

Desarrollo de una aplicación multiplataforma para el intercambio de libros.

Arnau Sintès Rello

Máster en desarrollo de aplicaciones para dispositivos móviles
Trabajo final de máster

Jordi Almirall López y David Escuer Latorre
Carles Garrigues Olivella.

26/12/2021



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivad a [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)

A) Creative Commons:



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada a [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](#)

B) GNU Free Documentation License (GNU FDL)

Copyright © AÑO TU-NOMBRE.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3

or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Desarrollo de una aplicación multiplataforma para el intercambio de libros.</i>
Nombre del autor:	<i>Arnau Sintés Rello</i>
Nombre del consultor/a:	<i>Jordi Almirall López, David Escuer Latorre</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	12/2021
Titulación:	<i>Plan de estudios del estudiante</i>
Área del Trabajo Final:	<i>Trabajo final de máster</i>
Idioma del trabajo:	Castellano
Palabras clave	<i>Libro, intercambio, multiplataforma</i>
Resumen del Trabajo	
<p>En este trabajo se documentan todas las fases del desarrollo de una aplicación para dispositivos móviles centrada en el intercambio de libros. En el mercado existen bastantes aplicaciones centradas en la venta de productos de segunda mano, pero no centradas al cien por cien en el intercambio de productos, concretamente libros, sin un fin económico.</p> <p>La aplicación ofrece a los usuarios la posibilidad de publicar libros usados con la finalidad de ser intercambiados, o bien, buscar libros en los que este interesado el usuario. Además de permitir la comunicación entre usuarios desde la aplicación.</p> <p>El proyecto se desarrollará siguiendo una metodología en cascada, de manera que las distintas fases del proyecto se ejecutarán de forma secuencial. Para este proyecto se han definido las fases de captura de requisitos, diseño de la solución, implementación y evaluación.</p> <p>La solución está diseñada bajo un stack tecnológico bastante novedoso donde se usa Flutter para implementar la aplicación móvil, Firebase para la creación de un backend basado en servicios y tecnología REST para la comunicación entre los distintos módulos.</p>	
Abstract	
<p>This paper documents all the development phases of an application for mobile devices focused on the book exchange. There are many applications on the market that focus on the sale of second-hand products, but they are not entirely focused on the exchange of products, specifically books, without an economic purpose.</p>	

The application offers users the possibility of publishing used books in order to be exchanged, or to search for books in which the user is interested. It also allows communication between users through the application.

The project will be developed following a waterfall methodology, so that the different phases of the project will be executed sequentially. For this project, the phases of requirements capture, solution design, implementation and evaluation have been defined.

The solution is designed under a fairly new technological stack where Flutter is used to implement the mobile application, Firebase for the creation of a backend based on services and REST technology for communication between the different modules.

Índice

1. Introducción	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo	4
1.3 Enfoque y método seguido	7
1.4 Planificación del Trabajo	7
1.5 Breve resumen de productos obtenidos	10
1.6 Breve descripción de los otros capítulos de la memoria	10
2. Usuarios y contexto de uso	12
2.2 Análisis de stakeholders y usuarios	14
2.3 Perfiles de stakeholders	14
2.4 Perfiles de usuarios	15
3. Diseño conceptual	16
3.1 Escenarios de uso	16
3.2 Flujos de interacción	18
4. Prototipado	24
4.1 Esbozos	24
4.2 Árbol de navegación	27
4.3 Prototipo de Alta Fidelidad	28
4.4 Diseños Alternativos	32
5. Definición de casos de uso	34
5.1 Diagramas de casos de uso	34
5.2 Listado de casos de uso	35
6. Diseño de la arquitectura	47
6.1 Arquitectura del sistema	47
6.2 Arquitectura de la Base de Datos	47
6.3 Arquitectura de la aplicación	48
7. Revisión de la planificación	50
7.1 Cambio de tecnología	50
7.2 Limitación de funcionalidades	50
7.3 Sección Administrador	51
7.4 Limitación de la evaluación del producto	51
8. Implementación	52
8.1 Estructura del proyecto	52
8.2 Librerías	53
8.3 Soluciones implementadas	54
9. Resultados de la implementación	60

10. Evaluación	65
10.1 Definición de Evaluación	65
10.2 Resultados de Evaluación	66
11. Conclusiones	68
11.1 Mejoras de la aplicación	69
13. Glosario	70
14. Bibliografía	71
15. Anexos	72

Lista de figuras

Figura 1: Tabla comparativa de aplicaciones	2
Figura 2: Tabla de requisitos funcionales	5
Figura 3: Tabla de requisitos no funcionales	6
Figura 4: Tabla de restricciones	6
Figura 5: Diagrama de Gantt del proyecto	9
Figura 6: Estimación de horas del proyecto	10
Figura 7: Ficha de usuario 1	12
Figura 8: Ficha de usuario 2	13
Figura 9: Ficha de usuario 3	13
Figura 10: Tabla de stakeholders	14
Figura 11: Tabla de usuarios	14
Figura 12: Escenario de uso 1	16
Figura 13: Escenario de uso 2	17
Figura 14: Escenario de uso 3	18
Figura 15: Flujo de autenticación	18
Figura 16: Flujo de filtrado	19
Figura 17: Flujo de contactar con usuario	20
Figura 18: Flujo de creación de una nueva publicación	21
Figura 19: Flujo de marcar publicación como reservada	22
Figura 20: Flujo de eliminar una publicación	23
Figura 21: Esbozos de la aplicación 1	24
Figura 22: Esbozos de la aplicación 2	25
Figura 23: Esbozos de la aplicación 3	26
Figura 24: Árbol de navegación	27
Figura 25: Diagrama casos de uso Usuario	34
Figura 26: Diagrama casos de uso Administrador	35
Figura 27: Diagrama del sistema	47
Figura 28: Diagrama de la base de datos	47
Figura 29: Diagrama de la aplicación	48
Figura 30: Tabla de requisitos no desarrollados	50
Figura 31: Tabla de requisitos gestionados por Firebase	51
Figura 32: Estructura del proyecto	52
Figura 33: Dependencias del proyecto	53
Figura 34: Función de filtrado de libros	54
Figura 35: Función de cálculo de distancia	55
Figura 36: Providers de la aplicación	56
Figura 37: Stream de eventos de la autenticación	57
Figura 38: Gestión de autenticación	57
Figura 39: Obtención de la localización	58

Figura 40: Consulta a la api de Google	59
Figura 41: Tabla de test funcionales	65
Figura 42: Resultado de los test funcionales	66

1. Introducción

1.1 Contexto y justificación del Trabajo

El trabajo busca desarrollar una aplicación móvil enfocada en el intercambio de libros de lectura. Actualmente nos encontramos sumergidos en un mundo de consumismo el cual nos está aportando muchos problemas tanto a la sociedad como al planeta tierra, esta aplicación pretende que los usuarios intercambien objetos, concretamente libros para darles una segunda vida.

En el mercado existen muchas aplicaciones móviles destinadas a la venta de productos usados como pueden ser Wallapop o Milanuncios.

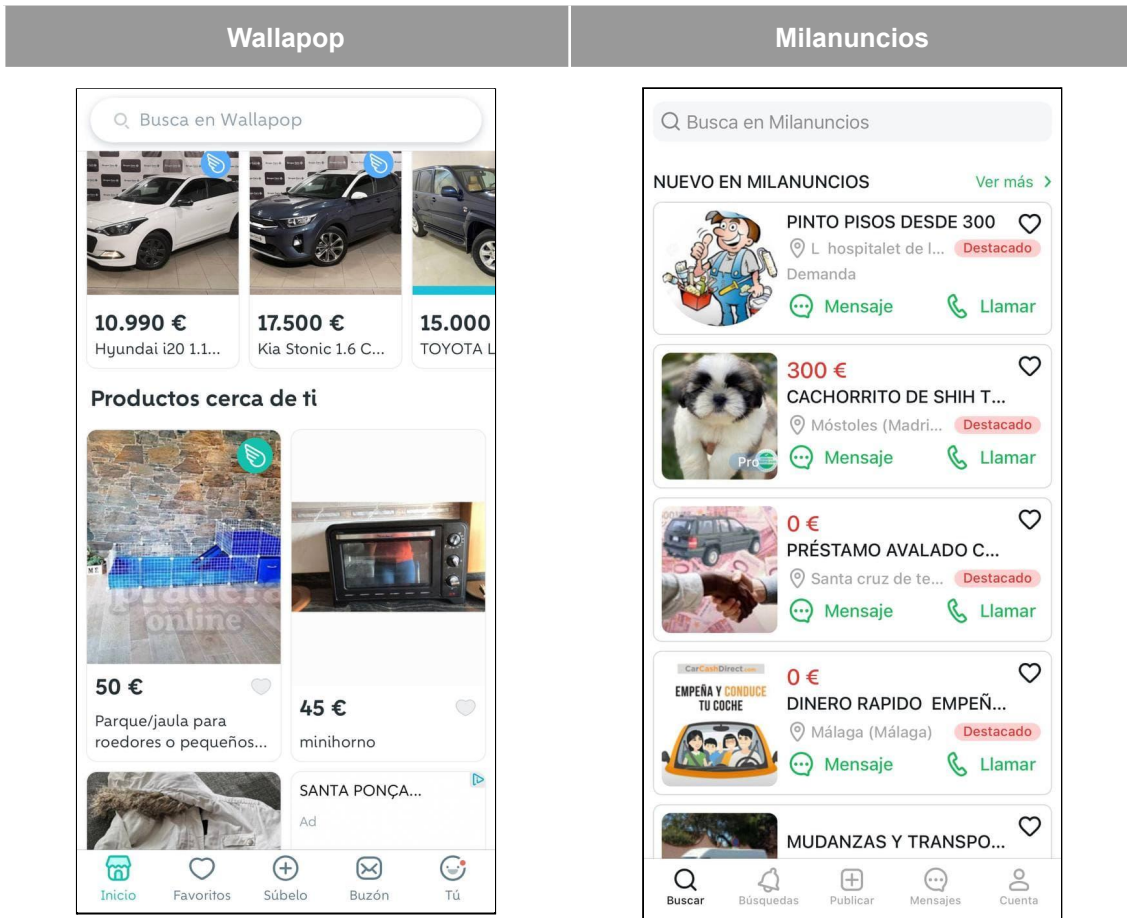
Todas las aplicaciones mencionadas tienen sus puntos fuertes y también débiles, pero la mayoría tienen un rol claramente definido, son una gran base de datos de productos en venta donde los usuarios pueden realizar búsquedas más o menos complejas y tienen el fin de poner en contacto a las dos partes interesadas en una compra o venta.

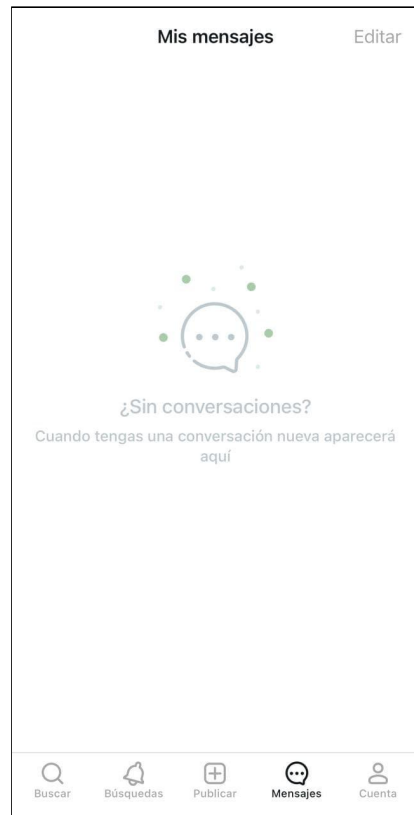
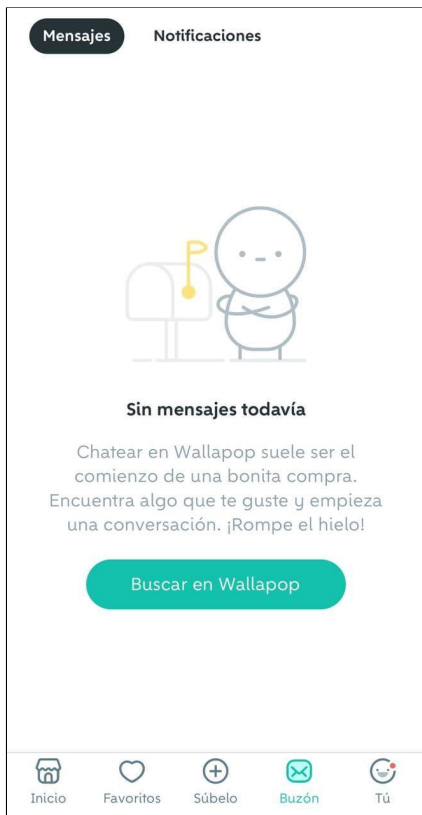
Con el objetivo de hacer más visual e intuitiva la comparativa entre aplicaciones se han seleccionado las características claves que debe tener este tipo de aplicaciones y en la tabla 1, se indican si aplicaciones seleccionadas cumplen las características:

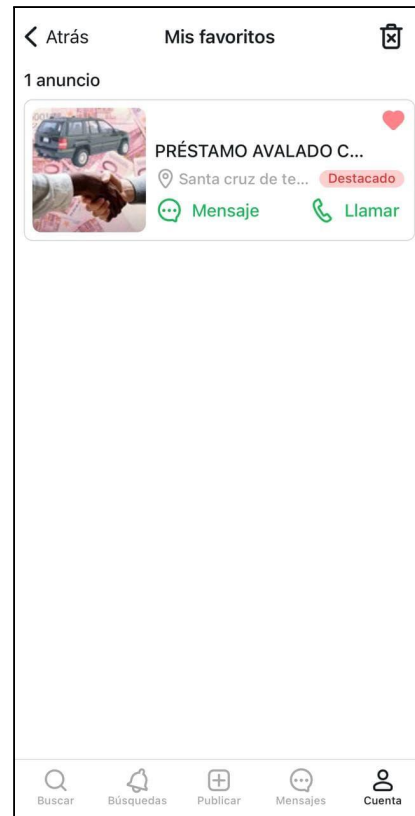
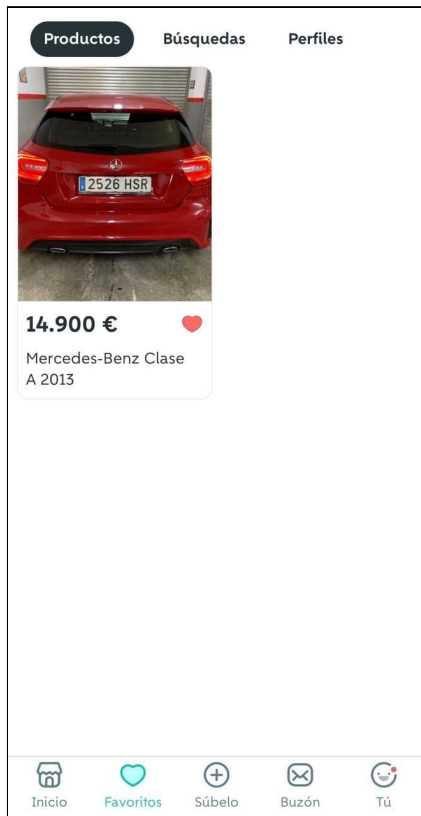
ID	Wallapop	Milanuncios	BookSharing
Propósito	Venta de productos usados	Venta de productos usados	Intercambio de libros
Tipo de productos	Todo tipo de productos	Todo tipo de productos	Libros.
Vista de resultados	Cuadrícula.	Lista.	Cuadrícula o Mapa.
Búsqueda avanzada	Si	Si	Si
Favoritos	Si	Si	Si
Descartar resultados	No	No	Si

Reservar	Si	Si	Si
Pago en App	Si	Si	No
Comunicación	Teléfono, chat interno y otros métodos.	Teléfono, chat interno y otros métodos.	Teléfono y chat interno.
Notificaciones	Si	Si	Si

Figura 1: Tabla comparativa de aplicaciones







Como se muestra tanto en la tabla comparativa como en las imágenes adjuntas, la aplicación a desarrollar, BookSharing, comparte muchas características con las aplicaciones anteriormente mencionadas, búsqueda avanzada, almacenamiento de favoritos, hacer reservas, vista de los resultados en lista o mapa. Donde radica la principal diferencia es en el propósito de las aplicaciones, mientras Wallapop y Milanuncios tienen como principal propósito la venta de productos usados, la aplicación a desarrollar, BookSharing, fomenta el intercambio de productos con usuarios cercanos.

De esta manera pretende promover un cambio de mentalidad frente al consumismo en el que se encuentra la sociedad actual y de esta manera ayudar a mitigar las secuelas que este genera como son la generación masiva de residuos o emisión de gases de efecto invernadero generados por los procesos de producción y transporte, entre otros.

Las aplicaciones comparadas pueden encontrarse en los enlaces [Wallapop - Compra y vende en App Store \(apple.com\)](https://www.apple.com/es/ios/apps/1444684142) y [Milanuncios - Comprar y vender en App Store \(apple.com\)](https://www.apple.com/es/ios/apps/1444684142).

1.2 Objetivos del Trabajo

A continuación se describen tanto los requisitos funcionales, no funcionales como restricciones del sistema a desarrollar.

Requisitos funcionales.

ID	Descripción
SR-FR-0001	La aplicación debe permitir publicar, actualizar y eliminar anuncios de productos a los usuarios.
SR-FR-0002	La aplicación debe permitir a los usuarios realizar búsquedas.
SR-FR-0003	La aplicación debe permitir realizar búsquedas avanzadas mediante un formulario.
SR-FR-0004	La aplicación debe permitir filtrar los resultados de la búsqueda a los usuarios.
SR-FR-0005	Los usuarios deben poder descartar resultados.
SR-FR-0006	Los usuarios deben poder guardar productos en favoritos.
SR-FR-0007	Los anuncios deben poder visualizar en una lista o en un mapa.
SR-FR-0008	Un usuario debe poder ver los detalles de un producto.
SR-FR-0009	Un usuario puede acceder al perfil de otro usuario.
SR-FR-0010	Un usuario puede ver el catálogo de productos disponible por otro usuario.
SR-FR-0011	Un usuario debe poder ver un historial de intercambios realizados.
SR-FR-0012	La aplicación debe permitir la comunicación entre usuarios con un chat propio de la aplicación.
SR-FR-0013	La aplicación debe permitir a un usuario reservar un producto.
SR-FR-0014	Un usuario puede bloquear a otros usuarios para que estos no vean un anuncio en sus búsquedas.
SR-FR-0015	Los usuarios deben poder reportar productos publicados.
SR-FR-0016	Un usuario administrador debe poder eliminar o pedir revisión por parte de los usuarios de un producto.
SR-FR-0017	Un usuario administrador debe poder revisar los productos reportados.
SR-FR-0018	Un usuario debe poder registrarse en la aplicación.
SR-FR-0019	Un usuario debe poder iniciar sesión.

Figura 2: Tabla de requisitos funcionales

Requisitos no funcionales

ID	Descripción
SR-NF-0001	Un usuario solo tiene un perfil. Ni dos ni ninguno.

SR-NF-0002	Un usuario solo se puede registrar una única vez con el mismo correo electrónico.
SR-NF-0003	Un usuario publicar uno o muchos productos en su perfil
SR-NF-0004	Un usuario puede establecer una o muchas conversaciones simultáneas en la aplicación.
SR-NF-0005	Los nuevos mensajes de una conversación se notificarán mediante notificaciones push.
SR-NF-0006	La aplicación deberá mantener la sesión activa.

Figura 3: Tabla de requisitos no funcionales

Restricciones

ID	Descripción
SR-RE-0001	La aplicación será una app móvil multidispositivo
SR-RE-0002	La aplicación se desarrollará en Flutter.
SR-RE-0003	La parte del servidor se realizará mediante Firebase
SR-RE-0004	La base de datos será de tipo no relacional
SR-RE-0005	La autenticación se realizará mediante Firebase.

