

Grau en Enginyeria Informàtica de Gestió i Sistemes d'Informació

Desenvolupament d'una aplicació mòbil de lectures conjuntes

Memòria

Maria Colom Gallardo

Tutor: Dr. David Ródenas Picó

Curs 2022-2023

Agraïments

En primer lloc, agraeixo al meu tutor, David Ródenas, per tot el suport i surgències que han col·laborat a la creació d'aquest projecte.

També vull agrair als meus pares i a la meva parella per ser sempre els primers en veure cada nou pas del desenvolupament.

Abstract

The objective of this project is to design and develop a mobile application to allow users to share opinions and comments that arise when reading a book. It has followed an agile development process based on iterations. The application has been developed with Flutter and it has used Firebase backend services to cover authentication, data persistence and image storage features.

Resum

L'objectiu d'aquest projecte és dissenyar i desenvolupar una aplicació mòbil que permeti a l'usuari compartir opinions i comentaris, amb altres usuaris, que van sorgint durant la lectura d'un llibre. S'ha seguit un procés de desenvolupament àgil basat en iteracions. L'aplicació s'ha desenvolupat amb Flutter i s'han utilitzat els serveis de backend de Firebase per cobrir les funcions d'autenticació, persistència de dades i emmagatzematge d'imatges.

Resumen

El objetivo de este proyecto es diseñar y desarrollar una aplicación móvil que permita al usuario compartir opiniones y comentarios, con otros usuarios, que van surgiendo durante la lectura de un libro. Se ha seguido un proceso de desarrollo ágil basado en iteraciones. La aplicación se ha desarrollado con Flutter y se han utilizado los servicios de backend de Firebase para cubrir las funciones de autenticación, persistencia de datos y almacenamiento de imágenes.

Índex

Índex de figures	IV
Índex de taules	VIII
Glossari de termes	IX
1. Introducció	1
2. Marc teòric i anàlisi dels referents	3
2.1 Context.....	3
2.2 Antecedents	3
2.2.1 Goodreads.....	3
2.2.3 Instagram.....	5
2.2.3 The StoryGraph	6
2.2.4 Síntesi dels antecedents	8
2.3 Necessitats d'informació.....	9
2.4 Estudi de tecnologies i serveis	10
2.4.1 Tipus de desenvolupament mòbil.....	10
2.4.2 Software de desenvolupament.....	11
2.4.3 Serveis de backend	12
2.4.4 Estudi de serveis de Firebase	13
Objectius i abast	16
3.1 Objectius	16
3.1.1 Objectius de client.....	16
3.1.2 Objectius de producte.....	16
3.2 Abast	17
3. Metodologia	18
4.1 Iteracions.....	19
4.1.1 Estudi de tecnologies.....	20

4.1.2 Configuració i model de navegació.....	20
4.1.3 Implementació de llibres	20
4.1.4 Implementació de sales de lectura.....	20
4.1.5 Navegació entre pàgines.....	21
4.1.6 Implementació d'usuaris	21
4.1.7 Implementació del xat	21
4.1.8 Incorporació d'imatges.....	21
4.1.9 Implementació de rols i sol·licituds	21
4.1.10 Elements addicionals del xat	22
4.1.11 Amics i notificacions	22
4.1.12 Denuncies i missatges a la desenvolupadora.....	22
4. Desenvolupament	23
5.1 Definició de requeriments funcionals i tecnològics.....	23
5.1.2 Requeriments tecnològics	25
5.2 Tria de tecnologies	25
5.3 Arquitectura	27
5.3.1 Arquitectura de serveis backend.....	27
5.3.2 Arquitectura del projecte	28
5.3 Casos d'ús	30
5.3.1 Actors	30
5.3.2 Diagrames dels casos d'ús.....	31
5.3.3 Descripció dels casos d'ús.....	32
6. Implementació	40
6.1 Preparació entorn de desenvolupament	40
6.2 Implementació de Firebase al projecte	41
6.3 Navegació	46
6.4 Llibres	50

6.5 Sales de Lectura	53
6.6 Autenticació	61
6.7 Perfil d'usuari	65
6.8 Sales de Xat	66
7. Anàlisi de les iteracions	69
7.1 Estudi de tecnologies	69
7.2 Configuració i model de navegació	69
7.3 Implementació de llibres.....	70
7.4 Implementació de sales de lectura	70
7.5 Navegació entre pàgines	71
7.6 Implementació d'usuaris.....	72
7.7 Implementació del xat.....	72
7.8 Incorporació d'imatges i altres millores	73
8. Conclusions	75
9. Possibles ampliacions.....	76
10. Bibliografia.....	78

Índex de figures

Figura 1. Sala de lectura conjunta web Goodreads. Font: Captura de pantalla.....	4
Figura 2. Sala de lectura conjunta app Goodreads. Font: Captura de pantalla.....	4
Figura 3. Publicació d'organització de lectura conjunta. Font: Captura de pantalla.....	6
Figura 4. Bloqueig de comentaris per usuaris enrederits. Font [7].....	7
Figura 5. Lectures conjuntes a The StoryGraph. Font: [8].....	7
Figura 6. Fases de l'Agile. Font: [27].....	19
Figura 7. Diagrama general dels processos de l'aplicació. Font: Elaboració pròpia	27
Figura 8. Comparació entre arquitectura feature-first i layer-first. Font: [32]	28
Figura 9. Captura de l'arquitectura final de l'aplicació. Font: Elaboració pròpia	29
Figura 10. Esquema de capes i comunicació de l'aplicació. Font: Elaboració pròpia.....	29
Figura 11. Tipus d'usuaris. Font: Elaboració pròpia	30
Figura 12. Casos d'ús d'autenticació i gestió de perfil. Font: Elaboració pròpia.....	31
Figura 13. Casos d'ús de les sales de lectura. Font: Elaboració pròpia	32
Figura 14. Eina Inspector de Flutter a Android Studio. Font: Elaboració pròpia.....	40
Figura 15. Plugin de Dart a Android Studio. Font: Elaboració pròpia	41
Figura 16. Plugin de Flutter a Android Studio. Font: Elaboració pròpia	41
Figura 17. Codi del main de l'aplicació amb Firebase integrat. Font: Elaboració pròpia ...	42
Figura 18. Web de creació de projecte de Firebase. Font: Elaboració pròpia	42
Figura 19. Procés d'addició de Firebase a projecte Android. Font: Elaboració pròpia	42
Figura 20. Botiga de productes de Firebase. Font: Elaboració pròpia	43

Figura 21. Consola de Realtime Database. Font: Elaboració pròpia.....	43
Figura 22. Consola de Firebase Authentication. Font: Elaboració pròpia.....	44
Figura 23. Consola de Firebase Cloud Storage. Font: Elaboració pròpia	45
Figura 24. Contingut d'una carpeta de Firebase Cloud Storage. Font: Elaboració pròpia ..	45
Figura 25. Consola de Firebase Cloud Firestore. Font: Elaboració pròpia	46
Figura 26. Esquema de navegació de l'aplicació mòbil. Font: Elaboració pròpia	46
Figura 27. Barra inferior de navegació de l'aplicació. Font: Elaboració pròpia.....	47
Figura 28. Estructura d'objectes del navegador de l'aplicació. Font: Elaboració pròpia.....	48
Figura 29. Botó de redirecció a la creació de sales de lectura. Font: Elaboració pròpia.....	48
Figura 30. Stepper 1. Font: Elaboració pròpia.....	49
Figura 31. Stepper 2. Font: Elaboració pròpia.....	49
Figura 32. Disseny del perfil de llibres. Font: Elaboració pròpia.....	50
Figura 33. Implementació final del perfil de llibres. Font: Elaboració pròpia	50
Figura 34. Implementació inicial del perfil de llibres. Font: Elaboració pròpia	50
Figura 35. Formulari de llibre amb errors destacats. Font: Elaboració pròpia.....	51
Figura 36. Formulari ampliat de creació de llibre. Font: Elaboració pròpia	51
Figura 37. Widget d'autocompletar gèneres. Font: Elaboració pròpia.....	52
Figura 38. Codi d'escriptura de sala de lectura a Cloud Storage. Font: Elaboració pròpia	52
Figura 39. Objecte llibre a la base de dades. Font: Elaboració pròpia	52
Figura 40. Implementació del formulari de les sales de lectura 1. Font: Elaboració pròpia	53
Figura 41. Implementació del formulari de les sales de lectura 2. Font: Elaboració pròpia	53

Figura 42. Formulari de creació d'esdeveniment. Font: Elaboració pròpia.....	54
Figura 43. Esdeveniment creat a la sala de lectura. Font: Elaboració pròpia.....	54
Figura 44. Tria d'hora. Font: Elaboració pròpia.....	54
Figura 45. Codi d'escriptura de sala a Realtime Database. Font: Elaboració pròpia	54
Figura 46. Codi de conversió d'objecte a JSON. Font: Elaboració pròpia	55
Figura 47. Codi de creació de sala de xat a Firebase Firestore. Font: Elaboració pròpia ...	55
Figura 48. Codi d'addició membre a sala a Realtime Database. Font: Elaboració pròpia ..	55
Figura 49. Objecte sala de lectura a la base de dades Font: Elaboració pròpia.....	56
Figura 50. Disseny del perfil de les sales de lectura 3. Font: Elaboració pròpia.....	57
Figura 51. Disseny del perfil de les sales de lectura 2. Font: Elaboració pròpia.....	57
Figura 52. Disseny del perfil de les sales de lectura 1. Font: Elaboració pròpia.....	57
Figura 53. Implementació inicial del perfil de les sales 1. Font: Elaboració pròpia	58
Figura 54. Implementació inicial del perfil de les sales 2. Font: Elaboració pròpia	58
Figura 55. Implementació inicial del perfil de les sales 2. Font: Elaboració pròpia	58
Figura 56. Implementació final del perfil de les sales 1. Font: Elaboració pròpia.....	58
Figura 57. Implementació final del perfil de les sales 2. Font: Elaboració pròpia.....	58
Figura 58. Implementació final del perfil de les sales 3. Font: Elaboració pròpia.....	58
Figura 59. Visualització d'accions usuari sense iniciar sessió. Font: Elaboració pròpia	59
Figura 60. Aaccions usuari identificat llistat de sales. Font: Elaboració pròpia.....	59
Figura 61. Accions a sala usuari identificat membre d'una sala. Font: Elaboració pròpia .	59
Figura 62. Accions a una sala usuari identificat propietari. Font: Elaboració pròpia.....	59

Figura 63. Formulari d'edició de sala de lectura. Font: Elaboració pròpia	60
Figura 64 Alerta d'esborrada de sala de lectura. Font: Elaboració pròpia	60
Figura 65. Notificació d'avis d'impossibilitat d'esborrada. Font: Elaboració pròpia.....	60
Figura 66. Efecte splash de les portades. Font: Elaboració pròpia.....	61
Figura 67. Sales de lectura públiques Font: Elaboració pròpia	61
Figura 68. Pantalla per iniciar sessió a l'aplicació. Font: Elaboració pròpia.....	62
Figura 69. Pantalla per crear un compte a l'aplicació. Font: Elaboració pròpia	62
Figura 70. Correu per validar la seva identitat. Font: Elaboració pròpia.	62
Figura 71. Pantalla de validació de correu electrònic. Font: Elaboració pròpia.....	63
Figura 72. Codi d'alta a Firebase Authentication. Font: Elaboració pròpia.....	63
Figura 73. Codi d'inici de sessió a Firebase Authentication. Font: Elaboració pròpia	64
Figura 74. Formulari d'edició del perfil d'usuari. Font: Elaboració pròpia	65
Figura 75. Perfil d'usuari amb perfil editat. Font: Elaboració pròpia	65
Figura 76. Perfil d'usuari amb compte nou. Font: Elaboració pròpia	65
Figura 77. Llistat de sales de xat. Font: Elaboració pròpia	66
Figura 78. Sala de xat amb teclat obert. Font: Elaboració pròpia	67
Figura 79. Sala de xat. Font: Elaboració pròpia	67
Figura 80. Codi d'escriptura de missatge a Firebase Firestore. Font: Elaboració pròpia....	68
Figura 81. Consola de Firestore mostrant una sala de xat. Font: Elaboració pròpia.....	68

Índex de taules

Taula 1. Comparació de frameworks de desenvolupament. Font: Elaboració pròpia	12
Taula 2. Comparació de serveis de backend. Font: [9]-[11]	13
Taula 3. Avantatges i inconvenients de Goodreads. Font: Elaboració pròpia	5
Taula 4 Avantatges i inconvenients d'Instagram. Font: Elaboració pròpia	6
Taula 5. Avantatges i inconvenients de The StoryGraph. Font: [22]-[25]	8
Taula 6. Comparativa entre els antecedents. Font: Elaboració pròpia	8
Taula 7. Avantatges i inconvenients de l'Agile. Font: [29]	19

Glossari de termes

Machine Learning	El Machine Learning o aprenentatge automàtic és una branca de la intel·ligència artificial que es centra en l'ús de dades i algorismes per imitar la manera com els humans aprenen, millorant gradualment la seva precisió.
Framework	Un framework és una plataforma que proporciona una base per desenvolupar aplicacions de software.
API	Application Programming Interface. Software intermediari que permet a dues aplicacions comunicar-se entre elles. És una manera accessible d'extreure i compartir dades dins i a través de les organitzacions.
SDK	Software development kit. Conjunt d'eines, llibreries i documentació que faciliten la creació d'aplicacions per a una plataforma o sistema determinat. Proporciona les eines necessàries per a crear software de manera eficient i efectiva.
MVP	Minimum Value Product o producte mínim viable és un prototip funcional d'un producte o servei que permet obtenir retroalimentació dels usuaris de manera ràpida i eficient. Aquesta estratègia ajuda a validar la viabilitat del producte i a reduir costos de desenvolupament.
CLI	Command-line Interface. Tipus d'interfície d'usuari que permet interactuar amb un programa o sistema a través d'ordres escrites en un terminal o consola.

1. Introducció

“Llegir pot ser una experiència molt solitària i hi ha llocs web que ofereixen una connexió social que resulta de molta ajuda.” Chasity Moreno, departament de Serveis de Lectors de la Biblioteca Pública de Nova York ^[1]

L'objectiu d'aquest projecte és dissenyar i desenvolupar una aplicació mòbil que permeti a l'usuari compartir opinions i comentaris, amb altres usuaris, que van sorgint durant la lectura d'un llibre. El xat, essent el mètode més utilitzat avui en dia de comunicació digital instantània, és la base sobre la que es dissenyen les sales de lectura conjunta de l'aplicació.

L'objectiu principal del producte final és cobrir una necessitat existent, ja que a nivell de comunitat lectora hispanoparlant l'organització de lectures conjuntes virtuals es fa a través d'eines no orientades a cobrir l'activitat, com per exemple Instagram. També cal destacar com a propòsit del producte socialitzar i dinamitzar la lectura, podent acostar-la a un nou públic.

L'aplicació consta de tres elements principals: usuaris, sales de lectura i xat. Una sala de lectura està formada per un llibre, un itinerari i un administrador. Els usuaris poden consultar les sales de lectura disponibles i afegir-se a qualsevol. Tots els membres de l'aplicació poden crear les seves pròpies sales de lectura i passen automàticament a ser el primer membre d'aquestes. Quan un usuari entra a una sala de lectura se l'afegeix a la sala de xat grupal de la sala. L'usuari pot veure en una pestanya totes les sales de xat en les que participa. Finalment, els usuaris tenen el seu propi perfil i també poden consultar el perfil del propietari d'una sala de lectura.

Les eines utilitzades per la creació del projecte han de poder-se adaptar al màxim de dispositius possibles sense tenir un gran nivell de complexitat degut als recursos limitats dels que es proporciona pel desenvolupament del producte. Així doncs, el requisit essencial que ha de complir el SDK final és sigui multiplataforma, és a dir, ha d'operar en diverses plataformes informàtiques, en aquest cas Android i iOS. El *framework* triat ha estat Flutter, que treballa amb el llenguatge de programació Dart.

Pels serveis de *backend* es requereix una baixa complexitat d'implementació i la interoperabilitat entre els diferents serveis per poder reduir el temps de producció. És

essencial una base de dades i un medi d'autenticació d'usuaris. La plataforma BaaS encarregada de proveir els serveis que requereix és Google Firebase. D'aquesta s'utilitza la base de dades Firebase Realtime Database per cobrir la necessitat d'una base de dades.

2. Marc teòric i anàlisi dels referents

2.1 Context

“Com a criatures socials que som, quan trobem alguna cosa que ens emociona o enfada, volem compartir-ho. Volem que els altres sàpiguen com ens ha fet sentir alguna cosa: i volem saber si algú més s’ha sentit igual.” [2]

La lectura és una activitat físicament solitària, però el lector és un ésser social. Per això mateix, tant articles com blogs i altres formats similars tenen un botó de compartir i la opció de publicar un comentari, perquè és natural voler fer saber i discutir el que un pensa.

S’entén com a lectura conjunta una activitat en la que participen dues o més persones que estan llegint un mateix llibre seguint el mateix ritme. El ritme ve marcat per capítols per període de temps i l’objectiu és poder compartir opinions i fer de la lectura una activitat social^[3].

Tradicionalment aquesta funció ha estat coberta per grups de lectura de biblioteques municipals o escoles que, després de fer la tria d’un llibre, els participants llegeixen i, en una data concreta, fan una posta en comú d’opinions de manera presencial.

2.2 Antecedents

Les lectures conjuntes han anat evolucionant i, gràcies al sorgiment de noves tecnologies, s’han adaptat a tota mena de plataformes com blogs, fòrums i xarxes socials. Avui en dia, la font predominant de lectures conjuntes són les xarxes socials, essent les principals Instagram i Goodreads. Tot i que no hi ha cap xarxa social especialment dedicada a les lectures conjuntes, hi ha una plataforma que sí ofereix aquest servei, The StoryGraph. De cada una de les diferents eines s’estudiaran els punts forts i mancances per poder determinar amb major precisió els objectius i abast del projecte.

2.2.1 Goodreads

Goodreads és una pàgina web subsidiària d’Amazon que permet a l’usuari cercar llibres, recomanacions, cites i ressenyes. Té la comunitat lectora més gran del món amb 90 milions de membres enregistrats^[4]. L’usuari es pot donar d’alta per publicar les seves pròpies

resenyes i catalogar llibres en forma de llistes. D'entre les funcions secundàries s'inclouen el suggeriment de llibres, creació d'enquestes, blogs i fòrums de debat.

Les lectures conjuntes de Goodreads s'organitzen en sales de tòpics concrets contingudes dintre dels grups. Les sales segueixen un estil de comentaris seqüencials i es poden endreçar per data. Encara que no hi ha característiques concretes per a lectures conjuntes, sí que hi ha certes funcions inspirades en els clubs de lectura tradicionals, com citar futures reunions o crear una llista amb els llibres que s'ha llegit el club de lectura. Es poden cercar grups amb sales dedicades a lectures conjuntes a través d'etiquetes creades pels propis usuaris (ex. book-club).

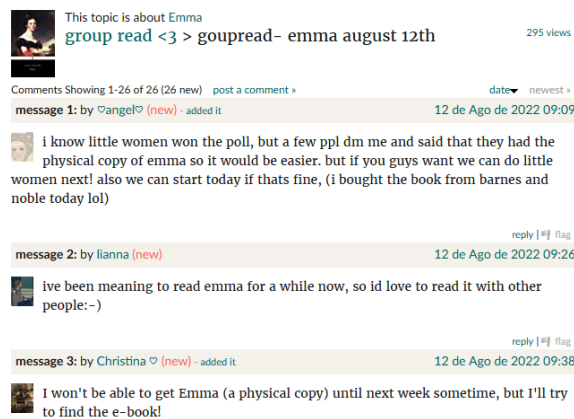


Figura 1. Sala de lectura conjunta web Goodreads. Font: Captura de pantalla

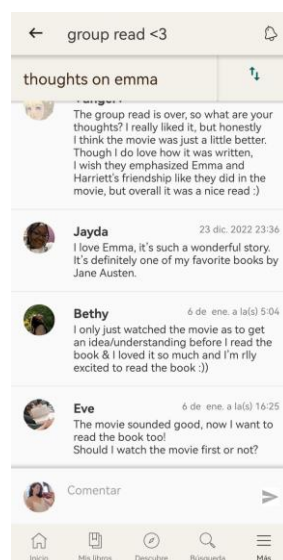


Figura 2. Sala de lectura conjunta app Goodreads. Font: Captura de pantalla

Avantatges	Inconvenients
Aplicació referent de la comunitat lectora	Aplicació mòbil amb funcionalitats limitades.
Gran base de dades de llibres	Sales de lectura no segueixen un estil de xat
Diversitat de funcionalitats (enquestes, llistes de lectura...)	Aplicació no orientada a la realització per fer lectures conjuntes

Taula 1. Avantatges i inconvenients de Goodreads per a la realització de lectures conjuntes. Font: Elaboració pròpia

2.2.3 Instagram

Instagram és una xarxa social propietat de Meta Platforms que permet a l'usuari compartir vídeos i imatges^[5]. La comunitat lectora de d'Instagram s'anomena Bookstagram, es pot identificar a través d'etiquetes en imatges o als perfils dels usuaris. Els membre de la comunitat de Bookstagram originalment només publicaven imatges estètiques de llibres amb petites ressenyes a la descripció, però pas del temps s'han començat a organitzar lectures conjuntes.

Les lectures conjuntes d'Instagram les solen organitzar usuaris amb certa rellevància a la comunitat. El procés sol ser el següent: primer s'anuncia a través d'una publicació o història el llibre triat per la lectura conjunta. Seguidament s'informa del seguit de dates en la que es comentarà el progrés del llibre fins a un capítol concret. Finalment, a cada data, s'habitua fer un vídeo en directe a la mateixa plataforma, on les organitzadores comenten entre elles la lectura i llegeixen esporàdicament comentaris de la resta d'usuaris que hi participen. Aquest mètode comporta una comunicació limitada entre tots els participants.



Figura 3. Publicació d'organització de lectura conjunta. Font: Captura de pantalla

Avantatges	Inconvenients
Milions d'usuaris actius dia a dia	Aplicació no orientada a la lectura
Gran comunitat lectora establerta	Sales de xat molt simples
Funcionalitats útils per les lectures conjuntes com vídeos directes o històries	Es complica fer un seguiment de l'estat de la lectura conjunta

Taula 2 Avantatges i inconvenients d'Instagram per a la realització de lectures conjuntes. Font: Elaboració pròpia

2.2.3 The StoryGraph

The StoryGraph és una web fundada l'any 2019 per Nadia Odunayo que va començar com un projecte per fer seguiment de lectures a través de llistes^[6]. Té un gran sistema de recomanació de llibres gràcies a una intel·ligència artificial de *machine learning* i també mostra suggeriments en base a l'estat d'ànim de l'usuari.

Recentment s'han incorporat una gran varietat de funcions per les lectures conjuntes, entre les que destaquen la de poder afegir reaccions a parts específiques d'un llibre i que els

participants desbloquegen quan arriben al mateix punt. Gràcies a la mateixa intel·ligència artificial de *machine learning* que mostra suggeriments de llibres, es pot cercar usuaris amb gustos similars per iniciar lectures conjuntes.

