

☕ la Cantina de Prodí� 🍔

Alumnes:

Jaume Cadafalch Miró
Adrian Ramirez Masagú

Tutores:

Xesca Tomàs
Esther Sanchez
Mercè Sitges

CFGS 2 DAM matí
2021-2022

INDEX

1. Introducció	6
1.1 Objectius de l'aplicació	6
1.2 Necessitats a resoldre	7
1.3 Target	8
1.4 Recollida de requisits	9
1.4.1 Actes reunions amb clients	11
1.4.1.1 Acta 18-10-2021	11
1.5 Anàlisi de requisits	14
1.5.1 Perfil d'usuari	14
1.5.2 Funcionalitats dels perfils	14
1.5.3 Requisits no funcionals	16
2. Pla de producte	17
2.1 Estudi de mercat	17
2.1.1 Producte / Servei	17
2.1.2 Client	17
2.1.3 Competència	17
2.2 Pressupost empresarial	17
2.2.1 Càcul de cost	17
2.2.2 Previsió de vendes	19
2.2.3 Càcul del punt mort	19
2.3 Dafo	20
2.4 Viabilitat	21
2.5 Promoció	21
2.6 Distribució	21
3. Disseny de l'aplicació	22
3.1 Mockups	22
3.1.1 Diagrama de navegació	22
3.1.2 Pantalles	24
3.2 Disseny de la base de dades NoSQL	48
4. Planificació i seguiment de les tasques de desenvolupament	50
4.1 Eines	50
4.1.1 Gestor de projectes Trello	50
4.1.2 Diagrama de gantt - TeamGantt	51
4.1.3 Issues del gitlab	52
4.2 Planificació Inicial	53
4.3 Seguiment Setmanal	54
4.3.1 Setmana 1	54
4.3.2 Setmana 2	56
4.3.3 Setmana 3	58

4.3.4 Setmana 4	60
4.3.5 Setmana 5	62
4.3.6 Setmana 6	64
4.3.7 Setmana 7	66
4.3.8 Setmana 8 i setmana 9	68
5. Desenvolupament de l'app	70
5.1 Control de versions	70
5.1.2 Git	70
5.1.2.1 Funcionalitats	71
5.1.2.1.1 Pull	71
5.1.2.1.2 Branches	72
5.1.2.1.3 Add	74
5.1.2.1.4 Commit	74
5.1.2.1.5 Push	75
5.1.2.1.6 Merge	76
5.1.2.2 Branques	77
5.1.2.2.1 Opcions comuns	77
5.1.2.2.2 Creació Branques	77
5.1.2.2.3 Creació Branques Remotes	78
5.1.2.2.4 Eliminació Branques	78
5.1.2.2.5 Treballant amb branques	80
5.1.2.3 Historial de versions	85
5.1.2.4 Issues	87
5.1.2.4.1 Seguiment dels issues	87
5.1.2.4.2 Tancament del issues	87
5.1.2.4.3 Etiquetes	87
5.1.2.4.4 Assignees	88
5.1.2.4.5 Milestone	88
5.1.2.4.6 Due date	89
5.1.2.5 Merge requests	89
5.1.2.5.1 Com fer un merge request	89
5.2 Estructura del projecte	90
5.2.1 Packages	90
5.2.1.1 ViewModel & Live Data	91
5.2.1.2 Corroutines	92
5.2.1.3 Data Class	92
5.2.1.4 Adapter	93
5.2.2 Layouts	94
5.2.3 Drawables	109
5.2.4 Mipmap	110
5.2.5 Menu	111
5.2.6 Values	111
5.2.6.1 Strings	112

5.2.6.3 Colors	112
5.2.6.3 Styles	113
5.2.6.4 Dimensions	114
5.2.7 Gradle	115
5.2.7.1 Gradle Project	115
5.2.7.1 Gradle Module	116
5.3 Codificació	118
5.3.1 Navegabilitat	118
5.3.3 Base de dades Firebase	122
5.3.3.0 Creació d'un projecte Firebase	123
5.3.3.1 Vinculació del projecte amb la app	126
5.3.3.2 Autenticació amb Firebase Authentication	131
5.3.3.3 Creació base de dades Cloud Firestore	133
5.3.3.4 Registre i accés a l'app	137
5.3.3.4.1 Pantalla Login	137
5.3.3.4.2 Pantalla Registre	138
5.3.3.4.3 Pantalla recuperar contrasenya I	139
5.3.3.4.4 Pantalla recuperar contrasenya II	140
5.3.3.4.5 Tancar Sessió	141
5.3.3.5 Perfil	141
5.3.3.5.1 Pantalla mostrar perfil	141
5.3.3.6 Inserció	142
5.3.3.6.1 Pantalla Formulari Afegir nou usuari	142
5.3.3.7 Consulta	142
5.3.3.8 Modificació	143
5.3.3.9 Eliminació	143
5.3.4 Firebase Storage	144
5.3.4.1 Càrrega d'imatges	144
5.3.4.1.1 Agafar imatges del Storage	144
5.3.4.1.2 Afegir imatges al storage	144
5.3.5 Dashboards - Llistats	145
5.3.7 Pas de dades entre pantalles	145
5.3.8 Animacions	146
5.4 Notificacions	148
5.5 Accessibilitat	150
5.6 Internacionalització	152
5.7 Disseny alternatius per varis tipus de pantalla	154
5.7.1 Versió vertical per a telèfon	155
5.7.2 Versió horitzontal per a telèfon	156
5.7.2 Versió per a tablet	156
5.8 Generació APK i publicar a google play	157
5.8.1 Generació APK	157
5.8.2 Publicar a Google play	160

5.9 Hipotètica monetització de l'app	173
6 Testeig	173
6.1 Testeig proves unitàries	174
6.2 Test amb usuaris	176
7 Aplicació final	177
8 Línies Futures	177
Possibles noves funcions	177
Corregir bugs	177
9 Conclusió	177
Webgrafia	178

1. Introducció

1.1 Objectius de l'aplicació

En la nostra visita a Prodis ens van comentar que voldrien una aplicació per gestionar les comandes de la cantina, com la idea ens va agradar hem decidit crear-la. L'objectiu és fer una aplicació perquè els usuaris de Prodis puguin fer comandes a la cantina des del seu lloc de treball, d'aquesta manera seria més còmode pels usuaris demanar a la cantina a la vegada que seria més fàcil per al personal de la cantina administrar les comandes dels clients.

1.2 Necessitats a resoldre

A Prodis ens van dir que volien una aplicació per poder fer comandes des de les oficines de la planta baixa a la cantina que hi és a la planta superior. El que volem aconseguir amb aquesta aplicació és agilitzar el procés de demanar a la cantina per ajudar als nois de Prodis a fer un treball més senzill.

També ens van dir que actualment utilitzen dos papers per tomar nota de la comanda, un per a les begudes i un altre per als entrepans. Amb la nostra aplicació eliminarem aquests papers i a més a més faran servir la seva tauleta per prendre nota de la comanda.

1.3 Target

L'aplicació “La Cantina de Prodis” està destinada a la *Fundació Privada Prodiscalacitats*, més coneguda com a **Prodis**.

La fundació Prodis, situada a la plaça del Tint, Terrassa, és una entitat social oberta a la societat, sense ànim de lucre que neix d'una iniciativa voluntària d'unes famílies que en els anys seixanta van lluitar perquè els seus fills poguessin formar part de ple dret, d'una forma activa i normalitzada de la societat.

Destinada a la promoció integral de persones adultes amb discapacitat intel·lectual, malaltia mental o paràlisi cerebral, administren recursos que provenen de diferents entitats i organismes, el que els obliga a gestionar la Fundació de forma rigorosa, transparent i sostenible.

La **seva missió** és fer possible la promoció integral de les persones adultes amb discapacitat, principalment intel·lectual, paràlisi cerebral o malaltia mental, més enllà del fet purament assistencial. A més els acompanyen de manera personalitzada en el seu recorregut vital, centrant-se a donar resposta i millorar totes les dimensions de la seva qualitat de vida.

Tot això gràcies a un equip de professionals i voluntaris, implicats, en formació continua, que treballen dia a dia convençuts que la seva missió és possible.

1.4 Recollida de requisits

El dilluns 27 de setembre de 2021 ens varem reunir amb els nostres clients, els quals ens van comentar com els hi agradaria l'aplicació, suggeriments per facilitar les seves tasques, la seva eficàcia i adaptar el sistema que ells fan servir a una aplicació.

Ens van comentar que els agradaria tenir diferents tipus d'usuaris:

- **Usuari Client**
- **Usuari Cambrer**
- **Usuari Cantina**

Els hi vam proposar tenir un usuari administrador per gestionar els usuaris i varem decidir tirar endavant la proposta amb una posterior revisió per acabar de veure si és viable o no.

Dins l'usuari cantina ens trobarem amb dos apartats importants: cuina i caixer, on cadascun tindrà les funcionalitats pertinents

També ens van comentar que volien una aplicació molt intuïtiva, molt semblant a les pantalles tàctils que tenen al McDonald per fer comandes, ja que l'aplicació la farien servir des de dispositius mòbils i tauletes.

Entre les funcions de l'aplicació volien que:

- Hi hagués l'opció per indicar que la comanda era per emportar
- Els clients puguin donar feedback
- En generar un tiquet, aparegui el nom del client, preu total i preu desglossat
- El cambrer primer tria el que volien els clients i posteriorment indicar de quin client era Aquella comanda, mitjançant un llistat amb el nom dels clients (nom + cognom)
- En la selecció de quin bocata, indicar si volien el pa sucat amb tomàquet o sense amb un check box
- Sistema per quadrar la caixa (vendes i els diners que hi ha a la caixa)

També ens van comentar els productes que venen:

Entrepans (grans i petits): Pernil dolç, Pernil salat/lbèric, Formatge, Fuet, Tonyina i Tonyina amb tomàquet

Begudes Fredes: Aigua, Llet soja, Llet sense lactosa, Llet normal, Suc de taronja natural

Begudes Calentes: Cafè, Té verd, Té Negre, Té vermel·l, Té de fruites, Mançanilla, Mançanilla amb mel, Romaní, Xocolata.

Complements: sucre blanc, sucre morè, sacarina

Ens comenten que potser en un temps, tindrien begudes en llauna estil coca-cola, fanta, etc.

Preus: Bocata gran i suc de taronja 2 €, la resta 1 €

El pagament sempre es fa en efectiu.

Finalment, necessiten tenir l'aplicació en català i castellà

1.4.1 Actes reunions amb clients

1.4.1.1 Acta 18-10-2021

Acta de reunió

Data	18/10/2021	Hora	10:45	Espai	2.01
------	------------	------	-------	-------	------

Persones assistents

- Crispi Carbonell (Usuari)
- Sergi Cañas (Usuari)
- Felipe Deza (Usuari)
- Mercè Miró (Trebballadora de Prodis)
- Jaume Cadafalch (Alumne 2 DAM matí)
- Adrián Ramírez (Alumne 2 DAM matí)
- Francesca Tomàs (Docent 2 DAM matí)
- Esther Sánchez (Docent 2 DAM matí)
- Jordi Cárdenas (Coordinador)

Seguiment de l'ordre del dia

- Presentacions dels assistents
- Presentació del primer prototip amb Marvel app
 - Mostra del funcionament del prototip
 - Aclariment de dubtes respecte al funcionament del prototip
 - Posada en comú de millores, noves funcionalitats i disseny
 - Recollida d'idees sorgides en la posada en comú
 - Nom del client en la pantalla de comandes
 - Opció per modificar la mida de la lletra
 - Pantalla principal amb els noms dels perfils d'usuari en majúscules
 - Bocata d'una única mida
 - Afegir opció sense sucre a la pantalla atributs beguda
 - En la pantalla d'introduir el nom del client (perfil cambrer), el cambrer introduceix el nom del client manualment i, a mesura que va escrivint es mostren els noms que coincideixen.
 - Pantalla comandes el nom del client més gran i el número de comanda més petit
 - Administrador gestionar cada quant es genera un resum de les comandes (diari, setmanal o mensual)
 - Administrador pugui afegir / gestionar productes
 - Afegir secció fruita i promocions a la pantalla selecció tipus de producte

- En la pantalla d'introduir el nom del client (perfil cambrer) si un client no és de Prodis, el cambrer escriurà "Extern" en l'apartat de nom del client i se li mostrarà un nou espai on podrà introduir el nom d'aquell client
- Nom perfils d'usuari en masculí i femení
- Contrasenya perfil Cambrer, Cuiner, Caixer serà: el nom del perfil (cambrer, caixer, cuiner) o seqüència numèrica senzilla (1234)
- Sol·licitud d'imatges dels productes per afegir-les a l'aplicació.
- Es demana un correu de contacte a la treballadora de prodis Mercè Miró (soi@prodis.cat).
- Dubtes generals sobre el funcionament de l'aplicació.
 - Pregunten si en l'apartat de comandes es poden separar les begudes dels entrepans. S'arriba a la conclusió que aquesta idea no es viable per com està dissenyada l'app.
- Registre d'acords
 - Jaume i Adrian acorden enviar l'enllaç del prototip amb les noves modificacions a la Mercè, a més d'indicar els seus correus electrònics de l'institut per posterior intercanvi d'informació. (nous suggeriments o canvis per ambdues parts)
 - Mercè acorda tremetre fotos de cada tipus de producte que tenen poder incloure-les dins de l'aplicació final.

Seguiment dels acords anteriors			
Acord	Responsable	Data	Estat

Acords establerts en aquesta reunió			
Acord	Responsable	Data	Comentaris
Realitzar l'acta	Jaume Cadafalch Adrián Ramírez	20/10/2021	Cap
Enviar enllaç del prototip amb les noves modificacions	Jaume Cadafalch Adrián Ramírez	Un cop les modificacions estiguin acabades	Cap
Tremetre fotos de cada tipus de producte.	Mercè Miró	Quan sigui possible	Cap
Enviar un email a l'adreça facilitada per la Mercè per un possible intercanvi d'informació / idees	Jaume Cadafalch Adrián Ramírez	Un cop les modificacions estiguin acabades	Cap

Fotos del dia:



1.5 Anàlisi de requisits

1.5.1 Perfil d'usuari

- **Usuari client:**

L'usuari client serà el que farà la comanda de manera remota des de fora de la cantina.

- **Usuari cambrer:**

L'usuari cambrer serà l'usuari que farà la comanda des de la cantina.

- **Usuari cantina:**

L'usuari cantina serà el que treballa a la cantina i el que rebrà les comandes fetes per als usuaris cambrers i clients.

- **Usuari administrador:**

Serà l'encarregat d'administrar els usuaris que tindran accés a l'aplicació.

1.5.2 Funcionalitats dels perfils

Usuari client

- Es registrará mitjançant el seu correu electrònic amb domini "@prodis.cat"
- Seleccionarà el producte que vol per a la seva comanda.
- Indicarà si vol algun producte específic si cal (tipus de llet, pa sense tomàquet, etc.)
- Indicarà si vol la comanda per emportar.
- Enviarà la comanda a l'usuari cantina
- Opcionalment podrà donar "feedback"

Usuari cambrer

- Indicaran els productes que aniran en la comanda.
- Indicarà si vol algun producte específic si cal (tipus de llet, pa sense tomàquet, etc.)
- Introduirà les dades personals del client per la comanda.
- Indicarà si vol la comanda per emportar.
- Enviaran la comanda a l'usuari cantina per poder imprimir el tiquet.

Usuari Administrador

- Donar de baixa els usuaris client en cas que sigui necessari
- Afegir o eliminar usuaris del llistat de clients.

Usuari cantina

- Rebrà la comanda de l'usuari client.
- Rebrà la comanda de l'usuari cambrer.
- Podrà imprimir el tiquet amb la comanda.
- Tindrà un sistema per quadrad la caixa
- Rebrà el "feedback" de l'usuari client.
- Podrà afegir productes en cas que sigui necessari.
- Podrà ocultar productes en cas que sigui necessari.

1.5.3 Requisits no funcionals

Aplicació visualment atractiva i intuïtiva, semblant a les pantalles per fer comandes del McDonald.

L'aplicació ha d'estar dissenyada per ser utilitzada tant amb mòbil com amb tauleta.

L'aplicació ha d'estar disponible tant en català com en castellà.

Colors agradables, en tons suaus.

2. Pla de producte

2.1 Estudi de mercat

2.1.1 Producte / Servei

La nostra aplicació va dirigida a una clientela molt específica, com són la gent de Prodis. No és una aplicació dirigida a un nivell de clientela individual sinó a organitzacions com és el cas de Prodis.

2.1.2 Client

La nostra aplicació va dirigida a una clientela molt específica, com són la gent de Prodis. No és una aplicació dirigida a un nivell de clientela individual sinó a organitzacions com és el cas de Prodis.

2.1.3 Competència

Dins del nostre sector no hi ha competència, ja que fem la nostra aplicació per a una empresa que ens ha demanat de manera específica aquesta app.

2.2 Pressupost empresarial

2.2.1 Càlcul de cost

IMMOBILITZAT MATERIAL		
Ordinadors ((1 unitat) x 2 persones)	700€ x 2	1400€
Mòbils ((2 unitats x persona) x 2 persones)	(250€ x 2) x 2	1000€
Tablets ((1 unitat) x 2 persones)	150€ x 2	300€
Total Immobilitat Material		2700€

IMMOBILITZAT IMMATERIAL		
Construcció de l'empresa		
Despeses de constitució de l'empresa	500€	500€
Total Immobilitat Immaterial		500€

Publicació del producte	
Compte de desenvolupador de Google Play	22,10€
Total publicació del producte	22,10€

Despeses elèctriques		
Llum	370x2persones x 0,35kWh	260€
Total despeses elèctriques		260€

Mà d'obra		
Mà d'obra	(370h x 6,25h€) x 2 persones	4625€
Total mà d'obra		4625€

INVERSIÓ INICIAL TOTAL	
Immobilitat Material	2700€
Immobilitat Immaterial	500€
Publicació del producte	22,10€
Despeses elèctriques	260€
Mà d'obra	4625€
Cost total	8107,10€

2.2.2 Previsió de vendes

Les nostres vendes no serien molt exitoses, ja que la clientela del producte és molt reduïda i l'aplicació està molt focalitzada a un tipus de públic molt concret i poc comú.

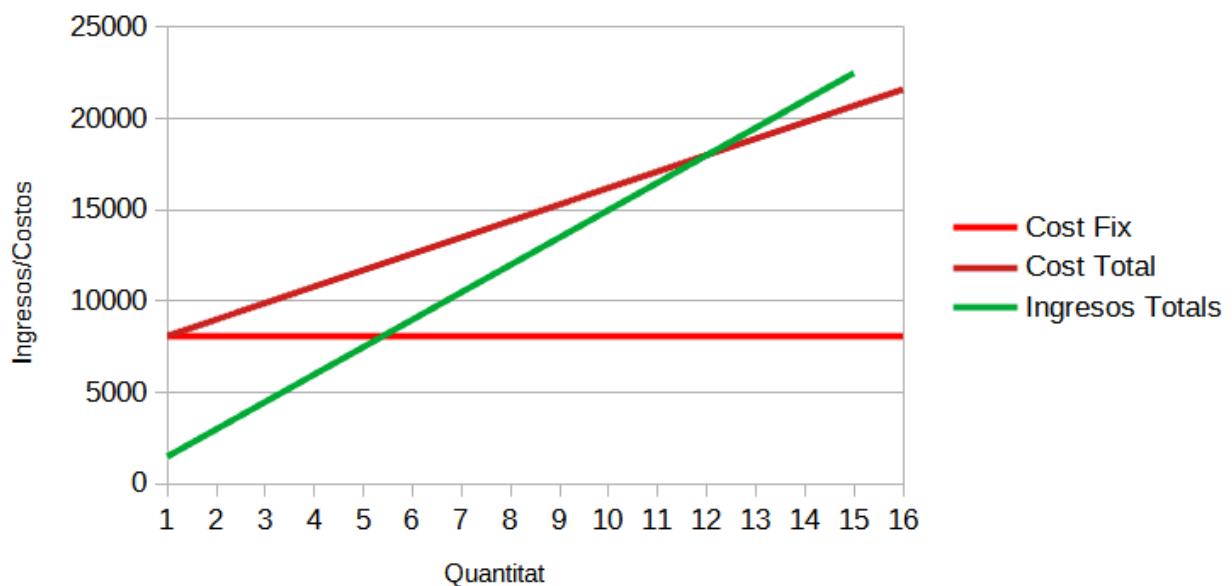
2.2.3 Càlcul del punt mort

2 versions de l'aplicació:

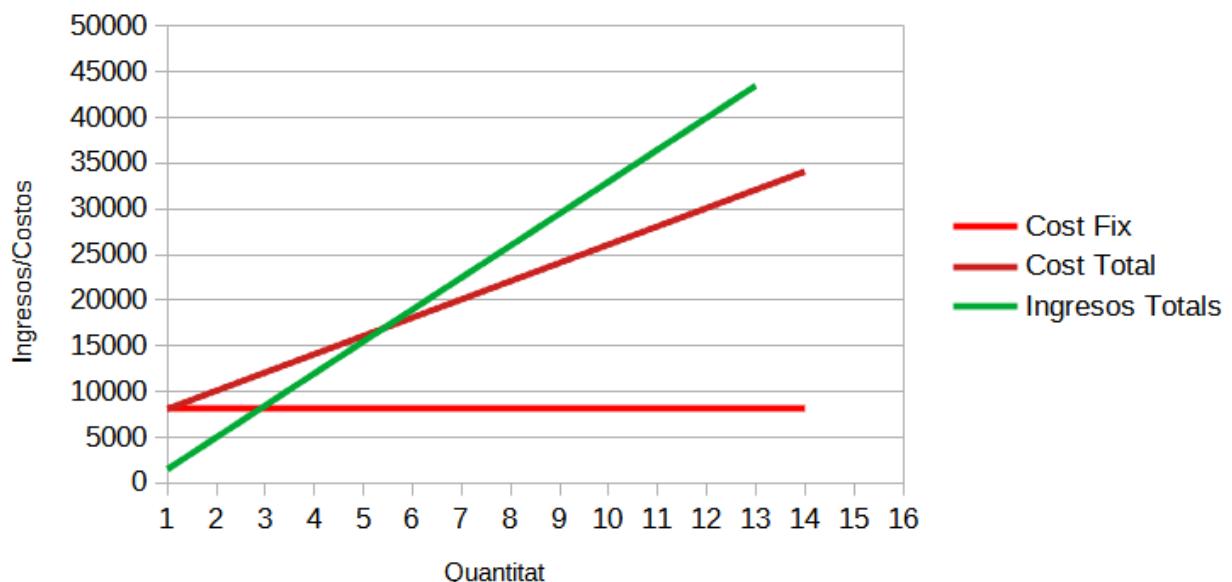
- Versió bàsica (els hi donem una plantilla i ells la customitzen) → 15€
- Versió PRO (ens diuen com la volen i nosaltres la customitzem) → 35€

Versió bàsica	Versió pro
$8107,10\text{€} / 15 - 9 \Rightarrow 1351,183$ Versions bàsiques [8107,10€ → Cost Fix (CF) 15€ → Preu (P) 9€ → Cost Variable unitari (CVu)]	$8107,10\text{€} / 35 - 20 \Rightarrow 540,473$ Versions PRO [8107,10€ → Cost Fix (CF) 35€ → Preu(P) 20€ → Cost Variable unitari (CVu)]
Comprovació	Comprovació
Inversió total = Preu producte * Quantitat = 15 * 1351,183 => 20267,745 Cost Total = CF + (CVu * Q) = 8107,1 + (9 * 1351,183) => 20267,747	Inversió total = Preu producte * Quantitat = 35 * 540,473 => 18916.555 Cost Total = CF + (CVu * Q) = 8107,1 + (20 * 540,473) => 18916,56

Punt mort (V. bàsica)



Punt mort (V. pro)



2.3 Dafo

Matriu Dafo			
Anàlisis intern	Fortaleses	Debilitats	
	<ul style="list-style-type: none"> - Coneixements - Treball en grup - Ganes d'aprendre / formarse - Iniciativa - La inversió inicial no és molt gran 	<ul style="list-style-type: none"> - Falta d'experiència - Falta de coneixements amplis - Competència - Baixa puntuació a Google Play 	
Analisis extern	Oportunitats	Amenaces	
	<ul style="list-style-type: none"> - Hi hauria institucions (com escoles) que potser estarien interessades per la facilitat d'ús de l'aplicació 	<ul style="list-style-type: none"> - És una aplicació que en qualsevol moment una empresa gran pot dir de fer-la i obtenir més vendes / beneficis 	

2.4 Viabilitat

Aquesta aplicació és un encàrrec de Prodis i s'ha fet de forma altruista, per tant, no s'ha obtingut cap prestació econòmica.

