



Alumnes: Aimar Benali Kharkhour
Laura Roldán García

Tutores: Lindsay Barrero, Ivan Cantón, Marisol López
CFGS 2DAM tarda 2022-2023

1. INTRODUCCIÓ	5
1.1. OBJECTIUS DE L'APLICACIÓ	5
1.2. NECESSITATS A RESOLDRE	5
1.3. RECOLLIDA DE REQUISITS	5
1.3.1. Actes de reunions amb clients	5
1.4. ANÀLISIS DE REQUISITS	7
1.4.1. Guions amb les funcionalitats dels perfils d'usuaris	7
1.4.2. Requisits no funcionals	9
2. PLA DE PROJECTE	9
2.1. ESTUDI DE MERCAT (F4)	9
2.1.1. Situació i evolució del mercat, en el que s' integren la empresa i àmbit geogràfic d'actuació del negoci.	9
2.1.2. Productes i / o serveis que s'ofereixen i els preus.	9
2.1.3. Clientela. Mapa d'empatia	9
2.1.4. Competència.	10
2.1.5. Distribució.	10
2.1.6 DAFO	10
2.2. ESTRATÈGIA COMERCIAL (F6)	10
2.2.1. Localització o ubicació del negoci.	10
2.2.2. Promoció i comunicació.	10
2.2.3. Previsió de vendes	10
2.3. PRODUCCIÓ, RECURSOS MATERIALS I HUMANS (F7)	10
2.3.1. Procés productiu.	10
2.3.2. Recursos materials.	10
2.3.3. Proveïdors.	10
2.3.4. Recursos humans	10
2.4. PLA JURÍDIC I NORMATIU (F7)	10
2.4.1. Forma jurídica i normativa a complir relacionada amb la pròpia activitat.	10
2.4.2. Normativa de Prevenció Riscos Laborals, normativa ambiental.	10
2.5. RESPONSABILITAT SOCIAL EMPRESARIAL / ACCIONS ECONOMIA SOCIAL (F8)	10
2.5.1. Responsabilitat Social Empresarial	10
2.5.2. Accions d'Economia Social	10
2.6. ANÀLISIS ECONÒMICA I FINANCERA (F9)	10
2.6.1. Càlcul del cost	11
2.6.2. Càlcul del punt mort	11
2.6.3. Viabilitat	11
2.6.4. Resum executiu	11
3. PLANIFICACIÓ I SEGUIMENT DE LES TASQUES DE DESENVOLUPAMENT	12

3.1. EINES	12
3.2. PLANIFICACIÓ INICIAL	12
3.2.1. Planificació amb Trello	12
3.2.2. Diagrama de Gantt	14
3.3. SEGUIMENT SETMANAL	14
3.3.1. SETMANA 1	14
3.3.2. SETMANA 2	16
3.3.3. SETMANA 3	17
3.3.4. SETMANA 4	19
3.3.5. SETMANA 5	22
3.3.6. SETMANA 6	24
3.3.7. SETMANA 7	25
3.3.8. SETMANA 8	26
3.3.8. SETMANA 9 i 10	27
3.3.9. SETMANA FINAL	27
4. DISSENY	28
4.1. MOCKUPS	28
4.1.1 PANTALLES	28
4.1.2. NAVEGACIÓ	33
4.1.2.1. Diagrama de navegació	34
4.1.2.2. Prototipus Marvel	34
4.2. DISSENY BASE DE DADES NO-SQL	34
5. DESENVOLUPAMENT DE L'APP	35
5.1. CONTROL DE VERSIONS	36
5.2. ESTRUCTURA DE PROJECTE (PACKAGES)	40
5.3. CODIFICACIÓ	40
5.3.1 NAVEGABILITAT. MENÚS	40
5.3.2. SPLASH SCREEN	40
5.3.3. BASE DE DADES FIRESTORE DE FIREBASE	41
● Authentication	41
5.3.3.1. CONNEXIÓ AMB LA BASE DE DADES	41
5.3.3.2. REGISTRE I ACCÉS A L'APP	46
5.3.2.3. PERFILE	57
5.3.3.2. INSERCIÓ	58
5.3.3.3. CONSULTA	58
5.3.3.4. MODIFICACIÓ	61
5.3.3.5. ELIMINACIÓ	61
5.3.4. FIREBASE STORAGE	61

5.3.4.1. CÀRREGA D'IMATGES	61
5.3.4.2. CÀRREGA DE VÍDEOS	61
5.3.5. BÚSQUEDA. FILTRES.	61
FINAL DE CODIFICACIÓ : 12A	62
. Creació de classes amb el constructor que forma part de la capçalera i/o altres constructors secundaris:	62
. Modificadors de visibilitat correctament aplicats. (private, public, protected):	62
. Atributs de classe. Encapsulació (getters and setters):	62
. Funcions. Classe Utils/Common amb funcions comuns a vàries classes:	62
. Herència:	62
. Interfaces. Abstract classes. Overridings.	63
. Data classes.	63
. Nested and inner classes.	63
. Enum classes	63
. Companion objects	64
Altres:	64
. Regular expressions	64
. Null safety	64
. Exceptions	64
. Annotations	64
. Implementació d'algoritmes d'ordenació (selection sort, bubble sort, merge sort...)	64
. Implementació d'algoritmes de cerca (sequential search, binary search,...)	64
. Callbacks	64
. Coroutines	64
. Destructuring declarations	64
5.3.6. PAS DE DADES ENTRE PANTALLES.	65
5.3.7. ANIMACIONS	65
5.3.8 NOTIFICACIONS	65
5.3.9 GEOLOCALITZACIÓ	65
5.4. STYLES I THEMES	65
5.5. DISSENY ALTERNATIUS PER VARIS TIPUS DE PANTALLA.	66
5.6. ACCESSIBILITAT	66
5.7. INTERNACIONALITZACIÓ.	66
5.8. GENERACIÓ APK I PUBLICAR A GOOGLE PLAY	66
6. TESTEIG	66
6.1. Tests Unitaris	66
6.2. Tests amb usuaris finals	66
7. APPLICACIÓ FINAL	67

7.1. Manual d'usuari	67
7.2. Vídeo-demo	67
8. LÍNIES FUTURES	67
9. GESTIÓ DEL PROJECTE	68
9.1. COMUNICACIÓ (F1)	68
9.2. PERFIL PROFESSIONAL (F2)	68
9.3. TREBALL EN EQUIP (F3)	68
9.4. CONFLICTES I RESOLUCIÓ (F5)	68
10. CONCLUSIÓ	68

1. INTRODUCCIÓ

1.1. OBJECTIUS DE L'APLICACIÓ

L'objectiu de la nostra aplicació és ajudar a la gent que més ho necessita o a la població amb diversitat funcional, ja que hem vist que, per exemple, les persones cegues no tenen tanta facilitat per utilitzar els mòbils o qualsevol dispositiu portàtil.

Estem desenvolupant una aplicació per donar facilitats i automatitzar les tasques de la gent que necessita una agenda per a la medicació, tenir un control de les visites mèdiques, una manera de trucar a l'hospital o trobar-ne el més proper.

1.2. NECESSITATS A RESOLDRE

Hi han moltes persones que pateixen de discapacitats, i hem trobat que poques aplicacions compleixen amb les característiques que necessita la gent incapacitada com a utilització diària.

També, que degut a la seva discapacitat no tenen cap sortida d'emergència (menys les persones del seu voltant que li puguin ajudar), ja sigui, trucades al/la cuidador/a, o un sistema de rastreig per a qui estigui al càrrec pugui anar-hi i estar a l'ajuda.

A més, hem vist que moltes d'aquestes persones careixen dels coneixements suficients com per utilitzar els nous dispositius que s'han anat fabricant últimament.

1.3. RECOLLIDA DE REQUISITS

1.3.1. Actes de reunions amb clients

ACTA DE REUNIÓ

CLIENT:

DIA: 19/09/2022

HORA: 19:12

LLOC: Industrial 43, Carrer Font Vella, 78 (Terrassa)

Assistents
<ul style="list-style-type: none">• Aimar Benali Kharkhour - Desenvolupador• Ángel Martínez González - Desenvolupador• Laura Roldán García - Administrativa• Adam Sanchez Po - Client
Ordre del dia
<ol style="list-style-type: none">1. Presentació dels assistents2. Presentació del projecte inicial

- 2.1. Brainstorming
- 2.2. Objectius
- 2.3. Necesitats que cobreix
- 2.4. Persones destinatàries
- 2.5. Planificació
- 2.6. Especificació de les funcions
 - 3. Estudis fets
 - 4. Torn obert de preguntes
- 5. Registre dels acords / programació propera reunió
- 6. Creació de llista per a la distribució del treball sobre el projecte.

Desenvolupament

- Presentació del projecte inicial
- Aplicació per a mòbil Android
- La idea inicial és oferir a les persones amb diversitat funcional o persones majors una aplicació que els ajudi en el seu dia a dia. Amb una alta capacitat de personalització.
- La primera interfície seria una de preguntes per saber que necessita el client.
- Després estaria la pantalla inicial on estaven totes les opcions per anar a la part que els interessi.
- Les principals funcions serien: recordatori dels medicaments que es pren, un mapa amb GPS incorporat, emergències, modes de suport a la diversitat funcional (narratiu, daltònic, etc.), agenda, tutorials sanitaris i configuració.
- Logo:
- Estudis fets: explicació del grau superior de DAM
- Torn obert de preguntes:
- Preguntes dels clients respecte a les diverses funcionalitats de l'app i idees diferents
- Registre dels acords:
- El coordinador es fa responsable de realitzar les actes de reunió, d'especificar els acords i la seva temporització i de convocar futures reunions
- Llista de funcions:
- S'acorda la divisió de funcions entre els programadors i la seva temporització

Acords

Acord	Responsable	temporització
Realitzar l'acta	Laura Roldán García	19/09/2022
Creació de la llista de desenvolupament de les funcions	Laura Roldán García	22/09/2022
Convocar una reunió més properament	Laura Roldán García	Abans de desembre
Seguiment dels acords anteriors		

Acord	Responsible	Data	Estat
Observacions			
Primera reunió realitzada			

1.4. ANÀLISIS DE REQUISITS

1.4.1. Guions amb les funcionalitats dels perfils d'usuaris

Actor
Client
Descripció
Usuari que s'ocupa d'administrar l'aplicació, de la seva funcionalitat i reportar-ho als desenvolupadors.
Guió
<ul style="list-style-type: none"> Aplicació per a mòbil Android Recordatori i seguiment dels medicaments GPS amb seguiment de l'usuari en ús de l'app Contactes d'emergència, on tindran els números de telèfon de les persones que estan cuidant a la persona que esta utilitzant l'app. Modalitats per l'adaptació de l'individu que està utilitzant, ja sigui per qualsevol discapacitat (diferenciació de colors, discapacitat ocular, etc...). Calendari, per el control de les cites del client, en cas de voler programar els seus dies de manera que organitza bé per no faltar a les cites mèdiques. Tutorials sanitaris, per a que l'usuari pugui sapiguer la importància de mantenir una bona higiene i sigui prudent amb la seva medicació, per fer conscient de que cuidar-se es important. Configuració, de manera que l'usuari podrà ajustar l'app, de la manera que li vingui més còmode.

Actor
Programadors
Descripció
Persona que s'encarrega del desenvolupament i monitorització de l'aplicació.
Guió
<ul style="list-style-type: none"> Mantener actualizada la app

- Fer una investigació per l'app de manera que quedi clar com començar a programar l'aplicació en qüestió.
- Disseny de l'aplicació, de manera que sigui "bonic" o mes comode per els ulls de l'usuari que utilitza l'app.
- Transcriure a llenguatge de programació, es à dir, traduir tot el llenguatge que tenia pensat, en el llenguatge que la màquina entén. (Java, HTML, JavaScript, C#, etc...)
- Monitoritzar l'aplicació de manera que tingui tot controlat.
- Buscar solucions als problemes que es poden presentar durant la programació o quan el Beta Tester detecti bugs.
- Lectura i anàlisis de l'aplicació.

Actor
Beta Tester
Descripció
Usuari que utilitza com a simulació i prova l'app abans de llençar-se a l'AppStore
Guió
<ul style="list-style-type: none"> ● Utilització de l'app ● Seguiment de millors ● Puntuació d'errors ● Decisió de llançament ● Parlar de punts a millorar ● Observació detallada de l'app ● Reportar bugs o problemes ● Posar comentaris sobre l'app en qüestió

Actor
Client
Descripció
Persona interessada en el desenvolupament de l'app, principalment per l'activitat econòmica
Guió
<ul style="list-style-type: none"> ● Explicació de com vol que sigui l'app ● Seguiment del desenvolupament setmanalment ● Decisions de mercat

- Monitoritzar el treball fet

1.4.2. Requisits no funcionals

- Connexió base de dades amb Firebase
- App amb Android Studio feta amb Kotlin
- Geolocalització
- Alarms
- Calendari customizable (agenda)
- Idiomes app: català

2. PLA DE PROJECTE

2.1. ESTUDI DE MERCAT (F4)

2.1.1. Situació i evolució del mercat, en el que s' integren la empresa i àmbit geogràfic d'actuació del negoci.

2.1.2. Productes i / o serveis que s'ofereixen i els preus.

2.1.3. Clientela. Mapa d'empatia



2.1.4. Competència.

2.1.5. Distribució.

2.1.6 DAFO

2.2. ESTRATÈGIA COMERCIAL (F6)

https://drive.google.com/file/d/1WtWdFWB936GQoSpcRZkdUc_T4RElx6KA/view

2.2.1. Localització o ubicació del negoci.

2.2.2. Promoció i comunicació.

2.2.3. Previsió de vendes

2.3. PRODUCCIÓ, RECURSOS MATERIALS I HUMANS (F7)

<https://drive.google.com/file/d/1ppLSzVuwSZNXOuS4dSmZpqRcBuhIDVnt/view?usp=sharing>

2.3.1. Procés productiu.

2.3.2. Recursos materials.

2.3.3. Proveïdors.

2.3.4. Recursos humans

2.4. PLA JURÍDIC I NORMATIU (F7)

<https://drive.google.com/file/d/1ppLSzVuwSZNXOuS4dSmZpqRcBuhIDVnt/view?usp=sharing>

2.4.1. Forma jurídica i normativa a complir relacionada amb la pròpia activitat.

2.4.2. Normativa de Prevenció Riscos Laborals, normativa ambiental.

2.5. RESPONSABILITAT SOCIAL EMPRESARIAL / ACCIONS ECONOMIA SOCIAL (F8)

<https://drive.google.com/drive/u/1/folders/1k37TQolpTUdIO5qhqjQJb63xGOPqgmUw>

2.5.1. Responsabilitat Social Empresarial

2.5.2. Accions d'Economia Social

2.6. ANÀLISIS ECONÒMICA I FINANCERA (F9)

<https://drive.google.com/file/d/1PYEiJ8E5dCe8b5gEJC6V2bAry2pE1iuX/view?usp=sharing>

2.6.1. Càlcul del cost

2.6.2. Càlcul del punt mort

2.6.3. Viabilitat

2.6.4. Resum executiu

3. PLANIFICACIÓ I SEGUIMENT DE LES TASQUES DE DESENVOLUPAMENT

Breu definició de les metodologies Kanban, Agile i Scrum.

- Kanban: La metodologia Kanban es tracta d'un mètode visual de gestió de projectes que permet als equips visualitzar els seus fluxos de treball i la càrrega de treball. En un tauler Kanban, el treball es mostra en un projecte en forma de tauler organitzat per columnes. Cada columna representa una etapa del treball. El tauler Kanban més bàsic està format per les columnes "To do", "In progress", un altre de treballs pendents i "Done". Les tasques individuals, representades per targetes visuals en el tauler, avancen a través de les diferents columnes fins que estiguin finalitzades.
- Agile: la metodologia Agile és la que permet adaptar la forma de treball a les condicions del projecte, aconseguint flexibilitat i immediatesa en la resposta per a emmotillar el projecte i el seu desenvolupament a les circumstàncies específiques de l'entorn.
- Scrum: és un procés al qual s'apliquen de manera regular un conjunt de bones pràctiques per a treballar col·laborativament en equip i obtenir el millor resultat possible de projectes, caracteritzat per (entre altres), adoptar una estratègia de desenvolupament incremental en lloc de la planificació i execució completa del producte, basar la qualitat del resultat en el coneixement de les persones més que en la qualitat dels processos emprats i ajuntar les diferents fases del desenvolupament.

3.1. EINES

- Trello: és una eina organitzativa web que té la funció de fer més fàcil l'organització de les tasques dels equips de treball. A través de la seva interfície els usuaris poden veure el projecte de forma organitzada i saber que és el que s'ha fet, el que s'està fent i el que s'ha de fer.
- TeamGantt: et permet posar sobre un calendari el diagrama de Gantt que et fa automàticament a partir del Trello que sincronitzi, facilitant veure que tasques s'han de fer cada setmana i si es bloqueja una com afecta a totes les altres, ja que et deixa fer que les següents tasques depenen directament del treball d'abans. A més a més, et deixa filtrar les tasques per només veure les d'una determinada persona, les que van tardo o han acabat abans del previst o filtrar per setmanes.