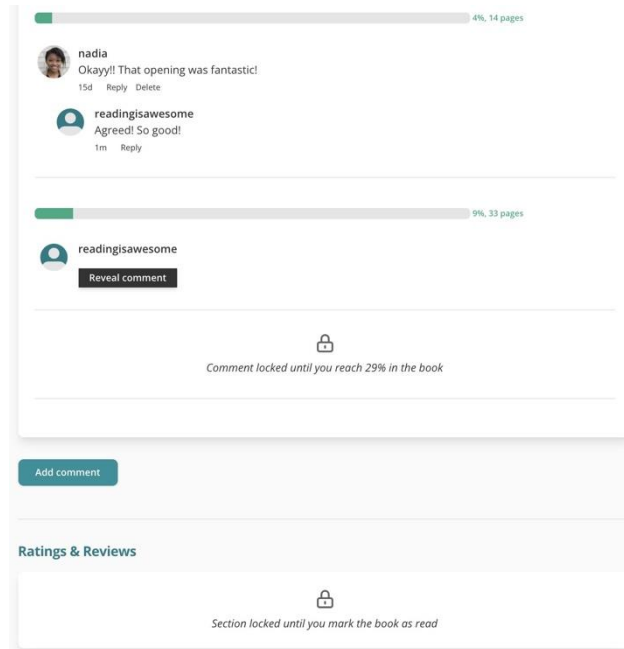


Figura 4. Bloqueig de comentaris per usuaris enrederits. Font [7]

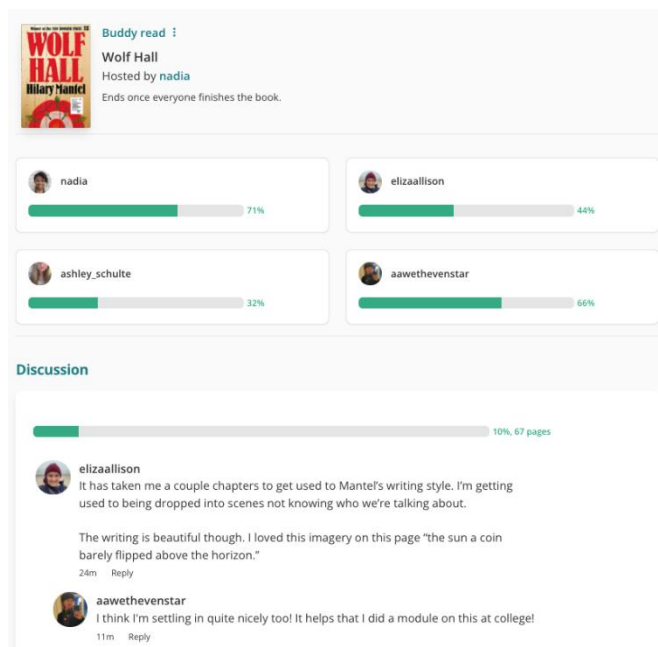


Figura 5. Lectures conjuntes a The StoryGraph. Font: [8]

Avantatges	Inconvenients
Aplicació molt nova amb actualitzacions constants.	L'aplicació es torna més complexa a mesura que incorporen actualitzacions.
Permet lectures conjuntes amb ritme individual.	L'aplicació no esta en espanyol.
Permet fer seguiment de les lectures a traves de llistes i estadístiques.	No hi ha un model de xat, sinó llistat de comentaris

Taula 3. Avantatges i inconvenients de The StoryGraph per a la realització de lectures conjuntes. Font: [9]-[12]

2.2.4 Síntesi dels antecedents

Característica	Goodreads	Instagram	The StoryGraph
Aplicació dedicada a la lectura	X		X
Lectures conjuntes implementades			X
Format de comunicació	Fil	Xat	Comentaris
Límit usuaris per lectura conjunta	Il·limitat	Il·limitat	5

Taula 4. Comparativa de característiques entre els antecedents. Font: Elaboració pròpia

2.3 Necessitats d'informació

La informació més rellevant que cal pel desenvolupament del projecte és estudiar els punts febles de cada recurs ja existent per la realització de lectures conjuntes. D'igual manera, també es necessari analitzar els punts forts de cada plataforma, per què els usuaris utilitzen una eina concreta per realitzar les lectures conjuntes i no una altra? A partir de les vulnerabilitats i fortaleces extretes es poden modelar amb més precisió els objectius i límits del producte.

Tota aquesta informació s'extreu de dues fonts: comentaris i opinions publicades online sobre dites plataformes i preguntes directes als usuaris tant que organitzen com que participen en les lectures conjuntes. En el moment que es disposa del MVP la informació s'extreu del *feedback* dels usuaris, donant peu a fer iteracions sobre el producte per millorar-lo i adaptar-lo a les necessitats del públic objectiu.

Per la banda tecnològica és necessari consultar la sintaxi i documentació del *framework* amb el que es desenvolupa l'aplicació. També cal comparar les diferents opcions que existeixen per donar suport als serveis de *backend*, com la persistència de dades o l'autenticació d'usuaris. De la tria final és important documentar-se del funcionament de cada servei i de la seva implementació en el *framework* de desenvolupament, pel que es primordial que l'elecció final disposi d'una quantitat considerable d'informació i suport per part del proveïdor.

2.4 Estudi de tecnologies i serveis

2.4.1 Tipus de desenvolupament mòbil

Es pot dir que una de les tries més importants a l'hora de desenvolupar qualsevol projecte és determinar quin objectiu o *target* té i, en el món del *software*, és el mateix. Determinar en quin entorn o sistema operatiu s'executarà el programari és essencial. Per les aplicacions mòbils hi ha tres opcions, desenvolupament natiu, híbrid o multiplataforma^[13].

Desenvolupament natiu

Les aplicacions natives son aquelles que han estat explícitament programades per un sistema operatiu concret^[14]. Un exemple pot ser Android, on les aplicacions son programades en Java o Kotlin. L'avantatge més gran del desenvolupament natiu és que s'aprofiten al màxim les qualitats del *hardware*, pel que es pot veure una millor eficiència en el programari. Per altre banda, el major inconvenient que hi ha és que es tracta d'una solució cara en quant temps i recursos. Cal tenir i fer tot per duplicat degut a que no es pot reciclar codi entre diferents llenguatges.

Desenvolupament multiplataforma

Les aplicacions multiplataforma poden operar en diferents sistemes operatius amb una sola font de codi. Aquestes es creen amb *frameworks* especialitzats com pot ser React Native. En contra del desenvolupament natiu, l'avantatge més evident és la gran reducció de costos degut a la concentració d'esforços en un sol objectiu. D'igual manera, és més fàcil de mantenir perquè no hi ha dues fonts de codi. L'inconvenient més destacat d'aquesta tècnica és que té un rendiment limitat comparat amb aplicacions natives.

Desenvolupament híbrid

El desenvolupament híbrid consisteix en la unió de solucions natives i web. A través d'eines com Ionic es pot inserir codi web en una aplicació nativa. Comparteix avantatges i inconvenients amb el sistema multiplataforma, el cost d'un projecte es veu substancialment abaratit degut a que només cal una font de codi però el rendiment es veu afectat.

2.4.2 Software de desenvolupament

Un cop presa la decisió d'utilitzar un tipus de desenvolupament multiplataforma o híbrid, especificat en l'apartat 5.2, s'han estudiat les diferents eines candidates que compleixen el requisit són Flutter, React Native, Ionic i Xamarin^[15].

Flutter

Flutter és un *framework* de codi obert creat per Google. Utilitza el llenguatge de programació Dart i compila les aplicacions de manera nativa. Presenta una gran quantitat de *widgets* per crear interfícies d'usuari responsives i ràpides. Una gran avantatge que ofereix és el *hot reload*, eina que permet veure els canvis realitzats en temps real durant el desenvolupament.

React Native

React Native és un *framework* de codi obert creat per Facebook que combina les avantatges de JavaScript i React.js. Té una comunitat molt gran pel que hi ha molt de suport i permet reutilitzar una gran quantitat de codi, fet que agilitza el desenvolupament. L'eina utilitza components nadius en lloc de components web, fet que permet una experiència d'usuari més responsiva i semblant a la nativa.

Ionic

Ionic és un *framework* de codi obert que utilitza tecnologies web com HTML, CSS i JavaScript per crear aplicacions híbrides per a iOS, Android i el web. Ionic està construït sobre Stencil.js però també suporta Angular i React, entre d'altres. Degut a la seva baixa complexitat és una bona eina per treure ràpidament un MVP al mercat.

Xamarin

Xamarin és un *framework* multiplataforma propietat de Microsoft que permet als desenvolupadors crear aplicacions multiplataforma mitjançant una única base de codi C#. Ofereix un conjunt ric d'eines i biblioteques per crear aplicacions mòbils d'alt rendiment i visualment atractives. S'integra amb Visual Studio pel que facilita l'ús de les eines existents.

La següent taula mostra una síntesi de les diferents característiques de cada *framework*:

Característica	Flutter	React Native	Ionic	Xamarin
Llenguatge de programació	Dart	Javascript	HTML/CSS/JS	C#
Rendiment	Molt alt	Alt	Moderat	Moderat
Reusabilitat de codi	Mitja	Alta	Molt alta	Molt alta
Renderització	Nativa	Nativa	<i>WebView</i>	Nativa
Complexitat aplicacions	Qualsevol	Qualsevol	Senzilles	Senzilles
Comunitat	En creixement	Molt gran	Gran	Gran
Preu	Codi obert	Codi obert	Codi obert + pagament	Codi obert + pagament

Taula 5. Comparació de *frameworks* de desenvolupament. Font: Elaboració pròpia

2.4.3 Serveis de backend

El *backend* com a servei (BaaS) és un model al núvol que ajuda els desenvolupadors a automatitzar les funcions de *backend*, com per exemple la base de dades, i a aconseguir una gestió de la infraestructura sense complicacions ja que el proveïdor de serveis BaaS és el responsable d'executar i portar el manteniment dels servidors^[16]. Els BaaS solen oferir una gran varietat de serveis i eines com ara l'autenticació d'usuaris o bases de dades en temps real i son fàcilment integrables a la interfície d'una aplicació.

Les major avantatges d'utilitzar serveis de *backend* en contra de servidors propis són el baix cost en projectes petits, l'escalabilitat, la seguretat i la substancial reducció de temps que suposa pel desenvolupador. Així doncs, degut a la limitació de recursos del projecte, es considera la opció més factible.

Un cop presa la decisió d'utilitzar Flutter, a l'apartat 5.2, s'han estudiat les diferents propostes que ofereixen serveis de *backend* que millor s'adapten al *framework*, AWS Amplify, Firebase i Back4App.

Tots ells proveeixen autenticació, base de dades en temps real i notificacions push. D'igual manera, també ofereixen plans gratuïts per petits projectes. La següent taula mostra un resum de les diferents característiques que es tenen en compte al moment de fer la tria final:

Característica	AWS Amplify	Firebase	Back4App
Complexitat de configuració	Mitja	Baixa	Mitja-baixa
Documentació	Extensa i més orientada a desenvolupament web.	Extensa i comprensible. Orientat a desenvolupament mòbil.	Enfoc més senzill i orientada al desenvolupament d'aplicacions.
Escalabilitat	Alta	Alta	Limitada
Mida projecte	Mig-gran	Petit-mig-gran	Petit-mig

Taula 6.Comparació de serveis de backend. Font:[17]-[19]

2.4.4 Estudi de serveis de Firebase

Un cop feta la tria del proveïdor de backend a l'apartat 5.2 cal analitzar els diferents productes que ofereix Firebase que poden cobrir les necessitats del projecte.

Firestore Realtime Database

Firestore Realtime Database és una base de dades NoSQL allotjada al núvol dissenyada per ser escalable i en temps real^[20]. Es pot utilitzar per crear aplicacions que requereixen de sincronització de dades entre diversos clients i plataformes, tal com iOS, Android i web. Es basa en un model de dades jeràrquic on aquestes s'emmagatzemen com a un gran arbre

JSON, a més de proporcionar una seguretat robusta, amb processos d'autenticació i autorització, per protegir dades sensibles.

Realtime Database ofereix APIs i llibreries fàcils d'utilitzar per a Flutter, cosa que simplifica la integració de la base de dades amb aquestes aplicacions. El SDK Firebase Flutter proporciona *widgets* prèviament creats que es poden utilitzar per mostrar dades en temps real.

Firestore Authentication

Firestore Authentication permet afegir autenticació d'usuaris a les aplicacions mòbils de manera senzilla mitjançant diversos mètodes, com ara correu electrònic/contrasenya, número de telèfon i Google, entre d'altres^[21].

Algunes del les avantatges que ofereix són la fàcil integració amb aplicacions i l'autenticació segura amb funcions de seguretat robustes, com ara la transmissió xifrada de dades sensibles i el segur emmagatzemant de credencials d'usuaris. També proveeix de components personalitzables per a la interfície d'autenticació, a banda d'una interfície ja construïda per a Android, iOS, i web per oferir una experiència d'usuari coherent i cohesionada.

Firestore Firestore

Firestore Firestore és una base de dades de documents NoSQL^[22]. Emmagatzema dades en forma de documents organitzats en col·leccions. Cada document consta de parells clau-valor similars als objectes JSON.

Una de les característiques clau de Firestore és la sincronització de dades en temps real. És altament escalable i pot gestionar grans volums de dades i trànsit pel que és la millor opció pel servei de xat de l'aplicació. També és reconegut per proporcionar un sistema més complex per fer cerques a la base de dades.

Firestore Cloud Storage

Firestore Cloud Storage serveix per emmagatzemar objectes al núvol. Permet emmagatzemar i mostrar contingut generat pels usuaris, com ara imatges, vídeos i fitxers d'àudio, amb una gran escalabilitat i disponibilitat^[23]. També garanteix la seguretat a través del el control d'accés i el xifrat. Ja que els usuaris no sempre tenen accés a internet, Cloud Storage atura i

reactiva automàticament les transferències de dades a mesura que l'aplicació perd i recupera la connectivitat, estalviant temps i ample de banda als usuaris.

Firestore Cloud Messaging

Firestore Cloud Messaging permet enviar missatges i notificacions a usuaris a través de diverses plataformes com ara Android, iOS i web^[24]. Dona la possibilitat d'enviar missatges multimèdia, inclosos vídeos, imatges, i enllaços, i missatges personalitzats a usuaris concrets o a grups d'usuaris, en funció del comportament d'aquests o d'altres criteris.

És fàcil d'integrar amb Firestore i altres serveis de Google, com ara Cloud Functions i Realtime Database. No cal codificar el sistema de notificacions degut a que està totalment integrat amb Google Analytics per a Firestore, el que permet accedir a un seguiment detallat de participació i de conversions.

3. Objectius i abast

3.1 Objectius

L'usuari final del producte son persones familiaritzades amb les lectures conjuntes i hi ha participat algun cop en una o tenen intenció de fer-ho. El públic objectiu secundari són persones amb interès per la lectura però no han sentit parlar mai d'una lectura conjunta. Es vol acostar aquesta eina a nous públics per poder-los oferir una alternativa a la idea tradicional de la lectura.

3.1.1 Objectius de client

Els objectius de client aquells que reflecteixen el que el client espera aconseguir o els problemes que vol resoldre utilitzant un producte o servei concret. Aquests objectius es centren en abordar les necessitats i preferències del client.

- Crear lectures conjuntes a partir d'un llibre concret.
- Descobrir lectures conjuntes actives.
- Participar a una lectura conjunta.
- Configurar la lectura conjunta per adaptar-la a les necessitats del client.

3.1.2 Objectius de producte

Els objectius de producte venen definits, en aquest cas, pel desenvolupador, amb el propòsit de cobrir els objectius de client.

3.1.2.1 Objectius principals

Els objectius mínims a complir per tenir una aplicació funcional són els següents:

- Crear i configurar sales de lectura conjunta dedicades a un llibre concret.
- Gestió d'usuaris: els usuaris es poden enregistrar i configurar el seu perfil. També poden iniciar i tancar sessió.
- Veure un llistat de les sales de lectura obertes.
- Permetre als usuaris unir-se a les sales de lectura i xatejar. Allotjar un mínim de deu persones a les sales de xat.

- Configurar esdeveniments per les sales de lectura en format de calendari.
- Veure informació d'una sala de lectura concreta.
- Veure les dades del llibre al qual es dedica una sala de lectura.

3.1.2.2 Objectius secundaris

- Dividir les sales de lectura per estat.
- Gestió d'usuaris a les sales de lectures conjuntes. La gestió inclou l'expulsió d'usuaris d'una sala de xat.
- Poder incorporar una imatge de les portades dels llibres.
- Permetre a l'usuari pujar una imatge per al seu de perfil.
- Cercar entre totes les sales de lectura disponibles a partir de diversos filtres.
- Enviar notificacions i recordatoris sobre les lectures conjuntes.
- Crear enquestes o votacions a les sales de xat.
- Censurar texts a les sales de xat.
- Enviar una sol·licitud per poder entrar a una sala de lectura conjunta.
- Gestió d'usuaris a les sales de lectures conjuntes segons el rol de l'usuari. La gestió inclou donar o treure permisos a usuaris i acceptar sol·licituds d'admissió a la sala.
- Afegir usuaris a la llista d'amics d'un perfil.
- Denunciar missatges maliciosos o conductes inapropiades a les sales de xat.

3.2 Abast

Com s'ha pogut veure en el apartat 2.2, en el mercat ja hi ha eines enfocades a la comunitat lectora. L'objectiu d'aquest projecte es cobrir una necessitat i, per tant, el límit es marca en funcionalitats les quals altres plataformes ja donen servei. Això inclou:

- Proporcionar un registre de lectures. Els usuaris no tindran llistes de lectures al seu perfil com llibres pendents o llegits.
- Permetre als usuaris llegir llibres a l'aplicació. El projecte no és una plataforma de lectura online.
- Recomanacions de llibres a partir de lectures anteriors. Tal com tampoc hi haurà un històric de llibres llegits, no es faran recomanacions en base a aquests.

4. Metodologia

La metodologia triada pel desenvolupament d'aquest projecte és el Lean Agile^[25]. L'Agile es basa en iteracions les quals donen resultats des d'etapes primerenques del projecte i incrementen el valor del producte final poc a poc. Tant els requeriments com els resultats es reavaluen i analitzen al final de cada iteració per decidir com procedir amb el desenvolupament. El Lean té com a objectiu reduir costos y entregar el producte de manera eficient. Aquesta metodologia es l'adient pel desenvolupament del projecte ja que:

- Un dels recursos més rellevants del projecte és el temps. Degut a la limitació de temps és primordial tenir un producte funcional mínim lo abans possible.
- Es tracta d'un projecte el qual necessita constant seguiment i reavaluació per aconseguir lliurar el producte més adient per als usuaris. A mesura que es desenvolupa el producte es pot avaluar millor les necessitats següents.

Tot i que l'Agile té diferents maneres d'implementar-se depenent dels recursos, les etapes que es segueixen durant cada iteració la implementació de l'aplicació son les següents^[26]:

1. **Planificació:** S'inicia el procés identificant els objectius que es vol assolir en la iteració.
2. **Disseny:** Cal trobar i entendre bé els requisits funcionals i tecnològics necessaris per completar l'objectiu i s'ha d'estudiar la millor alternativa per implementar cada una de les funcionalitats.
3. **Desenvolupament:** En aquesta etapa es desenvolupen els requisits que s'han estudiat durant l'etapa de disseny. En les primeres iteracions aquesta fase ocuparà un gran espai de temps degut al desconeixement inicial de l'eina de desenvolupament. Aquest temps s'ha de veure substancialment reduït a mesura que s'avança en la producció del projecte.
4. **Testing:** Tot i que una part important del testeig es fa simultàniament amb desenvolupament, cal verificar el correcte funcionament del conjunt de les funcionalitats i buscar situacions no tingudes en compte durant el disseny, és a dir, *bugs* o errors.
5. **Revaluació:** En la fase final de la iteració cal fer una retrospectiva dels objectius que s'han pogut assolir i les complicacions que han sorgit durant el procés. Aquest

anàlisi permet fer una reavaluació dels objectius i reconsiderar les prioritats del projecte.

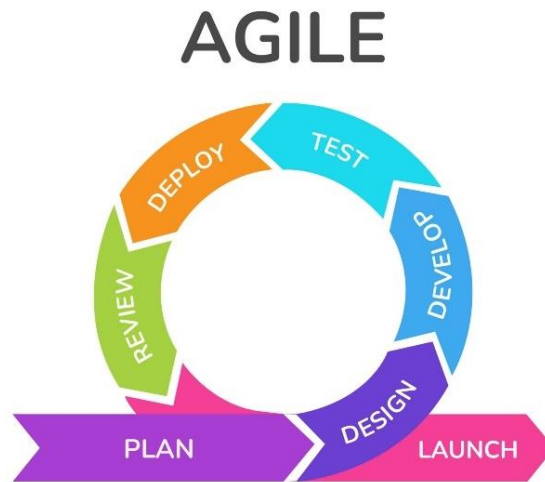


Figura 6. Fases de l'Agile. Font: [27]

L'aplicació d'Agile a aquest projecte presenta les següents avantatges i inconvenients:

Avantatges	Inconvenients
Està garantit un lliurament mínim del producte.	Com Agile està orientat al lliurament de producte, la documentació pot quedar en segon lloc.
El procés està adaptat per admetre a canvis.	Dificultat a l'hora de predir quin serà l'estat del producte final.
Hi ha una millora ràpida i constant del producte.	És una metodologia més fàcil d'aplicar en equip.

Taula 7. Avantatges i inconvenients de l'Agile. Font: [28]

4.1 Iteracions

Cada iteració durant el desenvolupament engloba un conjunt de funcionalitats a implementar. És important seguir l'ordre concret ja que iteracions posteriors tenen

dependències amb iteracions prèvies. A cada una de les iteracions cal fer una retrospectiva dels objectius assolits i una reavaluació dels objectius posteriors. Així doncs, la següent estructura es tracta d'una referència, però no unes pautes inalterables.

4.1.1 Estudi de tecnologies

Aquesta és la primera iteració després de l'entrega de l'Avantprojecte. Cal fer un estudi de les diferents alternatives tecnològiques llistades a l'apartat 2.4 que hi ha per al desenvolupament de l'aplicació i, a part, la implementació dels serveis de *backend*, tal com la persistència de dades. La tria final es documenta amb els motius i característiques d'aquesta.

4.1.2 Configuració i model de navegació

La segona iteració és on es dur a terme el primer contacte amb el *framework* i el nou llenguatge de programació, s'assumeix que les primeres passes seran lentes degut a la poca familiaritat amb el mètode de desenvolupament. En aquest primer punt es fa la creació del projecte i el repositori a Github. També es fa tria de l'arquitectura que més s'adapta al projecte. Finalment, és necessari programar el primer element de l'aplicació, la navegació.

4.1.3 Implementació de llibres

En aquest punt del desenvolupament es crea un altre element imprescindible de l'aplicació, els llibres. Els llibres emmagatzemen són creats pels usuaris el mateix moment que creen les sales de lectura. Es crea la interfície del perfil dels llibres i el formulari per a la creació d'aquests. També és necessari crear el model del domini i de dades i cal fer la connexió a la base de dades per poder-hi escriure la informació. També cal fer la lògica necessària perquè no hagi llibres duplicats a la base de dades.

4.1.4 Implementació de sales de lectura

La quarta iteració és crucial degut a que es fa el disseny de les sales de lectura, tant a nivell artístic com funcional. Igual com en la iteració prèvia, es crea la interfície del perfil de les sales de lectura i el formulari per a la creació d'aquestes. Degut a la complexitat del conjunt i essent una de les primeres etapes, es tracta d'una de les iteracions més llargues.

4.1.5 Navegació entre pàgines

Un cop implementats els elements anteriors cal crear els enllaços entre ells. Un exemple n'és que l'usuari ha de poder crear un llibre i, a partir d'aquest, una sala de lectura. En aquest punt també es desenvolupa un llistat de sales de lectura existents i un botó que redirigeixi al formulari de creació d'aquestes.

4.1.6 Implementació d'usuaris

Aquesta iteració està dedicada a la creació i disseny dels usuaris. Cal crear totes les pantalles relacionades amb aquests i les accions que poden dur a terme tal com donar-se de baixa o editar el seu perfil. També s'introdueix l'inici de sessió.

4.1.7 Implementació del xat

La setena iteració és la darrera per concloure el MVP. En aquesta fase es creen les sales de xat dins les sales de lectura i s'adapta la navegació. També cal mostrar una interfície adient pel llistat de sales de xat. Al final d'aquesta etapa es fan proves exhaustives i es solucionen errades i *bugs*. La documentació final ha d'estar al dia.

4.1.8 Incorporació d'imatges

La vuitena iteració és la primera iteració prevista després de rebre el primer *feedback* del MVP. Ja es disposa d'un producte mínim viable i ara es millora per incrementar el seu valor. A part de treballar en errades o detalls que es detectin en l'entrega del MVP, és necessari incorporar els elements més visuals de l'aplicació, portades de llibres i fotos de perfil dels usuaris.

4.1.9 Implementació de rols i sol·licituds

Per a aquesta iteració, després d'haver fet una altra avaluació amb els usuaris, es redefeixen els requisits adients i s'incorpora el sistema de rols i privilegis i les sol·licituds a les sales de lectura.

4.1.10 Elements addicionals del xat

Per a la desena iteració del desenvolupament del projecte, a part de seguir el mateix procés de reavaluació de requeriments, s'incorporen els dos principals elements que donen vida a les sales de lectura conjunta, les enquestes o votacions i la censura de texts.

4.1.11 Amics i notificacions

En aquest punt del projecte s'incorporen les notificacions i el sistema de llistes d'amics i sales de l'aplicació. És necessari adaptar els perfils dels usuaris per poder incloure aquestes noves funcionalitats.