Figura 4: Tabla de restricciones

Funcionalidades principales

A continuación se describen las funcionalidades básicas que debe cubrir la aplicación:

- Publicación de libros a intercambiar.
- Búsqueda de libros avanzada, con filtros por localización, categorías y demás características del libro.
- Visualización de resultados en una lista o mapa.
- Guardar resultados en favoritos.
- Descartar resultados.
- Permitir la comunicación entre usuarios, mediante chat interno.
- Reserva de un libro.

El listado de funcionalidades se ha obtenido a partir de la definición de requisitos funcionales que se puede consultar en la sección Requisitos Funcionales del documento.

1.3 Enfoque y método seguido

Una vez analizadas las aplicaciones existentes en el mercado y definir los objetivos del proyecto se ha decidido hacer una app nueva basándose en apps existentes y contrastadas, de esta manera podemos ofrecer una aplicación que pueda rivalizar con las existentes en el mercado y pueda ser atractiva y fácil de usar para los usuarios.

Se desarrollará con tecnologías modernas como, Flutter, que nos permite crear aplicaciones multidispositivo sin tener que pasar por crear proyectos específicos para cada tipo de dispositivo, Android o iOS. Por lo que repercute a la parte servidor del sistema, se ha decidido hacer uso de un BaaS, concretamente de Firebase, ya que se trata de una aplicación que principalmente funciona como una gran base de datos de anuncios y por la facilidad y flexibilidad de integración que ofrecen servicios como Firebase a la hora de crear servidores para nuestros proyectos, pudiendo tener operativo un servidor con una base de datos en pocas horas. Además de ofrecer servicios de autenticación útiles para controlar los accesos a la aplicación y los tipos de roles que pueden tener los usuarios.

Se ha decidido seguir la metodología de desarrollo en cascada, ya que se trata de un proyecto con unas fases muy marcadas. También al tratarse de un proyecto donde no existe un cliente o interesados que puedan solicitar cambios en los requisitos durante la ejecución del proyecto, se han descartado otras metodologías más ágiles como SCRUM.

1.4 Planificación del Trabajo

La planificación del proyecto está basada en 5 bloques marcados por cada una de las PEC de la asignatura.

El primer bloque nos centraremos en la conceptualización y planificación del proyecto, que constará de las siguientes tareas:

- Definir el contexto de qué se va a desarrollar y analizar la oferta actual.
- Definir objetivos y enfoque del proyecto.
- Definir la planificación del proyecto y los entregables a realizar.

En el segundo bloque nos centraremos en el diseño conceptual de la aplicación el cual se realizará a través de las siguientes tareas:

- Análisis de posibles usuarios y perfiles.
- Definición de casos de uso a partir de usuarios y perfiles detectados.
- Realización de prototipos de baja y alta fidelidad.

- Documentación técnica de la arquitectura, tecnología aplicadas en el desarrollo del proyecto y diagramas de flujo.

En el tercer bloque nos centraremos en el desarrollo de aplicación y parte del servidor. Este bloque tendremos las siguientes tareas:

- Definición de modelo de datos que intervienen en la aplicación y diseño de la base de datos.
- Creación del backend mediante el servicio Firebase.
- Desarrollo de la aplicación móvil con Flutter.
- Realización de pruebas end-to-end de sistema.

En el cuarto bloque se finalizarán tanto el sistema como la memoria del proyecto repartido en las siguientes tareas:

- Realización del informe final del proyecto.
- Creación de manual de usuario de la aplicación.
- Creación del manual de compilación de la aplicación.
- Presentación del proyecto.

En el último bloque se responderán todas las cuestiones y preguntas recibidas por el tribunal evaluador:

- Defensa del trabajo realizado.

A continuación podemos ver en el siguiente diagrama de Gantt cómo se han planificado las tareas anteriormente descritas a lo largo de los meses que dura el proyecto. También se pueden observar las fechas de entrega de los distintos bloques marcadas con líneas verticales de color rojo.

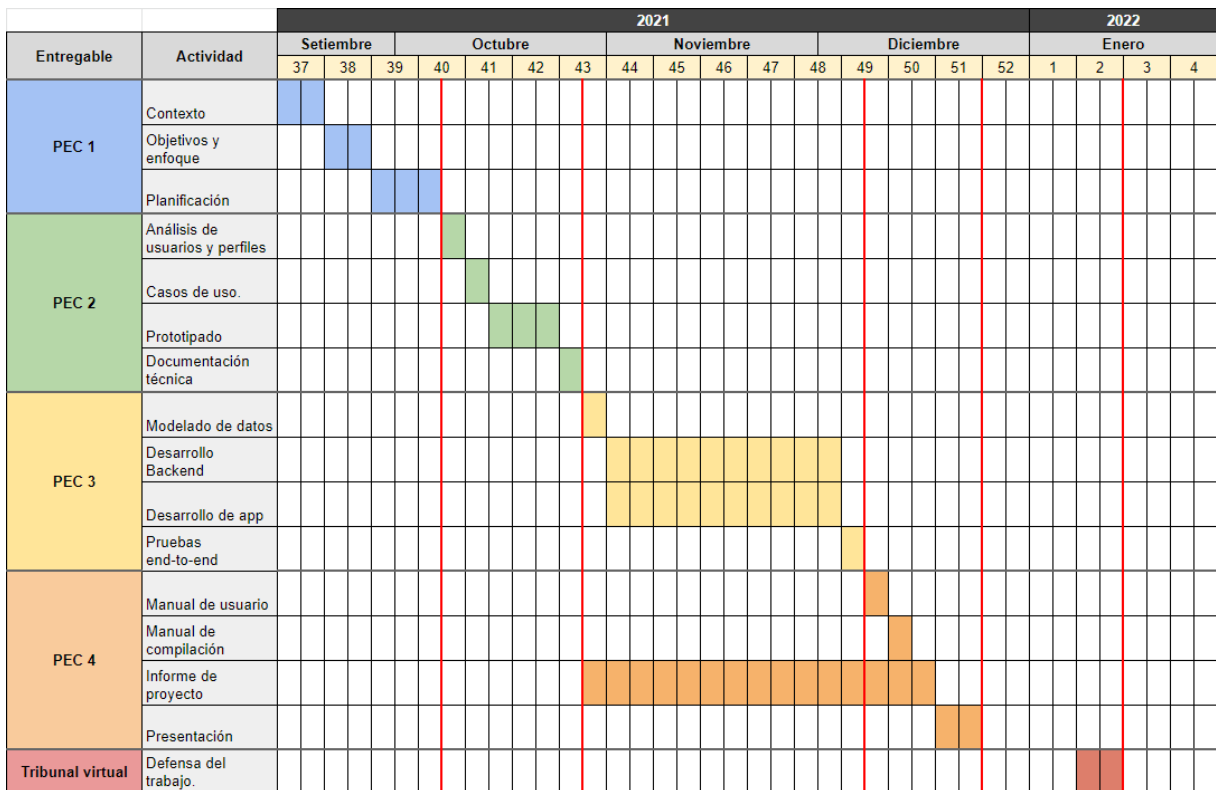


Figura 5: Diagrama de Gantt del proyecto

En el diagrama destacan tres actividades sobre las otras, las primeras, Desarrollo Backend y Desarrollo de app, se solapan en el tiempo, ya que se pretende ir avanzando en ambas tareas a medida que la aplicación va creciendo. La otra que destaca es Informe de proyecto, es debido a que se pretende ir redactando el informe a medida que se tomen decisiones técnicas y se completen partes importantes del proyecto.

Se ha estimado una duración de 197 horas para la realización del proyecto. Para ello se tiene en cuenta la duración de cada actividad y una dedicación media de 2 horas diarias, a excepción de épocas de festivas, donde la dedicación puede aumentar en caso de ser necesario.

Bloque	Actividad	Horas Estimadas
PEC 1	Propuesta y Contexto	10
	Objetivos y enfoque	10
	Planificación	15
PEC 2	Análisis de usuarios y perfiles	5
	Casos de uso.	5
	Prototipado	15
	Documentación técnica	5
PEC 3	Modelado de datos	5

	Desarrollo Backend	15
	Desarrollo de app	35
	Pruebas end-to-end	5
PEC 4	Manual de usuario	5
	Manual de compilación	5
	Informe de proyecto	38
	Presentación	20
Tribunal Virtual	Defensa del trabajo.	4
Total de horas estimadas		197

Figura 6: Estimación de horas del proyecto

1.5 Breve resumen de productos obtenidos

Al finalizar el proyecto se entregarán los siguientes productos:

- Código fuente de la aplicación desarrollada con Flutter.
- Archivo *apk* de la aplicación una vez compilada.
- Memoria del proyecto.
- Presentación del proyecto.
- Manual de usuario.
- Manual de instalación.

1.6 Breve descripción de los otros capítulos de la memoria

Los capítulos que conforman la memoria de este proyecto son los siguientes:

- **Introducción:** Se definirán el contexto, objetivos y planificación del proyecto desarrollado.
- **Usuarios y contexto de uso:** Se analizarán los posibles usuarios y perfiles que usarán la aplicación.
- **Diseño conceptual:** Se analizarán los posibles escenarios de uso de la aplicación y los flujos de interacción.
- **Prototipado:** Se realizarán los esbozos y prototipos de la aplicación una vez analizados los posibles escenarios de uso.
- **Definición de caso de uso:** Se identificarán y analizarán los casos de uso identificados.
- **Diseño de la arquitectura:** En este apartado se definirá la arquitectura del sistema y de los elementos que lo componen como la base de datos o la aplicación móvil.

- **Revisión de la planificación:** Se presentan las desviaciones, respecto la planificación inicial, surgidas durante la implementación, así como los motivos que han llevado a estas.
- **Implementación:** En este apartado se describen detalles de la implementación como estructura del proyecto, librerías usadas y detalles de código en detalle.
- **Resultados de la implementación:** Se presentarán las capturas de la aplicación una vez desarrollada.
- **Evaluación:** Se definirá el plan de evaluación y pruebas que debe pasar el sistema para validar que su comportamiento es el esperado.
- **Conclusiones:** Se expondrán las conclusiones obtenidas durante el desarrollo del proyecto.
- **Glosario:** Se citarán términos y acrónimos usados en el documento, aportando la descripción de los mismos.
- **Bibliografía:** Se citarán las fuentes consultadas durante el desarrollo del proyecto.
- **Anexo:** Se adjuntará la documentación adicional generada durante el desarrollo del proyecto como manuales de usuario o de instalación, además de posibles apartados que sean demasiados extensos.

2. Usuarios y contexto de uso

Con la finalidad de crear un producto que se ajuste a las necesidades de los posibles usuarios, es necesario identificar e involucrar a todos los posibles usuarios y participantes del proyecto como parte del proceso conceptualización del producto a desarrollar.

En esta sección se analizan los interesados en el proyecto y posibles usuarios, así como los perfiles de estos a partir de fichas de usuario. Este proceso nos permite justificar decisiones tomadas en procesos de diseño que se desarrollaran en otras secciones.

2.1 Fichas de usuario

Nombre	Laura
Edad	24 años
Nivel de estudios	Grado universitario
Profesión	Bióloga
Descripción	<p>Laura tiene pareja, pero vive con sus padres. Vive en Ripollet y ha empezado a trabajar en una empresa química como ayudante de laboratorio. De 8.00 a 17.00 realizar sus tareas en el laboratorio donde se encarga de realizar controles de calidad, para ello dispone de un ordenador donde debe documentar todos los procesos y resultados.</p> <p>En casa dispone de conexión a internet por fibra óptica. También dispone de un ordenador portátil que usó durante los estudios, pero que ahora usa para realizar distintos trámites en línea. Lo que más le gusta es leer y ver series en su televisor o móvil, el cual también usa como medio de comunicación o para hacer uso de las redes sociales.</p> <p>Cuando tiene dudas sobre tecnología consulta a su grupo de amigos y se deja aconsejar por ellos.</p>

Figura 7: Ficha de usuario 1

Nombre	Daniel
Edad	22 años
Nivel de estudios	Bachillerato
Profesión	Estudiante de grado universitario

Descripción	<p>Daniel no tiene pareja y vive con sus padres y su hermano menor de 17 años. Vive en Cerdanyola del Vallés y está finalizando un grado de ingeniería informática en la Universidad Autónoma de Barcelona.</p> <p>En casa dispone de conexión a internet con fibra óptica, tiene un ordenador propio para los estudios que además usa para jugar con sus amigos a videojuegos en línea. También dispone de un teléfono móvil que usa sobre todo para comunicarse, realizar trámites y compras en línea además de entretenerse en redes sociales. Además dispone de una videoconsola la cual raramente usa.</p> <p>Le gusta estar informado de las últimas tendencias y noticias sobre tecnología y videojuegos, además sus familiares y amigos le piden consejos sobre este tema.</p>
--------------------	--

Figura 8: Ficha de usuario 2

Nombre	Antonio
Edad	54 años
Nivel de estudios	Graduado escolar
Profesión	Administrativo
Descripción	<p>Antonio está separado y tiene un hijo de 26 años el cual está independizado y vive en otra ciudad. Vive en Barcelona, en el barrio del Poblenou, y trabaja de administrativo para una empresa del sector farmacéutico. De 9.00 a 18.00 se encuentra delante de su puesto de trabajo con un ordenador y un teléfono.</p> <p>En casa dispone de conexión a internet por fibra óptica y dispone de una torre que solo hace uso para realizar trámites o alguna compra en línea. Dispone de un teléfono móvil que usa para comunicarse y estar conectado con viejos amigos mediante las redes sociales. Además dispone de televisor Smart TV que usa para ver series o películas.</p> <p>Está interesado en la nueva tecnologías y dispositivos que salen al mercado, pero siempre pide consejo a su hijo. En su móvil dispone de todo tipo de aplicaciones, desde juegos a redes sociales, no suele tener problemas para usarlos ni configurarlas.</p>

Figura 9: Ficha de usuario 3

2.2 Análisis de stakeholders y usuarios

Resumen de stakeholders

Nombre	Descripción	Aportación
Personas consultadas para las fichas de usuario.	Personas que han respondido a preguntas para la creación de las fichas de usuario.	Fuente con peso importante en los requisitos definidos de la aplicación.
Usuario de la aplicación.	Los usuarios finales de la aplicación a desarrollar.	Aportan información sobre los requisitos funcionales del sistema.
Equipo de desarrollo	Integrantes del equipo de desarrollo del proyecto, tanto desarrolladores como consultores.	Aportan requisitos más técnicos y la propia experiencia en otros proyectos realizados

Figura 10: Tabla de stakeholders

Resumen de usuarios

Nombre	Descripción	Perfil
Publicador de un libro.	Usuario que publica un libro para intercambiar	Usuario de la aplicación
Usuario interesado en un producto.	Usuario que usa la aplicación para buscar un libro a intercambiar por otros.	Usuario de la aplicación
Administrador o moderador de contenido.	Usuario que gestiona el contenido de la aplicación.	Usuario administrador de la aplicación

Figura 11: Tabla de usuarios

2.3 Perfiles de stakeholders

Participantes en las fichas de usuario

Representante	Participantes en las fichas de usuario.
Descripción	Tipo de stakeholder del cual se puede extraer una cantidad de requisitos importante y de forma rápida.
Tipo	Pueden abarcar todo tipo de perfiles, desde expertos en tecnología a personas interesadas en intercambiar un libro o personas ajenas al mundo de la tecnología y sin interés en intercambiar productos.

Responsabilidades	No tienen responsabilidades en el proyecto.
Criterios de éxito	La versión final de la aplicación ha de satisfacer sus necesidades. Sus intereses son aportar respuestas útiles para los desarrolladores de la aplicación.

Usuarios de la aplicación

Representante	Futuros usuarios de la aplicación.
Descripción	Personas interesadas en la utilización de la aplicación, de la cual saldrán las ideas sobre la interfaz y uso de la aplicación.
Tipo	No es un experto en tecnología, pero sí pueden tener ciertos conocimientos en tecnologías e interés en el mundo de la lectura.
Responsabilidades	Sus intereses son el buen funcionamiento de la aplicación y que sea fácil de usar.
Criterios de éxito	Tener una experiencia de uso con la aplicación agradable e intuitiva.

Desarrolladores de la aplicación

Representante	Desarrollador del proyecto.
Descripción	Stakeholder encargado del control y desarrollo del producto durante todo su desarrollo.
Tipo	Es un experto en la materia tecnológica, pero puede tener poco conocimiento sobre el mundo de la literatura.
Responsabilidades	Tiene responsabilidades directas en la calidad y buen funcionamiento de la aplicación. Su interés es satisfacer las necesidades y expectativas del cliente.
Criterios de éxito	Que el desarrollo de la aplicación se adecúe a la planificación realizada y a las expectativas de resultados definidos.

2.4 Perfiles de usuarios

Publicador de un libro

Representante	Publicador de un libro.
Descripción	Persona que publica un libro para intercambiar

Tipo	Persona que puede no tener conocimientos técnicos de la aplicación, pero sí interés en literatura.
Responsabilidades	Mantener actualizada la información del anuncio.
Criterios de éxito	Que el uso de la aplicación sea intuitiva y fácil, además que el proceso de intercambio sea sencillo.

Usuario interesado en un producto

Representante	Usuario interesado en un producto.
Descripción	Persona que usa la aplicación para buscar intercambiar un libro por otro.
Tipo	Persona que puede no tener conocimientos técnicos de la aplicación, pero sí interés en literatura.
Responsabilidades	Mantener actualizada la información del anuncio.
Criterios de éxito	Que el uso de la aplicación sea intuitiva y fácil, además que el proceso de intercambio sea sencillo.

Administrador

Representante	Usuario administrador o moderador de contenido.
Descripción	Persona que gestiona el contenido de la aplicación.
Tipo	Experto sobre el funcionamiento del sistema, además debe tener ciertos conocimientos del contenido de la aplicación.
Responsabilidades	Es responsable del contenido de la aplicación, de que esté correcto y se adecúe a la calidad que se quiere ofrecer. Sus intereses son preservar la calidad de la aplicación y variedad de contenido de esta.
Criterios de éxito	Que el contenido y el uso de la aplicación siempre sea correcto.

3. Diseño conceptual

3.1 Escenarios de uso

A continuación se describen los posibles escenarios de uso de la aplicación partiendo de los usuarios descritos en las fichas.

Id. Escenario	E-0001
Usuario	Laura
Descripción	<p>Laura se ha instalado la aplicación desde el mercado de aplicaciones y se ha dado de alta en el sistema.</p> <p>Una vez ha iniciado sesión ha rellenado los datos personales de su perfil. Se dispone a buscar libros que no haya leído aún.</p> <p>Como no dispone de vehículo personal y está interesada en los libros de misterio y terror decide hacer uso de los filtros para buscar libros de la categoría que le interesa y que se encuentren en su población o poblaciones del alrededor a las que se pueda desplazar caminando.</p> <p>Tras intercambiar mensajes con distintos usuarios, ha llegado a un acuerdo con dos usuarios para intercambiar libros.</p>

Figura 12: Escenario de uso 1

Id. Escenario	E-0002
Usuario	Daniel
Descripción	<p>Daniel se ha instalado la aplicación desde el mercado de aplicaciones y se ha dado de alta en el sistema.</p> <p>Una vez ha iniciado sesión ha rellenado los datos personales de su perfil. Con la ayuda de su hermano se dispone a publicar los libros del curso anterior de su hermano para intercambiarlos por otros libros de su interés, por ello lo indica en la descripción de las publicaciones que ha creado.</p> <p>Al rato de publicar los libros recibe un mensaje de un usuario interesado en los libros publicados por Daniel. Tras un rato hablando desde la aplicación han encontrado un intercambio satisfactorio para ambos y deciden quedar al día siguiente para efectuar dicho intercambio.</p> <p>Tras el intercambio, Daniel se encuentra muy satisfecho con la aplicación.</p>

Figura 13: Escenario de uso 2

Id. Escenario	E-0003
Usuario	Antonio
Descripción	<p>Antonio con la ayuda de su hijo han instalado la aplicación desde el mercado de aplicaciones y se ha registrado correctamente y ha iniciado sesión.</p>

Antonio quiere deshacerse de unos libros que tiene cogiendo polvo en la estantería, por ello, tras actualizar la información de su perfil, publica distintos anuncios en la aplicación haciendo énfasis tanto en la descripción como en las etiquetas que permite añadir la aplicación al crear una publicación que no está interesado en un intercambio, sino que los ofrece a otros usuarios de forma totalmente gratuita sin necesidad de dar un libro a cambio.

