay acceso a Internet, cuando el usuario pulsa el botón de guardar en un anuncio que no es suyo, entonces, se muestra un mensaje de error indicando que no se ha podido guardar el anuncio manteniendo al usuario en la misma pantalla.

• Escenario 3: Sin acceso al servidor

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de guardar en un anuncio que no es suyo, entonces, se muestra un mensaje de error indicando que no se ha podido guardar el anuncio manteniendo al usuario en la misma pantalla.

Tabla 25: Guardar anuncio.

Eliminar anuncio guardado	
Identificador de historia: 14.	Sprint asignado: 2.
Riesgo de desarrollo: Medio.	Valor de la historia: 6/10.
Horas estimadas: 4,0 horas.	Horas reales: 5,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder eliminar los anuncios que he guardado de forma que ya no estén en mi lista de anuncios guardados.	
Criterios de aceptación:	
• Escenario 1: Anuncio guardado eliminado.	
Dado un anuncio guardado por el usuario, cuando el usuario pulsa el botón de eliminar anuncio guardado, entonces, se muestra un mensaje de éxito y se elimina del servidor los datos necesarios.	
• Escenario 2: Sin acceso a Internet.	
Dado que no hay acceso a Internet, cuando el usuario pulsa el botón de eliminar anuncio guardado, entonces, se muestra un mensaje de error indicando que no se ha podido eliminar el anuncio de la lista de anuncios guardados y se mantiene al usuario en la misma pantalla.	
• Escenario 3: Sin acceso al servidor.	
Dado que el servidor no está disponible, cuando el usuario pulsa el botón de eliminar anuncio guardado, entonces, se muestra un mensaje de error indicando que no se ha podido eliminar el anuncio de la lista de anuncios guardados y se mantiene al usuario en la misma pantalla.	

Tabla 26: Eliminar anuncio guardado.

13.2.4. Listas de Anuncios

Ver lista de anuncios de se necesita ayuda	
Identificador de historia: 15.	Sprint asignado: 2.
Riesgo de desarrollo: Medio.	Valor de la historia: 8/10.
Horas estimadas: 6,5 horas.	Horas reales: 7,5 horas.
<p>Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver los anuncios donde se pide ayuda publicados por otros usuarios de la aplicación ordenados de más a menos cerca de mi actual posición, de más nuevos a más antiguos o de más antiguos a más nuevos, de forma que pueda acceder a su información.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> ● Escenario 1: Obtención de anuncios válido. Dada una lista de anuncios donde se pide ayuda no vacía ordenadas según un filtro, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de “se necesita”, entonces, se obtiene dicha lista de anuncios publicados por otros usuarios y se muestra. ● Escenario 2: Lista de anuncios vacía. Dada una lista de anuncios de donde se pide ayuda vacía, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de “se necesita”, entonces, se muestra un mensaje indicando que no hay anuncios de dicho tipo actualmente y se mantiene al usuario en la misma pantalla. ● Escenario 3: Obtención de anuncios por distancia sin ubicación del usuario. Dado que la localización del usuario no se ha determinado, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de “se necesita”, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo su localización y se mantiene al usuario en la misma pantalla. ● Escenario 4: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de “se necesita”, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla. ● Escenario 5: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de “se necesita”, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla. 	

Tabla 27: Ver lista de anuncios donde se pide ayuda.

Ver lista de anuncios de se ofrece ayuda	
Identificador de historia: 16.	Sprint asignado: 2.

Riesgo de desarrollo: Medio.	Valor de la historia: 8/10.
Horas estimadas: 6,5 horas.	Horas reales: 7,5 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver los anuncios donde se ofrece ayuda publicados por otros usuarios de la aplicación ordenados de más a menos cerca de mi actual posición, de más nuevos a más antiguos o de más antiguos a más nuevos, de forma que pueda acceder a su información.	
Criterios de aceptación:	
<ul style="list-style-type: none"> ● Escenario 1: Obtención de anuncios válido. Dada una lista de anuncios donde se ofrece ayuda no vacía ordenadas según un filtro, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de “se ofrece”, entonces, se obtiene dicha lista de anuncios publicados por otros usuarios y se muestra. 	
<ul style="list-style-type: none"> ● Escenario 2: Lista de anuncios vacía. Dada una lista de anuncios de donde se ofrece ayuda vacía, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de “se ofrece”, entonces, se muestra un mensaje indicando que no hay anuncios de dicho tipo actualmente y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> ● Escenario 3: Obtención de anuncios por distancia sin ubicación del usuario. Dado que la localización del usuario no se ha determinado, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de “se ofrece”, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo su localización y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> ● Escenario 4: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de “se ofrece”, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> ● Escenario 5: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de “se ofrece”, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla. 	

Tabla 28: Ver lista de anuncios de se ofrece ayuda.

Ver lista de anuncios donde pido ayuda	
Identificador de historia: 17.	Sprint asignado: 2.
Riesgo de desarrollo: Medio.	Valor de la historia: 8/10.
Horas estimadas: 6,5 horas.	Horas reales: 6,5 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver los anuncios donde pido ayuda publicados por mi ordenados de más a menos cerca de mi actual posición, de más nuevos a más antiguos o de más antiguos a más nuevos, de forma que pueda acceder a su información.	

más nuevos a más antiguos o de más antiguos a más nuevos, de forma que pueda acceder a su información.

Criterios de aceptación:

- Escenario 1: Obtención de anuncios válido.**

Dada una lista de anuncios donde se pide ayuda del usuario no vacía, cuando el usuario accede a la pantalla de mis anuncios en la pestaña de “necesito”, entonces, se muestran los diferentes anuncios publicados por él.

- Escenario 2: Lista de anuncios vacía.**

Dada una lista de anuncios donde se pide ayuda del usuario vacía, cuando el usuario accede a la pantalla de la aplicación de mis anuncios en la pestaña de “necesito”, entonces, se muestra un mensaje indicando que no tiene anuncios de dicho tipo actualmente y se mantiene al usuario en la misma pantalla.

- Escenario 3: Sin acceso a Internet.**

Dado que no hay conexión a Internet, cuando el usuario accede a la pantalla de la aplicación de mis anuncios en la pestaña de “necesito”, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla.

- Escenario 4: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario accede a la pantalla de la aplicación de mis anuncios en la pestaña de “necesito”, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla.

Tabla 29: Ver lista de anuncios donde pido ayuda.

Ver lista de anuncios donde ofrezco mi ayuda	
Identificador de historia: 18.	Sprint asignado: 2.
Riesgo de desarrollo: Medio.	Valor de la historia: 8/10.
Horas estimadas: 6,5 horas.	Horas reales: 6,5 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver los anuncios donde ofrezco mi ayuda publicados por mí ordenados de más a menos cerca de mi actual posición, de más nuevos a más antiguos o de más antiguos a más nuevos, de forma que pueda acceder a su información.	
Criterios de aceptación:	
<ul style="list-style-type: none"> Escenario 1: Obtención de anuncios válido. 	
Dada una lista de anuncios donde se ofrece la ayuda del usuario no vacía, cuando el usuario accede a la pantalla de mis anuncios en la pestaña de “ofreces”, entonces, se muestran los diferentes anuncios publicados por él.	
<ul style="list-style-type: none"> Escenario 2: Lista de anuncios vacía. 	

Dada una lista de anuncios donde se ofrece la ayuda del usuario vacía, cuando el usuario accede a la pantalla de la aplicación de mis anuncios en la pestaña de “ofreces”, entonces, se muestra un mensaje indicando que no tiene anuncios de dicho tipo actualmente y se mantiene al usuario en la misma pantalla.

- **Escenario 3: Sin acceso a Internet.**

Dado que no hay conexión a Internet, cuando el usuario accede a la pantalla de la aplicación de mis anuncios en la pestaña de “ofreces”, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla.

- **Escenario 4: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario accede a la pantalla de la aplicación de mis anuncios en la pestaña de “ofreces”, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla.

Tabla 30: Ver lista anuncios donde ofrezco mi ayuda.

Ver lista de anuncios de personas que sigo	
Identificador de historia: 19.	Sprint asignado: 3.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,5 horas.	Horas reales: 5,5 horas.
<p>Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver los anuncios publicados por los usuarios que sigo ordenados de más a menos cerca de mi actual posición, de más nuevos a más antiguos o de más antiguos a más nuevos, de forma que pueda acceder a su información.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Escenario 1: Obtención de anuncios válido. Dada una lista de anuncios de cualquier tipo no vacía ordenadas según un filtro, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de los usuarios que sigue, entonces, se obtiene dicha lista de anuncios publicados y se muestra. • Escenario 2: Lista de anuncios vacía. Dada una lista de anuncios vacía, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de los usuarios que sigue, entonces, se muestra un mensaje indicando que no hay anuncios actualmente y se mantiene al usuario en la misma pantalla. • Escenario 3: Obtención de anuncios por distancia sin ubicación del usuario. Dado que la localización del usuario no se ha determinado, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de los usuarios que sigue, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo su localización y se mantiene al usuario en la misma pantalla. 	

- Escenario 4: Sin acceso a Internet.**

Dado que no hay conexión a Internet, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de los usuarios que sigue, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla.

- Escenario 5: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario accede a la pantalla inicial de la aplicación en la pestaña de anuncios de los usuarios que sigue, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla.

Tabla 31: Ver lista de anuncios de personas que sigo.

Ver lista de anuncios guardados	
Identificador de historia: 20.	Sprint asignado: 2.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,5 horas.	Horas reales: 5,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver los anuncios que he guardado ordenados de más a menos cerca de mi actual posición, de más nuevos a más antiguos o de más antiguos a más nuevos, de forma que pueda acceder a su información.	
Criterios de aceptación:	
<ul style="list-style-type: none"> Escenario 1: Obtención de anuncios válido. Dada una lista de anuncios guardados por el usuario no vacía ordenadas según un filtro, cuando el usuario accede a la pantalla de mis anuncios de la aplicación en la pestaña de anuncios guardados, entonces, se obtiene dicha lista de anuncios y se muestra. 	
<ul style="list-style-type: none"> Escenario 2: Lista de anuncios vacía. Dada una lista de anuncios guardados por el usuario vacía, cuando el usuario accede a la pantalla de mis anuncios de la aplicación en la pestaña de anuncios guardados, entonces, se muestra un mensaje indicando que no hay anuncios guardados y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> Escenario 3: Obtención de anuncios por distancia sin ubicación del usuario. Dado que la localización del usuario no se ha determinado, cuando el usuario accede a la pantalla de mis anuncios de la aplicación en la pestaña de anuncios guardados, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo su localización y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> Escenario 4: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario accede a la pantalla de mis anuncios de la aplicación en la pestaña de anuncios guardados, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla. 	

- **Escenario 5: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario accede a la pantalla de mis anuncios de la aplicación en la pestaña de anuncios guardados, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla.

Tabla 32: Ver lista de anuncios guardados.

Buscar anuncios	
Identificador de historia: 21.	Sprint asignado: 2.
Riesgo de desarrollo: Medio.	Valor de la historia: 8/10.
Horas estimadas: 5,5 horas.	Horas reales: 10,0 horas.
<p>Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder buscar anuncios que no son míos a través de un filtro de búsqueda por tipo de anuncio y ordenación, de forma que pueda acceder a la información de los anuncios.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Escenario 1: Búsqueda válida. Dado un texto de búsqueda no vacío, cuando el usuario pulsa el botón de buscar, entonces, se muestra una lista de los anuncios que cumplen los filtros de búsquedas seleccionados. • Escenario 2: Búsqueda vacía. Dado un texto de búsqueda vacío, cuando el usuario pulsa el botón de buscar, entonces, no se produce ninguna búsqueda. • Escenario 3: Búsqueda válida y lista de anuncios vacía. Dado un texto de búsqueda válido y una lista de anuncios vacía, cuando el usuario pulsa el botón de buscar, entonces, se muestra un mensaje indicando que ningún anuncio contiene el texto buscado. • Escenario 4: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de buscar, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la lista de anuncios y se mantiene al usuario en la misma pantalla. • Escenario 5: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario pulsa el botón de buscar, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la lista de anuncios y se mantiene al usuario en la misma pantalla. 	

Tabla 33: Buscar anuncios.

13.2.5. Valoraciones

Valorar a un usuario	
Identificador de historia: 22.	Sprint asignado: 3.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,0 horas.	Horas reales: 6,0 horas.
<p>Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder valorar a un usuario que me ha ayudado, de forma que él y otros usuarios puedan ver la opinión y tomar decisiones respecto a esta.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> Escenario 1: Valoración válida. Dada una valoración y una puntuación del 1 al 5 válidas, cuando el usuario pulsa el botón de enviar valoración, entonces, se muestra un mensaje de éxito, se envían los datos al servidor donde se almacenan y se redirige al usuario a la pantalla principal de la aplicación. Escenario 2: Valoración vacía. Dada una valoración vacía, cuando el usuario pulsa el botón de enviar valoración, entonces, se muestra un mensaje de error indicando que la valoración no puede estar vacía y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. Escenario 3: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de enviar valoración, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema creando la valoración y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. Escenario 4: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario pulsa el botón de enviar valoración, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema creando la valoración y se mantiene al usuario en la misma pantalla sin alterar los datos introducidos. 	

Tabla 34: Valorar a un usuario.

Ver valoraciones dirigidas a mí	
Identificador de historia: 23.	Sprint asignado: 3.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,0 horas.	Horas reales: 5,5 horas.
<p>Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver las valoraciones que otros usuarios han hecho de la ayuda que les otorgué, de forma que pueda mejorar y ser consciente de la opinión de los otros usuarios de mi.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> Escenario 1: Obtención de valoraciones válida. 	

Dado que el usuario ha recibido valoraciones, cuando el usuario accede al apartado de valoraciones de su perfil, entonces, se muestra la lista.
• Escenario 2: No hay valoraciones.
Dado que el usuario no ha recibido valoraciones, cuando el usuario accede al apartado de valoraciones de su perfil, entonces, se muestra un mensaje indicando que no tiene valoraciones.
• Escenario 3: Sin acceso a Internet.
Dado que no hay conexión a Internet, cuando el usuario accede al apartado de valoraciones de su perfil, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla.
• Escenario 4: Sin acceso al servidor.
Dado que el servidor no está disponible, cuando el usuario accede al apartado de valoraciones de su perfil, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla.

Tabla 35: Ver valoraciones dirigidas a mi.

Ver valoraciones dirigidas a un usuario	
Identificador de historia: 24.	Sprint asignado: 3.
Riesgo de desarrollo: Medio.	Valor de la historia: 8/10.
Horas estimadas: 5,0 horas.	Horas reales: 5,5 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver las valoraciones que tienen otros usuarios de forma que pueda decidir que usuario escoger para que me ayude.	
Criterios de aceptación:	
• Escenario 1: Obtención de valoraciones válida.	
Cuando el usuario accede al apartado de valoraciones del perfil de otro usuario, entonces, se obtienen las valoraciones que ha recibido el otro usuario y se muestran.	
• Escenario 2: No hay valoraciones.	
Dado un usuario sin valoraciones, cuando el usuario accede al apartado de valoraciones del perfil del usuario que no ha recibido valoraciones, entonces, se muestra un mensaje indicando que dicho usuario no tiene valoraciones.	
• Escenario 3: Usuario ha dejado de existir.	
Cuando el usuario accede al apartado de valoraciones del perfil de un usuario que ha dejado de existir, entonces, se muestra un mensaje de error indicando que ya no existe y se redirige al usuario a la pantalla anterior.	
• Escenario 4: Sin acceso a Internet.	
Dado que no hay conexión a Internet, cuando el usuario accede al apartado de valoraciones	

del perfil de otro usuario, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla.

- **Escenario 5: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario accede al apartado de valoraciones del perfil de otro usuario, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema obteniendo dicha información y se mantiene al usuario en la misma pantalla.

Tabla 36: Ver valoraciones dirigidas a un usuario.

13.2.6. Seguimiento de Actividad

Seguir a un usuario	
Identificador de historia: 25.	Sprint asignado: 3.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,0 horas.	Horas reales: 5,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder seguir la actividad de otros usuarios de forma que pueda estar al tanto de los anuncios que publiquen.	
Criterios de aceptación:	
<ul style="list-style-type: none"> ● Escenario 1: Éxito al seguir. Cuando el usuario presiona el botón de seguir en el perfil de un usuario que no sigue, entonces se crea una entrada nueva en el servidor que indica el nuevo seguimiento y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> ● Escenario 2: Usuario ha dejado de existir. Cuando el usuario pulsa el botón de seguir en el perfil de un usuario que ha dejado de existir, entonces, se muestra un mensaje de error indicando que ya no existe y se redirige al usuario a la pantalla anterior. 	
<ul style="list-style-type: none"> ● Escenario 3: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario presiona el botón de seguir en el perfil de un usuario que no sigue, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema procesando la solicitud y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> ● Escenario 4: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario presiona el botón de seguir en el perfil de un usuario que no sigue, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema procesando la solicitud y se mantiene al usuario en la misma pantalla. 	

Tabla 37: Seguir a un usuario.

Dejar de seguir a un usuario	
Identificador de historia: 26.	Sprint asignado: 3.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,0 horas.	Horas reales: 4,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder dejar de seguir a un usuario al que sigo, de forma que ya no esté al tanto de las publicaciones que publique.	
Criterios de aceptación:	
<ul style="list-style-type: none"> Escenario 1: Éxito al dejar de seguir. Cuando el usuario pulsa el botón de dejar de seguir en el perfil del usuario que quiere dejar de seguir, entonces, se elimina la información necesaria correspondiente al seguimiento que existía del servidor y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> Escenario 2: Usuario ha dejado de existir. Cuando el usuario pulsa el botón de dejar de seguir en el perfil de un usuario que ha dejado de existir, entonces, se muestra un mensaje de error indicando que ya no existe y se redirige al usuario a la pantalla anterior. 	
<ul style="list-style-type: none"> Escenario 3: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario presiona el botón de dejar de seguir en el perfil de un usuario que sigue, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema procesando la solicitud y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> Escenario 4: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario presiona el botón de dejar de seguir en el perfil de un usuario que sigue, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema procesando la solicitud y se mantiene al usuario en la misma pantalla. 	

Tabla 38: Dejar de seguir a un usuario.

Ver los usuarios que me siguen	
Identificador de historia: 27.	Sprint asignado: 3.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,5 horas.	Horas reales: 5,5 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver los usuarios que me siguen en ese momento, de forma que pueda ver sus perfiles o realizar otras acciones como seguir su actividad.	
Criterios de aceptación:	
<ul style="list-style-type: none"> Escenario 1: Obtención de usuarios válido. Dada una lista no vacía de usuarios que siguen al usuario, cuando este pulsa el botón de ver 	

los usuarios que le siguen en la pantalla de su perfil, entonces se obtienen dichos usuarios y se muestran.

- **Escenario 2: No hay usuarios.**

Dada una lista de usuarios vacía que siguen al usuario, cuando este pulsa el botón de ver los usuarios que le siguen en la pantalla de su perfil, entonces se muestra un mensaje indicando que no hay usuarios.

- **Escenario 3: Sin acceso a Internet.**

Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de ver los usuarios que le siguen en la pantalla de su perfil, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la lista de usuarios y se mantiene al usuario en la misma pantalla.

- **Escenario 4: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de ver los usuarios que le siguen en la pantalla de su perfil, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la lista de usuarios y se mantiene al usuario en la misma pantalla.

Tabla 39: Ver los usuarios que me siguen.

Ver los usuarios que sigo	
Identificador de historia: 28.	Sprint asignado: 3.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,5 horas.	Horas reales: 5,5 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder ver los usuarios a los que sigo, de forma que pueda ver sus perfiles o realizar otras acciones como dejar de seguir su actividad.	
Criterios de aceptación:	
<ul style="list-style-type: none"> ● Escenario 1: Obtención de usuarios válido. <p>Dada una lista no vacía de usuarios que el usuario sigue, cuando este pulsa el botón de ver los usuarios que sigue en la pantalla de su perfil, entonces se obtienen dichos usuarios y se muestran.</p>	
<ul style="list-style-type: none"> ● Escenario 2: No hay usuarios. <p>Dada una lista de usuarios vacía que sigue el usuario, cuando este pulsa el botón de ver los usuarios que sigue en la pantalla de su perfil, entonces se muestra un mensaje indicando que no hay usuarios.</p>	
<ul style="list-style-type: none"> ● Escenario 3: Sin acceso a Internet. <p>Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de ver los usuarios que sigue en la pantalla de su perfil, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la lista de usuarios y se mantiene al usuario en la misma pantalla.</p>	

- **Escenario 4: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de ver los usuarios que sigue en la pantalla de su perfil, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la lista de usuarios y se mantiene al usuario en la misma pantalla.

Tabla 40: Ver los usuarios que sigo.