4.1.12 Denuncies i missatges a la desenvolupadora

Per a complir els últims objectius previstos en el disseny inicial, s'introdueix la possibilitat de denunciar missatges maliciosos i enviar missatges a la desenvolupadora. Juntament amb aquests nous elements es creen noves pantalles per recuperar la contrasenya de l'usuari en cas d'oblit. A partir d'aquest punt es fan més iteracions en base a les observacions i comentaris dels usuaris de l'aplicació.

5. Desenvolupament

5.1 Definició de requeriments funcionals i tecnològics

Per diferenciar quins són els requisits mínims que cal que el projecte implementi per poder oferir el producte mínim viable, aquests s'han classificat per prioritat. Els requeriments d'alta prioritat impliquen que la seva implementació és essencial. Els requeriments de baixa prioritat incrementen la qualitat del producte i atorguen valor afegit.

5.1.1 Requeriments funcionals

Els requeriments funcionals s'han dividit, a part de per prioritat, per àmbit de l'aplicació. S'ha fet diferència entre les funcionalitats que necessita l'aplicació en àmbit general i les funcionalitats necessàries perquè el servei que ofereix la plataforma, que són les lectures conjuntes, estigui cobert.

5.1.1.1 Requeriments funcionals d'alta prioritat

Requeriments generals de l'aplicació

- L'usuari ha de poder-se donar d'alta a l'aplicació.
- L'usuari ha de poder iniciar sessió a l'aplicació.
- L'usuari ha de poder configurar el seu perfil.
- L'usuari ha de poder verificar la llista de sales actives i obertes.
- L'usuari ha de tenir una secció personal amb el llistat de lectures conjuntes en les que participa.
- L'usuari no enregistrat pot visualitzar les sales actives però no pot fer cap acció respecte aquestes.

Requeriments envers les sales de lectures conjuntes

- L'usuari ha de poder crear una sala de lectura.
- L'usuari ha de poder consultar la informació d'una sala de lectura.
- L'usuari ha de poder entrar a una sala de lectura.
- L'usuari ha de poder xatejar amb els membres de la lectura conjunta.
- L'usuari ha de poder abandonar una sala de lectura conjunta.

- L'usuari administrador ha de poder configurar l'estructura de la lectura conjunta.
- L'usuari administrador pot crear esdeveniments en un calendari adjunt a la lectura conjunta.
- L'usuari ha de poder afegir un llibre a la base de dades durant la creació d'una lectura conjunta.

5.1.1.2 Requeriments funcionals de baixa prioritat

Requeriments generals de l'aplicació

- L'usuari ha de poder filtrar les sales de lectura actives per diferents filtres.
- L'usuari ha de poder tenir una imatge de perfil.
- L'usuari ha de poder-se donar de baixa a l'aplicació.
- L'usuari ha de poder guardar una sala en esborrany per consultar-la després.
- L'usuari ha de poder enviar comentaris i suggeriments a la desenvolupadora.
- L'usuari ha de poder consultar, un cop a l'any, estadístiques globals i personals sobre diferents aspectes de les lectures conjuntes.
- L'usuari ha de poder demanar verificació de perfil com a persona reconeguda a la comunitat lectora.
- L'usuari ha de poder afegir usuaris a una llista personal d'amics.
- L'usuari ha de poder consultar un llistat de llibres llegits en lectures conjuntes ja acabades.
- Creació de portal web de presentació de l'aplicació.

Requeriments envers les sales de lectures conjuntes

- L'usuari administrador ha de poder cedir i enretirar privilegis.
- L'usuari ha de poder afegir una portada de llibre durant la creació de la sala de lectura.
- L'usuari ha de poder marcar una part d'un missatge o tot un missatge com a *spoiler*.
- L'usuari ha de poder sol·licitar accés a una sala de lectura.
- L'usuari administrador ha de poder crear enquestes o votacions sobre temes relacionats amb el llibre o la lectura conjunta.
- L'usuari ha de poder valorar el llibre de 0 a 5 un cop acabada finalitzada la lectura conjunta.

- L'usuari amb privilegis ha de poder enviar i programar notificacions.
- L'usuari ha de poder denunciar missatges o usuaris que incompleixin les normes de conducta.
- L'usuari ha de poder-se connectar a una API per recopilar les dades d'un llibre.

5.1.2 Requeriments tecnològics

5.1.2.1 Requeriments tecnològics d'alta prioritat

- Utilitzar de desenvolupament de l'aplicació que permeti el màxim d'adaptabilitat entre diferents sistemes operatius mòbils.
- Persistir les dades tant dels usuaris, com les sales de lectura i els llibres, al núvol.
- Mantenir la comunicació per xat segura.
- Allotjar un mínim de deu usuaris per sala de lectura.

5.1.2.2 Requeriments tecnològics de baixa prioritat

- Tenir una còpia de seguretat de les dades actualitzada de manera diària.
- Permetre a l'usuari enregistrar-se a l'aplicació amb comptes d'altres entitats com Google.
- Connectar-se a una externa API per col·lectar les dades dels llibres.
- Tenir un sistema d'allotjament web per allotjar el portal presentació de l'aplicació.

5.2 Tria de tecnologies

En base als requeriments definits a l'apartat 5.1 s'ha fet tria de les tecnologies a utilitzar per desenvolupar l'aplicació.

Tipus de desenvolupament mòbil

En primer lloc, per poder cobrir el requeriment de maximitzar l'adaptabilitat entre sistemes operatius tenint en compte la limitació de recursos que té el projecte, s'ha triat desenvolupar l'aplicació amb un *framework* multiplataforma o híbrid.

Software de desenvolupament

En base a la primera elecció es concreten diferents alternatives per escollir l'eina de desenvolupament, especificades a l'apartat 2.4.2. Tenint en compte les diferents avantatges i inconvenients de cada tecnologia s'ha decidit fer ús de Flutter. Es tracta d'una opció factible per presentar un MVP de la manera més ràpida possible i no es considera un *framework* complicat d'aprendre^[29] a diferència d'altres, com React Native. També mostra millor previsió en escalabilitat que Ionic i un rendiment superior a Xamarin.

Backend

Un altre requisit del projecte és dur a terme la persistència de dades al núvol. Tenint en consideració que l'aplicació és un projecte petit, però que opta a escalar i, a part, hi ha un temps limitat pel desenvolupament d'aquest, es requereix que la complexitat de configuració sigui baixa i no convé malbaratar recursos quan poden estalviar-se utilitzant l'alternativa adient. Així doncs es decideix fer ús de Firebase com a proveïdor de serveis de *backend*^[30].

5.3 Arquitectura

5.3.1 Arquitectura de serveis backend

A continuació es mostra un diagrama de seqüència molt bàsic del procés d'escriptura i lectura a l'aplicació.

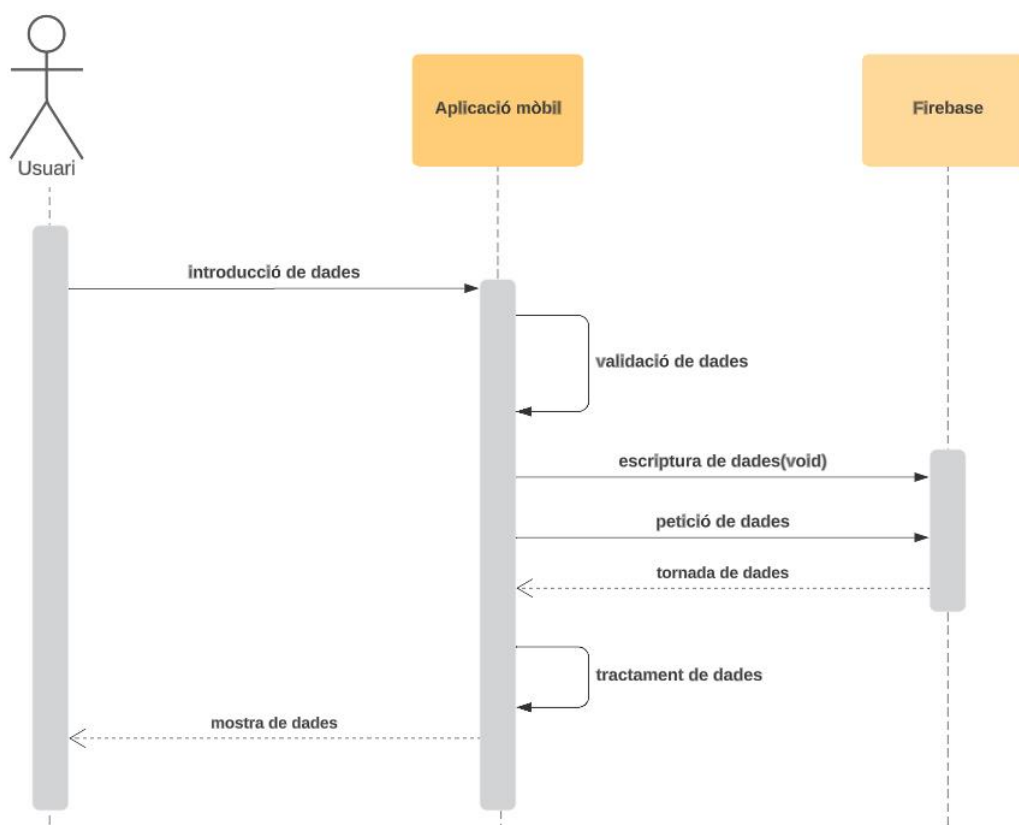


Figura 7. Diagrama general dels processos de l'aplicació. Font: Elaboració pròpia

Presa la decisió de treballar amb Firebase i en base als requeriments de l'aplicació, s'han seleccionat els següents serveis per donar suport a l'aplicació.

Firestore Database per a persistir les dades de l'aplicació. En aquesta base de dades s'emmagatzemen les dades que conformen el perfil de l'usuari, les sales de lectura, incloent esdeveniments, i els llibres. És la opció adient degut a que es tracta d'una base de dades ràpida i fàcil de configurar pensada sobretot per fer-hi consultes simples.

Authentication per permetre als usuaris autenticar-se a l'aplicació mitjançant un correu i contrasenya. És una tria més convenient ja que el propi servei s'encarrega

de la gestió d'estat de sessió de l'usuari i d'enviar correus de validació entre d'altres funcions.

Firestore per gestionar els missatges del xat i les sales de xat en sí. Tot i que ja es compte amb una base de dades, s'ha decidit tenir dues bases de dades separades per a l'aplicació és perquè, en general, no es requereix de *queries* complexes i el model de dades de Realtime Database és més simple. Pel xat doncs, com presenta un flux de dades superior i Firestore té grans avantatges a l'hora d'escalar, així com de velocitat a l'hora de fer *queries*, es considera l'eina més adient.

Cloud Storage es fa servir per incorporar imatges de les portades dels llibres i dels perfils dels usuaris a l'aplicació.

Cloud Messaging, tot i estar esmentat com a servei necessari, al no haver-se implementat el sistema de notificacions, no se n'ha fet ús.

5.3.2 Arquitectura del codi

Hi ha tantes maneres d'estructurar una aplicació com desenvolupadors existeixen. No hi ha una resposta correcta, però existeixen diversos enfocaments que es recomanen per poder fer el codi més fàcil de mantenir i escalar.

L'estructura utilitzada en aquest projecte és *feature-first*, és a dir, es crea un directori exclusiu per a cada *feature* o element de l'aplicació i, dins de cadascun, s'inclouen les distintives capes d'un projecte per capes: persistència, domini, presentació i aplicació. S'ha seguit la guia d'Andrea Bizzotto^[31], desenvolupador de Flutter i Firebase.

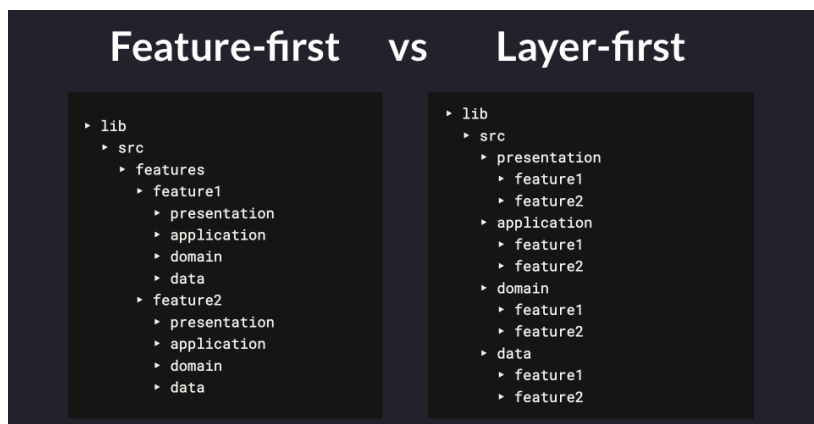


Figura 8. Comparació entre arquitectura feature-first i layer-first. Font: [32]

Com és habitual en molts projectes de software, pot haver-hi codi compartit o requerit per diferents *features*. En el cas de Flutter es veu destacat ja que, degut a que tot funciona amb *widgets*, molts d'ells es reciclen per evitar codi duplicat. Per aquests casos es creen carpetes addicionals, al mateix nivell que el directori que recull tots els *features*.

A continuació es pot veure com ha quedat la distribució final:

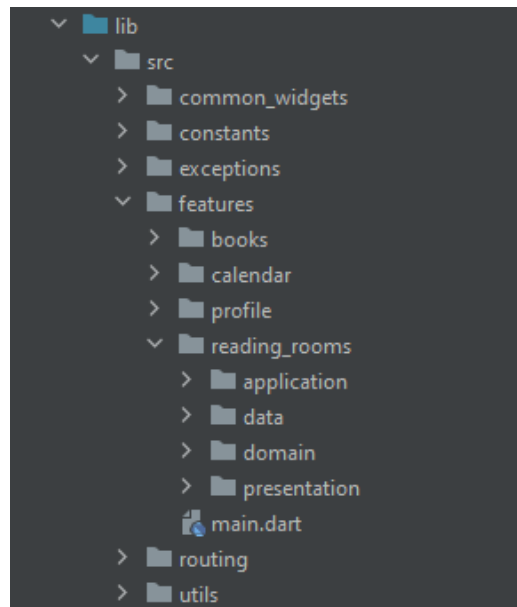


Figura 9. Captura de l'arquitectura final de l'aplicació. Font: Elaboració pròpia

Cada una de les capes dels *features*, de superior a inferior, estan conformades pel següent contingut:

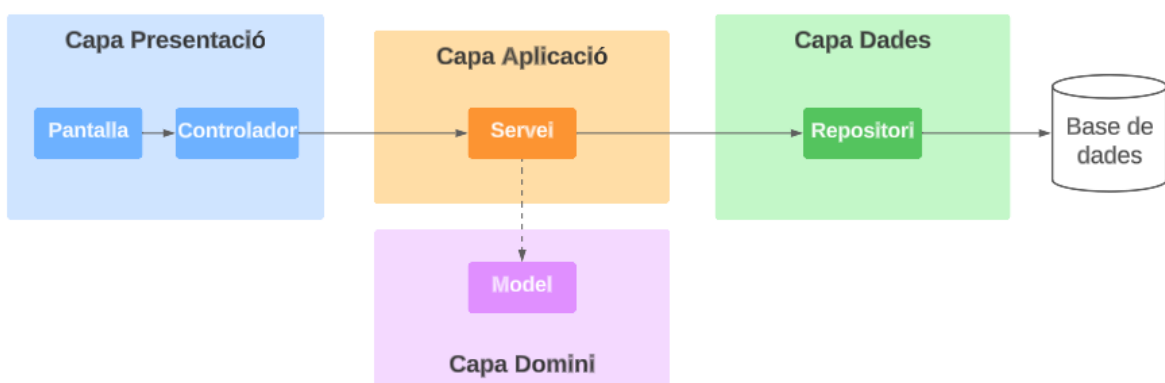


Figura 10. Esquema de capes i comunicació de l'aplicació. Font: Elaboració pròpia

- **Presentació:** Conté les pantalles visibles per a l'usuari, la interfície de l'aplicació. També es troben els controladors, encarregats de contactar amb el servei per dur a terme les accions de persistència o peticions.

- **Aplicació:** Aquesta capa conté el servei que contacten amb els diferents repositoris. Un exemple en pot ser que, quan l'usuari crea una nova sala de lectura, cal crear tant una entrada amb les dades de la sala a Realtime Database, i una sala de xat a Firestore. La capa de servei crida als dos repositoris encarregats d'executar les comandes pertinents. També contacta amb el domini per transformar les dades d'entrada al format adient per ser enviades a la base de dades.
- **Domini:** Aquesta capa conté els models de dades i es troba a la mateixa alçada que l'aplicació ja que es cridada per aquesta. Cada membre del domini té la capacitat de transformar-se d'un objecte a un mapa JSON o en un objecte a partir d'un dataSnapshot, és a dir, la tornada de les bases de dades.
- **Dades:** La capa de dades o persistència està composta d'un o més repositoris que parlen directament amb els serveis de backend externs de Firebase. Cada funció dur a terme una acció com lectura, escriptura, o actualització de dades.

5.3 Casos d'ús

5.3.1 Actors

Un cas d'ús és la descripció d'una interacció específica entre un actor i un sistema per aconseguir un objectiu específic. Per tant, els actors d'un cas d'ús són els elements externs, com persones o un sistema extern, que interactuen amb un sistema concret.

En aquest projecte es diferencien els següents dos actors:

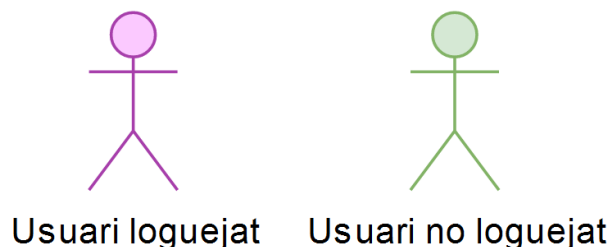


Figura 11. Tipus d'usuaris. Font: Elaboració pròpia

- **Usuari loguejat:** aquest actor és un usuari que ja ha iniciat sessió a la plataforma i, per tant, té accés a una sèrie de funcionalitats específiques per a les que es requereix un perfil al sistema.

- Usuari no loguejat: aquest actor és un usuari que no ha iniciat sessió a la plataforma. Té un conjunt limitat d'accions disponibles.

Un mateix usuari pot prendre el rol de qualsevol dels dos actors depenent del seu estat dins de la plataforma.

5.3.2 Diagrames dels casos d'ús

Els esquemes a continuació mostren les diferents accions que poden fer els dos actors a l'aplicació segons el servei.

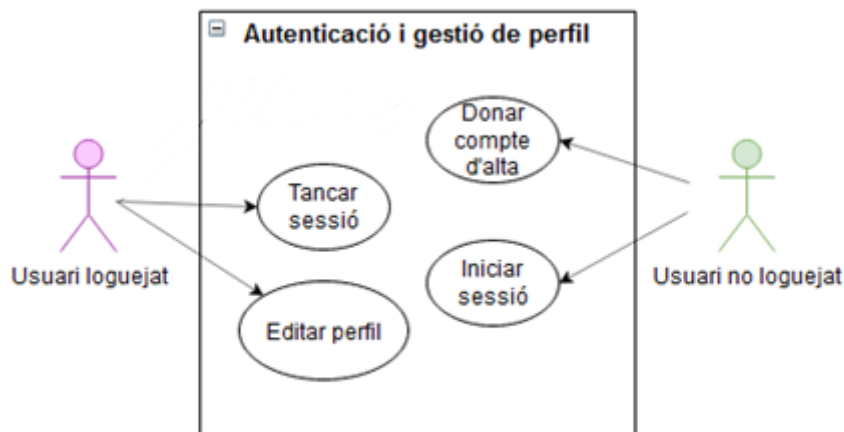


Figura 12. Casos d'ús d'autenticació i gestió de perfil. Font: Elaboració pròpia

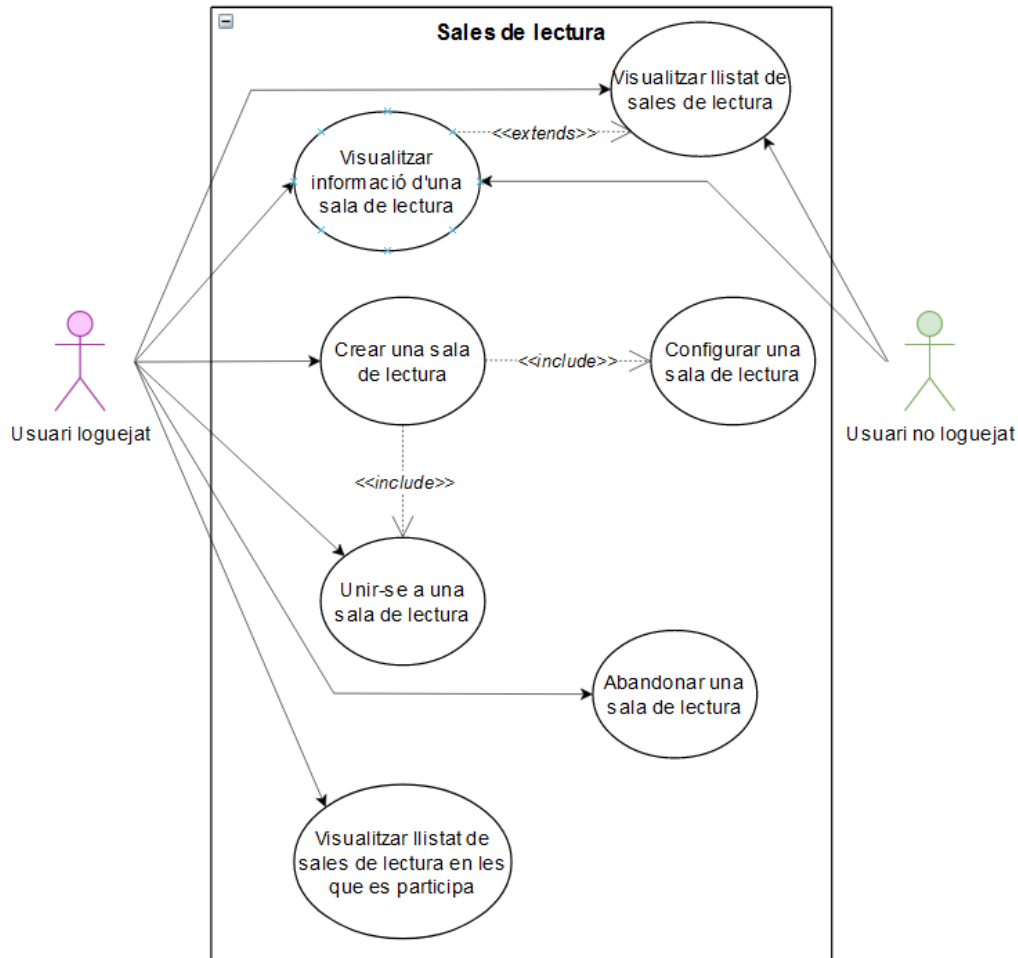


Figura 13. Casos d'ús de les sales de lectura. Font: Elaboració pròpia

5.3.3 Descripció dels casos d'ús

A continuació, cadascun dels casos d'ús exposats a l'apartat previ, es veuen explicats amb detall:

5.3.3.1 Autenticació i gestió de perfil

Donar compte d'alta

Actor: Usuari no loguejat

Pre-condicions: L'usuari no loguejat ha d'haver iniciat l'aplicació.

Post-condicions: L'usuari no loguejat té un compte al sistema.

Flux normal d'execució:

1. L'usuari prem el botó de "Perfil" de l'aplicació.
2. El sistema redirigeix a l'usuari al formulari per crear un nou compte ja que no està donat d'alta.
3. L'usuari omple les dades i prem validar.
4. El sistema valida les dades, aquestes son correctes i envia un correu electrònic al correu proporcionat. El sistema mostra una pantalla informat que cal validar el correu electrònic.
5. L'usuari valida el correu electrònic i torna a l'aplicació.
6. El sistema comprova que s'ha validat el correu electrònic, afegeix les dades de l'usuari a la base de dades i mostra a l'usuari el seu perfil. L'usuari té un compte a l'aplicació.

Flux alternatiu d'execució:

- 4.b. El sistema valida les dades, aquestes no son correctes i ressaltar els errors. Torna al punt anterior.
- 5.b. L'usuari no valida el correu electrònic.
6. El sistema no afegeix l'usuari a la base de dades.