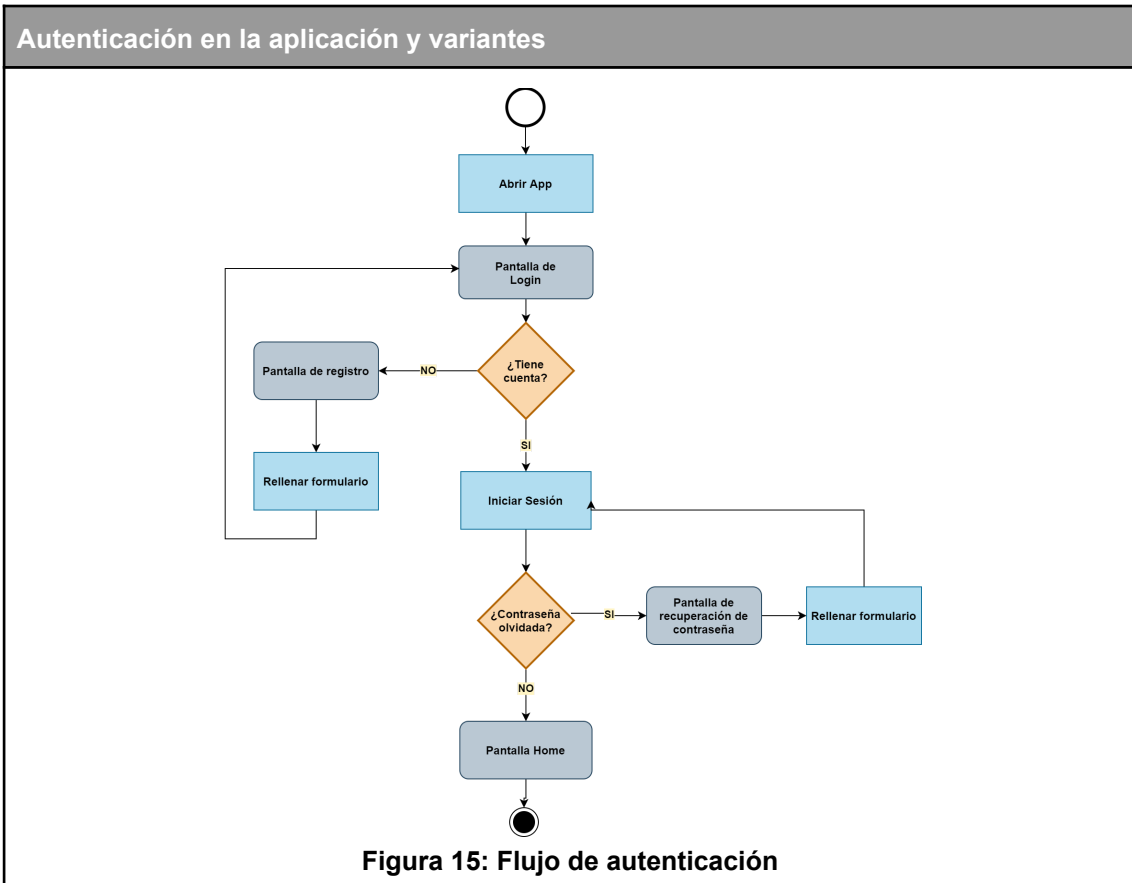
Antonio recibe a lo largo de la semana mucho mensaje de usuarios interesados en los libros que ha publicado, por ello decide marcar como reservados algunos de los libros publicados para evitar recibir más mensajes.

Tras realizar los encuentros con los otros usuarios, procede a eliminar los anuncios de los libros que ha cedido a otros usuarios.

Figura 14: Escenario de uso 3

3.2 Flujos de interacción

A partir de los escenarios descritos en el punto anterior, se han detectado flujos de interacción internos de la aplicación, los cuales se representan a continuación mediante diagramas.



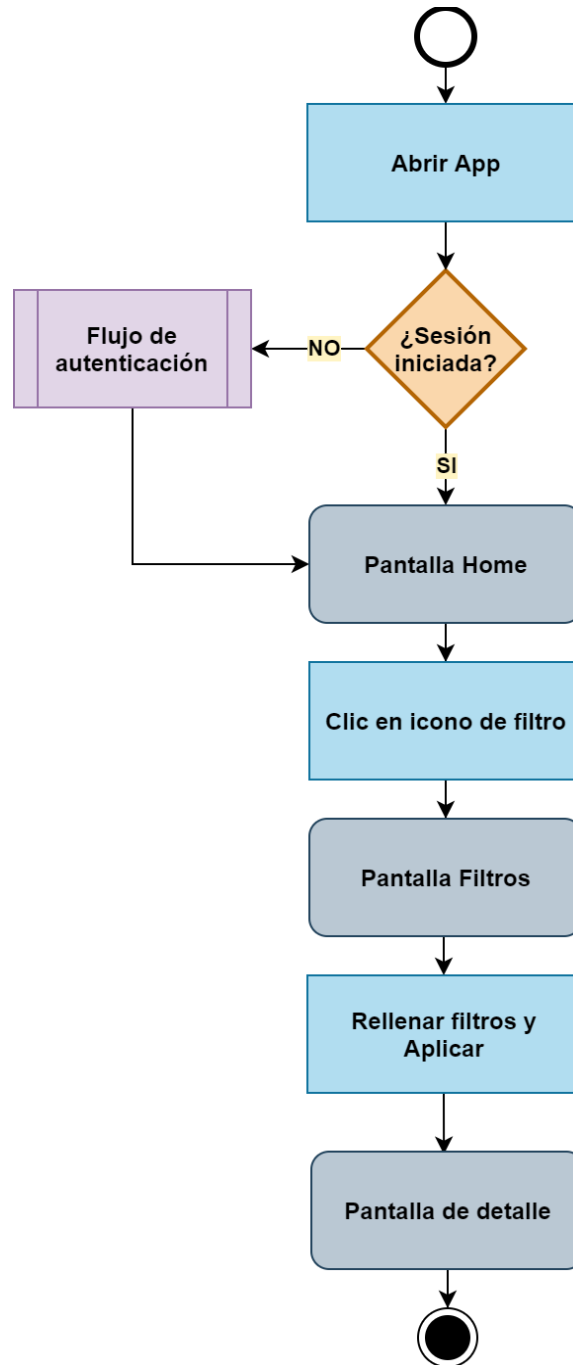
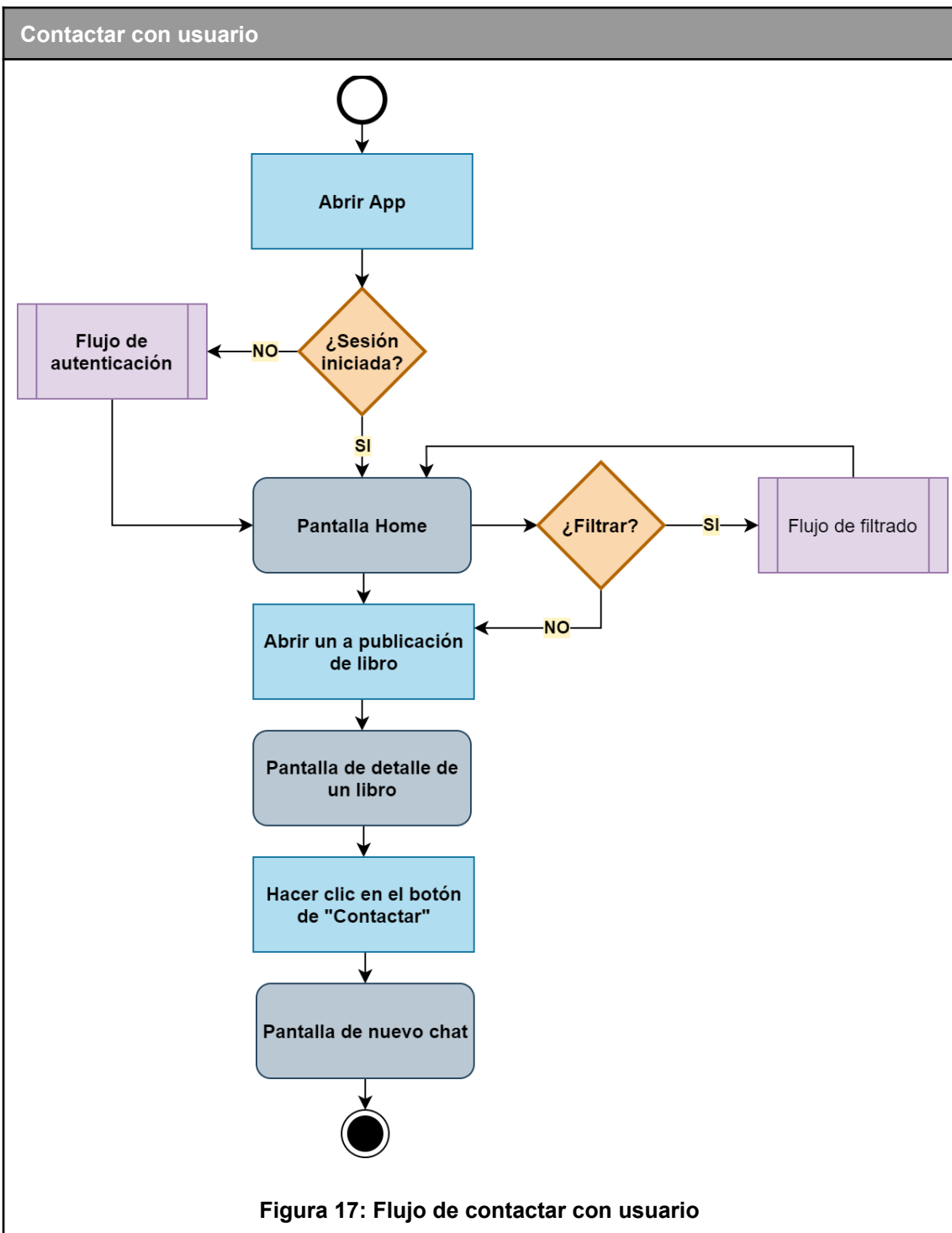