Buscar a un usuario que sigo	
Identificador de historia: 29.	Sprint asignado: 3.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,0 horas.	Horas reales: 4,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder buscar a un usuario que sigo, de forma que pueda ver su perfil o realizar otras acciones como dejar de seguir su actividad.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Escenario 1: Búsqueda válida. Dado un texto de búsqueda no vacío, cuando el usuario pulsa el botón de buscar, entonces, se muestra una lista de los usuarios que sigue que contienen dicho texto en su nombre de usuario. 	
<ul style="list-style-type: none"> • Escenario 2: Búsqueda vacía. Dado un texto de búsqueda vacío, cuando el usuario pulsa el botón de buscar, entonces, no se produce ninguna búsqueda. 	
<ul style="list-style-type: none"> • Escenario 3: Búsqueda válida y lista de usuarios vacía. Dado un texto de búsqueda válido y una lista de usuarios vacía, cuando el usuario pulsa el botón de buscar, entonces, se muestra un mensaje indicando que ningún usuario de los que sigue contiene el texto en su nombre de usuario. 	
<ul style="list-style-type: none"> • Escenario 4: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de buscar, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la lista de usuarios y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> • Escenario 5: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario pulsa el botón de buscar, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la lista de usuarios y se mantiene al usuario en la misma pantalla. 	

Tabla 41: Buscar a un usuario que sigo.

Buscar a un usuario que me sigue	
Identificador de historia: 30.	Sprint asignado: 3.
Riesgo de desarrollo: Medio.	Valor de la historia: 7/10.
Horas estimadas: 5,0 horas.	Horas reales: 4,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder buscar a un usuario que me sigue, de forma que pueda ver su perfil o realizar otras acciones como seguir su actividad.	
Criterios de aceptación:	
<ul style="list-style-type: none"> ● Escenario 1: Búsqueda válida. Dado un texto de búsqueda no vacío, cuando el usuario pulsa el botón de buscar, entonces, se muestra una lista de los usuarios que le siguen que contienen dicho texto en su nombre de usuario. 	
<ul style="list-style-type: none"> ● Escenario 2: Búsqueda vacía. Dado un texto de búsqueda vacío, cuando el usuario pulsa el botón de buscar, entonces, no se produce ninguna búsqueda. 	
<ul style="list-style-type: none"> ● Escenario 3: Búsqueda válida y lista de usuarios vacía. Dado un texto de búsqueda válido y una lista de usuarios vacía, cuando el usuario pulsa el botón de buscar, entonces, se muestra un mensaje indicando que ningún usuario de los que le sigue contiene el texto en su nombre de usuario. 	
<ul style="list-style-type: none"> ● Escenario 4: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de buscar, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la lista de usuarios y se mantiene al usuario en la misma pantalla. 	
<ul style="list-style-type: none"> ● Escenario 5: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario pulsa el botón de buscar, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la lista de usuarios y se mantiene al usuario en la misma pantalla. 	

Tabla 42: Buscar a un usuario que me sigue.

13.2.7. Conversaciones

Abrir conversación	
Identificador de historia: 31.	Sprint asignado: 4.
Riesgo de desarrollo: Alto.	Valor de la historia: 9/10.
Horas estimadas: 8,0 horas.	Horas reales: 8,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder abrir una conversación con otro usuario respecto a un anuncio publicado, de forma que pueda ayudarle o que el pueda ayudarme.	

Criterios de aceptación:**• Escenario 1: Conversación abierta.**

Cuando el usuario pulsa el botón de conversar dentro de la pantalla de un anuncio y no existe una conversación con los usuarios asociada con el anuncio, entonces, en el servidor crea una nueva conversación con el anuncio correspondiente, los usuarios que participan y el estado de la conversación como por ejemplo si han visto los usuarios los últimos mensajes o si han cerrado la conversación.

• Escenario 2: Anuncio ha dejado de existir.

Dado un anuncio que ha dejado de existir, cuando el usuario pulsa el botón de conversar dentro de la pantalla de un anuncio, entonces, se muestra un error indicando que el anuncio ha dejado de existir y no se puede abrir una conversación y se redirige al usuario a la pantalla anterior.

• Escenario 3: Sin acceso a Internet.

Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de conversar dentro de la pantalla de un anuncio, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando abrir una conversación y se mantiene al usuario en la misma pantalla.

• Escenario 4: Sin acceso al servidor.

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de conversar dentro de la pantalla de un anuncio, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando abrir una conversación y se mantiene al usuario en la misma pantalla.

Tabla 43: Abrir conversación.

Enviar mensajes	
Identificador de historia: 32.	Sprint asignado: 4.
Riesgo de desarrollo: Alto.	Valor de la historia: 9/10.
Horas estimadas: 13,0 horas.	Horas reales: 10,5 horas.

Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder conversar con otro usuario enviándole mensajes, de forma que pueda ayudarle o que el pueda ayudarme.

Criterios de aceptación:**• Escenario 1: Envío de mensajes válido.**

Dada una conversación abierta con un usuario y un mensaje válido, cuando el usuario pulsa el botón de enviar mensaje, entonces, se envía al servidor el mensaje, se actualizan los datos de estado de la conversación, y se transmite el mensaje al servidor de Firebase para que este comunique al receptor de su existencia

• Escenario 2: Conversación ha dejado de existir.

Dada una conversación que ha dejado de existir, cuando el usuario pulsa el botón de enviar

mensaje, entonces este no se transmite y se mantiene al usuario en la misma pantalla.

- **Escenario 3: Sin acceso a Internet.**

Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de enviar mensaje, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando enviar el mensaje y se mantiene al usuario en la misma pantalla.

- **Escenario 4: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario pulsa el botón de enviar mensaje, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando enviar el mensaje y se mantiene al usuario en la misma pantalla.

Tabla 44: Enviar mensajes.

Cerrar conversación	
Identificador de historia: 33.	Sprint asignado: 4.
Riesgo de desarrollo: Alto.	Valor de la historia: 9/10.
Horas estimadas: 7,0 horas.	Horas reales: 7,0 horas.
Descripción: Como usuario que ha iniciado sesión, quiero poder cerrar la conversación que tenga con otro usuario respecto a un anuncio, de forma que la vea en mi lista de conversaciones.	
Criterios de aceptación:	
● Escenario 1: Conversación cerrada.	
Dada una conversación con un usuario, cuando el usuario pulsa el botón de cerrar conversación en la pantalla de la lista de conversaciones, entonces, se actualiza en el servidor los datos correspondientes para que esta permanezca oculta hasta que el usuario vuelve a abrirla o el anuncio se elimine.	
● Escenario 2: Conversación ha dejado de existir.	
Dada una conversación que ha dejado de existir, cuando el usuario pulsa el botón de cerrar conversación, entonces se muestra un mensaje de error indicando que la conversación ya no existe y se mantiene al usuario en la pantalla de la lista de conversaciones.	
● Escenario 2: Sin acceso a Internet.	
Dado que no hay conexión a Internet, cuando el usuario pulsa el botón de cerrar conversación en la pantalla de la lista de conversaciones, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando cerrar la conversación y se mantiene al usuario en la misma pantalla.	
● Escenario 3: Sin acceso al servidor.	
Dado que el servidor no está disponible, cuando el usuario pulsa el botón de cerrar conversación en la pantalla de la lista de conversaciones, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando cerrar la conversación y se mantiene al usuario en la misma pantalla.	

Tabla 45: Cerrar conversación.

Ver mis conversaciones	
Identificador de historia: 34.	Sprint asignado: 4.
Riesgo de desarrollo: Alto.	Valor de la historia: 9/10.
Horas estimadas: 13,0 horas.	Horas reales: 11,5 horas.
<p>Descripción: Como usuario que ha iniciado sesión, quiero poder ver las conversaciones que estoy manteniendo con otros usuarios así como si han llegado nuevos mensajes, de forma que pueda acceder a ellas y conversar.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> Escenario 1: Obtención de conversaciones válida. Dado que el usuario tiene conversaciones abiertas, cuando el usuario accede a la pantalla de la lista de conversaciones, entonces, estas se muestran indicando el usuario con quien se habla, el título del anuncio asociado, el último mensaje y si hay nuevos mensajes. Escenario 2: No hay conversaciones. Dado que el usuario no tiene conversaciones abiertas, cuando el usuario accede a la pantalla de la lista de conversaciones, entonces, se muestra un error indicando que no hay conversaciones abiertas y se mantiene al usuario en la misma pantalla. Escenario 3: Sin acceso a Internet. Dado que no hay conexión a Internet, cuando el usuario accede a la pantalla de la lista de conversaciones, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando cerrar la conversación y se mantiene al usuario en la misma pantalla. Escenario 4: Sin acceso al servidor. Dado que el servidor no está disponible, cuando el usuario accede a la pantalla de la lista de conversaciones, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando cerrar la conversación y se mantiene al usuario en la misma pantalla. 	

Tabla 46: Ver mis conversaciones.

Ver conversación	
Identificador de historia: 35.	Sprint asignado: 4.
Riesgo de desarrollo: Alto.	Valor de la historia: 9/10.
Horas estimadas: 15,0 horas.	Horas reales: 13,0 horas.
<p>Descripción: Como usuario que ha iniciado sesión, quiero poder ver la conversación que tengo con otro usuario, de forma que pueda leer los mensajes recibidos y enviados.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> Escenario 1: Obtención de conversación válida. 	

Dado que el usuario tiene una conversación abierta, cuando el usuario accede a la pantalla de la lista de conversaciones y pulsa sobre una conversación, entonces, se muestran los mensajes intercambiados entre él y el otro usuario

- **Escenario 2: Sin acceso a Internet.**

Dado que no hay conexión a Internet, cuando el usuario accede a la pantalla de la lista de conversaciones y pulsa sobre una conversación, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la conversación y se redirige al usuario a la pantalla anterior.

- **Escenario 3: Sin acceso al servidor.**

Dado que el servidor no está disponible, cuando el usuario accede a la pantalla de la lista de conversaciones y pulsa sobre una conversación, entonces, se muestra un mensaje de error indicando que ha ocurrido un problema intentando obtener la conversación y se redirige al usuario a la pantalla anterior.

Tabla 47: Ver conversación.

Buscar conversación	
Identificador de historia: 36.	Sprint asignado: 4.
Riesgo de desarrollo: Medio.	Valor de la historia: 8/10.
Horas estimadas: 7,0 horas.	Horas reales: 7,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación, quiero poder buscar conversaciones que tenga abiertas, de forma que pueda acceder a dichas conversaciones.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Escenario 1: Búsqueda válida. Dado un texto de búsqueda no vacío, cuando el usuario pulsa el botón de buscar, entonces, se muestra una lista de las conversaciones que contienen el texto buscado. 	
<ul style="list-style-type: none"> • Escenario 2: Búsqueda vacía. Dado un texto de búsqueda vacío, cuando el usuario pulsa el botón de buscar, entonces, no se produce ninguna búsqueda. 	
<ul style="list-style-type: none"> • Escenario 3: Búsqueda válida y lista de anuncios vacía. Dado un texto de búsqueda válido y una lista de conversaciones vacía, cuando el usuario pulsa el botón de buscar, entonces, se muestra un mensaje indicando que no hay ninguna conversación que cumpla los requisitos. 	

Tabla 48: Buscar conversación.

13.2.8. Notificaciones

Recibir notificación de anuncio cercano	
Identificador de historia: 37.	Sprint asignado: 5.
Riesgo de desarrollo: Alto.	Valor de la historia: 8/10.
Horas estimadas: 9,0 horas.	Horas reales: 9,0 horas.
<p>Descripción: Como usuario que ha iniciado sesión en la aplicación quiero poder recibir notificaciones de anuncios donde se pide ayuda creados cerca de mi, de forma que pueda acceder a su información y ofrecer mi ayuda.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> Escenario 1: Notificación enviada y recibida. Dada una notificación recibida, cuando el usuario pulsa sobre esta tanto si la aplicación está suspendida o encendida, entonces, se le redirige a la pantalla del anuncio donde se muestra su información. Escenario 2: Anuncio ha dejado de existir. Dada una notificación recibida y el anuncio asociado ha dejado de existir, cuando el usuario pulsa sobre esta entonces, se muestra un error indicando que el anuncio ha dejado de existir y se redirige al usuario a la pantalla anterior si la aplicación estaba abierta o a la pantalla principal si estaba cerrada. Escenario 3: Sin acceso a Internet. Dado que no hay conexión a Internet, la notificación no es recibida por el usuario hasta el momento en que la conexión de este se restaura. Escenario 4: Servidor de la aplicación y Firebase no disponibles. Dado que el servidor de la aplicación o el servidor de Firebase no estén disponibles, la notificación no es recibida por el usuario en ningún momento. 	

Tabla 49: Recibir notificación de anuncio cercano.

Recibir notificación de valoración	
Identificador de historia: 38.	Sprint asignado: 5.
Riesgo de desarrollo: Alto.	Valor de la historia: 8/10.
Horas estimadas: 8,0 horas.	Horas reales: 7,0 horas.
<p>Descripción: Como usuario que ha iniciado sesión, quiero poder recibir notificaciones de las valoraciones que me hagan los usuarios a los que he ayudado, de forma que pueda acceder al contenido de la valoración de forma rápida.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> Escenario 1: Notificación enviada y recibida. Dada una notificación recibida, cuando el usuario pulsa sobre esta tanto si la aplicación está suspendida o encendida, entonces, se le redirige a la pantalla de su perfil más concretamente 	

a la pestaña de valoraciones donde se muestra su información.

- **Escenario 2: Sin acceso a Internet.**

Dado que no hay conexión a Internet, la notificación no es recibida por el usuario hasta el momento en que la conexión de este se restaura.

- **Escenario 3: Servidor de la aplicación y Firebase no disponibles.**

Dado que el servidor de la aplicación o el servidor de Firebase no estén disponibles, la notificación no es recibida por el usuario en ningún momento.

Tabla 50: Recibir notificación de valoración.

Recibir notificación de nuevo mensaje de conversación	
Identificador de historia: 39.	Sprint asignado: 5.
Riesgo de desarrollo: Alto.	Valor de la historia: 8/10.
Horas estimadas: 9,0 horas.	Horas reales: 10,0 horas.
Descripción: Como usuario que ha iniciado sesión, quiero poder recibir notificaciones cuando usuarios con los que mantengo una conversación me envíen mensajes, de forma que pueda acceder de forma rápida a las conversaciones y estar al tanto de estas.	
Criterios de aceptación:	
<ul style="list-style-type: none"> ● Escenario 1: Notificación enviada y recibida. Dada una notificación recibida, cuando el usuario pulsa sobre esta tanto si la aplicación está suspendida o encendida, entonces, se le redirige a la pantalla de la conversación donde se puede escribir nuevos mensajes y ver los anteriores. 	
<ul style="list-style-type: none"> ● Escenario 2: Conversación ha dejado de existir. Dada una notificación recibida y una conversación asociada a esta que ha dejado de existir, cuando el usuario pulsa sobre la notificación, entonces, se muestra un mensaje de error indicando que la conversación ha dejado de existir y se redirige al usuario a la pantalla anterior si la aplicación se encontraba abierta o a la pantalla principal si estaba cerrada. 	
<ul style="list-style-type: none"> ● Escenario 3: Sin acceso a Internet. Dado que no hay conexión a Internet, la notificación no es recibida por el usuario hasta el momento en que la conexión de este se restaura. 	
<ul style="list-style-type: none"> ● Escenario 4: Servidor de la aplicación y Firebase no disponibles. Dado que el servidor de la aplicación o el servidor de Firebase no estén disponibles, la notificación no es recibida por el usuario en ningún momento. 	

Tabla 51: Recibir notificación de nuevo mensaje de conversación.

Recibir notificación de nuevo seguidor	
Identificador de historia: 40.	Sprint asignado: 5.
Riesgo de desarrollo: Alto.	Valor de la historia: 8/10.

Horas estimadas: 8,0 horas.	Horas reales: 8,0 horas.
Descripción: Como usuario que ha iniciado sesión, quiero poder recibir notificaciones cuando los usuarios empiecen a seguir mi actividad, de forma que pueda acceder a sus perfiles de forma rápida.	
Criterios de aceptación:	
<ul style="list-style-type: none"> ● Escenario 1: Notificación enviada y recibida. Dada una notificación recibida, cuando el usuario pulsa sobre esta tanto si la aplicación está suspendida o encendida, entonces, se le redirige a la pantalla del perfil del usuario que ha comenzado a seguir su actividad. 	
<ul style="list-style-type: none"> ● Escenario 2: Usuario ha dejado de existir. Dada una notificación recibida y un usuario asociado a esta que ha dejado de existir, cuando el usuario pulsa la notificación, entonces, se muestra un mensaje de error indicando que ya no existe el usuario y se redirige al usuario a la pantalla anterior si la aplicación estaba abierta o a la pantalla principal si se encontraba cerrada. 	
<ul style="list-style-type: none"> ● Escenario 3: Sin acceso a Internet. Dado que no hay conexión a Internet, la notificación no es recibida por el usuario hasta el momento en que la conexión de este se restaura. 	
<ul style="list-style-type: none"> ● Escenario 4: Servidor de la aplicación y Firebase no disponibles. Dado que el servidor de la aplicación o el servidor de Firebase no estén disponibles, la notificación no es recibida por el usuario en ningún momento. 	

Tabla 52: Recibir notificación de nuevo seguidor.

Bloquear notificación de anuncio cercano	
Identificador de historia: 41.	Sprint asignado: 5.
Riesgo de desarrollo: Alto.	Valor de la historia: 8/10.
Horas estimadas: 7,0 horas.	Horas reales: 6,0 horas.
Descripción: Como usuario que ha iniciado sesión en la aplicación quiero poder dejar de recibir notificaciones de anuncios donde se pide ayuda creados cerca de mi bloqueándolas, de forma que deje de estar informado de estos.	
Criterios de aceptación:	
<ul style="list-style-type: none"> ● Escenario 1: Notificaciones bloqueadas. Dado un usuario en la pantalla de ajustes de notificaciones, cuando el usuario pulsa sobre el botón de dejar de recibir notificaciones de tipo anuncio, entonces, se guarda la información necesaria en el dispositivo móvil y ya no se reciben más notificaciones de este tipo. 	

Tabla 53: Bloquear notificación de anuncio cercano.

Bloquear notificación de valoración	
Identificador de historia: 42.	Sprint asignado: 5.
Riesgo de desarrollo: Alto.	Valor de la historia: 8/10.
Horas estimadas: 7,0 horas.	Horas reales: 6,0 horas.
Descripción: Como usuario que ha iniciado sesión, quiero poder dejar de recibir notificaciones de las valoraciones que me hagan los usuarios a los que he ayudado bloqueándolas, de forma que deje de estar informado de estas.	
Criterios de aceptación:	
<ul style="list-style-type: none"> Escenario 1: Notificaciones bloqueadas. Dado un usuario en la pantalla de ajustes de notificaciones, cuando el usuario pulsa sobre el botón de dejar de recibir notificaciones de tipo valoración, entonces, se guarda la información necesaria en el dispositivo móvil y ya no se reciben más notificaciones de este tipo. 	

Tabla 54: Bloquear notificación de valoración.

Bloquear notificación de nuevo mensaje de conversación	
Identificador de historia: 43.	Sprint asignado: 5.
Riesgo de desarrollo: Alto.	Valor de la historia: 8/10.
Horas estimadas: 7,0 horas.	Horas reales: 7,0 horas.
Descripción: Como usuario que ha iniciado sesión, quiero poder dejar de recibir notificaciones cuando los usuarios me envíen mensajes bloqueándolas, de forma que deje de estar informado de estos.	
Criterios de aceptación:	
<ul style="list-style-type: none"> Escenario 1: Notificaciones bloqueadas. Dado un usuario en la pantalla de ajustes de notificaciones, cuando el usuario pulsa sobre el botón de dejar de recibir notificaciones de tipo mensajes de conversación, entonces, se guarda la información necesaria en el dispositivo móvil y ya no se reciben más notificaciones de este tipo. 	

Tabla 55: Bloquear notificación de nuevo mensaje de conversación.

Bloquear notificación de nuevo seguidor	
Identificador de historia: 44.	Sprint asignado: 5.
Riesgo de desarrollo: Alto.	Valor de la historia: 8/10.
Horas estimadas: 7,0 horas.	Horas reales: 6,0 horas.
Descripción: Como usuario que ha iniciado sesión, quiero poder dejar de recibir notificaciones cuando los usuarios empiecen a seguir mi actividad bloqueándolas, de forma que deje de estar informado de estos.	

Criterios de aceptación:

- **Escenario 1: Notificaciones bloqueadas.**

Dado un usuario en la pantalla de ajustes de notificaciones, cuando el usuario pulsa sobre el botón de dejar de recibir notificaciones de tipo nuevo seguidor, entonces, se guarda la información necesaria en el dispositivo móvil y ya no se reciben más notificaciones de este tipo.

Tabla 56: Bloquear notificación de nuevo seguidor.

13.3. Requisitos No Funcionales

En este apartado se comentan los requisitos no funcionales relacionados con el proyecto. Se ha realizado una división por temas como puede ser la eficiencia, la usabilidad o la apariencia para que sea más fácil su lectura.

13.3.1. Diseño y Apariencia

Este grupo engloba aquellos requisitos que tienen que ver con como se ve la aplicación, los colores utilizados, el tamaño de letra utilizado etc.

- El texto debe de ser de como máximo 3 colores: negro, gris o blanco.
- La aplicación no debe exceder en el uso de colores en su interfaz. Como máximo debe de utilizar un total de 5 colores que jueguen con tonalidades similares y no mostrar más de 2 colores con tonalidades que contrasten bruscamente.
- Los colores utilizados por la aplicación deben de verse correctamente y no ser incómodos a la vista cuando se utilicen con texto, por lo tanto los colores deben de verse correctamente con los colores negro, gris y blanco.
- No debe haber mezclas de colores como gradientes u otras técnicas existentes.
- La aplicación no debe utilizar más de 4 tamaños distintos de letra y estos deben de ser correctamente visibles.
- La aplicación no debe de utilizar colores que puedan ocasionar confusión a las personas con daltonismo.