En cas d'haver desenvolupat l'aplicació per aconseguir beneficis, com hem vist en el punt mort tindríem un cost inicial de 8107,10€ i necessitaríem vendre 1352 versions bàsiques o 541 versions pro per començar a tenir beneficis.

2.5 Promoció

Com la nostra app va dirigida un públic molt específic la nostra promoció en tot cas es farà dins de Prodis.

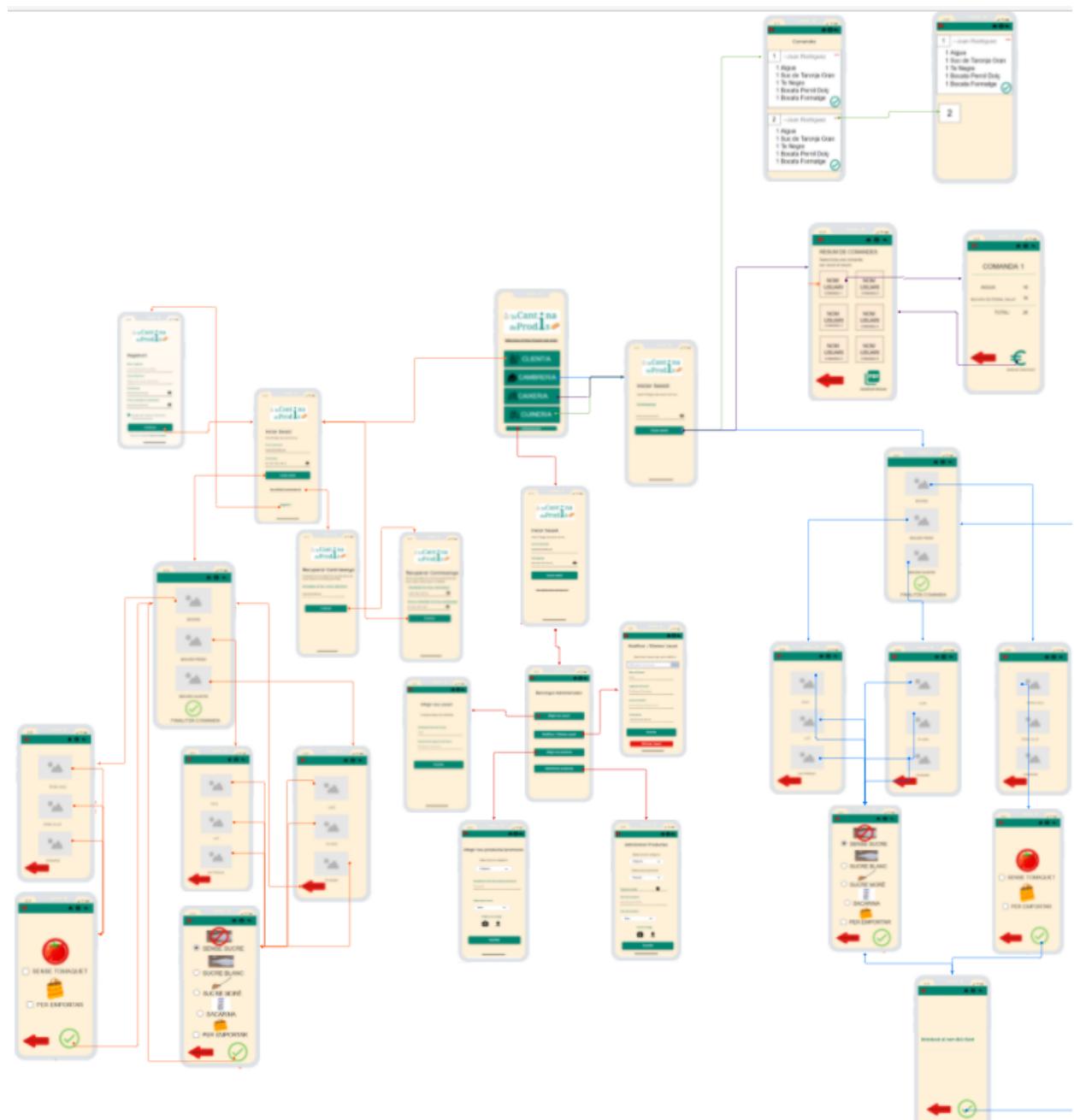
2.6 Distribució

En el cas de la distribució és ben senzill, l'aplicació es trobarà publicada al Google Play i els usuaris de Prodis només se l'hauran de descarregar.

3. Disseny de l'aplicació

3.1 Mockups

3.1.1 Diagrama de navegació



<https://app.moqups.com/vOw0QQ7h5z/view/page/ae8fe8eb0> (Navegació + pantalles cuiner)

<https://app.moqups.com/nKHKEoRc89/view/page/ae8fe8eb0> (Navegació pantalles selecció de productes)

<https://app.moqups.com/F85QPtq2Gg/view/page/ae8fe8eb0> (Pantalla principal, inicis de sessió, registre i recuperar contrasenya)

<https://app.moqups.com/feJktGYJWV/view/page/ae8fe8eb0> (Pantalla edició de usuari + resum de comandes)

https://marvelapp.com/2bhj6_c0q (Simulació de navegació entre pantalles)

3.1.2 Pantalles

3.1.2.1 Pantalla Principal

	<p>Pantalla que es mostrarà només iniciar l'aplicació.</p> <p>Amb cinc botons, quatre principals i un secundari.</p> <p>Cada botó principal té una imatge perquè sigui més intuïtiu.</p> <p>Cada botó obrirà una nova pantalla perquè puguin iniciar sessió.</p>
--	--

3.1.2.2 Pantalla Inici Sessió Client



Pantalla que es mostrarà quan l'usuari client vulgui iniciar sessió.

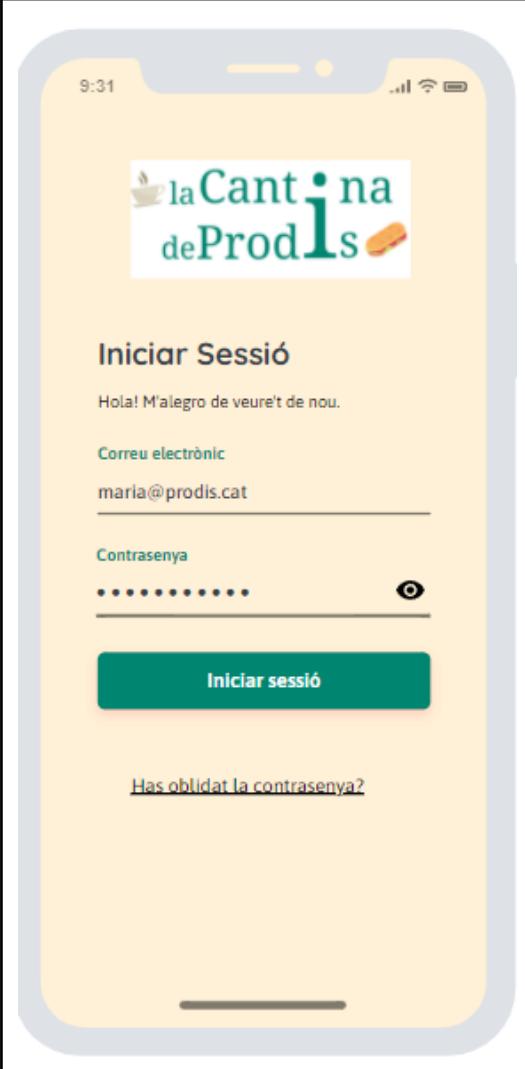
En aquesta pantalla l'usuari client pot:

- Iniciar Sessió introduint les dades necessàries i pitjant el botó d'iniciar sessió
- Recuperar la contrasenya prement sobre el text
- Recuperar la contrasenya pitjant sobre el text
- Registrar-se pitjant sobre el text

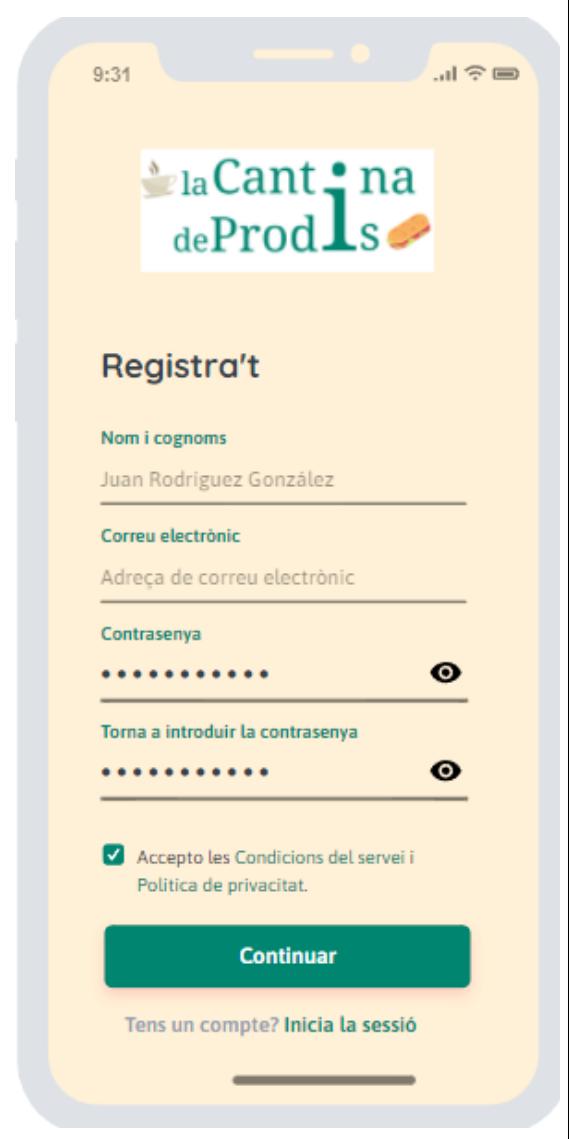
En pitjar sobre iniciar sessió, s'obrirà la següent pantalla, en cas contrari apareixerà un missatge d'error.

En pitjar sobre "Has oblidat la contrasenya?" s'obrirà la pantalla per poder recuperar-la.

3.1.2.3 Pantalla Inici Sessió Administrador

 A smartphone screen displaying the "Iniciar Sessió" (Login) page of the La Cantina de Prodís app. The screen shows the following fields: <ul style="list-style-type: none">Correu electrònic: maria@prodis.catContrasenya: (redacted)A button labeled "Iniciar sessió" (Start session).A link labeled "Has oblidat la contrasenya?" (Forgot password?). The background of the screen is light yellow.	<p>Pantalla que es mostrarà quan l'usuari administrador vulgui iniciar sessió.</p> <p>En aquesta pantalla l'usuari Administrador pot:</p> <ul style="list-style-type: none">- Iniciar Sessió introduint les dades necessàries i pitjant el botó d'iniciar sessió- Recuperar la contrasenya pitjant sobre el text <p>En pitjar sobre iniciar sessió, s'obrirà la següent pantalla, en cas contrari apareixerà un missatge d'error.</p> <p>En pitjar sobre "Has oblidat la contrasenya?" s'obrirà la pantalla per poder recuperar-la.</p>
---	--

3.1.2.4 Pantalla Registre

 <p>The image shows a smartphone displaying the registration page of the 'La Cantina de Prodís' app. The screen has a light yellow background. At the top, there's a header with the app's logo 'la Cantina de Prodís' featuring a coffee cup icon. Below the header, the title 'Registra't' is displayed. There are four input fields: 'Nom i cognoms' with the value 'Juan Rodriguez González', 'Correu electrònic' with the value 'Adreça de correu electrònic', 'Contrasenya' with a masked password, and another 'Contrasenya' field below it also with a masked password. Below these fields is a checkbox labeled 'Accepto les Condicions del servei i Política de privacitat.' followed by a checked checkbox icon. At the bottom of the screen is a large green button labeled 'Continuar'.</p>	<p>Pantalla que es mostrarà quan l'usuari client vulgi registrar-se.</p> <p>Per fer-ho haurà d'introduir:</p> <ul style="list-style-type: none">- Nom i cognoms- Correu electrònic amb domini "@prodis.cat"- Contrasenya- Tornar a introduir la contrasenya <p>Finalment, haurà d'acceptar les Condicions del servei i Política de privacitat, si no ho fa, no podrà continuar.</p> <p>Si l'usuari ja té un compte, podrà iniciar sessió pitjant sobre "inicia sessió" que hi ha a la part d'avall de la pantalla.</p>
--	---

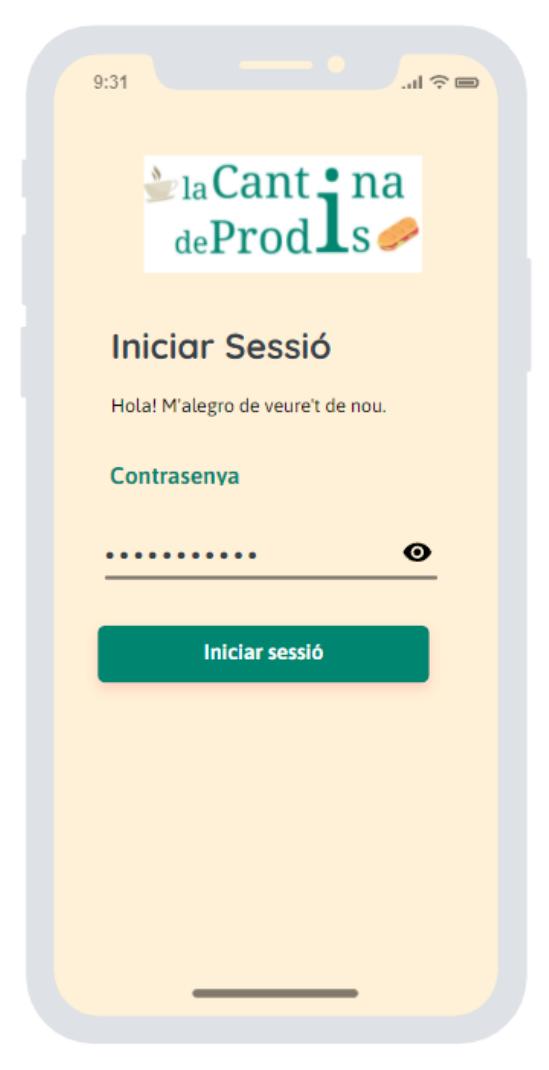
3.1.2.5 Pantalla Recuperar Contrasenya I

	<p>Pantalla que es mostrarà quan l'usuari client/administrador vulgui recuperar la contrasenya.</p> <p>Primer de tot se li demanarà que introdueixi el seu correu electrònic.</p> <p>Si el correu consta a la base de dades, deixarà a l'usuari seguir.</p> <p>En cas contrari se li mostrarà per pantalla un missatge avisant que aquell correu no consta a la base de dades.</p>
--	--

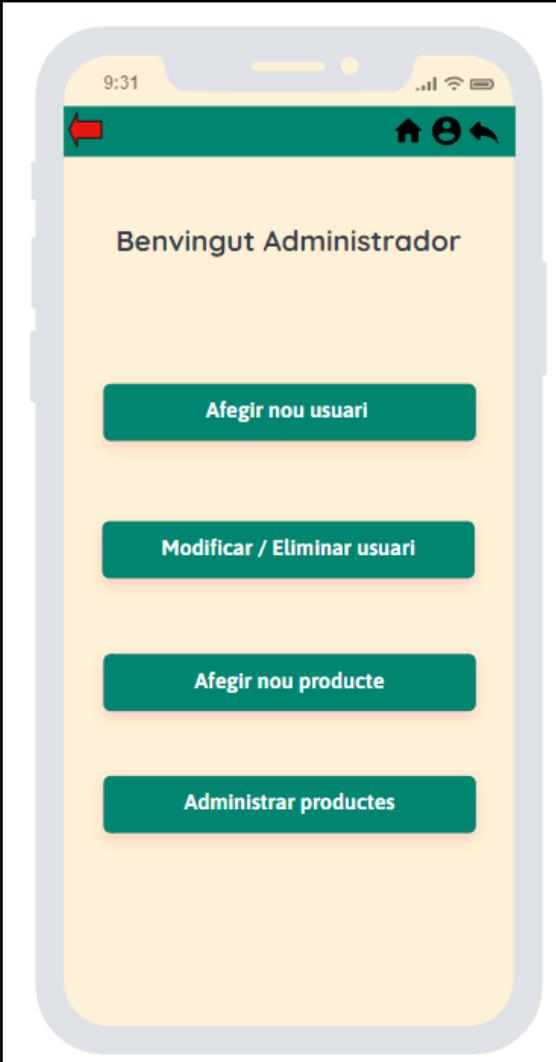
3.1.2.6 Pantalla Recuperar Contrasenya II

	<p>Pantalla que es mostrarà quan l'usuari client/administrador estigui intentant recuperar la contrasenya.</p> <p>En aquesta pantalla s'obliga a l'usuari a introduir dues vegades la nova contrasenya, verificant que siguin la mateixa i diferent de la que tenia.</p> <p>Un cop l'usuari ho hagi fet, se li habilitarà el botó "Continuar", el qual el portarà a la pantalla d'inici de sessió de l'usuari Client o Administrador.</p>
--	---

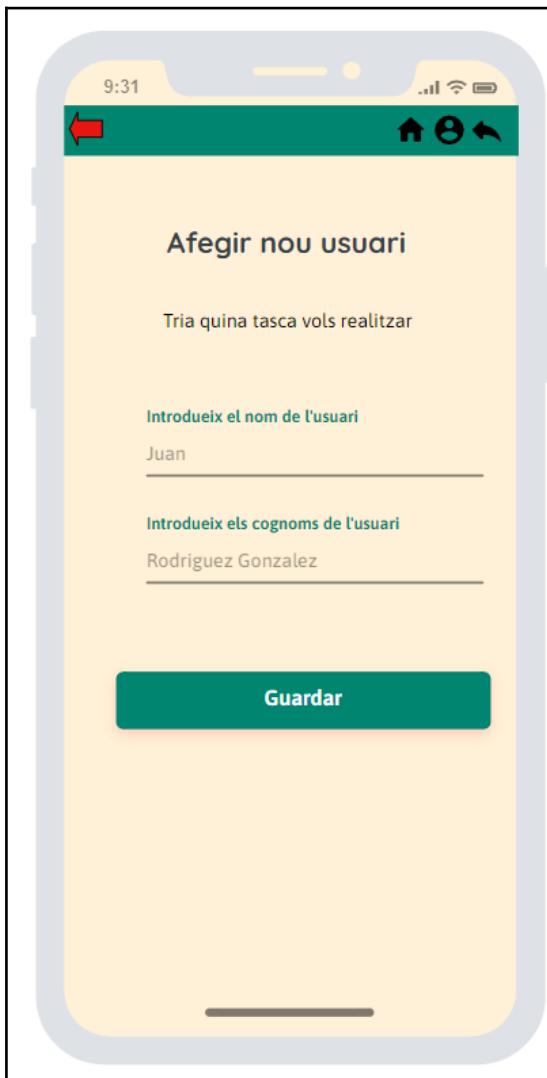
3.1.2.7 Pantalla Inici Sessió Cambrer - Cantina-Caixer

	<p>Pantalla que es mostrarà quan els usuaris: cambrer o cantina vulguin iniciar sessió.</p> <p>En aquesta pantalla els usuaris només podran introduir una contrasenya la qual serà fàcil (exemple: cantina).</p> <p>Un cop introduïda la contrasenya s'habilitarà el botó d'iniciar sessió.</p> <p>Si la contrasenya és la correcta, se li mostrà la següent pantalla, en cas contrari apareixerà un missatge d'error.</p>
--	--

3.1.2.8 Pantalla Administrador I

	<p>Pantalla que es mostrarà un cop l'usuari administrador hagi iniciat sessió.</p> <p>En aquesta pantalla l'usuari administrador seleccionarà l'acció que vol fer:</p> <ul style="list-style-type: none"> - Afegir nou usuari - Modificar / Eliminar usuari - Afegir nou producte <p>En cas de voler tancar la sessió, només haurà de pitjar el botó de menú i fer clic en tancar sessió.</p> <p>Botó “home”.</p> <p>Botó per editar perfil.</p> <p>Botó per tancar la sessió.</p> <p>Botó per tornar enrere</p>
--	---

3.1.2.9 Pantalla Administrador II

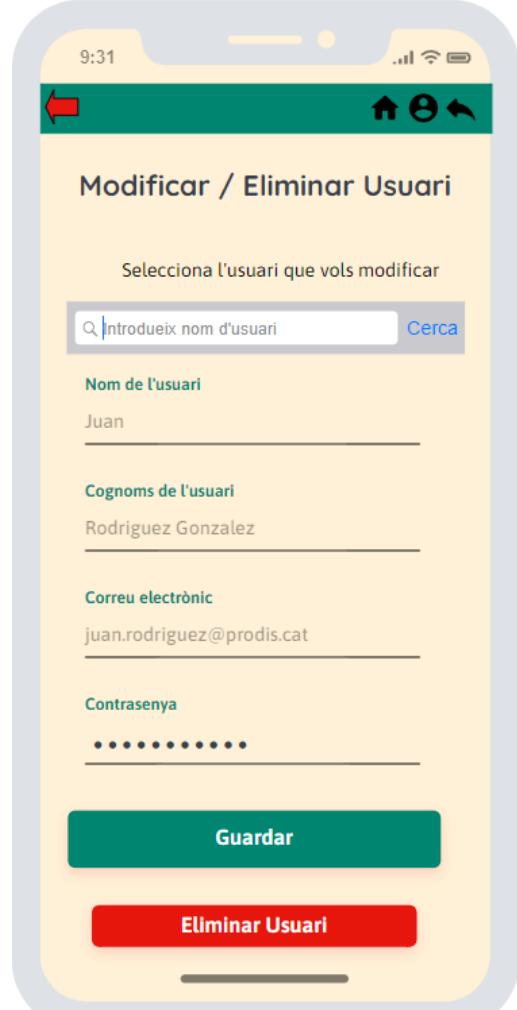


Pantalla que es mostrarà a l'usuari administrador quan vulgui afegir un nou usuari, el qual apareixerà al llistat d'usuaris en el perfil del cambrer.

Per afegir un nou usuari, l'usuari administrador haurà d'introduir el nom de l'usuari i els cognoms de l'usuari.

Un cop ho hagi fet se li habilitarà el botó "Fet" el qual afegirà el nou usuari a la base de dades.

3.1.2.10 Pantalla Administrador III

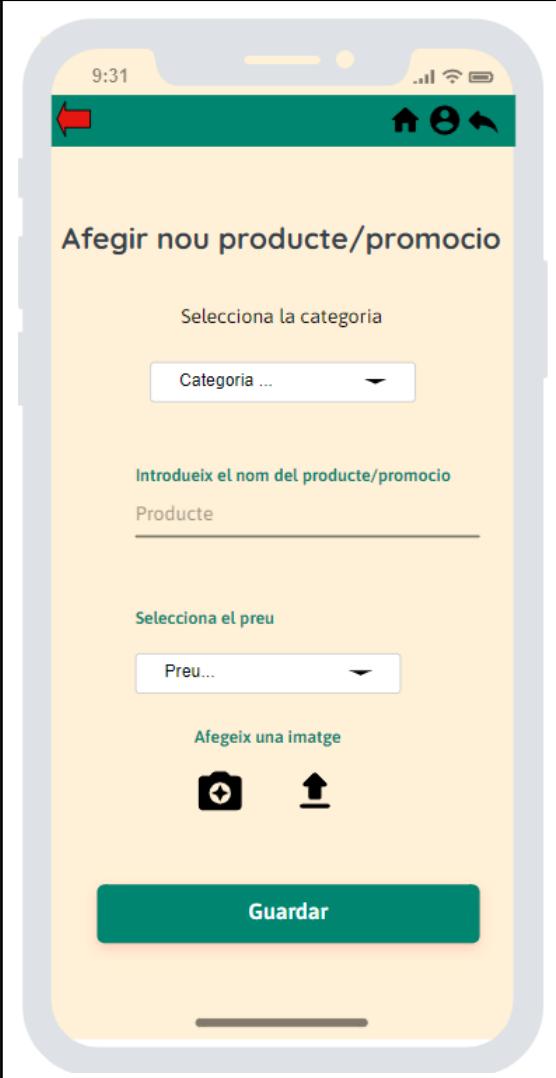


Pantalla que es mostrarà a l'usuari administrador quan vulgui modificar o eliminar un usuari.