Figura 16: Flujo de filtrado



Crear una nueva publicación

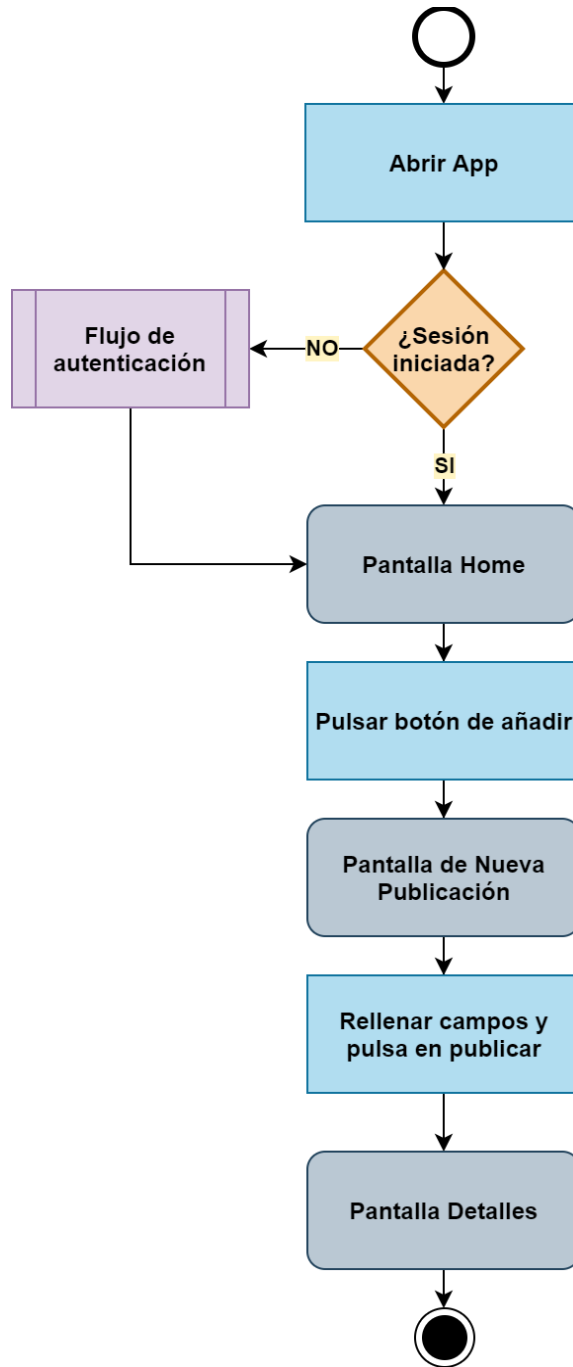


Figura 18: Flujo de creación de una nueva publicación

Marcar libro como reservado

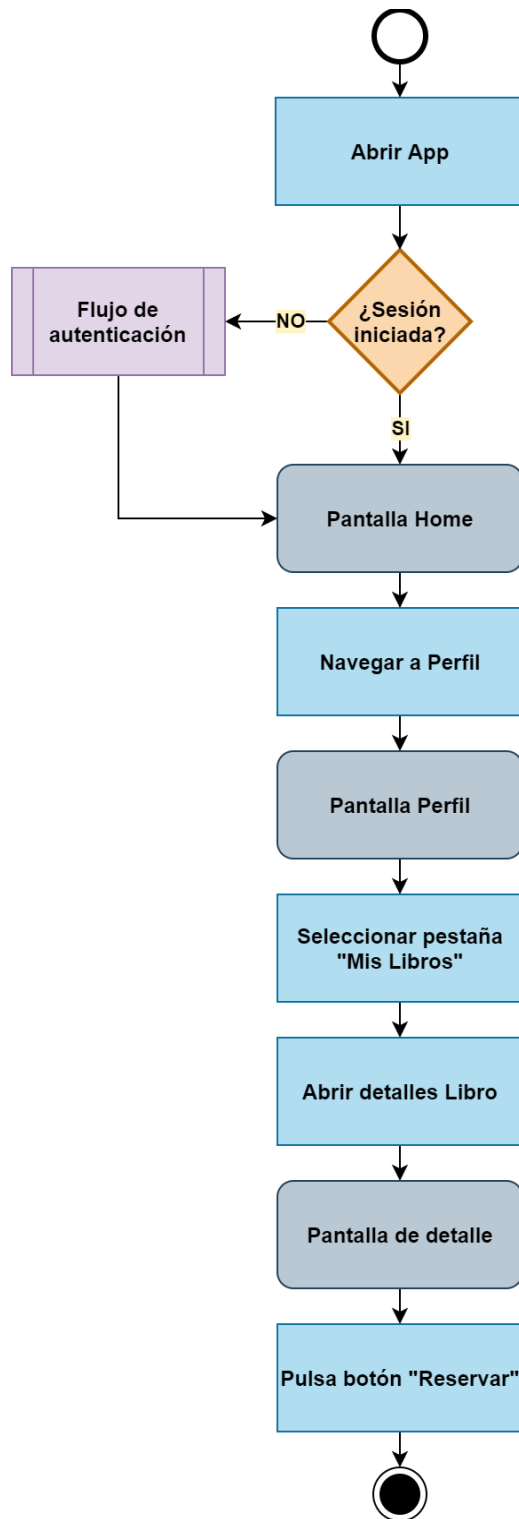


Figura 19: Flujo de marcar publicación como reservada

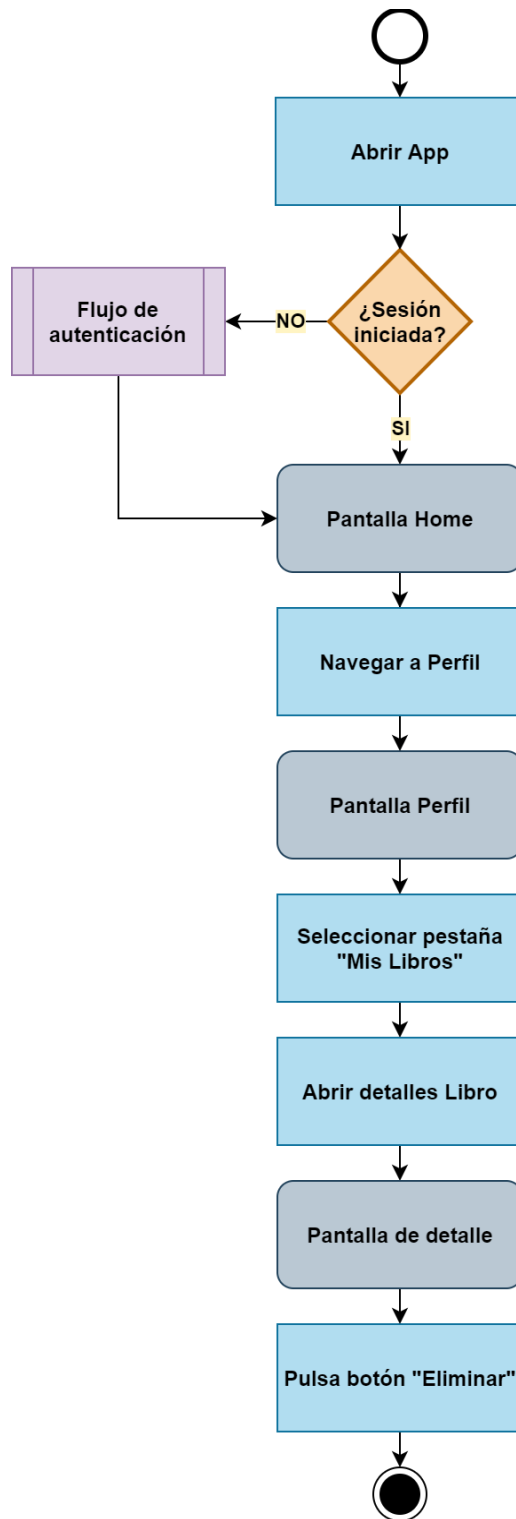


Figura 20: Flujo de eliminar una publicación

4. Prototipado

4.1 Esbozos

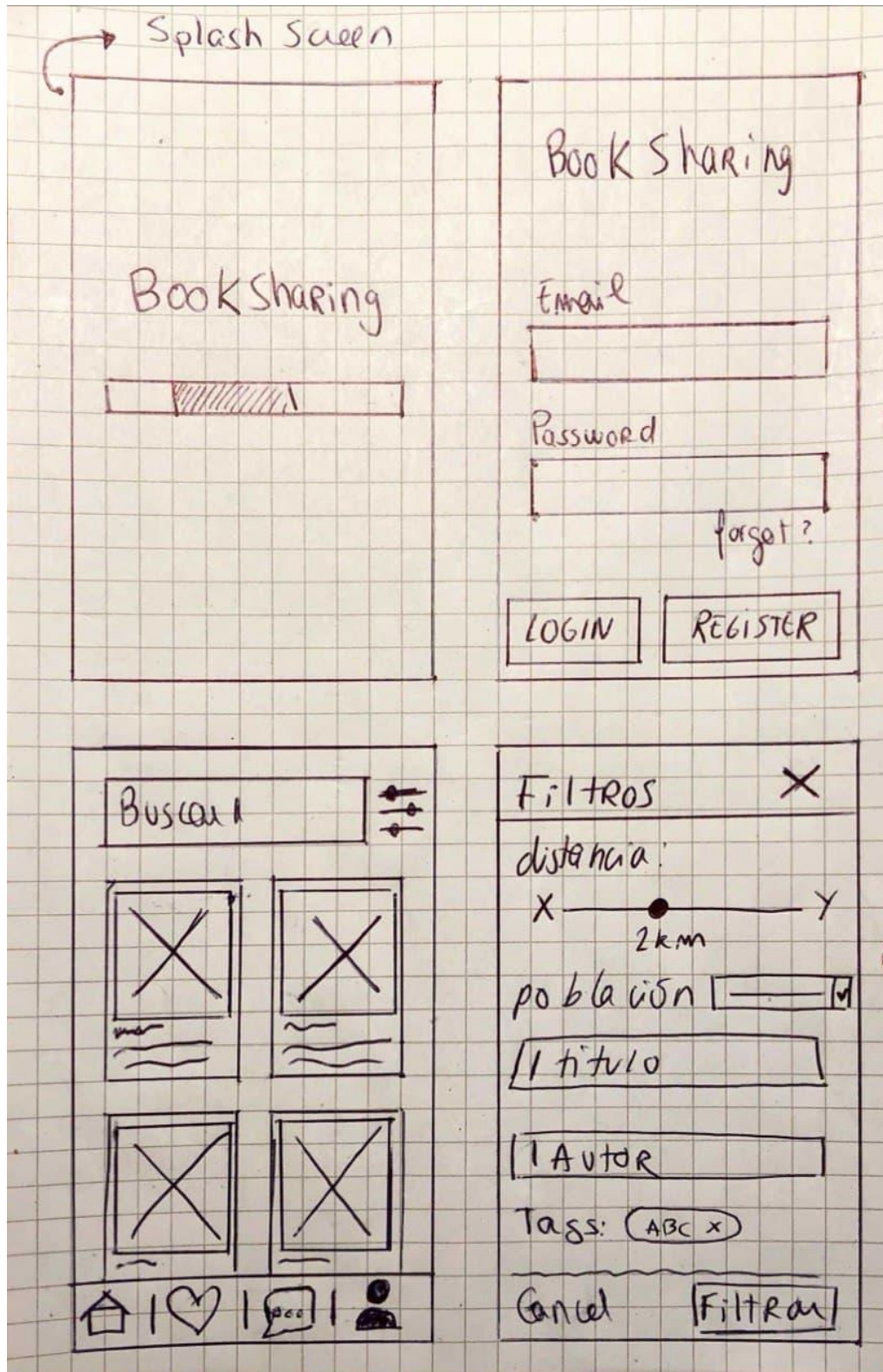


Figura 21: Esbozos de la aplicación 1

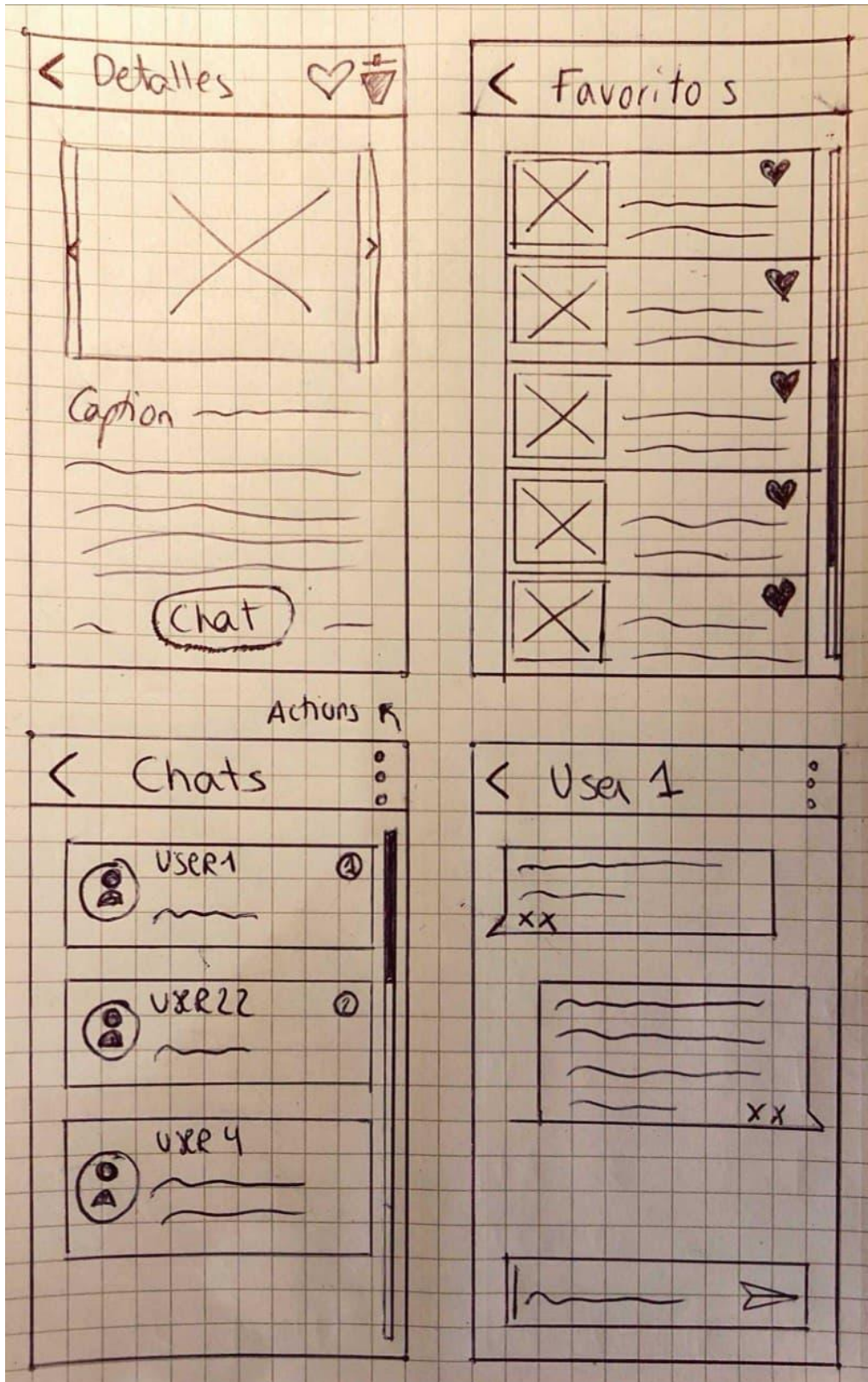


Figura 22: Esbozos de la aplicación 2

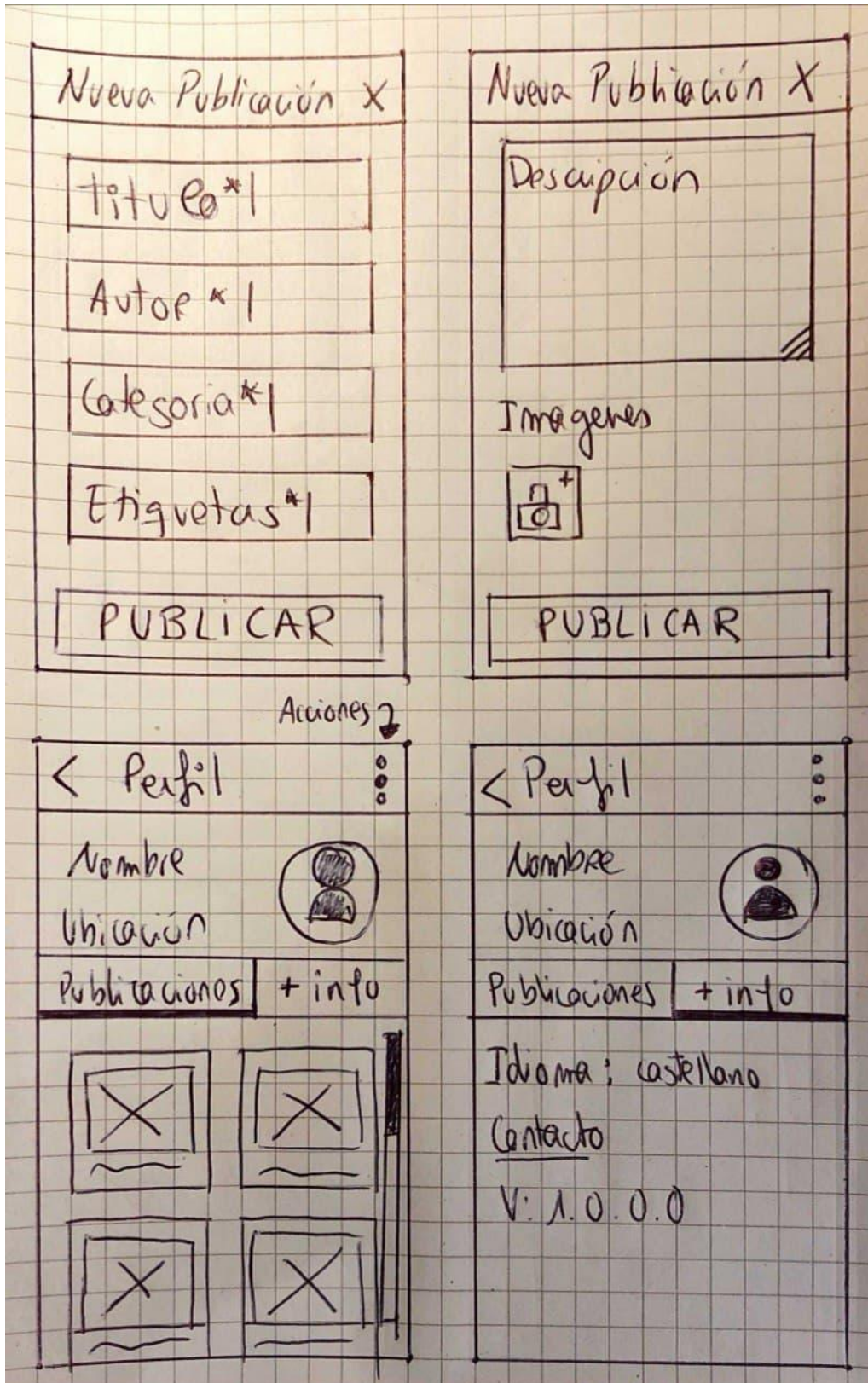


Figura 23: Esbozos de la aplicación 3

4.2 Árbol de navegación

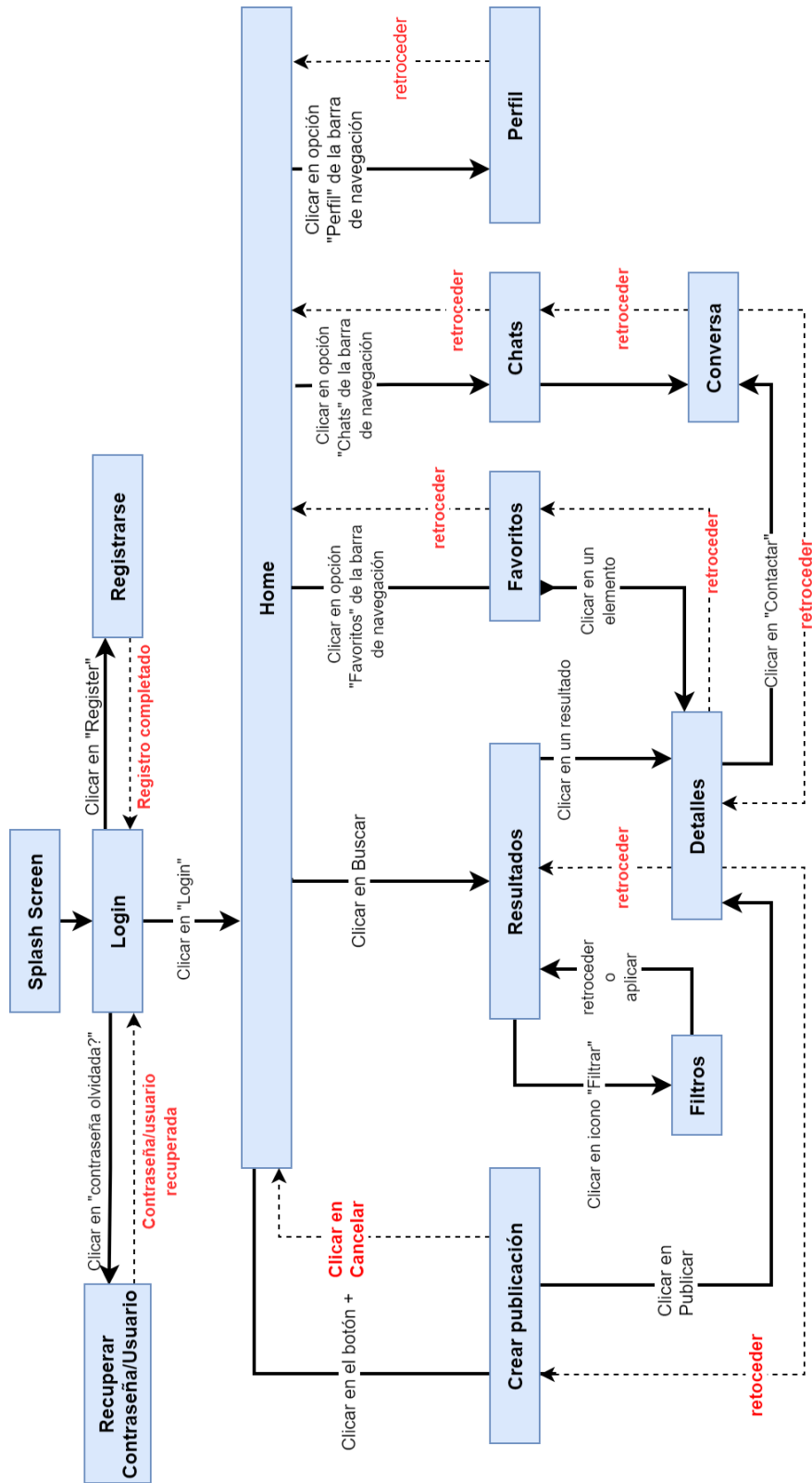


Figura 24: Árbol de navegación

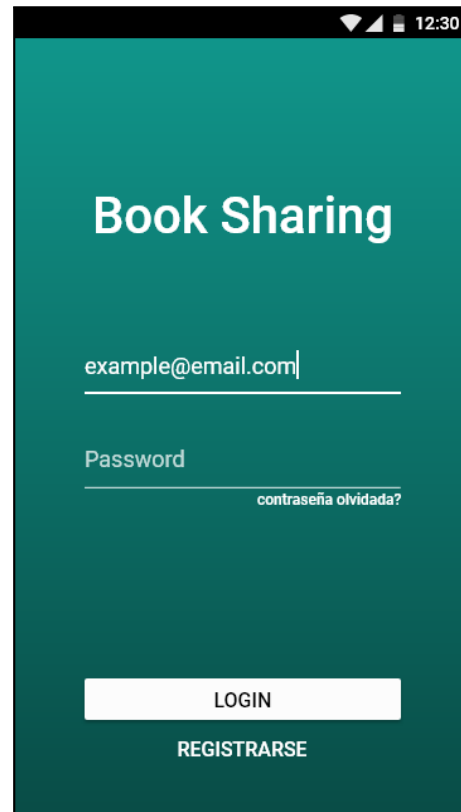
4.3 Prototipo de Alta Fidelidad

Tras realizar los primeros esbozos y definir el árbol de navegación, se ha creado el prototipo de alta fidelidad de la aplicación. Para realizar este prototipo se han tomado las siguientes decisiones de diseño:

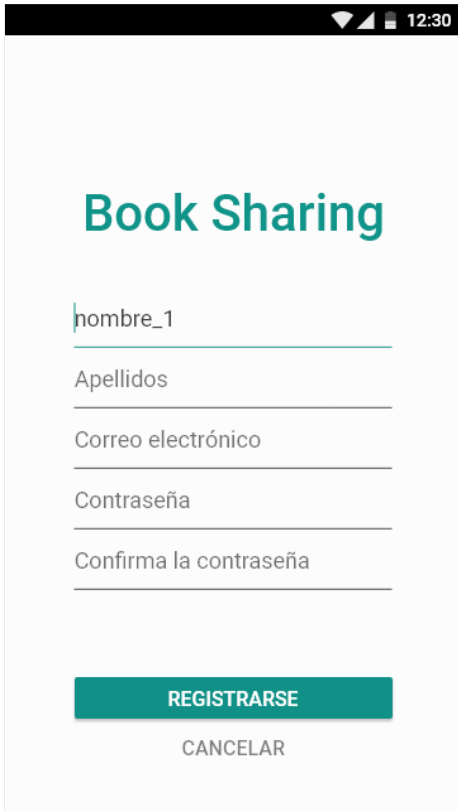
- Uso del patrón “maestro-detalle” para las principales pantallas que contienen una cuadrícula o lista como resultados, favoritos o el apartado “mis libros” de la pantalla perfil.
- También se ha optado por usar una barra de navegación inferior en las pantallas principales o “maestras”.