Iniciar sessió

Actor: Usuari no loguejat

Pre-condicions: L'usuari no loguejat ha d'haver iniciat l'aplicació i haver-se donat d'alta al sistema i validat el seu correu electrònic.

Post-condicions: L'usuari ara es troba loguejat al sistema.

Flux normal d'execució:

1. L'usuari prem el botó de "Perfil" de l'aplicació.
2. El sistema redirigeix a l'usuari al formulari per iniciar sessió.
3. L'usuari omple les dades i prem validar.
4. El sistema valida les dades i aquestes son correctes. Es recupera la sessió de l'usuari i ara aquest es troba loguejat a l'aplicació.

Flux alternatiu d'execució:

- 4.b. El sistema valida les dades, aquestes no son correctes i ressaltar els errors.

Tancar sessió

Actor: Usuari loguejat

Pre-condicions: L'usuari ha d'estar loguejat a l'aplicació i a la secció de "Perfil".

Post-condicions: S'ha expulsat a l'usuari de la sessió activa.

Flux normal d'execució:

1. L'usuari prem el botó de "Sortir del compte".
2. El sistema treu a l'usuari de la sessió activa i el redirigeix a la pàgina d'inici de sessió.

Editar perfil

Actor: Usuari loguejat

Pre-condicions: L'usuari ha d'estar loguejat a l'aplicació i a la secció de "Perfil".

Post-condicions: S'ha actualitzat el perfil de l'usuari a la base de dades.

Flux normal d'execució:

1. L'usuari prem el botó de "Editar perfil".
2. El sistema mostra el perfil de l'usuari en format editable.
3. L'usuari fa els canvis convenients i prem "Desar".
4. El sistema valida els canvis actualitza el perfil de l'usuari a la base de dades.

Flux alternatiu d'execució:

- 3.b. L'usuari fa els canvis convenients i prem "Cancel·lar".
- 4.b. El sistema mostra una alerta de confirmació on s'indica que els canvis no desats es perdran.

4.c. El sistema valida els canvis i no son correctes. Es ressalten els errors i es torna al format de perfil editable.

5.b. L'usuari confirma i es torna a la secció de configuració de perfil.

5.3.3.2 Sales de lectura

Visualitzar llistat de sales de lectura

Actor: Usuari loguejat/no loguejat

Pre-condicions: L'usuari ha d'haver iniciat l'aplicació.

Post-condicions: Es visualitza el llistat de sales de lectura existents.

Flux normal d'execució:

1. L'usuari prem el botó de "Sales de lectura".
2. El sistema mostra un llistat amb les sales de lectura recuperades de la base de dades.

Visualitzar informació d'una sala de lectura

Actor: Usuari loguejat/no loguejat

Pre-condicions: L'usuari ha d'haver iniciat l'aplicació.

Post-condicions: L'usuari visualitza la informació d'una sala de lectura.

Flux normal d'execució:

1. L'usuari prem el botó de "Sales de lectura".
2. El sistema un llistat amb les sales de lectura que hi ha al sistema.
3. L'usuari prem la sala de lectura de la qual vol veure més informació.
4. El sistema mostra una pàgina amb tota la informació relacionada amb la sala de lectura.

Crear una sala de lectura

Actor: Usuari loguejat

Pre-condicions: L'usuari ha d'estar loguejat a l'aplicació i es troba a la secció personal de sales de lectura.

Post-condicions: S'ha creat una nova sala de lectura.

Flux normal d'execució:

1. L'usuari prem el botó de "Crear una nova sala de lectura".
2. El sistema mostra el formulari de creació d'una sala de lectura.
3. L'usuari introdueix la informació pertinent i prem "Crear".
4. El sistema valida els canvis i afegeix una nova sala de lectura a la base de dades.
L'usuari és afegit com a participant de la sala de lectura.

Flux alternatiu d'execució:

- 3.b. L'usuari fa els canvis convenients i prem "Cancel·lar".
- 4.b. El sistema mostra una alerta de confirmació on s'indica que els canvis no desats es perdran.
- 4.c. El sistema valida els canvis i no son correctes. Es ressalten els errors i es torna al format de perfil editable.

Configurar/actualitzar una sala de lectura

Actor: Usuari loguejat

Pre-condicions: L'usuari ha d'estar loguejat a l'aplicació i es troba a la secció personal de sales de lectura amb al menys una sala de lectura existent creada per ell mateix.

Post-condicions: La sala de lectura s'ha actualitzat a la base de dades.

Flux normal d'execució:

1. L'usuari prem la sala de lectura la qual vol actualitzar. Aquesta ha de ser creació de l'usuari.
2. El sistema mostra el xat de la sala de lectura.
3. L'usuari prem la capçalera de la sala de xat.
4. El sistema mostra el perfil de la sala de lectura en format de lectura.

5. L'usuari prem el botó "Editar".
6. El sistema mostra el perfil de la sala de lectura el formulari en format edició.
7. L'usuari actualitza la informació pertinent i prem "Desar".
8. El sistema valida els canvis i actualitza la sala de lectura a la base de dades.

Flux alternatiu d'execució:

- 6.b. El sistema valida els canvis i no son correctes. Es ressalten els errors i es torna al format de perfil editable.

Unir-se a una sala de lectura

Actor: Usuari loguejat

Pre-condicions: L'usuari ha d'haver iniciat l'aplicació.

Post-condicions: L'usuari s'ha unit a una sala de lectura.

Flux normal d'execució:

1. L'usuari prem el botó de "Sales de lectura".
2. El sistema mostra un llistat amb les sales de lectura recuperades de la base de dades.
3. L'usuari tria la sala de lectura i prem l'entrada.
4. El sistema el perfil de la sala de lectura mostra en format de lectura.
5. L'usuari prem el botó de "Unir-se".
6. El sistema mostra una pantalla demanant confirmació per unir-se a la sala de lectura.
7. L'usuari prem "Sí".
8. El sistema afegeix a l'usuari a la sala de lectura.

Flux alternatiu d'execució:

7. L'usuari prem "No"
8. El sistema torna l'usuari al perfil de la sala de lectura.

Abandonar una sala de lectura

Actor: Usuari loguejat

Pre-condicions: L'usuari ha de ser participant d'una sala de lectura.

Post-condicions: L'usuari ja nos serà membre de la sala de lectura abandonada.

Flux normal d'execució:

1. L'usuari tria la sala de lectura la qual vol abandonar de la secció de sales de lectura en les que es participa.
2. El sistema mostra el perfil de la sala de lectura en format de lectura.
3. L'usuari prem el botó "Abandonar".
4. El sistema mostra una alerta per confirmar la decisió de l'usuari.
5. L'usuari reafirma la decisió.
6. El sistema elimina l'usuari de la sala de lectura i l'actualitza a la base de dades.

Flux alternatiu d'execució:

- 5.b. L'usuari prem "Cancel·lar".
- 6.b. El sistema torna a l'usuari al perfil de la sala de lectura en format de lectura.

Visualitzar llistat de sales de lectura en les que es participa

Actor: Usuari loguejat

Pre-condicions: L'usuari ha de ser participant d'almenys una sala de lectura.

Post-condicions: L'usuari visualitza el llistat de sales de lectura en les que participa.

Flux normal d'execució:

1. L'usuari prem el botó de "Les meves sales de lectura".
2. El sistema mostra un llistat amb les sales de lectura, recuperades de la base de dades, en les que l'usuari participa.

Eliminar una sala de lectura

Actor: Usuari loguejat

Pre-condicions: L'usuari ha d'estar loguejat a l'aplicació i es troba a la secció personal de sales de lectura amb al menys una sala de lectura existent creada per ell mateix.

Post-condicions: La sala de lectura s'ha eliminat de la base de dades

Flux normal d'execució:

1. L'usuari prem la sala de lectura la qual vol actualitzar. Aquesta ha de ser creació de l'usuari.
2. El sistema mostra el xat de la sala de lectura.
3. L'usuari prem la capçalera de la sala de xat.
4. El sistema mostra el perfil de la sala de lectura en format de lectura.
5. L'usuari prem el botó "Eliminar".
6. El sistema verifica que la sala sigui apta per eliminar, és a dir, que sigui data després de la data de finalització de la sala de lectura o que l'administrador es trobi sol a la sala de lectura. Si la sala és apta s'elimina la sala de lectura.

Flux alternatiu d'execució:

6. La sala no és apta per eliminar i es mostra un missatge que informa a l'usuari.

6. Implementació

6.1 Preparació entorn de desenvolupament

L'entorn triat pel desenvolupament del projecte és Android Studio, eina recomanada per Google pel desenvolupament d'Android^[33]. Aquest IDE, o entorn de desenvolupament integrat, ve amb funcions integrades i simplificades per donar suport al desenvolupament mòbil, com la creació d'emuladors. Degut a que Android Studio està construït sobre IntelliJ IDEA, es pot gaudir d'una gran quantitat de funcionalitats de desenvolupament pròpies del software sense necessitat de pagar-hi una llicència. També integra components exclusius de Flutter, com el rendiment, inspector o esquema de l'aplicació.

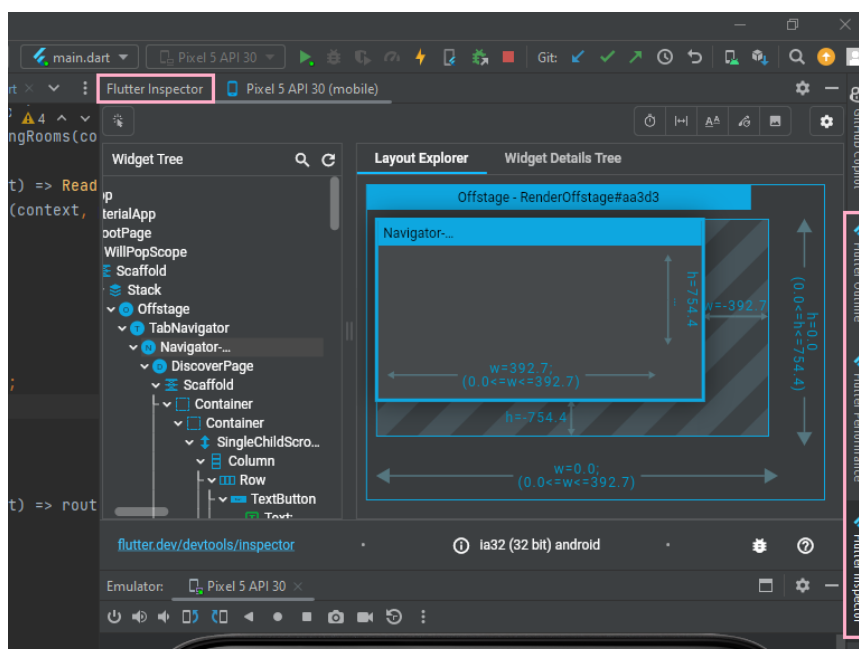


Figura 14. Eina Inspector de Flutter a Android Studio. Font: Elaboració pròpia

Tot i que hi ha altres entorns que ofereixen més *plugins* externs com VS Code es considera que no és motiu de suficient pes com per renunciar a la gran quantitat de solucions pràctiques que ofereix Android Studio.

Per deixar l'IDE llest pel desenvolupament cal instal·lar els *plugins* de Dart i Flutter del *Marketplace*. També és necessari instal·lar el SDK de Flutter des de qualsevol font oficial com docs.flutter.dev i actualitzar les variables del sistema.

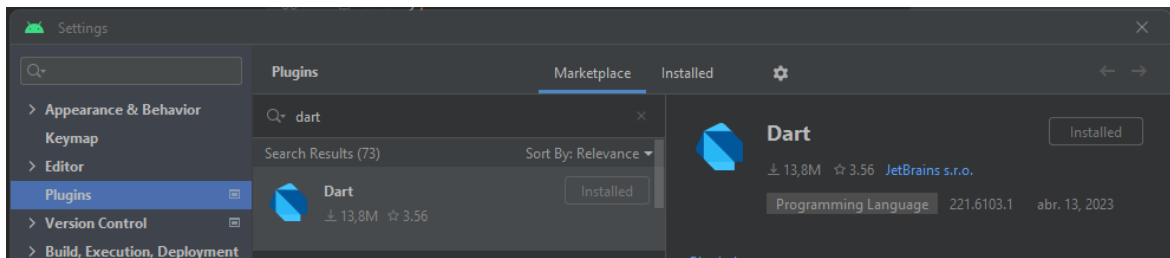


Figura 15. *Plugin* de Dart a Android Studio. Font: Elaboració pròpia

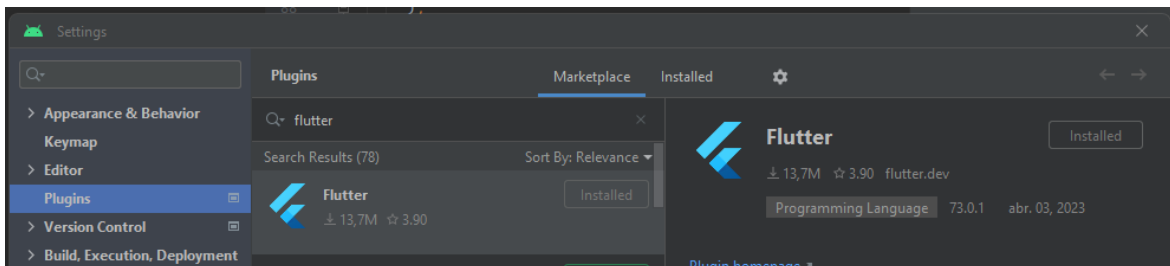


Figura 16. *Plugin* de Flutter a Android Studio. Font: Elaboració pròpia

Un cop llest pot executar-se la comanda `#flutter doctor` per veure si es compleixen tots els requeriments per començar a desenvolupar amb Flutter.

6.2 Implementació de Firebase al projecte

Per integrar els serveis de Google Firebase a Flutter cal seguir les següents instruccions^[34]:

1. Executar `#npm i -g firebase-tools` per instal·lar el CLI de Firebase, és a dir, poder executar comandes `#firebase`.
2. Executar `#firebase login` per iniciar sessió al compte de Firebase.
3. Executar `#dart pub global activate flutterfire_cli` per instal·lar el CLI de Firebase en la carpeta `pub-cache`.
4. Ara, en el directori del projecte, cal executar `#flutter pub add firebase_core` per instal·lar plugin encarregat de comunicar el projecte amb els serveis de Firebase.
5. Seguidament, amb `#flutterfire configure` es configura Firebase per a Flutter. Aquesta comanda crea el document `firebase_options.dart`, el qual associa l'aplicació local amb el descriptor corresponent de cada plataforma en el projecte Firebase.

Un cop finalitzat el procés, incorporar qualsevol servei és ara tan senzill com inicialitzar el producte a la consola de Firebase i afegir la dependència del servei desitjat al fitxer del projecte `pubspec.yaml`.

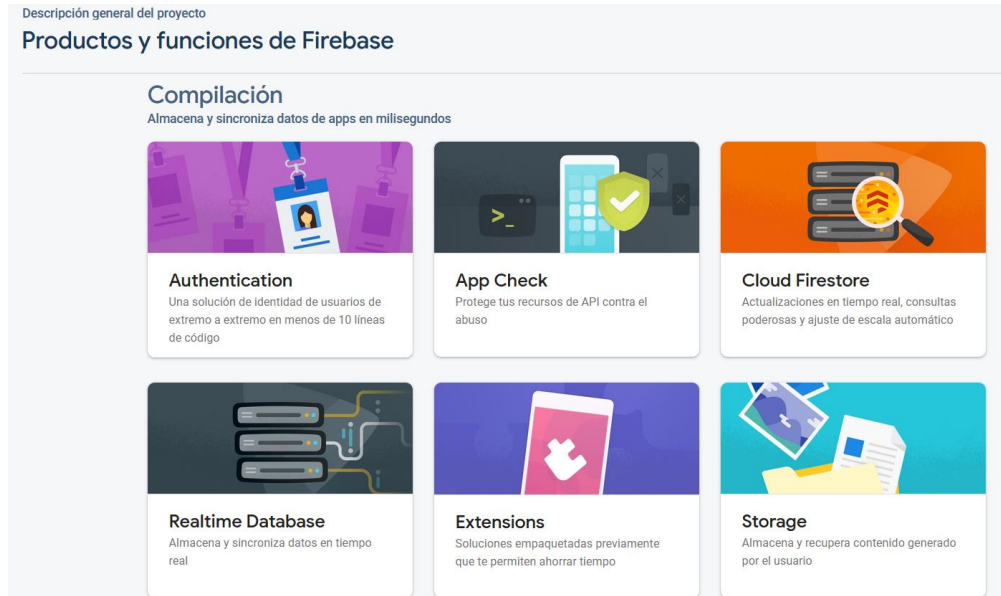


Figura 20. Botiga de productes de Firebase. Font: Elaboració pròpia

Firestore Realtime Database

Per tenir la base de dades a l'aplicació cal, en primer lloc, prémer la opció de la Figura 120 pertinent i, després, fa triar la ubicació d'on serà emmagatzemada la base de dades. Les opcions són Estats Units, Singapur o Bèlgica. Després es trien les regles de lectura i escriptura de la base de dades, per ala fase de producció es deixen les dues accions disponibles sense restriccions. Ara la base de dades ja està creada.

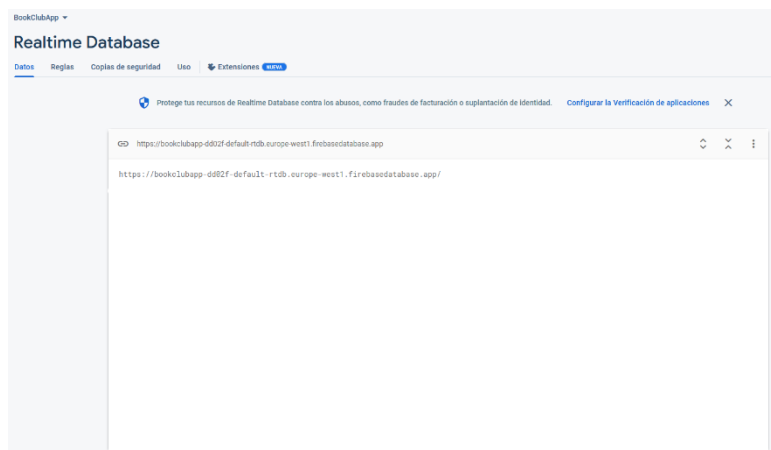


Figura 21. Consola de Realtime Database. Font: Elaboració pròpia

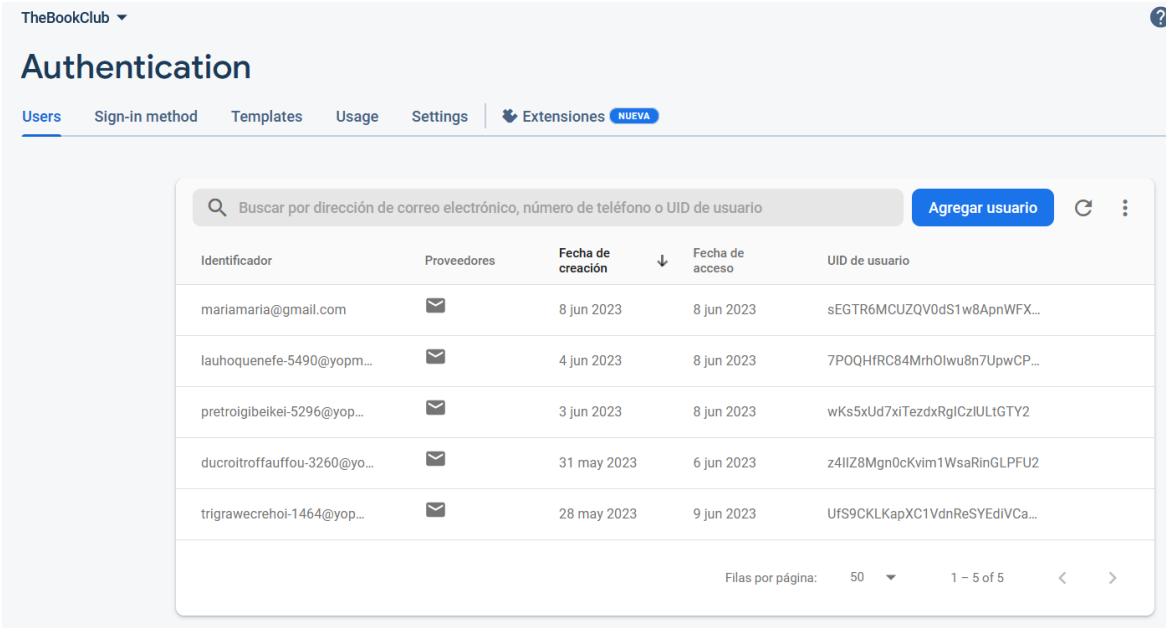
La manera més senzilla d'incorporar la base de dades al projecte local és executant la comanda `#flutter pub add firebase_database` en el directori del projecte, fet que incorporarà al fitxer de dependències la última versió de la llibreria.

Firestore Authentication

Tal com en el cas anterior, per incloure Firestore Authentication només cal prémer la opció corresponent en el catàleg de productes i, després, triar el mètode d'autenticació desitjat. Per a l'aplicació es farà servir el model correu electrònic i contrasenya, però hi ha moltes altres opcions, com ara Google amb OAuth.

Altre cop, per incorporar el sistema al projecte cal executar `#flutter pub add firebase_auth` al directori perquè s'inclogui al llistat de dependències.

A continuació es mostra el portal de Firestore Authentication amb uns quants usuaris donats d'alta:



The screenshot shows the 'Authentication' page in the Firebase console for 'TheBookClub'. It features a navigation menu with 'Users', 'Sign-in method', 'Templates', 'Usage', 'Settings', and 'Extensiones' (with a 'NUEVA' button). A search bar is present with the text 'Buscar por dirección de correo electrónico, número de teléfono o UID de usuario' and an 'Agregar usuario' button. Below is a table of users with the following data:

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
mariamaria@gmail.com	✉	8 jun 2023	8 jun 2023	sEGTR6MCUZQV0dS1w8ApnWFX...
lauhoquenefe-5490@yopm...	✉	4 jun 2023	8 jun 2023	7POQHfRC84MrhOlwu8n7UpwCP...
pretroigibeikei-5296@yop...	✉	3 jun 2023	8 jun 2023	wKs5xUd7xiTezdxRglCzIULtGTy2
ducroitroffauffou-3260@yo...	✉	31 may 2023	6 jun 2023	z4lIZ8Mgn0cKvim1WsaRinGLPFU2
trigrawecrehoi-1464@yop...	✉	28 may 2023	9 jun 2023	UFS9CKLKapXC1VdnReSYEdiVCa...

At the bottom of the table, there is a pagination control showing 'Filas por página: 50' and '1 - 5 of 5'.

Figura 22. Consola de Firestore Authentication. Font: Elaboració pròpia

Firestore Cloud Storage

Per inicialitzar Cloud Storage el procediment és el mateix que amb els serveis previs. Es tria la opció a la pantalla de serveis de Firestore i després condueix a una curta passarel·la on demana concretar les regles de seguretat de la base de dades i la localització d'aquesta. En

aquest cas la ubicació és i serà la mateixa per la que es configuri Firebase Firestore, *eu-west* pel cas d'aquest projecte.

Finalment, és necessari executar `#flutter pub add firebase_storage` per poder accedir a les funcions de Cloud Storage.

A continuació es pot veure la consola de Firebase Storage, unes carpetes que contenen les imatges de l'aplicació.