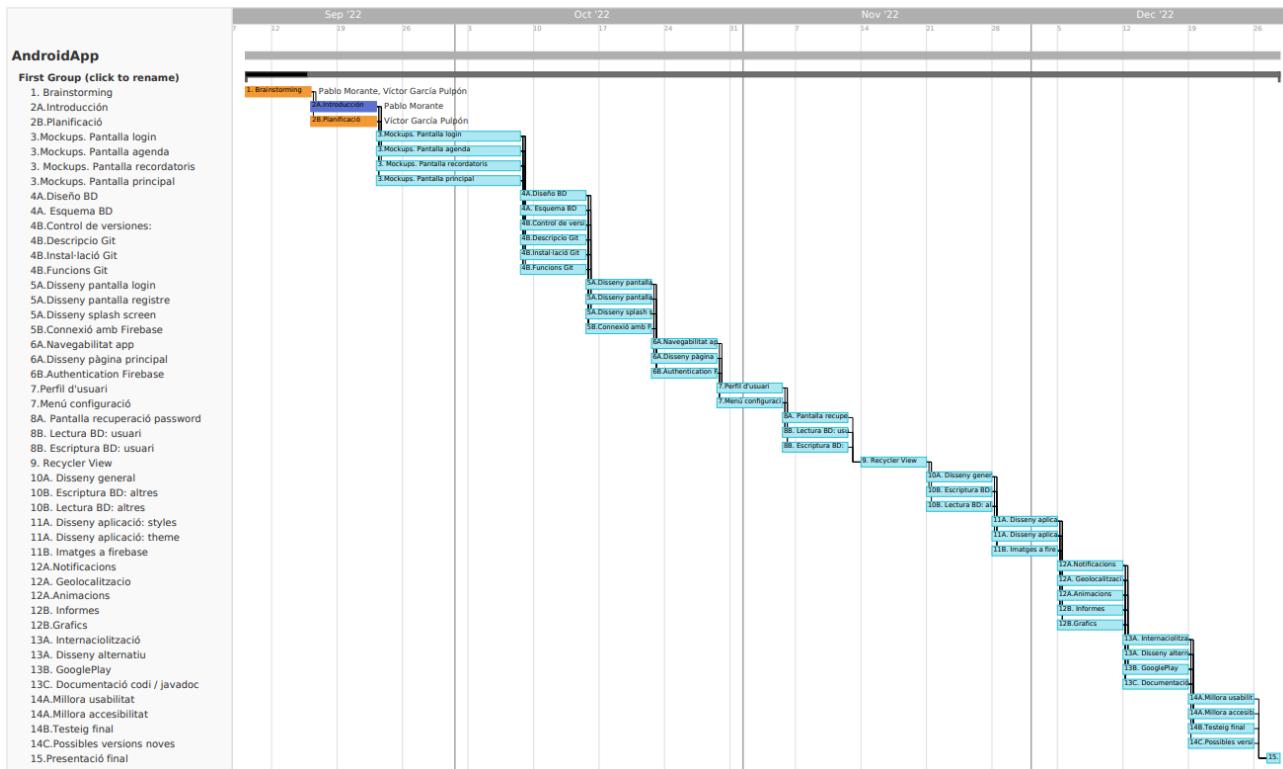
3.2. PLANIFICACIÓ INICIAL

3.2.1. Planificació amb Trello

Link al tauler de Trello → <https://trello.com/b/acGKbNMT/planificaci%C3%B3>

Column	Task Description	Due Date	Team Members
To do	Setmana 1	8 de sep. - 14 de sep.	AK, LG, PAB, VIC
	1.Brainstorming		
	Setmana 2	15 de sep. - 21 de sep.	PAB, VIC
	2B.Planificación		
	Setmana 2	15 de sep. - 21 de sep.	AK, LG
	2A.Introducción		
	Setmana 3	22 de sep. - 28 de sep.	LG
Setmana 3	22 de sep. - 28 de sep.	AK	
Setmana 3	22 de sep. - 28 de sep.	PAB	
Setmana 3	22 de sep. - 28 de sep.	VIC	
Setmana 3	22 de sep. - 28 de sep.		
Setmana 3	22 de sep. - 28 de sep.		
Setmana 3	22 de sep. - 28 de sep.		
+ Añada una tarjeta			

3.2.2. Diagrama de Gantt



3.3. SEGUIMENT SETMANAL

3.3.1. SETMANA 1

To do	In Progress	Blocked	Done
44 tarjetas	0 tarjeta	0 tarjeta	2 tarjetas
1. Brainstorming ① 8 de sep. - 14 de sep. Sep 14 AK LG PAB VIC	+ Añada una tarjeta	+ Añada una tarjeta	+ Añada una tarjeta

The screenshot shows a project management interface with a Kanban board at the top and a Gantt chart below it.

Kanban Board:

- To do:** 43 tarjetas. Action: + Añada una tarjeta.
- In Progress:** 0 tarjeta. Action: + Añada una tarjeta.
- Blocked:** 0 tarjeta. Action: + Añada una tarjeta.
- Done:** 3 tarjetas. Sub-task: 1. Brainstorming (8 de sep. - 14 de sep., 100%, Sep 14). Participants: AK, LG, PAB, VIC.

Gantt Chart:

Task	Assigned	Progress	SEPTEMBER 2022																
			8	9	12	13	14	15	16	19	20	21	22	23	26	27	28	29	30
AndroidApp		6%																	
First Group (click to rename)		6%																	
1. Brainstorming	Pablo Morante, Víctor García Pulpón	100%	1. Brainstorming																

3.3.2. SETMANA 2

Board view:

Column	Task	Progress
To do	45 tarjetas	0%
In Progress	0 tarjeta	0%
Blocked	0 tarjeta	0%
Done	1 tarjeta	0%

Column	Task	Progress
To do	43 tarjetas	0%
In Progress	0 tarjeta	0%
Blocked	0 tarjeta	0%
Done	3 tarjetas	100%

Gantt chart for September 2022:

Day	8	9	12	13	14	15	16	19	20	21	22	23	26	27	28	29	30	3	4
Assigned																			
Progress	6%	6%																	
Tasks	1. Brainstorming	Pablo Morante, Víctor García Pulpón	100%	1. Brainstorming	Pablo Morante, Víctor García Pulpón	100%	2A.Introducción	Pablo Morante	100%	2A.Introducción	Pablo Morante	100%	2B.Planificació	Víctor García Pulpón	Víctor García Pulpón				

3.3.3. SETMANA 3

To do	In Progress	Blocked	Done
39 tarjetas	11 tarjetas	0 tarjeta	3 tarjetas
+ Añada una tarjeta	3. Mockups. Reunión para acordar estructura base de la app, colores y diseño en general. ① 29 de sep. 5% Sep 29 AK, LG, PAB, VIC	+ Añada una tarjeta	1. Brainstorming ① 8 de sep. - 14 de sep. 100% Sep 14 AK, LG, PAB, VIC
	3. Mockups. Splash screen. ① 29 de sep. - 6 de oct. 0% Oct 6 VIC		2A.introducción ① 15 de sep. - 21 de sep. 100% Sep 21 AK, LG
	3.Mockups. Pantalla login. ① 22 de sep. - 6 de oct. 5% Oct 6 LG		2B.Planificació ① 15 de sep. - 21 de sep. 100% Sep 21 PAB, VIC
	3.Mockups. Pantalla principal. ① 29 de sep. - 6 de oct. 5% Oct 6 VIC		+ Añada una tarjeta
	3. Mockups. Pantalla de registro. ① 29 de sep. - 6 de oct. 0% Oct 6 AK		
	3. Mockups. Pantalla de + Añada una tarjeta		

Planificació | Público | Tablero | View in TeamGantt | Automatización | Pov

To do	In Progress	Blocked	Done
39 tarjetas	0 tarjeta	0 tarjeta	13 tarjetas
+ Añada una tarjeta	+ Añada una tarjeta	+ Añada una tarjeta	

Setmana 3
3. Mockups. Reunión para acordar estructura base de la app, colores y diseño en general.
Due 5 @ 29 de sep. 100%
Sep 29

AK LG PAB VIC

Setmana 3
3. Mockups. Splash screen.
@ 29 de sep. - 6 de oct. 100%
Oct 6

VIC

Setmana 3
3. Mockups. Pantalla login.
@ 29 de sep. - 6 de oct. 100%
Oct 6

LG

Setmana 1
1. Brainstorming
@ 8 de sep. - 14 de sep.
100% Sep 14

AK LG PAB VIC

Setmana 3
3. Mockups. Pantalla principal.
@ 29 de sep. - 6 de oct. 100%
Oct 6

VIC

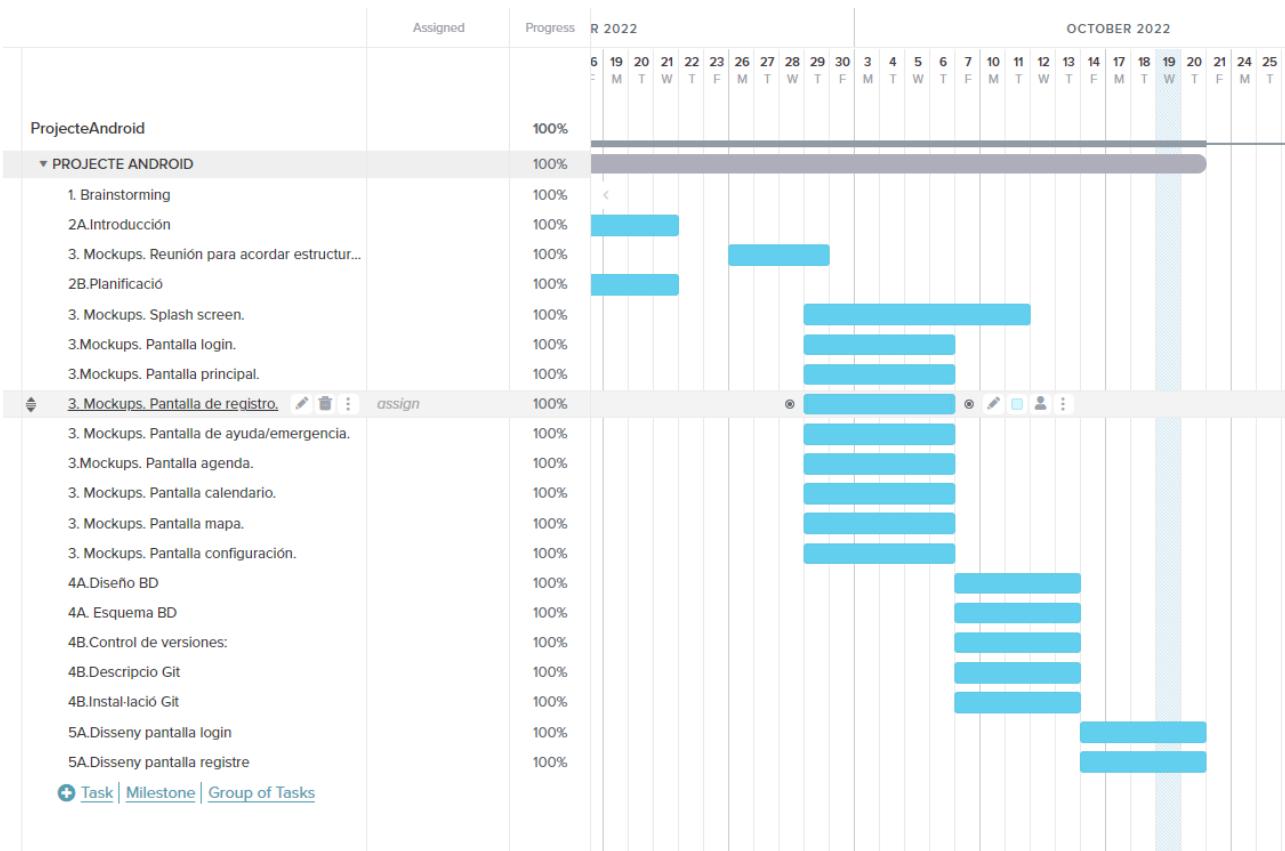
+ Añada una tarjeta

3.3.4. SETMANA 4

To do	In Progress	Blocked	Done
Setmana 6 6A. Navegabilitat app ⌚ 26 de oct. - 1 de nov.	Setmana 4 4A. Disseny pàgina principal ⌚ 7 de oct. - 13 de oct. LG	+ Añada una tarjeta	Setmana 3 3. Mockups. Pantalla login. ⌚ 29 de sep. - 6 de oct. LG
Setmana 6 6B. Disseny pàgina principal ⌚ 26 de oct. - 1 de nov. AK	Setmana 4 4A. Esquema BD ⌚ 7 de oct. - 13 de oct. LG		Setmana 3 3. Mockups. Pantalla principal. ⌚ 29 de sep. - 6 de oct. LG
Setmana 8 8B. Lectura BD: usuari ⌚ 9 de nov. - 15 de nov. LG	Setmana 4 4B. Control de versiones: ⌚ 7 de oct. - 13 de oct. AK LG		Setmana 3 3. Mockups. Pantalla de registro. ⌚ 29 de sep. - 6 de oct. AK
Setmana 8 8B. Escriptura BD: usuari ⌚ 4 de nov. - 10 de nov. LG	Setmana 4 4B. Descripció Git ⌚ 7 de oct. - 13 de oct. LG		Setmana 3 3. Mockups. Pantalla de ayuda/emergencia. ⌚ 29 de sep. - 6 de oct. LG
Setmana 9 9. Recycler View ⌚ 17 de nov. - 23 de nov. LG	Setmana 4 4B. Instal·lació Git ⌚ 7 de oct. - 13 de oct. AK		Setmana 3 3. Mockups. Pantalla agenda. ⌚ 29 de sep. - 6 de oct. AK
Setmana 10 10A. Disseny general ⌚ 24 de nov. - 30 de nov. AK	Setmana 4 4B. Funcions Git ⌚ 12 de oct. - 18 de oct. AK		Setmana 3 3. Mockups. Pantalla calendario. ⌚ 29 de sep. - 6 de oct. AK
Setmana 10 10B. Escriptura BD: altres ⌚ 24 de nov. - 30 de nov. LG	Setmana 5 5A. Disseny pantalla registre ⌚ 4 de oct. - 10 de oct. AK LG		Setmana 3 3. Mockups. Pantalla mapa. ⌚ 29 de sep. - 6 de oct. AK
Setmana 10 10B. Lectura BD: altres ⌚ 24 de nov. - 30 de nov. LG	Setmana 5 5B. Connexió amb Firebase ⌚ 2 de oct. - 8 de oct. B	+ Añada una tarjeta	Setmana 3 3. Mockups. Pantalla configuración. ⌚ 29 de sep. - 6 de oct. AK
Setmana 11 11A. Disseny aplicació: styles ⌚ Añada una tarjeta		+ Añada una tarjeta	+ Añada una tarjeta

The screenshot displays a project management board with the following sections:

- To do:**
 - Setmana 6: 6A. Navegabilitat app (Due: 26 de oct. - 1 de nov., AK)
 - Setmana 6: 6A. Disseny pàgina principal (Due: 26 de oct. - 1 de nov., AK)
 - Setmana 8: 8B. Lectura BD: usuari (Due: 9 de nov. - 15 de nov., LG)
 - Setmana 8: 8B. Escriptura BD: usuari (Due: 4 de nov. - 10 de nov., LG)
 - Setmana 9: 9. Recycler View (Due: 17 de nov. - 23 de nov., LG)
 - Setmana 10: 10A. Disseny general (Due: 24 de nov. - 30 de nov., AK)
 - Setmana 10: 10B. Escriptura BD: altres (Due: 24 de nov. - 30 de nov., LG)
 - Setmana 10: 10B. Lectura BD: altres (Due: 24 de nov. - 30 de nov., LG)
 - Setmana 11: 11A. Disseny aplicació: styles (Due: -, LG)
- In Progress:**
 - Setmana 5: 5A. Disseny pantalla registre (Due: 14 de oct. - 20 de oct., AK, LG)
 - Setmana 5: 5B. Connexió amb Firebase (Due: 14 de oct. - 20 de oct., B)
 - Setmana 5: 5A. Disseny pantalla login (Due: 14 de oct. - 20 de oct., AK, LG)
 - Setmana 5: 5A. Disseny splash screen (Due: 14 de oct. - 20 de oct., LG)
 - Setmana 6: 6B. Authentication Firebase (Due: 21 de oct. - 27 de oct., B)
 - Setmana 7: 7. Menú configuració (Due: 2 de nov. - 8 de nov., LG)
 - Setmana 7: 7. Perfil d'usuari (Due: 2 de nov. - 8 de nov., LG)
 - Setmana 8: + Añada una tarjeta
- Blocked:**
 - + Añada una tarjeta
- Done:**
 - 29 de sep. - 6 de oct.: 1. Setmana 3 (Due: 29 de sep. - 6 de oct., AK)
 - 29 de sep. - 6 de oct.: 3. Mockups. Pantalla mapa. (Due: 29 de sep. - 6 de oct., AK)
 - Setmana 3: 3. Mockups. Pantalla configuració. (Due: 29 de sep. - 6 de oct., AK)
 - Setmana 4: 4A. Disseny BD (Due: 7 de oct. - 13 de oct., LG)
 - Setmana 4: 4A. Esquema BD (Due: 7 de oct. - 13 de oct., LG)
 - Setmana 4: 4B. Control de versions: (Due: 7 de oct. - 13 de oct., AK, LG)
 - Setmana 4: 4B. Descripció Git (Due: 7 de oct. - 13 de oct., LG)
 - Setmana 4: 4B. Instal·lació Git (Due: 7 de oct. - 13 de oct., AK)
 - Setmana 4: 4B. Funcions Git (Due: 12 de oct. - 18 de oct., AK)
 - + Añada una tarjeta



3.3.5. SETMANA 5

Column	Task Description	Due Date	Assignee
To do	6A. Navegabilitat app	26 de oct. - 1 de nov.	AK
To do	6A. Disseny pàgina principal	26 de oct. - 1 de nov.	AK
To do	8B. Lectura BD: usuari	9 de nov. - 15 de nov.	LG
To do	8B. Escriptura BD: usuari	4 de nov. - 10 de nov.	
To do	9. Recycler View	17 de nov. - 23 de nov.	
To do	10A. Disseny general	24 de nov. - 30 de nov.	AK
To do	10B. Escriptura BD: altres	24 de nov. - 30 de nov.	LG
To do	10B. Lectura BD: altres	24 de nov. - 30 de nov.	
To do	11A. Disseny aplicació: styles		
In Progress	5A. Disseny pantalla registre	14 de oct. - 20 de oct.	AK, LG
In Progress	5B. Connexió amb Firebase	14 de oct. - 20 de oct.	B
In Progress	4B. Funcions Git	12 de oct. - 18 de oct.	AK
In Progress	6B. Authentication Firebase	21 de oct. - 27 de oct.	B
In Progress	7. Menú configuració	2 de nov. - 8 de nov.	LG
In Progress	7. Perfil d'usuari	2 de nov. - 8 de nov.	
In Progress	8A. Pantalla recuperació password	9 de nov. - 15 de nov.	AK
In Progress	+ Añada una tarjeta		
Blocked	+ Añada una tarjeta		
Done	3. Mockups. Pantalla configuració.	29 de sep. - 6 de oct.	AK
Done	4A. Diseño BD	7 de oct. - 13 de oct.	LG
Done	4A. Esquema BD	7 de oct. - 13 de oct.	LG
Done	4B. Control de versiones:	7 de oct. - 13 de oct.	AK, LG
Done	4B. Descripció Git	7 de oct. - 13 de oct.	LG
Done	4B. Instal·lació Git	7 de oct. - 13 de oct.	AK
Done	5A. Disseny pantalla login	14 de oct. - 20 de oct.	AK, LG
Done	5A. Disseny splash screen	14 de oct. - 20 de oct.	LG
Done	+ Añada una tarjeta		

The board consists of four columns:

- To do**: Contains tasks for weeks 6 through 11, including "Setmana 6" (6A. Navegabilitat app), "Setmana 7" (7. Menú configuració), "Setmana 8" (8B. Lectura BD: usuari), "Setmana 9" (9. Recycler View), "Setmana 10" (10A. Disseny general), "Setmana 10" (10B. Escriptura BD: altres), "Setmana 10" (10B. Lectura BD: altres), and "Setmana 11" (11A. Disseny aplicació: styles).
- In Progress**: Contains tasks for weeks 6 through 8, including "Setmana 6" (6B. Authentication Firebase), "Setmana 7" (7. Perfil d'usuari), and "Setmana 8" (8A. Pantalla recuperació password).
- Blocked**: Contains a single task: "+ Añada una tarjeta".
- Done**: Contains tasks for weeks 4 through 11, including "Setmana 4" (4B. Control de versiones), "Setmana 4" (4B. Descripció Git), "Setmana 4" (4B. Instal·lació Git), "Setmana 4" (4B. Funcions Git), "Setmana 5" (5A. Disseny pantalla registre), "Setmana 5" (5B. Connexió amb Firebase), "Setmana 5" (5A. Disseny pantalla login), and "Setmana 5" (5A. Disseny splash screen).