Pantalla de carga



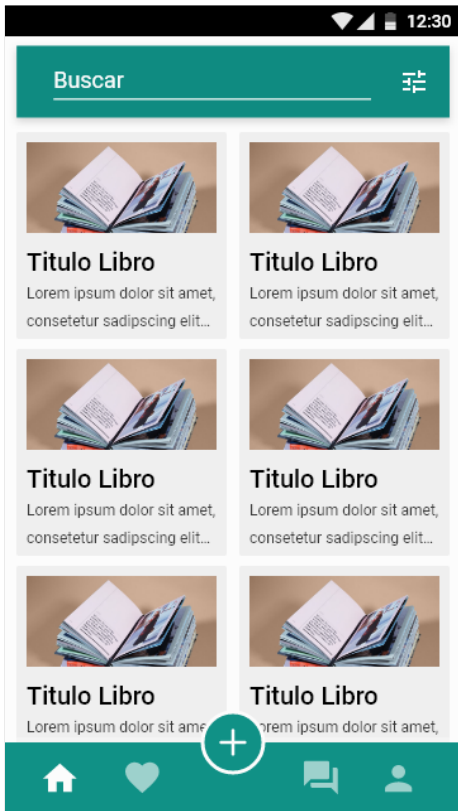
Pantalla de login



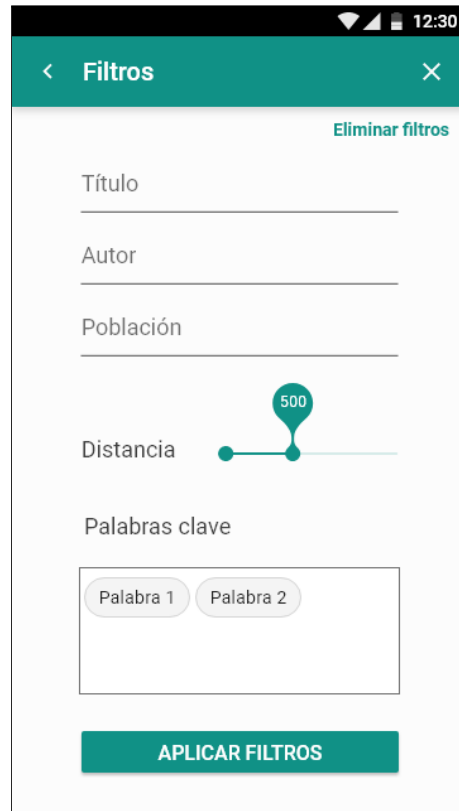
Pantalla de registro



Pantalla de recuperación



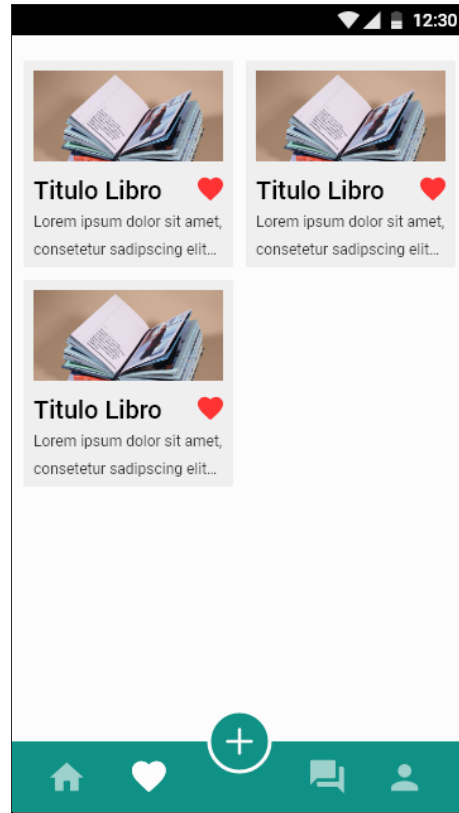
Pantalla de inicio



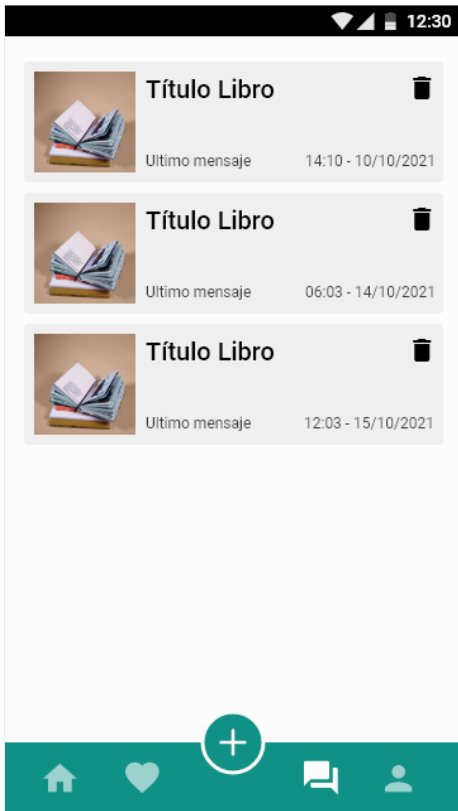
Pantalla de filtros



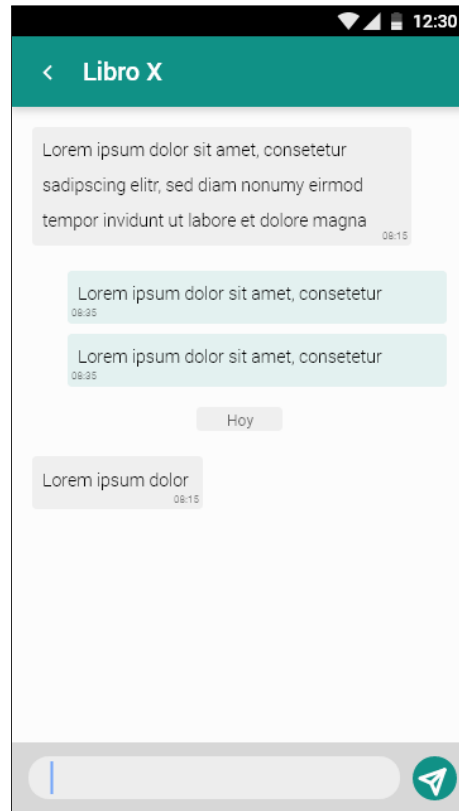
Pantalla de detalle



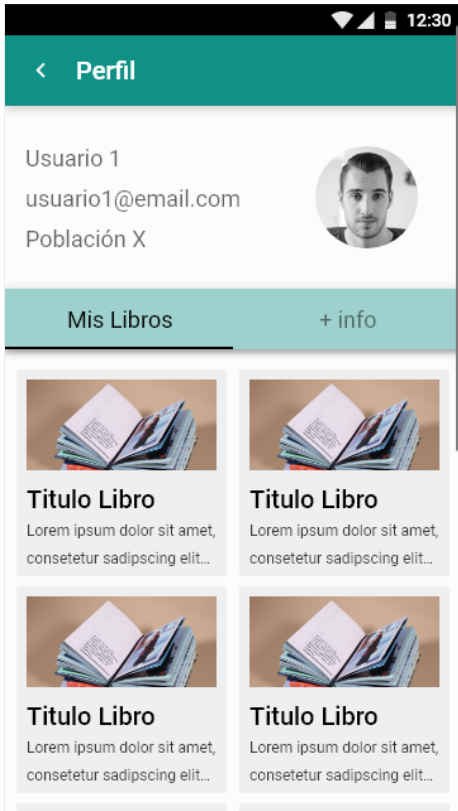
Pantalla de favoritos



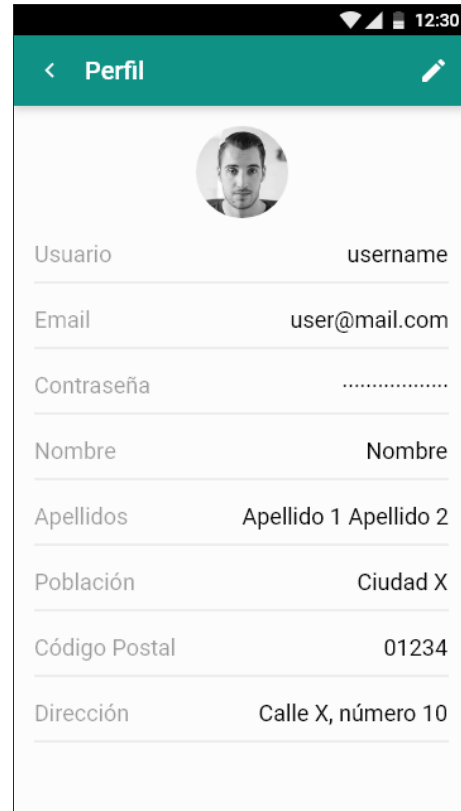
Pantalla de chats



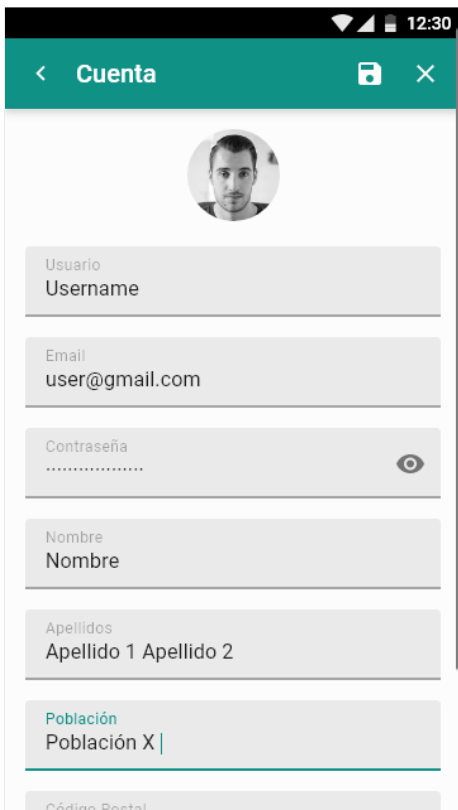
Detalle de chat



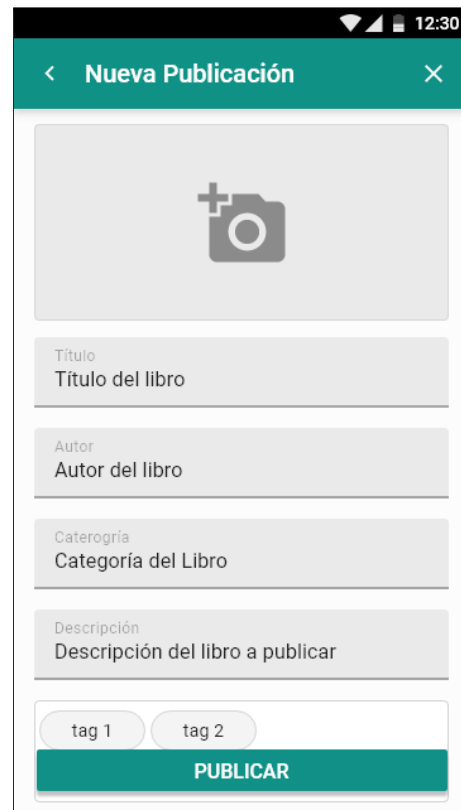
Pantalla de Perfil



Detalles de perfil



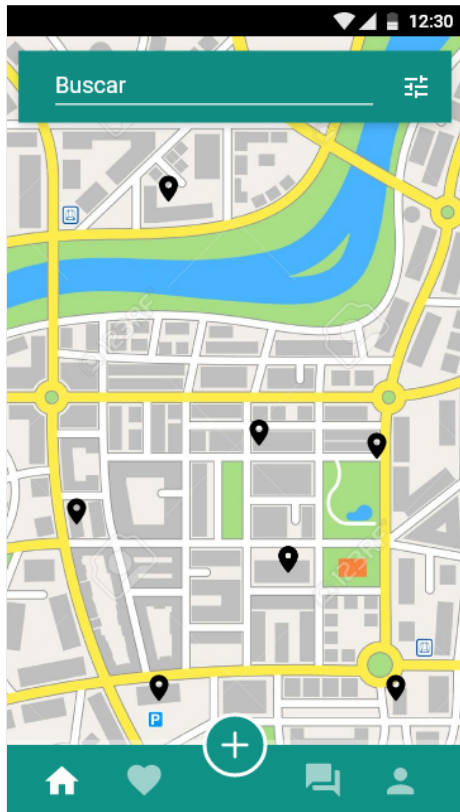
Pantalla de edición del perfil



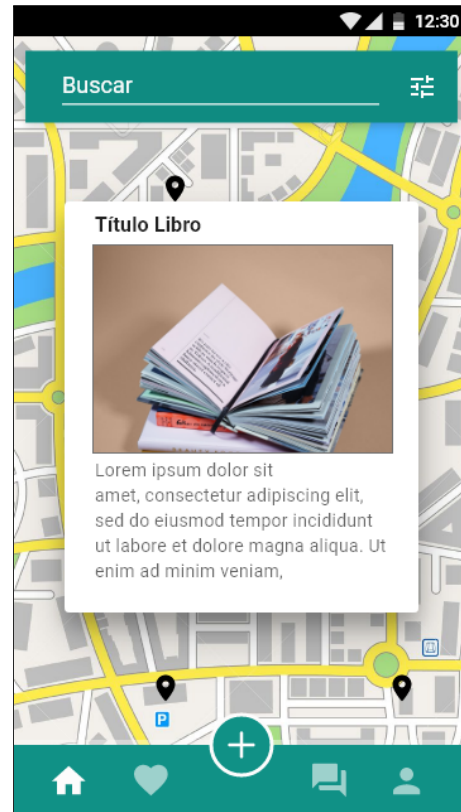
Pantalla de creación de libro

4.4 Diseños Alternativos

A continuación se muestran los distintos diseños alternativos, de algunas pantallas, estudiados para realizar el prototipo.



Pantalla de inicio



Pantalla de inicio con detalles

El prototipo se ha diseñado de manera que los resultados de una búsqueda se muestren en una cuadrícula, donde cada elemento contiene una pequeña imagen, un título del anuncio y descripción de la publicación, todo esto en un contenedor que permite hacer scroll hacia abajo de forma “infinita.” Sin embargo, aplicaciones similares proponen otros diseños donde los resultados se muestran en un mapa con marcadores.

Este diseño alternativo permite ver los anuncios que están cerca del usuario y abrir una previsualización del libro en una tarjeta emergente.



Pantalla de perfil

La pantalla de “Perfil” se ha diseñado como una vista donde se puede previsualizar los detalles más importantes del usuario como la imagen, el nombre o el correo electrónico y unas pestañas, dónde de una manera rápida pueden cargar datos como sus libros o información de la aplicación.

Una opción alternativa es montar un diseño lineal, como se puede ver en la imagen, para organizar los diferentes elementos de forma jerárquica donde se proporciona una navegación secuencial a las diferentes opciones de la pantalla. Este tipo de navegación es sencilla e intuitiva cuando no hay muchos niveles de navegación.

5. Definición de casos de uso

5.1 Diagramas de casos de uso

Tras analizar los interesados, posibles usuarios, escenarios de uso y los flujos de interacción de la aplicación se han extraído los casos de uso que a continuación se representan mediante diagramas.

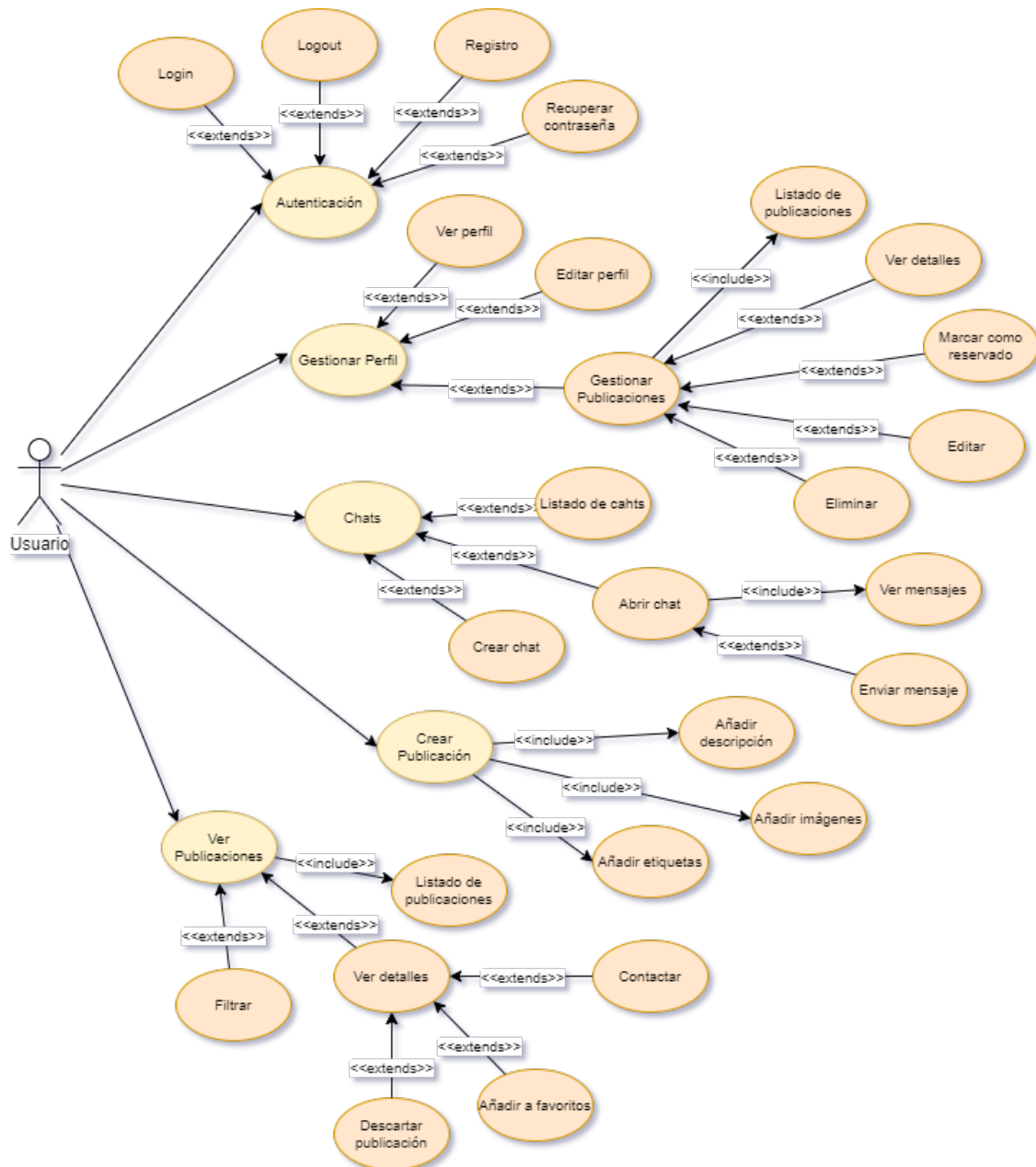


Figura 25: Diagrama casos de uso Usuario

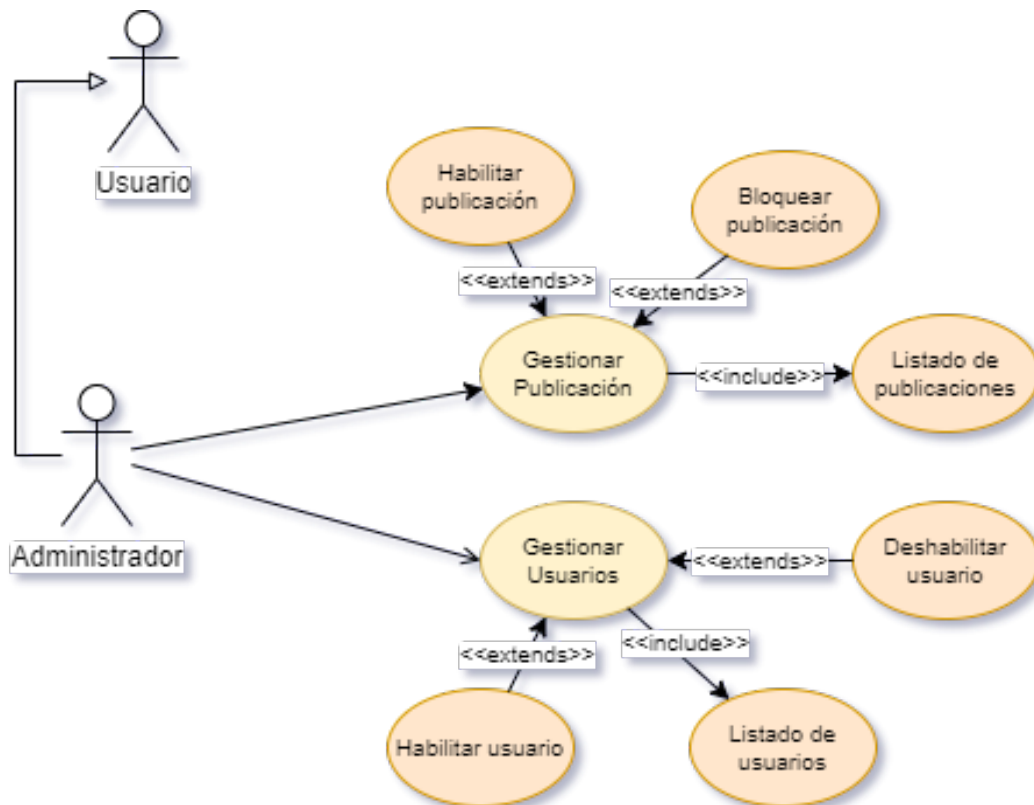


Figura 26: Diagrama casos de uso Administrador

5.2 Listado de casos de uso

Id. Caso de uso	UC-0001
Título	Un usuario se registra en el sistema
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario no está registrado ni ha iniciado sesión en la aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario introduce sus datos personales en el formulario de registro. 2. El usuario introduce sus datos bancarios. 3. El usuario introduce una contraseña. 4. El usuario presiona el botón de registro. 5. El sistema muestra la pantalla de "Inicio de sesión".
Postcondiciones	Los datos del usuario son registrados en el sistema. Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0002
Título	Un usuario inicia sesión en el sistema.
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado, pero no ha iniciado sesión en la aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario introduce su ID de usuario o correo electrónico. 2. El usuario introduce su contraseña. 3. El usuario presiona el botón de inicio de sesión.
Postcondiciones	Se comprueba que el usuario exista en la BD. Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0003
Título	Recuperar contraseña del usuario.
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado, pero no ha iniciado sesión en la aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario introduce su ID de usuario o correo electrónico. 2. El usuario presiona el botón de recuperar contraseña. 3. El sistema envía un correo con los pasos a seguir para resetear la contraseña.
Postcondiciones	Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0004
Título	Cerrar sesión
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación.

Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de perfil. 2. El usuario pulsa en el icono de “Cerrar sesión”. 3. El sistema muestra la pantalla de “Inicio de sesión”.
Postcondiciones	Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0005
Título	Ver listado de libros publicados
Actores	Cualquier tipo de usuario
Precondiciones	<p>El usuario tiene la aplicación descargada e instalada en su dispositivo.</p> <p>El usuario está registrado y ha iniciado sesión en la aplicación.</p>
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla principal de la aplicación.
Postcondiciones	Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0006
Título	Filtrar listado de libros publicados
Actores	Cualquier tipo de usuario
Precondiciones	<p>El usuario tiene la aplicación descargada e instalada en su dispositivo.</p> <p>El usuario está registrado y ha iniciado sesión en la aplicación.</p> <p>El usuario ha accedido a la pantalla principal.</p>
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa en el icono de “Filtrado”. 2. El usuario indica los filtros que quiere. 3. El usuario pulsa en el botón “Aplicar”. 4. El sistema devuelve los resultados filtrados.
Postcondiciones	Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0007
Título	Ver detalles de una libro publicado
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo.