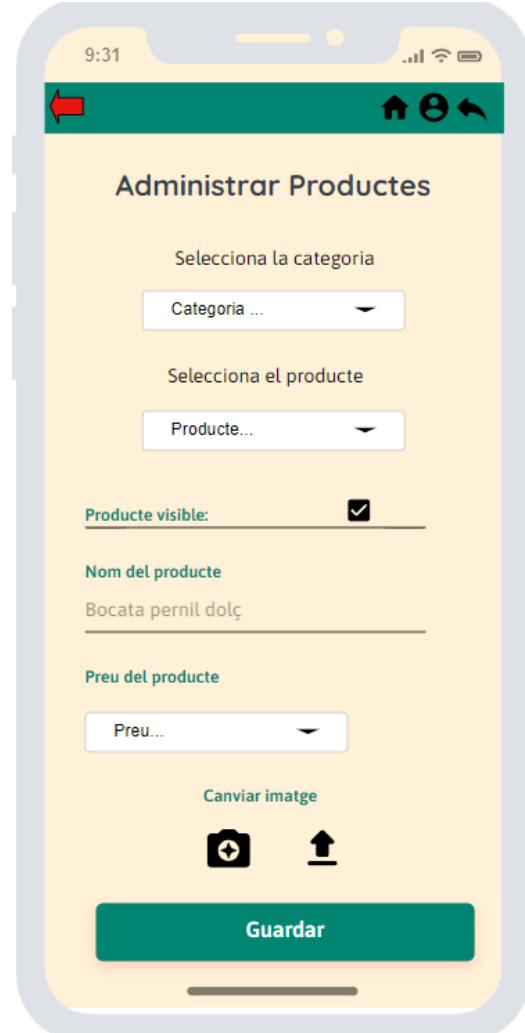
Per modificar un usuari, primer haurà de buscar el nom i cognoms de l'usuari, mitjançant un llistat d'usuaris, un cop fet, se li mostraran els camps d'aquell usuari amb la seva informació; en funció del tipus d'usuari se li mostrarà més o menys informació.

Per altra banda, per eliminar aquell usuari, haurà de fer clic en "Eliminar Usuari" i se li demanarà que confirmi l'eliminació de l'usuari.

3.1.2.11 Pantalla Administrador IV

	<p>Pantalla que es mostrarà a l'usuari administrador quan vulgui afegir un nou producte / promoció.</p> <p>Per fer-ho, primer haurà de seleccionar la categoria.</p> <p>Un cop seleccionada la categoria, només haurà d'introduir el nom del producte o promoció, seleccionar el preu i afegir una imatge, ja sigui fent servir la càmera del mòbil o pujant-la des de la galeria.</p> <p>Un cop fet tot, només haurà de pitjar en guardar.</p>
--	---

3.1.2.12 Pantalla Administrador V



Pantalla que es mostrarà a l'usuari administrador quan vulgui modificar algun producte.

Per fer-ho, primer seleccionarà la categoria, a continuació el producte.

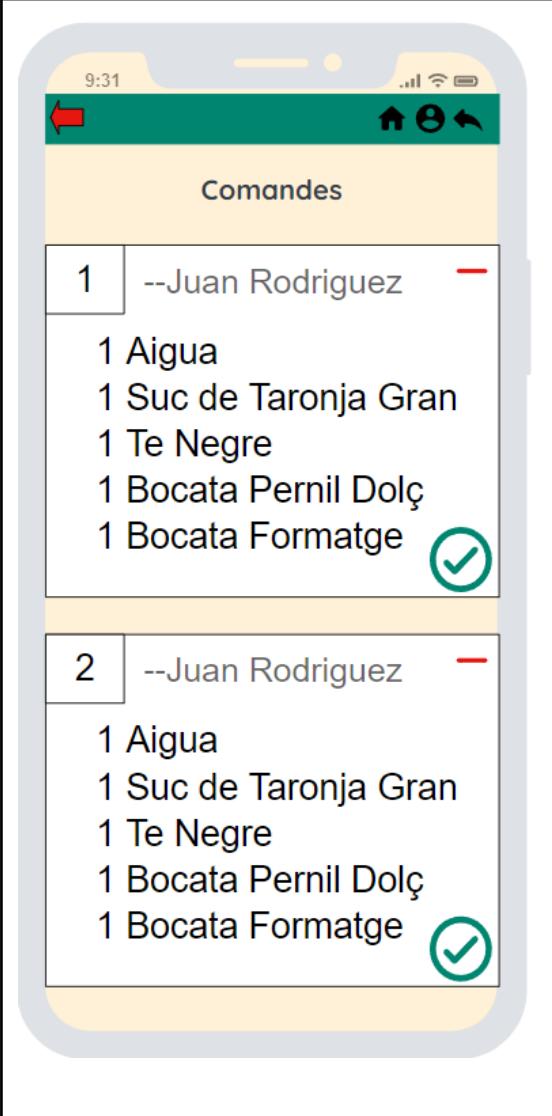
Un cop fet, se li mostrarà:

- Si el producte està visible
- El nom del producte
- El preu

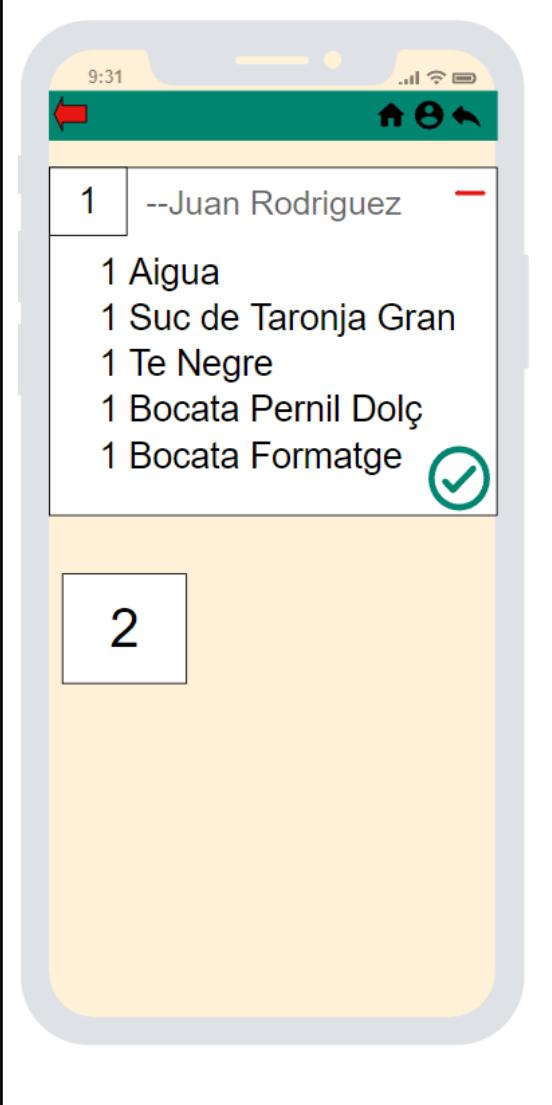
També tindrà l'opció de canviar la imatge, sigui fent una nova foto, o pujant-la des de la galeria.

Finalment, haurà de pitjar en guardar per aplicar els canvis.

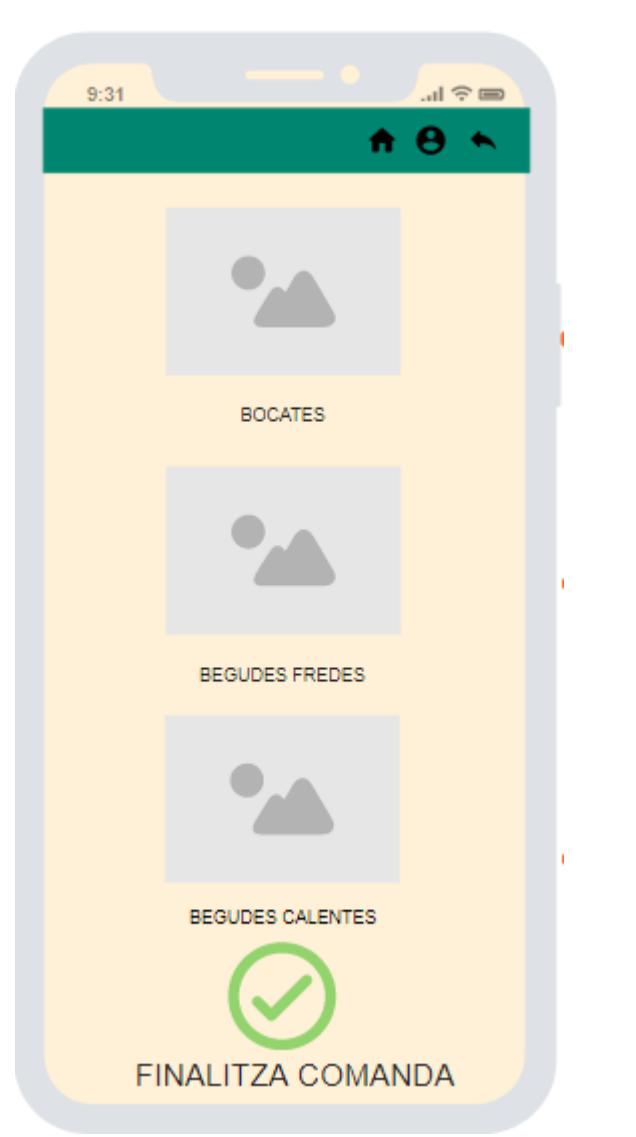
3.1.2.13 Pantalla Comandes /

	<p>Pantalla que es mostrerà a l'usuari cantina quan pitgi sobre el número d'una comanda.</p> <p>Igual que la pantalla anterior tenim dues seccions: begudes i entrepans.</p> <p>En aquest cas en cada secció se'n mostra:</p> <ul style="list-style-type: none"> Número de comanda Nom del client Productes que ha demanat el client <p>També se'n mostra un botó en forma de tic el qual en pitjar-lo s'eliminarà la comanda i es tornaran a veure els números de comandes pendents.</p> <p>En cas que l'usuari vulgui tancar la comanda sense eliminar-la s'haurà de pitjar a sobre de la comanda o el botó vermell de tornar enrere.</p> <p>També tenim un botó per obrir el menú lateral.</p>
--	--

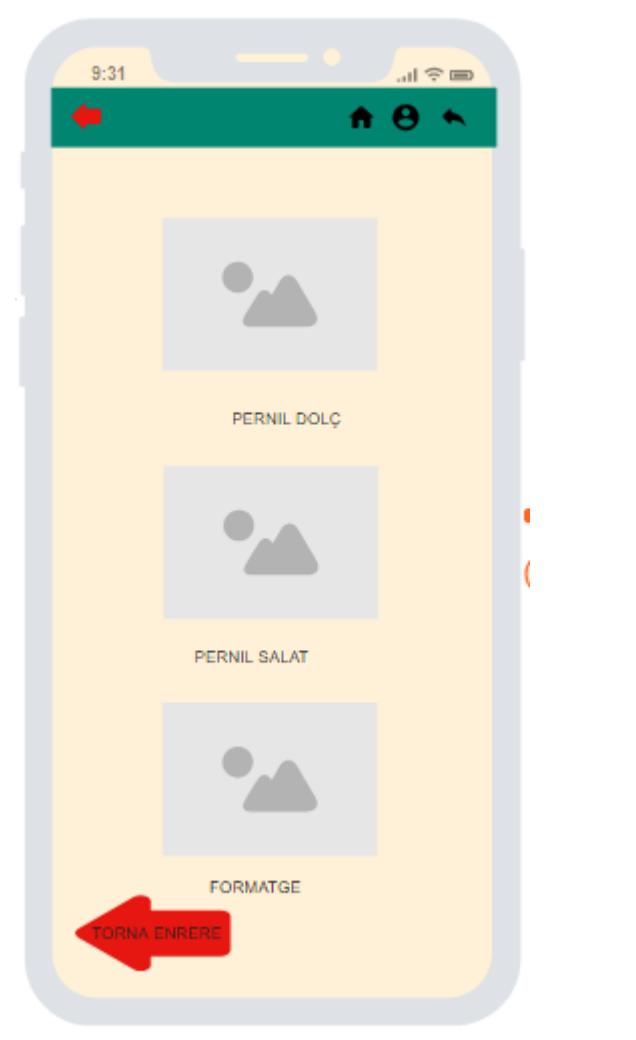
3.1.2.14 Pantalla Comandes II

	<p>Pantalla que es mostrarà a l'usuari cantina quan pitgi sobre el número d'una comanda.</p> <p>Igual que la pantalla anterior tenim dues seccions: begudes i entrepans.</p> <p>En aquest cas en cada secció se'n mostra:</p> <ul style="list-style-type: none"> Número de comanda Nom del client Productes que ha demanat el client <p>També se'n mostra un botó en forma de tic el qual en pitjar-lo s'eliminarà la comanda i es tornaran a veure els números de comandes pendents.</p> <p>En cas que l'usuari vulgui tancar la comanda sense eliminar-la s'haurà de pitjar a sobre de la comanda o el botó vermell de tornar enrere.</p> <p>També tenim un botó per obrir el menú lateral.</p>
--	--

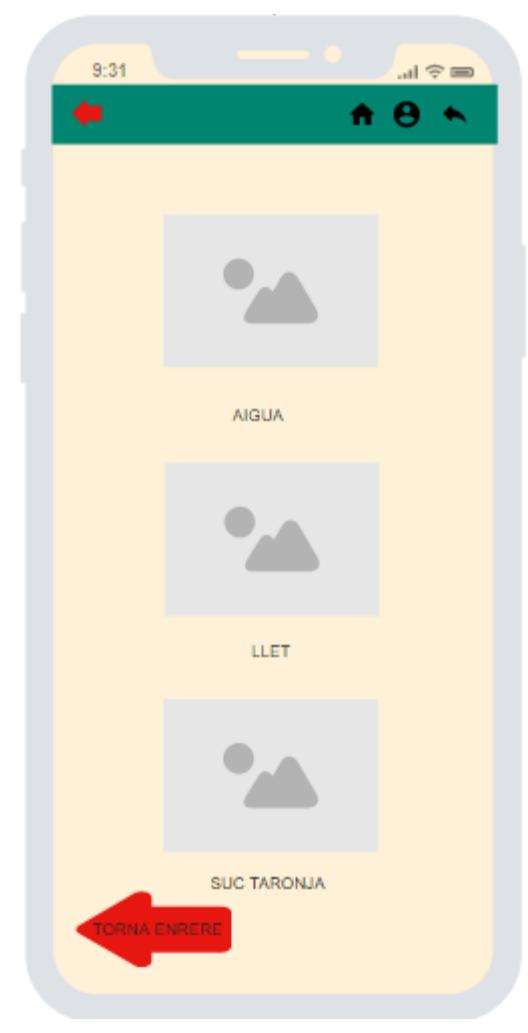
3.1.2.15 Pantalla selecció tipus de producte

	<p>Pantalla que sortirà en registrar-se un client o cambrer.</p> <p>En aquesta pantalla sortiran els tres tipus de productes que es venen en la cantina. Els botons seran tant la imatge com el text.</p> <p>En la part superior esquerra hi ha un botó que obre el menú desplegable.</p> <p>Botó que finalitza la comanda.</p> <p>Botó imatge/text.</p>
--	--

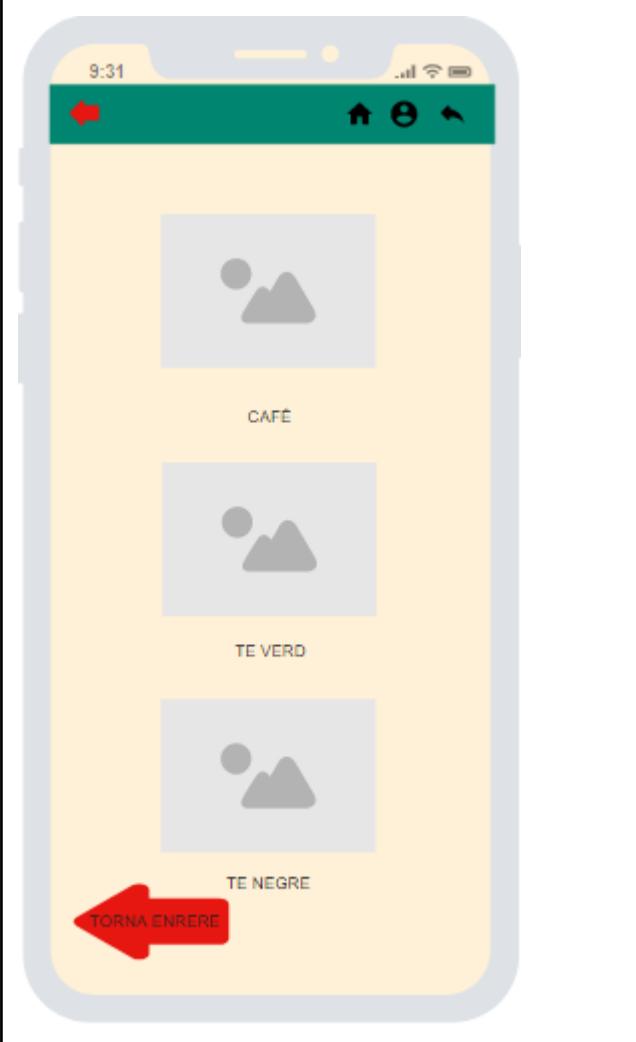
3.1.2.16 Pantalla selecció bocata

	<p>Pantalla a la qual arribarà l'usuari en seleccionar bocates a la pàgina principal.</p> <p>En aquesta pantalla hi haurà tots els bocates que hi ha en la cantina de Prodis. Serà un menú amb "scroll".</p> <p>Els botons seran tant la imatge com el text.</p> <p>En la part superior esquerra hi ha un botó que obre el menú desplegable.</p> <p>Botó per tornar enrere en la part superior i inferior esquerre.</p> <p>Botó imatge i text</p> <p>Botó menú desplegable</p> <p>Botó per tornar enrere</p>
--	---

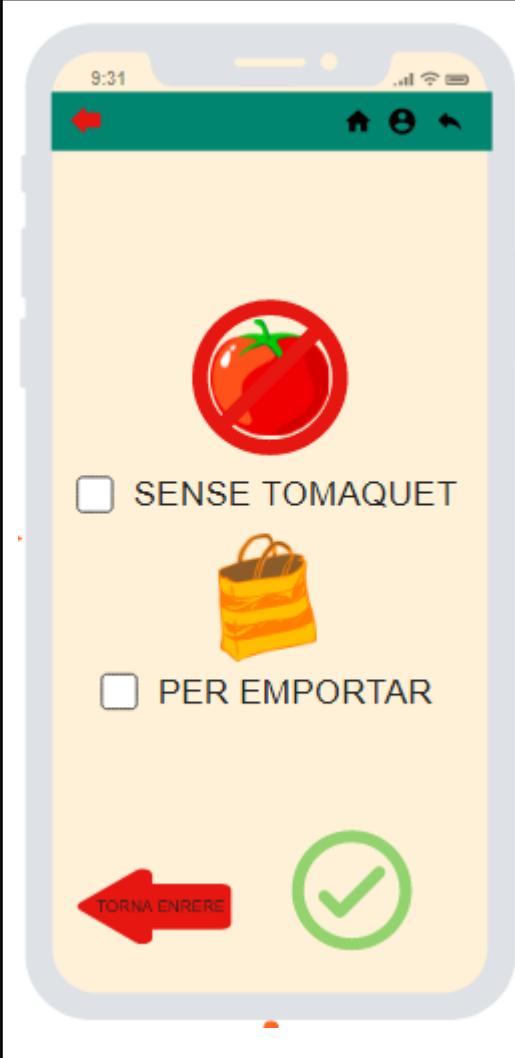
3.1.2.17 Pantalla selecció beguda freda

	<p>Pantalla a la qual arribarà l'usuari en seleccionar begudes fredes a la pàgina principal.</p> <p>En aquesta pantalla hi haurà totes les begudes fredes que hi ha en la cantina de Prodis. Serà un menú amb "scroll".</p> <p>Els botons seran tant la imatge com el text.</p> <p>En la part superior esquerra hi ha un botó que obre el menú desplegable.</p> <p>Botó per tornar enrere en la part superior i inferior esquerre.</p> <p>Botó imatge i text</p> <p>Botó menú desplegable</p> <p>Botons per tornar enrere</p>
--	--

3.1.2.18 Pantalla selecció beguda calenta

	<p>Pantalla a la qual arribarà l'usuari en seleccionar begudes calentes a la pàgina principal.</p> <p>En aquesta pantalla hi haurà totes les begudes calentes que hi ha en la cantina de Prodis. Serà un menú amb "scroll".</p> <p>Els botons seran tant la imatge com el text.</p> <p>En la part superior esquerra hi ha un botó que obre el menú desplegable.</p> <p>Botó per tornar enrere en la part superior i inferior esquerre.</p> <p>Botó imatge i text</p> <p>Botó menú desplegable</p> <p>Botons per tornar enrere</p>
--	--

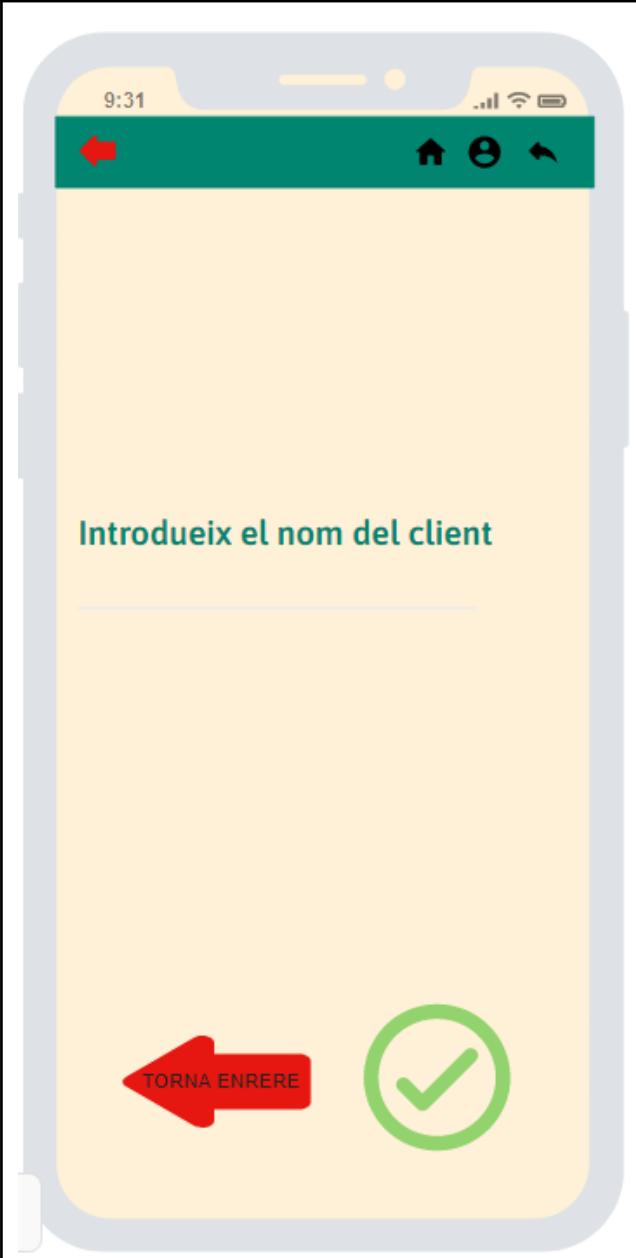
3.1.2.19 Pantalla selecció atributs del bocata

	<p>Pantalla a la qual s'arribarà en seleccionar el bocata en la pantalla de selecció de bocates.</p> <p>En aquesta pantalla se seleccionaran els atributs que ha demanat el client del bocata seleccionat.</p> <p>Els botons seran tant la imatge com el text.</p> <p>En la part superior esquerra hi ha un botó que obre el menú desplegable.</p> <p>Botó per tornar enrere en la part superior i inferior esquerre.</p> <p>Diferents tipus de botons per seleccionar els atributs del bocata i la comanda.</p> <p>Botó per confirmar la comanda.</p> <p>Botó imatge i text.</p> <p>Botó menú desplegable.</p> <p>Botons per tornar enrere.</p> <p>"Check box" per seleccionar si vol tomàquet i si vol la comanda per emportar.</p> <p>Botó en forma de tic per confirmar la comanda.</p>
--	--

3.1.2.20 Pantalla selecció atributs de la beguda

	<p>Pantalla a la qual s'arribarà en seleccionar tant la beguda calenta com la beguda freda en la pantalla de selecció de begudes.</p> <p>En aquesta pantalla se seleccionaran els atributs que ha demanat el client sobre la beguda seleccionada.</p> <p>Els botons seran tant la imatge com el text.</p> <p>En la part superior esquerra hi ha un botó que obre el menú desplegable.</p> <p>Botó per tornar enrere en la part superior i inferior esquerre.</p> <p>Diferents tipus de botons per seleccionar els atributs del bocata i la comanda.</p> <p>Botó per confirmar la comanda.</p> <p>Botó imatge i text.</p> <p>Botó menú desplegable.</p> <p>Botons per tornar enrere.</p> <p>"Check box" per seleccionar si vol tomàquet i si vol la comanda per emportar.</p> <p>Botó en forma de tic per confirmar la comanda.</p>
--	--

3.1.2.21 Pantalla selecció nom client comanda

 <p>The image shows a smartphone screen with a light blue background. At the top, there is a green navigation bar with a back arrow icon, a home icon, a power button icon, and a recent apps icon. Below the navigation bar, the text "Introduceix el nom del client" is displayed in a teal font. A horizontal line follows this text. At the bottom, there are two large, rounded rectangular buttons: a red one on the left with the text "TORNA ENRERE" and a green one on the right with a white checkmark icon.</p>	<p>Pantalla per introduir el nom del client.</p> <p>Pantalla on s'introduirà el nom del client de la comanda.</p> <p>Text per introduir el nom del client.</p> <p>Botons per tornar enrere.</p> <p>Botó en forma de tic per confirmar la comanda.</p>
--	--

3.1.2.22 Pantalla de edició de perfil

	<p>Pantalla per editar el compte de l'usuari.</p> <p>En aquesta pantalla es podrà tomar una foto per al perfil, editar el nom, cognom i correu electrònic de l'usuari.</p> <p>Botó per tornar enrere en la part superior esquerra.</p> <p>Text per canviar el nom, cognom i correu electrònic.</p> <p>Botó per recuperar la contrasenya.</p> <p>Botó per acceptar els canvis.</p> <p>Botó per canviar de foto.</p> <p>Espais de text per escriure la nova informació.</p> <p>Botó per canviar la mida de la lletra.</p>
--	---

3.1.2.23 Pantalla resum comanda



The screenshot shows a mobile application interface. At the top, there's a green header bar with a back arrow, a home icon, and a refresh/clock icon. Below the header, the text 'NOM USUARI' and 'COMANDA 1' is displayed. The main content area lists items with their prices:

AIGUA	1€
BOCATA DE PERNIL SALAT	1€
TOTAL:	2€

At the bottom left is a large red arrow pointing left, labeled 'Torna enrere'. To its right is a teal euro symbol with the text 'MARCAR COM PAGAT' below it.