Each card includes a due date, a status indicator (orange for In Progress, green for Done), and assigned team members (AK or LG). There are also buttons to add a new card (+ Añada una tarjeta) and edit existing cards (edit icon).

3.3.6. SETMANA 6

To do	In Progress	Blocked	Done
<ul style="list-style-type: none"> Setmana 8 <ul style="list-style-type: none"> 8B. Lectura BD: usuari <ul style="list-style-type: none"> 9 de nov. - 15 de nov. Setmana 9 <ul style="list-style-type: none"> 8B. Escriptura BD: usuari <ul style="list-style-type: none"> 4 de nov. - 10 de nov. Setmana 10 <ul style="list-style-type: none"> 9. Recycler View <ul style="list-style-type: none"> 17 de nov. - 23 de nov. 10A. Disseny general <ul style="list-style-type: none"> 24 de nov. - 30 de nov. 10B. Escriptura BD: altres <ul style="list-style-type: none"> 24 de nov. - 30 de nov. Setmana 10 <ul style="list-style-type: none"> 10B. Lectura BD: altres <ul style="list-style-type: none"> 24 de nov. - 30 de nov. Setmana 11 <ul style="list-style-type: none"> 11A. Disseny aplicació: styles <ul style="list-style-type: none"> 1 de dic. - 7 de dic. 11A. Disseny aplicació: theme <ul style="list-style-type: none"> 1 de dic. - 7 de dic. 	<ul style="list-style-type: none"> Setmana 7 <ul style="list-style-type: none"> 7.Perfil d'usuari <ul style="list-style-type: none"> 2 de nov. - 8 de nov. Setmana 8 <ul style="list-style-type: none"> 8A. Pantalla recuperació password <ul style="list-style-type: none"> 9 de nov. - 15 de nov. Setmana 6 <ul style="list-style-type: none"> 6A.Navegabilitat app <ul style="list-style-type: none"> 26 de oct. - 1 de nov. 		<ul style="list-style-type: none"> Setmana 4 <ul style="list-style-type: none"> 4B.Funcións Git <ul style="list-style-type: none"> 12 de oct. - 18 de oct. Setmana 5 <ul style="list-style-type: none"> 5A.Disseny pantalla registre <ul style="list-style-type: none"> 2 de nov. - 8 de nov. Setmana 5 <ul style="list-style-type: none"> 5B.Connexió amb Firebase <ul style="list-style-type: none"> 14 de oct. - 20 de oct. Setmana 5 <ul style="list-style-type: none"> 5A.Disseny pantalla login <ul style="list-style-type: none"> 2 de nov. - 8 de nov. Setmana 5 <ul style="list-style-type: none"> 5A.Disseny splash screen <ul style="list-style-type: none"> 2 de nov. - 8 de nov. Setmana 6 <ul style="list-style-type: none"> 6B.Authentication Firebase <ul style="list-style-type: none"> 21 de oct. - 27 de oct. Setmana 7 <ul style="list-style-type: none"> 7.Menú configuració <ul style="list-style-type: none"> 2 de nov. - 8 de nov.

3.3.7. SETMANA 7

The image shows a digital task board with four columns: To do, In Progress, Blocked, and Done.

- To do:**
 - Setmana 10
 - 0A. Disseny general
⌚ 24 de nov. - 30 de nov.
AK
 - 0B. Escriptura BD: altres
⌚ 24 de nov. - 30 de nov.
LG
 - Setmana 10
 - 0B. Lectura BD: altres
⌚ 24 de nov. - 30 de nov.
 - Setmana 11
 - 1A. Disseny aplicació: styles
⌚ 1 de dic. - 7 de dic.
 - Setmana 11
 - 1A. Disseny aplicació: theme
⌚ 1 de dic. - 7 de dic.
AK
 - Añada una tarjeta
- In Progress:**
 - Setmana 7
 - 7.Perfil d'usuari
⌚ 2 de nov. - 8 de nov.
 - Setmana 6
 - 6A.Disseny pàgina principal
⌚ 26 de oct. - 1 de nov.
AK
- Blocked:**
 - + Añada una tarjeta
- Done:**
 - 8A. Pantalla recuperació password
⌚ 9 de nov. - 15 de nov.
AK
 - Setmana 8
 - 8B. Escriptura BD: usuari
⌚ 4 de nov. - 10 de nov.
AK
 - Setmana 8
 - 8B. Lectura BD: usuari
⌚ 9 de nov. - 15 de nov.
AK
 - Setmana 9
 - 9A. Menus
⌚ 17 de nov. - 23 de nov.
LG
 - Setmana 9
 - 9. Recycler View
⌚ 17 de nov. - 23 de nov.
B
LG

To do

13C. Documentació codi / javadoc
⌚ 15 de dic. - 21 de dic.

14A. Millora accesibilitat
⌚ 22 de dic. - 28 de dic.

14A. Millora usabilitat
⌚ 22 de dic. - 28 de dic.
AK

14B. Testeig final
⌚ 22 de dic. - 28 de dic.
LG

14C. Possibles versions noves
⌚ 22 de dic. - 28 de dic.

15 Presentació final
+ Añada una tarjeta

In Progress

7A. Navigation Graph
⌚ 8 de nov. - 15 de nov.
AK B LG

+ Añada una tarjeta

Blocked

+ Añada una tarjeta

Done

8B. Lectura BD: usuari
⌚ 9 de nov. - 15 de nov.
AK LG

9A. Menus
⌚ 17 de nov. - 23 de nov.
LG

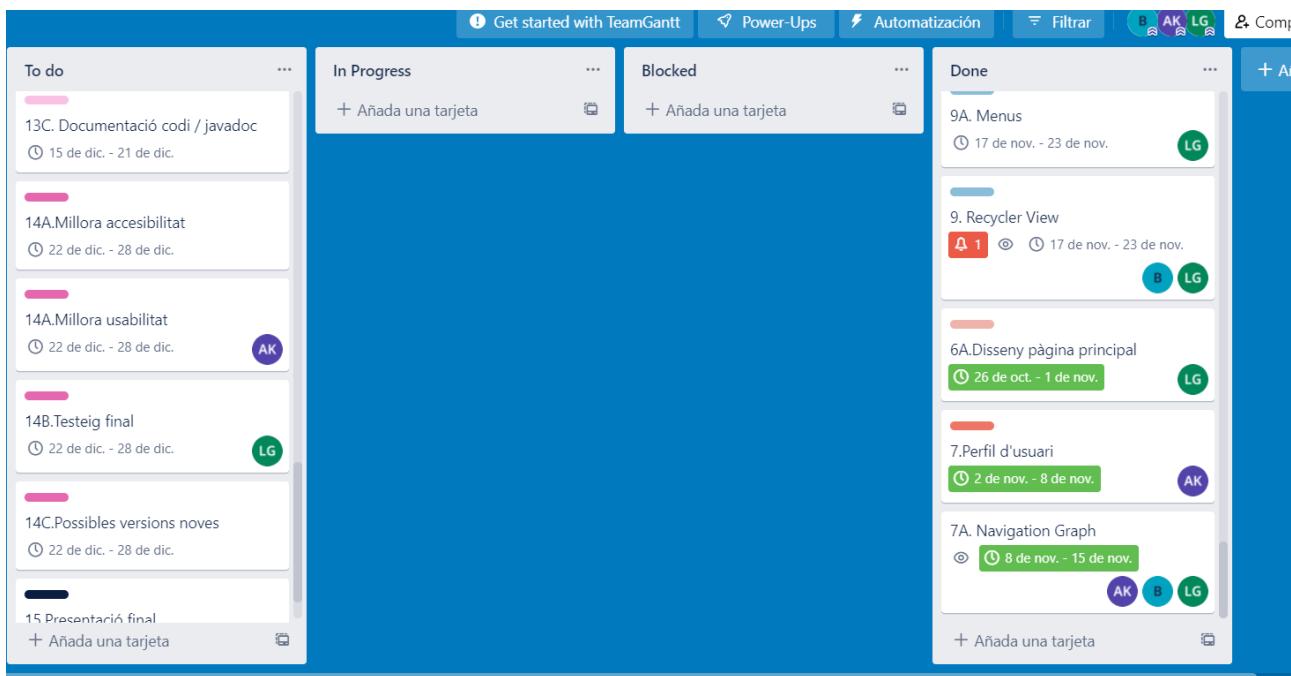
9. Recycler View
⌚ 17 de nov. - 23 de nov.
A 1 B LG

6A. Disseny pàgina principal
⌚ 26 de oct. - 1 de nov.
LG

7.Perfil d'usuari
⌚ 2 de nov. - 8 de nov.
AK

+ Añada una tarjeta

+ Añada otra lista



3.3.8. SETMANA 8

Este tablero está configurado como público, pero los administradores del tablero pueden cambiar la visibilidad en cualquier momento. [Más información aquí](#)

Planificación Público Tablero

Get started with TeamGantt Power-Ups Automatización Filtrar Compartir

To do In Progress Blocked Done

1. Geolocalitzacio 2. Animacions 3. Informes 4. Grafics

Setmana 12 Setmana 12 Setmana 12 Setmana 12

Setmana 9 6A. Disseny pàgina principal 9. Recycler View

Setmana 6 7A. Navigation Graph 7. Perfil d'usuari

Setmana 12 8A. Recycler Views con Safe Args

3.3.8. SETMANA 9 i 10

This screenshot shows a Scrum board with four columns: To do, In Progress, Blocked, and Done.

- To do:**
 - Setmana 10: 10A. Disseny general (AK, 24 de nov. - 30 de nov.)
 - Setmana 11: 11A. Disseny aplicació: styles (1 de dic. - 7 de dic.)
 - Setmana 11: 11A. Disseny aplicació: theme (1 de dic. - 7 de dic., AK)
 - Setmana 11: 11B. Imatges a firebase (1 de dic. - 7 de dic., LG)
 - Setmana 12: 12A. Animacions (8 de dic. - 14 de dic.)
- In Progress:**
 - + Añada una tarjeta
- Blocked:**
 - + Añada una tarjeta
- Done:**
 - Setmana 9: 9A. Menus (17 de nov. - 23 de nov., LG)
 - Setmana 10: 10B. Escriptura BD: altres (24 de nov. - 30 de nov., AK)
 - Setmana 10: 10B. Lectura BD: altres (24 de nov. - 30 de nov., AK)
 - Setmana 12: 12A. Geolocalitzacio (8 de dic. - 14 de dic., B)

3.3.9. SETMANA FINAL

This screenshot shows a Scrum board with four columns: To do, In Progress, Blocked, and Done.

- To do:**
 - + Añada una tarjeta
- In Progress:**
 - Setmana 13: 13C. Documentació codi / javadoc (15 de dic. de 2022 - 21 de dic. de 2022)
 - Setmana 14: 14B. Testeig final (22 de dic. de 2022 - 28 de dic. de 2022, LG)
 - Setmana 15: 15. Presentació final (30 de dic. de 2022 - 30 de dic. de 2022, AK, LG)
- Blocked:**
 - + Añada una tarjeta
- Done:**
 - Setmana 13: 13A. Internaciolització (15 de dic. de 2022 - 21 de dic. de 2022, AK, LG)
 - Setmana 14: 14A. Millora accesibilitat (22 de dic. de 2022 - 28 de dic. de 2022, AK)
 - Setmana 14: 14A. Millora usabilitat (22 de dic. de 2022 - 28 de dic. de 2022, AK)

4. DISSENY

4.1. MOCKUPS

4.1.1 PANTALLES

- **Splash screen:** pantalla de càrrega amb el color corporatiu i el logo en gran.

 <p>Iniciar sessió</p> <p><input type="text"/> Usuari</p> <p><input type="password"/> Contrasenya</p> <p>Reestableix la teva contrassenya.</p> <p><input type="checkbox"/> Recorda la teva contrassenya</p> <p>→</p> <p>Nou usuari? Regista't.</p>	<p>Pantalla inici de sessió</p> <p>Característiques:</p> <ul style="list-style-type: none">- Iniciar sessió amb usuari i contrasenya- Funció per restablir la contrasenya en cas d'oblidar-la- Botó per mantenir la sessió oberta sempre- Enllaç a la pantalla de creació d'usuari
--	--

← Registra't

Nom

Cognoms

Data de naixement:

12 - 5 - 2014



Correu electrònic

Contrasenya

Confirma contrasenya

Codi Postal

CIP

Accepta els termes de privacitat?

Condicions d'ús



Pantalla de registre

Característiques:

- Creació d'usuari
- Nom
- Cognoms
- Data naixement
- Correu electrònic
- etc
- Termes de privacitat
- Condicions d'ús
- Botó per regressar a l'anterior pantalla en cas d'haver-se equivocat

≡ Home

Dilluns, 3 d'octubre del 2022



CALENDARI



FAMILIARS



FARMÀCIES



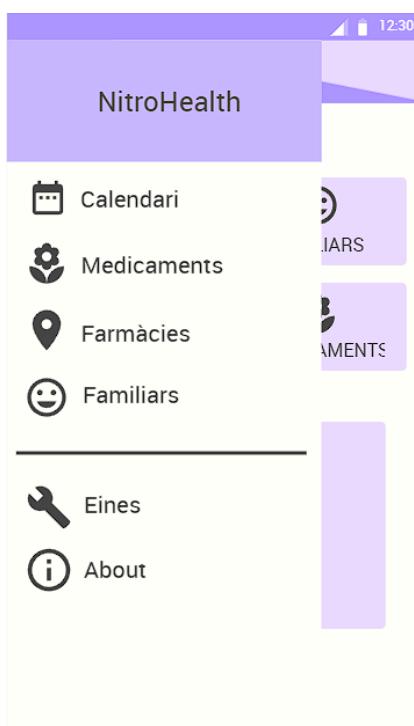
MEDICAMENTS

resum diari

Pantalla home

Característiques:

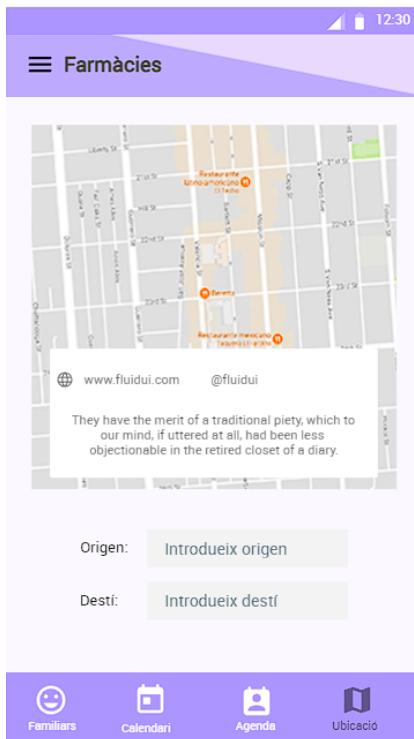
- Petita introducció amb data actual / nom usuari / etc
- Botons a les funcionalitats principals de l'app
- Slider amb un resum diari de què s'ha de fer / resum mensual amb tota la informació



Menú “burger”

Característiques:

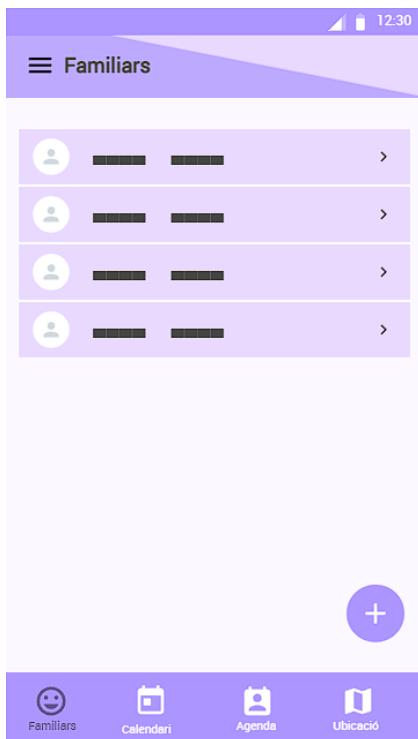
- Enllaços a totes les pàgines
- Enllaç a l'apartat d'eines de l'app
- About amb les condicions d'ús / la nostra missió
- Clicant el logo es retorna a la home



Pantalla farmàcies/mapa

Característiques:

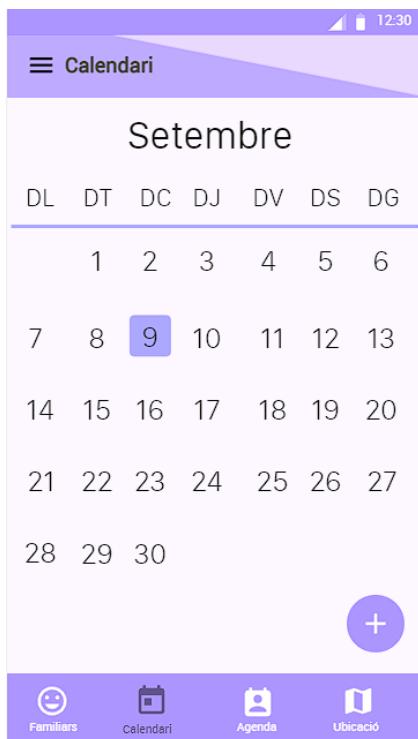
- Mapa amb la ubicació actual
- El mapa amb un filtre per veure farmàcies / CAPs / hospitals / centres mèdics propers
- Indicacions cap a aquests llocs



Pantalla familiars/contactes

Característiques:

- Llista amb contactes de l'usuari
- Ordenats tal com vulgui (capacitat de moure la llista)
- Creació de nous contactes o importació directa des del mòbil



Pantalla calendari

Característiques:

- Calendari amb el mes actual
- Els dies amb consultes mèdiques ressaltats depenen de la causa
- Informació del Departament de Salut de quan comença la temporada de vacunació etc
- Creació de noves dates importants
- Recordatori de compra de medicaments

Sat 5 Sep	6	56
Sun 6 Sep	7	57
Mon 7 Sep	8	58
Tue 8 Sep	9	59
Wed 9 Sep	10	00
Thu 10 Sep	11	01
Fri 11 Sep	12	02
Sat 12 Sep	13	03
Sun 13 Sep	14	04

Tipus visita

Nom centre sanitari

CANCEL-LAR GUARDAR

Pantalla creació calendari

Característiques:

- Creació d'una cita mèdica al calendari
- Tipus de visita: revisió, vacunació, consulta, etc...
- Escollar centre sanitari

DL DT DX DJ DV DS DM

PARACETAMOL

SINTRON

AMRINONE

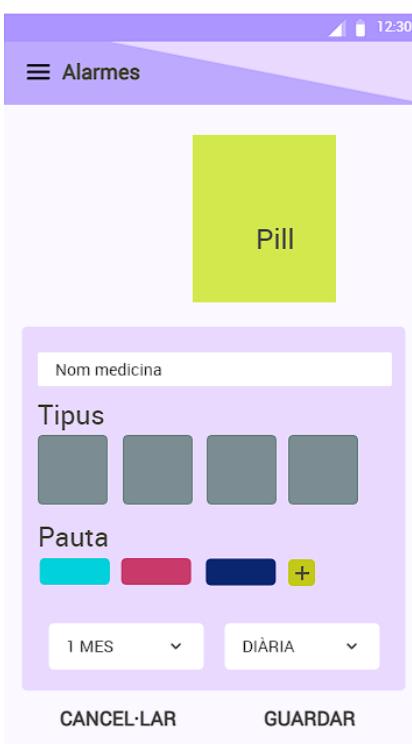
+

Familiars Calendari Agenda Ubicació

Pantalla medicaments

Característiques:

- Segons el dia informació de quins medicaments s'ha de prendre
- A cada medicament tags de l'hora a la qual s'ha de prendre (matí/tarda, abans de dinar, després de, etc.)
- Dibuix del medicament per facilitar comprensió
- Creació de nous medicaments



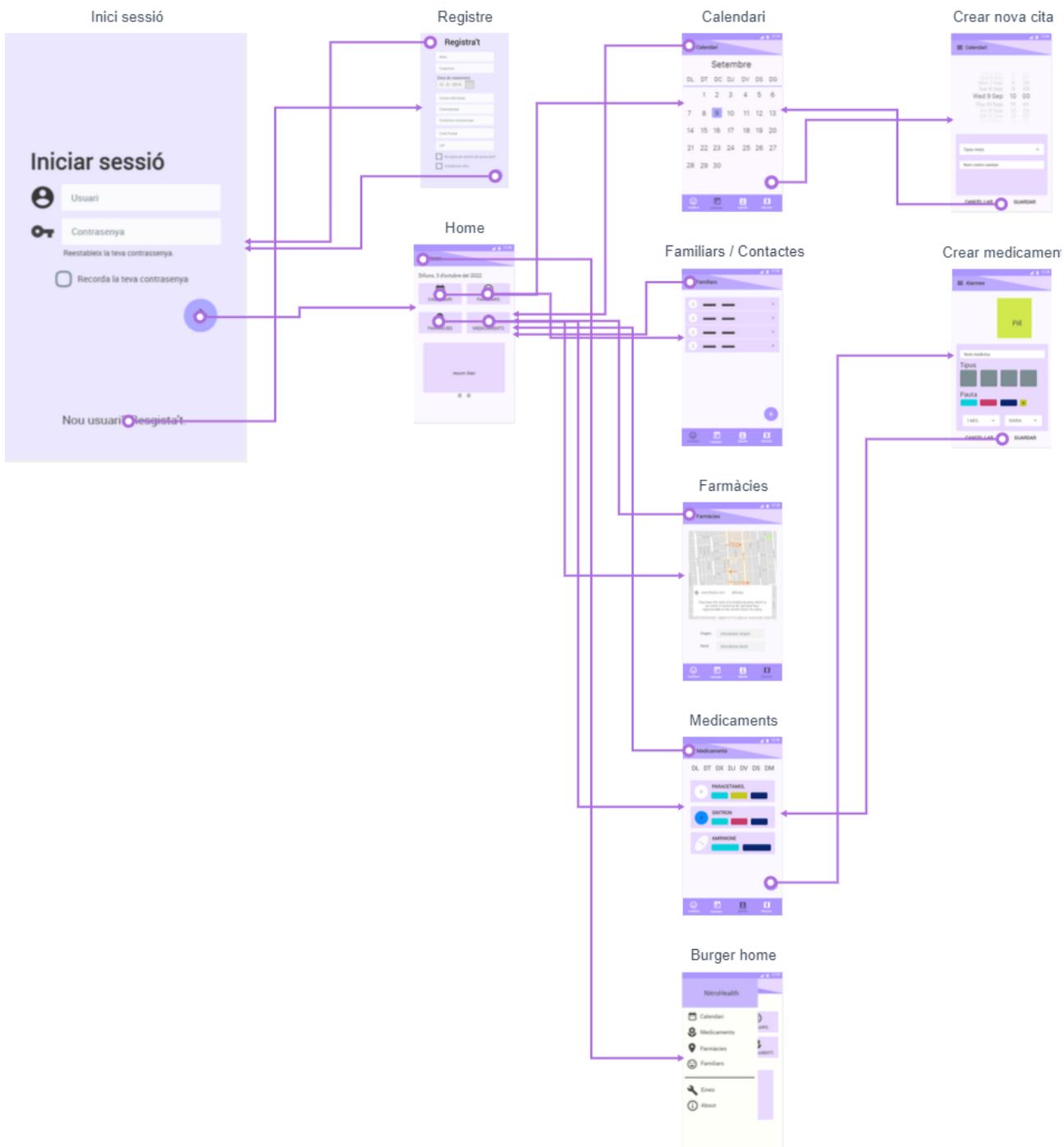
Pantalla creació medicaments

Característiques:

- Creació de nou medicament a prendre:
- Nom
- Tipus (pastilla, solució, etc).
- Pauta: cada quan
- Escol·lir durant quant de temps i si és diari o no

4.1.2. NAVEGACIÓ

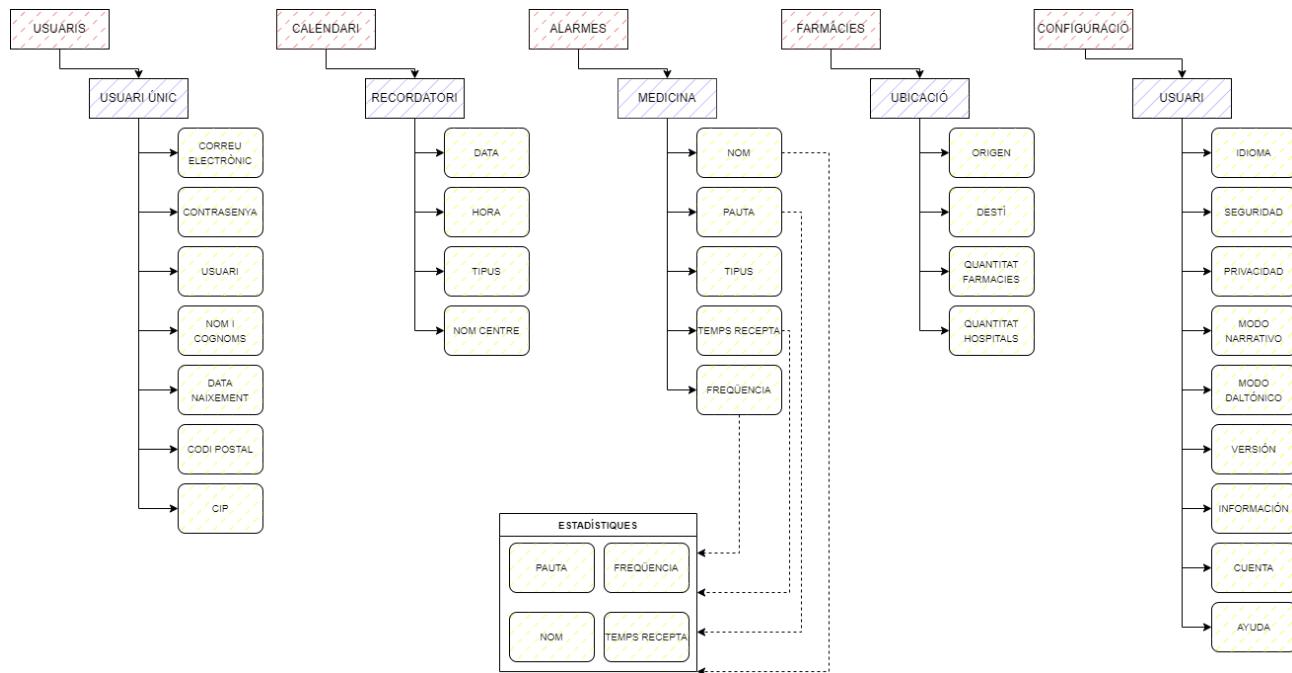
4.1.2.1. Diagrama de navegació



4.1.2.2. Prototipus Marvel

Enllaç al prototip fet a Marvel: <https://marvelapp.com/prototype/71ha0fe>

4.2. DISSENY BASE DE DADES NO-SQL



Per començar, tenim la taula d'usuaris que correspon a les dades que introduïm a la pantalla de registre i les dades que es buscarien a l'hora que l'usuari inici sessió.

La taula de calendari guarda la data en la qual l'usuari introduceix un recordatori per a la medicina.

La pantalla d'alarms recull les dades de les medicines que l'usuari va introduint, i depenent de la freqüència de la medicina, l'aplicació posarà més o menys alarms. En conjunt amb aquesta pantalla, hi ha les estadístiques que apareixen a la pantalla principal del menú, on fa una petita taula amb les estadístiques de les medicines que s'han pres, la freqüència i el temps total.

La pantalla de les farmàcies recull la ubicació de l'usuari per proporcionar una aproximació de farmàcies i hospitals propers.

La pantalla de configuració és la típica on està tota la informació de l'usuari amb la seva seguretat, privacitat i opcions personalitzades, tant com una opció d'ajuda per si tenen preguntes sobre l'aplicació.

5. DESENVOLUPAMENT DE L'APP

5.1. CONTROL DE VERSIONS

- **Què és el Git. Descripció breu. Característiques.**

Git és un projecte de codi obert com un sistema de control de versions, té com a característiques la ramificació, fusió o comparació de versions d'un projecte/arxiu/bbdd/etc...

Amb Git, podem controlar les versions amb total seguretat, i també té un sistema per mantenir un historial complet per a cada una de les versions del projecte.

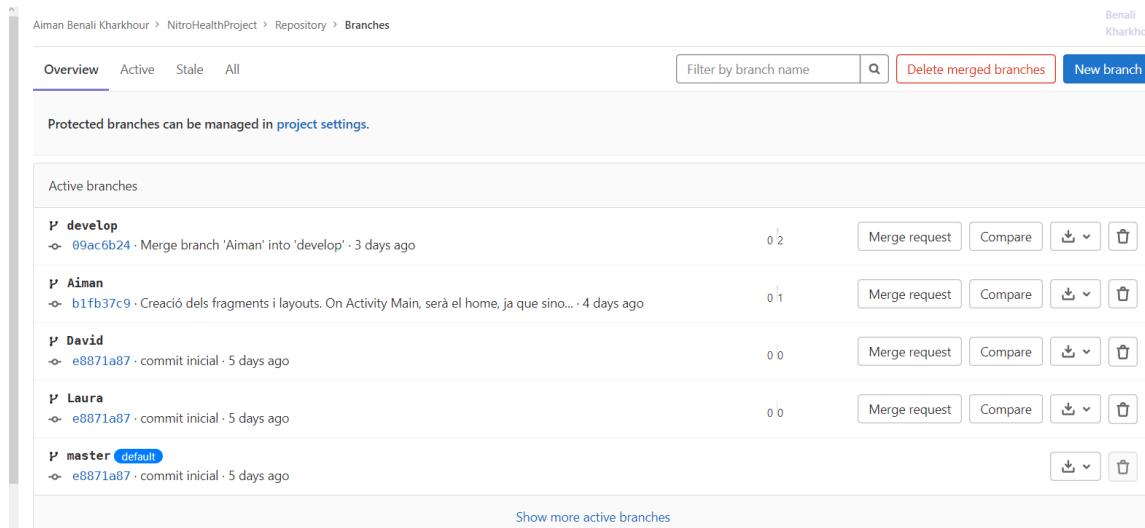
- **Què és el Gitlab. Descripció breu. Característiques.**

Gitlab és una plataforma on han treballat junts Git amb DevOps, i aquesta també és un servei de control de versions tant com per supervisar, fer proves i gestionar errors.

Una de les característiques del GitLab és que poden haver-hi diferents persones treballant en un projecte, i permet la gestió de permisos i accés dels usuaris que accedeixen als arxius. També compta amb un historial i una interície gràfica que fa molt fàcil la navegació per la plataforma.

Per poder treballar junts de manera progressiva hem aplicat el git en el nostre projecte, d'aquesta manera els treballs tindran més progressos i millor qualitat. Nosaltres ens hem organitzat un repositori on farem tots els push del commits en la branca "develop", així, només haurem de fer un merge i amb el permís dels companys i que tinguin el treball (sense fer purge del seu codi ja creat sense commitar) fet fins ara fent pull en l'app.

S'han creat un total de 3 usuaris (Aiman, Laura, David), i aquests serem els developers i desenvoluparem el projecte d'aquesta manera.

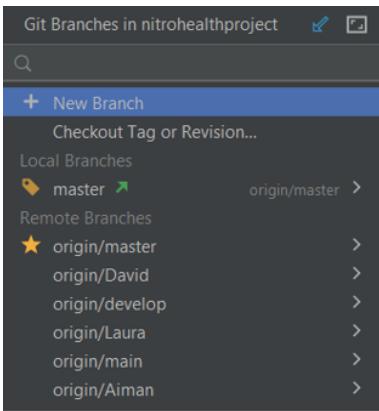


The screenshot shows the 'Branches' page in Gitlab. At the top, there are navigation tabs for 'Overview', 'Active', 'Stale', and 'All'. Below these are buttons for 'Filter by branch name' (with a search icon), 'Delete merged branches', and 'New branch'. A note says 'Protected branches can be managed in project settings.' The main section is titled 'Active branches' and lists four branches:

- develop**: Last commit was 3 days ago by Aiman (commit 09ac6b24). Actions: Merge request, Compare, More, Delete.
- Aiman**: Last commit was 4 days ago by Aiman (commit b1fb37c9). Actions: Merge request, Compare, More, Delete.
- David**: Last commit was 5 days ago by David (commit e8871a87). Actions: Merge request, Compare, More, Delete.
- Laura**: Last commit was 5 days ago by Laura (commit e8871a87). Actions: Merge request, Compare, More, Delete.

At the bottom of the list, there is a link 'Show more active branches'.

Així es veurien les branques des de l'Android studio



Descripció de les funcions:

- Add → Afegeix els canvis dels fitxers al repositori abans de fer el commit.
- Commit → Tots els canvis en els fitxers del projecte es queden guardats al repositori local i han de venir amb un comentari/descrició del que s'ha fet per fer un seguiment dels canvis.
- Push → Després del commit fem un push perquè s'actualitzin els canvis al GitLab.
- Pull → Recull tots els canvis d'una branca i els importa per que es pugui treballar sobre els canvis que s'han fet per altres persones.
- Clone → Copia una branca en una altre exactament igual en el ordinador en funció.
- Revert → Desfà els canvis d'un commit.
- Merge → Uneix dues branques en una per a que aquesta pugui rebre els canvis realitzats i ajuntar les dues versions.
- Rebase → Substitueix una branca amb els canvis des d'on fas el rebase, ajunta les dues versions i elimina la branca d'origen.