	El usuario está registrado y ha iniciado sesión en la aplicación. El usuario ha realizado una búsqueda de libros.
Flujo	1. El usuario selecciona una publicación de la cuadrícula. 2. El sistema muestra la pantalla con los detalles del libro.
Postcondiciones	Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0008
Título	Añadir un libro publicado a favoritos
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación. El usuario ha accedido a la pantalla de detalles de una publicación
Flujo	1. El usuario pulsa el icono de favoritos situado en la barra de navegación superior. 2. El sistema actualiza el color del icono.
Postcondiciones	Se añade la publicación como favoritos del usuario en la BD. Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0009
Título	Descartar la publicación de un libro para futuras búsquedas.
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación. El usuario ha accedido a la pantalla de detalles de una publicación
Flujo	1. El usuario pulsa el icono de descartar situado en la barra de navegación superior. 2. El sistema navega a la vista anterior, mostrando los resultados de la búsqueda.

Postcondiciones	Se añade la publicación como descartada del usuario en la BD. Se registra la acción del usuario para posteriores analíticas.
------------------------	---

Id. Caso de uso	UC-0010
Título	Crear una publicación de un libro
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón “+” situado en la parte inferior. 2. El sistema muestra la pantalla con el formulario para crear una publicación. 3. El usuario completa los campos requeridos. 4. El usuario pulsa en el botón “Publicar” 5. El sistema muestra la pantalla con los detalles del libro publicado.
Postcondiciones	Se crea una nueva entrada de libro publicada asociada al usuario en la BD. Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0011
Título	Contactar con un usuario
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación. El usuario ha accedido a la pantalla de detalles de una publicación
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Contactar” 2. El sistema muestra la pantalla con el chat entre el usuario y publicador del libro.
Postcondiciones	Se crea un nuevo chat en la BD entre el usuario y el usuario que ha publicado el libro. Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0012
Título	Ver listados de chats
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación.
Flujo	1. El usuario pulsa el icono de chats, situado en la barra de navegación inferior. 2. El sistema muestra la pantalla con el listado de chats
Postcondiciones	Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0013
Título	Ver los mensajes de un chat
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación. El usuario ha accedido a la pantalla de "Chats"
Flujo	1. El usuario selecciona un chat. 2. El sistema muestra la pantalla con el chat entre los dos usuarios.
Postcondiciones	Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0014
Título	Enviar un mensaje
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación. El usuario ha accedido a los detalles de un chat desde la pantalla de "Chats"
Flujo	1. El usuario pulsa en el espacio para escribir el mensaje.

	<ol style="list-style-type: none"> 2. Una vez escrito, el usuario pulsa en el botón enviar. 3. El sistema actualiza la pantalla con los nuevos mensajes.
Postcondiciones	<p>Se registra el nuevo mensaje en la BD. Se envía una notificación al otro usuario. Se registra la acción del usuario para posteriores analíticas.</p>

Id. Caso de uso	UC-0015
Título	Ver el perfil del usuario
Actores	Cualquier tipo de usuario
Precondiciones	<p>El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación.</p>
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el icono de perfil, situado en la barra de navegación inferior. 2. El sistema muestra la pantalla con los detalles del perfil del usuario.
Postcondiciones	Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0016
Título	Editar el perfil del usuario
Actores	Cualquier tipo de usuario
Precondiciones	<p>El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación. El usuario ha accedido a la pantalla de "Perfil".</p>
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa en el botón de editar. 2. El sistema muestra la pantalla de edición. 3. El usuario realiza cambios en el perfil. 4. El usuario pulsa en el botón "Guardar". 5. El sistema muestra un mensaje de confirmación. 6. El sistema muestra la pantalla con los detalles del usuario.
Postcondiciones	<p>Se actualizan los datos del usuario en la BD. Se registra la acción del usuario para posteriores analíticas.</p>

Id. Caso de uso	UC-0017
Título	Ver las publicaciones creadas por un usuario.
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación. El usuario ha accedido a la pantalla de "Perfil".
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa en la pestaña "Mis libros". 2. El sistema muestra el listado de publicaciones del usuario.
Postcondiciones	Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0018
Título	Editar publicación creada por un usuario.
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo. El usuario está registrado y ha iniciado sesión en la aplicación. El usuario ha accedido a la pantalla de "Mis libros".
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona un libro. 2. El sistema muestra la pantalla con los detalles del libro. 3. El usuario pulsa el botón "Editar libro". 4. El sistema muestra la pantalla de edición. 5. El usuario realiza los cambios y pulsa el botón "Guardar". 6. El sistema muestra un mensaje de confirmación. 7. El sistema carga la pantalla con los detalles del libro.
Postcondiciones	Se actualizan los detalles del libro en la BD, Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0019
Título	Eliminar publicación creada por un usuario.
Actores	Cualquier tipo de usuario
Precondiciones	El usuario tiene la aplicación descargada e instalada en su dispositivo.

	<p>El usuario está registrado y ha iniciado sesión en la aplicación.</p> <p>El usuario ha accedido a la pantalla de “Mis libros”.</p>
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona un libro. 2. El sistema muestra la pantalla con los detalles del libro. 3. El usuario pulsa el botón “Eliminar libro”. 4. El sistema muestra un mensaje de confirmación. 5. El sistema carga la pantalla con el listado de publicaciones del usuario.
Postcondiciones	<p>Se elimina el libro de la BD.</p> <p>Se registra la acción del usuario para posteriores analíticas.</p>

Id. Caso de uso	UC-0020
Título	Marcar como reservada una publicación creada por un usuario.
Actores	Cualquier tipo de usuario
Precondiciones	<p>El usuario tiene la aplicación descargada e instalada en su dispositivo.</p> <p>El usuario está registrado y ha iniciado sesión en la aplicación.</p> <p>El usuario ha accedido a la pantalla de “Mis libros”.</p>
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona un libro. 2. El sistema muestra la pantalla con los detalles del libro. 3. El usuario pulsa el botón “Marcar como reservado”. 4. El sistema muestra un mensaje de confirmación. 5. El sistema carga la pantalla con el listado de publicaciones del usuario.
Postcondiciones	<p>Ser marca como reservado el libro en la BD.</p> <p>Se registra la acción del usuario para posteriores analíticas.</p>

Id. Caso de uso	UC-0021
Título	Ver listado de publicaciones marcadas como favoritas.
Actores	Cualquier tipo de usuario
Precondiciones	<p>El usuario tiene la aplicación descargada e instalada en su dispositivo.</p> <p>El usuario está registrado y ha iniciado sesión en la</p>

	aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el icono de favoritos, situado en la barra de navegación inferior. 2. El sistema muestra la pantalla con el listado de libros marcados como favoritos.
Postcondiciones	Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0022
Título	Eliminar publicación de favoritos
Actores	Cualquier tipo de usuario
Precondiciones	<p>El usuario tiene la aplicación descargada e instalada en su dispositivo.</p> <p>El usuario está registrado y ha iniciado sesión en la aplicación.</p> <p>El usuario ha accedido a la pantalla de “Favoritos”.</p>
Flujo	<ol style="list-style-type: none"> 1. El usuario arrastra de manera horizontal la tarjeta del libro. 2. El sistema muestra un mensaje de confirmación. 3. El sistema actualiza la lista de libros favoritos.
Postcondiciones	<p>Se elimina de favoritos el libro en la BD.</p> <p>Se registra la acción del usuario para posteriores analíticas.</p>

Id. Caso de uso	UC-0023
Título	Bloquear publicación de un libro.
Actores	Usuario administrador
Precondiciones	<p>El usuario está registrado y ha iniciado sesión en la aplicación.</p> <p>El usuario tiene el perfil administrador.</p> <p>El usuario ha accedido a la pantalla con los detalles un libro.</p>
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Bloquear”. 2. El sistema muestra un mensaje de confirmación.
Postcondiciones	<p>Se actualiza la información del libro en la BD.</p> <p>Se envía un correo electrónico al usuario creador de la publicación notificándole la acción.</p> <p>Se registra la acción del usuario para posteriores analíticas.</p>

Id. Caso de uso	UC-0024
Título	Desbloquear publicación de un libro.
Actores	Usuario administrador
Precondiciones	El usuario está registrado y ha iniciado sesión en la aplicación. El usuario tiene el perfil administrador. El usuario ha accedido a la pantalla con los detalles un libro que ha sido bloqueado.
Flujo	1. El usuario pulsa el botón “Desbloquear”. 2. El sistema muestra un mensaje de confirmación.
Postcondiciones	Se actualiza la información del libro en la BD. Se envía un correo electrónico al usuario creador de la publicación notificándole la acción. Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-00245
Título	Ver el listado de usuarios
Actores	Usuario administrador
Precondiciones	El usuario está registrado y ha iniciado sesión en la aplicación. El usuario tiene el perfil administrador.
Flujo	1. El usuario pulsa el botón “Bloquear”. 2. El sistema muestra un mensaje de confirmación.
Postcondiciones	Se actualiza la información del libro en la BD. Se envía un correo electrónico al usuario creador de la publicación notificándole la acción. Se registra la acción del usuario para posteriores analíticas.

Id. Caso de uso	UC-0026
Título	Deshabilitar un usuario.
Actores	Usuario administrador
Precondiciones	El usuario está registrado y ha iniciado sesión en la aplicación. El usuario tiene el perfil administrador. El usuario ha accedido al perfil de un usuario.

Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Deshabilitar”. 2. El sistema muestra un mensaje de confirmación.
Postcondiciones	<p>Se deniega el acceso a la aplicación en Firebase. Se envía un correo electrónico al usuario notificándole la acción. Se registra la acción del usuario para posteriores analíticas.</p>

Id. Caso de uso	UC-0027
Título	Habilitar un usuario deshabilitado.
Actores	Usuario administrador
Precondiciones	<p>El usuario está registrado y ha iniciado sesión en la aplicación. El usuario tiene el perfil administrador. El usuario ha accedido al perfil de un usuario.</p>
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Habilitar”. 2. El sistema muestra un mensaje de confirmación.
Postcondiciones	<p>Se autoriza el acceso a la aplicación en Firebase. Se envía un correo electrónico al usuario notificándole la acción. Se registra la acción del usuario para posteriores analíticas.</p>

6. Diseño de la arquitectura

6.1 Arquitectura del sistema

El siguiente diagrama muestra los distintos elementos que forman el ecosistema de la solución diseñada.

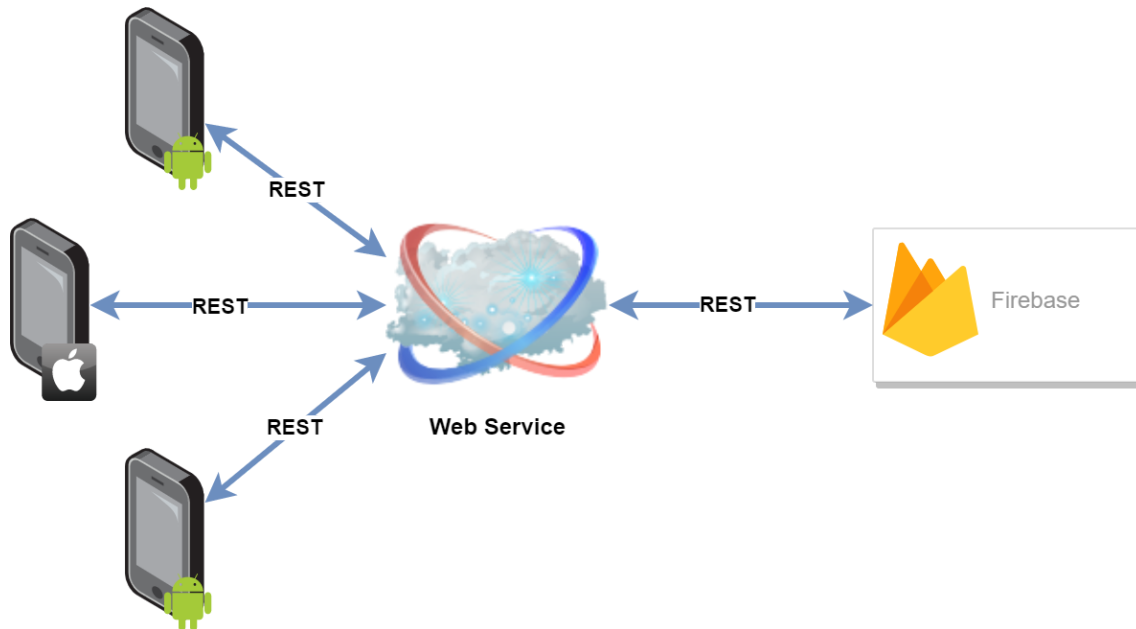


Figura 27: Diagrama del sistema

6.2 Arquitectura de la Base de Datos

El diagrama se centra en mostrar las relaciones entre los distintos documentos en vez de mostrar todos los elementos o atributos de estos, ya que pueden variar a medida que la aplicación se va desarrollando.

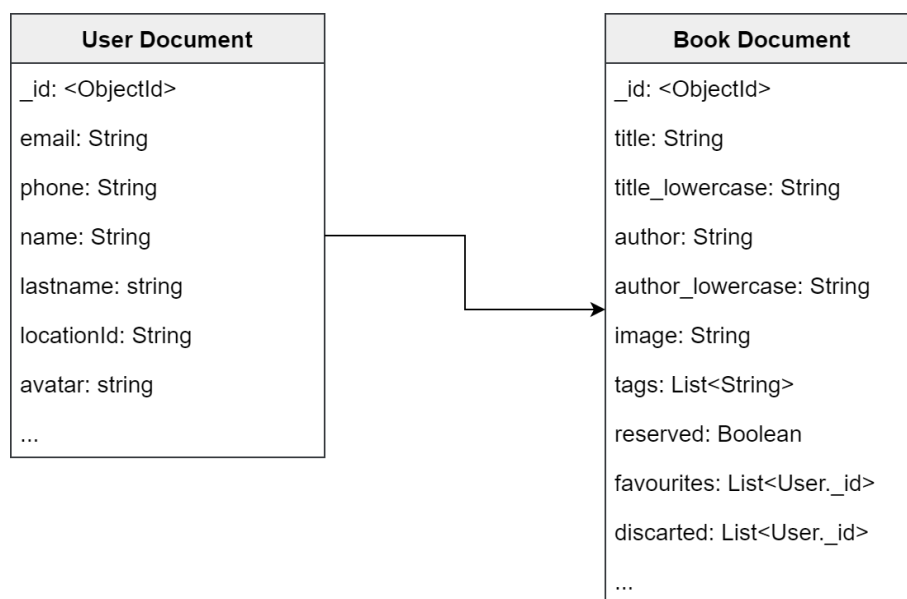


Figura 28: Diagrama de la base de datos

Destacar que Firebase ofrece un servicio de base de datos no relacional, por lo que se trabaja con documentos en vez de clases.

6.3 Arquitectura de la aplicación

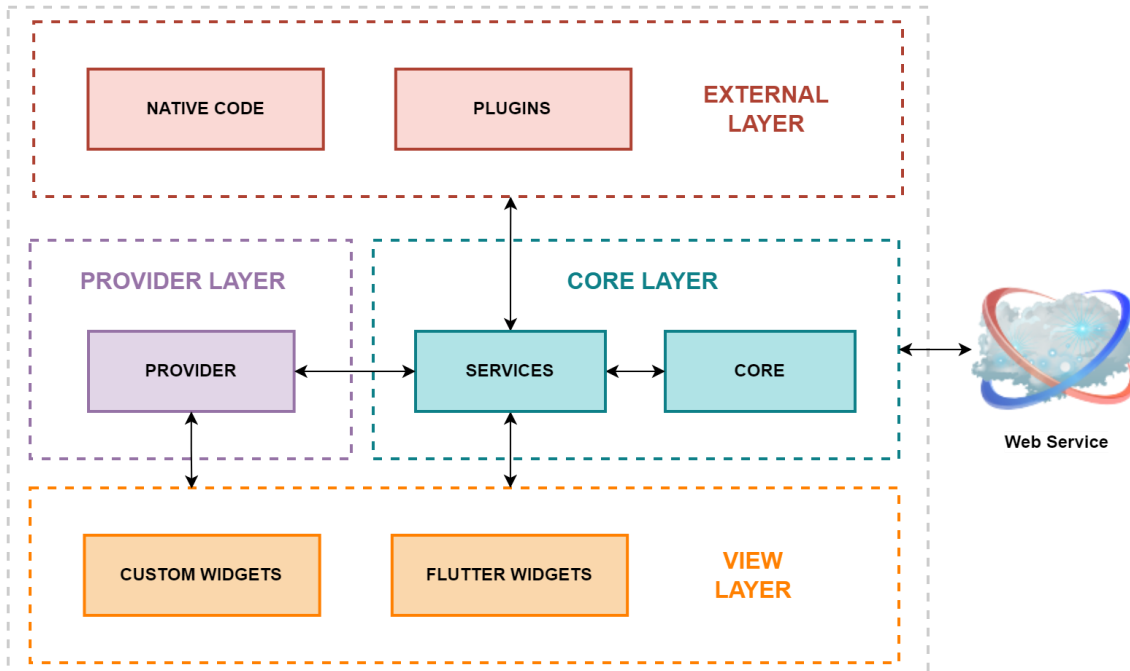


Figura 29: Diagrama de la aplicación

El diagrama *Figura 29: Diagrama de la aplicación* muestra como se distribuyen los diferentes componentes que forman la aplicación móvil. Estos están distribuidos en cuatro capas:

- **View layer:** encargada de mostrar los datos al usuario y reaccionar a las acciones que este realiza.
- **Provider layer:** capa encargada de proveer de datos a la vista, e interactuar con los distintos servicios de la aplicación.
- **Core o Service layer:** capa que concentra la lógica de mercado de la aplicación y de comunicación con servicios externos, y modelos de datos de la aplicación. Además de ofrecer datos a la capa provider y la vista.
- **External Layer:** Capa que contiene todas las librerías externas, código nativo de la plataforma con las que interactúa la aplicación.

Gracias a este diseño por capas, podemos decir que nuestra aplicación es:

- **Clara:** La separación del código en distintos componentes, permite obtener clases más reducidas y simples que facilita la comprensión del código.

- **Reutilizable:** Con la creación de widgets simples y atomizados, estos pueden ser reutilizados en múltiples vistas agilizando el proceso de desarrollo.
- **Eficiente:** El uso de la librería Provider y la programación reactiva mediante RxDart permite que la aplicación solo reconstruya las vistas y widgets necesarios cuando los datos son actualizados.
- **Escalable:** La baja dependencia entre el modelo y los otros componentes permite simplificar el esfuerzo necesario para expandir el proyecto y su código.
- **Mantenible:** El bajo acople y bajas dependencias entre elementos de la aplicación, junto con la claridad del código, favorece el mantenimiento de la aplicación.

7. Revisión de la planificación

Durante la implementación del proyecto se han tomado distintas decisiones que han implicado cambios respecto a la planificación inicial descrita en este documento. A continuación procedo a enumerar las decisiones tomadas y los principales motivos que me han llevado a tomarlas.

7.1 Cambio de tecnología

Se ha sustituido la tecnología inicialmente escogida para desarrollar la aplicación, React Native, por Flutter. Los motivos que han llevado a tomar dicha decisión son:

- La experiencia y conocimientos, previos al inicio de la implementación, del framework Flutter respecto a React Native.
- El sobre esfuerzo que suponía aprender el funcionamiento y conocimientos de React Native junto al limitado tiempo de desarrollo que disponía.
- El amplio soporte de Flutter a otras plataformas como aplicaciones escritorio para Linux, Mac o Windows y desarrollo Web, que puede beneficiar en futuras mejoras del producto implementado.

7.2 Limitación de funcionalidades

Durante la fase de desarrollo han surgido distintas dificultades, como por ejemplo en la integración con Firebase para la autenticación de usuarios, almacenaje de recursos y base de datos, a demás de los problemas obtenidos durante la integración de la API de Google Places para la gestión de la localización de los usuarios y anuncios publicados, que han consumido un mayor tiempo, del inicialmente planificado, para poder ser desarrolladas correctamente.

Estos contratiempos han supuesto algunos requisitos identificados en la planificación no hayan sido implementados:

ID	Descripción
SR-FR-0012	La aplicación debe permitir la comunicación entre usuarios con un chat propio de la aplicación.
SR-FR-0014	Un usuario puede bloquear a otros usuarios para que estos no vean un anuncio en sus búsquedas.