Pantalla per veure el resum de la comanda.

En aquesta pantalla es podrà veure el preu total de la comanda seleccionada i també es podrà imprimir la comanda.

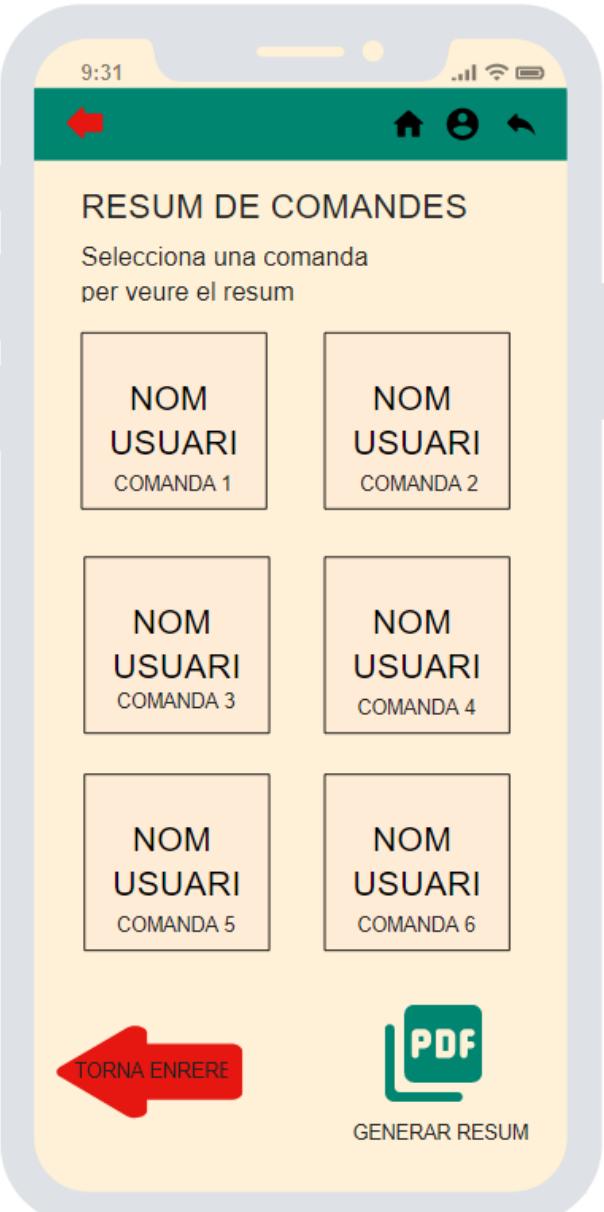
Botó per tornar enrere en la part superior i inferior esquerre.

També hi ha un botó per imprimir.

Botons per tornar enrere.

Botó per marcar com pagat.

3.1.2.24 Pantalla selecció resum de la comanda



The screenshot shows a mobile application interface. At the top, there's a green header bar with a back arrow, a home icon, and a refresh/clock icon. Below it, the title "RESUM DE COMANDES" is centered. A sub-instruction "Selecciona una comanda per veure el resum" is displayed. The main content area contains a 3x2 grid of boxes, each labeled "NOM USUARI" and "COMANDA X". The boxes are arranged as follows:

NOM USUARI COMANDA 1	NOM USUARI COMANDA 2
NOM USUARI COMANDA 3	NOM USUARI COMANDA 4
NOM USUARI COMANDA 5	NOM USUARI COMANDA 6

At the bottom left is a large red "Torna enrere" button with a left arrow. At the bottom right is a green "GENERAR RESUM" button with a PDF icon.

Pantalla per seleccionar el resum de la comanda.

En aquesta pantalla sortiran totes les comandes fets. Es podrà seleccionar un número per veure el resum de la comanda.

Botó per tornar enrere en la part superior i inferior esquerre.

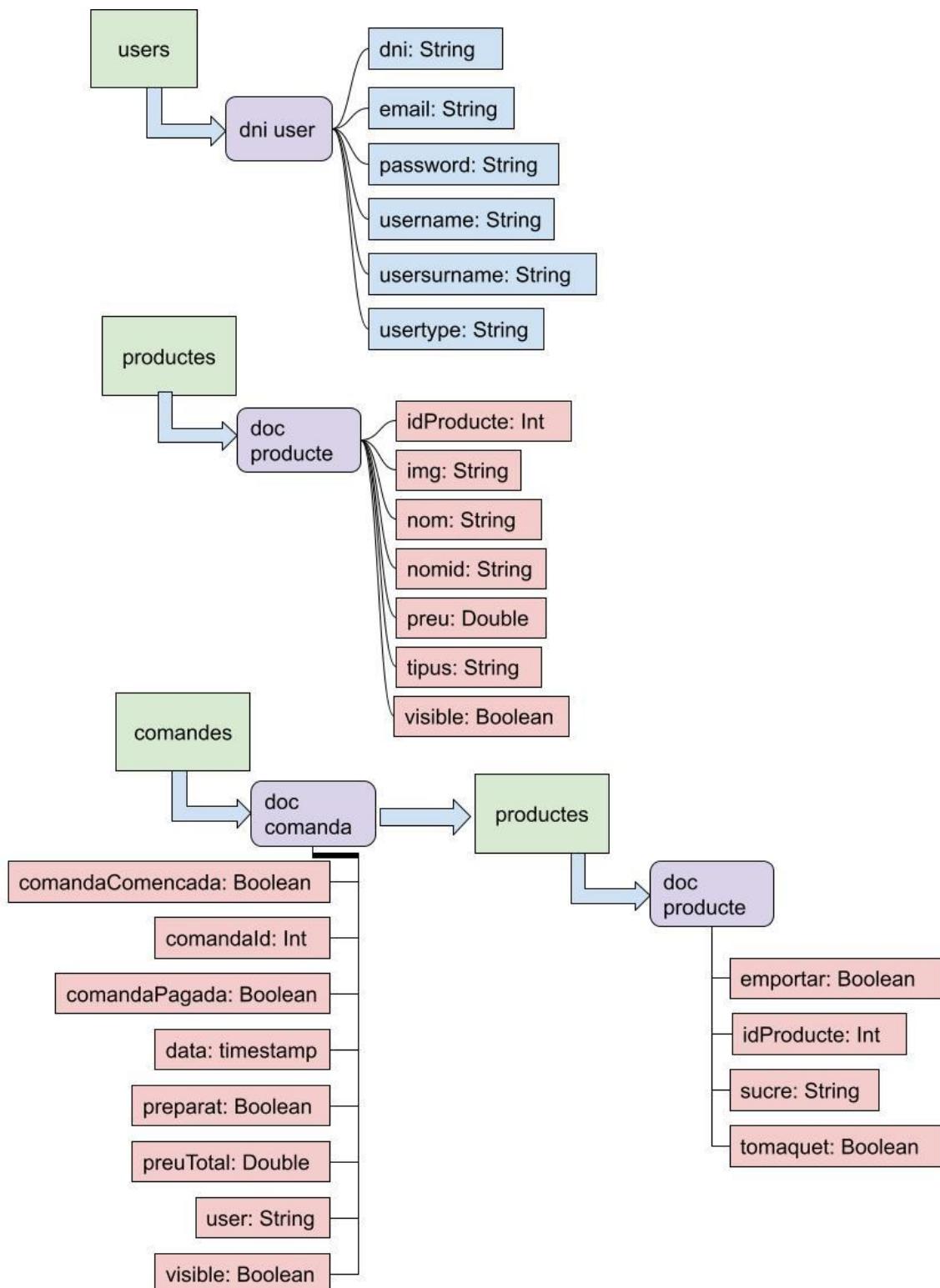
Es podrà generar un resum diari de totes les comandes del dia.

Botons per tornar enrere.

Botó per generar el resum.

Botons per veure el resum de la comanda.

3.2 Disseny de la base de dades NoSQL



Home > comandes > 1

lacantinadeprodis-dae70	comandes	1
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
comandes >	1 >	+ Agregar campo
productes		- 1
users		emportar: true
		quantitat: 2
		► sucre: ["SS", "SC"]
		- 3
		emportar: true
		quantitat: 2
		▼ tomaquet
		0 true
		1 false
		preuTotal: 3.5
		status: true
		user: "00000000T"

4. Planificació i seguiment de les tasques de desenvolupament

4.1 Eines

4.1.1 Gestor de projectes Trello

Funcionalitats:

Trello és un administrador de projectes amb interfície web, Android i iOS desenvolupat en JavaScript. Utilitzant el sistema kanban (sistema d'organització harmònic japonès) permet:

- Organitzar tasques
- Afegir llistes
- Adjuntar arxius
- Etiquetar esdeveniments
- Afegir comentaris
- Compartir taulers.

A més permet treballar de manera cooperativa permetent compartir els espais de treball amb altres persones.

Característiques:

Gràcies a una interfície molt intuitiva, permet gestionar totes les tasques de l'espai de treball molt fàcilment. Amb l'ajut d'etiquetes es poden administrar les tasques per colors, cosa que a nosaltres ens ha servit de gran ajuda. També permet introduir quina persona treballa a cada tasca, posant a la part inferior esquerra la persona involucrada en aquesta. Addicionalment, ens ha agradat que és en cada tasca es poden ficar les dates d'inici i finalització, a més a més de ficar si han estat completades, si s'estan fent en aquests moments o si estan fora de termini.



<https://trello.com/>

4.1.2 Diagrama de gantt - TeamGantt

TeamGantt és una eina especialitzada en la creació de diagrames de Gantt per a l'optimització en l'organització del teu projecte. Amb aquest mètode de gestió de projectes s'estalvia temps i esforç i es garanteix el desenvolupament programat dels teus projectes. TeamGantt forma part dels plugins de Trello, per tant, en crear el nostre projecte Trello podrem gestionar l'organització del projecte mitjançant aquesta eina de creació de diagrames de Gantt.

- Indicar el percentatge de realització de la tasca
- Indicar quin usuari dur a terme cada tasca
- Assignar colors a la tasca
- Assignar dia d'inici i dia de fi de la tasca
- Creació de grups per una organització de les tasques més ordenada i visualment simple d'entendre.
- Maneig fàcil, intuïtiu i àgil.
- Pots realitzar qualsevol acció des de qualsevol apartat de l'eina
- Enfocada a l'eficàcia de l'usuari
- Eina col·laborativa (encara que la versió gratuïta només admeti 3 usuaris).

Característiques:

Hi ha moltes funcions que giren entorn de facilitar la col·laboració com: conversacions, gestió de documents, seguiment de temps i dependències (projectes de mapeix que depenen d'altres projectes).



<https://www.teamgantt.com/>

4.1.3 Issues del gitlab

Funcionalitats:

Els issues del gitlab s'utilitzen per:

- Col·laborar en idees.
- Resoldre problemes.
- Planificar el treball.
- Compartir i discutir les propostes de treball amb el teu equip o altres col·laboradors.

Característiques:

També es poden utilitzar els issues per moltes altres funcions, adaptades a les teves necessitats i especificacions del teu treball:

- Discutir les implementacions de les idees.
- Fer el seguiment de l'estat del treball.
- Acceptar les propostes de característiques, idees, peticions del suport o reportar bugs.
- Elaborar implementacions per al codi.

Cada issue està associat al seu projecte. Si tens diferents projectes en un grup, pots veure tots els issues del projecte a la vegada

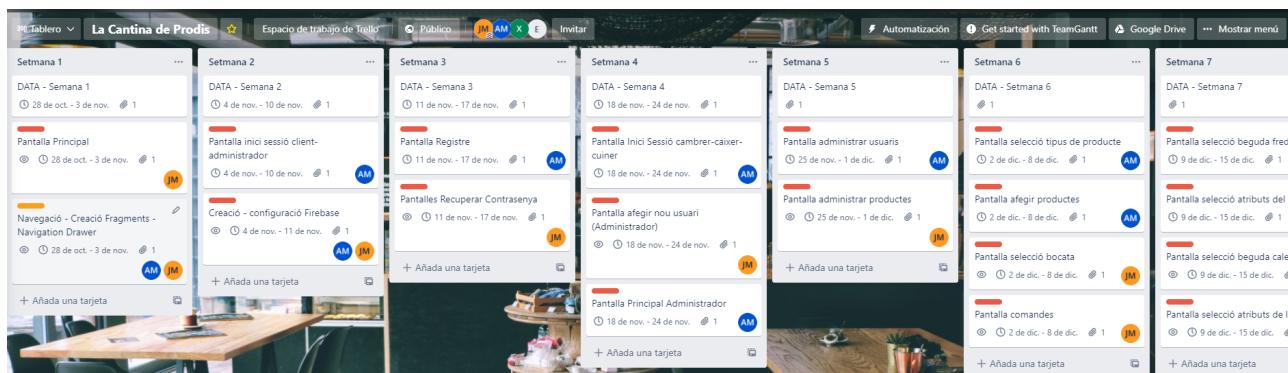
Funcionament dels Issues:

https://forge.etsi.org/rep/help/user/project/issues/create_new_issue.md



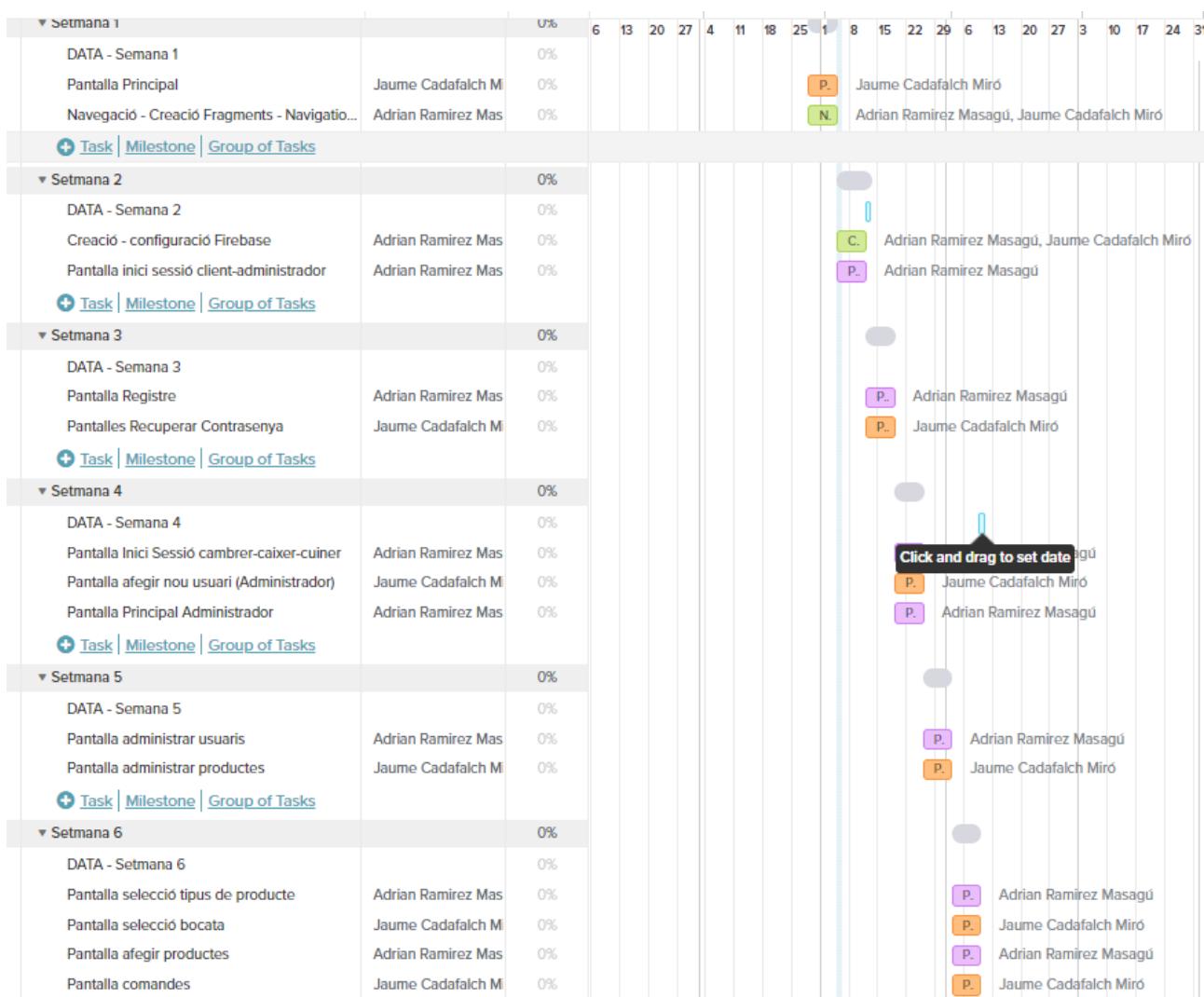
<https://gitlab.com/gitlab-org/gitlab/-/issues>

4.2 Planificació Inicial



The screenshot shows a Trello board with 7 columns representing weeks. Each column contains several cards with task details and due dates. The tasks include various screen recordings and configurations for a mobile application.

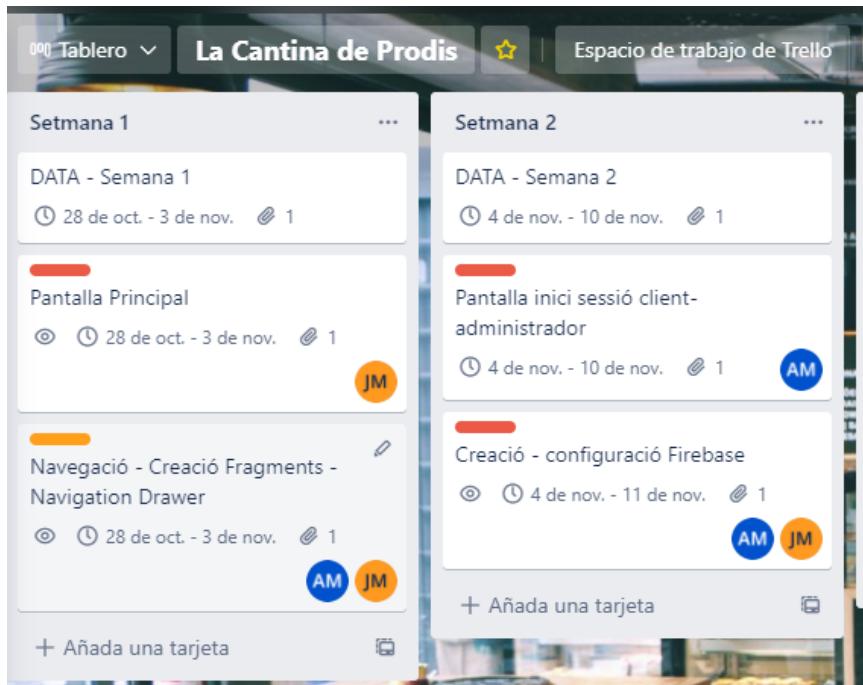
[Clic aquí per accedir a la visualització del tauler](#)



[Clic aquí per accedir al projecte TeamGantt \[Jaume - Adrián - Xesca\]](#)

4.3 Seguiment Setmanal

4.3.1 Setmana 1

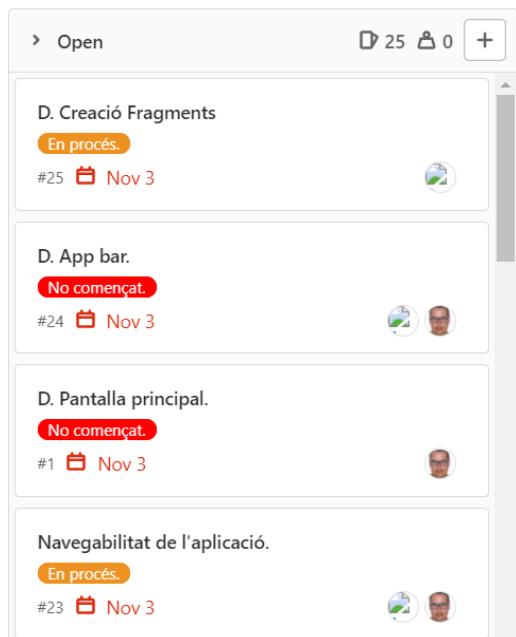


Setmana 1

- DATA - Semana 1
 - 28 de oct. - 3 de nov.
 - 1
- Pantalla Principal
 - 28 de oct. - 3 de nov.
 - 1

Setmana 2

- DATA - Semana 2
 - 4 de nov. - 10 de nov.
 - 1
- Pantalla inici sessió client-administrador
 - 4 de nov. - 10 de nov.
 - 1
- Creació - configuració Firebase
 - 4 de nov. - 11 de nov.
 - 1



Open

D. Creació Fragments
En procés.
#25 Nov 3

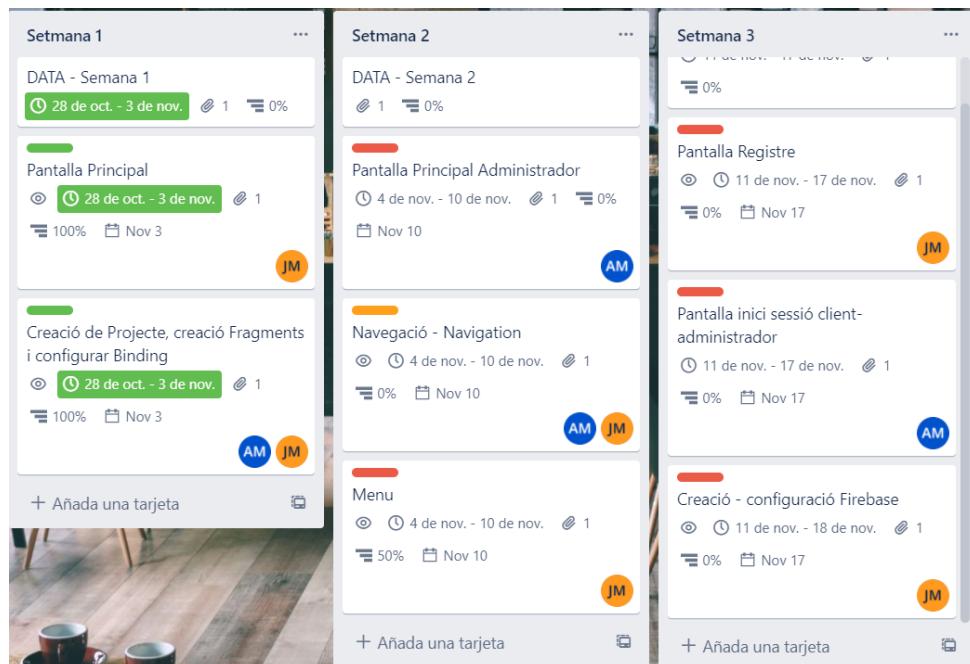
D. App bar.
No començat.
#24 Nov 3

D. Pantalla principal.
No començat.
#1 Nov 3

Navegabilitat de l'aplicació.
En procés.
#23 Nov 3

En aquesta setmana havíem de crear el projecte, connectar-lo amb GitLab, decidir quins activities i fragments tindrem, dissenyar la pantalla principal i fer la navegació d'ells.