Diferències entre:

pull i push: El pull recull els canvis fets al repositori i els importa en el teu projecte. El push puja els canvis del repositori a la seva branca corresponent.

merge i rebase: El rebase el que fa és substituir la branca en la que volem fer el rebase, sense tenir en compte els canvis que hi hagi en comparació des d'on es fa el rebase, en canvi el merge uneix els canvis de dues branques sense eliminar res.

Branques:

Les branques son els diferents camins d'un projecte, entorns on pots treballar independentment dels canvis que faci alguna altre persona, per fer proves i per tenir més organització en general .

Amb el tema de les branques tenim en total → master, Aimana, Laura, David i develop.

D'aquesta manera, totes les feines que anem fent durant el transcurs, poden ser commitats des d'on l'usuari pertany.

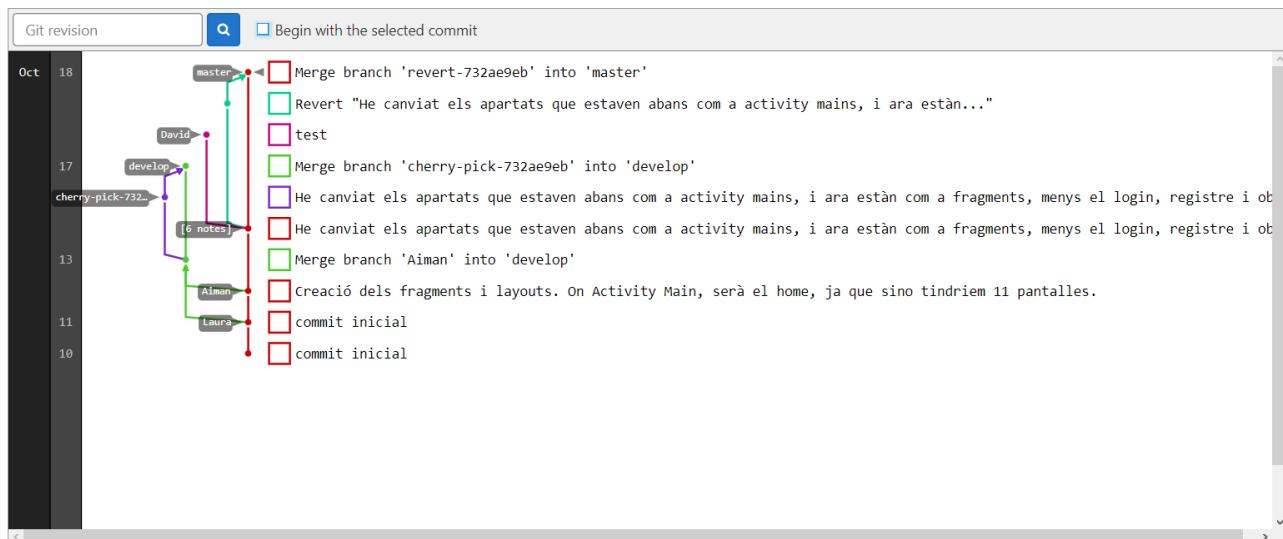
Les branques personals (David,Aiman,Laura), són per fer els commits per després fer un merge request amb el develop un cop ja repassat.

Develop, serà la branca comú on tots el canvis produïts pels usuaris desenvolupant el projecte, pujaran els canvis, i a la vegada baixaran els canvis dels companys, acceptant un merge request (sense trencar cap estructura local de l'ordinador).

El master l'utilitzarem per al projecte final una vegada ja tinguem totes les pantalles fetes amb el seu funcionament i no hi hagin problemes ni errors.

Historial de versions:

L'historial de versions ho mostrem en una gràfica on es pot veure tots els push que s'han fet durant la creació del projecte, així d'aquesta manera, queda tot més visual i comprensiu per l'enteniment lògic. Es veurà que tots els canvis acaben sempre en develop, de moment només n'hi ha un canvi fet, que és la creació de les pantalles, però mitjançant avancem en el projecte es veuen tots els progrès fets en la branca de develop. Sempre podrem tornar a una versió anterior fent un revert en cas d'un error o incident.



Merge request del gitlab:

Merge és una petició que fa l'usuari que ha fet commit i push del projecte i aquests canvis es necessiten aplicar en la rama de develop, sacrificant un treball en concret, sempre aquesta petició es fa perquè l'usuari que vagi a acceptar el merge sapgui que alguna feina que pot ser que hagi fet ell, pot ser anulada i substituïda per al projecte pujat. Per això se li diu merge request, ja que per fer-ho es necessita aprovació dels usuaris que treballen en el projecte.

Issues:

Nosaltres tractem amb els issues de manera que llegim l'avís que ens dona de per què el projecte en cas d'entrega fallaria, o donaria un error que no accepta. Aleshores, una vegada ja llegit l'error, ens comuniquem per veure qui pot tractar amb l'error i poder corregir, per a què així tots podem acceptar el merge (acceptant el trencament del treball en local no commitat).

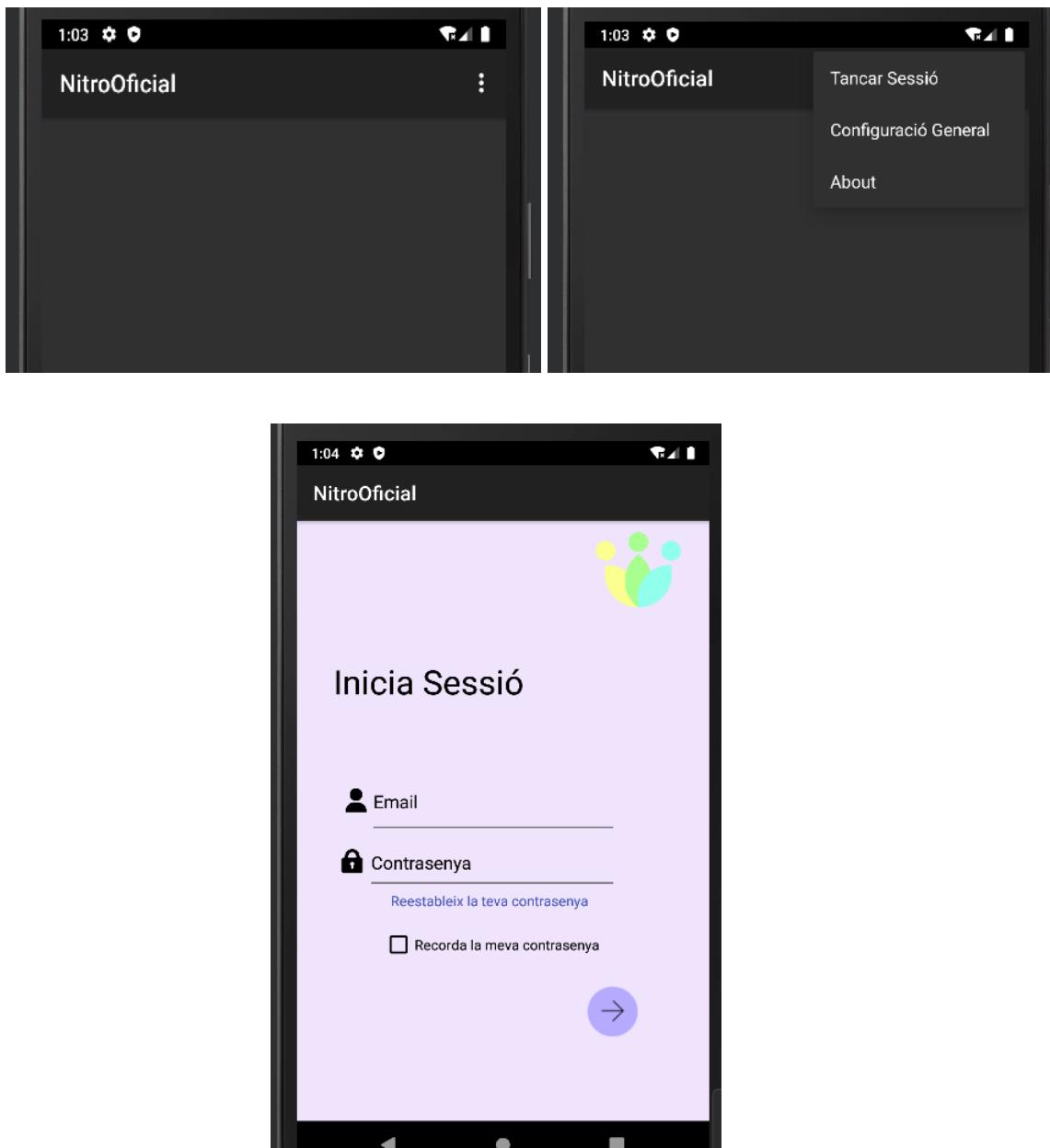
D'aquesta manera ens assegurem tots de què el projecte que tenim en comú segueix funcionant correctament.

5.2. ESTRUCTURA DE PROJECTE (PACKAGES)

5.3. CODIFICACIÓ

5.3.1 NAVEGABILITAT. MENÚS

Hem implementat un menú on ens portarà a les diferents pantalles de la nostre aplicació, el tancar sessió ens porta a l'inici de sessió, la opció de configuració general ens porta a la pantalla de configuració de l'aplicació, i l'about ens porta a la informació general. (De moment la configuració general i l'about no l'hem fet perquè son fragments i no ens porta en lloc, però el de tancar sessió com es un activity si que ho hem pogut fer).



5.3.2. SPLASH SCREEN

5.3.3. BASE DE DADES FIRESTORE DE FIREBASE

<https://git.copernic.cat/benali.kharkhour.aiman/nitrohealthproject>

- **Authentication**

- L'autentificació de Firebase té com a finalitat gestionar els usuaris a través dels seus comptes de correu, en el nostre cas comptes de Google. Gràcies a ella podem connectar la nostra aplicació Android a la base de dades i podem gestionar els comptes d'usuaris que treballaran amb aquesta.

- **Cloud Firestore**

- És una base de dades NoSQL allotjada al núvol, s'utilitza per guardar l'informació de la teva aplicació Android o Apple. Aquesta base de dades com és NoSQL funciona amb col·leccions. Una col·lecció seria com una carpeta/contenidor on tu guardaries diferents documents que serien les dades que vols guardar de la teva aplicació. Una vegada creada la teva estructura de col·leccions i documents, Cloud Firestore et dóna la possibilitat de fer consultes a la teva informació aplicant diferents filtres per trobar exactament les dades que has guardat. És una molt bona opció per protegir les teves dades de la teva aplicació i poder accedir-hi en línia.

- **Cloud Storage**

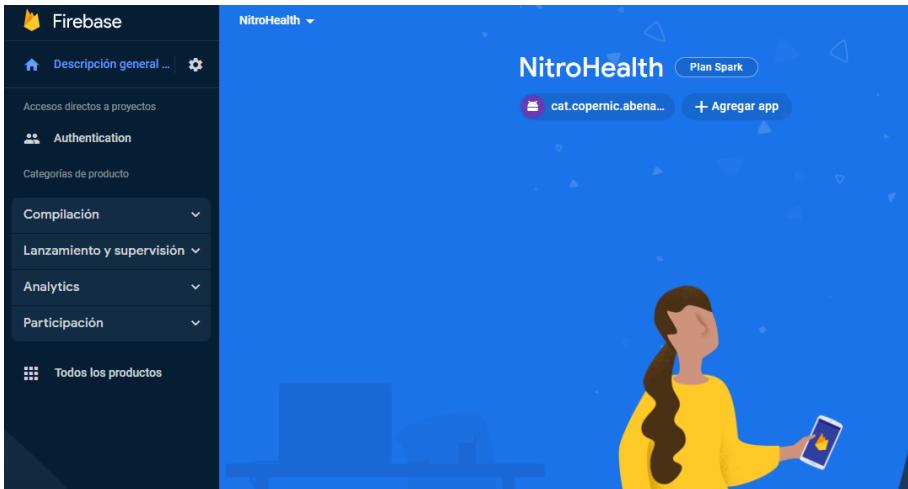
- És una forma de guardar de forma remota els teus documents i dades (al núvol), actualment moltíssimes empreses confien en aquest mètode, ja que es despréocupen de tindre un servidor local, que pot fallar en qualsevol moment, on guardar l'informació. És molt més ràpid i efectiu tindre la teva informació en un repositori en línia perquè així tothom que tingui accés pot accedir d'una forma segura i ràpida. Amb Cloud Storage, pots guardar qualsevol informació (fotos, arxius, documents, etc.) en una quantitat il·limitada d'espai que s'encarrega de gestionar el proveïdor de l'aplicació i tota aquesta informació que es puja estan protegides per claus d'encriptació i sofisticades mesures de seguretat, com he comentat, és una forma molt segura i ràpida d'accendir i guardar l'informació.

- **Cloud Messaging**

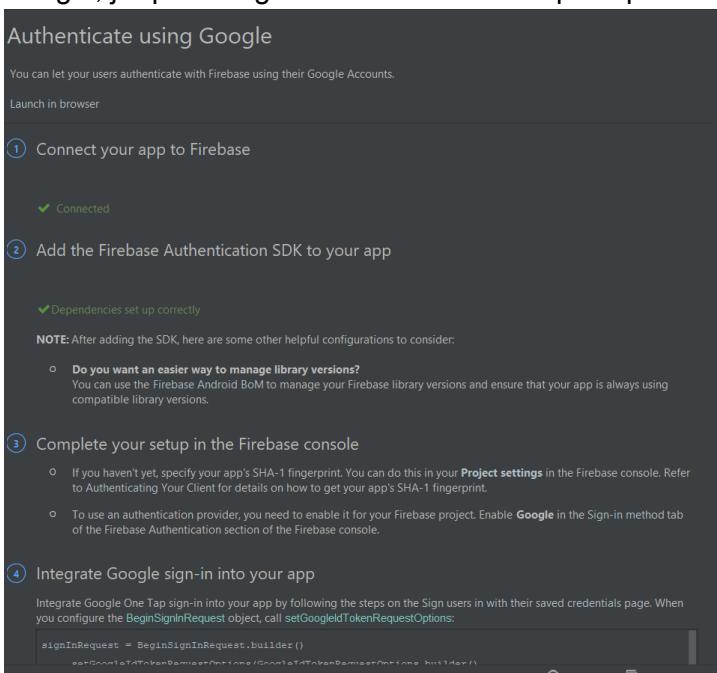
- És una solució de missatgeria multiplataforma que et permet enviar missatges de forma segura i sense cost. Principalment, s'utilitza per enviar missatges a través de notificacions de l'app o directament al correu de l'usuari, perquè usuaris envíin missatges des de la seva aplicació al nostre servidor i per distribuir els missatges, és a dir, per trametre missatges a un grup d'usuaris específics o a un usuari de forma individual per alguna acció en específic dins de l'aplicació. Cloud Messaging funciona amb un entorn de confiança de generació, segmentació i enviament de missatges i una app client que serà el destinatari que rebrà els missatges.

5.3.3.1. CONNEXIÓ AMB LA BASE DE DADES

- Per començar, anirem a la base de dades Firebase i crearem una nova base de dades al domini de l'institut Nicolau Copèrnic amb el nom de “NitroHealth”. Una vegada creat arribem a aquest menú on podrem gestionar la base de dades.



- Abans de fer res connectarem la nostra aplicació amb la base de dades, anem a l'Android studio i seleccionem l'opció Firebase, dintre de les diferents opcions que ens ofereix l'aplicació, seleccionarem “Authentication” i ens autentificarem a través del compte de Google, ja que Google és el nostre domini principal.



- Una vegada connectada l'aplicació amb la base de dades, anirem a Firebase a la configuració del teu projecte i comprovarem que ens hem connectat a la perfecció en

l'apartat de “Tus apps”.

Tus apps

Configuración del SDK

¿Necesitas volver a configurar los SDK de Firebase en tu app? Revisa las instrucciones de configuración del SDK o descarga el archivo de configuración con las claves y los identificadores de tu app.