13.3.2. Usabilidad y Humanidad

Este grupo de requisitos tiene que ver con aquellos relacionados con la facilidad de uso de la aplicación, la internacionalización, el aprendizaje que debe tener el usuario o la accesibilidad.

- El tiempo de aprendizaje de la aplicación debe de ser como máximo 2 horas.
- El usuario debe de poder realizar cada funcionalidad con menos de 6 clics.
- La tasa de errores cometidos por el usuario debe de ser menor al 5% de las transacciones totales ejecutadas por el sistema.
- La aplicación debe de proporcionar mensajes de error que sean orientativos.
- La aplicación debe de proporcionar información acerca de la entrada de información en la aplicación por ejemplo el máximo número de caracteres de un campo.

- La aplicación debe de estar traducida en al menos 3 idiomas: castellano, catalán e inglés.

13.3.3. Rendimiento

Este grupo hace referencia a la eficiencia de la aplicación a la hora de realizar tareas, la precisión, la tolerancia a fallos etc.

- La aplicación, cuando se produce interacción con el servidor, debe de mostrar una respuesta al usuario en menos de 15 segundos.
- La aplicación debe de estar preparada para controlar gran cantidad de datos. Para ello debe de utilizar técnicas que permitan esto como el uso de la técnica de paginación.
- La aplicación debe de mostrar al usuario cuando se está realizando una tarea costosa a través de la interfaz con métodos como barras de progreso.

13.3.4. Seguridad y Privacidad

Este grupo hace referencia a las condiciones que debe de tener la aplicación para garantizar la seguridad de los datos y la información de los usuarios.

- El acceso a datos de la aplicación solo debe de estar disponible para usuarios registrados en la aplicación.
- El usuario registrado en la aplicación debe ser capaz de acceder a su información.
- La aplicación no debe de compartir los datos de sus usuarios con otros sistemas a menos que sea indispensable para el funcionamiento de la aplicación.
- La información de los usuarios no debe ser utilizada para fines ajenos a la aplicación.
- Los usuarios no deben ver datos privados de otros usuarios.

14. Diseño del sistema

En este apartado se pretende mostrar el diseño que ha seguido la aplicación. Primero se comenta el diseño del sistema en general, los diferentes sistemas que actúan y las comunicaciones entre estos. Posteriormente se describen los patrones de arquitectura utilizados tanto en la parte cliente como en la parte del servidor. Finalmente se muestran distintos diagramas UML de la parte cliente y del servidor para mostrar el comportamiento y un diagrama UML representativo de la base de datos.

14.1. Visión General

El sistema que se ha construido se compone de diferentes recursos que actúan entre sí para poder ofrecer a los usuarios las funcionalidades deseadas. El sistema está formado por una aplicación Android, un servidor de recursos de uso restringido únicamente a la aplicación construido a partir de una API Rest y una base de datos que se encuentra en el mismo servidor. Además también se compone de un servidor del servicio Cloudinary para almacenar las fotografías de los usuarios y un servidor del servicio Firebase para la mensajería instantánea y el envío de notificaciones.

Como se puede ver en la Figura 2, todas las solicitudes pasan directamente por el servidor de recursos. Esto puede resultar un problema en el caso de tener una cantidad de usuarios enorme ya que el servidor supondría un cuello de botella para las peticiones de los usuarios. No obstante, debido a que es necesaria una sincronización entre el contenido del servicio de Firebase, Cloudinary y la base de datos, se ha optado por este método. Esto permite que el servidor puede controlar si alguna petición a estos servicios ha fallado y realizar un *rollback* de los datos subidos a estos servicios o almacenados en la base de datos. Lo ideal sería tener un sistema distribuido con un sistema de *rollback* para mantener la coherencia entre el contenido de todos los servidores.

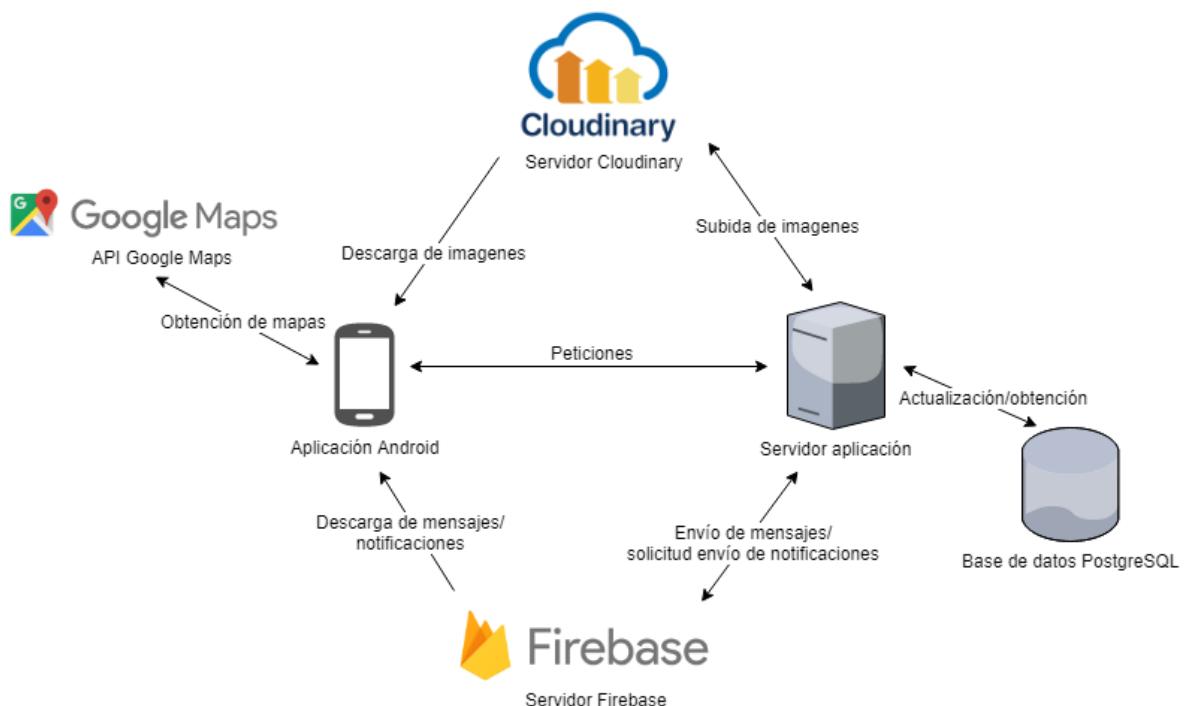


Figura 2: Visión general del sistema.

El servidor de recursos se comunica con el servicio de Cloudinary para subir las fotografías de los usuarios ya que no es una buena idea tener archivos pesados almacenados en una base de datos no destinada a ella. Cloudinary está preparado para ello y cumple todas las legislaciones europeas de protección de datos con lo que no debería de suponer ningún problema. En cuanto a Firebase es el encargado de transmitir las notificaciones que se producen en la aplicación a los distintos usuarios, también se encarga de almacenar los mensajes de las conversaciones entre los usuarios. Al igual que el servicio anterior, Firebase cumple todas las legislaciones europeas de protección de datos.

Gracias al uso de estos servicios, la aplicación de Android tan solo debe de obtener las imágenes de Cloudinary a través del vínculo URL almacenado en la base de datos del servidor de recursos único de cada usuario y debe de estar al tanto de las notificaciones de Firebase o a si se ha añadido un mensaje en algún chat si se encuentra en dicha pantalla gracias a su base de datos real. Más adelante se explicará la implementación del uso de estos servicios de forma más concreta.

14.2. Patrones de Arquitectura y Diseño

En este apartado se explican los patrones de diseño que se utilizan a lo largo de la implementación de la aplicación tanto en la parte *frontend* como en la parte *backend*. Algunos de estos se obliga su uso por el entorno donde se trabaja y mientras que otros se han implementado por uno mismo.

14.2.1. Modelo-Vista-Controlador (MVC)

El Modelo-Vista-Controlador [28] o también conocido como MVC de forma abreviada, es un patrón de arquitectura de código que separa la aplicación en tres capas distintas: el modelo, la vista y el controlador. Esta separación supone la separación de los datos y la lógica de negocio de las interfaces o pantallas de una aplicación.

El controlador es el centro de todo y se encarga de recibir y de responder a las peticiones del usuario a través de la interfaz y de mostrar la vista apropiada a la acción. Para ello le comunica al modelo lo que necesita obtener o actualizar y este se lo da. Posteriormente el controlador modifica estos datos para que la vista solamente tenga que mostrarlos y renderiza la vista.

El modelo constituye la información con la que el sistema trabaja, es decir, gestiona todos los accesos a la información ya sean consultas, modificaciones, eliminaciones o inserciones en una base de datos o llamadas a servicios. Su actuación viene determinada por lo que le comunica el controlador. Una vez que finaliza la tarea este da el resultado al controlador para que haga lo que deba de hacer.

La vista es pasiva, es decir, no contiene lógica y es creada a petición del controlador. La vista solo muestra el contenido que recibe del controlador y contiene todos los elementos de la interfaz gráfica. La vista únicamente se debe encargar de como se deben de ver los datos. Normalmente se piensa en esta capa como los componentes HTML, CSS, etc.

En este proyecto, este patrón es aplicado en la parte *backend*, ya que como se menciona más adelante en el apartado Tecnologías Utilizadas de forma más concreta, se utiliza el *framework* Spring Boot que implementa este patrón de forma predeterminada y obligatoria. En este caso debido a que

no hay interfaz en el servidor no es necesaria la capa de la vista y solamente se utilizan la capa de controlador y modelo.

En la Figura 3 se puede ver el comportamiento descrito en este apartado del patrón Modelo-Vista-Controlador de forma visual.

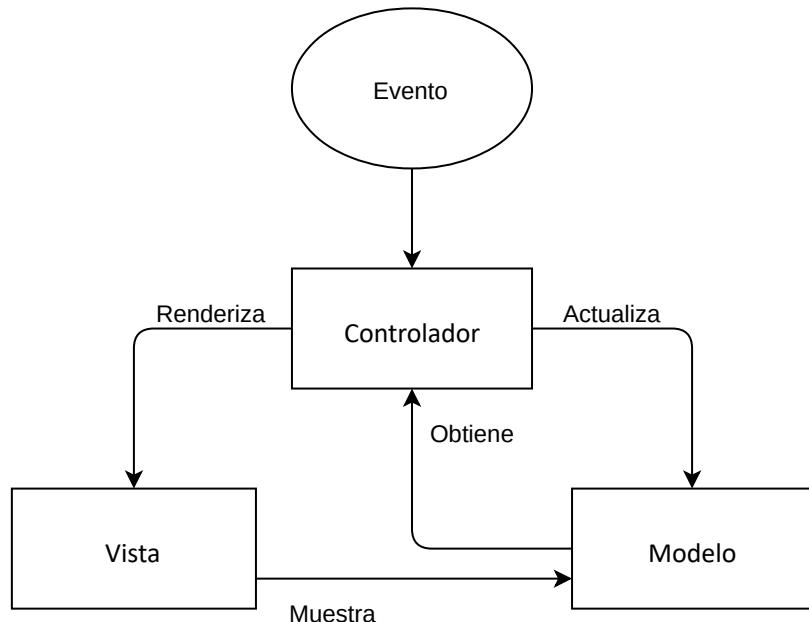


Figura 3: Patrón Modelo-Vista-Controlador.

Como se puede observar, el evento es capturado por el controlador que es el encargado de llamar el modelo para que se actualice u obtener aquello que necesite. Una vez hecho esto el controlador escoge la vista que debe mostrar al usuario junto con la información obtenida. La vista una vez es renderizada muestra aquello que el controlador le ha determinado y que forma parte del modelo.

14.2.2. Modelo-Vista-Presentador (MVP)

El Modelo-Vista-Presentador [28] o también conocido como MVP de forma abreviada, es un patrón de arquitectura de código evolucionado del Modelo-Vista-Controlador que separa la aplicación en tres capas distintas: el modelo, la vista y el presentador.

El presentador actúa como intermediario entre la vista y el modelo. En esta capa es donde se encuentra la lógica de negocio que atiende a los eventos del usuario y esta lógica es llamada por la vista. Es también el encargado de pedir la información al modelo y reorganizarla para que la vista solo deba mostrarla.

El modelo sigue actuando como en el patrón Modelo-Vista-Controlador. Este gestiona los accesos a la información como consultas, modificaciones, eliminaciones o inserciones en una base de datos o llamadas a servicios que se utilizan. Su actuación viene determinada por lo que le comunica el presentador. Una vez que finaliza la tarea este da el resultado al presentador para que haga lo que deba de hacer.

La vista es la encargada de capturar los eventos que el usuario realiza en la interfaz. Este comportamiento es diferente al del Modelo-Vista-Controlador donde se encargaba el controlador. La vista es pasiva y casi no tiene lógica salvo aquella necesaria para poder llamar al presentador cuando un evento se produce. La vista al igual que el patrón anterior, debe encargarse de mostrar la información que le llega del presentador.

En el proyecto este patrón es aplicado en la parte de *frontend* donde se realiza la aplicación móvil en si misma dejando de lado el servidor. Este patrón se ha escogido ya que en Android los eventos producidos por el usuario con la interfaz son capturados siempre por las vistas. Además este patrón permite de forma más sencilla el poder realizar comprobaciones y pruebas del código para ver que funcione como se espera.

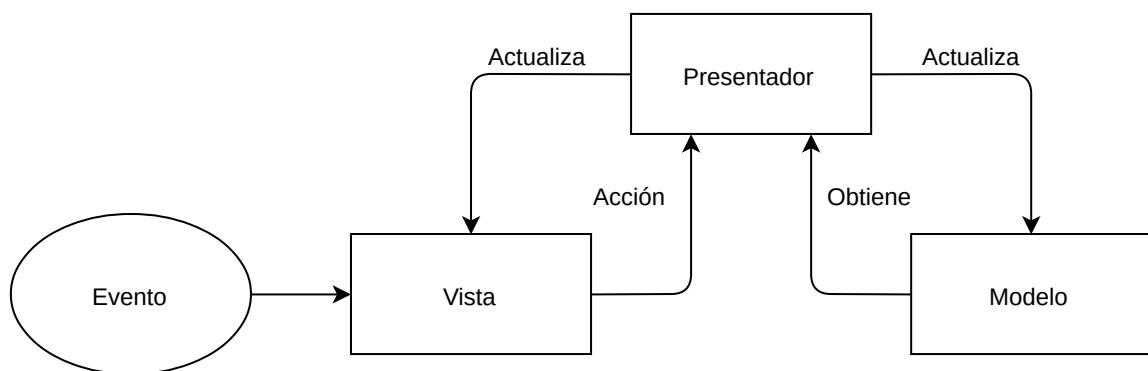


Figura 4: Patrón Modelo-Vista-Presentador.

Como se puede observar, el evento es capturado por la vista y esta indica al presentador que se ha producido. Este indica al modelo lo que necesita y este le devuelve información si es necesario o no. Una vez hecho esto, el presentador le indica a la vista que es lo que debe actualizar.

14.2.3. Singleton

El patrón Singleton [29] consiste en una clase que es responsable de declararse ella misma y de asegurarse que no hay más de una instancia de ella. Además debe proporcionar un punto de acceso global a su instancia para que pueda utilizarse desde cualquier lugar del código y sin tener que volver a crear una nueva instancia.

Como se puede observar en la Figura 5, la clase Singleton implementa el patrón Singleton. Esta clase contiene un atributo privado instancia que hace referencia a sí misma. También tiene un constructor privado el cual es ejecutado una vez se llama al método público obtenerInstancia(). El funcionamiento es el siguiente. En algún lugar del código se llama a la función pública obtenerInstancia() y este comprueba internamente si el atributo privado instancia es nulo o no. Si lo es, llama al constructor y se le asigna el valor al atributo y se devuelve. En cambio si no es nulo se devuelve directamente el valor del atributo privado instancia.

Como comentario final hay que asegurarse que la llamada al constructor de una clase que implementa el patrón Singleton debe de realizarse de manera síncrona. En caso contrario podría darse el caso en que se crearan dos instancias dejando de cumplir el patrón arquitectónico.

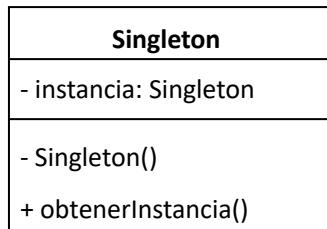


Figura 5: Patrón Singleton.

14.2.4. Adaptador

El patrón adaptador es un patrón que permite utilizar la interfaz de un clase en otra. En otras palabras permite usar objetos aparentemente incompatibles. Este patrón es especialmente útil y además de obligado uso en Android, específicamente en las listas que se visualizan en la aplicación donde se debe realizar un adaptador para que el sistema operativo pueda mostrarlas.

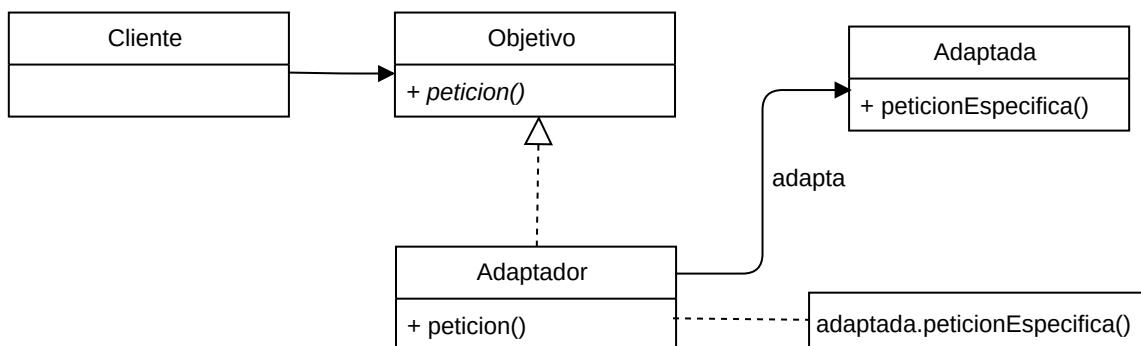


Figura 6: Patrón adaptador.

Como se puede observar en la Figura 6, la clase adaptador implementa la interfaz Objetivo. A su vez tiene un objeto de la clase Adaptada y cuando el cliente llama a la función `peticion()`, como es implementada por la clase Adaptador, esta puede realizar lo que necesite para poder llamar dentro de dicha función de forma satisfactoria a la función `peticionEspecificada()`.

14.2.5. Repositorio

El patrón repositorio es un patrón que permite separar de la lógica de negocio el acceso, modificación y eliminación de datos de la base de datos. Es un patrón muy sencillo que consta de una interfaz que contiene todos los métodos necesarios para actuar sobre una entidad específica. La interfaz es implementada por un clase que se encarga de definir la lógica de dichos métodos y de utilizar la base de datos. Debido que se utiliza el *framework* Spring Boot que más adelante se comenta, la implementación de la clase que implementa la interfaz ya lo hace el propio *framework* y solo se debe realizar la interfaz.

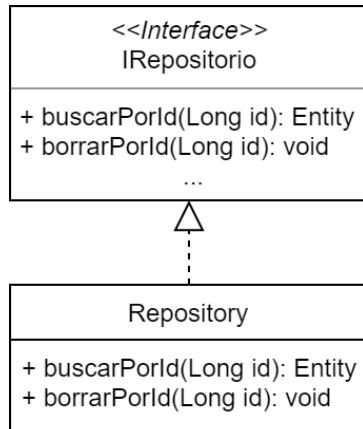


Figura 7: Patrón repositorio.

14.3. Diseño de la Interfaz Gráfica

En este apartado se describe la paleta de colores empleada en la interfaz gráfica de la aplicación Android y se muestran capturas de las diferentes pantallas que la forman.

14.3.1. Paleta de Colores

Para la interfaz gráfica de la aplicación se utilizan colores que hacen contraste entre sí para poder visualizar correctamente los elementos y los textos. En la siguiente tabla se muestran los principales colores utilizados así como su uso.

Código Hexadecimal	Color	Uso
68aaff	Azul	Elementos de la interfaz
5d99e5	Azul oscuro	Barra de estado de Android
9fdbff	Azul claro	Destacar elementos
de0000	Negro	Texto sobre fondos claros
fafafa	Blanco	Fondos y texto sobre fondos oscuros
808080	Gris	Texto sobre fondos claros
ef5350	Rojo	Errores

Tabla 57: Paleta principal de colores.

Estos colores son los principales de la aplicación no obstante en algunas ocasiones se utilizan variaciones de estos hacia más claro o más oscuros para adaptarlos a colores predeterminados de Android.

14.3.2. Capturas de Pantalla

En este apartado se muestran capturas de pantalla que se han realizado de la aplicación para mostrar la interfaz de forma completa. Algunas pantallas se han obviado ya que son muchas las que forman

la aplicación y otras son muy similares especialmente las listas o la información que se muestra en los perfiles. También se han incluido capturas de los dos tipos de anuncios, los de pedir o buscar ayuda y los de ofrecer ayuda aunque sean parecidas.