Figura 30: Tabla de requisitos no desarrollados

Para el requisito *SR-FR-0012* se ha implementado dos vías de comunicación alternativa, mediante el uso de la librería *url_launcher*,

podemos realizar una llamada telefónica al usuario creador del anuncio en caso de que indique su número de teléfono o bien enviar un correo electrónico.

Sin embargo, para el requisito *SR-FR-0014* no se ha implementado ninguna alternativa.

7.3 Sección Administrador

Las funciones de administración han sido delegadas al servicio Firebase, ya que dispone herramientas que permiten ejecutar dichas funciones sin necesidad de implementar un módulo de dedicado en la aplicación. Los requisitos que han pasado a estar gestionados desde Firebase son:

ID	Descripción
SR-FR-0016	Un usuario administrador debe poder eliminar o pedir revisión por parte de los usuarios de un producto.
SR-FR-00147	Un usuario administrador debe poder revisar los productos reportados.

Figura 31: Tabla de requisitos gestionados por Firebase

7.4 Limitación de la evaluación del producto

Se ha sustituido la validación de test automáticos end-to-end por una validación funcional donde posibles usuarios participen en el proceso de validación de la aplicación.

Destacar que debido a una limitación de recursos que se disponían durante la fase de implementación la aplicación únicamente se ha podido generar para la plataforma Android. Por lo mismo, la evaluación del producto desarrollado únicamente se ha podido hacer sobre la plataforma mencionada.

8. Implementación

8.1 Estructura del proyecto

Se ha estructurado el código de la aplicación por módulos, estos módulos a su vez pueden contener sub módulos o carpetas para organizar los ficheros por tipología y/o funcionalidad. El proyecto quedaría estructurado en los siguientes módulos:

- **Core:** contiene los modelos de datos, servicios y los métodos o clases comunes de la aplicación.
- **Providers:** contiene las clases que son usadas en el patrón provider.
- **View:** contiene las distintas pantallas de la aplicación, los widgets comunes y los componentes de navegación. Las pantallas puede tener widgets locales.

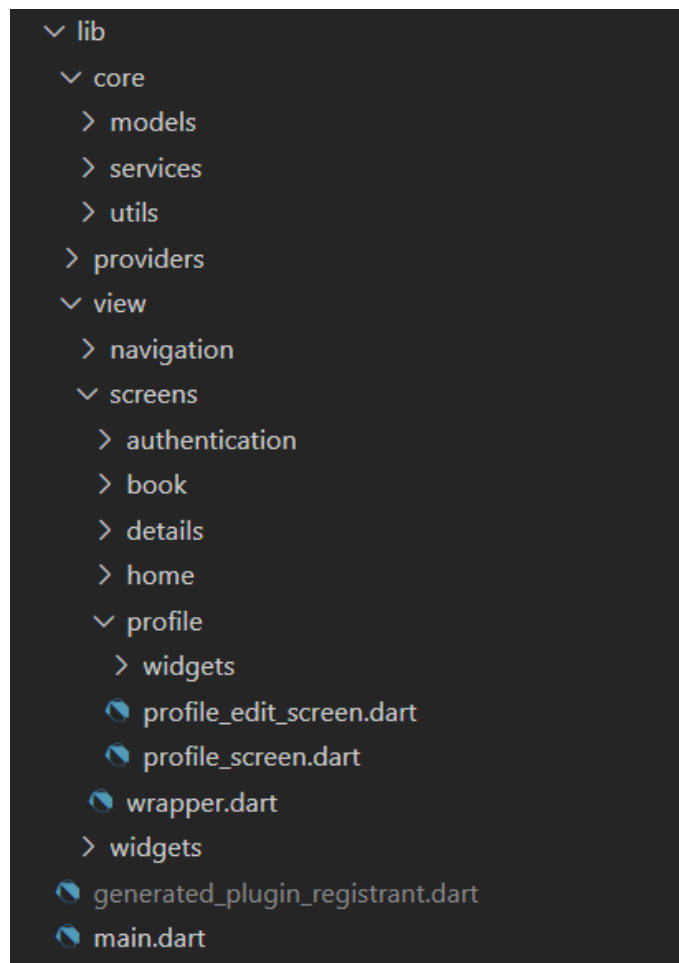


Figura 32: Estructura del proyecto

8.2 Librerías

Para el desarrollo de la aplicación se han usado unas cuantas librerías, la mayoría de ellas son imprescindibles para el funcionamiento de la aplicación, mientras que otras se han escogido por facilitar el trabajo y ahorrar el desarrollo de ciertos componentes.

```
firebase_core: ^1.2.0
firebase_auth: ^1.2.0
cloud_firestore: ^2.2.0
get: ^4.3.8
provider: ^6.0.1
flutter_spinkit: ^5.1.0
image_picker: ^0.8.4+4
firebase_storage: ^10.1.0
http: ^0.13.4
uuid: ^3.0.5
geolocator: ^7.7.1
intl_phone_field: ^2.1.0
url_launcher: ^6.0.13
rxdart: ^0.27.2
flutter_dotenv: ^5.0.2
```

Figura 33: Dependencias del proyecto

Las librerías, `firebase_core`, `firebase_auth`, `cloud_firestore` y `firebase_storage` son librerías necesarias para la comunicación con Firebase. A continuación se listan el resto de librerías con una breve descripción:

- **get:** Librería para la gestión del contexto y navegación, además de incluir muchas utilidades que facilitan el desarrollo.
- **provider:** Implementación del patrón provider en Flutter.
- **flutter_spinkit:** librería que permite crear distintos indicadores de carga de forma sencilla.
- **image_picker:** Librería usada para permitir al usuario seleccionar imágenes de la galería.
- **http:** Librería que habilita la comunicación mediante el protocolo HTTP para Flutter.
- **uuid:** Librería que genera identificadores uuid de manera aleatoria.
- **geolocator:** Gestión de la geolocalización del dispositivo.
- **intl_phone_field:** Campo de texto que contiene todas las validaciones para un número de teléfono además de permitir seleccionar el prefijo.
- **url_launcher:** Librería para la ejecución de URI.
- **rxdart:** Implementación de programación reactiva para Dart.
- **flutter_dotenv:** Librería para la gestión de ficheros de entorno `.env`

8.3 Soluciones implementadas

Filtrado de resultados

En la siguiente captura se muestra la lógica de la función encargada de filtrar los libros de la aplicación. Esta función realiza dos operaciones de filtrado, la primera es montar la consulta de Firebase con los filtros recibidos por parámetro y obtener los datos de Firebase.

Una vez obtenidos los datos de Firebase, se aplican filtros que Firebase no soporta como el de libros descartados o el filtro por distancia.

```
// Defult filter instance is 15km --> 15.0000m
double distance = 15000;
bool filterByDistance = false;
// Basic query structure
Query query = bookCollection.where('user', isNotEqualTo: _uid)
    .where('changed', isEqualTo: false);
// Create Filters based oon FilterObject
for (var element in filters) {
    if (element.value is List) {
        // Manage List filter
        query = query.where(element.key, arrayContainsAny: element.value);
    } else if (element.value is String &&
        element.key.contains('_lowercase')) {
        // Manage Spring filters
        query = query
            .where(element.key, isEqualTo: (element.value as String).toLowerCase());
    } else if (element.key == 'distance') {
        // Manage distance filter
        filterByDistance = true;
        distance = element.value * 1000;
    } else {
        query = query.where(element.key, isEqualTo: element.value);
    }
}
// Sort result by timestamp --> newest on top
query = query.orderBy('user').orderBy('timestamp', descending: true);
//Perform request
QuerySnapshot books = await query.get();
List<Book> result = books.docs
    .map((doc) =>Book.fromJson(doc.data() as Map<String, dynamic>, doc.id)).toList();
// Remove User discarted results
result = result.where((element) => !element.discarted.contains(_uid)).toList();
// Apply distance filter
if (filterByDistance) {
    result = result
        .where((element) => distanceFilter(element, _position!) < distance).toList();
}
// Sort Results by nearest
result.sort((a, b) => distanceFilter(a, _position!)
    .compareTo(distanceFilter(b, _position!)));
return result;
```

Figura 34: Función de filtrado de libros

Es importante remarcar una limitación encontrada con Firebase, y es que en las comparaciones en campos de texto distingue mayúsculas y minúsculas, además de hacer comparaciones exactas y no parciales. Para solucionar el primer problema se han tenido que duplicar los atributos de texto por los que se puede filtrar, es por ello que existen atributos con el sufijo *_lowercase*.

Estos atributos almacenan la misma información que su homólogo, pero en minúsculas para permitir realizar búsquedas sin distinguir mayúsculas y minúsculas.

Filtrado y Ordenado por distancia

Con el fin de filtrar los resultados por distancia u ordenarlos de más cercano a más lejano, se hace uso de la librería *Geolocator*. Esta dispone de una función, *distanceBetween*, que dadas dos geolocalizaciones devuelve la distancia que hay entre ellas en metros.

```
double distanceFilter(Book book, Position position) {
    // Use Geolocator to calculate distance between to location points
    final double distance = Geolocator.distanceBetween(book.location[0],
        book.location[1], position.latitude, position.longitude);
    return distance;
}
```

Figura 35: Función de cálculo de distancia

Providers

La aplicación hace uso del patrón Provider para que distintos componentes de la aplicación reaccionen a cambios sin necesidad de volver a compilar todo el árbol de widgets.

```
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    // Return providers
    return MultiProvider(
      providers: [
        ChangeNotifierProvider<Filter>(create: (_) => Filter()),
        StreamProvider<MyUser?>.value(
          value: AuthService().user, initialData: null)
      ],
      child: GetMaterialApp(
        debugShowCheckedModeBanner: false,
        theme: ThemeData(
          brightness: Brightness.light,
          primarySwatch: Colors.teal,
          inputDecorationTheme: const InputDecorationTheme(
            hintStyle: TextStyle(
              color: Color(0xFF666666),
              fontSize: 14),
          ),
          iconTheme: const IconThemeData(size: 17),
        ),
        title: 'BookSharing',
        home: const Wrapper(),
        routes: {
          HomeScreen.route: (context) => const HomeScreen(),
          DetailsScreen.route: (context) => const DetailsScreen(),
          ProfileScreen.route: (context) => const ProfileScreen(),
          CreateBookScreen.route: (context) => const
CreateBookScreen(),
          FilterScreen.route: (context) => const FilterScreen(),
          ResetPassword.route: (context) => const ResetPassword()
        },
      ),
    );
  }
}
```

Figura 36: Providers de la aplicación

Los providers aplicados en la aplicación son dos, el primero para gestionar y mantener los filtros aplicados entre pantallas de la aplicación. Y el último para gestionar la autenticación del usuario en la aplicación.

Gestión de la autenticación

Para gestionar la autenticación en la aplicación, se escuchan los cambios que pueda haber en la autenticación de Firebase Auth.

```
Stream<MyUser?> get user {  
  // Listen to Firebase auth changes.  
  return _auth  
    .authStateChanges()  
    .map((User? user) => _userfromFirebase(user));  
}
```

Figura 37: Stream de eventos de la autenticación

De manera que, en función del valor emitido por Firebase Auth se muestra la aplicación o se redirige a la pantalla de inicio de sesión.

```
class Wrapper extends StatelessWidget {  
  const Wrapper({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    final user = Provider.of<MyUser?>(context);  
    if (user == null) {  
      return const AuthenticateScreen();  
    } else {  
      // Init Books Service with auth user  
      BooksService().uid = user.uid;  
      // Gather curren position  
      return FutureBuilder<Position>(  
        future: LocationService.determinePosition(),  
        builder: (context, snapshot) {  
          // Store location in Books Service  
          final position = snapshot.data;  
          BooksService().position = position!;  
          // Display navigation Widget  
          return const NavigationWidget();  
        });  
    }  
  }  
}
```

Figura 38: Gestión de autenticación

Localización

Para el buen funcionamiento de la app es necesario que el usuario permita que la aplicación haga uso de su localización.

Para ello se ha seleccionado la librería *Geolocator*, que permite obtener la localización del dispositivo con gran precisión en caso de que el permiso de localización este habilitado. Si este no está habilitado, se emite una solicitud de permisos y se recupera dicha localización una vez habilitados los permisos pertinentes.

```
static Future<Position> determinePosition() async {
  bool serviceEnabled;
  LocationPermission permission;

  // Test if location services are enabled.
  serviceEnabled = await Geolocator.isLocationServiceEnabled();
  if (!serviceEnabled) {
    // Return location error
    return Future.error('Location services are disabled.');
```

```
  }

  permission = await Geolocator.checkPermission();
  if (permission == LocationPermission.denied) {
    //Request Location permission
    permission = await Geolocator.requestPermission();
    if (permission == LocationPermission.denied) {
      // Return location error
      return Future.error('Location permissions are denied');
```

```
    }
  }
  if (permission == LocationPermission.deniedForever) {
    // Permissions are denied forever
    return Future.error(
      'Location permissions are permanently denied, we cannot request permissions.');
```

```
  }
}
```

Figura 39: Obtención de la localización

En la implementación, se usan otras soluciones para obtener la localización de un libro en el momento de crearlo o editarlo, o bien, al editar el perfil de usuario.

En este caso se hace uso de una API de Google llamada Places, que nos ha permitido desarrollar un buscador de localizaciones, el cual, consulta a la API posibles resultados a medida que el usuario va completando el nombre de una población.

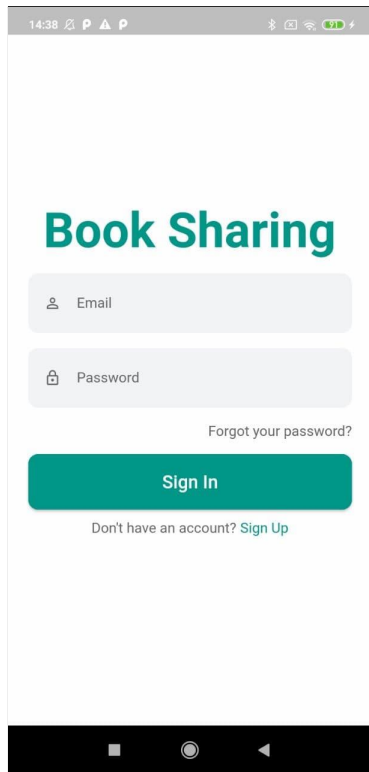
```

Future<List<Suggestion>> fetchSuggestions(String input, String lang) async {
    // Define type of information
    const String type = '(cities)';
    // Define country to search in
    const String country = 'es';
    // Create request URI
    final request =
        'https://maps.googleapis.com/maps/api/place/autocomplete/json?input=$input&types=$
type&language=$lang&components=country:$country&key=$_apiKey&sessiontoken=$sessionToken';
    final response = await http.get(Uri.parse(request));
    if (response.statusCode == 200) {
        final result = jsonDecode(response.body);
        if (result['status'] == 'OK') {
            // Create List of Suggestions from returned prefictions
            return result['predictions']
                .map<Suggestion>((p) => Suggestion(p['place_id'], p['description']))
                .toList();
        }
        if (result['status'] == 'ZERO_RESULTS') {
            // return empty list
            return [];
        }
        throw Exception(result['error_message']);
    } else {
        throw Exception('Failed to fetch suggestion');
    }
}

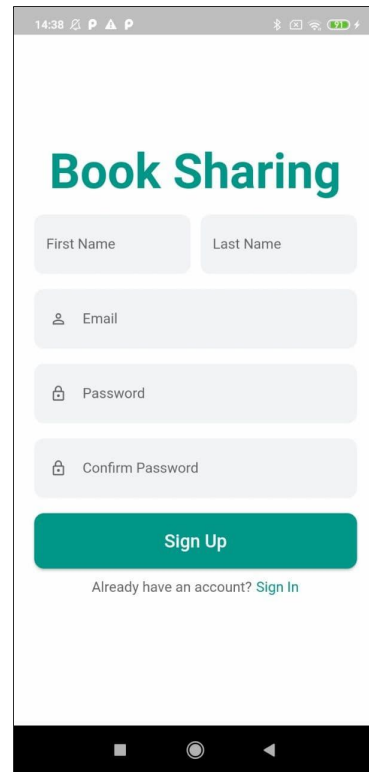
```