Per problemes amb el binding no hem pogut fer la navegació, però hem començat a dissenyar visualment les pantalles d'inici de sessió i de registre.



Setmana 1

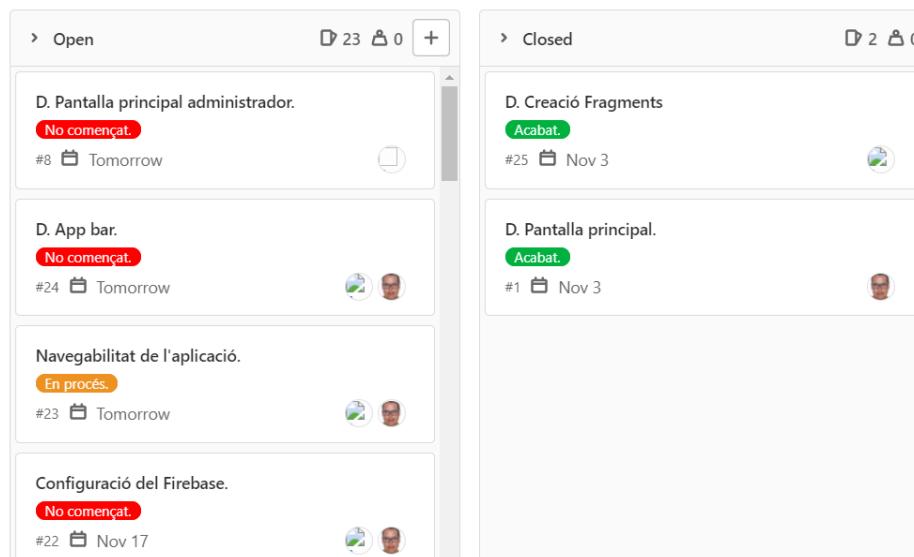
- DATA - Semana 1
 - ⌚ 28 de oct. - 3 de nov. ⏳ 1 └ 0%
- Pantalla Principal
 - ⌚ 28 de oct. - 3 de nov. ⏳ 1
 - ─ 100% ─ Nov 3
- Creació de Projecte, creació Fragments i configurar Binding
 - ⌚ 28 de oct. - 3 de nov. ⏳ 1
 - ─ 100% ─ Nov 3

Setmana 2

- DATA - Semana 2
 - ⌚ 1 └ 0%
- Pantalla Principal Administrador
 - ⌚ 4 de nov. - 10 de nov. ⏳ 1 └ 0%
 - ─ Nov 10
- Navegació - Navigation
 - ⌚ 4 de nov. - 10 de nov. ⏳ 1
 - ─ 0% ─ Nov 10
- Menu
 - ⌚ 4 de nov. - 10 de nov. ⏳ 1
 - ─ 50% ─ Nov 10

Setmana 3

- DATA - Semana 3
 - ⌚ 1 └ 0%
- Pantalla Registre
 - ⌚ 11 de nov. - 17 de nov. ⏳ 1
 - ─ 0% ─ Nov 17
- Pantalla inici sessió client-administrator
 - ⌚ 11 de nov. - 17 de nov. ⏳ 1
 - ─ 0% ─ Nov 17
- Creació - configuració Firebase
 - ⌚ 11 de nov. - 18 de nov. ⏳ 1
 - ─ 0% ─ Nov 17



Open □ 23 ⏳ 0 +

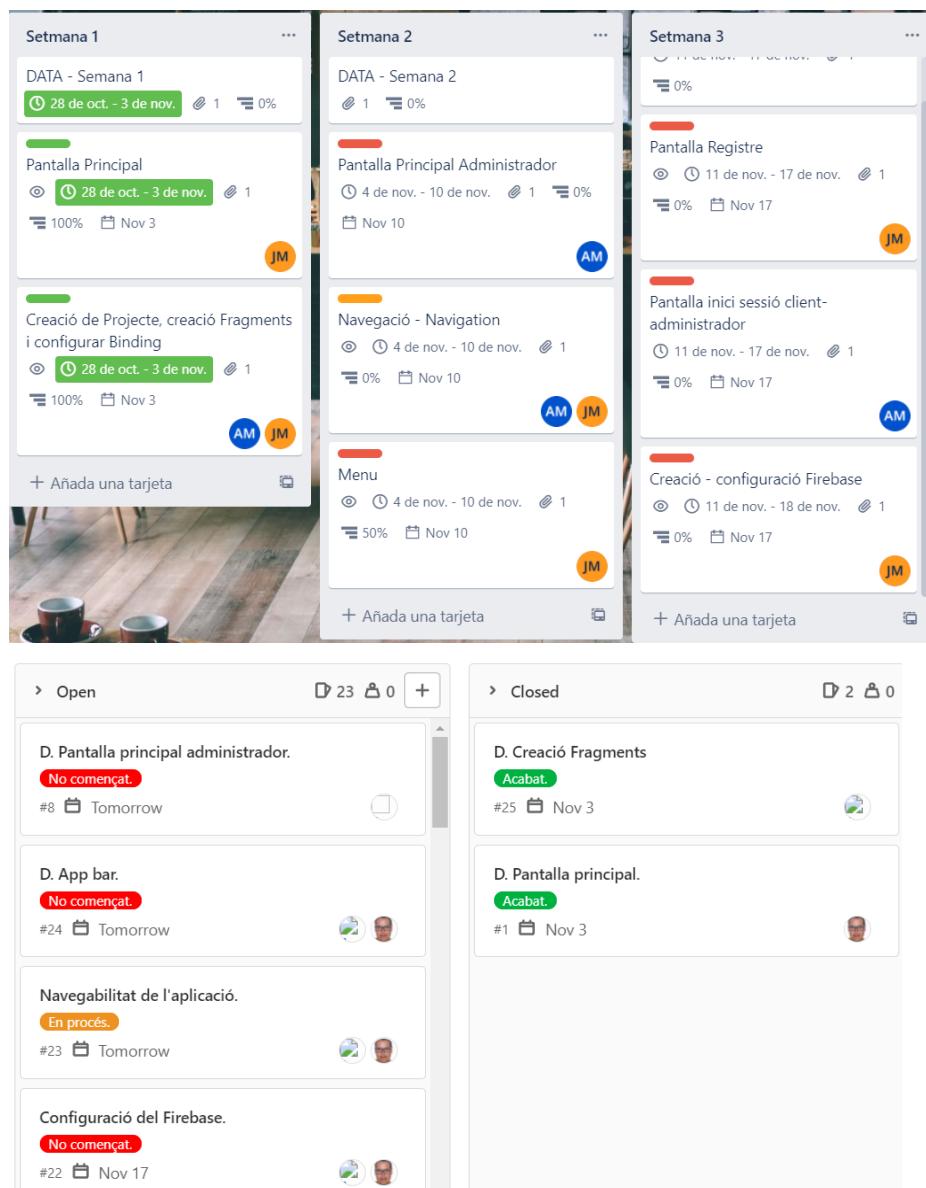
- D. Pantalla principal administrador.
No començat.
#8 ─ Tomorrow
- D. App bar.
No començat.
#24 ─ Tomorrow
- Navegabilitat de l'aplicació.
En procés.
#23 ─ Tomorrow
- Configuració del Firebase.
No començat.
#22 ─ Nov 17

Closed □ 2 ⏳ 0

- D. Creació Fragments
Acabat.
#25 ─ Nov 3
- D. Pantalla principal.
Acabat.
#1 ─ Nov 3

En la pròxima setmana farem la navegació i la pantalla principal de l'usuari Administrador, amb la previsió que tinguem temps a poder avançar alguna de les tasques de la següent setmana.

4.3.2 Setmana 2

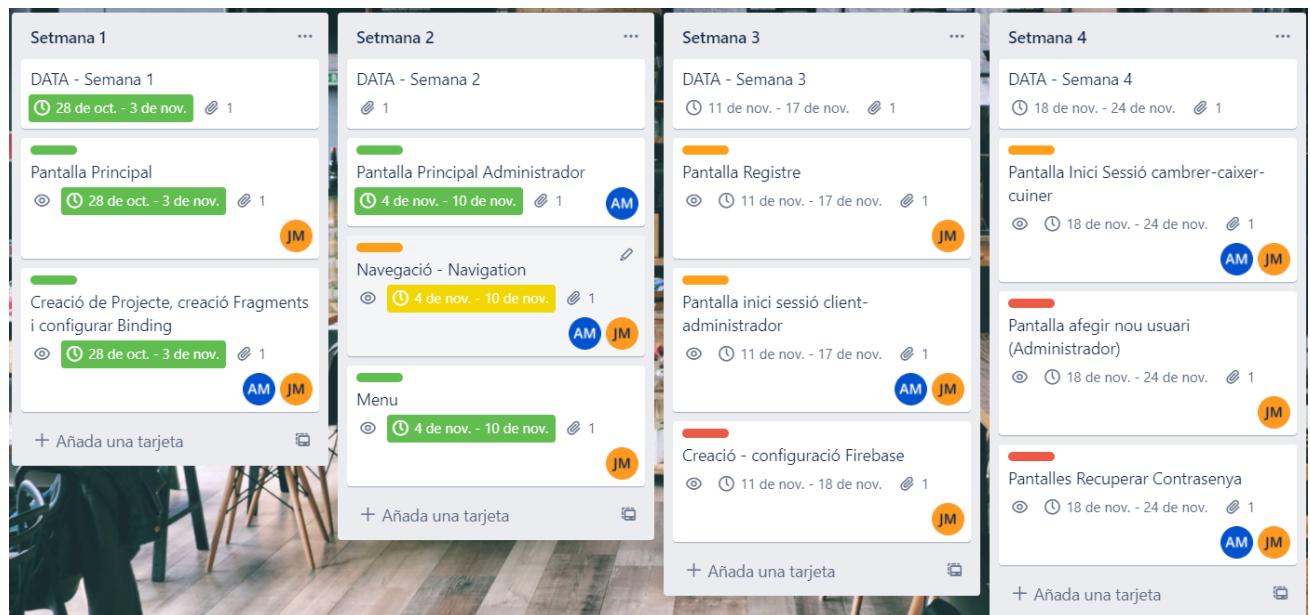


Setmana 1	Setmana 2	Setmana 3
Pantalla Principal	Pantalla Principal Administrador	Pantalla Registre
Creació de Projecte, creació Fragments i configurar Binding	Navegació - Navigation	Pantalla inici sessió client-administrador
Menu	Menu	Creació - configuració Firebase

Open	Closed
D. Pantalla principal administrador. No començat. #8 🗓 Tomorrow	D. Creació Fragments Acabat. #25 🗓 Nov 3
D. App bar. No començat. #24 🗓 Tomorrow	D. Pantalla principal. Acabat. #1 🗓 Nov 3
Navegabilitat de l'aplicació. En procés. #23 🗓 Tomorrow	
Configuració del Firebase. No començat. #22 🗓 Nov 17	

Per aquesta setmana havíem d'acabar la navegació, fer l'navbar i fer la pantalla principal de l'administrador.

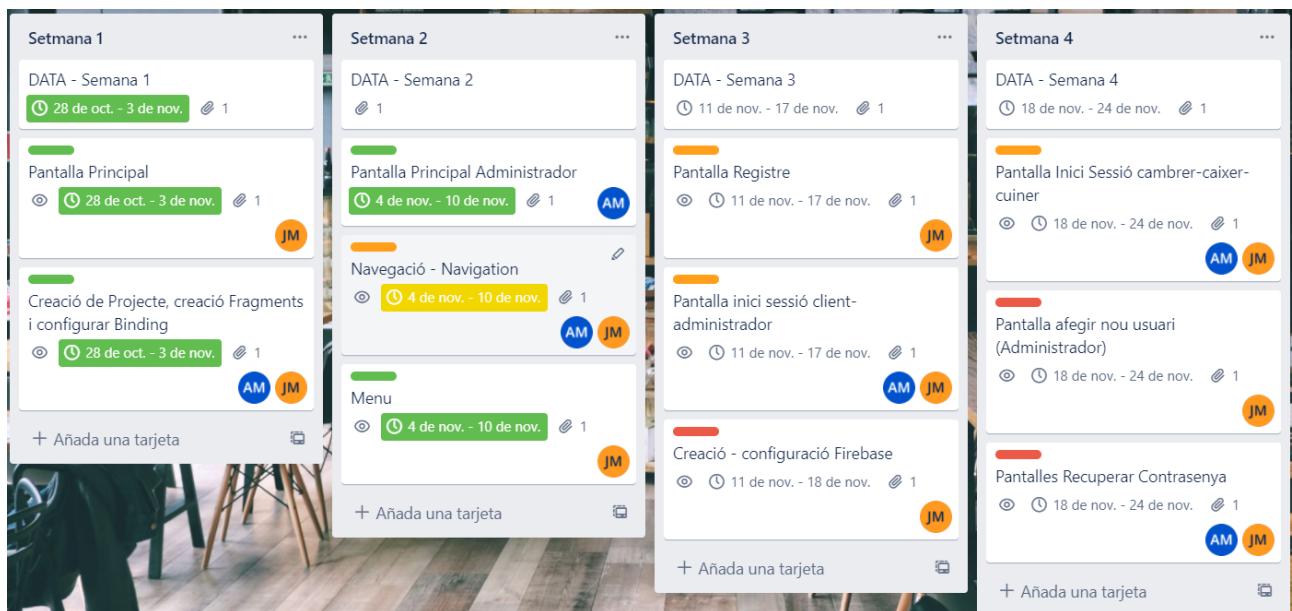
Seguim amb el problema de la navegació, tot i això, hem acabat el disseny de les pantalles inici de sessió i registre, hem fet l'navbar i la pantalla principal de l'administrador (visualment i funcionalment).



Open	Closed
Navegabilitat de l'aplicació. En procés. #23 Today	D. App bar. Acabat. #24 Today
D. Pantalla d'inici de sessió del client-administrador. En procés. #2 Nov 17	D. Pantalla principal administrador. Acabat. #8 Today
D. Pantalla registre. En procés. #4 Nov 17	D. Creació Fragments Acabat. #25 Nov 3
Configuració del Firebase. No començat. #22 Nov 17	D. Pantalla principal. Acabat. #1 Nov 3
D. Pantalla inci sessió cambrer, caixer i cuiner. En procés. #5 Nov 24	

En la pròxima setmana ens centrem a acabar la navegació, crear la base de dades Firebase i connectar-la amb Android Studio i fer les pantalles d'inici de sessió i registre.

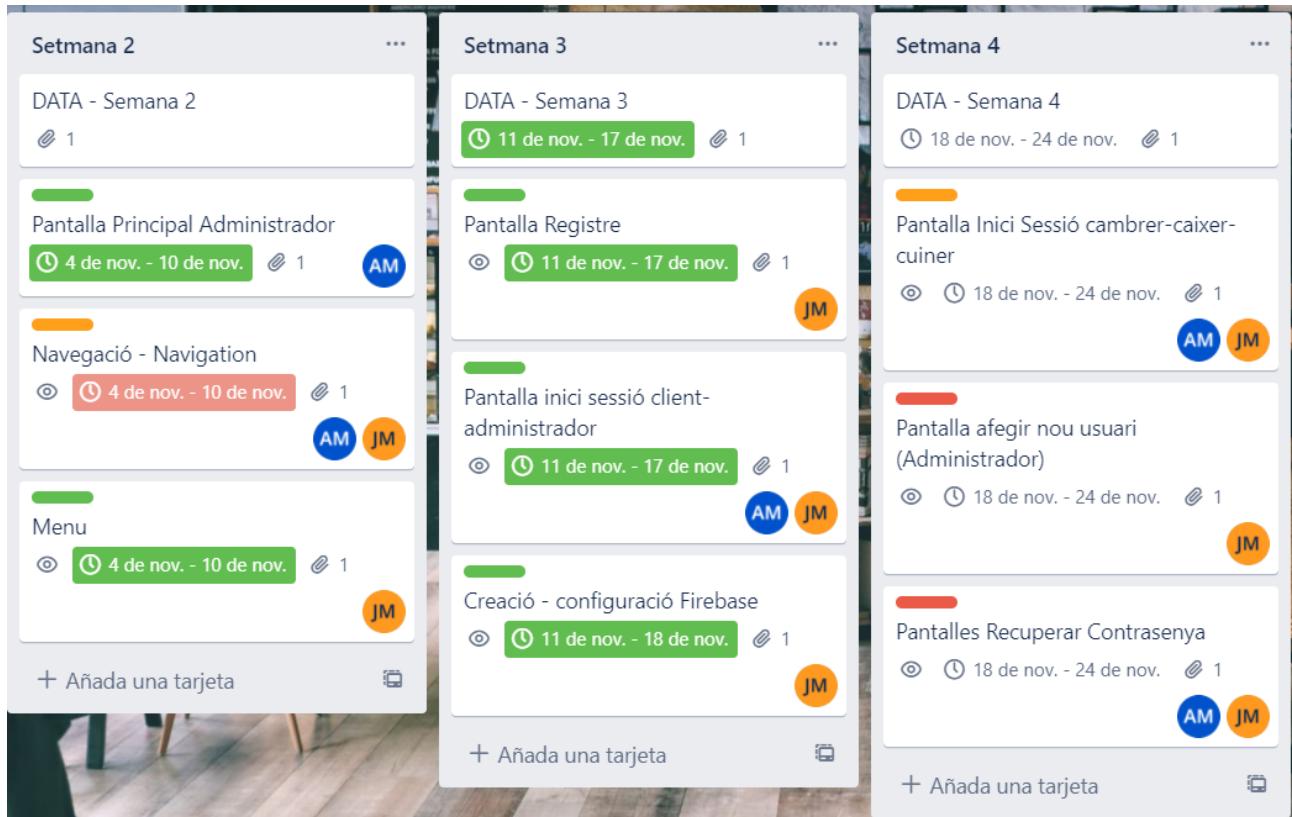
4.3.3 Setmana 3



Open	Closed
<p>Navegabilitat de l'aplicació. En procés. #23 Today</p> <p>D. Pantalla d'inici de sessió del client-administrador. En procés. #2 Nov 17</p> <p>D. Pantalla registre. En procés. #4 Nov 17</p> <p>Configuració del Firebase. No començat. #22 Nov 17</p> <p>D. Pantalla inci sessió cambrer, caixer i cuiner. En procés. #5 Nov 24</p>	<p>D. App bar. Acabat. #24 Today</p> <p>D. Pantalla principal administrador. Acabat. #8 Today</p> <p>D. Creació Fragments Acabat. #25 Nov 3</p> <p>D. Pantalla principal. Acabat. #1 Nov 3</p>

Per aquesta setmana havíem d'acabar la pantalla de registre, la pantalla d'inici de sessió del client i administrador i per últim havíem de crear i configurar la base de dades de Firebase. També havíem d'acabar la navegabilitat que ens havia quedat de la setmana passada.

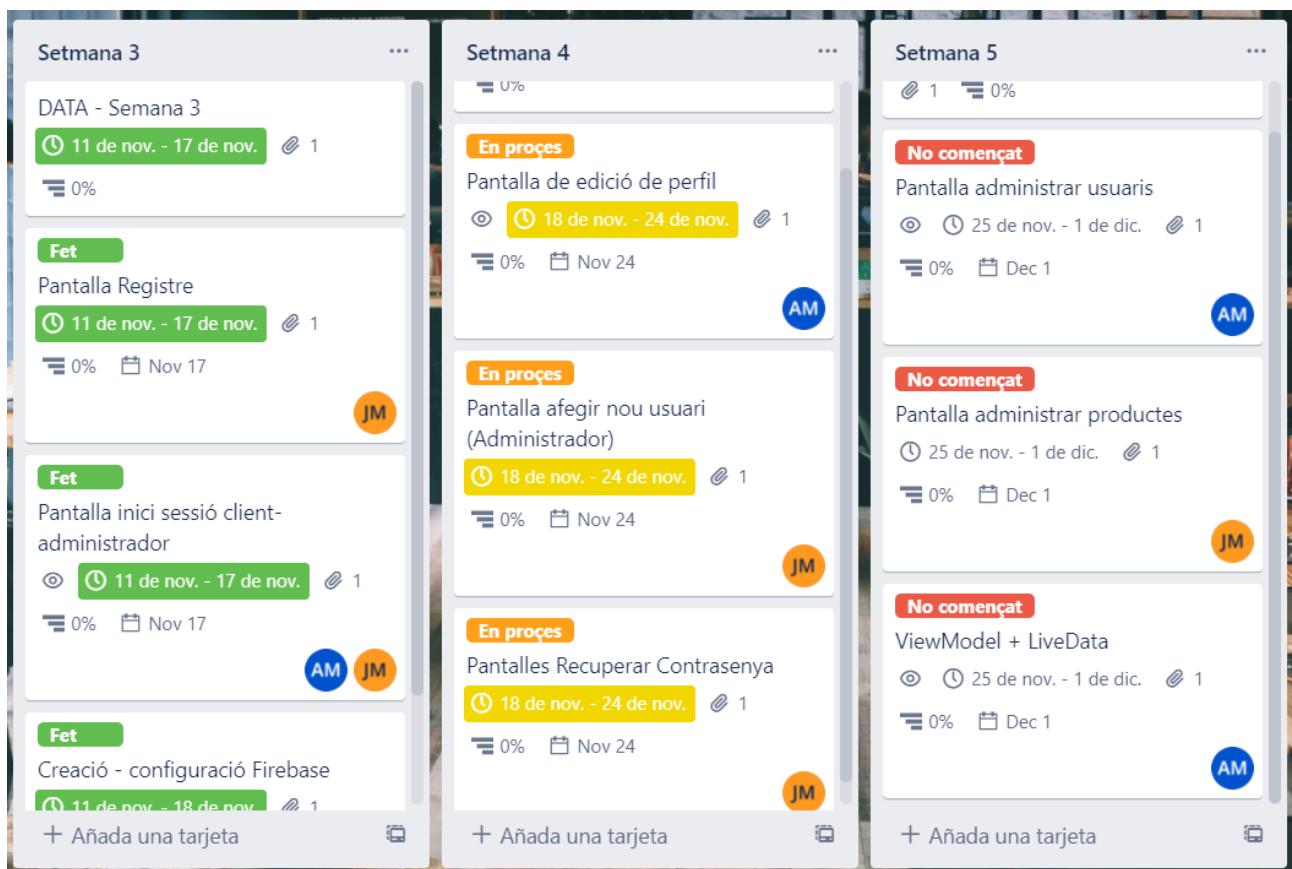
La navegació està quasi acabada, falten de polir els intents amb el pas de paràmetres per poder navegar per cada tipus d'usuari correctament.



Open	Closed
<p>Navegabilitat de l'aplicació. En procés. #23 Nov 10 </p> <p>D. Pantalla inci sessió cambrer, caixer i cuiner. En procés. #5 Nov 24 </p> <p>D. Recuperar contrassenya. No començat. #3 Nov 24 </p>	<p>D. Pantalla registre. Acabat. #4 Today </p> <p>D. Pantalla d'inici de sessió del client-administrador. Acabat. #2 Today </p> <p>Configuració del Firebase. Acabat. #22 Today </p>

En la setmana vinent ens centrarem a fer l'inici de sessió del cambrer, caixer i cuiner, la pantalla de l'administrador per afegir nous usuaris i la pantalla per recuperar contrasenyes.