Figura 8: Pantalla de carga.



Figura 9: Pantalla de inicio de sesión.

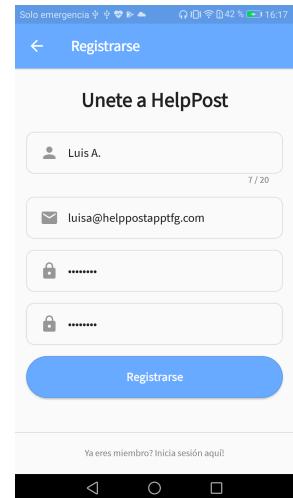


Figura 10: Pantalla de registro.



Figura 11: Pantalla de recuperación de contraseña (I).

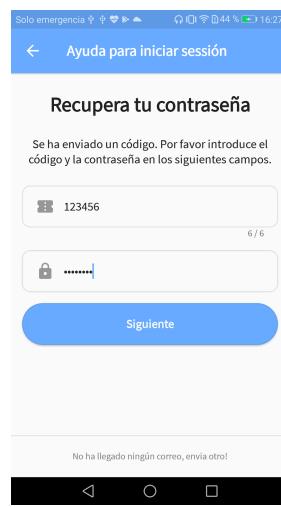


Figura 12: Pantalla de recuperación de contraseña (II).

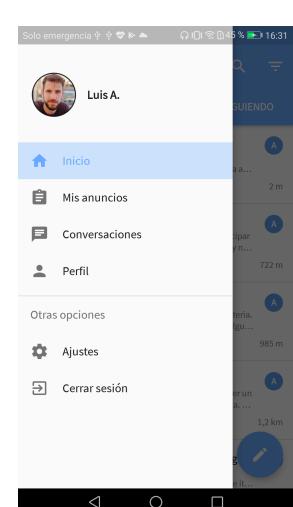


Figura 13: Menú desplegable.

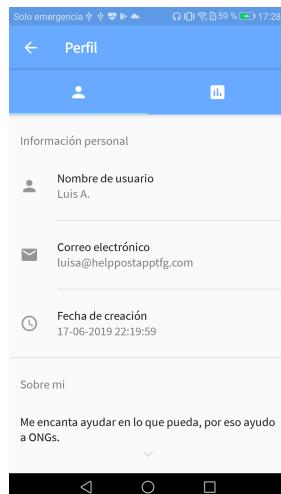
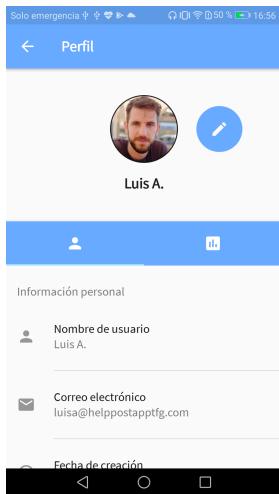


Figura 14: Pantalla de perfil del usuario.

Figura 15: Pantalla de información personal.

Figura 16: Pantalla de valoraciones recibidas.

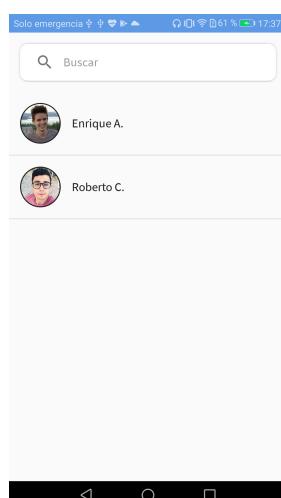


Figura 17: Pantalla de editar perfil.

Figura 18: Pantalla de seguidores y gente que sigue al usuario.

Figura 19: Pantalla de perfil de otro usuario.



Figura 20: Pantalla de inicio.

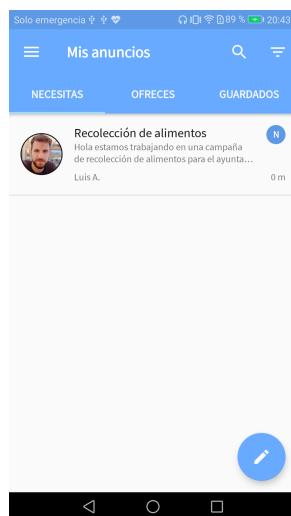


Figura 21: Pantalla de mis anuncios.



Figura 22: Pantalla de búsqueda de anuncios.

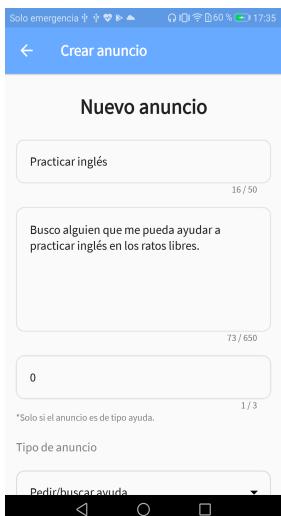


Figura 23: Pantalla de crear anuncio.



Figura 24: Pantalla de selección de localización.



Figura 25: Pantalla de anuncio de otro usuario donde se pide o busca ayuda.

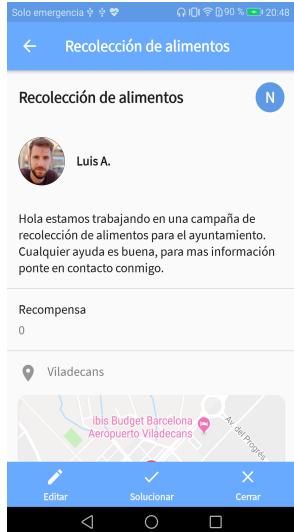


Figura 26: Pantalla de anuncio del usuario donde pide o busca ayuda.



Figura 27: Pantalla de anuncio de otro usuario donde se ofrece ayuda.



Figura 28: Pantalla de anuncio del usuario donde se ofrece ayuda.

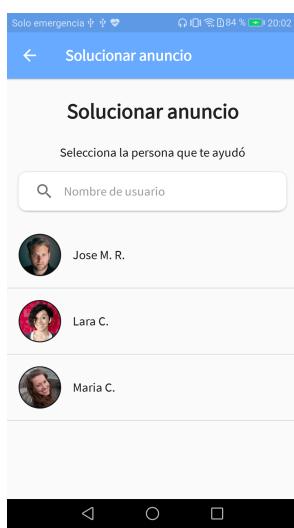


Figura 29: Pantalla de solución de anuncio.

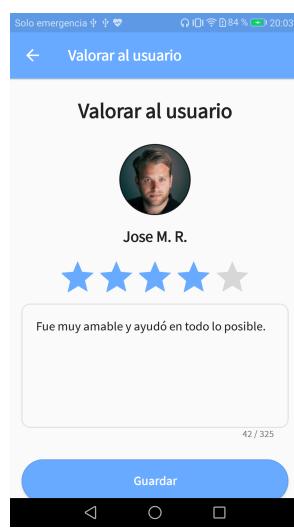


Figura 30: Pantalla de valoración de un usuario.

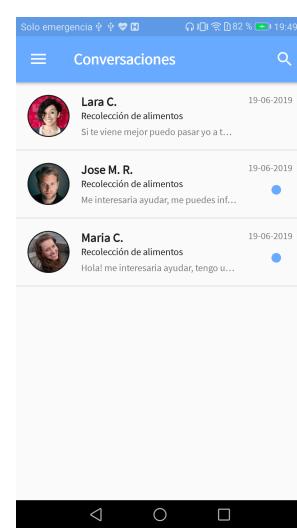


Figura 31: Pantalla de conversaciones.

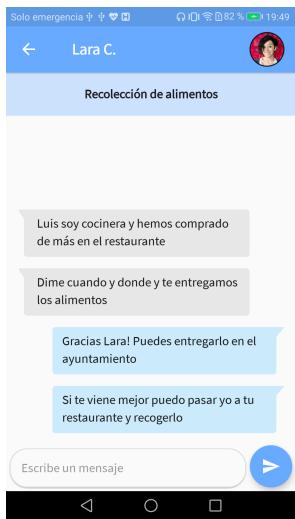


Figura 32: Pantalla de conversación.



Figura 33: Pantalla de cambiar contraseña.



Figura 34: Pantalla de eliminar cuenta.

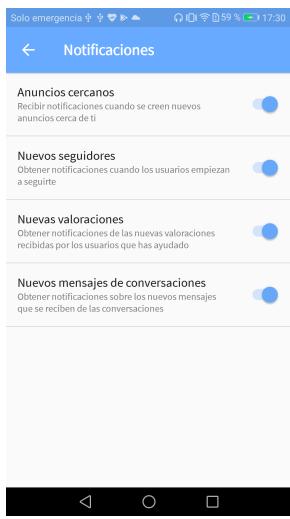


Figura 35: Pantalla de ajustes de notificaciones.



Figura 36: Pantalla de cerrar sesión.

14.4. Diseño del Cliente (*Frontend*)

En la Figura 8 se puede encontrar un diagrama muy básico y general de los componentes que forman la parte de la aplicación Android. Este diagrama es un diagrama muy general de la estructura que sigue la aplicación Android y que se va repitiendo a partir de las diferentes actividades de la aplicación así como los fragmentos. Como se ha comentado anteriormente, la aplicación Android se ha construido en base al patrón de arquitectura MVP. En dicha figura se puede encontrar la separación entre las tres capas.

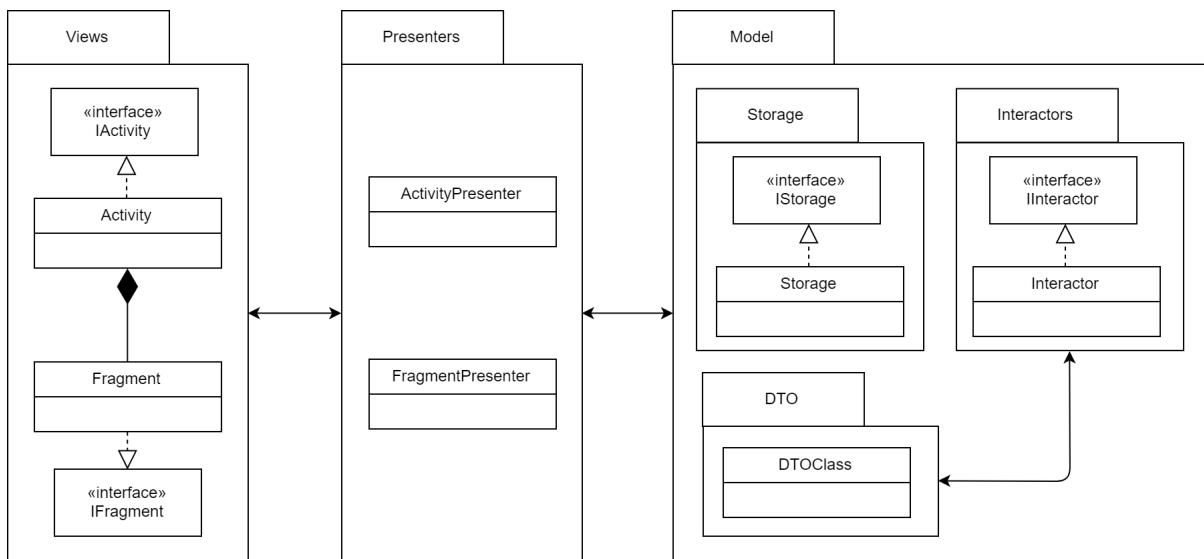


Figura 37: Diagrama de componentes del frontend.

La capa de la vista contiene actividades que representan las vistas en Android. A su vez las actividades pueden tener fragmentos que son vistas menores que se adhieren para poder favorecer a la reutilización de código. Debido a que no se utiliza ningún método de inyección de dependencias en este proyecto aunque de cara al futuro podría ser una buena idea, cada actividad y fragmento contiene una referencia a su propio presentador en el cual delega las acciones que el usuario hace sobre la interfaz de la aplicación.

El presentador se encarga de realizar la lógica necesaria y mostrarla a las actividades y a los fragmentos cada vez que recibe un evento de dichas vistas. Esto se hace a partir de la interfaz que implementan estas clases y sirve para desacoplar las vistas del presentador. El presentador obtiene la referencia de la vista a la que tiene que indicarle las acciones una vez la actividad o el fragmento crea al presentador. De igual modo que pasa con las vistas, al no utilizar inyector de dependencias, los presentadores contienen referencias al modelo como a los *interactor* o a la base de datos de la aplicación.

El modelo está formado principalmente por los *interactor* que se encargan de entregar al presentador observables a los cuales se pueden suscribir y de esta forma hacer peticiones al servidor de forma asíncrona y obtener los resultados a través de las clases DTO del modelo. Hay diferentes tipos de *interactors* dependiendo del tipo de petición que se quiera realizar al servidor, por ejemplo, autenticación, relacionado con el usuario, anuncios, valoraciones, etc. También se encuentra la

clase Storage que permite actuar con la base de datos de las SharedPreferences de Android que se utiliza por ejemplo para guardar los *tokens* de la sesión. Esta clase es especial pues es creada por las vistas ya que es necesario el contexto de la aplicación que solo se puede obtener de estas. Una vez la vista que corresponda la crea es transportada hasta el presentador el cual puede acceder a través de su interfaz. Del mismo modo que antes las interfaces se han creado para desacoplar las clases.

Como ejemplo para entender mejor esta estructura se muestra a continuación la historia de usuario de iniciar sesión así como una pequeña descripción de lo que sucede entre los diferentes componentes.

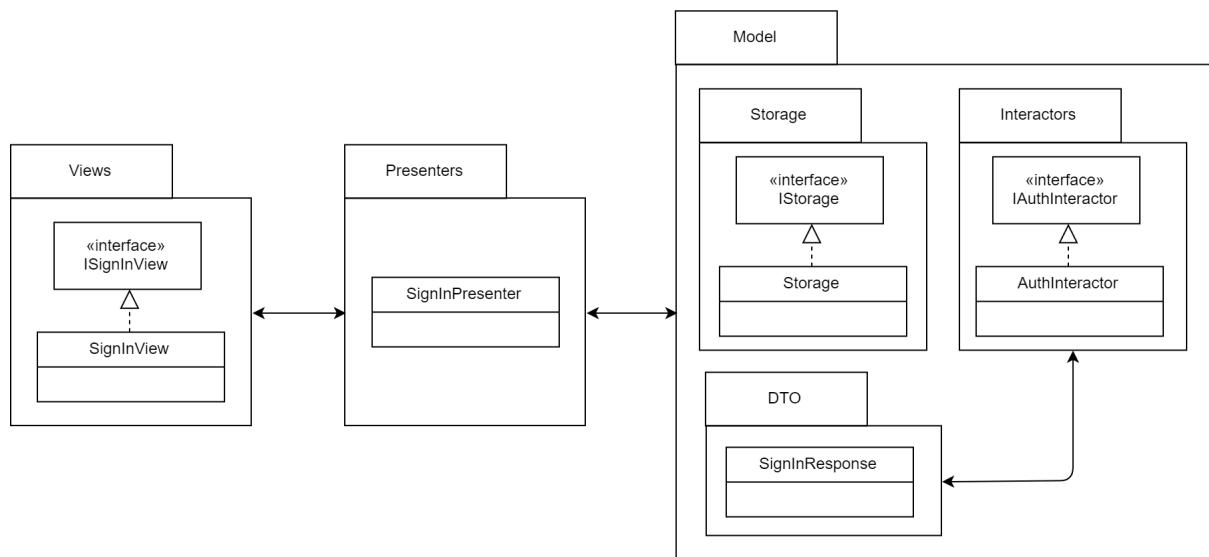


Figura 38: Diagrama de componentes del frontend de inicio de sesión.

Como se puede observar en la Figura 9, hay una actividad llamada SignInView que representa la vista del inicio de sesión. El presentador tiene una referencia de la vista a través de su interfaz para desacoplar las dos clases y tiene dos referencias más que son de AuthInteractor y de Storage a través de sus interfaces.

Cuando el usuario introduce sus credenciales, la vista comunica el evento al presentador y este se suscribe al observable que entrega AuthInteractor que indica cuando finaliza la petición y así puede recibir la respuesta del servidor que consta de los *tokens* de acceso. Estos *tokens* vienen en una clase DTO llamada SignInResponse que es entregada al presentador una vez el AuthInteractor finaliza la petición. Una vez recibida la respuesta, el presentador llama al Storage para que guarde dichos *tokens* en la base de datos de Android y le indica a la vista que redirija al usuario a la pantalla principal de la aplicación.

14.5. Diseño del Servidor (*Backend*)

En la Figura 10 se puede encontrar un diagrama de los componentes de la parte del servidor de forma muy básica y general. Al igual que con el diagrama de la parte del *frontend* este diagrama es muy básico y la estructura se va repitiendo. La arquitectura del servidor está centrada en la

implementación del patrón MVC. Normalmente la parte del modelo hace referencia a aquello que se muestra en la vista que es un subconjunto de la información recibida del Domain Model y que no tiene porqué corresponde con este. No obstante, como en este proyecto el servidor no muestra ninguna vista y además no se sigue una estructura muy compleja de se ha considerado como modelo al Domain Model directamente.

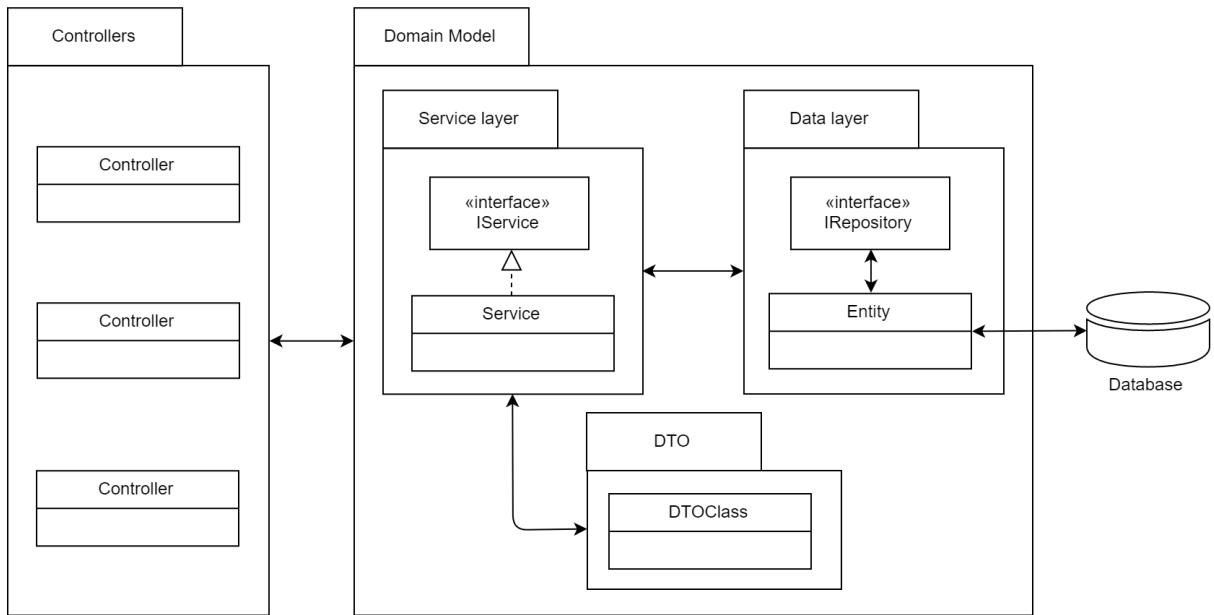


Figura 39: Diagrama de componentes del backend.

Como se puede observar, se encuentran los controladores que se encargan de capturar las peticiones de la aplicación Android. Los controladores si es necesario se comunican con el Domain Model para poder enviar la respuesta al cliente y realizar las operaciones necesarias. En sí el controlador se encarga de pedir los recursos necesarios y una vez obtenidos crear una respuesta que la aplicación Android sea capaz de entender.

En cuanto al Domain Model, este se divide en diversas capas. Se ha hecho así para facilitar el cambio de funcionalidades en un futuro y según las recomendaciones del framework Spring Boot. Primero está la capa de servicios que es con la que contactan los controladores y se encargan de realizar toda la lógica. Cada controlador tiene su propio servicio a través de una referencia mediante la interfaz del servicio para desacoplar el código.

Después de los servicios está la capa de datos que es donde se encuentran los repositorios y las entidades. Los repositorios son los encargados de realizar las llamadas a la base de datos y obtener los resultados. Dichos resultados pueden ser una fila de una tabla de la base de datos o varias llegando a tener listas de resultados. Cada fila es asociada con una instancia de una entidad que contiene los atributos de la fila y de esta forma tratar siempre con objetos en la parte del código. Los encargados de llamar a los repositorios y tratar las entidades son los servicios.

Una vez el repositorio obtiene todo lo necesario, es entregado al servicio y este realiza las operaciones necesarias, devuelve un resultado al controlador. De esta forma el controlador crea una respuesta para enviar al cliente o utiliza este resultado para realizar otras operaciones. Para entender mejor esta estructura se muestra una vez más un diagrama de la historia de usuario de iniciar sesión en la parte del servidor.