[Ver las instrucciones del SDK](#) [google-services.json](#)

ID de la app ⓘ 1:967570066917:android:c30c9e2ab172927f17b30c

Sobrenombre de la app Agregar un sobrenombre

Nombre del paquete cat.copernic.abenali.nitrohealthproject

Huellas digitales del certificado SHA ⓘ Tipo ⓘ

[Agregar huella digital](#)

[Quitar esta app](#)

- A continuació gestionarem els usuaris i els seus permisos, dins de la configuració de Firebase seleccionarem la pestanya de “Usuarios i permisos”. Aquí és on podrem agregar usuaris i donar-li permisos

Configuración de proyecto

General Cloud Messaging Integraciones Cuentas de servicio Privacidad de los datos Usuarios y permisos Verificación de aplicaciones

Buscar miembros Agregar miembro

Miembro	Funciones
David Bel Ortega bel.ortega.david@alumnat.copernic.cat	Propietario ⓘ

4 cuentas de servicio también tienen acceso a este proyecto

- Agregem tots els usuaris del projecte i li donem els permisos de propietari

Direcciones de correo electrónico

Propietario ⓘ

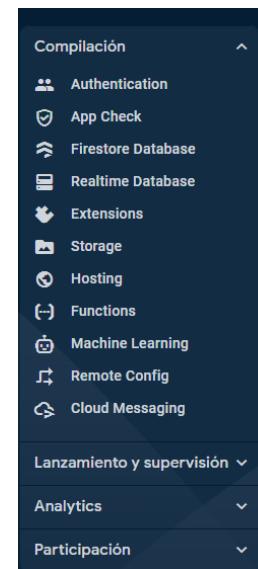
Cancelar Agregar miembro

[Agregar miembro](#)

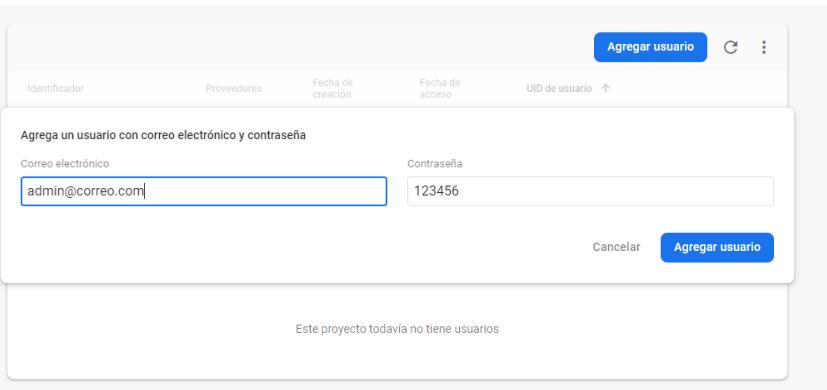
Miembro ↑	Funciones
 benali.kharkhour.aiman@alumnat.copernic.cat	Propietario 
 David Bel Ortega bel.ortega.david@alumnat.copernic.cat	Propietario 
 ivan.canton@copernic.cat	Propietario 
 lindsay.barrero@copernic.cat	Propietario 
 roldan.garcia.laura@alumnat.copernic.cat	Propietario 

[4 cuentas de servicio](#) también tienen acceso a este proyecto

- Per gestionar el tema d'usuaris registrats dintre de l'app, en el menú principal de Firebase, dintre de "Compilació" tenim diverses opcions per fer ajustos a la nostra base de dades. Per poder gestionar la informació dels usuaris que es registraran a la nostra aplicació, seleccionarem l'opció "Authentication"



- Tots els usuaris que es registrin a la nostra aplicació sortiran aquí, el mètode implementat és per correu electrònic, també podem crear a mà els usuaris seleccionant



Agrega un usuario con correo electrónico y contraseña

Correo electrónico: admin@correo.com

Contraseña: 123456

Cancelar Agregar usuario

Este proyecto todavía no tiene usuarios

l'opció "Agregar usuario".

- La comanda des de l'aplicació seria així

```
createuser( email: "pedro@gmail.com" , password: "23432543543")
```

5.3.3.2. REGISTRE I ACCÉS A L'APP

- PANTALLA LOGIN I PANTALLA REGISTRE

The image shows two side-by-side screenshots of a mobile application's login and registration screens.

Login Screen ('Inicia Sessió'):

- Header: 'Inicia Sessió'
- Fields:
 - User icon: 'Usuari' (Email) - Text input field
 - Lock icon: 'Contraseña' (Password) - Text input field
 - Text link: 'Reestableix la teva contraseña'
 - Checkboxes:
 - 'Recorda la meva contraseña'
- Large blue arrow button with a white circle containing a right-pointing arrow.
- Text at the bottom: 'Nou usuari? [Registra't](#)'

Registration Screen ('Registra't'):

- Header: 'Registra't' with a back arrow icon.
- Fields:
 - Text input fields for 'Nom' (Name), 'Cognoms' (Last Name), 'DD/MM/AAAA' (Date of Birth), 'Correu electrònic' (Email), 'Contraseña' (Password), 'Repeteix contraseña' (Repeat Password), 'Codi Postal' (Postal Code), and 'CIP'.
 - Checkboxes:
 - 'Accepto els termes de privacitat' (I accept the privacy terms)
 - 'Condicions d'ús' (Usage conditions)
- Large blue arrow button with a white circle containing a right-pointing arrow.

- Per poder fer la pantalla de Login i la pantalla de registre hem utilitzat la base de dades Firebase. Una vegada feta la connexió entre la nostra aplicació i la base de dades, hem hagut de programar la funcionalitat dintre del Main Activity de cada pantalla.

El que fa aquesta funcionalitat, simplement és agafar l'informació dels camps on escriu l'usuari de la pantalla de registre, comprovar que els ha posat de forma correcta, i finalment enviar l'informació a la base de dades perquè l'agafi i creï l'usuari. Una vegada creat, en la pantalla de login, haurà de posar el seu nom i la seva contrasenya, si s'ha creat correctament i l'informació posada és correcta entrarà dins de l'aplicació, si l'informació no és correcta o l'usuari no s'ha creat correctament, donarà error i haurà de corregir i tornar a iniciar sessió.

- Captura codi pantalla login

```
class InicioUsuario : AppCompatActivity() {

    private lateinit var auth: FirebaseAuth
    private lateinit var binding: ActivityInicioUsuarioBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        auth = FirebaseAuth.getInstance()
        binding = ActivityInicioUsuarioBinding.inflate(layoutInflater)
        setContentView(binding.root)
        auth = FirebaseAuth.getInstance()

        binding.botonConfirmarLogin.setOnClickListener { it: View! ->
            var email = this.binding.emailInicio.text.toString().trim()
            var password = this.binding.ContraInicio.text.toString().trim()
            if (email.isNotEmpty() & password.isNotEmpty()) {
                signIn(email, password)
            } else {
                Toast.makeText(baseContext, text = "No has omplert tots els camps!", Toast.LENGTH_SHORT)
                    .show()
            }
        }
        val Registrarse = findViewById<TextView>(R.id.RegistratLink)

        Registrarse.setOnClickListener{ it: View!
            val intent = Intent(packageContext: this, Registro::class.java)
            startActivity(intent)
            finish()
        }
    }
}
```

Per començar, tenim els “private lateinit var auth: FirebaseAuth” (la qual declara una instància del Firebase per mes endavant inicialitzar la base de dades a la funció OnCreate) i “private lateinit var binding: ActivityInicioUsuarioBinding”(per donar-li un valor a una variable amb el valor que vulguem).

La funció onCreate es on tenim totes les funcions principals que farà el nostre login. Iniciem la base de dades i el binding per a continuació agafar el botó de confirmar i fer un setOnClickListener que conté les variables i els if corresponents per fer el Login.

Creem els bindings per l'email i el password, fem un if per saber quan estan els camps buits o no, si no estan buits, inicia sessió amb l'usuari que has fet o farás al registre, si estan buits, dóna un missatge de que els camps estan buits.

A continuació tenim el setOnClickListener del text de registre :

Nou usuari? [Registra't](#)

Que envia a l'usuari a la pantalla de registre quan li donen clic.

```
private fun signIn(email: String, password: String) {
    auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                Log.d(Registro.TAG, msg: "signInUserWithEmail:success")
                val user = auth.currentUser
                updateUI(user)
            } else {
                Log.w(Registro.TAG, msg: "signInUserWithEmail:failure", task.exception)
                Toast.makeText(
                    baseContext, text: "Authentication failed.", Toast.LENGTH_SHORT
                ).show()
                updateUI(user: null)
            }
        }
}

confirmaUsuario()

private fun updateUI(user: FirebaseUser?) {
    val intent = Intent(packageContext: this, MainActivity::class.java)
    startActivity(intent)
    finish()
}

private fun confirmaUsuario() {
    var usuario = auth.currentUser
    if(usuario != null){
        Log.w(Registro.TAG, msg: "USUARIO INFO: ${usuario.email}")
    }
}
```

La funció de signIn recull els valors de email i password per fer la autentificació de l'usuari, si està a la base de dades donarà un “Success” sinó donarà un “Authentication failed” i hauràs de posar un usuari vàlid.

La funció de updateUI el que fa és canviar de pantalla quan l'usuari es confirmat i li donem al botó.

I finalment per la codificació del Login, tenim la funció de confirmarUsuari, que mostra al Log el correu del usuari per confirmar que és correcte i està a la base de dades.

Captura codi Pantalla registre

```
class Registro : AppCompatActivity() {
    companion object {
        internal const val TAG = "EmailPassword"
    }
    private lateinit var auth: FirebaseAuth
    private lateinit var binding: ActivityRegistroBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_registro)
        //FirebaseApp.initializeApp(this)
        auth = Firebase.auth
        binding = ActivityRegistroBinding.inflate(layoutInflater)
        setContentView(binding.root)
        auth = FirebaseAuth.getInstance()
    }
}
```

La codificació del Registre per començar és igual que la part del login.

```
binding.BotonConfirmarRegistro.setOnClickListener { it: View ->
    val email = this.binding.EmailRegistre.text.toString().trim()
    val password = this.binding.ContrasenaRegistre.text.toString().trim()
    val passwordConf = binding.RepetirContraRegistre.text.toString().trim()
    val fecha = this.binding.DataRegistre.text.toString().trim()
    val nom = this.binding.NomRegistre.text.toString().trim()
    val cognom = this.binding.CognomRegistre.text.toString().trim()
    val codpostal = this.binding.CodiPostalRegistre.text.toString().trim()
    val cip = this.binding.CipRegistre.text.toString().trim()
    val _check_terms = this.binding.TermePrivacitatRegistre.isChecked
    val _check_conduse = this.binding.CondicionsUsRegistre.isChecked

    val isCheck = checkCampo(email, password, passwordConf, fecha, nom, cognom, codpostal, cip, _check_terms, _check_conduse)
}
```

Fem un bind amb el botó de confirmar el registre i fem variables amb els binds de tots els camps del registre per després poder-ho guardar a la base de dades, i fem una

variable booleana on fa una crida a la funció checkCampo que valida si tots els camps estan omplerts (retornarà true), si no estan omplerts retornarà false i un missatge a la part inferior a la pantalla de: “no has omplert tots els camps”

```
if(isCheck.equals(true) /*email.isEmpty() && password.isEmpty()*/) {
    Log.d(TAG, msg: "Content")
    Log.w(TAG, msg: "content2")
    createAccount(email,password)
    /*if (password == passwordConf) {
        Log.d(TAG, "content")
    }else {
        Toast.makeText(baseContext, "Contrasenya incorrecta", Toast.LENGTH_SHORT).show()
    }*/
}else {
    Toast.makeText(baseContext, text: "No has omplert tots els camps!", Toast.LENGTH_SHORT).show()
}

}
```

```
public override fun onStart() {
    super.onStart()
    // Check if user is signed in (non-null) and update UI accordingly.
    val currentUser = auth.currentUser
    if(currentUser != null){
        reload();
    }
}

private fun createAccount(email: String, password: String) {
    print(email)
    print(password)

    auth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                Log.d(TAG, msg: "createUserWithEmailAndPassword:success")
                val user = auth.currentUser
                updateUI(user)
            } else {
                Log.w(TAG, msg: "createUserWithEmailAndPassword:failure", task.exception)
                Toast.makeText(baseContext, text: "Authentication failed.", Toast.LENGTH_SHORT).show()
                updateUI( user: null)
            }
        }
    confirmaUsuario()
}
```

(La funció onStart no la estem utilitzant)

Seguim amb la funció createAccount on es crea el compte amb l'email i el password que hem escrit a l'imput del registre i si ho fa retornarà un success, sino retorna un "Authentication failed", crida a confirmaUsuario que fa el mateix que en la pantalla de Login.

```
private fun checkCampo(  
    email: String,  
    password: String,  
    passwordConf: String,  
    fecha: String,  
    nom: String,  
    cognom: String,  
    codPostal: String,  
    cip: String,  
    _check_terms: Boolean,  
    _check_condus: Boolean  
): Boolean {  
    var isChecked = false  
    if(fecha.contains( other: "/")){  
        //Log.w(TAG, "ERROR: El campo fecha no contiene /")  
        isChecked = true  
    } else {  
        //Log.w(TAG, "CORRECTO: El campo fecha contiene /")  
        return false  
    }  
  
    if(email.contains( other: "@gmail.com" ) || email.contains( other: "@gmail.es" ) || email.contains( other: "@hotmail.com" )){  
        isChecked = true  
    } else {  
        isChecked = false  
        return false  
    }  
    if(password.length > 5 && passwordConf.length > 5 && (password == passwordConf)) isChecked = true  
    else return false  
}
```

La funció de checkCampo agafa tots els valors dels inputs del registre i confirma si el valor de data té les barres per que siguin correctes, que el email tingui el domini, el password sigui igual a la confirmació de password y sigui més llarga de 5 caràcters.

```

        if(nom.length > 1 && cognom.length > 1) isCheck = true
        else return false

        if(codpostal.length > 4 && cip.length == 9) isCheck = true
        else return false

        if(_check_terms && _check_condus) isCheck = true
        else return false

        return isCheck
    }

private fun confirmaUsuario() {
    var usuario = auth.currentUser
    if(usuario != null){
        Log.w(TAG, msg: "USUARIO INFO: ${usuario.email}")
    }
}

private fun reload() {

}

private fun updateUI(user: FirebaseUser?) {
    val intent = Intent(packageContext: this, InicioUsuario::class.java)
    startActivity(intent)
    finish()
}

```

Seguim a la mateixa funció, amb les condicions. Hi ha una perquè el nom i el cognom siguin més llargs d'un caràcter, si el codi postal és més llarg de 4 caràcters i el CIP de 9, per acabar la funció si els 2 checkbox del final estan o no marcats, totes aquestes condicions si es compleixen retornaran true, sino retornaran false i la crida del principi no es farà per tant sortirà el missatge de que “no s’han omplert tots els camps”.

(La funció de reload no la fem servir)

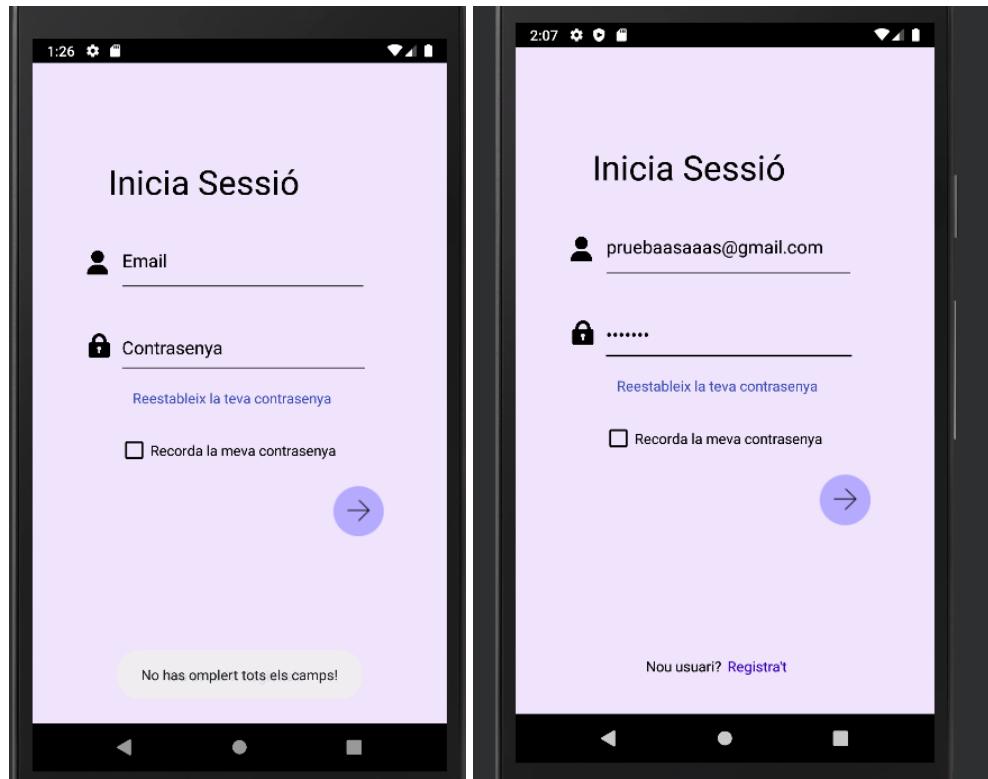
I per finalitzar el codi del registre tenim el updateUI que canvia de pantalla al



IniciUsuario.



- Captura Snackbar Missatge error (camps buits o contrasenya incorrecte)

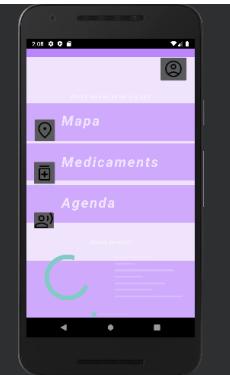


Aquí tenim les comprovacions, on si no omplies els camps surt el missatge a la part inferior de la pantalla.

Una vegada ja omplerts li donem al botó per continuar i si l'usuari és correcte pasará al menú.

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
pruebaaaaas@gmail.com	✉	28 oct 2022	28 oct 2022	1tuHjoL7tVPg9j87S7bbta5wwfD3

Filas por página: 50 ▾ 1 - 1 of 1 < >



- PANTALLA RECUPERAR CONTRASENYA



- SPLASH SCREEN

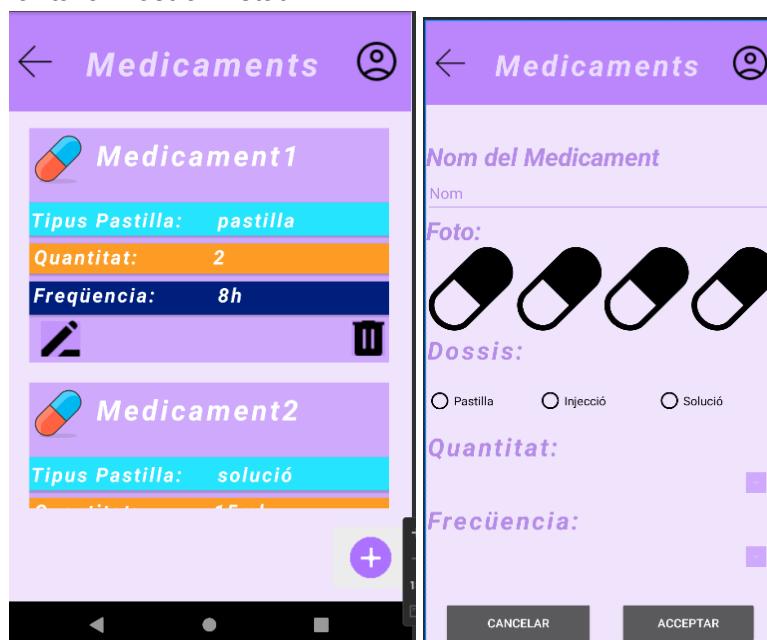


5.3.2.3. PERFIL

5.3.3.2. INSERCIÓ

5.3.3.3. CONSULTA

- Pantalla mostrar llistat...