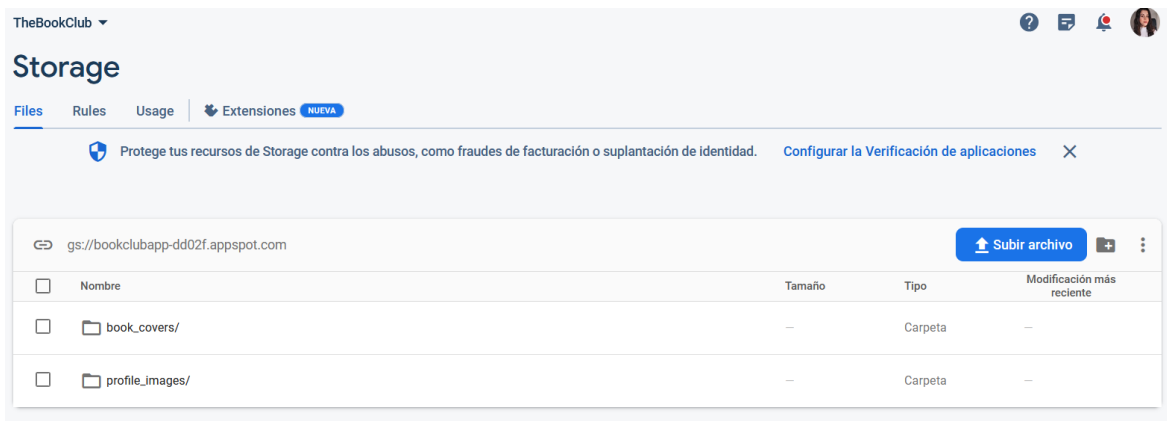


Figura 23. Consola de Firebase Cloud Storage. Font: Elaboració pròpia

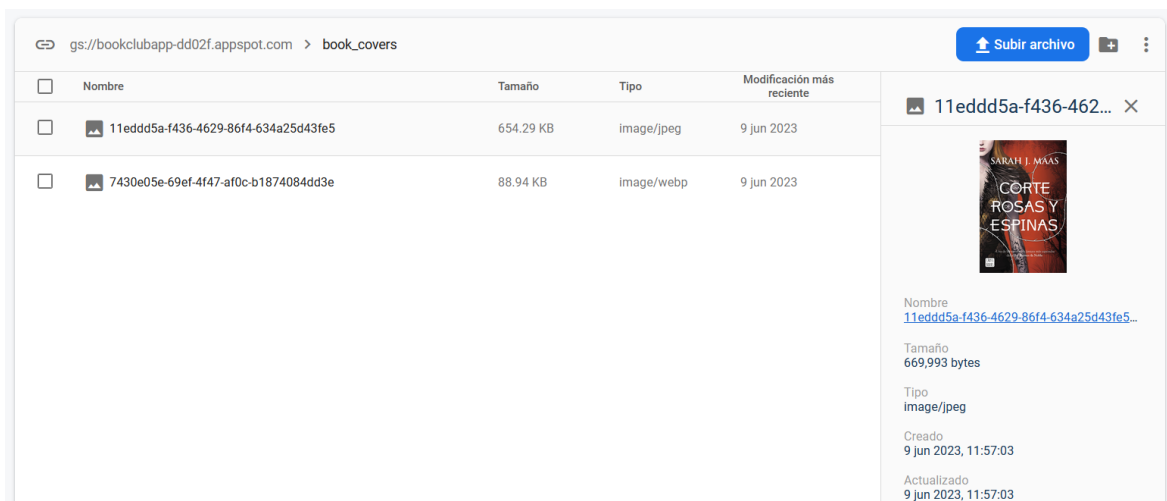


Figura 24. Contingut d'una carpeta de Firebase Cloud Storage. Font: Elaboració pròpia

Firestore

L'últim element requerit pel projecte és Firestore, necessari per donar suport al xat. Per poder-ne fer ús es segueix el mateix camí que amb les altres bases de dades, es tria la opció al catàleg de serveis, s'especifiquen les normes d'escriptura i lectura i, en cas d'haver

configurat prèviament Cloud Storage, ja no cal fer més configuració addicional ja que la base de dades s'ubica a la mateixa localització.

Per incloure la llibreria al projecte s'executa `#flutter pub add cloud_firestore` i ja es pot començar a treballar amb la base de dades.

La captura a continuació mostra el portal de Cloud Firestore:

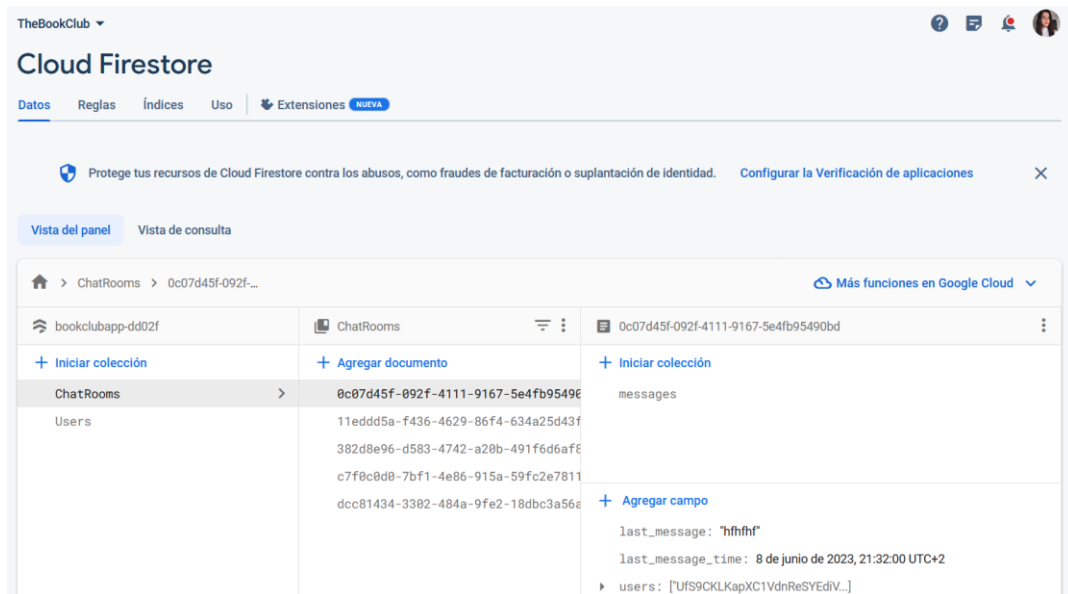


Figura 25. Consola de Firebase Cloud Firestore. Font: Elaboració pròpia

6.3 Navegació

Navegació de l'aplicació

La següent figura mostra l'esquema de navegació de l'aplicació mòbil.

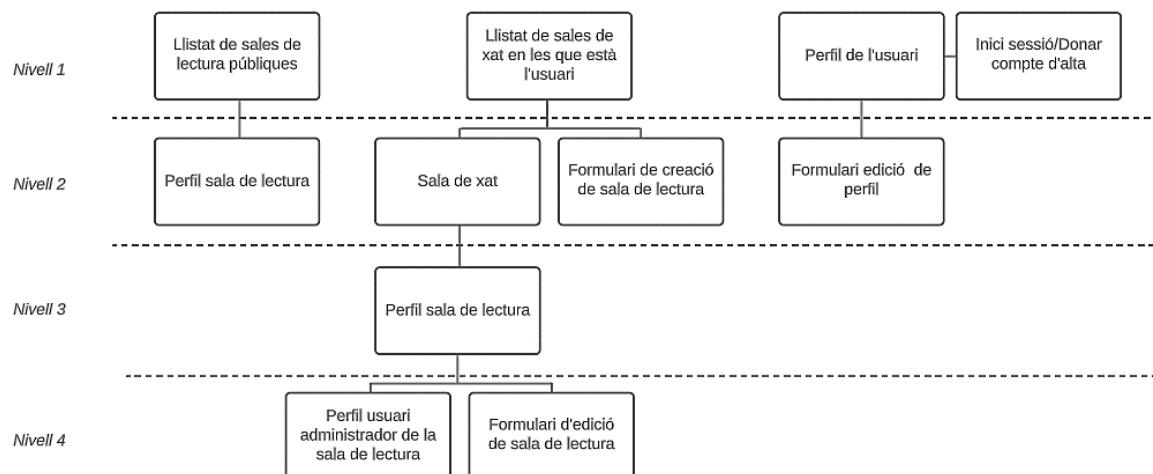


Figura 26. Esquema de navegació de l'aplicació mòbil. Font: Elaboració pròpia

El primer nivell representa per la barra de navegació inferior de l'aplicació. Cada element del primer nivell és el contingut de la pantalla que es mostra al prémer el botó corresponent.

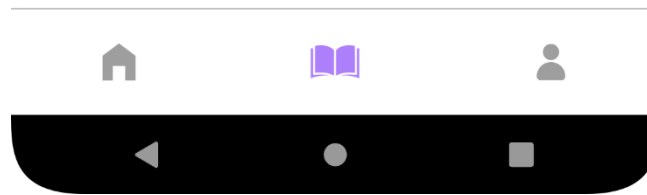


Figura 27. Barra inferior de navegació de l'aplicació. Font: Elaboració pròpia

La resta de nivells són les possibles sortides que hi ha a cada pantalla. Es pot canviar de pantalla a través d'un botó o un element ampliable, com per exemple l'entrada d'una llista.

A Flutter tots els objectes són *widgets*. Es pot entendre un *widget* com un objecte amb unes funcionalitats preestablertes per fer el desenvolupament més senzill. Tot i que hi ha moltes solucions ja implementades, en cas de voler un sistema més personalitzat cal buscar solucions alternatives, i aquest és el cas del sistema de navegació de l'aplicació.

Navigator^[35] és un objecte que manté un historial de *widgets* en una pila, pel que a través de les accions *pop* i *push*, es carreguen o treuen pantalles de l'arbre general. El problema és que, degut a que hi ha una sola pila, l'estat de les seccions globals de l'aplicació (nivell 1 de la Figura 26) es perd cada vegada quan es fa un canvi de secció. Per exemple, si l'usuari es troba llegint el perfil d'una sala de lectura, decideix anar a la secció de perfil de l'usuari, i després torna a la secció on es trobava, tornarà a carregar el llistat de sales de lectura i haurà perdut el perfil de la sala de lectura que estava llegint.

Així doncs, per solucionar aquest inconvenient es munta una pila (Stack) d'objectes *OffStage Navigator*. *Offstage*^[36] és un tipus de *widget* que té una propietat booleana la qual fa que es mostri, o no, l'element en l'arbre d'objectes. Això vol dir que hi ha tres Navigators, un sobre l'altre, només que només un d'ells és visible segons la secció de l'aplicació on es trobi l'usuari. Aquesta solució comporta que no es perd l'estat de les seccions, però existeix l'inconvenient de l'impacte addicional en el rendiment de l'aplicació ja que cada cop que es canvia de pestanya es tornen a renderitzar les tres seccions.

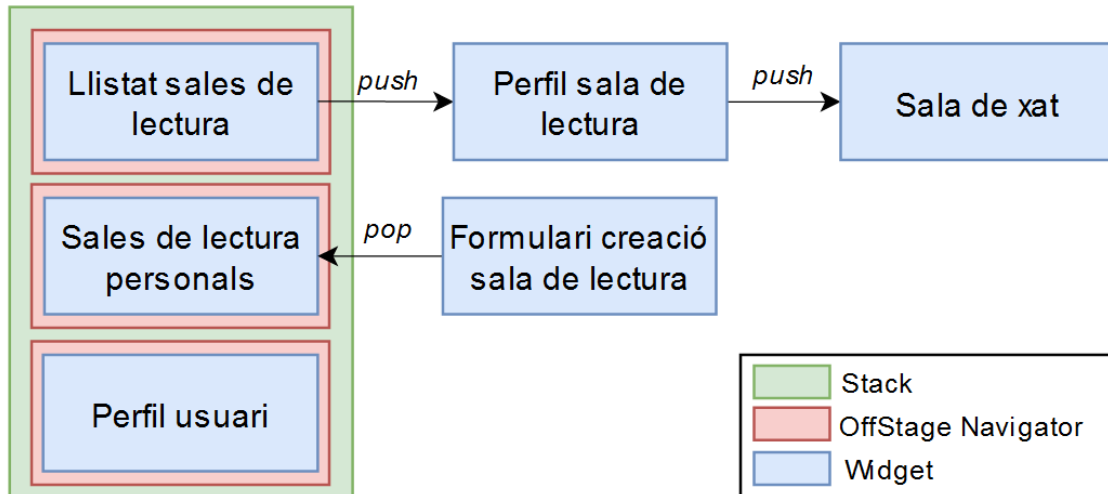


Figura 28. Estructura d'objectes del navegador de l'aplicació. Font: Elaboració pròpia

Navegació formulari de creació de sala de lectura

Per accedir al formulari de creació d'una sala de lectura es fa a través de un `FloatingActionButton`^[37], *widget* propi de Flutter, ubicat a la secció de sales de xat dels usuaris.

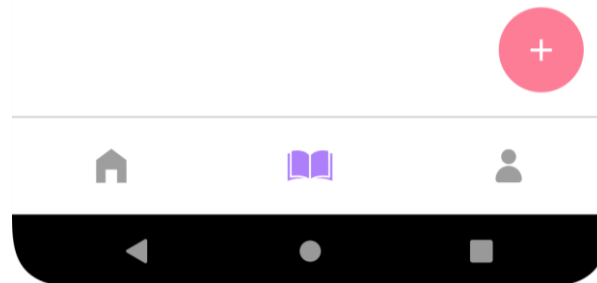


Figura 29. Botó de redirecció a la creació de sales de lectura. Font: Elaboració pròpia

Per crear una sala de lectura, cal concretar els detalls del llibre i de la pròpia lectura conjunta. Per millorar l'experiència d'usuari s'ha dividit en dues pàgines diferents. L'eina utilitzada ha estat un `Stepper`^[38], un *widget* propi de Flutter que mostra el progrés, en aquest cas, d'un formulari, a través d'una seqüència de passos.

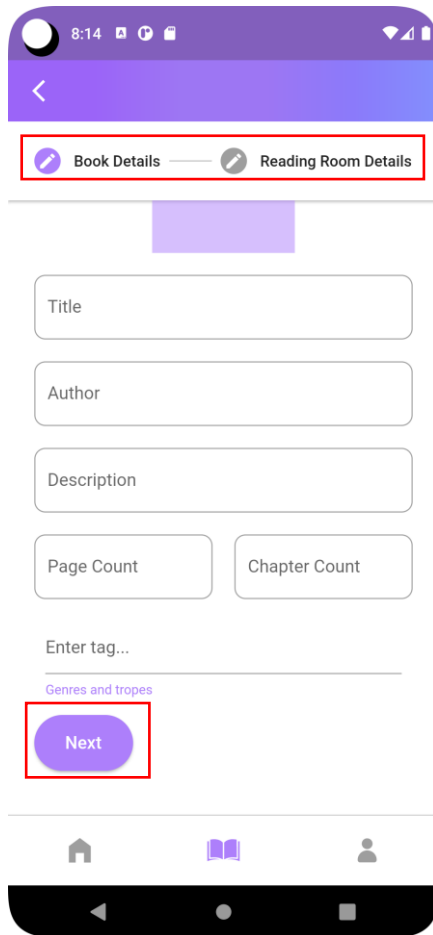


Figura 30. Stepper 1. Font: Elaboració pròpia

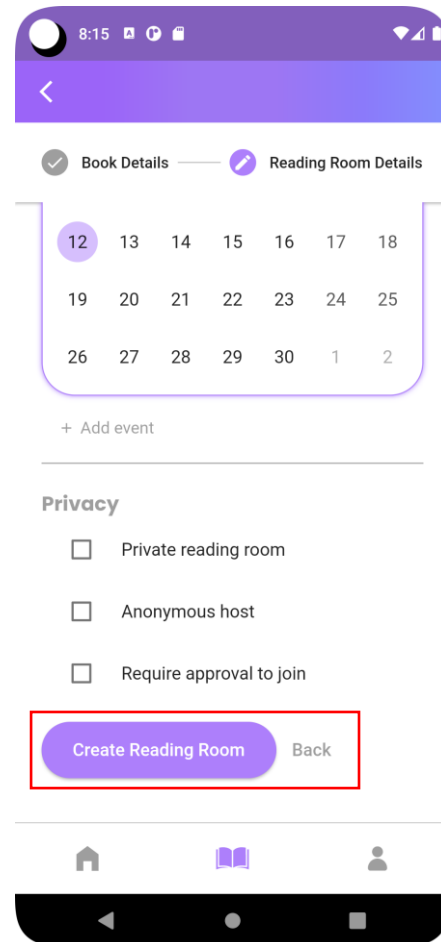


Figura 31. Stepper 2. Font: Elaboració pròpia

Té dos modes, vertical o horitzontal. Per a la implementació s'ha utilitzat el format horitzontal. El *widget* també proveeix d'altres eines com el botó de Next, Back o Submit, a part de poder canviar de pàgina a través de les icones que es mostren a la part superior.

6.4 Llibres

Els llibres son un element que van molt de la mà amb les lectures conjuntes, fins el punt que pot ser complicat desacoblar-los del desenvolupament amb les sales. Segons el disseny plantejat, tant la informació dels llibres com de la sala de lectura es troba a la mateixa pàgina. La Figura 34 mostra el disseny plantejat per a la interfície d'usuari, creat amb Adobe Photoshop. Es veu com es mostra el títol del llibre, autor, nombre de pàgines, descripció, gèneres i tòpics i, finalment, portada i puntuació. A la Figura 32 es mostra el resultat de la codificació final de la interfície en Flutter.

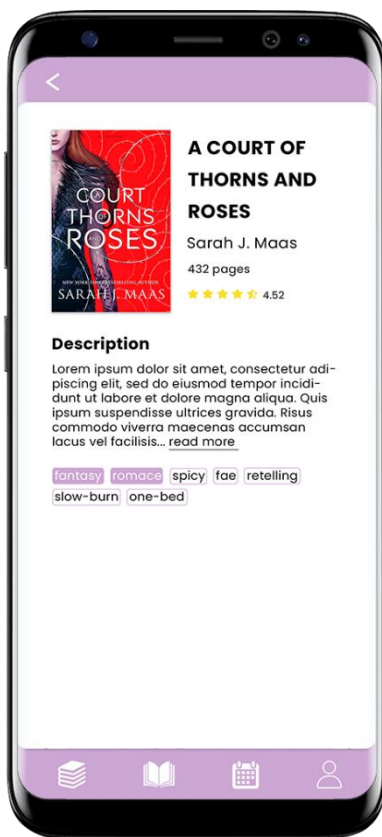


Figura 32. Disseny del perfil de llibres. Font: Elaboració pròpia

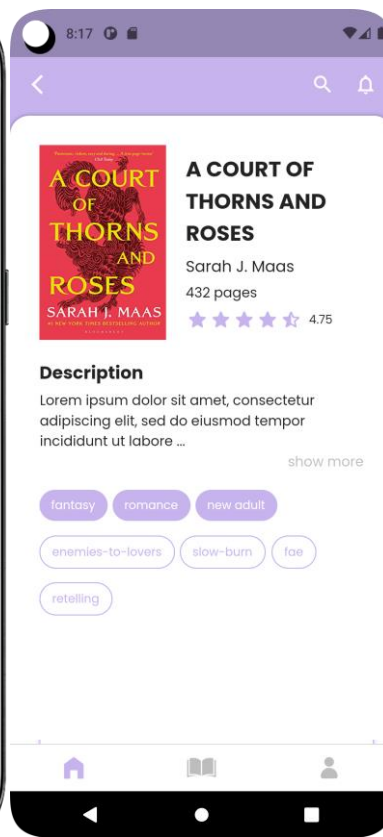


Figura 34. Implementació inicial del perfil de llibres. Font: Elaboració pròpia

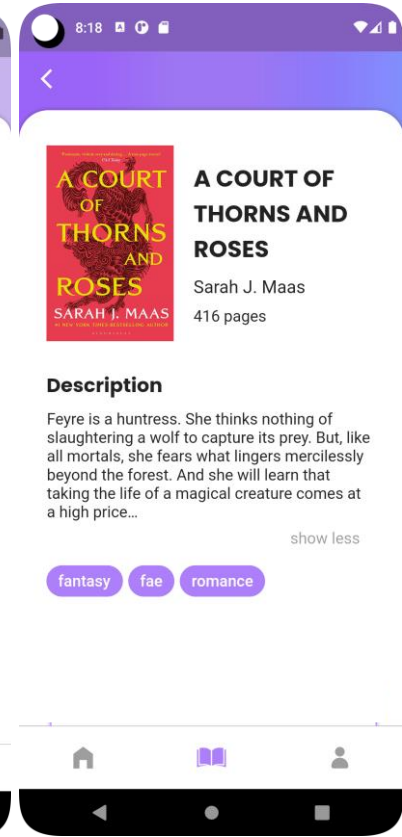


Figura 33. Implementació final del perfil de llibres. Font: Elaboració pròpia

Tot i que a les Figures 34 i 33 es mostra com el llibre té una puntuació, s'ha valorat al llarg de les iteracions i les seves pròpies reavaluacions, que es tracta d'un element addicional no necessari per a l'entrega del MVP, fet que es veu reflectit a la Figura 32.

La Figura 36 mostra la pàgina de detalls del llibre per a la creació d'una sala de lectura conjunta, que es compon de portada, títol, autor, descripció del llibre, nombre de pàgines i capítols i llistat de gèneres. Flutter proveeix d'un *widget* anomenat Form^[39] el qual proporciona, entre d'altres funcionalitats, un validador. Cada camp en conté un de propi que, quan es prem el botó per desar, verifica que el valor introduït compleix les condicions adients, en aquest cas omplir els camps obligatoris. En cas d'error es ressalten els camps pertinents, tal com es mostra a la Figura 35.

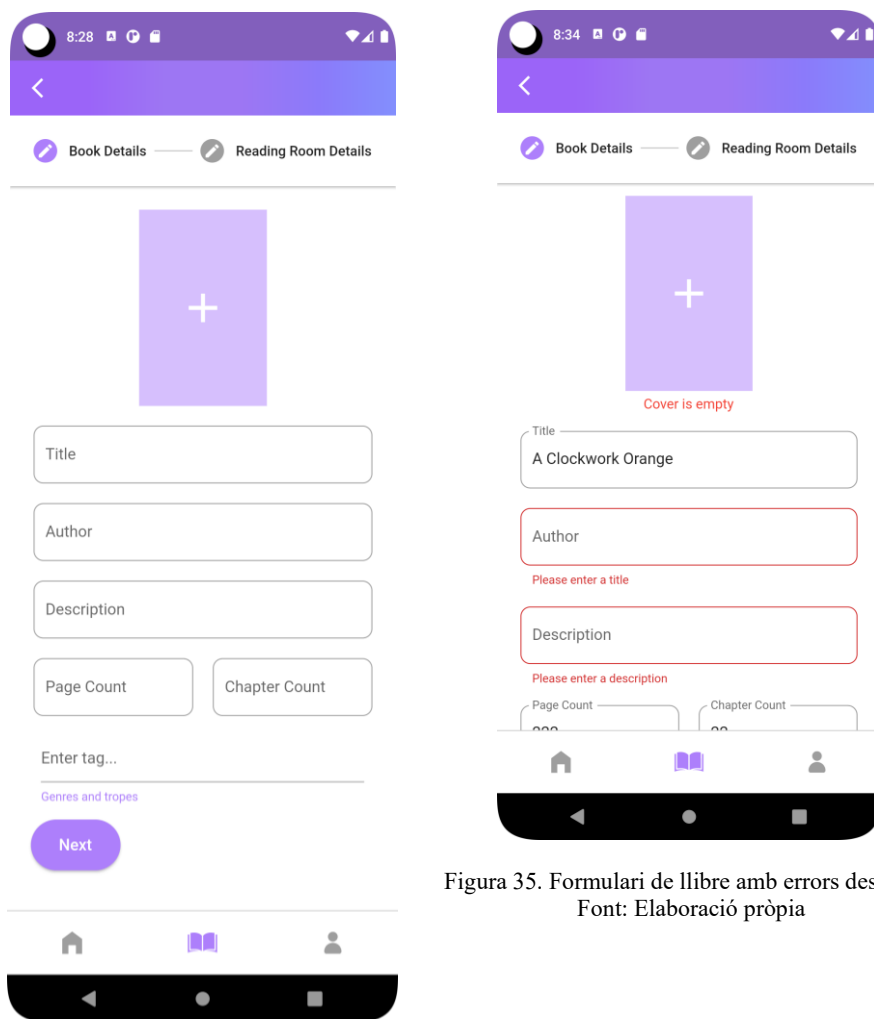


Figura 35. Formulari de llibre amb errors destacats.
Font: Elaboració pròpia

Figura 36. Formulari ampliat de creació de llibre. Font: Elaboració pròpia

Per permetre als usuaris pujar imatges a l'aplicació cal el *plugin* `image_picker`^[40]. Aquest *plugin* ja incorpora la funcionalitat d'obrir la galeria de l'usuari i carregar una o més imatges a l'aplicació. El format de pujada de les imatges és `XFile`, el qual cal transformar a `File` per poder penjar la imatge a Cloud Storage.

Hi ha una categoria destinada als gèneres i tòpics. Cada pestanya que es veu s'anomena Chip i és un *widget* ja proveït per Flutter.