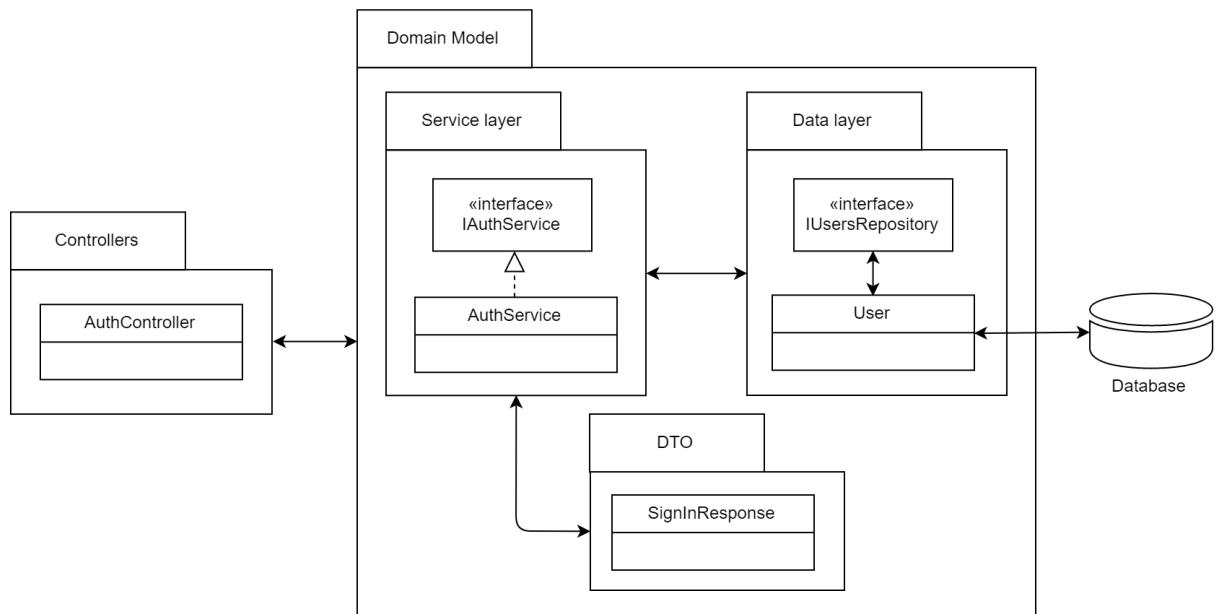


Figura 40: Diagrama de componentes del backend de inicio de sesión.

Como se puede observar en la Figura 11, cuando la aplicación Android envía una solicitud de inicio de sesión, el controlador `AuthController` que se encarga de recibir las peticiones relacionadas con el registro, inicio de sesión entre otros, captura dicha petición y llama a `AuthService` a través de su interfaz para poder realizar el proceso de inicio de sesión.

El servicio llama al repositorio para comprobar si existe el usuario que quiere iniciar sesión a través del nombre de usuario o correo y la contraseña otorgados. Si es así, el servicio genera un *token* de acceso, actualiza el de refresco y crea el objeto `SignInResponse` que devuelve al controlador como respuesta. El controlador una vez recibe la respuesta directamente la envía al cliente pues no debe de realizar ningún cambio sobre esta.

14.6. Diseño de la Base de Datos

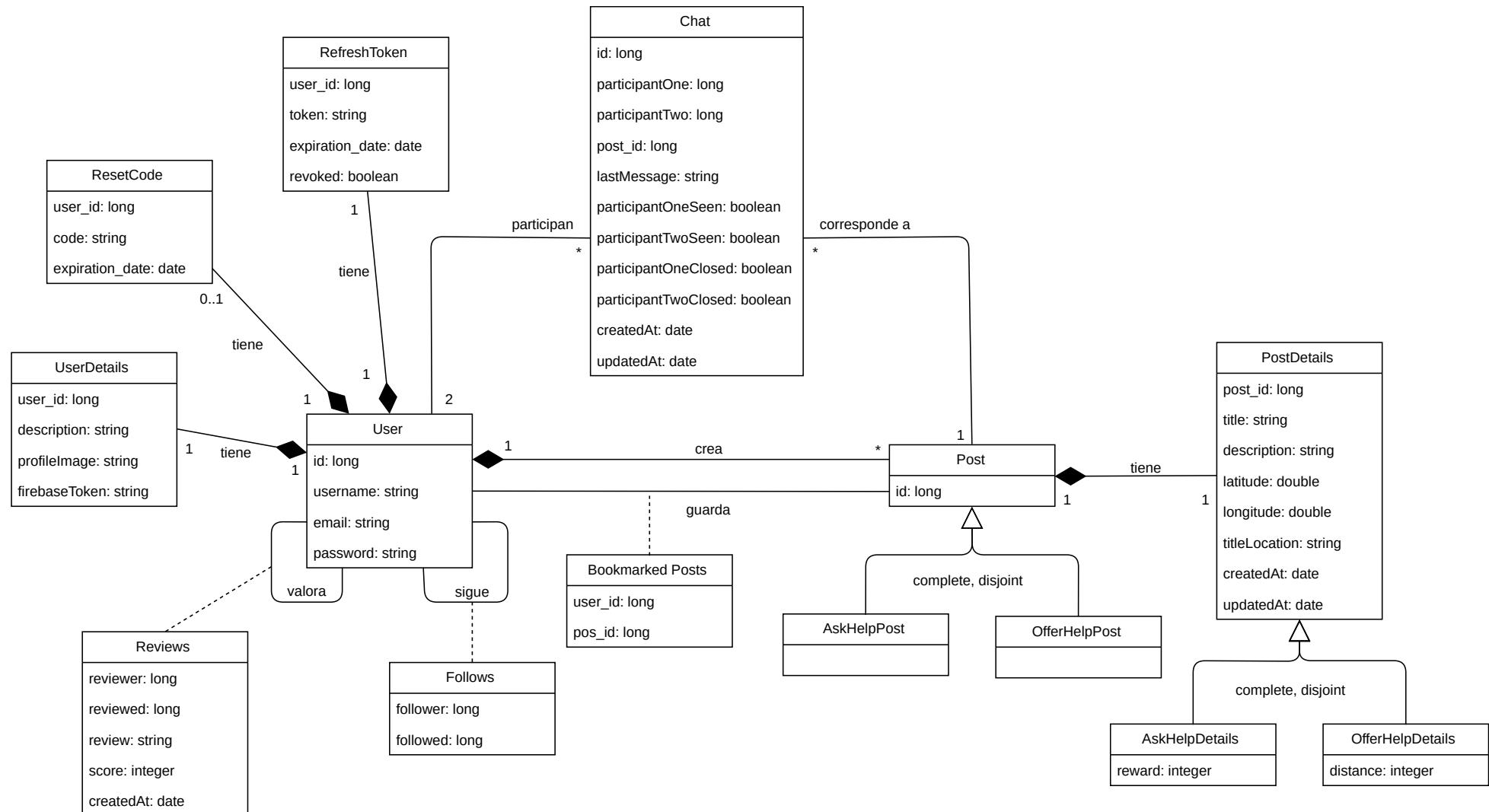


Figura 41: Diagrama UML de la base de datos.

Restricciones textuales:

1. Claves primarias: (User: id), (Chat: id), (Post: id), (RefreshToken: user_id), (ResetCode: user_id), (UserDetails: user_id), (Reviews: reviewer, reviewed), (Follows: follower, followed), (PostDetails: post_id), (BookmarkedPosts: user_id, post_id)
2. Un usuario solo puede guardar anuncios que no haya guardado previamente.
3. Un usuario solo puede seguir a otro si no le seguía previamente.
4. Un anuncio donde se pide ayuda solo puede tener detalles de tipo pedir ayuda. (AskHelpPost → AskHelpDetails)
5. Un anuncio donde se ofrece ayuda solo puede tener detalles de tipo ofrecer ayuda. (OfferHelpPost → OfferHelpDetails)
6. Un usuario solamente puede valorar a otro usuario si han conversado y no han cerrado los dos la conversación.
7. Las fechas de actualización tienen que ser superiores o iguales a las creación. (updatedAt >= createdAt)
8. Una conversación se cierra completamente cuando dos usuarios cierran la conversación.

15. Tecnologías Utilizadas

En este apartado se describen las tecnologías que se utilizan durante el desarrollo del proyecto. Primero se definen los entornos de desarrollo que se utilizan. Seguidamente se describen las tecnologías de desarrollo como *frameworks*, librerías o bases de datos utilizadas. Finalmente se explican los lenguajes de programación utilizados.

15.1. Entornos de Desarrollo

Para facilitar el trabajo de la creación del código de la aplicación se utilizan entornos de desarrollo ya preparados que otorgan características y funcionalidades que de otra forma no se tendrían. En el siguiente apartado se explican los entornos que se han utilizado, específicamente, Android Studio para la parte *frontend* y Eclipse para la parte *backend*.

15.1.1. Android Studio

Para facilitar la creación de código para la parte *frontend* se utiliza un entorno de desarrollo llamado Android Studio. Tal y como comenta la página de desarrolladores oficial de Android [30], Android Studio es el entorno oficial para el desarrollo de aplicaciones para el sistema operativo Android. Está basado en el *software* de IntelliJ IDEA de JetBrains y se publicó en 2014 de forma gratuita.

Algunas de las características que presenta este entorno de desarrollo son que incluye el kit de desarrollo de Android automáticamente que incluye herramientas importantes como un emulador o las librerías de código. También tiene completo soporte con Java y Kotlin, lenguajes de programación oficiales de Android. Además, incorpora integración con Git, es decir con Github o Bitbucket, con lo que esta herramienta es muy útil para equipos de desarrollo.

15.1.2. Eclipse

Al igual que se utiliza Android Studio para la parte *frontend*, para la parte del *backend* se utiliza un entorno de desarrollo llamado Eclipse. Eclipse [31] es un entorno de desarrollo genérico sin ánimo de lucro que no tiene asignado ningún lenguaje determinado aunque si se quiere considerar uno sería Java ya que tiene mucha popularidad entre los desarrolladores de este lenguaje. Inicialmente fue desarrollado por la compañía IBM pero ahora es desarrollado por la Fundación Eclipse desde 2003.

Presenta características como un depurador de código, la posibilidad de crear pruebas unitarias con el *framework* JUnit ya incorporado y otras características propias del lenguaje Java. Debido a que fue diseñado para que fueran creciendo sus funcionalidades, se puede obtener a partir de Internet otras características de forma gratuita a partir de complementos. De esta forma, este entorno de desarrollo no solo se puede utilizar para Java sino para muchos otros e utilizar herramientas que desde un inicio no estaban añadidas.

15.2. Tecnologías para Desarrollo

Para el desarrollo de la aplicación se utilizan tecnologías como librerías de código, *frameworks*, APIs y bases de datos que posibilitan al proyecto ser desarrollado más rápido y con mayor calidad. En el siguiente apartado se describen aquellas tecnologías principales y más destacadas de las cuales se hace uso.

15.2.1. Android Software Development Kit

Se ha decidido realizar una aplicación Android principalmente porque es la plataforma donde más libertad se otorga al desarrollador. Al contrario que sus competidoras, Android es completamente accesible por cualquiera de forma gratuita y posibilita el uso de aplicaciones propias sin tiempo de caducidad. Además, solo se dispone de teléfonos Android para comprobar de forma física el funcionamiento y no de teléfonos con sistema iOS. Para la realización es necesario el *kit* de desarrollo oficial.

El Software Development Kit de Android [32], llamado comúnmente por los desarrolladores Android SDK, es un conjunto de herramientas oficial de desarrollo para aplicaciones Android. Este kit contiene un depurador de código, un simulador de teléfono, documentación, ejemplos de código y las librerías necesarias y básicas para llevar a cabo una aplicación Android. Está disponible para sistemas operativos Windows, Linux y Mac.

El SDK es actualizado periódicamente de manera coordinada con las actualizaciones del sistema operativo Android. También, mediante estas actualizaciones se va dando soporte a anteriores versiones de Android, permitiendo así que funcionalidades que solo estaban disponibles para dispositivos nuevos puedan utilizarse en antiguos.

15.2.2. Retrofit y GSON

En este proyecto, la parte *frontend* de la aplicación realiza peticiones al servidor que consiste en una Rest API. Para llevar a cabo estas peticiones es necesario crear las operaciones necesarias para realizar una conexión a Internet y utilizar un analizador de datos JSON que es el formato en que se obtienen las respuestas. Debido a que es una tarea costosa y que ya se ha realizado múltiples veces por otros desarrolladores, en este proyecto se utiliza una librería llamada Retrofit y otra llamada GSON que se encargan de este proceso.

Retrofit [33] es una librería de cliente HTTP seguro para Android y Java desarrollado por Square. Mediante esta librería se pueden traducir fácilmente Rest APIs en interfaces Java. De esta forma, las peticiones GET, POST, DELETE, PUT, y HEAD se pueden realizar de forma sencilla sin tener que crear ninguna operación de conexión ya que esta es realizada directamente por Retrofit.

GSON [34], en cambio, es una librería de código abierto creada por Google para Java que permite la deserialización de los objetos Java y viceversa en formato JSON. Esta se adhiere a la creación de Retrofit y automáticamente es capaz de convertir un objeto Java que contiene datos en formato JSON, el cual el servidor creado con Spring Boot es capaz de leer. Del mismo modo, una vez que el

servidor lanza una respuesta en formato JSON, esta librería automáticamente convierte la respuesta en un objeto Java.

15.2.3. RxJava

La librería Retrofit por si misma permite las peticiones asíncronas mediante *callbacks*, algo totalmente necesario en la parte *frontend* de esta aplicación para que el usuario no esté obligado esperar a que esta termine la petición al servidor para hacer otra cosa. No obstante, esta implementación puede derivar en problemas en algunos casos, por ejemplo, si queremos realizar dos peticiones al mismo tiempo ya que supondría una gran cantidad de código difícil de comprobar y mantener. Para corregir esto se utiliza la librería RxJava la cual Retrofit permite hacer uso de ella.

RxJava [35] es una implementación para la Java Virtual Machine de ReactiveX, una librería para crear programas asíncronos y basados en eventos mediante la creación de cadenas de observadores. Esta librería mediante el patrón observador permite no tan solo hacer lo que Retrofit realiza mediante *callbacks* sino hacer varias peticiones a la vez o ir adquiriendo los datos de forma progresiva. Los observadores son notificados únicamente cuando los datos llegan o se produce un error. Por lo tanto, esta librería facilita la creación de código asíncrono para hacerlo de forma más breve. También disminuye la dificultad de comprobación de código y el mantenimiento.

15.2.4. Picasso

Picasso es una librería para Android que permite descargar imágenes a partir de una URL de forma rápida y transparente al programador en segundo plano. Además, esta librería evita la descarga continua de imágenes proporcionando un método de cache y se encarga de eliminar y liberar la memoria del dispositivo móvil cuando la imagen ya no es necesaria.

15.2.5. Spring y Spring Boot

Debido a que se realiza una Rest API en la parte del servidor para crear un servicio web que permita la obtención, el guardado y el procesamiento de los datos, se ha decidido emplear un *framework* que implemente el patrón Modelo Vista Controlador. Hay muchos disponibles en el mercado con esta capacidad como Ruby on Rails o Django. Finalmente se ha optado por Spring, más concretamente Spring Boot ya que utiliza Java y es compatible con Eclipse.

Spring [36] es un *framework* con lenguaje de programación Java y Kotlin creado por programadores y para programadores en 2003 disponible de forma gratuita. Esta herramienta facilita a los desarrolladores a crear aplicaciones de forma más rápida y eficaz evitando la realización de código que ya se ha realizado múltiples veces por otros desarrolladores. Este permite la realización del patrón Modelo Vista Controlador que otorga la posibilidad de reutilización de código en un futuro y la facilidad en el mantenimiento de este. En otras palabras, Spring ayuda a crear aplicaciones Java y facilita la reutilización de código.

En cuanto a Spring Boot [37] es un proyecto que surge de Spring debido a la dificultad que tiene este de configurarse por archivos XML. Este proyecto supone una facilidad aún mayor de la que ya hacía

Spring a la hora de construir, ejecutar y configurar las aplicaciones. Para ello proporciona herramientas a través de nuevas librerías de manera directa y mantiene aquellas que ya estaban como la implementación del Modelo Vista Controlador. Algunas librerías que proporciona están relacionadas con pruebas o creación de servicios web.

15.2.6. Java Persistence API (JPA) y Hibernate

Por tal de automatizar la creación de tablas y secuencias SQL de las bases de datos y centrarse únicamente en el código de la parte del *backend*, se ha decidido utilizar una implementación de Java Persistence API o también conocido como JPA junto con Spring Boot en la parte del servidor.

JPA [38] es una solución estándar que hace de puente entre el uso de orientación a objetos y sistemas de base de datos relacionales. La idea que propone JPA es convertir clases Java en tablas de una base de datos y los atributos de las clases en filas de estas. No obstante, como se ha comentado, JPA es tan solo una especificación y es necesario una implementación para poder llevar a cabo este proceso.

En este proyecto se utiliza el *framework* Hibernate [39] como implementación de JPA. El *software* apareció en 2001 a manos de Red Hat y funciona a partir de anotaciones que se añaden en el código. Viene incorporado ya en la instalación de Spring Boot por lo que resulta ser el más atractivo de usar. Este permite el poder mapear clases Java con tablas de bases de datos relacionales. De esta forma permite abstraerse de las sentencias SQL propias de las bases de datos facilitando la obtención y el almacenamiento de datos.

15.2.7. API de Cloudinary

En la aplicación se utiliza el servicio web Cloudinary para el almacenamiento y gestión de imágenes de los usuarios. Para ello es necesario poder comunicar el *software* que se crea junto con el de los servidores del servicio. Debido a que el código puede variar entre los dos sistemas, los propietarios de Cloudinary disponen de lo que se denominan APIs para poder realizar la comunicación entre los dos sistemas de forma satisfactoria. Una API es un conjunto de funcionalidades escritas en el mismo lenguaje de programación que otro programa para que pueda hacer uso de un servicio escrito en un lenguaje de programación diferente.

15.2.8. API de Firebase

Para la parte de mensajería instantánea de la aplicación, así como para el envío de las notificaciones a los distintos usuarios registrados en la aplicación, es necesario la comunicación con el servicio de Firebase de Google. Para ello se ha utilizado la API Admin de Firebase en la parte del servidor de recursos de la aplicación. Esta API al igual que la de Cloudinary existe para los propietarios de una cuenta en dicho servicio y con ella se puede modificar cualquier dato de la base de datos de Firebase o realizar peticiones para registrar usuarios o enviar mensajes.

15.2.9. API de Google Maps

En la parte de la aplicación Android se utiliza a menudo la localización y es muy útil complementarlo con un mapa en el cual indicar la posición determinada por el usuario o anuncio. Por ello, se han intentado utilizar varias herramientas de mapas como por ejemplo la librería OpenStreetMaps la cual es de libre uso. No obstante las diferentes alternativas no dieron buenos resultados ya que no eran compatibles con las últimas actualizaciones de las librerías de Android. Por ello se decidió usar la API de Google Maps para Android la cual a día de hoy es gratuita. Esta API permite la obtención de mapas muy detallados de los servidores de Google además de poder dibujar formas y poner marcadores en estos.

15.2.10. Base de Datos PostgreSQL

Como es obvio, es necesario una base de datos para almacenar la información de los usuarios en el servidor. De esta forma el cliente Android puede realizar peticiones para poder obtenerlos o modificarlos. Debido a la experiencia que se tiene con bases de datos relacionales se ha decidido hacer uso en este proyecto de PostgreSQL para producción.

PostgreSQL [40] es un sistema de base de datos relacional de código abierto desarrollado en lenguaje C y publicado en 1995 que utiliza y extiende el lenguaje SQL combinándolo con otras características que le permiten almacenar de forma segura. Este *software* cumple con las propiedades ACID que asegura que ningún dato se pierda o se transmita a otro lugar del sistema en caso de fallo. También cumple en gran medida con el estándar SQL permitiendo así poder migrar de base de datos de forma sencilla.

15.3. Tecnologías para Pruebas

Para poder comprobar que la aplicación realiza lo que se espera es necesario utilizar herramientas que faciliten esta tarea y ayuden. A continuación se describen aquellas tecnologías que se han utilizado para la realización de pruebas tanto en la parte del cliente como del servidor.

15.3.1. JUnit

Tanto para la parte *frontend* como la de *backend* de este proyecto se realizan pruebas para comprobar que las funcionalidades creadas funcionan correctamente a nivel de producción. Para ello se usa una herramienta muy común en la programación de pruebas llamada JUnit.

JUnit [41] es un framework diseñado para poder escribir y ejecutar pruebas unitarias de cada funcionalidad. Inicialmente fue creado para el lenguaje de programación Java pero se ha ido exportando a otros. Cada prueba que se crea con esta herramienta es independiente de las otras, de esta forma se puede determinar de forma segura que realmente las funcionalidades funcionan correctamente y no están alteradas por otras. Además, se ha convertido en un estándar ya en la industria para las pruebas unitarias lo que le ha permitido tener gran soporte por parte de otras librerías y aumentar así su facilidad de uso.

15.3.2. Mockito

A la hora de hacer pruebas unitarias con el *framework* JUnit muchas veces encontramos que el código depende de otro. De esta forma es difícil comprobar que la funcionalidad funciona correctamente por sí misma ya que existen dependencias. Para solucionar este problema se ha optado por la ayuda de un *software* llamado Mockito.