4.3.4 Setmana 4



D. Pantalla d'edició del perfil.

#21 · created 4 weeks ago by Jaume Cadafalch Miró  Codificació i desenvolupament de l'aplicació.  Nov 24, 2021  En procés.

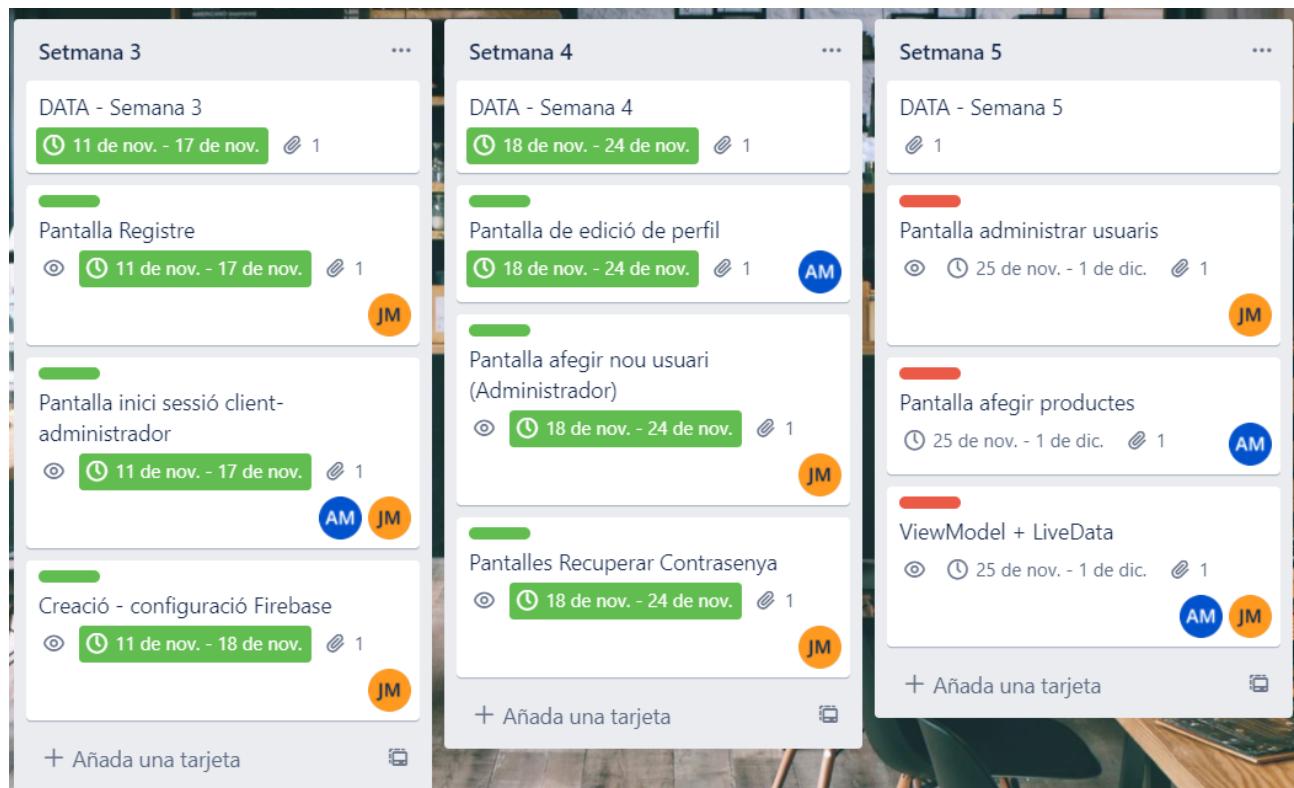
D. Pantalla afegir nou usuari.

#6 · created 4 weeks ago by Jaume Cadafalch Miró  Codificació i desenvolupament de l'aplicació.  Nov 24, 2021  En procés.

D. Recuperar contrassenya.

#3 · created 4 weeks ago by Jaume Cadafalch Miró  Codificació i desenvolupament de l'aplicació.  Nov 24, 2021  En procés.

Aquesta setmana havíem de fer la pantalla d'edició de perfil, la pantalla per afegir nou usuaris des de l'administrador i la pantalla per recuperar contrasenyes.



D. Recuperar contrassenya.

#3 · created 4 weeks ago by Jaume Cadafalch Miró  Codificació i desenvolupament de l'aplicació.  Nov 24, 2021  Acabat.

D. Pantalla afegir nou usuari.

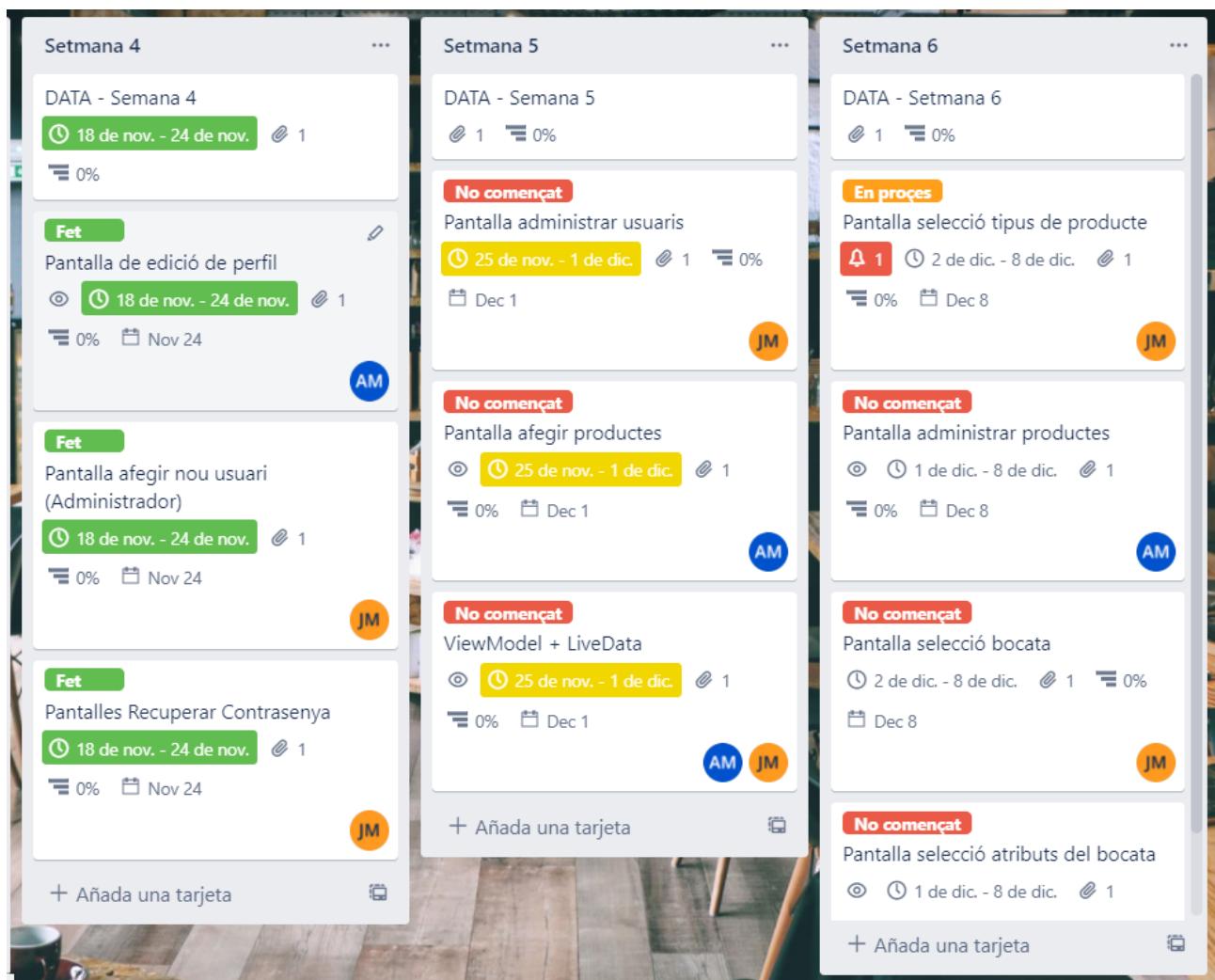
#6 · created 4 weeks ago by Jaume Cadafalch Miró  Codificació i desenvolupament de l'aplicació.  Nov 24, 2021  Acabat.

D. Pantalla d'edició del perfil.

#21 · created 4 weeks ago by Jaume Cadafalch Miró  Codificació i desenvolupament de l'aplicació.  Nov 24, 2021  Acabat.

En la setmana vinent, la nostra prioritat serà el ViewModel i el LiveData, a més farem la pantalla d'administrar usuaris i la d'afegir productes.

4.3.5 Setmana 5



Setmana 4	Setmana 5	Setmana 6
DATA - Semana 4 ⌚ 18 de nov. - 24 de nov. 0% Fet Pantalla de edició de perfil ⌚ 18 de nov. - 24 de nov. 0% Nov 24 JM	DATA - Semana 5 ⌚ 1 0% No començat Pantalla administrar usuaris ⌚ 25 de nov. - 1 de dic. 0% Dec 1 JM	DATA - Setmana 6 ⌚ 1 0% En proces Pantalla selecció tipus de producte ⌚ 1 0% Dec 8 JM
Fet Pantalla afegir nou usuari (Administrador) ⌚ 18 de nov. - 24 de nov. 0% Nov 24 JM	No començat Pantalla afegir productes ⌚ 25 de nov. - 1 de dic. 0% Dec 1 AM	No començat Pantalla administrar productes ⌚ 1 0% Dec 8 AM
Fet Pantalles Recuperar Contrasenya ⌚ 18 de nov. - 24 de nov. 0% Nov 24 JM	No començat ViewModel + LiveData ⌚ 25 de nov. - 1 de dic. 0% Dec 1 JM	No començat Pantalla selecció bocata ⌚ 2 de dic. - 8 de dic. 0% Dec 8 JM
+ Añada una tarjeta	+ Añada una tarjeta	+ Añada una tarjeta

Disseny de la base de dades

#27 · created 10 minutes ago by Adrian Ramirez Masagú ⌚ Codificació i desenvolupament de l'aplicació. Dec 1, 2021 No començat.

updated just now

ViewModel + LiveData

#26 · created 11 minutes ago by Adrian Ramirez Masagú ⌚ Codificació i desenvolupament de l'aplicació. Dec 1, 2021 No començat.

updated just now

D. Pantalla afegir productes.

#10 · created 1 month ago by Jaume Cadafalch Miró ⌚ Codificació i desenvolupament de l'aplicació. Dec 1, 2021 No començat.

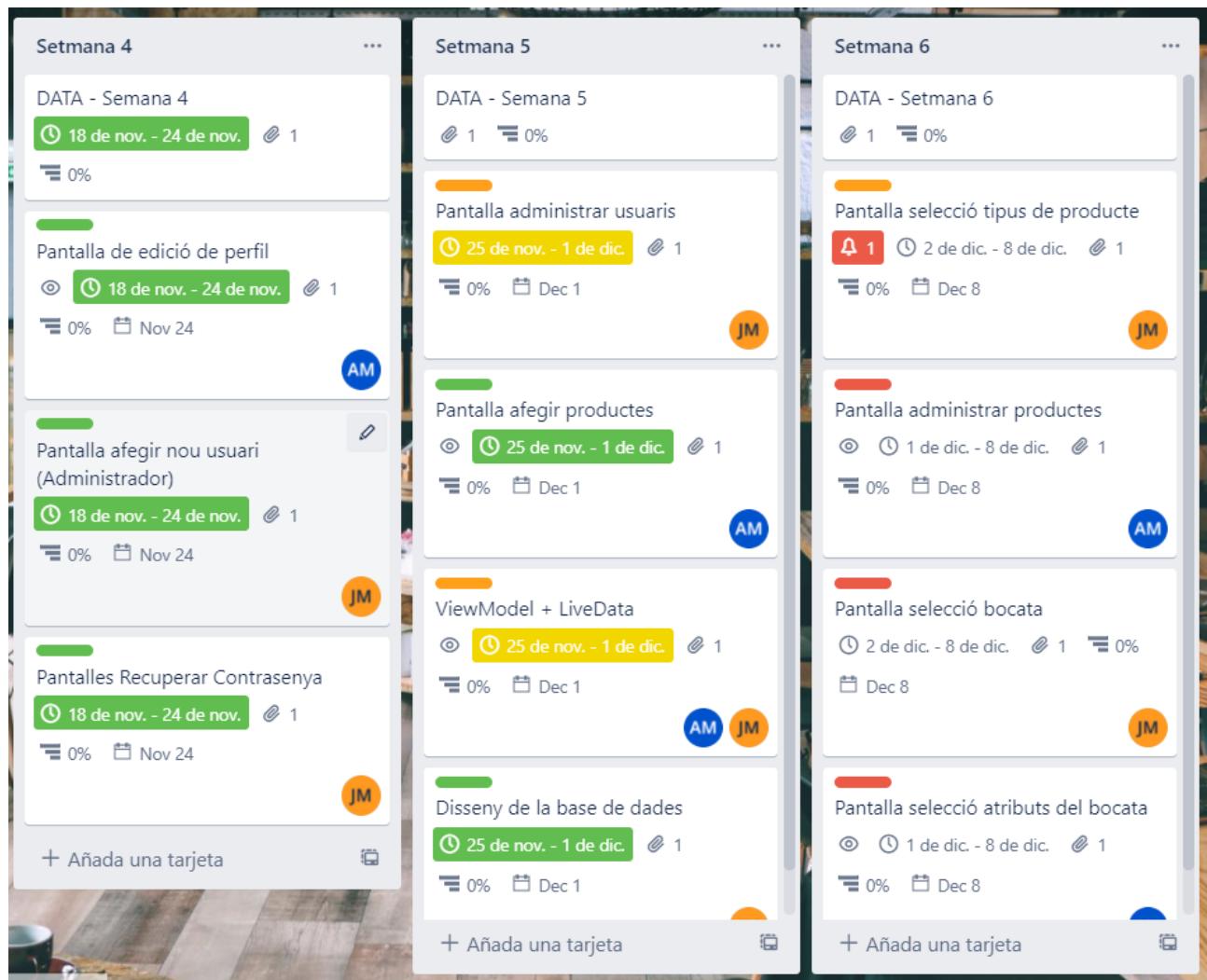
updated just now

D. Pantalla administrar productes.

#9 · created 1 month ago by Jaume Cadafalch Miró ⌚ Codificació i desenvolupament de l'aplicació. Dec 1, 2021 No començat.

updated 1 month ago

Aquesta setmana havíem de fer el ViewModel + LiveData, la pantalla d'afegir producte i la pantalla d'administrar usuaris.



Disseny de la base de dades

#27 · created just now by Adrian Ramirez Masagú (i) Codificació i desenvolupament de l'aplicació. (d) Dec 1, 2021 (a) Acabat.

ViewModel + LiveData

#26 · created 1 minute ago by Adrian Ramirez Masagú (i) Codificació i desenvolupament de l'aplicació. (d) Dec 1, 2021 (e) En procés.

D. Pantalla afegir productes.

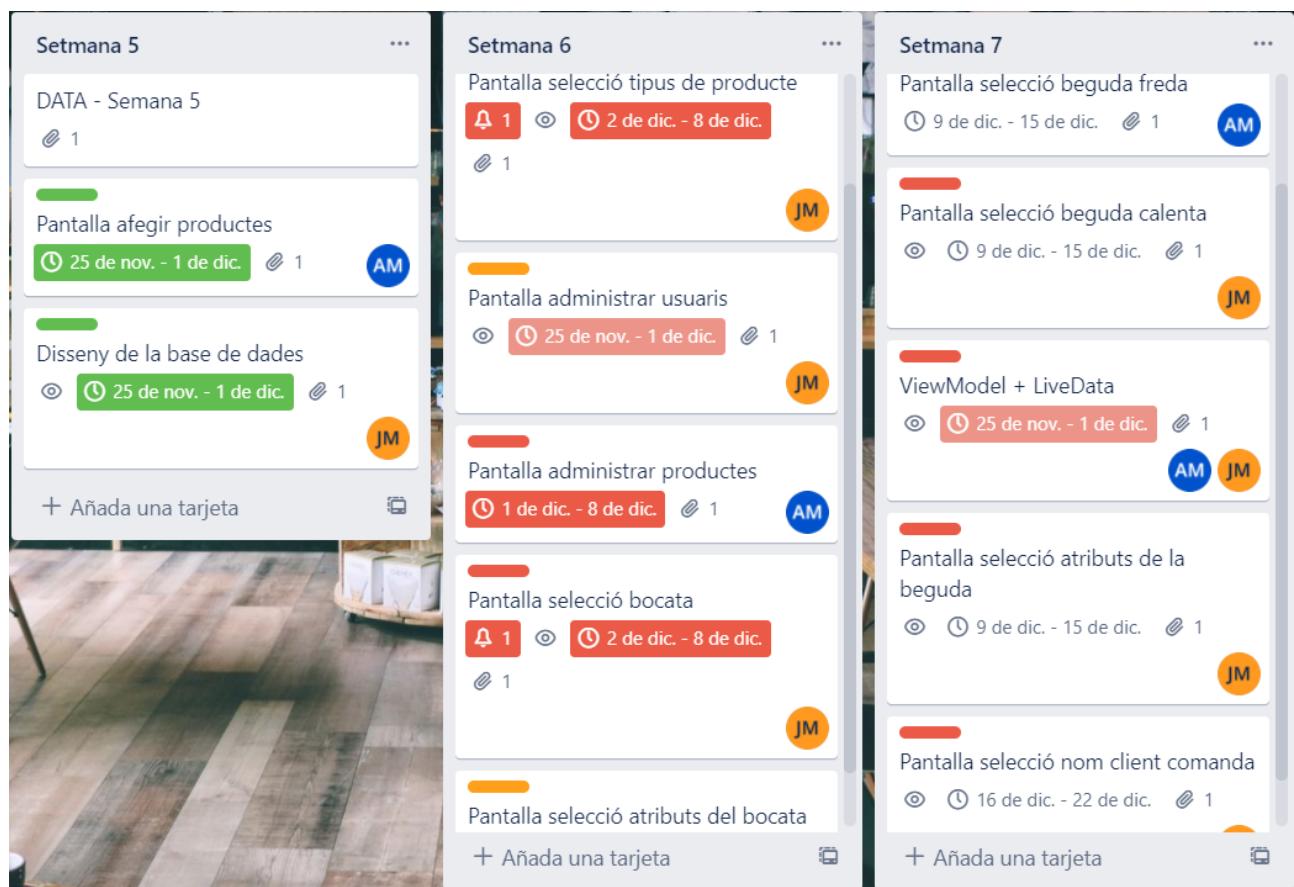
#10 · created 1 month ago by Jaume Cadafalch Miró (i) Codificació i desenvolupament de l'aplicació. (d) Dec 1, 2021 (a) Acabat.

D. Pantalla eliminar-administrar usuaris.

#7 · created 1 month ago by Jaume Cadafalch Miró (i) Codificació i desenvolupament de l'aplicació. (d) Dec 1, 2021 (e) En procés.

Al final no hem fet el ViewModel + LiveData perquè hem tingut problemes per implementar-ho i ens hem dedicat a fer les pantalles. Per un altre costat, hem fet el disseny de la base de dades. Per la setmana que ve la nostra prioritat és acabar el ViewModel + LiveData i començar a fer les pantalles que requereixen el Cloud Storage.

4.3.6 Setmana 6



D. Pantalla administrar productes.

#9 · created 1 month ago by Jaume Cadafalch Miró Codificació i desenvolupament de l'aplicació. Dec 1, 2021 No començat.

0
updated just now

D. Pantalla eliminar-administrar usuaris.

#7 · created 1 month ago by Jaume Cadafalch Miró Codificació i desenvolupament de l'aplicació. Dec 1, 2021 En procés.

0
updated 1 week ago

D. Pantall selecció de bocata.

#13 · created 1 month ago by Jaume Cadafalch Miró Codificació i desenvolupament de l'aplicació. Dec 8, 2021 No començat.

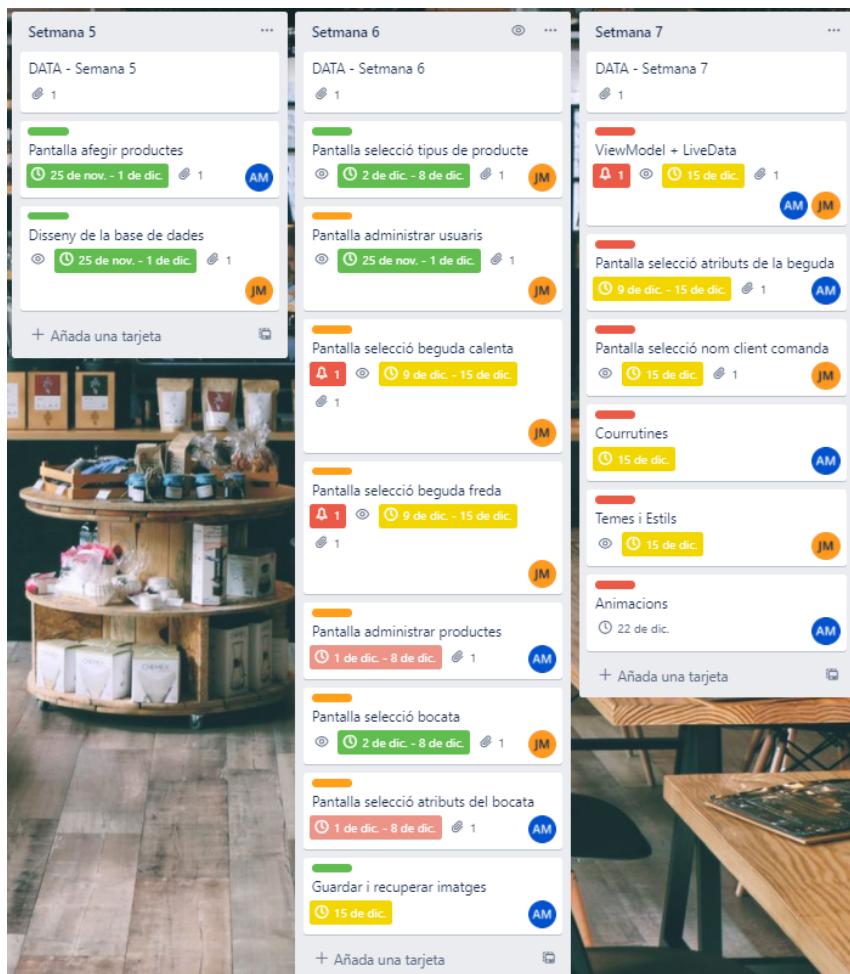
0
updated 1 month ago

D. Pantalla selecció tipus de producte.

#12 · created 1 month ago by Jaume Cadafalch Miró Codificació i desenvolupament de l'aplicació. Dec 8, 2021 En procés.