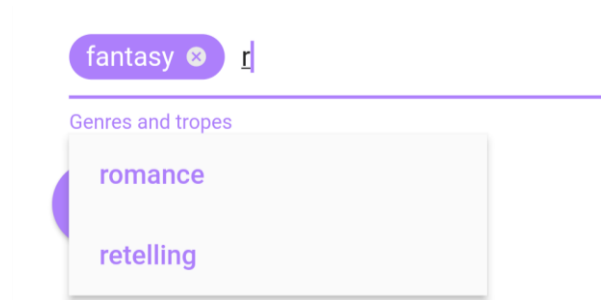


Figura 37. *Widget* d'autocompletar gèneres. Font: Elaboració pròpia

S'ha implementat un petit sistema de recomanació del qual, a partir del text introduït, es fan recomanacions per poder autocompletar prement la opció premuda.

Quan es desa un formulari vàlid, primer s'emmagatzema la portada a Cloud Storage amb el identificador de sala de lectura com a ruta. La crida de desa retorna el *link* de la ubicació de la imatge i aquesta s'afegeix al model de dades del llibre.

```
Future<String> uploadBookCover(String readingRoomId, File cover) async {
  return await storageRef
    .child('book_covers/$readingRoomId')
    .putFile(cover)
    .then((value) {
      return value.ref.getDownloadURL();
    });
}
```

Figura 38. Codi d'escriptura de sala de lectura a Cloud Storage. Font: Elaboració pròpia

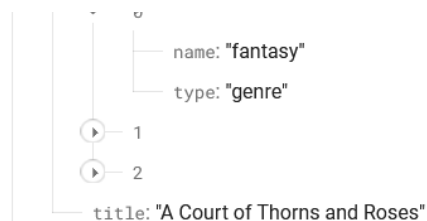


Figura 39. Objecte llibre a la base de dades. Font: Elaboració pròpia

Finalment s'executa el mètode que introdueix les dades a Firebase Realtime Database. La Figura 35 mostra un exemple d'un llibre emmagatzemat a la base de dades.

6.5 Sales de Lectura

Les sales de lectura son el pilar de l'aplicació. Hi ha tres elements principals que conformen dit sistema: el perfil de cada sala, el formulari de creació d'aquestes i el xat. El xat es tracta en un apartat diferent degut a la rellevància individual de la funcionalitat.

Formulari de creació de sales de lectura

El formulari per crear les sales està conformat per dues pàgines: llibre i detalls de sala. Els detalls de la sala venen conformats per un text presentació, un calendari per introduir els dies clau de la lectura conjunta i diversos elements de privacitat els quals no es troben implementats.

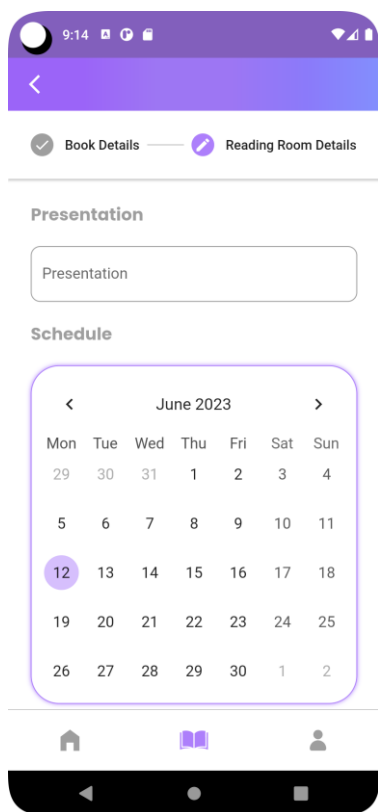


Figura 40. Implementació del formulari de les sales de lectura 1.
Font: Elaboració pròpia

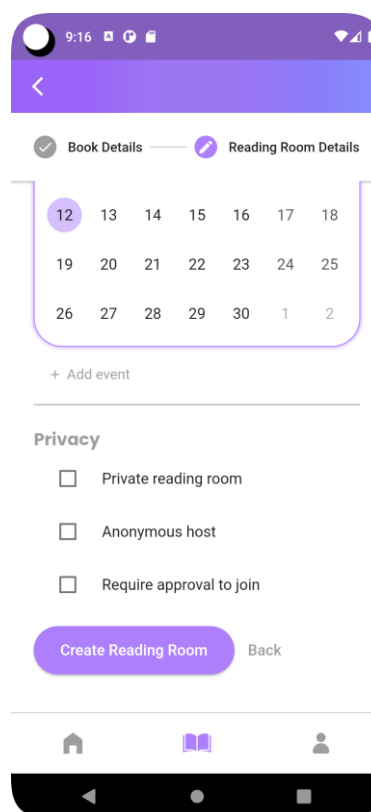


Figura 41. Implementació del formulari de les sales de lectura 2. Font: Elaboració pròpia

El calendari que conforma l'itinerari està creat amb el *widget* TableCalendar^[41]. La classe crea l'estructura del calendari i permet allotjar esdeveniments. El formulari de creació d'esdeveniments és d'elaboració pròpia i permet introduir un títol, seleccionar el tipus de d'esdeveniment corresponent (data d'inici, data final i debat intermedi) i triar una hora amb un *widget* de tria d'hora. Els esdeveniments s'emmagatzemen dins de l'itinerari de la sala de lectura.

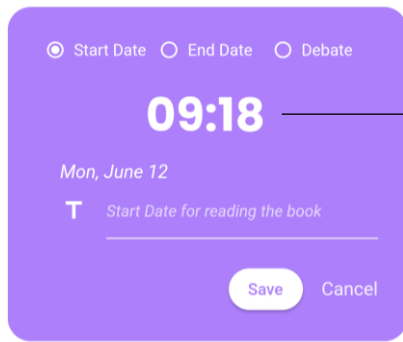


Figura 42. Formulari de creació d'esdeveniment. Font: Elaboració pròpia

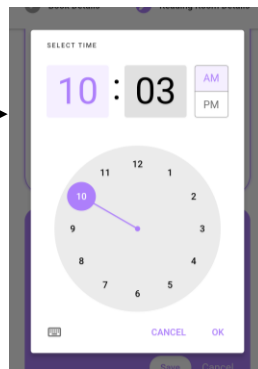


Figura 44. Tria d'hora. Font: Elaboració pròpia

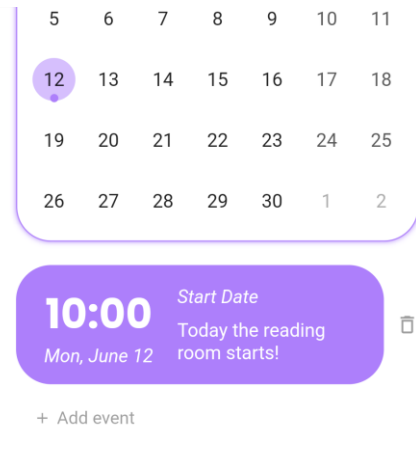


Figura 43. Esdeveniment creat a la sala de lectura. Font: Elaboració pròpia

La sala de lectura segueix el mateix model que la validació de llibres. Quan es vol enviar el formulari s'activa el validador i verifica que tots els camps compleixin les condicions adients. En cas d'haver-hi algun error automàticament apareix un missatge i canvia el camp a color vermell.

Si el formulari és vàlid s'executa un mètode que persisteix les dades introduïdes. Dit mètode invoca una referència de Firebase Realtime Database i crea un nou fill amb el identificador únic de la sala de lectura com a clau de manera asíncrona, el que vol dir que altres mètodes segueixen poder-se executar sense necessitat que acabi l'operació d'escriptura. Quan el procediment acaba s'inscriu a l'usuari creador en la sala de lectura per practicitat a l'hora de fer operacions de lectura, cal recordar que el sistema de *queries* d'aquesta base de dades és limitat. Un cop inscrita la sala de lectura a la base de dades es crea la sala de xat a Firestore amb l'usuari creador com a únic membre.

En la figura 45, es veu com es passa per paràmetre `readingRoom`, que és un mapa de *strings* on es troben les dades a persistir. Realtime Database és una base de dades NoSql i, per tant, tot el contingut d'aquesta està estructurada en un gran arbre d'objectes JSON.

```
Future createReadingRoom(Map<String, dynamic> readingRoom, String
readingRoomId, String userId) async {
  await ref.child("reading-room/$readingRoomId ")
    .set(readingRoom).then((value) async {
      await addUserToReadingRoom(readingRoomId, userId);
    });}
```

Figura 45. Codi d'escriptura de sala de lectura a Realtime Database. Font: Elaboració pròpia

```
Future addUserToReadingRoom(String readingRoomId, String userId) async {
  await ref.child("reading-room/$readingRoomId/members").push().set(userId);
}
```

Figura 48. Codi d'addició de membre a sala de lectura a Realtime Database. Font: Elaboració pròpia

```
Future createChatRoom(String readingRoomId, String hostId) async {
  await firestore.collection('ChatRooms').doc(readingRoomId).set({
    'users': [hostId],
  });
}
```

Figura 47. Codi de creació de sala de xat a Firebase Firestore. Font: Elaboració pròpia

El mètode `.toJson()` no ve heretat pel que cal implementar-lo en cada element del domini que requereixi de persistència.

```
Map<String, dynamic> toJson({forUpdate = false}) {
  return {
    'book': book.toJson(),
    'schedule': schedule.toJson(),
    if (!forUpdate) 'hostId': hostId,
    if (!forUpdate) 'members': members,
    if (!forUpdate) 'creationDate': creationDate.toString(),
    'description': presentation,
    'isPrivate': isPrivate,
    'isAnonymous': isAnonymous,
    'requireApproval': requireApproval,
  };
}
```

Figura 46. Codi de conversió d'objecte a JSON. Font: Elaboració pròpia

La captura a continuació mostra com una sala de lectura a la base de dades:

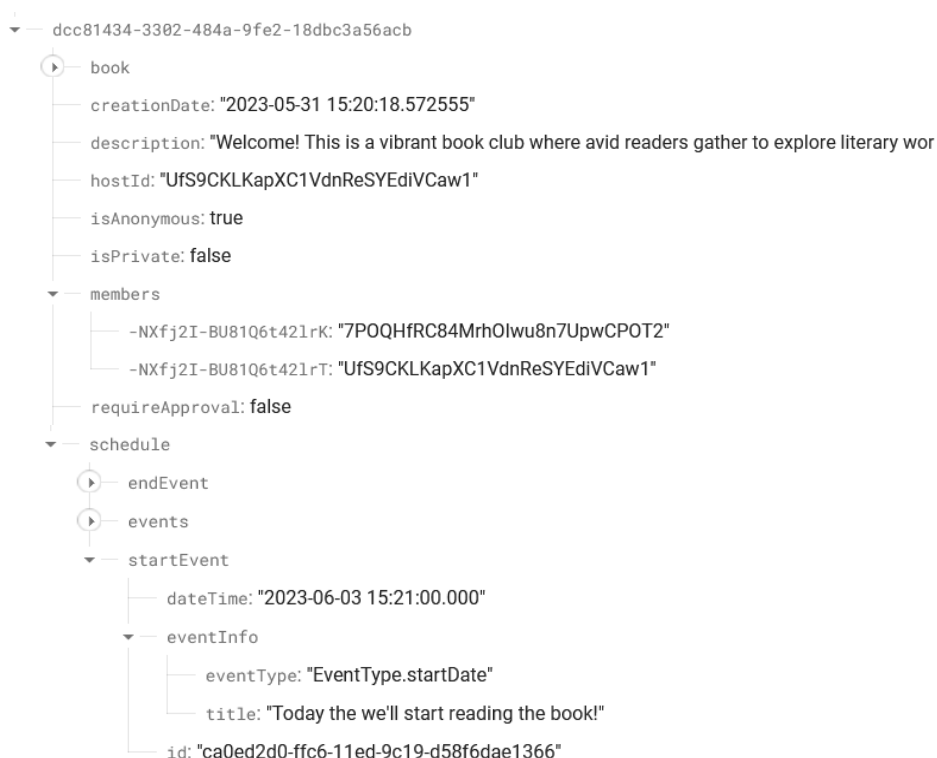


Figura 49. Objecte sala de lectura a la base de dades Font: Elaboració pròpia

Perfil sala de lectura

El desenvolupament d'aquest component comença pel *frontend*. Degut que es tracta d'una aplicació mòbil, l'experiència d'usuari és un dels factors més importants del projecte^[42]. Seguir aquest flux ajuda a identificar les funcionalitats més importants i prioritzar-les en funció de les necessitats de l'usuari. També cal tenir en compte com s'accelera el procés de prototipatge i iteració de la interfície i experiència d'usuari, el que ajuda a perfeccionar el disseny i la funcionalitat de l'aplicació abans d'invertir temps i recursos en el *backend*.

Abans de començar a programar es fa el disseny de la UI per tenir una guia del que es vol implementar. Pel disseny de pantalles s'ha utilitzat l'eina Adobe Photoshop.

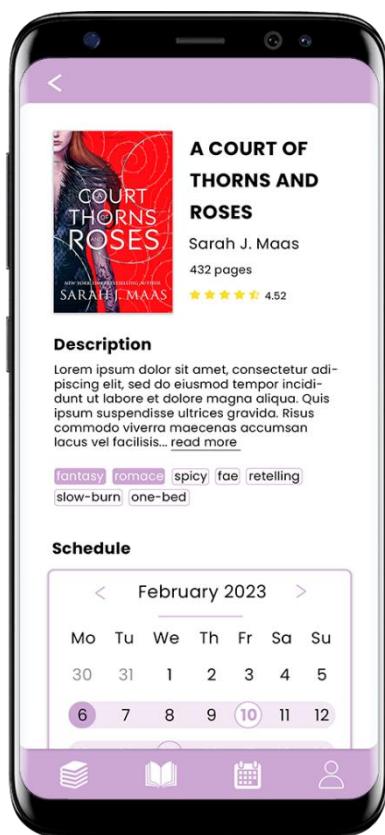


Figura 52. Disseny del perfil de les sales de lectura 1. Font: Elaboració pròpia

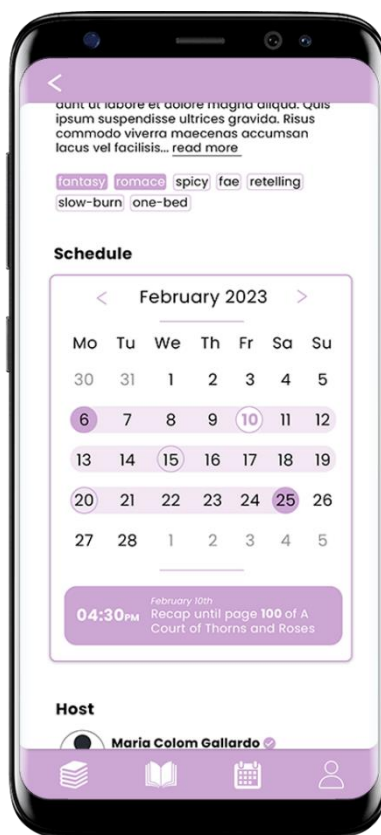


Figura 51. Disseny del perfil de les sales de lectura 2. Font: Elaboració pròpia

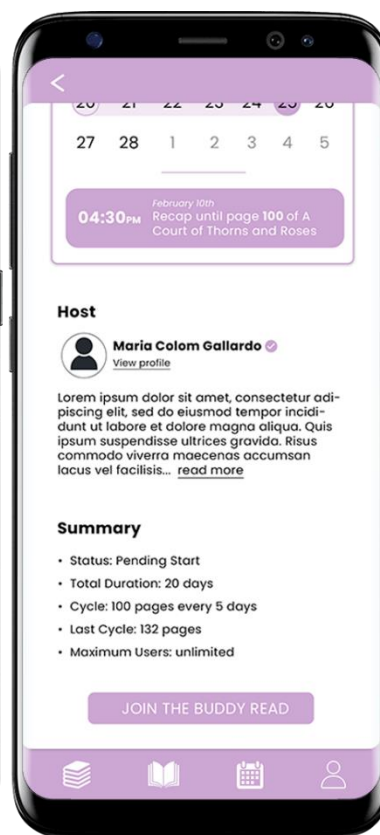


Figura 50. Disseny del perfil de les sales de lectura 3. Font: Elaboració pròpia

Com es pot veure a les imatges del disseny, el perfil consisteix en quatre parts: informació del llibre, itinerari de la lectura conjunta, informació de l'usuari organitzador i una secció de resum o informació addicional. Les figures a continuació mostren el resultat de la construcció inicial i final en Flutter.

Tal com s'indica a la secció 6.3, als perfils s'hi accedeix a través d'una llista de la pàgina principal, la qual recull totes les sales de lectura de l'aplicació o a través de les sales de xat.

A l'apartat de "Host" es mostra la imatge de perfil del creador de la sala de lectura i el seu nom d'usuari. El text a continuació és la presentació de la sala de lectura i, a més, si es prem "View profile" redirigeix al perfil de l'usuari creador.

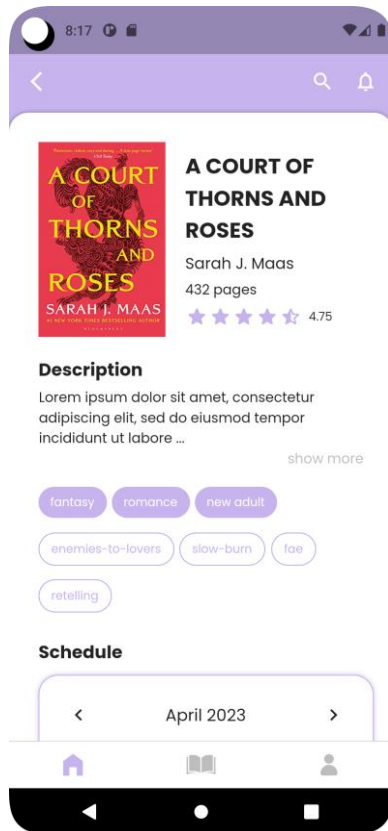


Figura 53. Implementació inicial del perfil de les sales de lectura 1. Font: Elaboració pròpia

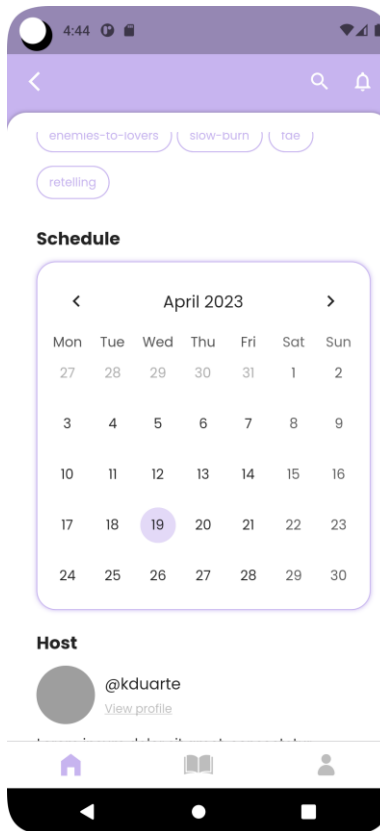


Figura 54. Implementació inicial del perfil de les sales de lectura 2. Font: Elaboració pròpia

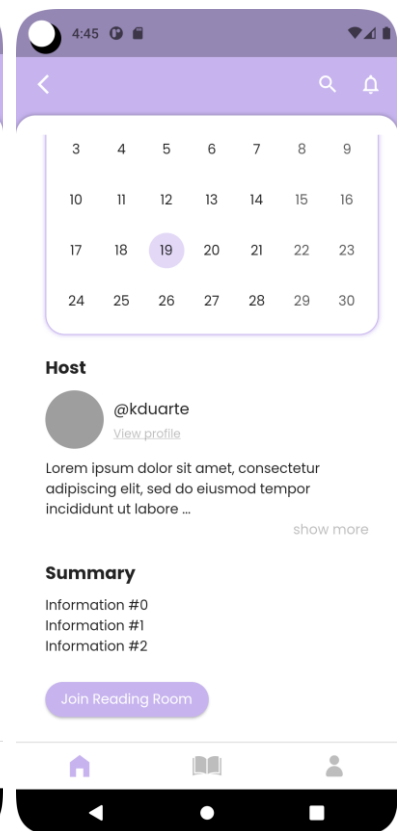


Figura 55. Implementació inicial del perfil de les sales de lectura 2. Font: Elaboració pròpia

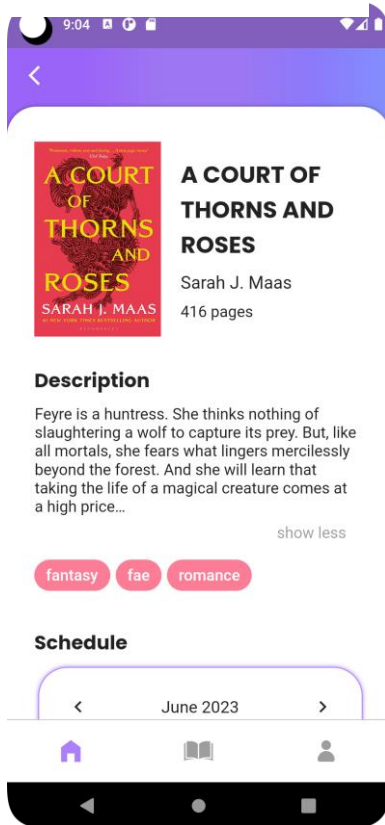


Figura 56. Implementació final del perfil de les sales de lectura 1. Font: Elaboració pròpia

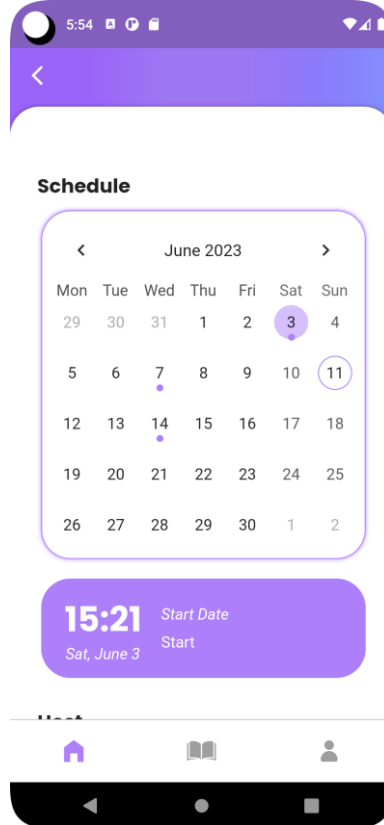


Figura 57. Implementació final del perfil de les sales de lectura 2. Font: Elaboració pròpia

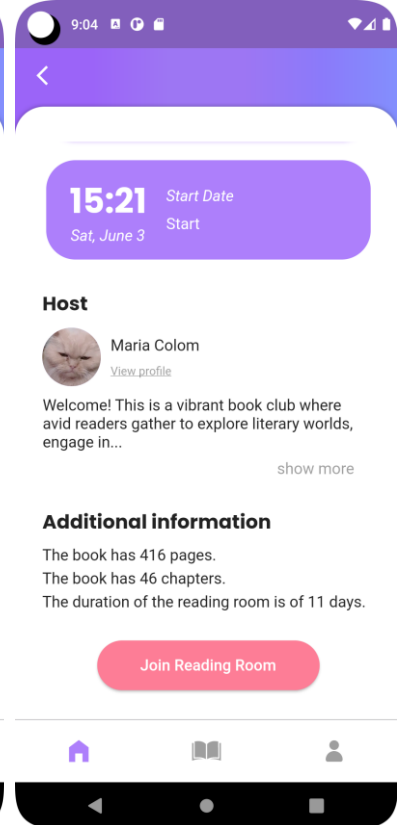


Figura 58. Implementació final del perfil de les sales de lectura 3. Font: Elaboració pròpia

Depenent de l'estat de l'usuari respecte la seva sessió o la seva participació a una sala de lectura, es mostren les següents diferents opcions al final del perfil de la sala de lectura.