Mockito [42] es un *framework* que permite suplantar funcionalidades o clases Java para poder realizar pruebas de funcionalidades que tienen dependencia de otras. Esta herramienta es muy útil ya que evita el tener que crear esas dependencias de forma trivial para poder comprobar las funcionalidades. Tan solo hay que indicarle que es aquello que se quiere suplantar y determinar que es lo que se quiere obtener si fuese necesario. Una vez hecho esto, él se ocupa de todo lo demás. El uso de esta herramienta supone un reducción de tiempo considerable y su compatibilidad con JUnit hace que sea muy fácil de utilizar.

15.3.3. Base de Datos H2

En cuanto a la realización de las pruebas de las funcionalidades en el *backend*, es necesario utilizar en algún momento algún mecanismo para comprobar que las sentencias SQL o comprobar que el almacenamiento y la obtención de datos funcionan correctamente. Por esto, es necesario el uso de una base de datos con la que interactuar. Debido a que el proceso de preparación es fastidioso y conlleva tiempo, en este proyecto se utiliza la base de datos H2 para pruebas que facilita esto.

La base de datos H2 [43] es una base de datos relacional de código abierto escrita en Java y creada en 2005 que puede ser incrustada en aplicaciones Java. Esta incrustación permite el no tener que crear una configuración tan extensa como se haría con una base de datos normal. También la velocidad de acceso a esta es rápido y es muy liviana y tiene la capacidad de cambiar de modo dependiendo de la base de datos que se utilizar en producción, es decir, al ponerla en modo PostgreSQL actuaría como tal. Además, una vez finalizadas las pruebas se elimina y deja de ser accesible en memoria con lo que la convierte en ideal para esta parte del trabajo.

15.4. Lenguajes de Programación

La elección de un lenguaje de programación puede ser clave en el desarrollo de la aplicación. Para la parte *frontend* se utiliza Kotlin mientras que para la parte *backend* Java. A continuación se describen estos dos lenguajes de programación y el por qué se ha optado por ellos.

15.4.1. Kotlin

Kotlin [44] es un lenguaje de programación de código abierto orientado a objetos para la Java Virtual Machine (JVM) creado por JetBrains y que surge de Java. Este lenguaje apareció en febrero de 2012 y surgió por que sus creadores no encontraban en ningún lenguaje las características que buscaban y que fuese rápido a la hora de compilar. En 2017 Google decidió nombrar a Kotlin como lenguaje oficial junto con Java para Android haciendo que su uso creciese de forma exponencial.

Este lenguaje presenta características y mejoras de Java [45] como el control de excepciones en especial el control de variables nulas. Las variables nulas son el principal problema que se encuentran en los programas desarrollados con Java ya que su control es mínimo. Kotlin, en cambio, hace un control exhaustivo con tan solo escribir un carácter en el código. Otra característica que proporciona es la reducción de código respecto a Java. Esto permite que los ficheros tengan tamaño menor permitiendo a la aplicación ser más liviana. Por último, el código Java puede utilizarse junto con el de Kotlin en el mismo proyecto ya que los dos producen el mismo *bytecode* permitiendo así tener diferentes ficheros en distintos lenguajes dependiendo de las características que se necesiten.

Dejando de lado los beneficios que pueda otorgar, se ha optado por este lenguaje para la parte *frontend* de la aplicación debido a dos motivos. Primero porque por decisión propia se ha propuesto como un objetivo el aprender un lenguaje de programación nuevo y aplicarlo en una situación real. Segundo y último porque en los últimos años se está dando mucha importancia en el mercado laboral el tener conocimiento de programación en Kotlin ya que es el futuro del desarrollo de aplicaciones móviles.

15.4.2. Java

Java [47] es un lenguaje de programación orientado a objetos diseñado para tener tan pocas dependencias de implementación como sea posible. Fue desarrollado originalmente por Sun Microsystems que posteriormente fue adquirida por Oracle la cual es la dueña actual. La intención de Java es la de que los desarrolladores escriban una única vez el programa y que este pueda ser utilizado en cualquier dispositivo a través de la Java Virtual Machine (JVM).

Este lenguaje tiene una gran comunidad detrás de él lo que permite facilidad a la hora de aprenderlo debido a la gran cantidad de documentación e información. Además, es fácil de entender y leer ya que todo está estructurado. También, la compilación de código es rápida y el programa puede ser ejecutado en cualquier dispositivo *hardware* que tenga una máquina virtual de java permitiendo así una portabilidad a gran escala.

A pesar de estos beneficios también tiene algunos defectos. Uno de los defectos más graves es el del tiempo de ejecución el cual es lento y el consumo de memoria. Esto es producido a que el programa es ejecutado en la Java Virtual Machine y es el precio a que se pueda utilizar el programa en cualquier *hardware*. También uno de los problemas más comunes en aplicaciones Java es el del control de variables nulas. Este problema puede causar que el programa pare instantáneamente de forma inesperada con lo que el programador debe tener especial cuidado.

Se ha decidido utilizar este lenguaje para la parte *backend* del proyecto por varios motivos. El primer motivo es que hay accesible mucha información útil para la realización del servidor a través de Java lo que permite mayor rapidez al desarrollar el código. También se ha escogido este lenguaje por decisión propia ya que durante el último año de carrera se ha dejado un poco de lado este lenguaje por el uso de otros y la realización de este proyecto puede servir para refrescarlo. Por último porque Java es uno de los lenguajes más expandidos y demandados en el ámbito laboral y es compatible con el *framework* escogido.

16. Implementación de Funcionalidades

Una vez descrita la tecnología que se utiliza puede ser una buena idea explicar como funcionan y como se aplican estas tecnologías a algunas de las funcionalidades más complejas del proyecto. En el siguiente apartado se recogen los elementos que han supuesto un mayor reto en su desarrollo y se explica de la forma más precisa como funcionan una vez realizado su desarrollo.

16.1. Inicio de Sesión, Sesión Continua y Acceso a Recursos

En los siguientes subapartados se explica la implementación realizada para que los usuarios puedan iniciar sesión en la aplicación y que esta se mantenga abierta así como el acceso a recursos del servidor.

16.1.1 Planteamiento

Para hacer uso de la aplicación es necesario que los usuarios inicien sesión al abrir la aplicación una vez que han cerrado sesión o una vez se han registrado. Utilizando el método trivial, los usuarios deben de estar realizando el proceso de inicio de sesión continuamente cada vez que abren la aplicación. Este proceso claramente no es bueno ya que supone una molestia grande para el usuario y además, puede ser una vulnerabilidad importante ya que deben de enviar su contraseña al servidor.

Para poder solucionar y mejorar el proceso de inicio de sesión se ha decidido realizar un mecanismo basado en *tokens*. Para poder explicarlo de forma detallada es necesario definir primero tres conceptos clave que son el núcleo de esta solución.

1. **JSON Web Token (JWT)**: Tal y como explica la página oficial de JSON Web Token [47], este es un estándar que define una manera compacta y segura de transmitir información a través de sistemas mediante *tokens* basado en JSON. La información que contiene está firmada digitalmente mediante una clave secreta que solo el sistema emisor conoce con lo que el *token* puede ser verificado y confiado. Una vez manipulado este deja de ser válido. Dicho *token* contiene todos los permisos necesarios para que el usuario pueda acceder a los recursos del servidor y consiste de tres partes: la cabecera, el cuerpo o contenido y la firma.
2. **Token de acceso**: Un *token* de acceso es aquel que se utiliza para poder acceder a los recursos protegidos de un servidor sin necesidad de transmitir las credenciales del usuario. Debido a esto, estos *tokens* deben de presentar toda la información necesaria para poder obtener la información y tienen un tiempo de caducidad muy bajo para evitar fallos de seguridad.
3. **Token de refresco**: Un *token* de refresco es aquel que se utiliza para poder obtener nuevos de tipo acceso y continuar accediendo a los recursos protegidos de un servidor sin necesidad de transmitir las credenciales del usuario. Estos tienen un tiempo de expiración mayor al de los de acceso y no contienen información alguna del usuario.

16.1.2. Solución Implementada

Cada vez que el usuario introduce sus credenciales en la aplicación para iniciar sesión, estas se envían al servidor y se comprueba si son válidas. Si estas lo son, el servidor envía al cliente un token de acceso que es un JSON Web Token y un *token* de refresco el cual es guardado también en la base de datos del servidor. En cambio, si no lo son, el servidor devuelve un error HTTP 401 negando el acceso.

Una vez el cliente ha obtenido estos datos se guardan en las Shared Preferences [48] que es un fichero de la aplicación que solo es accesible por esta misma. Debido a que se almacenan en texto plano se ha decidido otorgar más protección y encriptar estos datos antes de guardarlos. Por si fuera poco, para evitar la obtención de claves de encriptación, estas se han almacenado en el KeyStore [] de Android, un sistema de seguridad del sistema operativo que impide la extracción de estas.

Una vez realizado el proceso de obtención de *tokens* y guardado por parte del cliente, por cada petición al servidor para acceder a un recurso protegido se adhiere a la petición HTTP la cabecera Authorization con el *token* de acceso. El servidor determina si tiene los privilegios necesarios como roles o si está caducado y permite o niega el acceso según lo corresponda.

Cuando el *token* de acceso caduca y el servidor se lo notifica, el cliente debe enviar el de refresco al servidor. El *token* de refresco se adhiere en la cabecera “RefreshToken” y el servidor valida si se encuentra en la base de datos como válido. Una vez hecho esto, comprueba si es inválido o ha caducado. Si no es el caso y todo es correcto el servidor cambia el *token* de refresco de la base de datos y envía al cliente como respuesta un nuevo *token* de acceso y el de refresco. De forma contraria, se niega el acceso y el usuario debe de introducir de nuevo sus credenciales. A partir de aquí el proceso de repite de forma iterativa pudiendo mantener al usuario con la sesión iniciada sin tener que introducir las credenciales continuamente.

16.2. Recuperación de Contraseña

En los siguientes subapartados se explica la implementación realizada para que los usuarios puedan recuperar de contraseña.

16.2.1. Planteamiento

En toda aplicación que requiera que su usuario esté registrado y deba iniciar sesión es necesario que se proporcione algún tipo de mecanismo para poder recuperar la contraseña cuando el usuario no es capaz de recordarla. Un método trivial sería enviar una contraseña nueva a los correos de los usuarios cuando estos hayan pedido recuperarla. Este método, a pesar de que funciona, no es el mejor ya que cualquier otro usuario es capaz de pedir las veces que quiera que para un correo que no sea el suyo se cambie la contraseña. Esto provocaría que el usuario real de dicho correo no pueda iniciar sesión y deba realizar el proceso de recuperación de contraseña para poder acceder a la aplicación, con lo que se entraría en un bucle infinito. Además, este método supone una brecha de seguridad pues solo debe de ser capaz el usuario original de recuperar su contraseña.

Otros servicios que tienen la posibilidad de recuperar la contraseña, por ejemplo las páginas web, envían un correo electrónico con una dirección web de su dominio donde el usuario puede ingresar y cambiar allí su contraseña. Teniendo en cuenta esto, se ha pensado e implementado una solución similar para la aplicación considerando que es una aplicación móvil y no se dispone de un servicio web.

16.2.2. Solución Implementada

Cuando un usuario selecciona la opción de recuperar contraseña, se le muestra una pantalla donde debe de introducir el correo electrónico con el que se registró. Una vez introducido este se envía al servidor donde se busca al usuario que tiene asignado dicho correo si existe. Si es así, el servidor genera un código aleatorio de seis dígitos alfanuméricos incluyendo mayúsculas que se guarda en la base de datos asignado a dicho usuario y con un tiempo de expiración de 30 minutos. De forma contraria se devuelve un error HTTP 404 determinando que no se ha encontrado el usuario.

Una vez el código se guarda, este se envía al correo electrónico del usuario, este debe acceder al correo que ha introducido y obtenerlo. Al mismo instante, el usuario es redirigido a una nueva pantalla donde se le pide introducir su nueva contraseña y el código que se ha enviado. Hecho esto, el contenido es enviado al servidor y este comprueba que el código enviado existe y no ha caducado. Si es así, la contraseña del usuario es cambiada por la enviada y el *token* de refresco del usuario es revocado. Si es al contrario, si el código no existe, se envía al cliente un error HTTP 404 y si se encuentra y ha caducado un error HTTP 410 determinando que ya no es válido.

Gracias a este método solo los usuarios que tengan acceso al correo electrónico son los que pueden cambiar la contraseña. De este modo ya no se puede producir ninguna brecha de seguridad en lo que respecta al cambio de contraseña porque otro usuario lo haya solicitado en nombre de otro. Además esta forma de hacerlo posibilita que si en un futuro se dispusiese de página web tan solo habría que modificar el código por una dirección web.

16.3. Subida y Descarga de Imágenes

En los siguientes subapartados se describe la implementación realizada para la subida y descarga de imágenes de los usuarios en la aplicación.

16.3.1. Planteamiento

En la mayoría de aplicaciones actualmente en el mercado presentan la funcionalidad de mostrar, subir y descargar imágenes de sus usuarios o de la propia aplicación. Algunos ejemplos de aplicaciones son las redes sociales como Facebook, Instagram o SnapChat. En el caso de este proyecto también se ha querido implementar estas funcionalidades.

Un método para hacerlo posible pero muy básico es el de almacenar las imágenes directamente en la base de datos construida en el servidor de recursos de la aplicación en formato BLOB normalmente destinado para imágenes. No obstante esta solución no es la más apropiada ya que el servidor no está preparado para poder almacenar ficheros de grandes extensiones y esto puede

repercutir en el rendimiento, provocar brechas de seguridad y gastar muchos recursos para poder gestionar dichos archivos. De forma similar pasaría si se almacenaran documentos. Además a la hora de descargar dichas imágenes podría provocar que la aplicación Android no actuase de forma fluida.

Para poder solucionar estos problemas se ha decidido utilizar un servicio que está destinado a trabajar con imágenes, Cloudinary. Este se encarga de almacenar dichas imágenes y protegerlas. En cuanto a las descargas y el tratamiento de las imágenes en la aplicación Android se utiliza una librería llamada Picasso.

16.3.2. Solución Implementada

Cuando un usuario decide editar su perfil, escoge una fotografía de su teléfono para que le identifique en la aplicación y pulsa el botón de guardar cambios, la imagen es comprimida y reducida al tamaño necesario en un hilo secundario para no bloquear al principal que se encarga de notificar al usuario del progreso. Una vez comprimida, esta es enviada al servidor donde la procesa y la envía al servicio Cloudinary el cual crea una carpeta para el usuario si no existía anteriormente y almacena la imagen en esta.

Una vez guardada, el servicio devuelve al servidor una URL en formato de texto con el cual se puede acceder a la imagen de forma directa. Esta URL se almacena en la base de datos en el apartado del usuario que ha editado su perfil para que pueda acceder a su imagen de forma directa. En este caso al tratarse de un texto breve si puede ser almacenado en la base de datos ya que no su tamaño no es excesivo. Además tenemos la garantía de que las imágenes están protegidas por Cloudinary ya que cumplen con la normativa GDPR de la Unión Europea.

A partir del momento en que se guarda el hipervínculo en la base de datos, los usuarios pueden obtener las imágenes gracias a la librería Picasso que hace todo el tratamiento necesario. Tal y como se comentó anteriormente en el apartado de Tecnologías Utilizadas, esta librería solo necesita la URL, que es enviada a los usuarios cuando es necesario a través de peticiones HTTP al servidor y Picasso se encarga de descargar la imagen en segundo plano y tratarla como es necesario. Picasso es muy útil ya que evita la descarga continua de imágenes proporcionando un método de cache y además se encarga de eliminar y liberar la memoria del dispositivo móvil cuando la imagen ya no es necesaria.

16.4. Listas de Anuncios y Usuarios

En los siguientes subapartados se explica la implementación realizada para la obtención y visualización de las listas de anuncios que están muy presentes en la aplicación.

16.4.1. Planteamiento

A lo largo de la aplicación se presentan listas de anuncios y usuarios con los que el usuario puede actuar. Normalmente se piensa en mostrar una lista de forma directa sin pensar en lo que esto supone. Para listas de elementos que son pequeñas pedir a un servidor que traiga una lista con todos sus elementos puede ser factible, no obstante, cuando estas listas ya superan los cien elementos

estas peticiones pueden ser costosas y provocar problemas de rendimiento tanto en la parte cliente como en el servidor.

Los usuarios no deben de esperar a una gran cantidad de tiempo a que se reciba el contenido ya que sino querrán dejar de utilizar la aplicación. Además, estas solicitudes pueden provocar problemas de seguridad ya que a mayor tamaño la petición más tarda y más posibilidades hay de que sea interceptada por algún intruso. Para solucionar estos problemas se ha decidido optar por diferentes soluciones.

La primera solución consiste en el uso de paginación. La paginación divide los elementos de la lista en páginas, de esta forma solo se cogen los elementos de la lista que el usuario quiere ver. Esto sin duda hace que la fluidez de la aplicación sea mucho más alta y soluciona los problemas mencionados anteriormente. No obstante, al tratarse de una aplicación móvil los usuarios no suelen estar acostumbrados a esta solución ya que prefieren realizar gestos como deslizamientos que son más comunes. Esto ha conducido a la segunda solución llamada *scroll* infinito que es la que se ha adoptado.

16.4.2. Solución Implementada

La técnica de *scroll* infinito se creó para dispositivos táctiles y parte de la paginación ya que también va trayendo los elementos de las listas que corresponden a lo que el usuario quiere ver o está viendo. La diferencia entre esta solución y la paginación es que en vez de que el usuario pulse un botón para avanzar a la siguiente página, el usuario una vez baja o sube lo suficiente de la lista, entonces, automáticamente la aplicación realiza la petición de la siguiente página y añade los elementos a lista. Esto permite que el usuario vaya desplazándose por la lista sin tener que actuar más que para bajar o subir de ella. Una vez ya no hay más elementos que recoger del servidor la lista acaba. Esta solución además de mejorar el rendimiento del servidor y del cliente también permite utilizar únicamente la memoria necesaria en la memoria del cliente y evitar que se malgaste.

En este proyecto la implementación de la parte de la paginación por parte del servidor es muy simple ya que el *framework* Spring Boot e Hibernate proporcionan esta característica de forma directa. Para poder utilizarla tan solo hay que hacer una petición a la base de datos con la página y los elementos que se desea. En este proyecto se ha escogido un total de diez elementos cada vez. Una vez pasados, automáticamente se devuelven los diez elementos así como el número de página siguiente si hay.

En el caso de la parte Android, esto ha resultado un poco más complicado ya que no se ha utilizado la librería de paginación que utilizan muchos desarrolladores sino que se ha implementado manualmente. Primero de todo se ha controlado cuando el usuario baja o sube lo suficiente la lista para poder realizar la petición al servidor para obtener nuevos elementos. Una vez hecho esto, cuando el usuario activa este proceso y se realiza la petición obteniendo así los elementos nuevos se han añadido a la lista ya existente y se ha consultado si hay más páginas. Si las hay entonces una vez que el usuario vuelve activar este evento se repite el proceso, sino se obvia. Cabe destacar que mientras se realiza la petición y el usuario llega al límite de la lista tanto por arriba como por abajo se ha añadido una barra de progreso que indica que se están obteniendo los elementos.

16.5. Conversación entre Usuarios

En los siguientes subapartados se describe la implementación realizada para las conversaciones entre usuarios a través de un chat.

16.5.1. Planteamiento

Una de las funcionalidades que más gusta a los usuarios y que más aplicaciones tienen es la de poder conversar con otros usuarios. En este proyecto se ha querido realizar también un chat, una primera idea era el ir almacenando los mensajes que se enviaban los usuarios en la base de datos. Este sin duda es un método lógico pero el problema surgía al obtener los mensajes y hacer que la conversación se estuviese haciendo en tiempo real. Al no disponer de una base de datos a tiempo real esto no se podía realizar salvo haciendo una petición cada determinado tiempo pero esto no es una buena solución.

Otra idea era la de implementar sockets, no obstante, al no tener demasiado conocimiento sobre estos se decidió dejarlo a un lado y buscar otra solución más al alcance. Por ello se pensó en el uso del servicio Firebase que dispone de una base de datos a tiempo real donde los usuarios con permisos pueden leer datos cada vez que se produce un cambio.

16.5.2. Solución Implementada

Para poder implementar esta solución con Firebase primero se ha tenido que crear un proyecto nuevo en dicho servicio y registrar a los usuarios de la aplicación para que tengan los suficientes privilegios para poder leer de la base de datos. Debido a esto, cuando un usuario se registra en la aplicación el servidor que se encarga de comunicarse con los diferentes servicios como se comentó en el apartado de la Visión General del Sistema, también registra al usuario en Firebase con su correo electrónico y su contraseña.

Una vez los usuarios ya están registrados estos ya pueden comunicarse entre sí pulsando el botón de chat que se encuentra en los anuncios. Cuando se envía el primer mensaje entre dos usuarios, el servidor crea en la base de datos de Firebase una entrada con el anuncio y los usuarios asignados a la conversación. A partir de ese momento cada mensaje que se envía se añade a dicha entrada con el identificador del usuario que lo envía y el identificador de quien lo recibe. De esta forma la aplicación Android puede escuchar los mensajes que se van añadiendo a dicha entrada a tiempo real y así actualizar el historial según se va escribiendo.