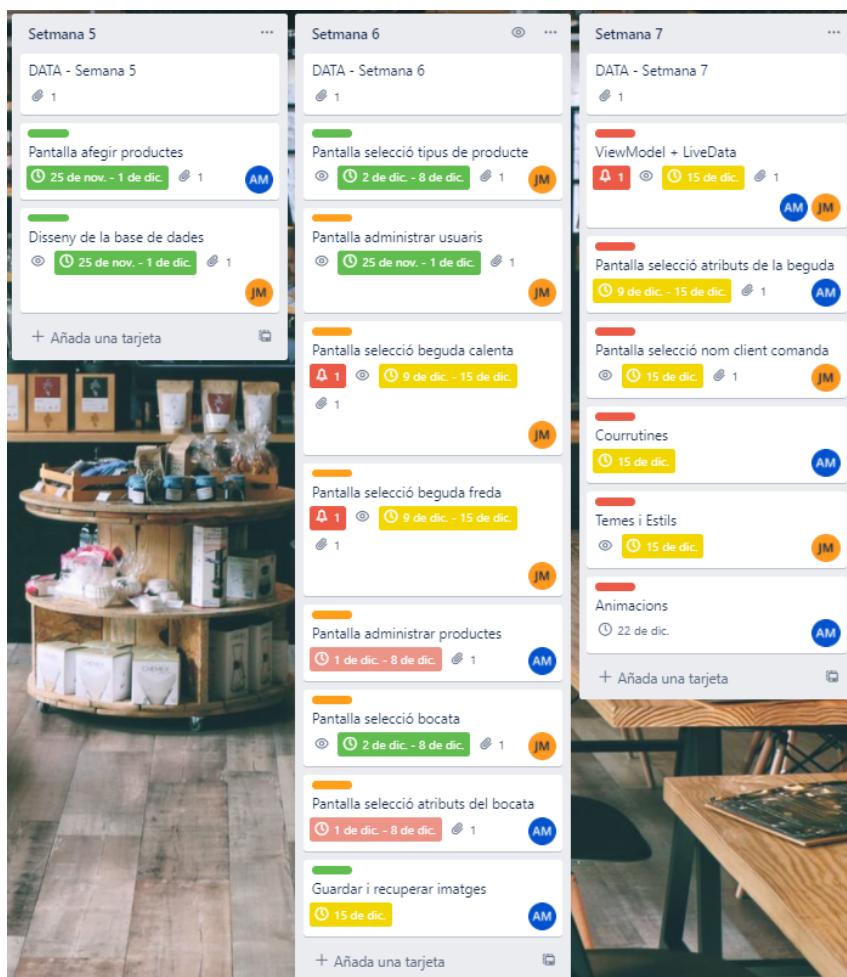
0
updated just now

En aquesta setmana havíem de corregir els errors de la setmana anterior, acabar les pantalles inacabades de la setmana anterior i fer les pantalles: pantalla administrar productes, pantalla selecció tipus de producte, pantalla selecció de bocata i pantalla d'atributs del bocata.

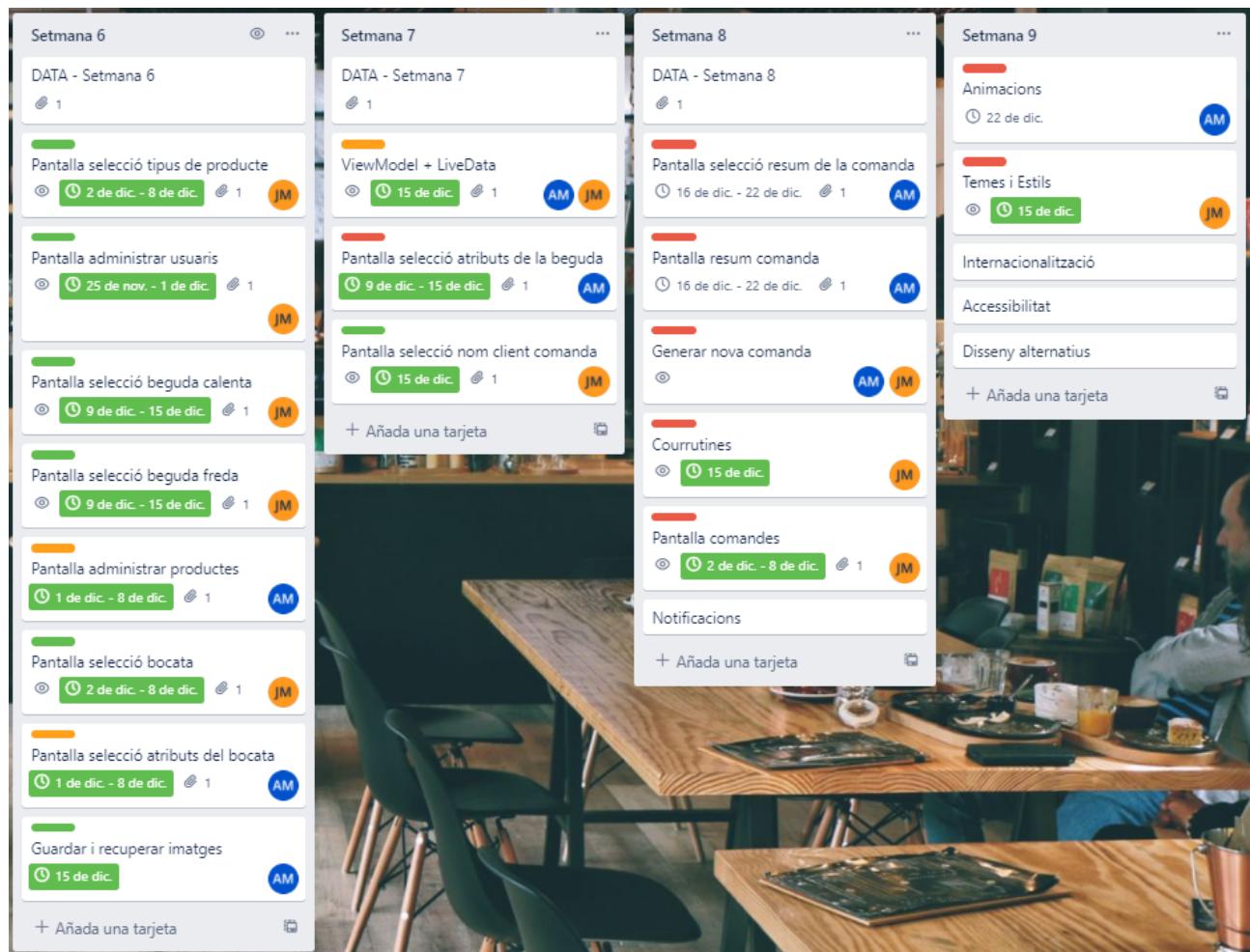


Acabem aquesta setmana amb les pantalles diferents pantalles de selecció de productes quasi acabades a falta d'afegir les imatges corresponents, pantalla administrar productes i administrar usuaris a mig fer i també les pantalles atributs bocates i begudes també a mig fer.

4.3.7 Setmana 7



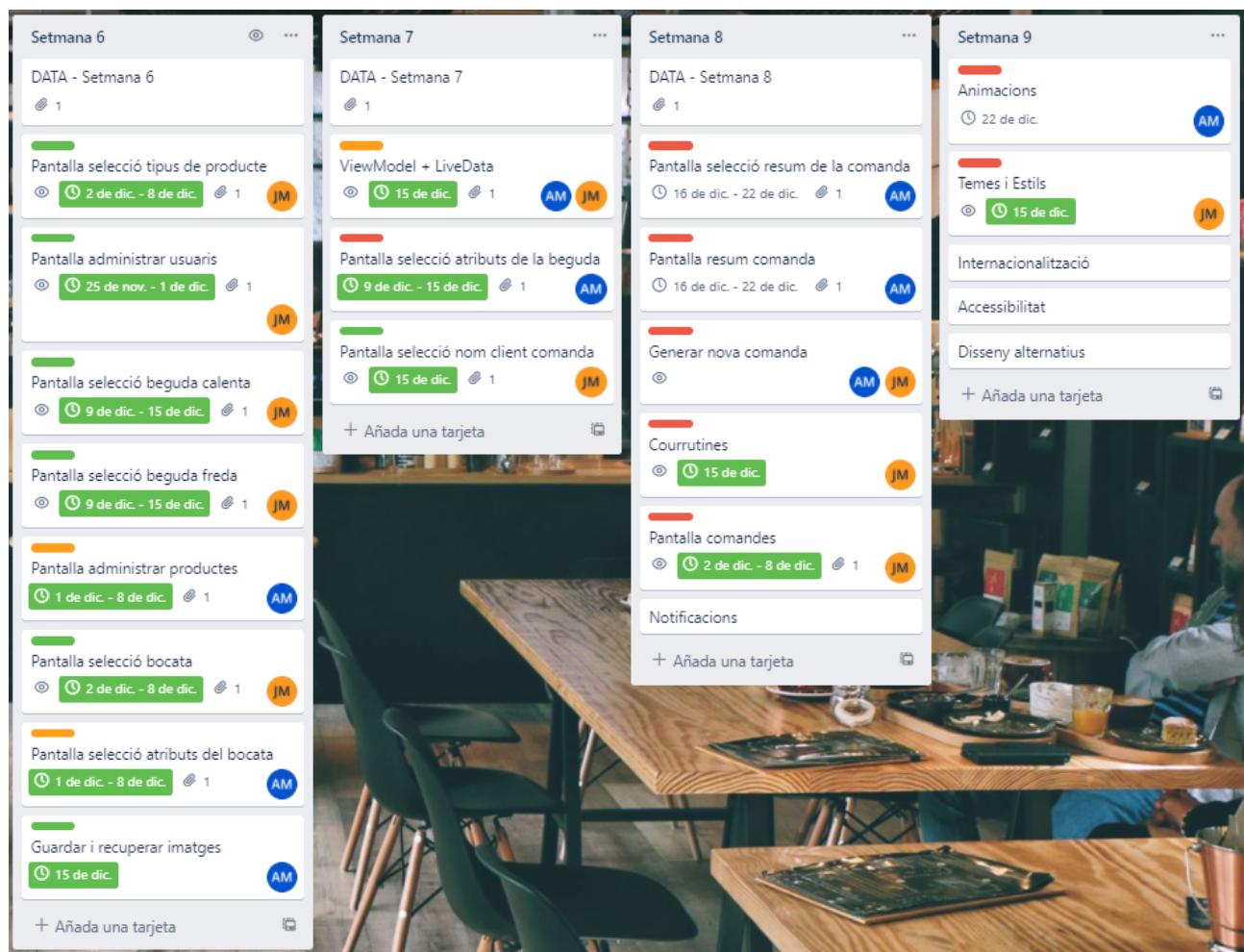
Aquesta setmana havíem d'acabar les coses de la setmana passada, i a més, fer la pantalla atribut beguda, pantalla "selecció nom client comanda", corutines, "temes i estils" i animacions.



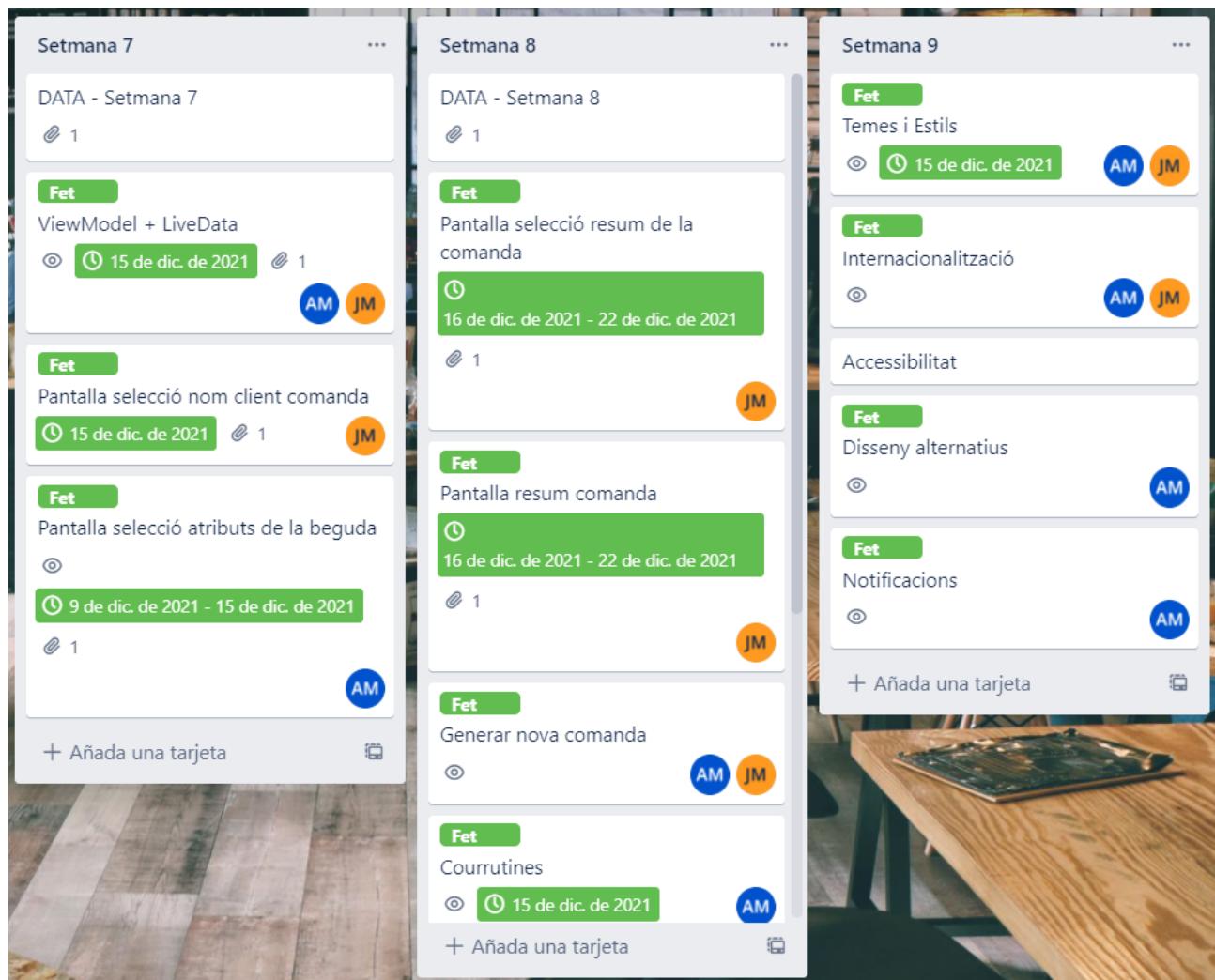
Aquesta setmana hem acabat gran part de les tasques pendents, hem fet la pantalla "Selecció nom client comanda" i hem començat amb el ViewModel. No hi ha hagut temps a fer la pantalla "selecció atributs de la beguda".

Per la setmana que ve ens centrarem a acabar el MVVM, acabar de programar la creació d'una comanda, farem les pantalles de "selecció resum comanda", "resum comanda" i pantalla comandes (cuiner) i si tenim temps, farem coroutines i notificacions.

4.3.8 Setmana 8 i setmana 9



Al final de la setmana 8 havíem d'acabar el MVVM, acabar de programar la creació d'una comanda, farem les pantalles de "selecció resum comanda", "resum comanda" i pantalla comandes (cuiner) i si tenim temps, farem corutines i notificacions.



Setmana 7

- Fet** ViewModel + LiveData
 - ⌚ 15 de dic. de 2021
 - AM JM
- Fet** Pantalla selecció nom client comanda
 - ⌚ 15 de dic. de 2021
 - JM
- Fet** Pantalla selecció atributs de la beguda
 - ⌚ 9 de dic. de 2021 - 15 de dic. de 2021
 - AM
- + Añada una tarjeta

Setmana 8

- Fet** Pantalla selecció resum de la comanda
 - ⌚ 16 de dic. de 2021 - 22 de dic. de 2021
 - JM
- Fet** Pantalla resum comanda
 - ⌚ 16 de dic. de 2021 - 22 de dic. de 2021
 - JM
- Fet** Generar nova comanda
 - ⌚ 15 de dic. de 2021
 - AM JM
- Fet** Courrutines
 - ⌚ 15 de dic. de 2021
 - AM
- + Añada una tarjeta

Setmana 9

- Fet** Temes i Estils
 - ⌚ 15 de dic. de 2021
 - AM JM
- Fet** Internacionalització
 - ⌚ 15 de dic. de 2021
 - AM JM
- Accessibilitat
- Fet** Disseny alternatius
 - ⌚ 15 de dic. de 2021
 - AM
- Fet** Notificacions
 - ⌚ 15 de dic. de 2021
 - AM
- + Añada una tarjeta

Al final de la setmana 9 hem acabat tot.

5. Desenvolupament de l'app

5.1 Control de versions

<https://git.copernic.cat/cadafalch.miro.jaume/lacantinadeprodis.git>

5.1.2 Git

Git és un sistema de control de versions distribuït dissenyat per controlar des de petits projectes fins a projectes grans amb molta rapidesa i eficiència.

Git és fàcil d'aprendre i és de mida molt reduïda amb una eficiència molt ràpida.

Característiques:

- És molt potent.
- No depèn d'un repositori central.
- És de software lliure.
- Es pot tenir un historial complet de versions.
- Ens podem moure a través de les versions del codi d'una manera molt àgil.
- És molt ràpida.
- Té un sistema de branques que ho fa realment potent.
- Les branques estan destinades a provocar projectes divergents d'un projecte principal per fer experiments o provar noves funcionalitats.
- Les branques poden tindre una línia de progrés diferent de la branca principal on és el core del nostre desenvolupament. En algun moment podem arribar a provar algunes d'aquestes millors o fer canvis en el nostre codi i fusionar aquestes branques amb el nostre projecte principal.

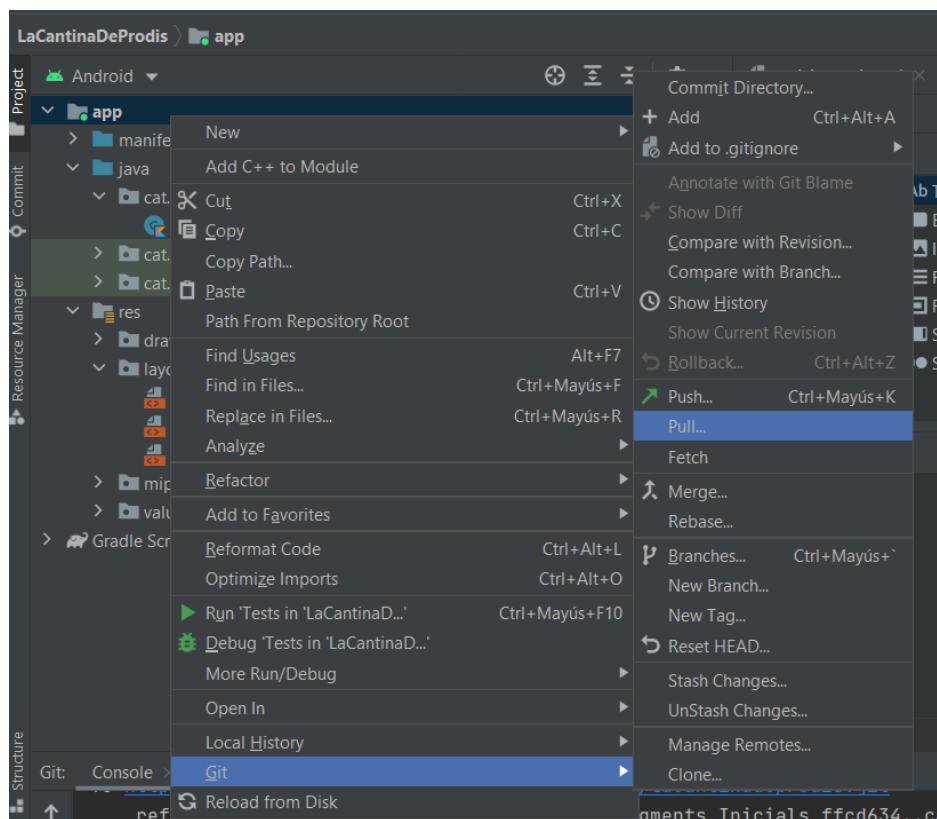
5.1.2.1 Funcionalitats

5.1.2.1.1 Pull

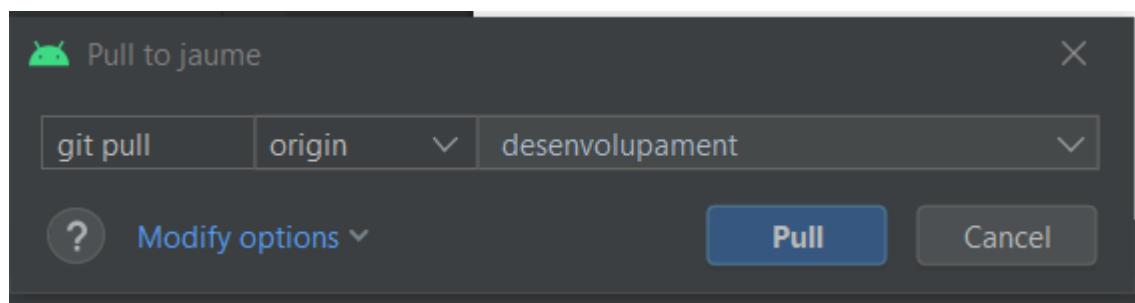
Pull: Extreu i descarrega el contingut del repositori remot i actualitza el repositori local.

Sempre que ens disposem a treballar en projectes en què estan ubicats remotament, el primer que hem de fer és un *pull* per assegurar-nos que el que tenim nosaltres és el que està en remot.

Per fer-ho, és tan fàcil com fer clic dret sobre app o projecte, seleccionar git i finalment seleccionar pull:



En seleccionar-ho serà tan fàcil com indicar de quina branca farem el pull i clicar sobre pull.

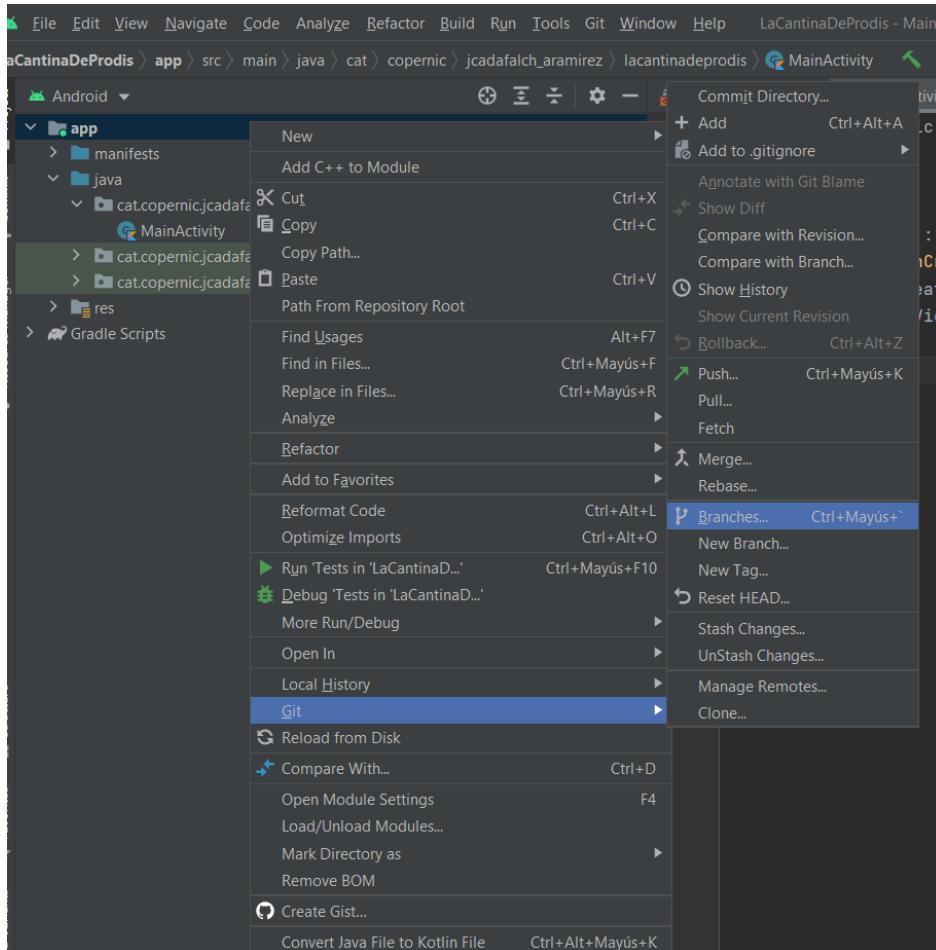


5.1.2.1.2 Branches

Branches: Una branca representa una línia independent de desenvolupament [\[5.1.2.2 Branques\]](#)

Un cop fet això, el més òptim és crear-nos una branca i treballar des de ella, per tal no interferir amb els altres companys de projecte.

Per fer-ho només hem de fer r clic dret sobre app o projecte, seleccionar git i seleccionar *branches*.