- Usuari sense iniciar sessió

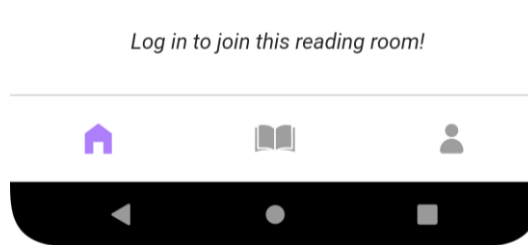


Figura 59. Visualització d'accions a una sala de lectura d'un usuari sense iniciar sessió. Font: Elaboració pròpia

- Usuari identificat a través del llistat de sales de lectura públiques

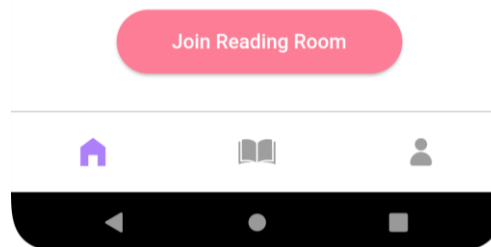


Figura 60. Visualització d'accions a una sala de lectura d'un usuari identificat a través del llistat de sales de lectura públiques. Font: Elaboració pròpia

- Usuari identificat membre d'una sala de lectura

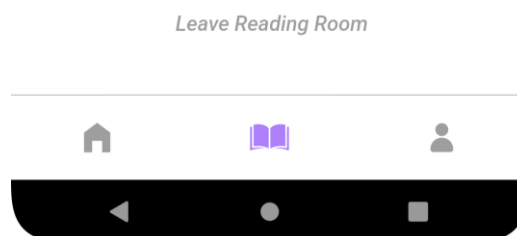


Figura 61. Visualització d'accions a una sala de lectura d'un usuari identificat membre d'una sala. Font: Elaboració pròpia

- Usuari identificat propietari d'una sala de lectura

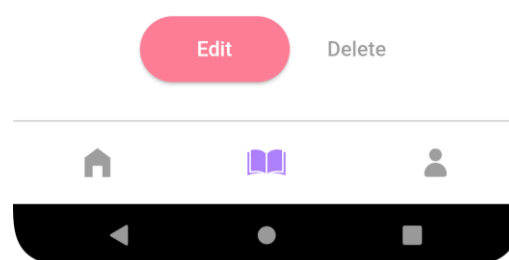


Figura 62. Visualització d'accions a una sala de lectura d'un usuari identificat propietari de sala. Font: Elaboració pròpia

L'usuari propietari de la sala de lectura pot editar en qualsevol moment la sala de lectura. Quan es prem el botó editar es redirigeix al mateix formulari de creació de sala de lectura però amb els camps existents coberts.

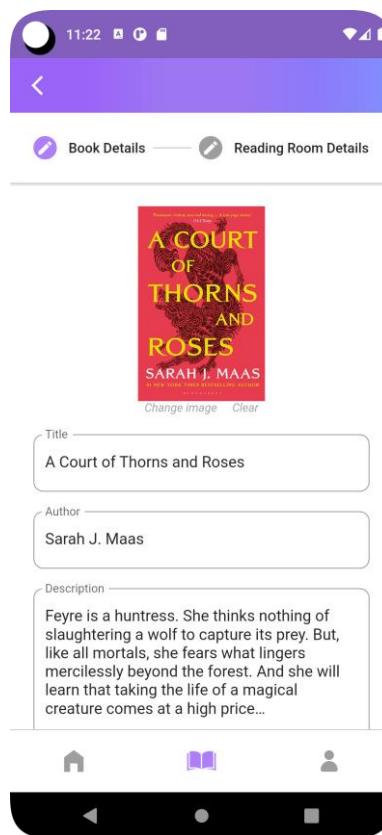


Figura 63. Formulari d'edició de sala de lectura. Font: Elaboració pròpia

No es pot eliminar una sala si ja hi ha membres. En cas d'intentar-ho apareix una notificació d'avís durant dos segons.

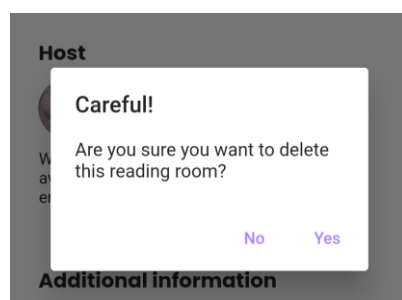


Figura 64 Alerta d'esborrada de sala de lectura. Font: Elaboració pròpia

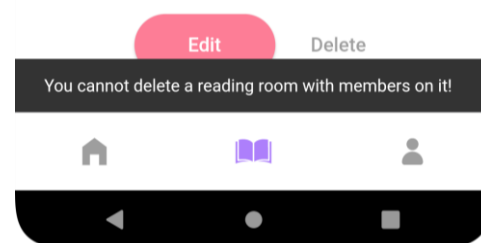


Figura 65. Notificació d'avís d'impossibilitat d'esborrada. Font: Elaboració pròpia

Llistat de sales de lectura públiques

Una sala de lectura pública és una sala de lectura en la qual l'usuari identificat no hi participa. Quan l'usuari s'inscriu a una sala de lectura, aquesta ja no apareix més al llistat general.



Figura 67. Sales de lectura públiques Font: Elaboració pròpia

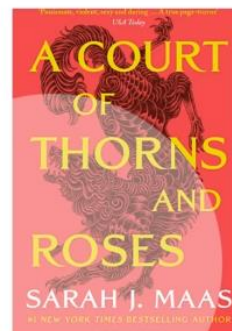


Figura 66. Efecte *splash* de les portades. Font: Elaboració pròpia

El llistat està conformat per les portades dels llibres al que es dedica cada sala de lectura. Quan es prem una de les portades, es mostra un efecte *splash* i redirigeix directament al perfil de la sala de lectura.

6.6 Autenticació

Quan s'inicia l'aplicació per primer cop i es prem la finestra de perfil, es mostren dues opcions, iniciar sessió o crear un compte. Gràcies a Firebase Authentication l'enregistrament i validació en el sistema es molt senzill.

Per donar d'alta un compte cal completar els camps que es mostren a la Figura 69, essent obligatoris el nom d'usuari, el correu electrònic i la contrasenya. Tal com els altres formularis de l'aplicació, aquest també conté validadors en cada camp.

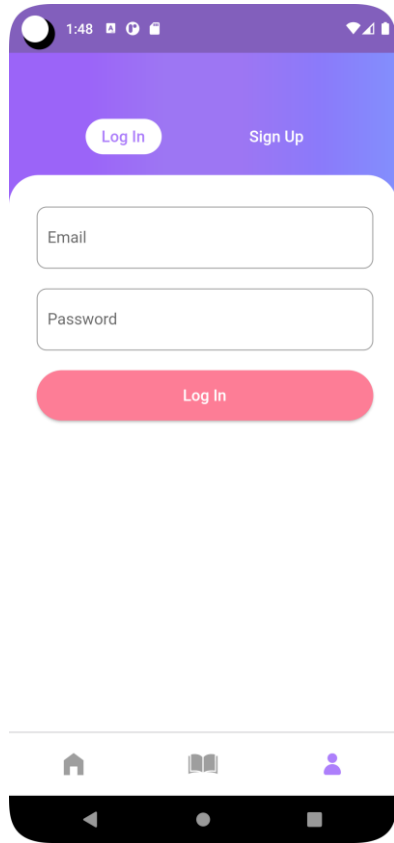


Figura 68. Pantalla per iniciar sessió a l'aplicació. Font: Elaboració pròpia

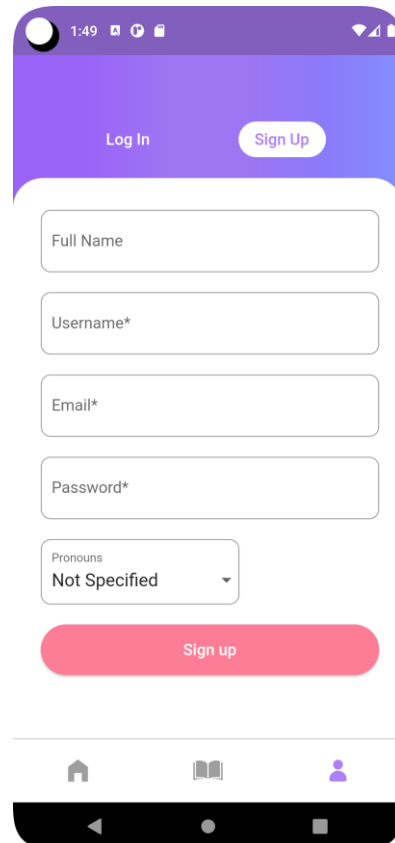


Figura 69. Pantalla per crear un compte a l'aplicació. Font: Elaboració pròpia

Quan es prem “Sign Up” primer es dona d'alta l'usuari a Firebase Authentication amb el correu electrònic i la contrasenya i, un cop finalitzat el procés amb èxit, es redirigeix a una pàgina on demana validar el correu electrònic. Aquesta pàgina té un temporitzador que cada tres segons revisa si l'usuari ha validat el correu prement l'enllaç que se li ha enviat. La

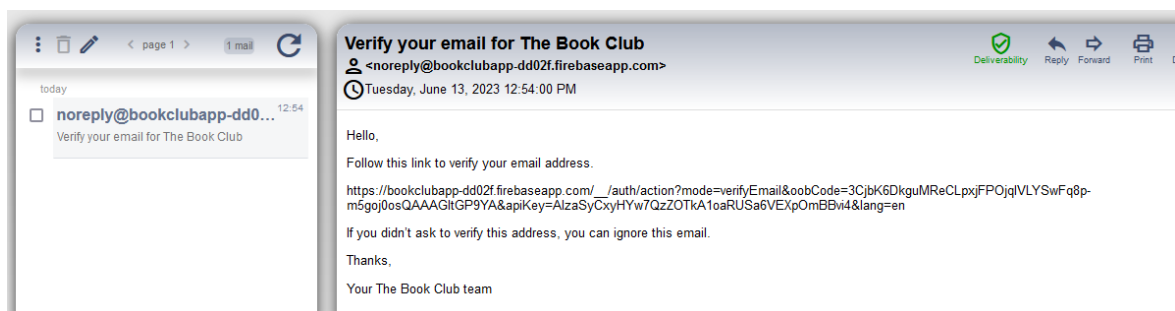


Figura 70. Correu que s'envia als usuaris per validar la seva identitat. Font: Elaboració pròpia.

variable proporcionada per Firebase Authentication `_auth.currentUser!.emailVerified` s'actualitza a `true` quan l'usuari es valida. Un cop l'usuari queda validat se'l redirigeix al seu perfil.

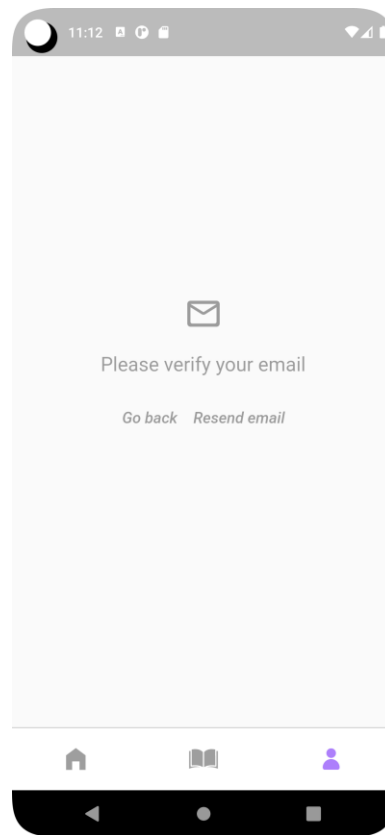


Figura 71. Pantalla de validació de correu electrònic. Font: Elaboració pròpia

Quan es dona d'alta un usuari amb Firebase Authentication es retorna un identificador únic, que és el que s'utilitza com a referència per emmagatzemar les dades addicionals de l'usuari a la base de dades Realtime Database.

```
Future<String?> signUp({required email, required password}) async {  
  try {  
    UserCredential userCredential = await _auth  
      .createUserWithEmailAndPassword(email: email, password: password);  
    return userCredential.user?.uid;  
  } on FirebaseAuthException catch (e) {  
    rethrow;  
  }  
}
```

Figura 72. Codi d'alta a Firebase Authentication. Font: Elaboració pròpia

Per iniciar sessió Firebase Authentication és l'encarregat de validar correu electrònic i contrasenya, i en cas d'error s'indica a l'usuari a través d'una notificació a la mateixa pantalla.

```
Future<User?> signIn({required email, required password}) async {  
  try {  
    UserCredential userCredential = await _auth.signInWithEmailAndPassword  
      (email: email, password: password);  
    return userCredential.user;  
  } on FirebaseAuthException {  
    rethrow;  
  }  
}
```

Figura 73. Codi d'inici de sessió a Firebase Authentication. Font: Elaboració pròpia

Firebase Authentication s'encarrega de conservar la sessió de l'usuari pel que, encara que es tanqui l'aplicació, no es tancarà la sessió de l'usuari a no ser que faci "Log Out". Per gestionar els canvis de sessió a l'aplicació s'utilitza el mètode `FirebaseAuth.instance.authStateChanges()`, el qual retorna un `Stream`^[43], un *widget* que rep esdeveniments de manera asíncrona. Així doncs, en la pantalla on es construeix el perfil, hi ha un *widget* `StreamBuilder` que escolta els esdeveniments del `Stream`, que indica canvis de sessió i mostra una pàgina o altre depenent de l'estat de la sessió.

6.7 Perfil d'usuari

El perfil de l'usuari es mostra un cop iniciada la sessió. Està compost per elements addicionals al del formulari de creació de compte com la descripció i la imatge de perfil. Aquests elements es poden editar prement el botó superior dret de la Figura 76. Aquest botó obre el formulari d'edició de perfil que es mostra en la Figura 74. És possible afegir una imatge de perfil, una descripció, i canviar els camps desitjats del perfil. El selector d'imatges funciona igual que el de la tria de portades de llibres i també emmagatzema les imatges a Firebase Cloud Storage.

A la part superior esquerra del perfil està el botó per sortir de la sessió, i aquest està lligat a Firebase Authentication, el qual provoca que quan es prem treu el perfil de la pila de pantalles per carregar la d'inici de sessió ja que s'ha tancat la sessió de l'usuari.

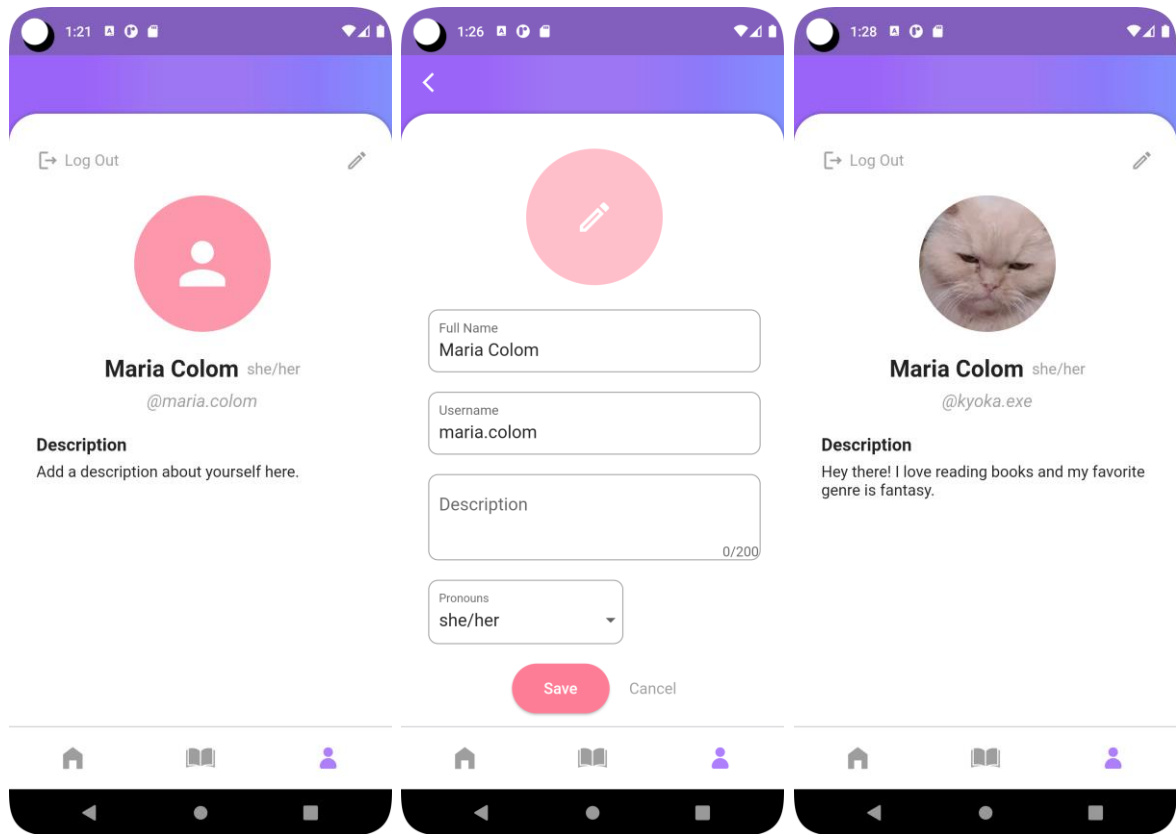


Figura 76. Perfil d'usuari amb compte nou. Font: Elaboració pròpia

Figura 74. Formulari d'edició del perfil d'usuari. Font: Elaboració pròpia

Figura 75. Perfil d'usuari amb perfil editat. Font: Elaboració pròpia

6.8 Sales de Xat

Llistat de sales de xat

Les sales de xat, o sales de lectura en les quals un usuari és membre, es troben a la pestanya d'enmig. Es mostren en forma de llistat endreçat per últim missatge rebut, amb la portada del llibre com a capçalera, el títol del llibre com a títol i l'últim missatge enviat a la sala. En cas de no haver-se enviat cap missatge, es mostra el text presentació de la sala de lectura.

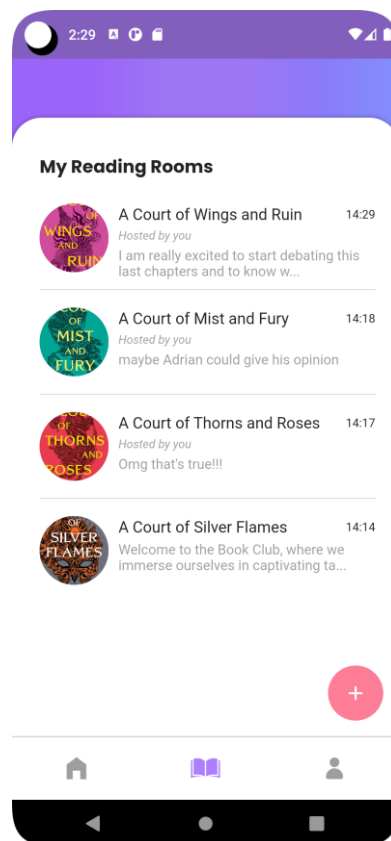


Figura 77. Llistat de sales de xat. Font: Elaboració pròpia

Per muntar la llista a partir del resultat de la base de dades s'utilitzen dos *widgets*. El primer és un *FutureBuilder*, un *widget* que es construeix basant-se en l'última instància de la interacció amb un objecte *Future*. Cal recordar que les crides a les bases de dades són asíncrones, pel que totes les tornades de dades són objectes de tipus *Future*. En aquest cas *Future* espera la resposta de *Realtime Database* per rebre el llistat de sales de lectura en les que participa l'usuari.

Dins del FutureBuilder hi ha un StreamBuilder, el *widget* que escolta un Stream i es va muntant a mesura que rep dades. El Stream escolta constantment Firebase Firestore per actuar en base als canvis que hi ha en les sales de xat.

Sales de Xat

Les sales de xat permeten enviar missatges i accedir al perfil de les sales de lectura a través de la barra superior. Cada missatge conté la imatge de perfil de l'usuari, el nom d'usuari, el missatge i l'hora en la que s'ha enviat el missatge.

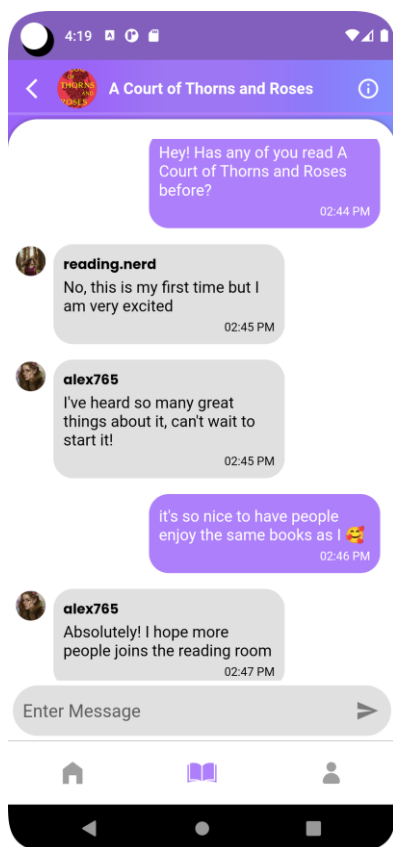


Figura 79. Sala de xat. Font: Elaboració pròpia

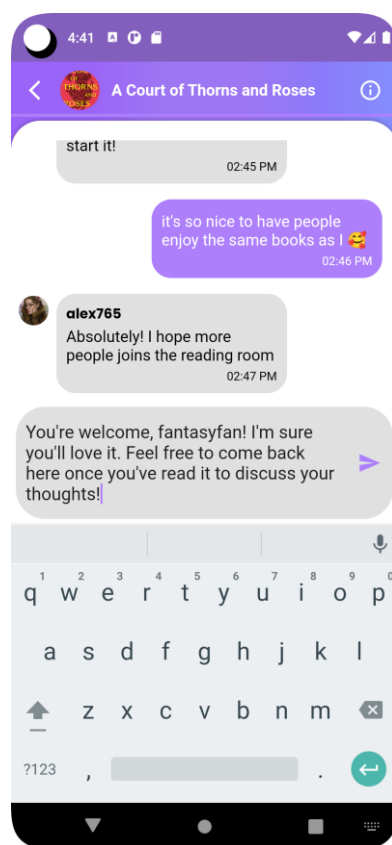


Figura 78. Sala de xat amb teclat obert. Font: Elaboració pròpia

Per construir el xat hi ha dos StreamBuilders. El primer recull la sala totes les sales de xat i selecciona la sala concreta a partir del identificador i el segon recull tots els missatges que conté la sala i els munta en un llistat de *widgets* en l'ordre invers en el que es va enviar cada missatge.

Per enviar un missatge s'instancia Firestore, s'actualitzen els camps d'últim missatge enviat i l'hora a la que s'ha enviat aquest i s'afegeix a la col·lecció de missatges el propi text enviat. Un missatge està format per text, hora d'emissió i usuari emissor.

```
void _submitMessage(Map<String, dynamic> message, String readingRoomId) {
  firestore.collection('ChatRooms').doc(readingRoomId).update({
    'last_message_time': message['datetime'],
    'last_message': message['message'],
  });
  firestore
    .collection('ChatRooms')
    .doc(readingRoomId)
    .collection('messages')
    .add(message);
}
```

Figura 80. Codi d'escriptura de missatge a Firebase Firestore. Font: Elaboració pròpia

La següent captura mostra com es veu una sala de xat a Firestore.

The screenshot displays the Firebase Firestore console interface. The breadcrumb path is: ChatRooms > dcc81434-3302-... > messages > DpJZzwMejGVA. The left pane shows a collection named 'messages' with a document containing the following data:

```
{
  last_message: "jff",
  last_message_time: "12 de junio de 2023, 17:19:29 UTC+2",
  users: [
    "Ufs9CKLKapXC1VdnReSYEdiVCaw1",
    "7POQHfRC84MrhOlwu8n7UpwCPOT2",
    "z4lIZ8Mgn0cKvim1WsaRinGLPFU2"
  ]
}
```

The right pane shows a document with the following data:

```
{
  datetime: "12 de junio de 2023, 16:45:40 UTC+2",
  message: "I've heard so many great things about it, can't wait to start it!",
  sent_by: "7POQHfRC84MrhOlwu8n7UpwCPOT2"
}
```

Figura 81. Consola de Firestore mostrant una sala de xat. Font: Elaboració pròpia

7. Anàlisi de les iteracions

7.1 Estudi de tecnologies

Planificació: L'objectiu de la primera iteració és fer la tria de les tecnologies que es fan ús al projecte.

Disseny: Per poder aquesta tria s'han perfilat els requisits tecnològics i funcionals i s'ha completat la documentació sobre les diferents alternatives llistades a l'apartat 2.4.

Desenvolupament i testeig: En aquesta iteració no hi ha desenvolupament de codi. El que més temps a consumit ha sigut la recerca i l'actualització de la documentació. La tria final de les diferents tecnologies es troba a l'apartat 5.2.