Debido a que puede darse el caso en que un usuario envíe un mensaje y cierre la aplicación antes de que se complete la acción se ha creado un servicio Android para poder realizar el envío. Un servicio en Android es una clase que se ejecuta en segundo plano ya sea con la aplicación abierta o cerrada de forma que en este caso se envía siempre el mensaje a menos que no haya conexión a Internet o haya ocurrido algún problema. Además, gracias a que los servicios pueden recibir colas de peticiones, en este caso de mensajes, el servicio solamente parará cuando todos los mensajes se envíen o se produzca un error facilitando así el envío sin tener que crear implementaciones complejas de colas.

Además, debido a que los mensajes se almacenan en una base de datos externa al servidor, se ha creado una tabla en la base de datos del proyecto donde se mantiene la información del estado de la conversación. Algunos de estos atributos son que usuario ha cerrado la conversación, quien ha visto el último mensaje, el último mensaje o la fecha de actualización. De esta forma cuando el usuario entre a la pantalla de sus conversaciones podrá ver que conversaciones tiene pendiente por ver de forma ordenada, el último mensaje enviado etc.

También cabe comentar que cuando un anuncio se cierra o un usuario elimina su cuenta todos los datos de las conversaciones o usuarios se eliminan de forma que siempre se mantiene actualizado y sincronizada la base de datos de Firebase con la base de datos del servidor.

Por último comentar que al igual que con las listas de anuncios se ha implementado la técnica de *scroll* infinito en el historial de las conversaciones el cual se recupera al entrar a la conversación. Cuando un usuario entra a la conversación y existe un historial de mensajes previo este carga tan solo veinte mensajes. Así, el historial se va cargando a medida que el usuario va desplazando hacia arriba la conversación.

16.6. Conversaciones del Usuario

En los siguientes subapartados se explica la implementación de la pantalla que contiene la lista de conversaciones del usuario.

16.6.1. Planteamiento

Al igual que con las listas comentadas anteriormente, la lista de conversaciones de un usuario no estaría de más implementarla con *scroll* infinito, no obstante, esta vez no se ha implementado ya que suponía renunciar a otras funcionalidades. Estas funcionalidades que se mencionan son características que presentan aplicaciones con un chat como por ejemplo WhatsApp, como puede ser la actualización automática de las conversaciones al llegar un nuevo mensaje o los iconos de nuevos mensajes. Al implementarlas con paginación resultaba muy dificultoso para la duración del proyecto y por tanto se ha descartado para esta lista.

16.6.2. Solución Implementada

Para dar la sensación de un chat real similar a otras aplicaciones, se ha querido otorgar a la lista indicativos de nuevos mensajes y actualizaciones de datos. La lista de conversaciones es actualizada cada vez que un usuario que se encuentra dentro de una conversación vuelve a la pantalla de sus conversaciones. Dicha lista actualiza la conversación que ha cambiado actualizando la fecha del último mensaje recibido, el último mensaje recibido y si se encontraba el ícono de nuevo mensaje este desaparece. También mueve la conversación hacia la parte más alta de la lista de manera que esta esté ordenada por conversaciones más recientes a menos recientes.

La lista de conversaciones no solo es actualizada cuando el usuario vuelve de un chat sino que también es actualizada cada vez que llega una notificación de un nuevo mensaje. Sin entrar en

mucho detalle ya que la implementación de las notificaciones se comentará a continuación cuando el servicio que se encarga de capturar dichas notificaciones detecta una nueva notificación de tipo conversación, entonces, se comunica a la pantalla de conversaciones si está activa que debe actualizarse. De esta forma se consigue lo que se quería realizar que es una implementación a tiempo real similar a otras aplicaciones con los recursos que se dispone.

16.7. Notificaciones

En los siguientes subapartados se explica la implementación realizada para el envío y el trato de notificaciones a los diferentes usuarios de la aplicación.

16.7.1. Planteamiento

Para poder enviar notificaciones a los usuarios de la aplicación es necesario que el servidor pueda enviar información cuando sea necesario. Una posible pero mala solución sería crear un servicio Android de forma similar a como se ha hecho para poder enviar mensajes entre usuarios que siempre estuviese activo ya sea con la aplicación encendida o apagada. Este servicio solicitaría al servidor cada cierto tiempo si el usuario tiene alguna notificación pendiente. Si es así esta se mostraría y repetiría este proceso al cabo del tiempo estimado. Debido a que esta solución parece poco apropiada se ha aprovechado una vez más la posibilidad de utilizar el servicio Firebase que dispone de envío de notificaciones a tiempo real.

16.7.2. Solución Implementada

Para poder utilizar este servicio que ofrece Firebase ha sido necesario almacenar en la base de datos un *token* que genera Firebase una vez la aplicación se instala por primera vez. Este *token* es único e identifica al dispositivo Android. Es seguro pues en caso de fallo de seguridad con eliminar la aplicación o borrar los datos del dispositivo Android el *token* es invalidado automáticamente. Cuando un usuario se registra a la aplicación se obtiene el *token* que es proporcionado por Firebase y es enviado al servidor donde se guarda. También, cuando el usuario inicia sesión se recupera el *token* y se vuelve a enviar para substituirlo por si ha cambiado.

Una vez el *token* es guardado, el servidor ya puede utilizar la API proporcionada por Firebase para poder enviar notificaciones a cada usuario de forma separada. No obstante, al igual que como se comentó en el planteamiento de este apartado es necesario crear un servicio en la parte Android que se dedique a escuchar los eventos producidos tanto si la aplicación está encendida como apagada. A diferencia del servicio comentado en el planteamiento este no realiza ninguna petición sino que solo se dedica a escuchar, consumiendo muchos menos recursos. Cuando una notificación llega al dispositivo esta es capturada y es mostrada permitiendo al usuario ir directamente a la pantalla pulsando sobre ella.

Este método ha permitido que las notificaciones relacionadas con el seguimiento de usuarios, de valoraciones recibidas y de nuevos mensajes sea muy simple. En cambio, para las notificaciones de anuncios cercanos se ha tenido que implementar un poco más de código. Estas notificaciones deben de enviarse a todos los usuarios de la aplicación y mostrarse únicamente si el usuario se encuentra

cerca del punto de creación del anuncio, por lo tanto, el enviar notificaciones a dispositivos de uno en uno no es buena idea. Para ello Firebase proporciona otro método de envío basado en temas.

Cuando los usuarios se registran o inician sesión al igual que enviaban el *token* y el servidor lo almacenaba, el servidor se encarga de suscribir automáticamente los dispositivos al tema de “anuncios cercanos” realizando una petición a Firebase. Gracias a esto, cuando se crea un anuncio nuevo se envía una única petición al servidor de Firebase y este último se encarga de enviar las notificaciones a los dispositivos Android. Cuando la notificación llega al dispositivo Android, el servicio se encarga de capturarla como a las demás y comprueba si la localización del usuario es cercana siempre que la localización esté activado o las notificaciones no estén desactivadas.

Todas las notificaciones proporcionan un acceso directo a la pantalla donde se puede consultar la información a la cual hacen referencia. Cabe destacar la implementación de las notificaciones del chat tal y como se comentó en el apartado anterior hablando de la lista de conversaciones. Cuando la aplicación está encendida o en reposo y el usuario se encuentra en la pantalla de la lista de conversaciones estas se actualizan de manera automática gracias a la comunicación proporcionada entre el servicio Android y la actividad. Además, cuando un usuario entra en un conversación, si la notificación hace referencia al mismo chat, dicha notificación es automáticamente eliminada conservando las demás. Esto no es automático y debe controlarse. En este proyecto se ha conseguido controlar a partir de un contador de notificaciones que se actualiza a medida que llegan y se eliminan ya sea por deslizamiento de estas o presionando sobre ellas.

17. Pruebas del sistema

Para producir y obtener un *software* que sea robusto y de esta forma minimizar los errores que puedan ocurrir es necesario realizar distintas pruebas del código del sistema. En este apartado se comenta las diferentes pruebas que se han realizado para la parte del servidor y la parte del cliente de la aplicación.

17.1. Pruebas del Cliente (*Frontend*)

Las pruebas de la parte cliente consisten primero de todo en pruebas de estrés para comprobar que la interfaz responde correctamente y esta no es bloqueada. También se realizan pruebas con usuarios reales, en específico los mismos usuarios que ayudaron a la obtención de los requisitos del proyecto. A continuación se describe en más profundidad estos dos tipos de pruebas.

17.1.1. Monkey Testing

Monkey Testing [49] es una técnica donde el usuario comprueba la aplicación mediante la creación de entradas aleatorias y de esta forma comprobar el comportamiento y la estabilidad de la aplicación. Esta técnica es útil para buscar errores en interfaces los cuales suelen pasar desapercibidos porque su aparición depende de otros eventos con la interfaz o simplemente se han pasado por alto. Debido a esto y a la falta de tiempo para implementar pruebas unitarias, se ha preferido realizar este tipo de pruebas que permitan detectar dichos errores y comprobar si la aplicación es realmente sólida.

17.1.2. Pruebas con Usuarios

Para estas pruebas se ha ido entregando la aplicación de forma fragmentada a una serie de compañeros que han actuado como beta testers para poder obtener comentarios de aquello que no funcionaba correctamente o que no era suficientemente claro. Al finalizar estas pruebas se ha recogido la siguiente información de los comentarios recibidos para la mejora de la aplicación:

- **Fallos en la interfaz:** Se han recopilado todos los fallos que han encontrado durante las pruebas en lo que respecta a la interfaz. Algunos errores pueden ser que un campo de texto estuviese deshabilitado o que las animaciones no se realizasen correctamente. Estos errores han sido solucionados completamente.
- **Fallos en el comportamiento:** Se han recopilado todos los fallos que han encontrado durante las pruebas en los que respecta a como se comporta la aplicación. Algunos de los errores proporcionados se centraban en el control de los errores. Por ejemplo cuando el usuario no introducía correctamente un campo y debía salir un mensaje de error o que una pantalla no llevaba a la que correspondía. Estos errores han sido solucionados completamente.
- **Posibles mejoras:** Se han recopilado todas las posibles mejoras que han comentado los usuarios. Algunas de estas mejoras tienen que ver con la estética de la aplicación o de temas más indicados a la implementación como el uso de paginación en ciertas listas. La mayoría de las mejoras se han implementado correctamente adaptándolas a la planificación sin que la alterasen.

17.2. Pruebas del Servidor (*Backend*)

Las pruebas del servidor consisten en la realización de pruebas unitarias de cada parte del patrón Modelo-Vista-Controlador mediante el uso de las tecnologías para pruebas comentadas en el apartado Tecnologías utilizadas de este documento. A continuación se describen las diferentes pruebas que se han realizado para la comprobación del código de la parte del servidor de la aplicación.

17.2.1. Pruebas del Modelo

Las operaciones que se deben de llevar a cabo para la manipulación de la base de datos como almacenamiento, modificación u obtención puede conducir a errores críticos provocando que el sistema falle completamente. Para comprobar que la parte del modelo de la parte del servidor funciona correctamente y no se producen errores indeseados se han realizado diferentes pruebas unitarias. Esto se ha realizado a partir de las herramientas JUnit, Mockito y la base de datos H2 mencionadas en el apartado de Tecnologías Utilizadas de este documento.

Primero se han realizado pruebas unitarias para comprobar el correcto funcionamiento de las entidades de la base de datos. También se han comprobado las sentencias SQL utilizadas en el código para poder acceder y realizar operaciones sobre las entidades. De tal forma, se han validado los siguientes aspectos:

- **Presencia de los atributos de las entidades:** Se ha comprobado que las entidades contienen los atributos obligatorios de cada una. Por ejemplo, una entidad que represente a un usuario debe de tener un nombre de usuario, una contraseña y un correo de forma obligatoria. En caso contrario se devuelve una excepción que indica que no puede existir dicha entidad.
- **Formato de los atributos de las entidades:** Se ha comprobado que los atributos de las entidades tienen el correcto formato. Por ejemplo, una contraseña debe de contener al menos 8 caracteres alfanuméricos. En caso contrario se devuelve una excepción que indica que no puede existir dicha entidad.
- **Asociaciones de las entidades:** Se ha comprobado que las asociaciones de cada entidad son correctas. Por ejemplo, un usuario debe de tener un único *token* de refresco y nunca más de uno pero, en cambio, un usuario puede tener un código de recuperación de contraseña o no tenerlo. Al no cumplirse estas restricciones se devuelve una excepción que indica dicho problema.
- **Restricciones de las entidades:** Se ha comprobado que las restricciones de cada entidad son correctas. Por ejemplo, no puede haber dos usuarios con el mismo nombre usuario. En caso contrario se devuelve una excepción que indica el problema ocurrido.
- **Sentencias SQL:** Se ha comprobado que las sentencias SQL que se utilizan son correctas. Por ejemplo, se ha comprobado que la sentencia SQL que permite saber si existe una entidad que representa a un usuario obtiene dicha entidad si esta existe o en caso contrario se lanza una excepción.

17.2.2. Pruebas del Controlador/Servicio y de Integración

Al igual que la comprobación del correcto funcionamiento de las entidades de la base de datos y de las operaciones sobre estas, es importante comprobar el correcto funcionamiento de los controladores que se encargan de recibir las peticiones al servidor y de los servicios que la procesan. Para ello se han realizado pruebas unitarias para los controladores y los servicios y también pruebas de integración como si el producto ya estuviese en producción. Al igual que en el caso anterior se han comprobado a través de las herramientas JUnit, Mockito y la base de datos H2 mencionadas en el apartado de Tecnologías Utilizadas de este documento. Se han validado los siguientes aspectos:

- **Códigos de respuesta:** Para cada petición se comprueba que el código de respuesta que se devuelve sea el indicado. Por ejemplo, si se hace una petición para iniciar sesión indicando un usuario que no existe se debe devolver un error 404. Si se hace una petición del mismo tipo pero esta vez indicando un usuario pero una contraseña errónea se debe devolver un error 401.
- **Presencia de los atributos de las peticiones:** Para cada petición, al igual que pasa con las pruebas del modelo, se comprueba que estas peticiones contengan todos los atributos necesarios y si no es el caso devolver el error indicado. Por ejemplo en la petición de inicio de sesión es necesario y obligatorio que lleguen los atributos de usuario o correo electrónico y contraseña. Si no es así se devuelve un error 400.
- **Formato de los atributos de las peticiones:** Para cada petición, al igual que pasa con las pruebas del modelo, se comprueba que los atributos tienen el formato correcto y si no es el caso devolver el error necesario. Por ejemplo en el caso de inicio de sesión se debe enviar una contraseña con al menos 8 caracteres alfanuméricos, en caso contrario se debe devolver un error 400.
- **Reglas de negocio:** Para cada petición, se ha comprobado que cada servicio que se utiliza funciona correctamente y cumple las restricciones impuestas por la aplicación. Por ejemplo en el caso de una petición de registro de usuario con un nombre de usuario solo se puede crear un usuario cuando el nombre de usuario y el correo no existen ya en la base de datos. En caso contrario el servicio devuelve una excepción que el controlador acaba traduciendo en un código de respuesta en este caso 409 indicando que hay conflicto ya que ya existe.

18. Identificación de Leyes y Regulaciones

La aplicación puede verse afectada por diversas leyes tanto europeas como estatales. En este apartado se describen aquellas que afectan y que se deberían cumplir si se quisiese sacar al mercado la aplicación. Cabe destacar que no se describe en profundidad las leyes y normativas ya que no se tiene pleno conocimiento de estos apartados. Lo mejor sería contactar con un profesional.

18.1. General Data Protection Regulation (GDPR)

La General Data Protection Regulation (GDPR) [50] es la normativa europea que regula la protección de los datos de los ciudadanos residentes en la Unión Europea. Entró en vigor el 24 de Mayo de 2016 y pasó a ser de obligado cumplimiento el 25 de Mayo de 2018. Esta normativa se aplica a todas las empresas que recoge, guardan, tratan, usan o gestionan datos de los ciudadanos de la Unión Europea independientemente de donde pertenezca la empresa.

En este proyecto se toman datos como el correo electrónico para el registro que se utiliza para poder recuperar la contraseña del usuario o para poder iniciar sesión. También se puede recoger fotografías del usuario si este quiere utilizar una para mostrarla en la aplicación. Por último se recogen datos como la localización del usuario actual para poder proporcionar a cuanta distancia se encuentra un anuncio de él pero estos datos no se almacenan en ningún lugar. En cuanto a datos de ubicación solo se almacenan aquellos que los usuarios seleccionan en la creación de un anuncio que pueden o no corresponder a localizaciones personales del usuario.

Debido a esta toma de datos la aplicación debe de cumplir la normativa GDPR si se quisiese lanzar al mercado. Para ello se debería de cumplir los derechos que determina la normativa que deben de tener los usuarios. Estos son mencionados a continuación de forma resumida:

- **Derecho a estar informado.** El usuario debe de saber que datos van a ser recolectados, para qué, como serán almacenados y procesados.
- **Derecho al acceso.** Una vez recolectados, los usuarios tienen derecho a poder obtener sus datos, como se han procesado, como se han almacenado y con qué propósito.
- **Derecho a la corrección.** El usuario tiene derecho a corregir aquellos datos que sean incorrectos o incompletos.
- **Derecho a la eliminación.** El usuario tiene derecho a eliminar sus datos personales de la aplicación como de los servicios de tercero que esta utilice.
- **Derecho a restringir el procesamiento.** El usuario tiene derecho a bloquear y suprimir sus datos en el momento que se están procesando o haciendo uso.
- **Derecho a la portabilidad de datos.** El usuario tiene derecho a copiar, mover o transferir sus datos de un controlador a otro de una forma segura, en un formato legible y comúnmente usado.
- **Derecho a objetar el procesamiento.** El usuario tiene derecho a objetar el hecho de ser sujeto para autoridades públicas o empresas que procesen sus datos sin un consentimiento explícito.

- **Derecho a no ser sujeto de toma de decisiones automatizadas.** El usuario tiene derecho a solicitar intervención humana en la toma de decisiones en vez de ser objetivo de la toma de decisiones por algoritmos.

Además del cumplimiento de estos derechos también hay que asegurarse de que si la aplicación utiliza servicios de terceros estos cumplan la normativa. En este caso se utiliza Firebase y Cloudinary los cuales cumplen la GDPR.

18.2. Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD)

De forma similar a la GDPR pero a nivel estatal se encuentra la Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD) [51]. Esta ley apareció el 6 de Diciembre de 2018 de manera oficial den el Boletín Oficial del Estado y sustituye a la anterior ley LOPD. Por el mismo motivo que se debe cumplir la GDPR para un lanzamiento al mercado también se debe cumplir esta. A continuación se describen algunos apartados del documento que suponen una novedad respecto a la ley anterior:

- **Acceso a datos de fallecidos (Art. 3).** Aquellas personas cercanas o familiares de una persona fallecida podrán solicitar la rectificación o eliminación de los datos de la persona fallecida.
- **Edad de consentimiento de menores (Art. 7).** La edad mínima en que una persona puede facilitar sus datos personales se estable en la edad de 14 años. En caso contrario deberá de ser consentido por el titular del menor.
- **Bloqueo de datos (Art. 32).** En el caso de que subsista una obligación legal, en lugar de suprimir los datos estos serán bloqueados durante el tiempo que el responsable esté obligado.

18.3. Impuesto Sobre Sucesiones y Donaciones

En la aplicación los usuarios pueden pedir ayuda a través de anuncios y ofrecer a aquellas personas que les ayuden algo a cambio como un trato de ayuda por ayuda u ofrecer algún tipo de recompensa monetaria para poder así motivar aún más a los usuarios a ayudar. Por ello, aunque la aplicación no se encarga de llevar a cabo la donación ni a ofrecer un servicio para hacerlo, no está de más notificar a sus usuarios de que deben cumplir con la ley y pagar el impuesto sobre sucesiones y donaciones si reciben una recompensa monetaria.

El impuesto sobre sucesiones y donaciones [52] es el impuesto que toda persona receptora debe de pagar al recibir una herencia o una donación. Cada comunidad autónoma tiene su impuesto de donaciones por lo que cada usuario que reciba una recompensa monetaria debe de atenerse a su respectiva comunidad autónoma y pagar dicho impuesto. En el caso de ONGs que se mencionaron en la documentación inicial están exentas por la Ley 49/2002 del 23 de diciembre del pago por donaciones. Esto es debido a que persiguen fines calificados de interés general como la cooperación al desarrollo o la asistencia social.

19. Planificación y Presupuesto Final

A lo largo de este apartado se comentan las diferentes desviaciones temporales producidas en el proyecto así como la planificación final y se comenta el presupuesto final. También se muestra una tabla resumen de las horas estimadas y reales por cada tarea así como el diagrama de Gantt final del proyecto donde se muestran los cambios temporales.

19.1. Desviaciones Temporales

A lo largo del proyecto no se han producido desviaciones que hayan perjudicado en gran medida al transcurso del proyecto. A pesar de esto si que ha habido cambios en algunas fechas y duración de Sprints pero que no han repercutido en el total de horas del proyecto.

Primero de todo, la fase de implementación tuvo que empezar más tarde de lo planificado inicialmente debido a que la defensa de la fase inicial se realizó más tarde de lo previsto. No obstante, debido a que todas las tareas de la primera fase estaban finalizadas correctamente según la planificación se pudo realizar de forma la fase de implementación como estaba previsto y realizar la defensa más tarde.