- Quan els usuaris registrin els nous medicaments, sortiran en forma de cascada en la mateixa pantalla. Dins del codi haurem de crear un llistat (recyclerview), i inserir els ítems que l'usuari registri en la pantalla de registrar medicaments.

```

private fun setupRecyclerView() {
    //Especificuem que els fills del RV seran del mateix tamany
    binding.rvMedicaments.setHasFixedSize(true)

    //indiquem que el RV es mostrarà en format llista
    binding.rvMedicaments.layoutManager = LinearLayoutManager(
        Orientation.VERTICAL, LinearLayoutManager.VERTICAL, false
    )

    //generem el adapter
    miAdapter = MedicamentsAdapter(getMedicaments(), context = this)

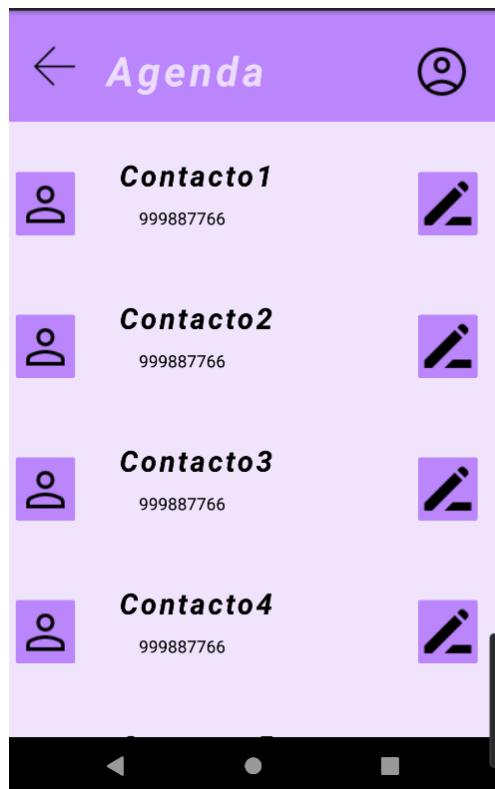
    //assignem el adapter al RV
    binding.rvMedicaments.adapter = miAdapter
}

private fun getMedicaments(): MutableList<Medicaments>{

    val medicaments: MutableList<Medicaments> = arrayListOf()
    medicaments.add(Medicaments( nombre: "Medicament1", image: "https://..."))
    medicaments.add(Medicaments( nombre: "Medicament2", image: "https://..."))
    medicaments.add(Medicaments( nombre: "Medicament3", image: "https://..."))
    medicaments.add(Medicaments( nombre: "Medicament4", image: "https://..."))
    medicaments.add(Medicaments( nombre: "Medicament5", image: "https://..."))

    return medicaments
}

```



- Aquesta pantalla funciona igual que l'anterior, però amb l'agregació de contactes, codi del llistat:

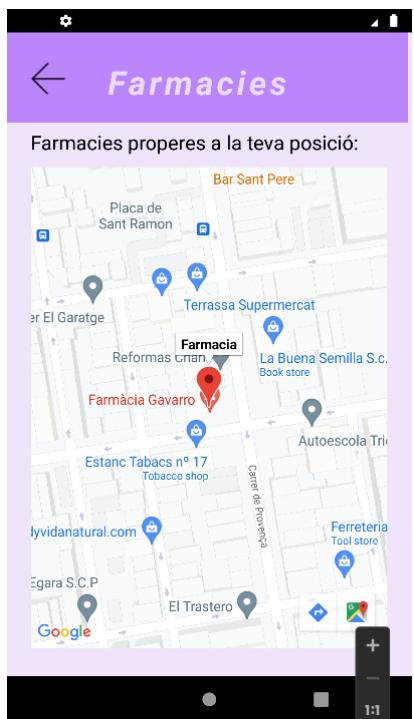
```
private fun getContactos():MutableList<Contacto>{
    val contacts: MutableList<Contacto> = arrayListOf()
    contacts.add(Contacto( name: "Contacto1", num: "999887766"))
    contacts.add(Contacto( name: "Contacto2", num: "999887766"))
    contacts.add(Contacto( name: "Contacto3", num: "999887766"))
    contacts.add(Contacto( name: "Contacto4", num: "999887766"))
    contacts.add(Contacto( name: "Contacto5", num: "999887766"))
    contacts.add(Contacto( name: "Contacto6", num: "999887766"))
    contacts.add(Contacto( name: "Contacto7", num: "999887766"))

    return contacts
}
```

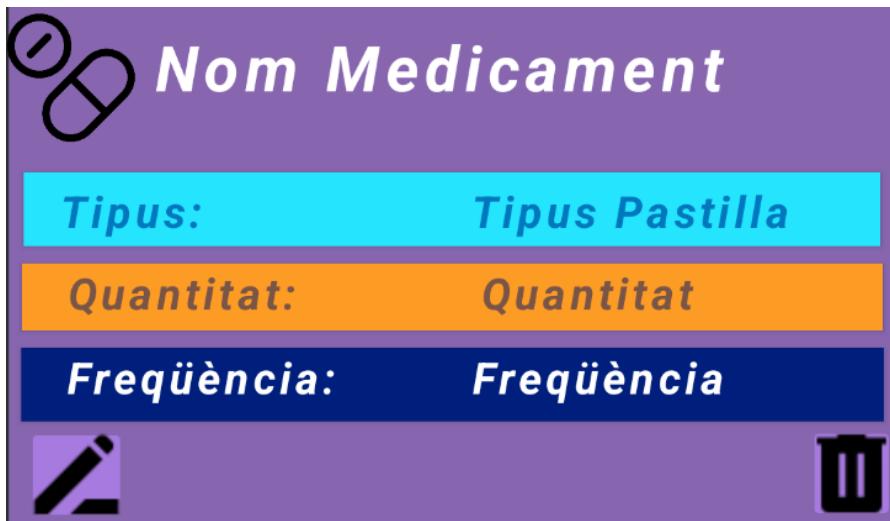
● Pantalla configuració



● Pantalla Mapa

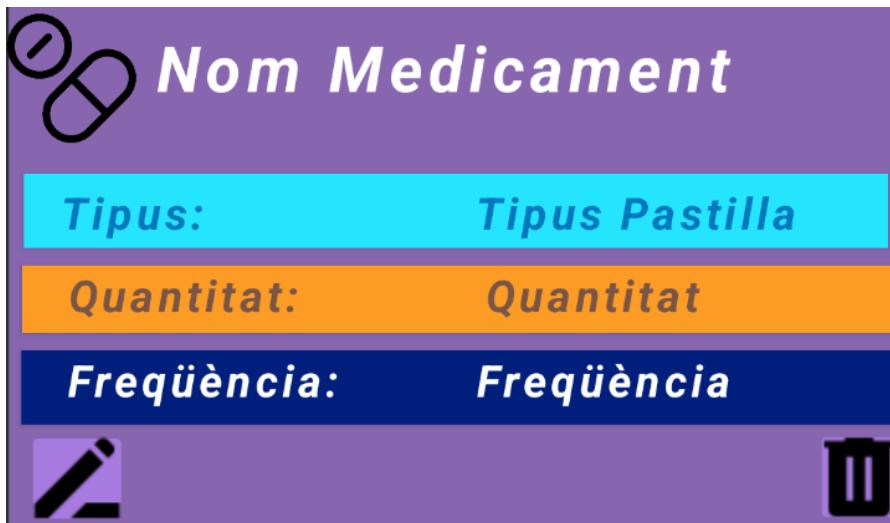


5.3.3.4. MODIFICACIÓ



Quan pulses la imatge de la paperera edita el medicament de la pantalla i de la base de dades

5.3.3.5. ELIMINACIÓ



Quan pulses la imatge de la paperera t'elimina el medicament de la pantalla i de la base de dades

5.3.4. FIREBASE STORAGE

5.3.4.1. CÀRREGA D'IMATGES

No tenim càrrega d'imatges amb el firebase

5.3.4.2. CÀRREGA DE VÍDEOS

No tenim càrrega de videos amb el firebase

5.3.5. BÚSQUEDA. FILTRES.

No tenim cap búsqueta ni filters a l'aplicació.

FINAL DE CODIFICACIÓ : 12A

. Creació de classes amb el constructor que forma part de la capçalera i/o altres constructors secundaris:

Quan es crea una nova instància a una classe, es fa un call al constructor para establir els valors inicials dels atributs de la classe.

No hem implementat la creació de classes amb un constructor.

. Modificadors de visibilitat correctament aplicats. (private, public, protected):

Serveixen per controlar qui pot accedir a la informació d'una classe:

- private: Només es podrà accedir a ella a la mateixa classe.
- public: Pot ser accedit des de qualsevol lloc
- protected: Només es podrà accedir a ella a la mateixa classe i subclasses.

Els hem aplicat, per a la creació de variables per tot el codi de l'aplicació; exemple:

```
private var _binding: FragmentCreacioContactesBinding? = null
private val binding get() = _binding!!
private val miAdapter: ContactosAdapter = ContactosAdapter()
```

. Atributs de classe. Encapsulació (getters and setters):

Serveixen per tractar amb objectes d'una classe més fàcilment, per afegir o recollir els valors dels objectes.

No hem utilitzat els getters i setters a la nostra aplicació.

. Funcions. Classe Utils/Common amb funcions comuns a vàries classes:

Serveixen per donar un tipus d'instruccions a un objecte i pot rebre una variable i retornar-la o no. Hem utilitzat les funcions a gairebé tots els fragments i activitats de la nostra aplicació; exemple:

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
```

. Herència:

Serveix perquè la classe amb la qual estàs tractant tingui les mateixes propietats d'aquella que estàs heretant, per exemple si una classe té d'herència un Fragment, totes les propietats i mètodes seran únics d'un Fragment.

Nosaltres hem aplicat l'herència en tots els arxius de l'aplicació, tant per crear Fragments, com Activities.

```
class CreacioContactes : Fragment() {
```

. Interfaces. Abstract classes. Overridings.

- Interfaces: en Kotlin serveixen per a definir un conjunt de membres (mètodes i propietats) que una classe ha d'implementar.

- Abstract classes: són classes que no es poden instanciar directament sinó que han de ser heretades per altres classes.
- Overridings: són la forma de reemplaçar el comportament d'un membre heretat en una subclasse. És a dir, una subclasse pot sobreescrivir un mètode o una propietat heretada d'una superclasse per a proporcionar la seva pròpia implementació.

No hem fet ús de classes abstractes ni interfaces, però sí que hem implementat els overridings a les diferents funcions del nostre codi:

```
override fun onCreateView(
```

. Data classes.

Són un tipus especial de classe que serveixen per emmagatzemar dades i tenen uns mètodes predefinits com el “equals”, “hashCode”, etc...

Hem utilitzat els data classes per rebre un text i passar-ho a String:

```
var email = this.binding.emailInicio.text.toString().trim()
```

. Nested and inner classes.

Són un tipus de classe que poden estar dins d'una altra classe, serveixen per organitzar el codi i fer recyclerviews; les classes poden ser estàtiques o no estàtiques:

- Nested classes: Les nested classes poden ser estàtiques o no, pot accedir als membres estàtics de la classe contenidora, però no als membres no estàtics.
- Inner classes: Les inner classes sempre són no estàtiques i tenen accés complet als membres de la classe contenidora, tan estàtics com no estàtics.

Un exemple a la nostra aplicació poden ser els recyclerviews que són simplement una llista de classes dins d'una classe contenidora.



. Enum classes

És un tipus especial de classe que permet definir un conjunt finit de constants. Aquestes constants són valors únics i no poden ser modificats durant l'execució del programa. Els Enum classes a Kotlin tenen algunes característiques addicionals en comparació a altres llenguatges, com la possibilitat de definir propietats i mètodes per a cada constant, i la capacitat d'implementar interfaces.

No hem fet servir els enum classes al nostre projecte.

. Companion objects

Són un objecte relacionat a una classe, és semblant a una classe estàtica perquè permet definir mètodes i propietats que no estan associades a una instància específica de la classe, i també permeten emmagatzemar dades compartides entre totes les instàncies de la classe.

Nosaltres hem implementat els companion objects per a una MutableList:

```
companion object {
    public lateinit var idsmed: MutableList<String>
}
```

Altres:

. Regular expressions

Són una seqüència de caràcters que formen un patró de cerca, s'utilitzen per buscar o manipular cadenes de text.

No hem implementat cap regex al nostre codi.

. Null safety

És un mecanisme de seguretat per prevenir errors de "null" en temps d'execució. S'utilitza el "?" per indicar que una variable es nullable i el "!!" per indicar que una variable no es nullable.

Hem implementat el Null Safety de les dues maneres al nostre codi.

```
medicaments.clear()
db.collection( collectionPath: "medicamentsUsuaris" ).document(FirebaseAuth.getInstance().currentUser!.uid)

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
```

. Exceptions

Són un mecanisme per tractar errors o situacions inesperades d'un programa, és un esdeveniment que interromp el flux normal d'execució del programa i s'utilitza per notificar a l'aplicació de l'error o la situació anormal.

No hem implementat cap excepció al nostre programa.

. Annotations

Són un mecanisme per afegir metadada a elements del codi, com classes, funcions, variables,etc. Són com una espècie d'etiqueta o marca per indicar una cosa especial a l'element, són similars als comentaris però les anotacions són processades pel compilador i poden ser utilitzades per a generar codi addicional.

```
@Suppress(...names: "DEPRECATION")
@Suppress(...names: "UNREACHABLE_CODE")
```

. Implementació d'algoritmes d'ordenació (selection sort, bubble sort, merge sort...)

Són un conjunt de tècniques per ordenar els elements d'una col·lecció, com una llista. L'objectiu dels algoritmes d'ordenació és organitzar els elements de tal manera que compleixin una condició, com que sigui ordre ascendent o descendent.

No l'hem implementat al nostre codi.

. Implementació d'algoritmes de cerca (sequential search, binary search,...)

Són un conjunt de tècniques per trobar un element específic dins d'una col·lecció. L'objectiu dels algoritmes d'ordenació és trobar un element específic a una col·lecció de tal manera que compleixin una condició.

No l'hem implementat al nostre codi.

. Callbacks

Són un mecanisme per passar una funció com argument a una altra funció, de manera que la funció rebuda pugui executar-se en algun punt dins de la funció que la rep. Els callbacks són una forma d'arribar a una programació asíncrona, permetent que una funció passi certa tasca a altra funció i continuar amb la seva execució sense esperar que la tasca sigui completada.

Hem implementat callbacks al nostre codi de diferents formes però el "setOnClickListener" és un d'ells:

```
binding.btnCancelarmedicement.setOnClickListener {
```

. Coroutines

Són un mecanisme per tractar la concorrència i la programació asíncrona de manera fàcil i eficient. Una coroutine és una tasca lleugera que s'executa en paral·lel amb altres tasques i pot ser suspesa i represa en qualsevol moment.

No l'hem implementat al nostre codi.

. Destructuring declarations

Són un mecanisme per descompondre un objecte en components individuals i assignar-los a variables individuals. És una manera de desempaquetar un objecte complex en parts més petites i tractar-les de manera més fàcil.

5.3.6. PAS DE DADES ENTRE PANTALLES.

El nostre passa de dades es farà d'una manera confirmant que tot estigui "quadrat", és a dir, per exemple si en l'inici de sessió, la credencial que afegeix l'usuari són incorrectes, abans que doni la resposta fa una verificació amb les dades que n'hi han emmagatzemades en la base de dades que hem creat, i si les dades coincideixen, aleshores arriba un "OK" mitjançant l'API i donarà una resposta en l'app, de manera que si coincideixen acceptarà l'usuari que ha introduït, per altra banda no acceptarà si qualsevol de les dades introduïdes no coincideixen, i no donarà un error de manera que no doni informació (usuario incorrecto o contraseña incorrecta) tampoc accessos no desitjats, per exemple, en cas de telèfon robat, perdut, trencat, etc...

Amb el tema dels medicaments, ubicacions i etc... També tenen comunicació amb la base de dades, i es necessaria, ja que per exemple, en el registre, s'ha d'emmagatzemar totes les dades de l'usuari que ha introduït en cas de que siguin correctes. Així si l'usuari fa login, ja tindrà un usuari registrat i no li caldrà fer un altre registre. En el cas dels medicaments, també s'emmagatzemen ja que és important saber que la persona que utilitza l'app, sàpiga quins medicaments té guardats i s'ha de prendre diàriament.

5.3.7. ANIMACIONS

La nostra app, per cada canvi de pantalla que hi hagi mostra una transició amb l'etiqueta "<fade>".

5.3.8 NOTIFICACIONS

L'app com a tal, només tindrà una notificació que serà la més evident que nosaltres considerem important i imprescindible, és el recordatori dels medicaments per l'usuari que està utilitzant l'app. Això s'organitza de manera que sempre que sigui l'hora del medicament, apareixerà en la part superior d'una notificació en rectangle amb el logo de NitroHealth a l'esquerra avisant de què s'ha de prendre els medicaments.

5.3.9 GEOLOCALITZACIÓ

Amb el tema de la geolocalització hem fet una pantalla on l'individu, agafa la secció automàticament el mapa agafa el seu punt on es troba actualment, i fa una recerca de farmàcies, hospitals o punts de salut més propers basats en la seva ubicació en temps real. Gràcies a això l'usuari pot estar constantment informat de tots els punts on pot anar en cas d'emergència.

5.4. STYLES I THEMES

El nostre estil de pantalles, serà més senzill on el nostre fons main serà "#f0e4fc", hem considerat que aquest color és el més suau per aquest temes de sanitat i on l'usuari podrà gaudir d'una manera més tranquil·la, i no amb colors intensos. Els temes, els hem fet nosaltres partint d'un fragment en Blank, les vores que estan en els extrems son toolbars on escrivim el títol de la pantalla, per a que l'usuari tingui clar en quina secció es troba actualment. Hem incorporat també una visualització més àmplia per l'usuari on no hi hagi molta complicació a l'hora d'utilitzar-ho, l'objectiu des d'un principi era i segueix sent aquest.

5.5. DISSENY ALTERNATIU PER VARIS TIPUS DE PANTALLA.

- Landscape
- Tablet

5.6. ACCESSIBILITAT

5.7. INTERNACIONALITZACIÓ.

En el nostre cas, hem posat tots els quadres de text en format de String, gràcies a això podem implementar la funcionalitat d'idioma, això permet que la funció detecti el text com a String i podem afegir un desplegable amb tots els idiomes disponibles i traduir l'app sencera en l'idioma de l'usuari que està utilitzant. Amb això considerem que l'aplicació és adaptativa de manera internacional, i més persones podran utilitzar l'app. Considerant que fos una app real on haguéssim d'obtenir un benefici, seria un gran punt pel projecte, ja que està més oberta a persones per tot el món, on poden recomanar l'app i descargarse'l més usuaris de mòbil.

5.8. GENERACIÓ APK I PUBLICAR A GOOGLE PLAY

6. TESTEIG

6.1. Tests Unitaris

1. Quina és la teva edat?
 - 10-15 anys
 - 15-18 anys
 - 18-30 anys
 - 30-60 anys
 - +60 anys
2. Quin és el teu sexe?
 - Masculí
 - Femení
 - Indefinit
 - Prefereixo no dir-ho
3. Quin coneixement tecnològic tens?
 - Molt Alt

- Alt
- Mitjà
- Baix
- Molt Baix/ No tinc

4. Navegar per l'aplicació se t'ha fet complicat?

- No, gens
- No gaire, algunes coses no estaven prou clares.
- Si, una mica, faltaven explicacions clares.
- Si, es molt difícil.

5. Creus que l'aplicació és apte per a totes les edats?

- Si, és molt adequada.
- Sí, però té algunes restriccions
- No, crec que es limitada per a una franja d'edat.

6. Creus que compleix amb totes les necessitats per a una persona amb una discapacitat(enfermetats, edat avançada,etc...) ?

- Si, l'aplicació és molt completa.
- Si, pero careix d'alguna cosa.
- No, faltarien molts aspectes i característiques.

7. De totes les opcions de l'aplicació quina t'ha agradat més?

- Mapa amb farmàcies/hospitals propers.
- Agenda per als contactes d'emergència.
- Introduir medicaments amb recordatoris.
- Canvi de vista a mode daltònic(4 modes).

8. Tú utilitzaries l'aplicació en la teva rutina diària?

9. Recomendaries l'app a alguna persona?

- Si, segur que interessa a algú.
- No, no crec que ningú del meu voltant la necessiti.
- Probablement si.

10. En una escala de l'1 al 10 quant satisfet estàs amb l'aplicació.

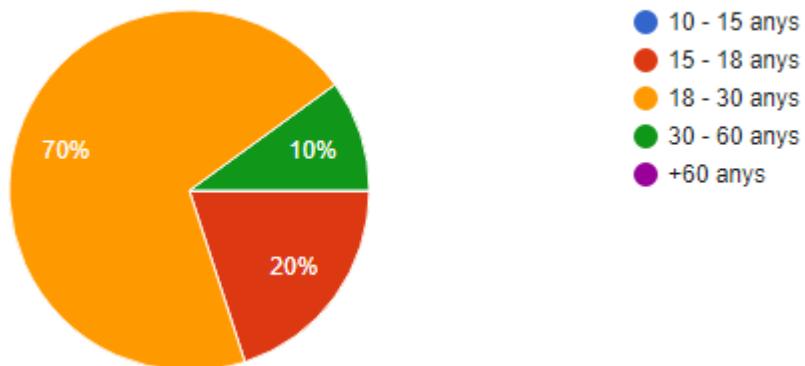
11. En una escala de l'1 al 10 quant calificaries l'aspecte de l'aplicació

12. Què característiques esperaves i has trobat a faltar a l'aplicació

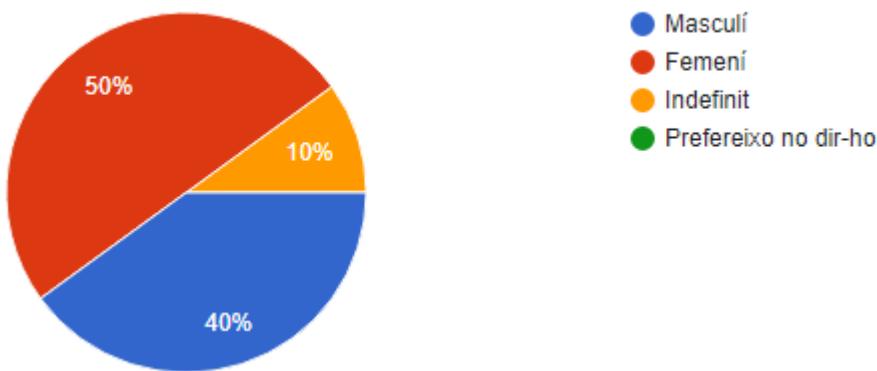
13. Observacions de l'aplicació

6.2. Tests amb usuaris finals

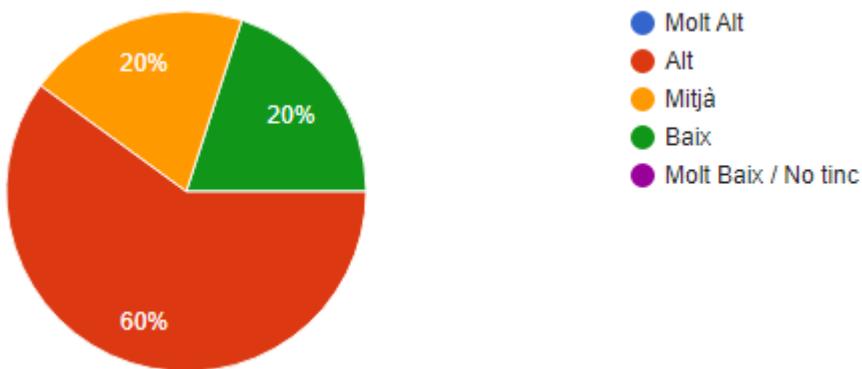
1. Quina és la teva edat?



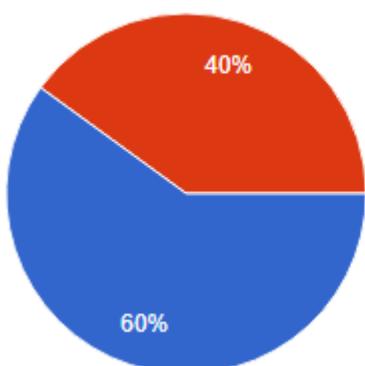
2. Quin és el teu sexe?



3. Quin coneixement tecnològic tens?

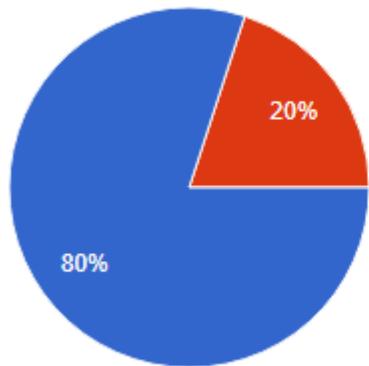


4. Navegar per l'aplicació se t'ha fet complicat?



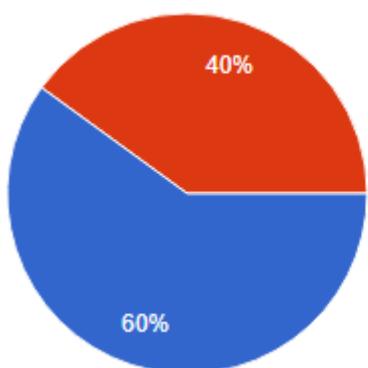
- No, gens.
- No gaire, algunes coses no estaven prou clares.
- Si, una mica, faltaven explicacions clares.
- Si, es molt difícil.

5. Creus que l'aplicació és apte per a totes les edats?



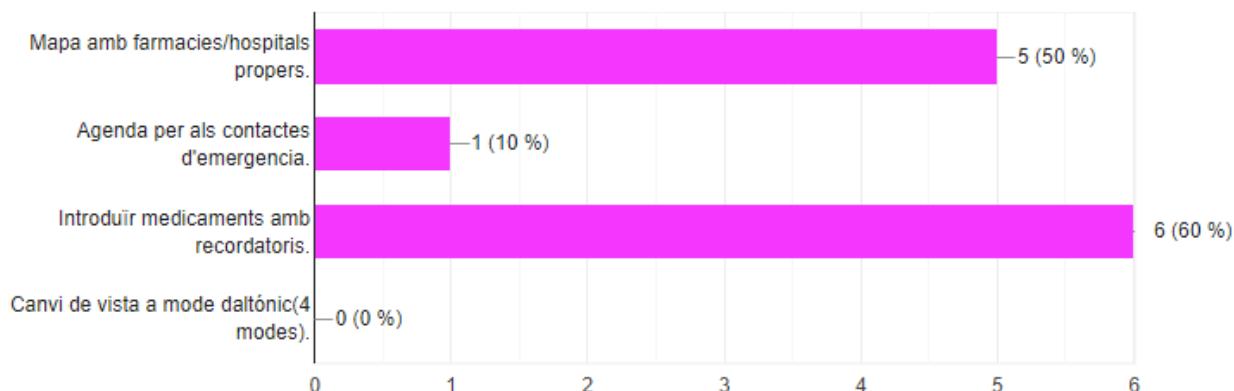
- Si, es molt adequada.
- Si, pero té algunes restriccions
- No, crec que es limitada per a una franja d'edat.

6. Creus que compleix amb totes les necessitats per a una persona amb una discapacitat(enfermetats, edat avançada,etc...) ?

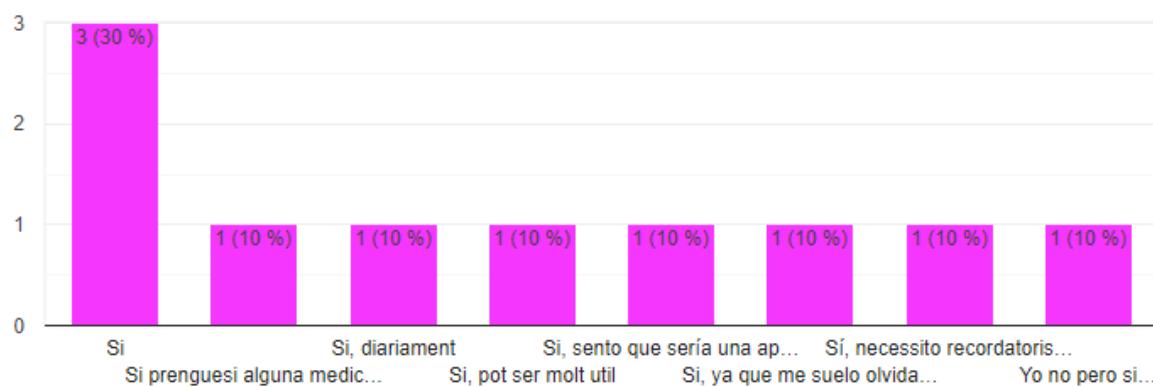


- Si, l'aplicació es molt completa.
- Si, pero careix d'alguna cosa.
- No, faltarien molts aspectes i característiques.

7. De totes les opcions de l'aplicació quina t'ha agradat més?



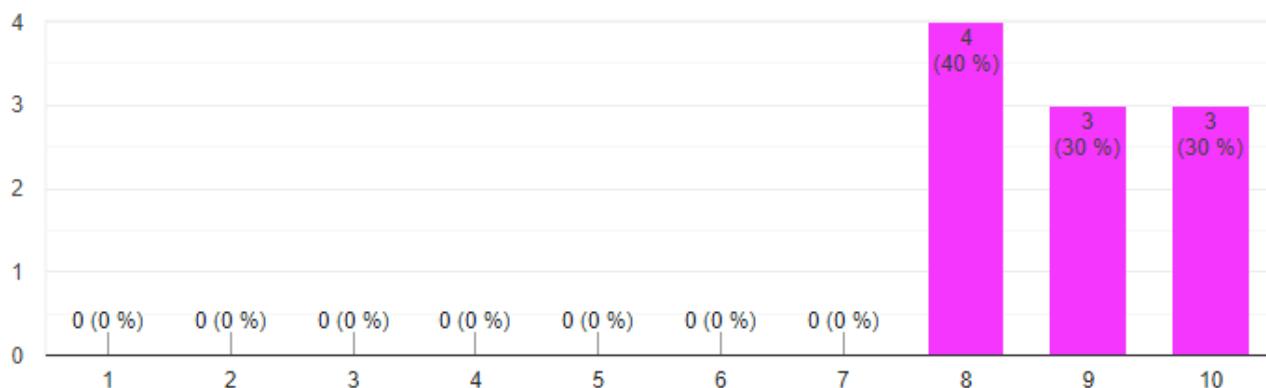
8. Tú utilitzaries l'aplicació en la teva rutina diària?



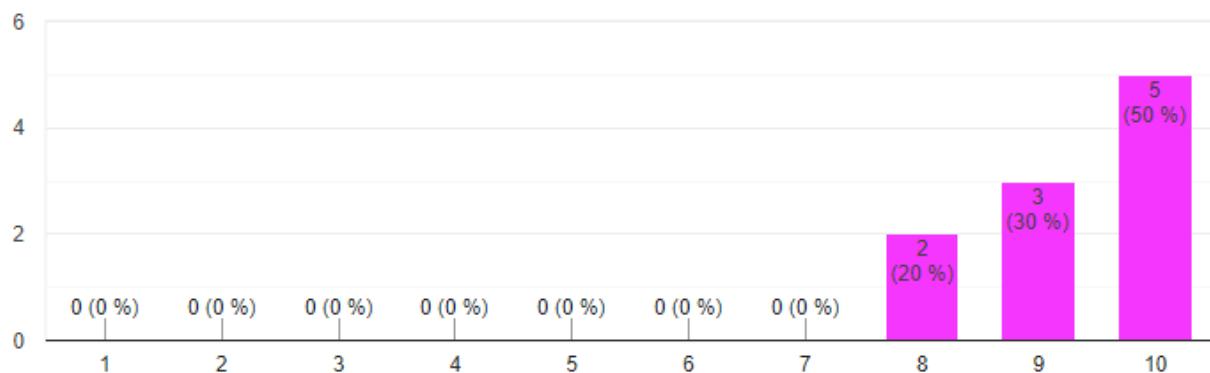
9. Recomendaries l'app a alguna persona?



10. En una escala de l'1 al 10 quant satisfet estás amb l'aplicació.



11. En una escala de l'1 al 10 quant calificaries l'aspecte de l'aplicació?



12. Qué característiques esperaves i has trobat a faltar a l'aplicació?

Nada, yo creo que es muy completa

Sento que es podría incloure alguna funció més, com activar el so dels semàfors per la gent invident vía veu, o una opció de traductor a llenguatge de seüles (o viceversa) per que la gent sordomuda que no entén les lletres es pugui comunicar amb la gent normal

Res que em vingui a la ment

Cap

No e trobat a faltar res

Nada

13. Observacions de l'aplicació

Es una aplicación muy completa para ser la primera versión y si se llega a desarrollar más puede llegar a ser muy útil

Per ara trobo tot correcte i molt genial

Podria ser una molt bona aplicació amb més temps

Es molt bona

Es una aplicació molt bona i completa

Es visualmente agradable y fácilmente puedes encontrar las herramientas que te ofrece

7. APLICACIÓ FINAL

7.1. Manual d'usuari

7.2. Vídeo-demo

<https://drive.google.com/file/d/1J41LuMOrLFDdly2u-Q0v9xdy--d6fu7l/view?usp=sharing>

8. LÍNIES FUTURES

No tenim pensat desenvolupar més l'aplicació, però creiem que si fos un projecte professional podríem fer una aplicació que fos molt útil per a la gent que realment ho necessita. La idea pot ajudar molt a què la gent amb algun tipus de malaltia o discapacitat pugui utilitzar el telèfon amb més comoditat.

9. GESTIÓ DEL PROJECTE

9.1. COMUNICACIÓ (F1)

9.2. PERFIL PROFESSIONAL (F2)

- Laura : <https://www.linkedin.com/in/laura-roldan-garcia-68429522b/>

9.3. TREBALL EN EQUIP (F3)

Torre d'espaguetis

9.4. CONFLICTES I RESOLUCIÓ (F5)

- Rols de l'equip:
 - David: Desenvolupador
 - Aiman: Desenvolupador
 - Laura: Desenvolupadora
- Conflictes:
 - A l'equip no tenim per al moment cap conflicte, intentem treballar tots al mateix ritme i decidim entre tots les tasques que fa cadascú, ja que no volem tindre problemes de què un mana més que un altre.

10. CONCLUSIÓ

Hem après molt en el transcurs del projecte, ja que hem fet una aplicació des de 0 i hem après a treballar amb Kotlin, que no l'havíem utilitzat mai.

Ha sigut un procés difícil i molt dens, però creiem que pel nostre aprenentatge ha sigut molt gratificant.

Un dels punts més importants que podem treure que ens ha servit molt, ha sigut l'autonomia que hem obtingut mitjançant anaven avançant el projecte.