Revaluació: S'ha consumit menys temps de l'esperat per aquesta iteració pel que ara es disposa de temps addicionals per enfocar en altres objectius. Les tries fetes no contradiuen cap requisit preestablert pel que no cal modificar cap objectiu.

7.2 Configuració i model de navegació

Planificació: Els objectius d'aquesta iteració són preparar l'entorn de desenvolupament, el repositori de Github, triar l'arquitectura de l'aplicació i implementar la navegació.

Disseny: El disseny d'aquesta etapa consisteix en una exhaustiva comparació de diferents possibles arquitectures i crear el mapa de navegació de l'apartat 6.3.

Desenvolupament i testeig: L'entorn de desenvolupament s'ha de preparar amb cura però hi ha una gran quantitat de fonts a Internet que guien pas a pas en el procés. S'ha organitzat el projecte i dividit en els diferents directoris de cop per facilitar la tasca durant el desenvolupament. Finalment, pel que comporta la codificació de la navegació, ha estat una feina lenta degut a la inexperiència amb el llenguatge i *framework*. De moment no s'ha implementat cap test.

Revaluació: Gràcies al temps extra proporcionat per la iteració prèvia s'ha pogut gastar més temps en aquesta, però el fet de veure la complexitat de començar amb una tecnologia nova implica una inversió extra de temps. S'ha reevaluat el temps esperat per a la implementació de certes funcionalitats per augmentar-lo, al menys a les etapes més

primerenques del projecte. També s'ha reformulat l'ordre d'implementació. En el plantejament inicial s'havia previst que les sales de lectura serien lo primer a desenvolupar després de la navegació, però degut a l'alta complexitat d'aquestes, s'ha decidit començar pels llibres, ja que tenen menys requeriments però comparteixen processos amb les sales.

7.3 Implementació de llibres

Planificació: En aquesta iteració es dissenyen i implementen els llibres, tant la vista com el formulari de creació d'aquests.

Disseny: El disseny d'aquesta etapa ha consistit en una avaluació dels elements imprescindibles per a la creació dels llibres. Amb aquests elements ha sigut possible crear, en primer lloc, la interfície de l'usuari i, després, el model del domini. Durant la fase de disseny s'ha decidit prescindir de dos components que no es consideren imprescindibles, la imatge de la portada i la puntuació mitja del llibre (aquests son visibles al disseny inicial de l'apartat 6.4).

Desenvolupament i testeig: Degut a que els llibres son una estructura de dades més aviat senzilla, el procés de creació del perfil i el formulari s'ha vist significativament simplificat. S'ha incorporat per primer cop la base de dades a l'aplicació i les accions de lectura i escriptura. Tot i així, el desenvolupament ha estat inevitablement enrederit a causa de la inexperiència.

Revaluació: Juntament amb la revaluació i el re-disseny del model dels llibres s'ha revisat també l'estructura prevista del projecte. En aquesta fase s'ha comprès la importància de d'adherir-se al concepte de "producte mínim viable" pel que en les properes iteracions és necessari reconsiderar requisits i objectius.

7.4 Implementació de sales de lectura

Planificació: En aquesta iteració es dissenyen i implementen les sales de lectura, tant el perfil de les sales com el formulari de creació d'aquestes.

Disseny: El disseny d'aquesta etapa ha sigut similar al dels llibres. Com s'ha pogut veure en la Iteració d'implementació de llibres, la informació de la sala de lectura i el perfil del llibre es troben a la mateixa pàgina. D'igual manera, és necessària la creació del model del domini.

Aquesta iteració ha patit molts canvis de disseny, les quals queden detallades a l'estudi de la viabilitat.

Per la part de la interfície d'usuari s'ha fet un model en Adobe Photoshop de com es preveu que sigui el resultat final del perfil de les sales de lectura.

Desenvolupament i testeig: Tal com ha passat amb la iteració prèvia, el desenvolupament s'ha vist enrederit, a part de pels canvis de disseny, per la inexperiència. Aquesta iteració ha permès experimentar amb una gran quantitat d'elements propis de Flutter i crear moltes eines que seran posteriorment reutilitzables en el codi, fet que agilitzarà la codificació en etapes finals del projecte. Un dels elements que més ha dificultat la finalització d'aquesta fase ha estat el calendari i els esdeveniments, el qual ha resultat més complex del que s'esperava.

Revaluació: Aquesta iteració s'ha hagut de reavaluar en nombroses ocasions. Els detalls queden especificats a l'estudi de viabilitat. S'ha invertit més temps del previst però també ha aportat el suficient coneixement com per reduir còmodament el temps estimat en posteriors implementacions.

7.5 Navegació entre pàgines

Planificació: En la cinquena iteració es creen els enllaços entre els diferents elements i el llistat de sales de lectura junt amb el botó de creació d'aquestes.

Disseny: La complexitat d'aquesta iteració és significativament inferior a la de les seves predecessores. S'ha plantejat fer una llista vertical molt senzilla per a les sales de lectura.

Desenvolupament i testeig: Tots tres elements a desenvolupar per aquesta fase han sigut poc complicats d'implementar. En primer lloc, el botó per a la creació d'una sala de lectura és un *widget* predefinit de Flutter pel que no requereix de configuració addicional més enllà de la presentació. Després, la llista, tot i que no és un element tant senzill d'implementar, el major gros de dificultat es centra en la recuperació de les dades de la base de dades. Finalment, l'enllaç entre el formulari de creació de llibre i sala de lectura s'ha fet a través d'un *Stepper*, el qual amb la correcte documentació és fàcil d'implementar, però lent.

Revaluació: En aquest punt del projecte s'ha pogut veure una millora en la despesa de temps per a la implementació de les diferents funcionalitats. A banda de l'agilitat per

programar adquirida gràcies a les iteracions prèvies, la reavaluació de prioritats i la definició de MVP han sigut dos elements essencials per poder encaminar i accelerar més el desenvolupament del projecte.

7.6 Implementació d'usuaris

Planificació: En aquesta iteració es creen i dissenyen els usuaris, així com el procés d'autenticació a l'aplicació.

Disseny: Per implementar els usuaris s'ha dissenyat l'ordre lògic a seguir explicat a l'apartat de "Desenvolupament". En quant a les interfícies s'ha dissenyat un model bàsic d'un perfil amb els elements més essencials i, a aquestes alçades, no s'ha tingut en compte encara la imatge de perfil.

Desenvolupament i testeig: En primer lloc s'ha implementat la interfície de perfil d'un usuari. Després els formularis d'inici de sessió i donar d'alta, desant a la base de dades els usuaris quan aquests validen el formulari per després poder escriure dites dades a la pàgina de perfil de l'usuari. Amb aquests elements creats s'ha incorporat Firebase Authentication per dur a terme l'enregistrament d'usuari, enviament de correu de validació i el control de sessió. Amb el control de sessió s'ha hagut d'alterar altres parts del codi, com per exemple mostrar missatges concrets en segons quines part de l'aplicació en cas que l'usuari no estigui identificat. L'última passa a incorporar ha estat el formulari d'edició de perfil, on els usuaris poden editar les seves dades o afegir-ne d'altres.

Reavaluació: Ja es comptava des de un inici tenir un perfil senzill amb únicament els camps més essencials. En aquesta etapa no hi ha hagut un canvi significatiu de prioritats en quant els usuaris.

7.7 Implementació del xat

Planificació: Per aquesta iteració es creen les sales de xat dins les sales de lectura i s'adapta la navegació. També cal mostrar una interfície adient pel llistat de sales de xat.

Disseny: La part més rellevant del disseny ha estat estudiar el format del xat i on allotjar els missatges. La part gràfica s'ha basat en un model genèric de xat .

Desenvolupament i testeig: Per implementar el xat s'ha hagut d'incorporar Firestore a l'aplicació. El desenvolupament del xat ha estat més simple de lo previst i l'experiència adquirida durant tot el projecte ha jugat un paper important. Al principi s'ha implementat un xat de dues persones i després s'ha convertit en una sala que admet diversos usuaris. A banda del xat ha calgut fer ajustos en la navegació ja que es pot accedir al perfil de la sala de lectura des de dins la sala de xat. També s'han incorporat elements accessoris a cada xat com l'hora d'enviada i el nom de l'usuari emissor.

En la revaluació s'ha considerat viable modificar la llista de sales de xat i afegir atributs com últim missatge rebut o hora de l'últim missatge rebut.

Revaluació: Quan ha finalitzat el desenvolupament essencial del xat s'ha fet una revaluació on s'ha trobat viable, degut a l'acceleració en el desenvolupament, afegir elements no essencials que proporcionen valor addicional al xat.

En aquesta iteració s'ha complert l'objectiu principal de lliurar un MVP i s'ha fet testejar a usuaris per rebre un *feedback*. En el procés s'han trobat errors i proposat petites millores que es treballaran desenvolupament d'aquesta iteració. També s'ha conclòs que l'addició d'elements gràfics com imatges és prioritari pel que és en el que s'enfocarà la següent iteració.

7.8 Incorporació d'imatges i altres millores

Planificació: En aquesta iteració post-MVP es solucionen errades i s'incorporen els elements més visuals de l'aplicació, portades de llibres i fotos de perfil dels usuaris.

Disseny: En aquest cas el disseny ha estat mínim ja que es tracta d'afegir dos *widgets* per pujar imatges als dos formularis de l'aplicació, el perfil i la sala de lectura. Com la única opció viable amb Firebase es Cloud Storage només ha estat necessari estudiar la documentació per emmagatzemar els fitxers. Per aprofitar l'element visual s'ha decidit redissenyar la pàgina de sales de lectura perquè mostri les portades dels llibres. També ha calgut canviar el model de dades a la base de dades per incloure l'enllaç de cada imatge al seu corresponent objecte.

Desenvolupament i testeig: Abans de començar amb la incorporació de les imatges ha estat necessari arreglar les errades que es van trobar en la fase prèvia, com per exemple pàgines

que no es renderitzaven correctament en horitzontal. Per les imatges ha calgut incorporar Firebase Cloud Storage a l'aplicació i configurar correctament la pujada d'imatges i mostra. Per exemple, en el formulari d'edició, quan es puja una imatge, s'espera que es mostri dita imatge en el espai que hi havia abans per seleccionar-ne una. També s'ha de mostrar la imatge existent quan s'obre un formulari per editar-lo.

En quant les millores proposades s'ha incorporat un efecte “splash” als elements interactius de l'aplicació per millorar l'experiència d'usuari.

Revaluació: En el procés de desenvolupament d'aquesta iteració s'han detectat diversos elements que es poden millorar i d'altre que s'haurien de prioritzar, però s'ha arribat al final del projecte. Tot i que procés iteratiu podria seguir endavant sense problema, s'ha aconseguit assolir els objectius principals plantejats i anar una mica més enllà.

8. Conclusions

Les conclusions s'enfocaran segons àrea del projecte.

Estat final de l'aplicació

L'estat final de l'aplicació es satisfactori degut a que s'han pogut assolir els objectius proposats, tot i haver estat reavaluats en cada iteració del desenvolupament. Encara que l'objectiu ha estat lliurar un MVP, s'ha pogut anar més enllà i incloure elements que sumen valor al producte gràcies a l'experiència adquirida amb el desenvolupament. L'aplicació està en un estat el qual es viable seguir en el seu desenvolupament per polir-la o, senzillament, deixar-la com està.

Software de desenvolupament

Tot i que mai s'havia treballat amb Flutter abans i que el procés d'aprendre pot haver estat més lent a l'inici, a etapes finals del projecte s'ha valorat molt la simplificació que proporciona aquest *framework* per implementar noves funcionalitats. La tria de Flutter com a software de desenvolupament ha estat encertada, no només pels resultats que ha proporcionat, sinó per tots els coneixements adquirits en el procés.

Firebase

Firebase ha estat l'eina que ha proporcionat els serveis de backend i, un altre cop, la tria ha estat encertada. Firebase és molt fàcil d'implementar, inclòs sense experiència prèvia. La documentació que proporciona és clara i concisa i no s'hauria pogut arribar tan lluny d'haver treballat amb un backend més complicat. Gràcies a la seva simplicitat s'ha pogut dedicar més temps a àrees més complexes o lentes.

Metodologia

El Lean Agile ha resultat una bona elecció com a metodologia de desenvolupament degut a les qualitats del projecte. Gràcies a l'Agile s'han pogut fer reavaluacions durant tot el projecte per reformular prioritats i objectius. El cas més evident ha estat en enfocar-se en lliurar un MVP, fet que ha permès deixar una aplicació simple, però polida.

9. Possibles ampliacions

Hi ha detalls de la implementació actual que manquen i és necessari tractar, un exemple en pot ser que els noms d'usuari no són únics i això pot portar a confusió entre els usuaris o problemes futurs. Per altre banda, tampoc es mostra el nombre de missatges nous en una sala de xat i, quan s'obre aquesta, sempre és a partir de l'últim missatge enviat a la sala, fet que complica l'experiència de l'usuari ja que no hi ha una guia que marca a partir de quin missatge no ha llegit l'usuari.

A part de les millores d'usabilitat, les ampliacions principals són els objectius secundaris que no s'han pogut implementar. Aquestes inclouen:

- Gestió d'usuaris a les sales de lectures conjuntes. Expulsar usuaris d'una sala de xat.
- Cercar entre totes les sales de lectura disponibles a partir de diversos filtres i un cercador.
- Enviar notificacions i recordatoris sobre les lectures conjuntes en base als esdeveniments programats.
- Crear enquestes o votacions a les sales de xat.
- Censurar texts a les sales de xat.
- Enviar una sol·licitud per poder entrar a una sala de lectura conjunta.
- Gestió d'usuaris a les sales de lectures conjuntes segons el rol de l'usuari. La gestió inclou donar o treure permisos a usuaris i acceptar sol·licituds d'admissió a la sala.
- Afegir usuaris a la llista d'amics d'un perfil.
- Denunciar missatges maliciosos o conductes inapropiades a les sales de xat.

A part de totes les funcionalitats extra llistades, una altra ampliació prioritària és la de oferir altres idiomes. Actualment l'aplicació està desenvolupada en anglès per tractar-se d'una llengua extensament coneguda però, tot i no utilitzar-s'hi un vocabulari complicat, hi ha usuaris que prefereixen el seu idioma matern.

Una bona manera de rebre feedback dels usuaris és a través d'una secció de comentaris a l'aplicació, pel que es pot incorporar un apartat de suggeriments. Aquesta funcionalitat facilita el procés de reavaluació de prioritats i dona una idea més específica d'on encaminar l'aplicació.

Ara mateix els llibres estan continguts dins d'una sala de lectura, però es poden treure fora per poder crear un repositori exclusiu de llibres i permetre als usuaris crear les lectures conjuntes sense necessitat d'introduir per si mateixos tota la informació del llibre.

A banda de les funcionalitats que es poden afegir, un objectiu important és el de optimitzar l'aplicació per evitar crides innecessàries a la base de dades o peticions demanant dades no rellevants. Això també aplica a millorar el navegador de l'aplicació, el qual amb la implementació actual fa masses renderitzacions innecessàries. Flutter és un *framework* nou i constantment es publiquen nous paquets o alternatives més eficients i segures a funcionalitats existents.

10. Bibliografia

- [1] Haupt, Angela: «8 Ways to Read More Books—And Why You Should» Time, 2023. [En línia]. Disponible: <https://time.com/6251239/how-to-read-more/>. [Consulta: 2023 Febrer 4].
- [2] Guilar, Josh: «Why reading is a social activity» Medium, 2015 [En línia]. Disponible: <https://medium.com/the-coffeelicious/why-reading-is-a-social-activity-76cea90d1407>. [Consulta: 2023 Gener 23].
- [3] Pearson, Jennifer; Owen, Tom; Thimbleby, Harold; Buchanan, George: «Co-reading: Investigating collaborative group reading» ResearchGate, 2012 [En línia]. Disponible: https://www.researchgate.net/publication/262408322_Co-reading_Investigating_collaborative_group_reading. [Consulta: 2023 Gener 23].
- [4] Goodreads: «About Goodreads» [En línia]. Disponible: <https://www.goodreads.com/about/us>. [Consulta: 2023 Febrer 06].
- [5] Wikipedia: «Instagram» [En línia]. Disponible: <https://en.wikipedia.org/wiki/Instagram>. [Consulta: 2023 Gener 25].
- [6] The StoryGraph: «The StoryGraph» [En línia]. Disponible: <https://thestorygraph.com/>. [Consulta: 2023 Febrer 07].
- [7] @thestorygraph: «Buddy reads are now live» Twitter, 2021. [En línia]. Disponible: <https://pbs.twimg.com/media/FH9xjnxXIA8GvI7?format=jpg&name=large>. [Consulta: 2023 Febrer 07].
- [8] The StoryGraph: «Read with friends» [En línia]. Disponible: <https://www.thestorygraph.com/assets/feature-buddy-reads-f06af15c3ca2dd5f96cebc9de27b04db68fa591b0506c143929b4c5dbace1b5c.png>. [Consulta: 2023 Febrer 07].
- [9] Tina: «Goodreads vs. The StoryGraph- Pros and Cons -» TBRETC, 2021 [En línia]. Disponible: <https://www.tbretc.com/etc/2021/3/28/goodreads-vs-the-storygraph-pros-and-cons>. [Consulta: 2023 Febrer 01].
- [10] The StoryGraph: «The StoryGraph Feedback» [En línia]. Disponible: <https://roadmap.thestorygraph.com/>. [Consulta: 2023 Febrer 01].
- [11] @myonnaslibrary: «Comentari» Twitter, 2023 [En línia]. Disponible: <https://twitter.com/myonnaslibrary/status/1622110009256460288?s=20&t=GMXy5HozB9p6koK9SgASQ>. [Consulta: 2023 Febrer 02].

- [12] Google Play: «StoryGraph - Apps on Google Play» [En línia]. Disponible: <https://play.google.com/store/apps/details?id=com.thestorygraph.thestorygraph>. [Consulta: 2023 Febrer 02].
- [13] Smith, Jim: «Native vs. Cross-Platform Mobile Dev: Pros, Cons, and Best Practices.» CircleCI Blog, 2022. Disponible: <https://circleci.com/blog/native-vs-cross-platform-mobile-dev/>. [Consulta: 2023 Març 12].
- [14] Extern Labs: «Native vs Hybrid vs Cross-Platform: Which One to Choose for Mobile App Development?» [En línia]. Disponible: <https://externlabs.com/blogs/native-vs-hybrid-vs-cross-platform/>. [Consulta: 2023 Març 12].
- [15] EZ Rankings: «Top 10 Cross-Platform App Frameworks for Mobile App Development» [En línia]. Disponible: <https://www.ezrankings.com/blog/cross-platform-app-frameworks>. [Consulta: 2023 Març 13].
- [16] Back4App Blog: «Flutter App Backend: How to Build Your Backend as a Service» [En línia]. Disponible: https://blog.back4app.com/flutter-app-backend/#What_is_a_Backend_as_a_Service. [Consulta: 2023 Març 15].
- [17] Back4App Blog: «The best 5 backend services for your Flutter app» [En línia]. Disponible: https://blog.back4app.com/flutter-app-backend/#The_best_5_backend_services_for_your_Flutter_app. [Consulta: 2023 Març 15].
- [18] Quora: «What is better, AWS Amplify or Firebase?» [En línia]. Disponible: <https://www.quora.com/What-is-better-AWS-Amplify-or-Firebase>. [Consulta: 2023 Març 15].
- [19] Elitech Systems: « Best Backend for a Flutter Application. Elitech Systems» [En línia]. Disponible: <https://www.elitechsystems.com/best-backend-for-a-flutter-application/>. [Consulta: 2023 Març 15].
- [20] Google Firebase: «Realtime Database» [En línia]. Disponible: <https://firebase.google.com/products/realtime-database>. [Consulta: 2023 Març 02].
- [21] Google Firebase: «Firebase Authentication» [En línia]. Disponible: <https://firebase.google.com/products/auth>. [Consulta: 2023 Maig 19].
- [22] Google Firebase: «Cloud Firestore» [En línia]. Disponible: <https://firebase.google.com/docs/firestore>. [Consulta: 2023 Juny 10].
- [23] Google Firebase: «Firebase Storage» [En línia]. Disponible: <https://firebase.google.com/products/storage>. [Consulta: 2023 Juny 22].

- [24] Google Firebase: «Firebase Cloud Messaging» [En línia]. Disponible: <https://firebase.google.com/products/cloud-messaging>. [Consulta: 2023 Març 22].
- [25] Atlassian: «What is Agile?» [En línia]. Disponible: <https://www.atlassian.com/agile>. [Consulta: 2023 Febrer 03].
- [26] Wrike: «The Agile Software Development Life Cycle: Wrike Agile Guide» [En línia]. Disponible: <https://www.wrike.com/agile-guide/agile-development-life-cycle/>. [Consulta: 2023 Febrer 03].
- [27] Krasamo: «Agile». [En línia]. Disponible: <https://www.krasamo.com/wp-content/uploads/agile-01-scaled.jpeg> . [Consulta: 2023 Febrer 03].
- [28] Indeed Editorial Team: «14 Pros and Cons of Agile Methodology (Plus Definition)» Indeed, 2021. [En línia]. Disponible: <https://www.indeed.com/career-advice/career-development/pros-cons-of-agile-methodology>. [Consulta: 2023 Febrer 04].
- [29] Hackr.io: «How to Learn Flutter: A Comprehensive Guide.» [En línia]. Disponible: <https://hackr.io/blog/how-to-learn-flutter>. [Consulta: 2023 Març 15].
- [30] Pinkiewicz, Mateusz: « Is Firebase the Best Back-End Choice for a Flutter App?» Brandmed. Blog, 2021. Disponible: <https://brandmed.com/blog/development/is-firebase-the-best-back-end-choice-for-a-flutter-app>. [Consulta: 2023 Març 15].
- [31] Bizzotto, Andrea: «Flutter project structure: tips & tricks» Code with Andrea, 2022. [En línia]. Disponible: <https://codewithandrea.com/articles/flutter-project-structure/>. [Consulta: 2023 Abril 02].
- [32] Bizzotto, Andrea: «Imatge» Twitter, 2022. Disponible: <https://twitter.com/biz84/status/1506285698558660609/photo/1>. [Consulta: 2023 Abril 02].
- [33] Cianci, Lewis: «Best IDEs for Flutter in 2022 » LogRocket Blog, 2022. Disponible: <https://blog.logrocket.com/best-ides-flutter-2022/#android-studio/>. [Consulta: 2023 Febrer 13].
- [34] Firebase: «Add Firebase to your Flutter app». [En línia]. Disponible: <https://firebase.google.com/docs/flutter/setup?platform=android>. [Consulta: 2023 Abril 10].
- [35] Flutter: «Navigator class - material library - Dart API». [En línia]. Disponible: <https://api.flutter.dev/flutter/widgets/Navigator-class.html>. [Consulta: 2023 Març 01].
- [36] Flutter: «Offstage class - material library - Dart API». [En línia]. Disponible: <https://api.flutter.dev/flutter/widgets/Offstage-class.html>. [Consulta: 2023 Març 01].

- [37] Flutter: «FloatingActionButton class - material library - Dart API». [En línia]. Disponible: <https://api.flutter.dev/flutter/material/FloatingActionButton-class.html>. [Consulta: 2023 Abril 12].
- [38] Flutter: «Stepper class - material library - Dart API». [En línia]. Disponible: <https://api.flutter.dev/flutter/material/Stepper-class.html>. [Consulta: 2023 Abril 14].
- [39] Flutter: «Build a form with validation». [En línia]. Disponible: <https://docs.flutter.dev/cookbook/forms/validation>. [Consulta: 2023 Març 18].
- [40] Pub.dev: «image_picker 0.8.8 | Flutter Package». [En línia]. Disponible: https://pub.dev/packages/image_picker. [Consulta: 2023 Juny 08].
- [41] Pub.dev: «table_calendar | Flutter Package». [En línia]. Disponible: https://pub.dev/packages/table_calendar. [Consulta: 2023 Maig 28].
- [42] Prerender.io: « Frontend vs. backend: Should I start with the front end or back end of my web app?». [En línia]. Disponible: <https://prerender.io/blog/frontend-vs-backend-should-i-start-with-the-front-end-or-back-end-of-my-web-app/>. [Consulta: 2023 Març 14].
- [43] Flutter: «Stream class - dart:async library». [En línia]. Disponible: <https://api.flutter.dev/flutter/dart:async/Stream-class.html>. [Consulta: 2023 Juny 01].