En lo que respecta a la fase de implementación se ha seguido la planificación indicada pero algunos Sprints han sufrido cambios en sus horas. El segundo Sprint tuvo que ampliar su número de horas retrasando el inicio de los demás. Esto fue debido a que las funcionalidades a realizar esas semanas llevaron más tiempo de lo previsto. En lo que respecta a los demás Sprints se han completado aproximadamente en las horas previstas a excepción de los dos últimos que han tenido un número menor de horas. Estos cambios no han repercutido demasiado ya que el total de horas ha sido de 319 en comparación a las 315 horas previstas inicialmente.

Por último, en la fase de cierre se han cumplido las horas previstas a excepción de la revisión ya que se ha hecho de forma muy rápida gracias a la ayuda de la directora.

Por lo tanto, el desarrollo del proyecto ha ido en gran medida a la par con la planificación inicial que se hizo al principio del proyecto. La planificación final tan solo se ha visto modificada por incrementos y decrementos de horas en la fase de implementación pero sin llegar a afectar en gran medida al desarrollo del proyecto.

19.2. Tabla Resumen de Horas por Tarea

En la siguiente tabla se puede encontrar un resumen de las horas estimadas durante la fase de planificación así como las reales de cada tarea. Los cambios a destacar se han mencionado anteriormente y corresponden principalmente a las horas de implementación de los Sprints. Los Sprints dos y tres han incrementado el número de horas mientras que el cuatro y el cinco han disminuido.

Como se puede ver al final de esta, el total del número de horas del proyecto ha sido de 556,5 aproximadamente, un valor muy aproximado al estimado de 560,5 horas.

Tarea	Horas estimadas	Horas reales
Análisis, contexto y alcance	17,5	17,5
Planificación temporal	10,0	10,0
Presupuesto y sostenibilidad	12,5	12,5
Preparación presentación, documento final	27,5	27,5
Defensa fase inicial	2,0	2,0
Reunión #1	2,0	2,0
Definición del Backlog	22,5	22,5
Diseño base de datos	12,5	12,5
Diseño <i>backend</i>	12,5	12,5
Diseño <i>frontend</i>	12,5	12,5
Diagramas de secuencia	10,0	10,0
Reunión #2	2,0	2,0
Sprint #1	63,0	63,0
Reunión #3	2,0	2,0
Sprint #2	63,0	76,5
Reunión #4	2,0	2,0
Sprint #3	63,0	63,5
Reunión #5	2,0	2,0
Sprint #4	63,0	57,0
Reunión #6	2,0	2,0
Sprint #5	63,0	59,0
Reunión #7	2,0	2,0
Finalización de la memoria	30,0	30,0
Revisión de la memoria	18,0	10,0
Preparación presentación y demostración	42,0	42,0
Defensa del proyecto	2,0	2,0
Total	560,5	556,5

Leyenda	
	Fase de planificación y especificación de requisitos
	Fase de implementación
	Fase de cierre

Tabla 58: Tabla resumen de horas por tarea.

19.3. Diagrama de Gantt Final

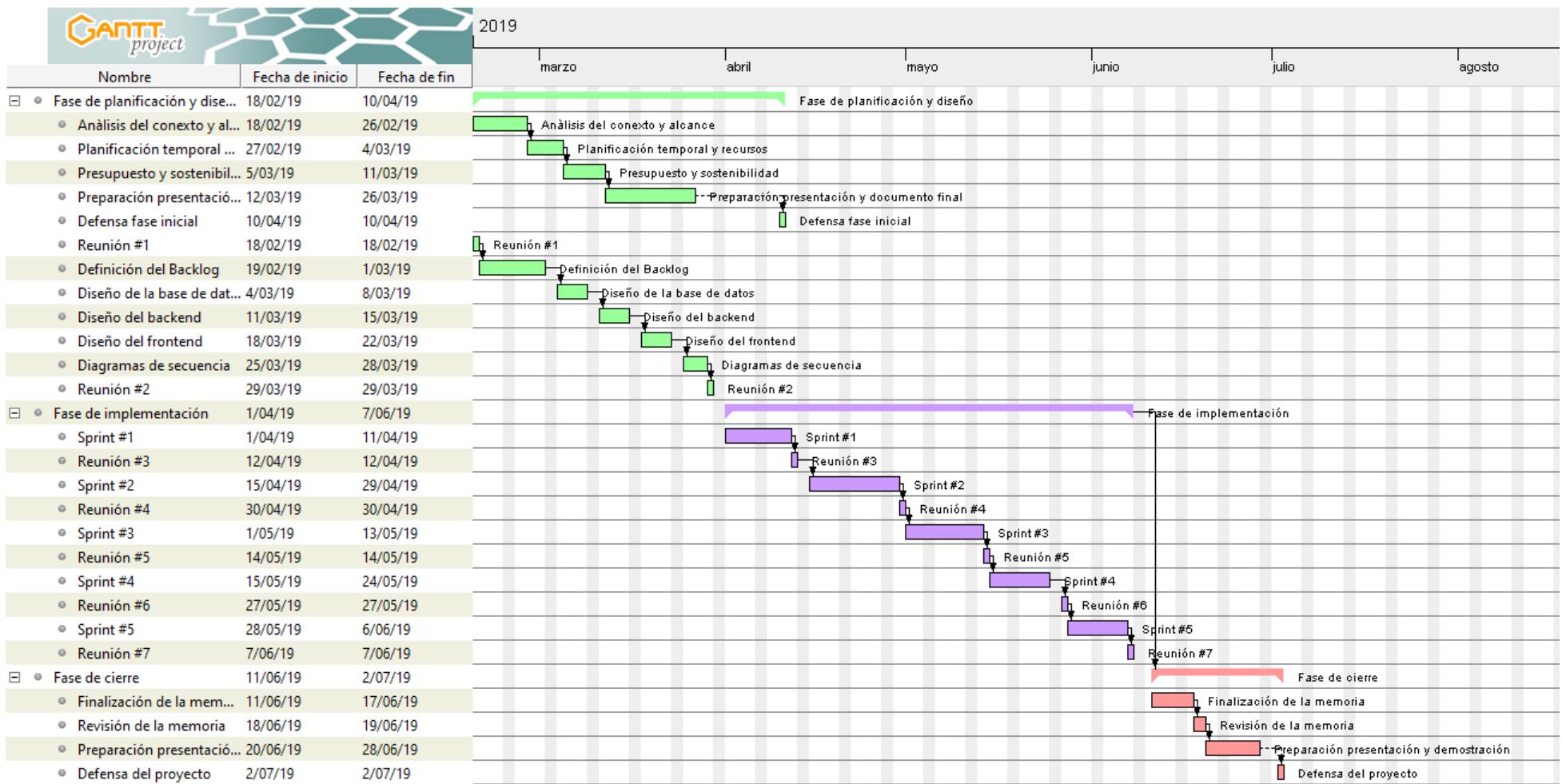


Figura 42: Diagrama de Gantt final.

19.4. Presupuesto Final

El presupuesto final se puede ver en la siguiente tabla. No se ha producido ningún imprevisto y no ha habido necesidad de utilizar el presupuesto de contingencia. Como se puede observar el presupuesto real es inferior al estimado. También cabe destacar que el presupuesto de recursos humanos, que forma parte de los directos, ha cambiado un poco y ha resultado ser menor, en total 10.840,6 € debido al cambio de horas mencionado anteriormente.

Tipo de Coste	Presupuesto estimado (€)	Presupuesto real (€)
Directos	12.951,4	12.258,5
Indirectos	851,7	851,7
Imprevistos	35,0	0
Contingencias	2.070,5	0
Total	15.908,6	13.110,2

Tabla 59: Tabla presupuesto final.

20. Conclusiones

En el siguiente apartado se hace una valoración del resultado del proyecto y como ha transcurrido y se comenta posibles mejoras o cambios que se pueden realizar en un futuro próximo.

20.1. Valoración Personal

El proyecto ha finalizado con la incorporación de todas las funcionalidades que se habían comentado en el apartado de Alcance del Proyecto. La realización de este ha sido fluido y se ha obtenido mucha experiencia realizándolo. Además, se han podido plasmar conocimientos que se han adquirido durante el grado universitario. También se han puesto en práctica técnicas que antes no se habían realizado nunca. Por ejemplo la funcionalidad del chat o el estado de sesión a partir de *tokens*. Gracias a la implementación de estas técnicas se ha podido también aprender sobre nuevas tecnologías y como utilizarlas.

20.2. Trabajo Futuro

A pesar de haber finalizado el proyecto aún hay funcionalidades que pueden modificarse en un futuro para mejorarlas o implementar nuevas para expandir la aplicación. También estaría bien corregir y estructurar código que no haya quedado suficientemente entendible para otros desarrolladores. A continuación se muestra una lista de algunas opciones para un trabajo futuro ordenadas de más a menos prioritarias.

- Aumentar la seguridad, por ejemplo mediante la contratación de un certificado de confianza o un dominio y el uso de Let's Encrypt para poder hacer uso de HTTPS que es más seguro que HTTP. El uso de HTTPS debe de ser obligatorio aunque actualmente no se usa, ya que la contratación del certificado de un dominio supone un coste monetario.
- Añadir funcionalidades como la recomendación de usuarios y mejorar las existentes.
- Traducción a más idiomas a parte de los tres presentes.
- Cumplir completamente las normativas y leyes tanto a nivel estatal como europeo que afectan a la aplicación.
- Inclusión de anuncios en caso de querer monetizar y publicación en la Play Store.

21. Bibliografía

- [1] "Smartphone", *Wikipedia*, 2019. [En línea].
Disponible en: <https://en.wikipedia.org/wiki/Smartphone>.
[Accedido: 21-02-2019]
- [2] "The History and Evolution of Mobile Apps", *Acodez: codes, ads, designs*, 2018. [En línea].
Disponible en: <https://acodez.in/evolution-mobile-apps/>.
[Accedido: 21-02-2019]
- [3] "Digital 2019: Global Internet Use Accelerates", *We Are Social*, 2019. [En línea]. Disponible en: <https://wearesocial.com/blog/2019/01/digital-2019-global-internet-use-accelerates>.
[Accedido: 21-02-2019]
- [4] "Impossible people", *Impossible people*, [En línea].
Disponible en: <https://www.impossible.com/impossible-community-app> .
[Accedido: 22-02-2019]
- [5] "FavForMe", *FavForMe*, n.d. [En línea].
Disponible en: <https://favforme.com/>.
[Accedido: 22-02-2019]
- [6] "Helpy", *Helpy*, n.d. [En línea].
Disponible en: <http://www.helpyapp.es/>.
[Accedido: 22-02-2019]
- [7] "Obsso", *Obsso*, n.d. [En línea].
Disponible en: <http://obsso.com/>.
[Accedido: 22-02-2019]
- [8] "¿Tienes Sal?", *¿Tienes Sal?*, n.d. [En línea]
Disponible en: <https://tienes-sal.es/>.
[Accedido: 22-02-2019]
- [9] Yu, L., Wooi, L., Wai, T & Soo, F., "Software Development Life Cycle AGILE vs Traditional Approaches", en International Conference on Information and Network Technology, Chennai, India, 2012. [En línea].
Disponible en:
<https://pdfs.semanticscholar.org/69b1/9ddc8a578f4c63d1dfe15252a465ee12fe5d.pdf>.
[Accedido en: 23-02-2019]
- [10] "Scrum en la metodología Ágil: te explicamos qué es y cómo funciona", *Clouding.io*, 2016. [En línea]. Disponible en: <https://clouding.io/blog/scrum-en-la-metodologia-agil-te-explicamos-que-es-y-como-funciona/>.
[Accedido: 23-02-2019]
- [11] "Bitbucket: What is Bitbucket?", *Atlassian Support*, 2018. [En línea].
Disponible en: <https://confluence.atlassian.com/confeval/development-tools-evaluator-resources/bitbucket/bitbucket-what-is-bitbucket>.
[Accedido: 23-02-2019]

[12] “Gitflow Workflow”, *Atlassian Bitbucket*, [En línea].
Disponible en: <https://es.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>.
[Accedido: 23-02-2019]

[13] “¿Qué es Trello?”, *Trello Help*, 2017. [En línea].
Disponible en: <https://help.trello.com/article/708-what-is-trello>.
[Accedido en: 24-02-2019]

[14] “The Definition of Done”, *Manifesto*, 2015. [En línea].
Disponible en: <https://manifesto.co.uk/definition-done/>.
[Accedido en: 24-02-2019]

[15] “What is Heroku”, *Heroku*, 2019. [En línea]
Disponible en: <https://www.heroku.com/what>.
[Accedido: 28-02-2019]

[16] “DigitalOcean”, *Wikipedia*, 2018. [En línea]
Disponible en: <https://es.wikipedia.org/wiki/DigitalOcean>.
[Accedido: 28-02-2019]

[17] “Qué es Adobe Illustrator?”, *Adobe*, 2019. [En línea]
Disponible en: <https://helpx.adobe.com/es/illustrator/how-to/what-is-illustrator.html>.
[Accedido: 28-02-2019]

[18] “Conoce Android Studio”, *Developers*, n.d. [En línea]
Disponible en: <https://developer.android.com/studio/intro/?hl=es-419>.
[Accedido: 28-02-2019]

[19] “What is Eclipse?”, *Help - Eclipse Platform*, n.d. [En línea]
Disponible en: https://help.eclipse.org/mars/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Fguide%2Fint_eclipse.htm.
[Accedido: 28-02-2019]

[20] Jose Ángel, Z., “¿Qué es Firebase? La mejorada plataforma de desarrollo de Google”, *El android libre*, 2016. [En línea]
Disponible en: <https://elandroidelibre.elespanol.com/2016/05/firebase-plataforma-desarrollo-android-ios-web.html>.
[Accedido: 28-02-2019]

[21] Félix, R., “Postman: gestiona y construye tus APIs rápidamente”, *Tecnología para desarrollo*, 2017. [En línea]
Disponible en: <https://www.paradigmadigital.com/dev/postman-gestiona-construye-tus-apis-rapidamente/>.
[Accedido 28-02-2019]

[22] “Tendencias del mercado laboral”, *PageGroup Tecnología*, 2018. [En línea].
Disponible en: https://www.pagepersonnel.es/sites/pagepersonnel.es/files/PG_ER_IT_2018.pdf.
[Accedido: 07-03-2019]

[23] “Precios y abonos de Adobe Cloud”, *Adobe*, 2019. [En línea].

Disponible en: <https://www.adobe.com/es/creativecloud/plans.html?promoid=CJ132JRD&mv=other>.
[Accedido: 07-03-2019]

[24] “Pricing on DigitalOcean - Cloud virtual machine & storage pricing”, *DigitalOcean*, 2019. [En línea].

Disponible en: <https://www.digitalocean.com/pricing/>.

[Accedido: 07-03-2019]

[25] “Precio tarjeta T-10 metro bus Barcelona”, *TMB*, 2019. [En línea]

Disponible en: <https://www.tmb.cat/es/tarifas-metro-bus-barcelona/sencillos-e-integrados/t-10>.

[Accedido: 07-03-2019]

[26] “Factor de emisión asociado a la energía eléctrica: el mix eléctrico”, *Generalitat de Catalunya*, 2019. [En línea]

Disponible en: http://canviclimatec.gencat.cat/es/reduceix_emissions/com-calcular-emissions-de-geh/factors_demissio_associats_a_lenergia/.

[Accedido: 07-03-2019]

[27] A. Durán, B. Bernárdez, “Metodología para la Elicitación de Requisitos de Sistemas Software”, *Universidad de Sevilla*, 2000. [En línea]

Disponible en: <http://www.lsi.us.es/docs/informes/lsi-2000-10.pdf>

[Accedido: 11-05-2019]

[28] J. T.Emmaty, “Differences between MVC and MVP for Beginners”, *Code Project*, 2011. [En línea]

Disponible en: <https://www.codeproject.com/Articles/288928/Differences-between-MVC-and-MVP-for-Beginners>

[Accedido: 11-05-2019]

[29] “Singleton Pattern”, *Object Oriented Design*, n.d. [En línea]

Disponible en: <https://www.oodesign.com/singleton-pattern.html>

[Accedido: 11-05-2019]

[30] “Conoce Android Studio”, *Developers*, n.d. [En línea]

Disponible en: <https://developer.android.com/studio/intro>

[Accedido: 11-05-2019]

[31] “Eclipse”, *Eclipse*, n.d. [En línea]

Disponible en: <https://www.eclipse.org/eclipseide/>

[Accedido: 11-05-2019]

[32] “Android SDK”, *Wikipedia*, 2019. [En línea]

Disponible en: https://es.wikipedia.org/wiki/Android_SDK

[Accedido: 11-05-2019]

[33] “Retrofit”, *Retrofit*, n.d. [En línea]

Disponible en: <http://square.github.io/retrofit/>

[Accedido: 11-05-2019]

[34] “Gson User Guide”, *Gson*, n.d. [En línea]

Disponible en: <https://sites.google.com/site/gson/gson-user-guide>

[Accedido: 12-05-2019]

[35] D. Karnok, “RxJava”, *Github RxJava*, 2017. [En línea]
Disponible en: <https://github.com/ReactiveX/RxJava/wiki>
[Accedido: 12-05-2019]

[36] J. Zea, “Spring Framework ¿Qué es y para qué sirve? - Java”, *Curiotek*, 2017. [En línea]
Disponible en: <https://curiotek.com/2017/06/16/java-que-es-spring/>
[Accedido: 11-05-2019]

[37] M. Debnath, “What is Spring Boot?”, *Developer*, 2018. [En línea]
Disponible en: <https://www.developer.com/java/ejb/what-is-spring-boot.html>
[Accedido: 12-05-2019]

[38] M. Debnath, “What is JPA?”, *Developer*, 2018. [En línea]
Disponible en: <https://www.developer.com/java/data/what-is-jpa-technology.html>
[Accedido: 11-05-2019]

[39] “Hibernate”, *Hibernate*, n.d. [En línea]
Disponible en: <http://hibernate.org/>
[Accedido: 12-05-2019]

[40] “What is PostgreSQL”, *PostgreSQL*, n.d. [En línea]
Disponible en: <https://www.postgresql.org/about/>
[Accedido: 11-05-2019]

[41] M. Rouse, “JUnit”, *Search Software Quality*, 2010. [En línea]
Disponible en: <https://searchsoftwarequality.techtarget.com/definition/JUnit>
[Accedido: 12-05-2019]

[42] “Mockito”, *Mockito*, n.d. [En línea]
Disponible en: <https://site.mockito.org/>
[Accedido: 12-05-2019]

[43] “H2 Database Engine”, *H2*, n.d. [En línea]
Disponible en: <http://www.h2database.com/html/main.html>
[Accedido: 12-05-2019]

[44] “Kotlin (Lenguaje de Programación)”, *Wikipedia*, 2019. [En línea]
Disponible en: [https://es.wikipedia.org/wiki/Kotlin_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Kotlin_(lenguaje_de_programaci%C3%B3n))
[Accedido: 13-05-2019]

[45] “Desarrolla Apps para Android con Kotlin”, *Developers*, n.d. [En línea]
Disponible en: <https://developer.android.com/kotlin?hl=es>
[Accedido: 13-05-2019]

[46] “Java (Lenguaje de Programación)”, *Wikipedia*, n.d. [En línea]
Disponible en: [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
[Accedido: 13-05-2019]

[47] “Introduction to JSON Web Tokens”, *JWT*, n.d. [En línea]
Disponible en: <https://jwt.io/introduction/>

[Accedido: 13-05-2019]

[48] "Sistema Android Keystore", *Developers*, n.d. [En línea]

Disponible en: <https://developer.android.com/training/articles/keystore?hl=es-419>

[Accedido: 13-05-2019]

[49] "MonkeyTesting", *Wikipedia*, n.d. [En línea]

Disponible en: https://en.wikipedia.org/wiki/Monkey_testing

[Accedido: 13-05-2019]

[50] "General Data Protection Regulation", *Intersoft Consulting*, 2018. [En línea]

Disponible en: <https://gdpr-info.eu/>

[Accedido: 23-05-2019]

[51] "Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.", *Boletín Oficial del Estado*, 2018. [En línea]

Disponible en: <https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673>

[Accedido: 23-05-2019]

[52] "Impuesto de sucesiones y donaciones: todo lo que hay que saber", *Ayuda en Acción*, 2018. [En línea]

Disponible en: <https://ayudaenaccion.org/ong/blog/solidaridad/impuesto-sucesiones-donaciones/>

[Accedido: 23-05-2019]

[53] L. Shklar y R. Rosen, *Web application architecture: principles, protocols, and practices*. 2^a edición. John Wiley, 2009

[Accedido: 10-03-2019]

[54] K. Pohl, *Requirements engineering: fundamentals, principles, and techniques*. 1^a edición. Berlin: Springer, 2010.

[Accedido: 20-03-2019]

[55] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design patterns: elements of reusable object-oriented software*. 1^a edición. Addison Wesley, 1995.

[Accedido: 25-03-2019]

[56] C. Walls, *Spring Boot in Action*. 1^a edición. Manning Publications, 2015.

[Accedido: 20-03-2019]

[57] N. Smyth, *Android Studio 3.0 Development Essentials – Android 8 Edition*. 1^a edición.

CreateSpace Independent Publishing Platform, 2017.

[Accedido: 25-03-2019]