Figura 40: Consulta a la api de Google

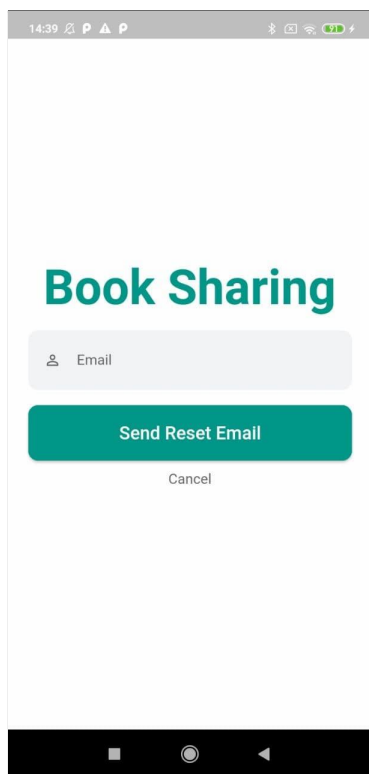
9. Resultados de la implementación



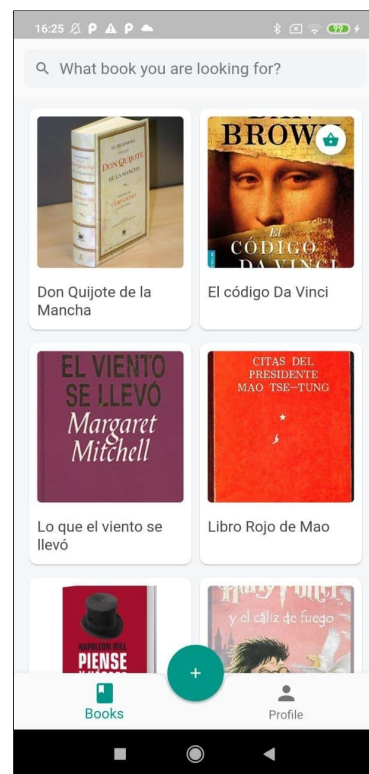
Pantalla de login



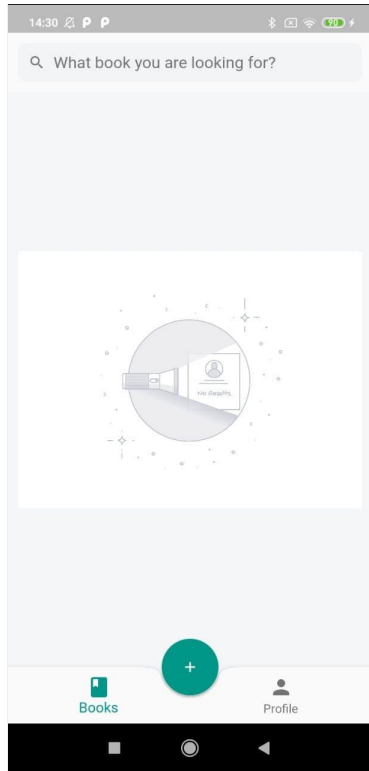
Pantalla de registro



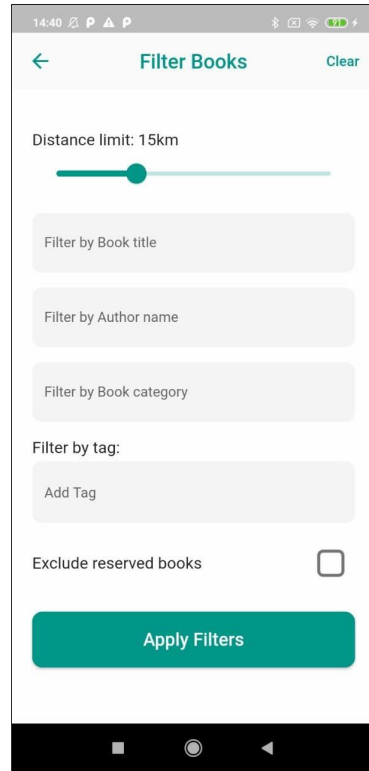
Pantalla de recuperación



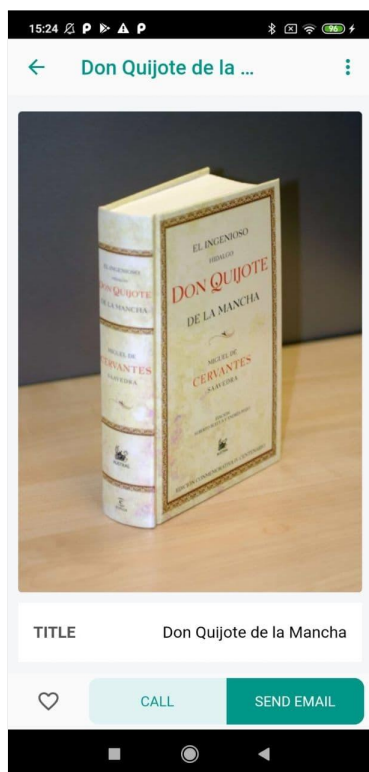
Pantalla de inicio



Pantalla de inicio sin resultados



Pantalla de filtros



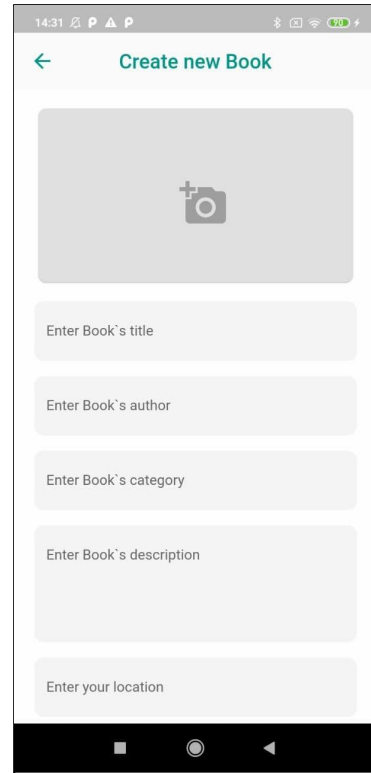
Pantalla de detalle



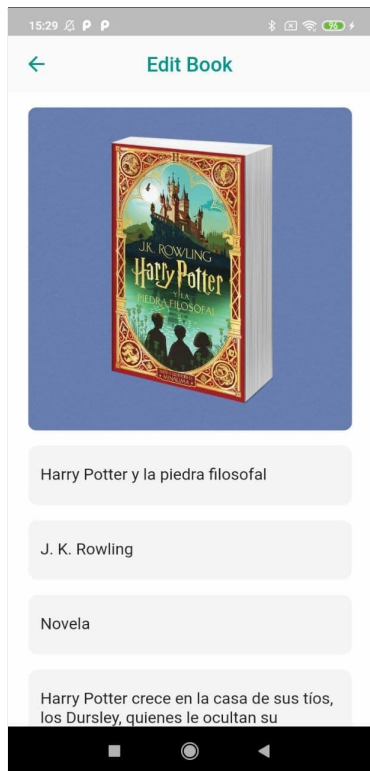
Pantalla de detalle



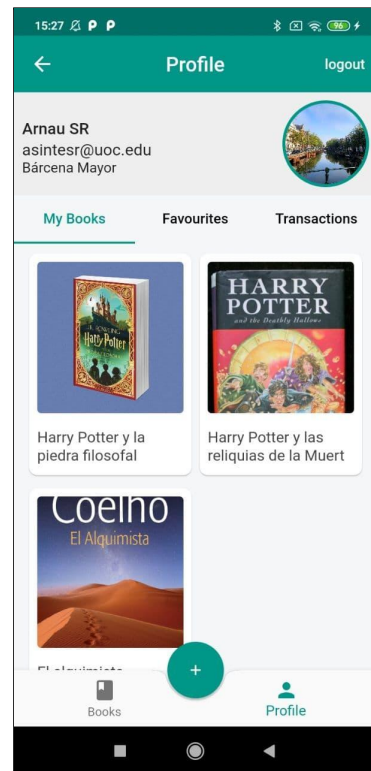
Detalles de libro de un usuario



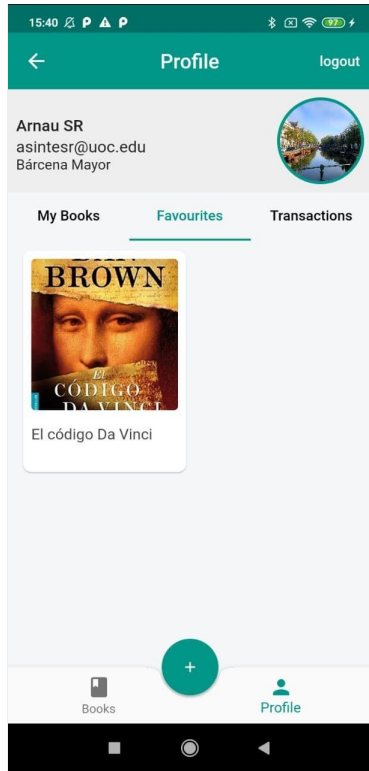
Pantalla de creación de libro



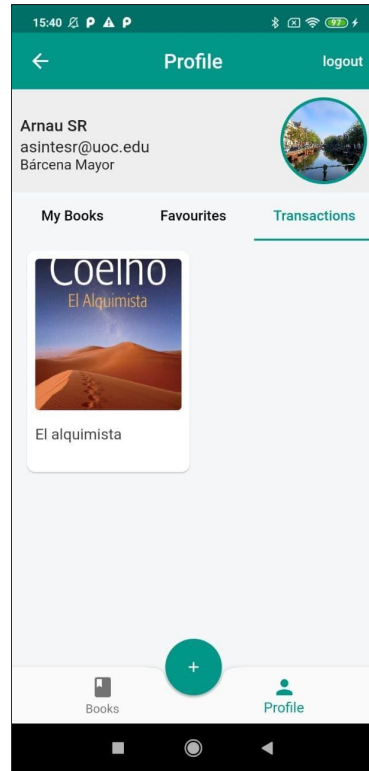
Pantalla de edición



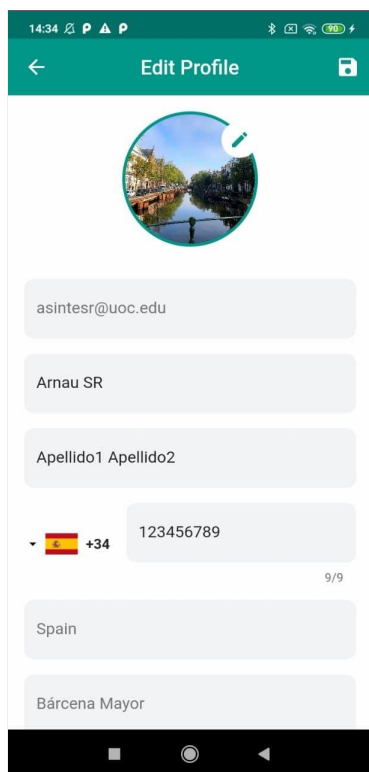
Pantalla de perfil



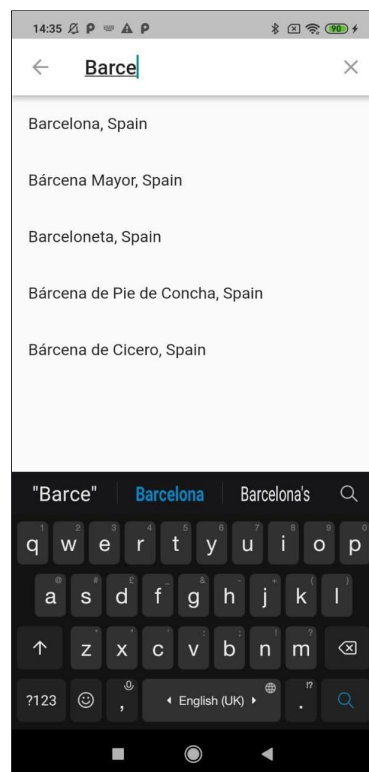
Libros favoritos



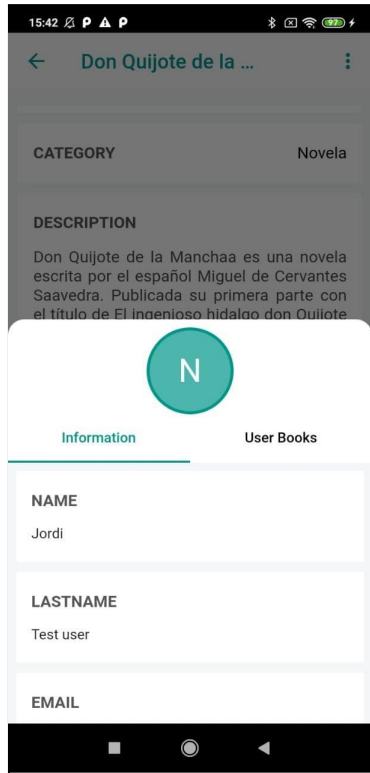
Libros intercambiados



Pantalla de edición del perfil



Sugerencias de ubicación



Información de usuario

10. Evaluación

10.1 Definición de Evaluación

Para evaluar que el producto final cumple las necesidades detectadas y definidas en los primeros apartados de este documento, se ha creado un plan de evaluación basado en los casos de uso definidos y analizados en el apartado 5. *Definición de casos de uso.*

La siguiente tabla muestra un listado de los distintos casos de uso a evaluar una vez finalizada la fase de implementación. Para ello tanto desarrolladores como posibles usuarios, realizarán una validación funcional de los casos de uso siguiendo el flujo definido para cada uno de ellos en el apartado 5.2 *Listado de casos de uso.*

ID	Funcionalidad a validar	Caso de uso
TF-0001	Registrarse en la aplicación.	UC-0001
TF-0002	Iniciar sesión en la aplicación.	UC-0002
TF-0003	Recuperar contraseña de la aplicación.	UC-0003
TF-0004	Cerrar sesión.	UC-0004
TF-0005	Ver listado de libros publicados.	UC-0005
TF-0006	Filtrar listado de libros publicados.	UC-0006
TF-0007	Ver detalle de libro publicado.	UC-0007
TF-0008	Añadir un libro publicado a favoritos.	UC-0008
TF-0009	Descartar una publicación para futuras búsquedas.	UC-0009
TF-0010	Publicar un libro.	UC-0010
TF-0011	Contactar con un usuario.	UC-0011
TF-0012	Ver el listado de chats.	UC-0012
TF-0013	Ver los mensajes de un chat.	UC-0013
TF-0014	Enviar un mensaje a un usuario.	UC-0014
TF-0015	Acceder al perfil del usuario.	UC-0015
TF-0016	Editar el perfil del usuario.	UC-0016
TF-0017	Ver las publicaciones creadas por un usuario.	UC-0017

TF-0018	Editar una publicación creada por un usuario.	UC-0018
TF-0019	Eliminar una publicación creada por un usuario.	UC-0019
TF-0020	Marcar como reservado un libro publicado por el usuario.	UC-0020
TF-0021	Ver el listado de publicaciones marcadas como favoritas.	UC-0021
TF-0022	Eliminar una publicación de favoritos.	UC-0022

Figura 41: Tabla de test funcionales

10.2 Resultados de Evaluación

En la siguiente tabla se puede observar los resultados de cada uno de los test funcionales anteriormente descrito. Estos test han estado se han realizado en distintos dispositivos móviles, de diferentes fabricantes y características técnicas, pero todos ellos bajo el sistema operativo Android.

ID	Funcionalidad a validar	Resultado
TF-0001	Registrarse en la aplicación.	Ok
TF-0002	Iniciar sesión en la aplicación.	Ok
TF-0003	Recuperar contraseña de la aplicación.	Ok
TF-0004	Cerrar sesión.	Ok
TF-0005	Ver listado de libros publicados.	Ok
TF-0006	Filtrar listado de libros publicados.	Ok
TF-0007	Ver detalle de libro publicado.	Ok
TF-0008	Añadir un libro publicado a favoritos.	Ok
TF-0009	Descartar una publicación para futuras búsquedas.	Ok
TF-0010	Publicar un libro.	Ok
TF-0011	Contactar con un usuario	Ok
TF-0012	Ver el listado de chats.	No Implementado
TF-0013	Ver los mensajes de un chat.	No Implementado

TF-0014	Enviar un mensaje a un usuario.	No Implementado
TF-0015	Acceder al perfil del usuario.	Ok
TF-0016	Editar el perfil del usuario.	Ok
TF-0017	Ver las publicaciones creadas por un usuario.	Ok
TF-0018	Editar una publicación creada por un usuario.	Ok
TF-0019	Eliminar una publicación creada por un usuario.	Ok
TF-0020	Marcar como reservado un libro publicado por el usuario.	Ok
TF-0021	Ver el listado de publicaciones marcadas como favoritas.	Ok
TF-0022	Eliminar una publicación de favoritos.	Ok

Figura 42: Resultado de los test funcionales

11. Conclusiones

Durante la implementación del proyecto me he dado cuenta de que para poder desarrollar un nuevo proyecto de forma estable es necesaria tener una buena base sobre la que empezar. Esta base nos la ofrece un buen análisis de requisitos, que en primer lugar mejora la capacidad de planificación del proyecto, disminuyendo los gastos y posibles atrasos en el proyecto. Todo esto implica una mejora de la calidad del producto resultante.

Una buena base, juntamente con el uso de herramientas y entornos de trabajos innovadores o relativamente modernos, nos aporta una idea de cómo se trabaja actualmente, a la vez que facilitan y acortan los procesos de desarrollo y despliegue de aplicaciones.

A pesar de tener buenas herramientas, entornos de trabajos, grandes análisis de requisitos o desarrollar software de calidad, lo que marca la satisfacción de un cliente y del equipo de desarrollo, respecto a un nuevo proyecto es una buena planificación, si esta se sigue de manera correcta, si no hay desviaciones o inconvenientes o bien si el producto final es lo pactado. Es importante que al encontrar alguna tarea bloqueante se dedique tiempo a resolverla, pero no tenemos que olvidar el objetivo global de la tarea, puesto que podríamos perder la visión global del proyecto. También es muy relevante en estos casos, saber priorizar entre tareas u objetivos del proyecto.

Como hemos descrito en el apartado 7. *Revisión de la planificación*, no ha sido posible implementar todos los objetivos analizados en el documento. El principal motivo ha sido las dificultades encontradas en la integración del BaaS Firebase y la integración de la aplicación ciertas APIs de Google.

Para llevar a cabo el desarrollo del proyecto se ha seguido la planificación inicialmente descrita en el apartado 1.4 *Planificación del Trabajo* sin apenas desviaciones, únicamente se han sustituido el tipo de test que a aplicar a nuestro sistema. Se han realizado una batería de test funcionales en vez de test end-to-end automatizados como se había planificado inicialmente, debido a las limitaciones de tiempo y los atrasos acumulados durante la implementación de la aplicación.

Durante la planificación se decidió apostar por una metodología en cascada, tras finalizar el desarrollo de la aplicación, puedo afirmar que ha sido la adecuada a las características de este proyecto, ya que tenía unas fases muy marcadas, junto a tratarse de una propuesta de proyecto personal por lo que no existe el rol de cliente que pueda introducir cambios de funcionalidades durante la implementación, disminuyendo así el riesgo a retrasos por cambios.

11.1 Mejoras de la aplicación

Las mejoras más importantes o prioritarias a aplicar en la aplicación son los requisitos funcionales identificados pero no implementados, descritos en el apartado *7.2 Limitación de funcionalidades*. Estos requisitos funcionales son:

- **SR-FR-0012:** La aplicación debe permitir la comunicación entre usuarios con un chat propio de la aplicación.
- **SR-FR-0014:** Un usuario puede bloquear a otros usuarios para que estos no vean un anuncio en sus búsquedas.

Además de la implementación de los requisitos pendientes, existen dos tipos de mejoras a aplicar, con el fin de mejorar la calidad del producto y su usabilidad:

- Mejoras de rendimiento.
- Mejoras de usabilidad.

En cuanto a las mejoras de rendimiento, la principal sería la implementación de un servicio de búsquedas e indexación intermedio, entre la aplicación y el proveedor de base de datos Firebase, con el fin de mejorar las limitaciones encontradas durante la implementación y mejorar la velocidad de ejecución de consultas y filtrado de datos en la app para grandes cantidades de resultados.

Se han detectado algunos puntos débiles en la usabilidad de la aplicación, por ello se han planteado las siguientes modificaciones con el fin de mejorar experiencia de usuario:

- Permitir subir múltiples imágenes al crear o editar el anuncio de un libro.
- Internacionalizar la aplicación, ya que actualmente solo está disponible en inglés.
- Integrar otros métodos de autenticación, actualmente únicamente se permite realizar un registro mediante correo electrónico, sin embargo, el proveedor de autenticación, Firebase Auth, permite otros métodos como con cuentas de Google, Apple o Facebook entre otros.
- Implementación de temas estéticos alternativos con el fin de que usuarios con problemas de visión puedan tener alternativas para poder seguir usando la aplicación sin dificultades.

13. Glosario

Término	Definición
Stack tecnológico	Conjunto de herramientas o tecnologías que conforman una solución de software.
Backend	Capa de acceso a los datos y lógica, comúnmente llamada como servidor.
Firebase	Plataforma en la nube que ofrece un conjunto de herramientas para el desarrollo web y móvil.
Framework	Entorno de trabajo estandarizado.
React Native	Framework de código abierto para el desarrollo de interfaces de usuario Android e iOS.
Flutter	Framework de código abierto para el desarrollo de aplicaciones multiplataforma capaces de correr en Android, iOS, Linux, Mac, Windows y web.
Android	Sistema operativo móvil propiedad de Google.
iOS	Sistema operativo móvil propiedad de Apple.
BaaS	Backend como Servicio.
Apk	Formato del archivo resultante de la compilación de una aplicación Android.
BD	Base de Datos
Stakeholder	Persona de interés para el proyecto.
App State	Información cargada en memoria de la aplicación.
Web Service	Servicio alojado en la red usado para intercambiar datos entre aplicaciones.
API	Interfaz de programación de aplicaciones es un conjunto de rutina, protocolos y herramientas.
REST	Transferencia de Estado Representación es un tipo de arquitectura para diseñar y crear software.
HTTP	Protocolo de Transferencia de Hipertexto para el intercambio de fichero si tribuidos.
Dart	Lenguaje de programación orientado al cliente desarrollado por Google.
Widget	Objeto propio del framework Flutter para producir una representación visual de los datos.
Provider	Patrón de diseño de software desarrollado por Microsoft.
RxDart	Librería para la implementación de la programación reactiva en Dart.

14. Bibliografía

Conceptualització: Jordi Flamarich Zampalo, FUOC - PID_00245391.

Arquitectura i wireframes: Jordi Flamarich Zampalo, FUOC - PID_00245392.

Disseny i prototipatge: Jordi Flamarich Zampalo, FUOC - PID_00245393.

Diseño centrado en el usuario: Muriel Garreta Domingo y Enric Mor Pera, CC-BY-SA - PID_00176058.

The Digital Project Manager: Un Blog De Gestión De Proyectos Digitales: [9 Ejemplos de Metodología de un Proyecto, Simplificados \(thedigitalprojectmanager.com\)](#), octubre del 2021.

Wikipedia: [Waterfall model - Wikipedia](#), octubre del 2021.

Smartsheet: [A Complete Guide to the Waterfall Project Method | Smartsheet](#), octubre del 2021.

Simplemente pregunta: Integración de la accesibilidad en el diseño: [Portada | Simplemente pregunta: Integración de la accesibilidad en el proceso de diseño \(uiaccess.com\)](#), octubre del 2021.

Simple app state management: [Simple app state management | Flutter](#), Noviembre 2021.

How to Use the Provider Pattern in Flutter: [How to Use the Provider Pattern in Flutter \(freecodecamp.org\)](#), Noviembre 2021.

Flutter architectural overview: [Flutter architectural overview | Flutter](#), Noviembre 2021.

15. Anexos

Además de la memoria del proyecto se han incluido los siguientes documentos y artefactos:

- Código fuente de la aplicación desarrollada con Flutter.
- Archivo *apk* de la aplicación una vez compilada.
- Manual de instalación.
- Manual de usuario.
- Presentación del proyecto.