A continuació clicar sobre *New Branch*, introduir el nom de branca i fer clic en *create per crear-la*

The screenshot shows a Git interface with the following elements:

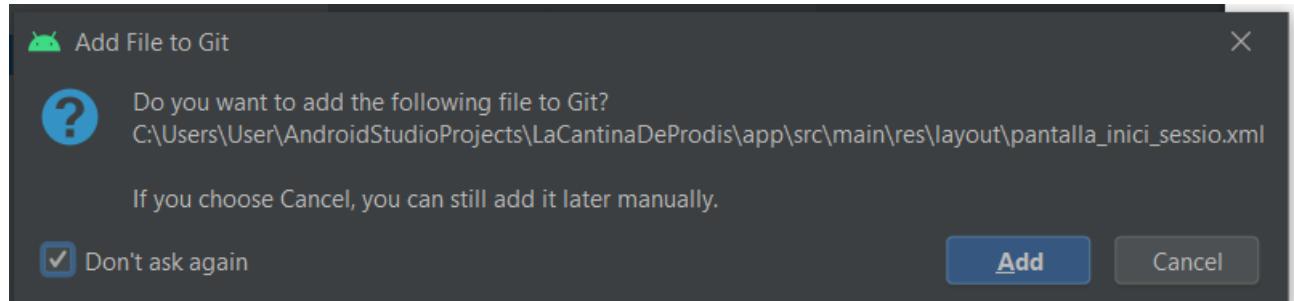
- Top Bar:** "Git Branches in LaCantinaDeProdis" with icons for refresh and settings.
- Search Bar:** A magnifying glass icon.
- New Branch Button:** "+ New Branch".
- Checkout Tag or Revision...** button.
- Local Branches:** "master" (key icon) and "origin/master" (arrow icon).
- Remote Branches:** "origin/master" (star icon).
- Create New Branch Dialog:**
 - Title: "Create New Branch" with a close button "X".
 - Text: "New branch name:"
 - Input Field: "Fragments_Inicial" (highlighted with a blue border).
 - Checkboxes:
 - Checkout branch
 - Overwrite existing branch
 - Buttons: "Create" (blue background) and "Cancel".
- Active Branches List:**
 - desenvolupament**
 - d9901e1e · Binding funcional + graddle 0 errors · 1 day ago
 - adrian**
 - 83d6d56b · binding · 2 days ago
 - jaume**
 - a2acb3aa · add pantalla_registre.xml · 3 days ago
 - master** (highlighted)
 - a2acb3aa · add pantalla_registre.xml · 3 days ago

5.1.2.1.3 Add

Add: Afegeix un fitxer al git.

Un cop hem creat la branca i comencem a treballar sobre ella el més comú és crear un fixer del que necessitem.

La primera vegada que ho fem (el primer fitxer que creem) ens apareix el següent missatge. Seleccionem “Don’t ask again” i fem apretem “Add”. Amb això estem indicant que cada fitxer que creem es faci un Add.

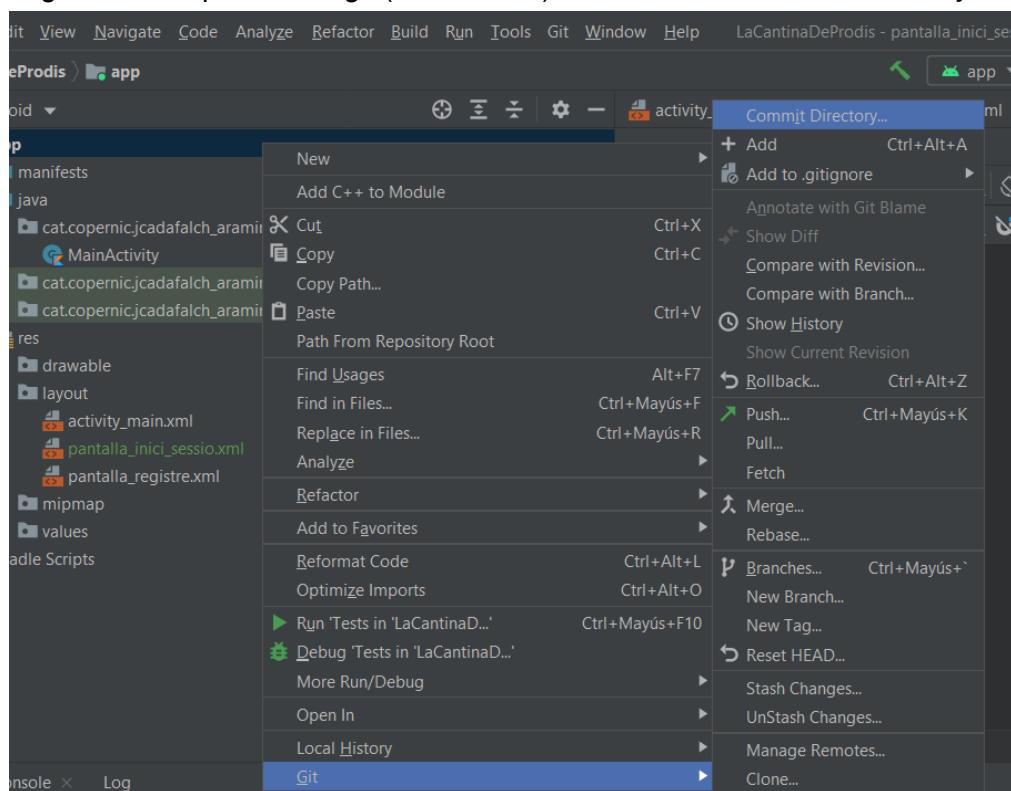


5.1.2.1.4 Commit

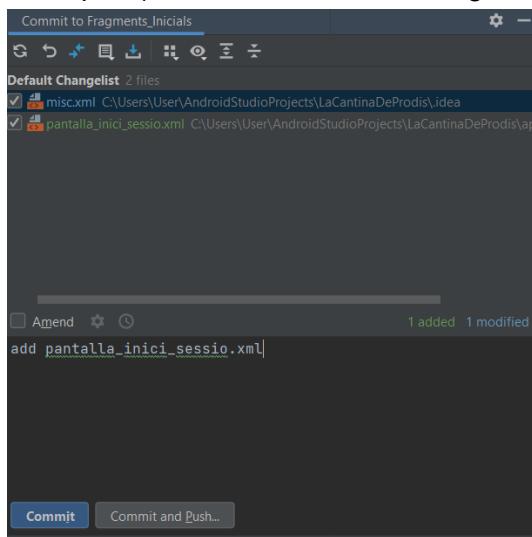
Commit: Captura una snapshot de l'estat en el qual està el projecte.

Un cop hem acabat per aquell dia, el més recomanable és sincronitzar el treball fet amb el repositori remot, per fer-ho fem un commit del directory arrel, és a dir de *Project* o *app*

Seguim la ruta per anar a git (vista abans) i seleccionem *Commit Directory*:



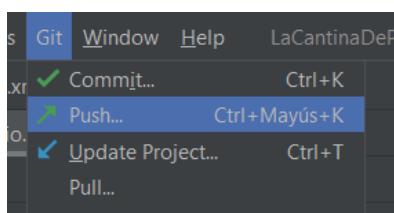
A continuació seleccionem els fitxers que volem incluir en el commit, afegim un missatge descriptiu (en el nostre cas hem afegir una nova pantalla) i pixem a commit.



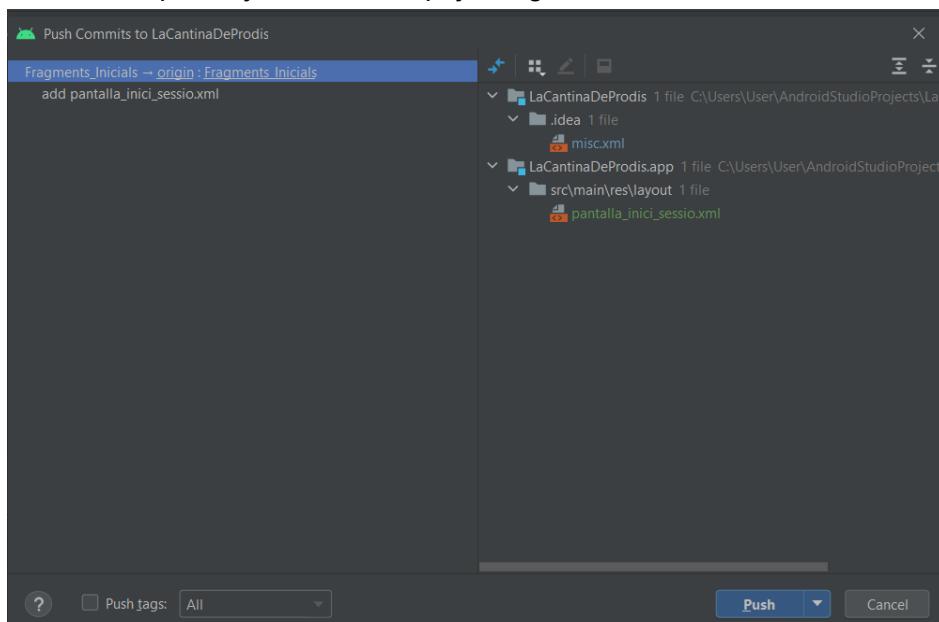
5.1.2.1.5 Push

Push: Puja els commits des de la teva branca local al teu repositori remot.

Finalment per penjar-ho al git definitivament és tan senzill com fer un push.



Fem click a push i ja ho haurem pujat al git



5.1.2.1.6 Merge

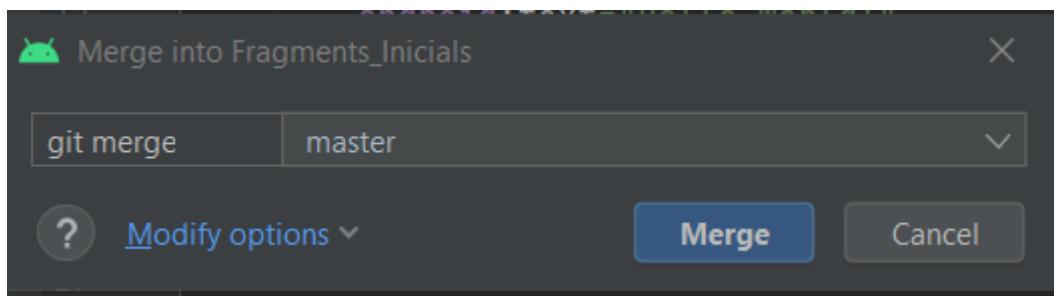
Merge: Agafa les línies independents de desenvolupament de les branques i les integra en una sola.

Per acabar, un cop hem acabat de treballar amb aquella branca que havíem creat al principi i enviar el contingut a la principal, només haurem de fer un merge.

Anant a git i seleccionant Merge:



En el menú que s'acaba d'obrir seleccionar la branca en la que volem fer un merge amb la branca master



5.1.2.2 Branques

Una branca representa una línia independent de desenvolupament. Les branques serveixen com una abstracció dels processos de canvi, preparació i confirmació. Pots concebre-les com una manera de sol·licitar un nou directori de treball, un nou entorn d'assaig o un nou historial de projecte. Les noves confirmacions es registren en l'historial de la branca actual, la qual cosa crea una bifurcació en l'historial del projecte.

La comanda `git branch` et permet crear, enumerar i eliminar branques, així com canviar el seu nom. No et permet canviar entre branques o tornar a unir un historial bifurcat. Per aquest motiu, `git branch` està estretament integrat amb els comandos `git checkout` i `git merge`.

5.1.2.2.1 Opcions comuns

<code>git branch</code>	Enumera totes les branques del teu repositori. Similar a " <code>git branch --list</code> "
<code>git branch <branch></code>	Crea una nova branca anomenada <code><branch></code> . Aquesta comanda no extreu la nova branca.
<code>git branch -d <branch></code>	Elimina la branca especificada. Aquesta és una operació segura, ja que Git evita que eliminis la branca si té canvis que encara no s'han fusionat.
<code>git branch -D <branch></code>	Força l'eliminació de la branca especificada, fins i tot si té canvis sense fusionar. Aquesta comanda la pots usar si vols eliminar de manera permanent totes les confirmacions associades amb una línia concreta de desenvolupament.
<code>git branch -m <branch></code>	Canvia el nom de la branca actual a <code><branch></code>
<code>git branch -a</code>	Enumera totes les branques remotes

5.1.2.2.2 Creació Branques

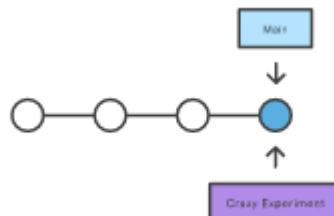
És important comprendre que les branques són només capdavanters a les confirmacions. Quan creguis una branca, tot el que Git ha de fer és crear un nou punter, no modifica el repositori de cap altra forma. Si comences amb un repositori que té aquest aspecte:



I, a continuació, crees una branca amb la següent comanda:

```
git branch crazy-experiment
```

L'historial del repositori no es modificarà. Tot el que necessites és un nou punter de la confirmació actual:



Tingues en compte que aquest comanda només crea la nova branca. Per a començar a afegir confirmacions, necessites seleccionar-la amb la comanda “*git checkout*” i, a continuació, utilitzar els commandos estàndard “*git add*” i “*git commit*”.

5.1.2.2.3 Creació Branques Remotes

La comanda “*git branch*” també funciona amb branques remotes. Per a treballar en branques remotes, primer cal configurar un repositori remot i afegir-lo a la configuració del repositori local.

```
$ git remote add new-remote-repo https://bitbucket.com/user/repo.git
# Add remote repo to local repo config
$ git push <new-remote-repo> crazy-experiment~
# pushes the crazy-experiment branch to new-remote-repo
```

Aquesta comanda enviarà una còpia de la branca local *crazy-experiment* al repositori remot *<remote>*.

5.1.2.2.4 Eliminació Branques

Una vegada que hagis acabat de treballar en una branca i l'hagis fusionat amb el codi base principal, pots eliminar la branca sense perdre cap història:

```
git branch crazy-experiment
```

No obstant això, si la branca no s'ha fusionat, la comanda anterior mostrarà un missatge d'error:

```
error: The branch 'crazy-experiment' is not fully merged. If you are sure you want to delete it, run 'git branch -D crazy-experiment'.
```

Això et protegeix davant la pèrdua d'accés a una línia de desenvolupament completa. Si realment vols eliminar la branca (per exemple, si es tracta d'un experiment fallit), pots usar l'indicador -D (en majúscula):

La següent comanda elimina la branca independentment del seu estat i sense avisos previs, així que usa'l amb cura.

```
git branch -D crazy-experiment
```

Les comandes anteriors eliminaran una còpia local de la branca. La branca continuarà existint en el repositori remot. Per a eliminar una branca remota, s'ha d'executar una de les següents comandes.

```
git push origin --delete crazy-experiment
```

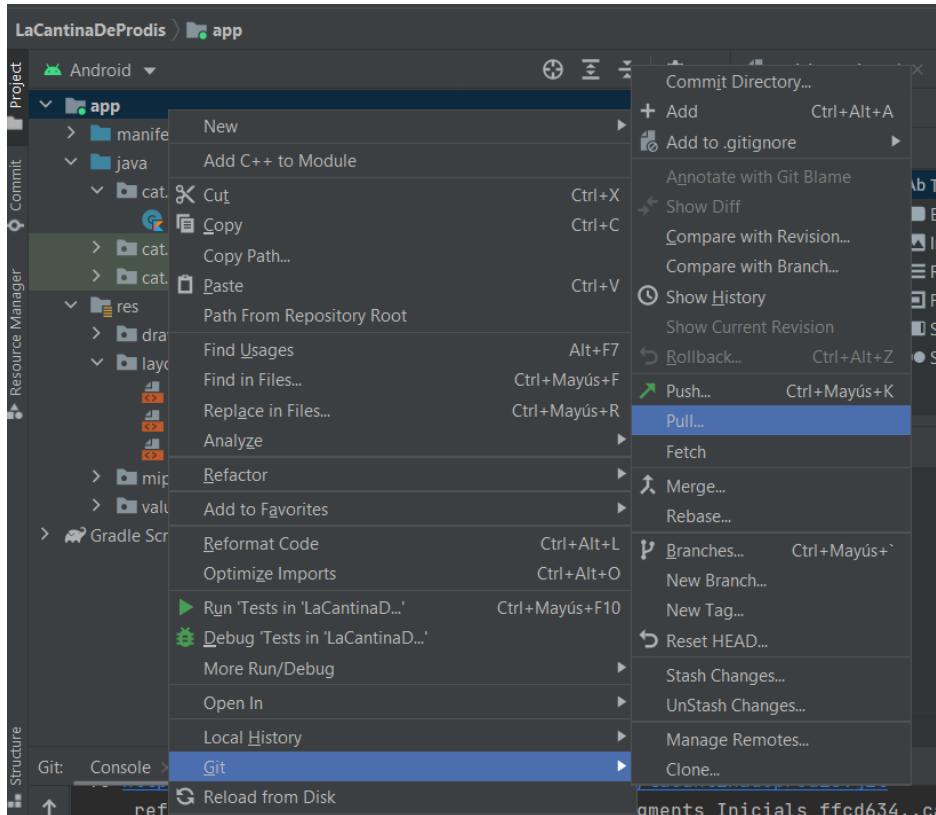
```
git push origin :crazy-experiment
```

Enviaran un senyal d'eliminació al repositori d'origen remot que desencadena l'eliminació de la branca remota crazy-experiment.

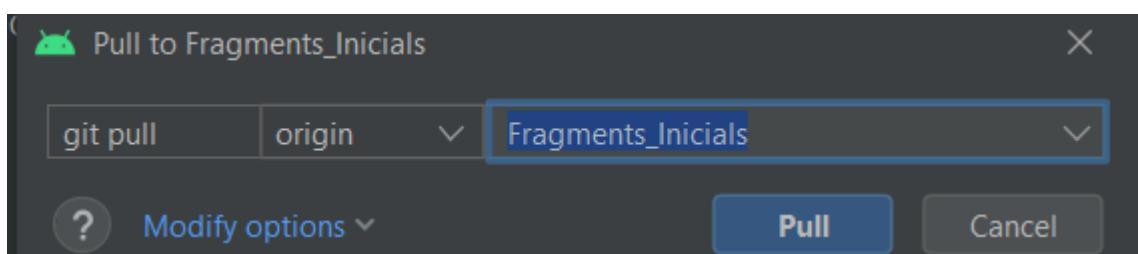
5.1.2.2.5 Treballant amb branques

Sempre que ens disposem a treballar en projectes en què estan ubicats remotament, el primer que hem de fer és un *pull* per assegurar-nos que el que tenim nosaltres és el que està en remot.

Per fer-ho, és tan fàcil com fer clic dret sobre app o projecte, seleccionar git i finalment seleccionar pull:

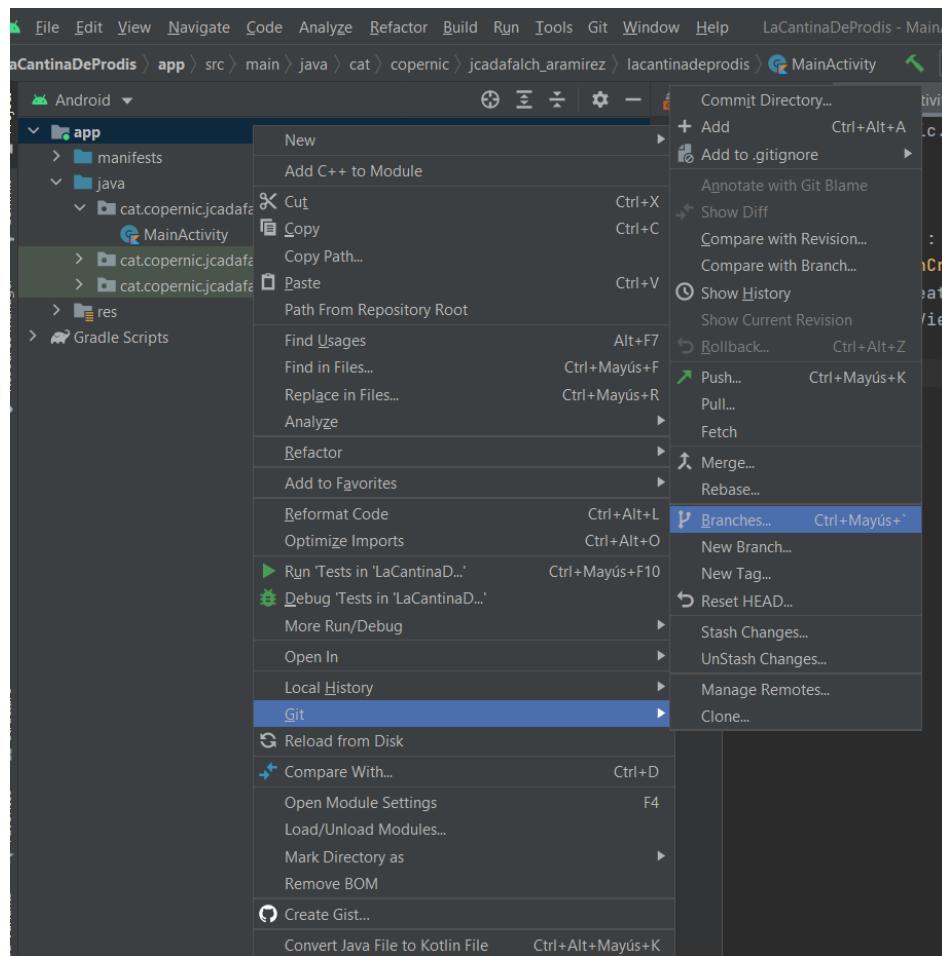


En seleccionar-ho serà tan fàcil com indicar de quina branca farem el pull i clicar sobre pull.

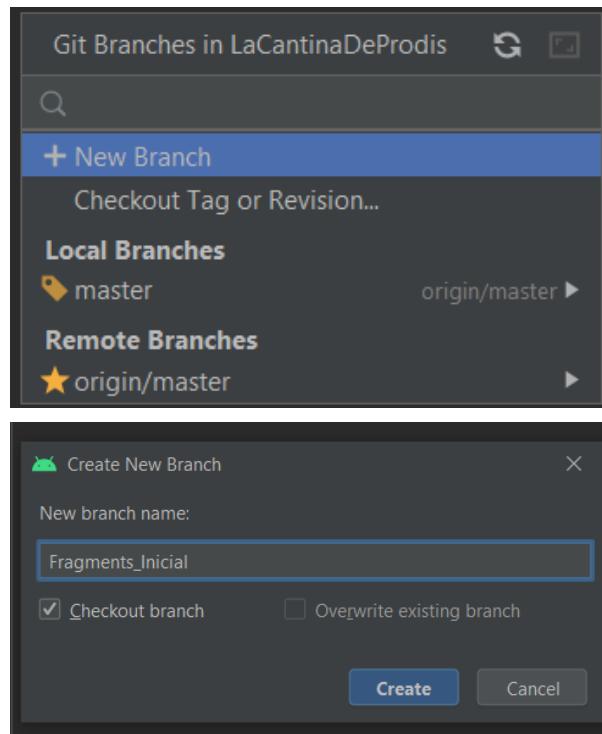


Un cop fet això, el més òptim és crear-nos una branca i treballar des d'ella, per tal no interferir amb els altres companys de projecte.

Per fer-ho només hem de fer clic dret sobre app o projecte, seleccionar git i seleccionar *branches*.



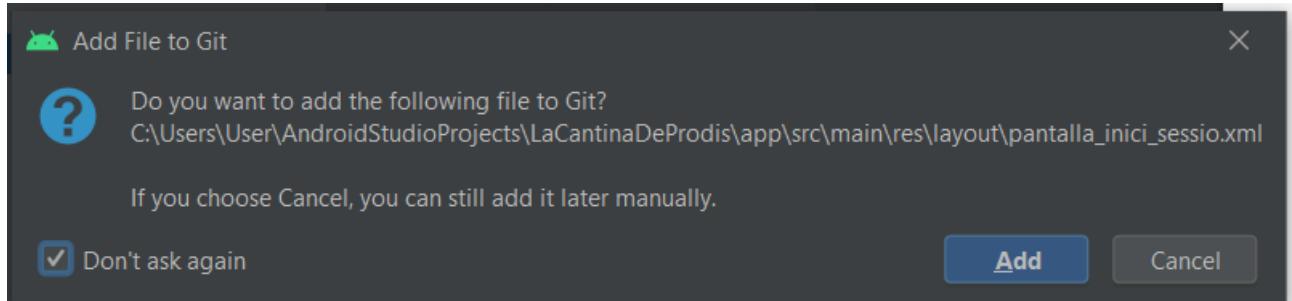
A continuació clicar sobre *New Branch*, introduir el nom de branca i fer clic en *create per crear-la*



The top screenshot shows the 'Git Branches' screen for the 'LaCantinaDeProdis' repository. It displays local branches ('master') and remote branches ('origin/master'). A blue highlight is on the '+ New Branch' button. The bottom screenshot shows a modal dialog titled 'Create New Branch'. It has a text input field containing 'Fragments_Inicial', a checked checkbox for 'Checkout branch', and an unchecked checkbox for 'Overwrite existing branch'. At the bottom are 'Create' and 'Cancel' buttons.

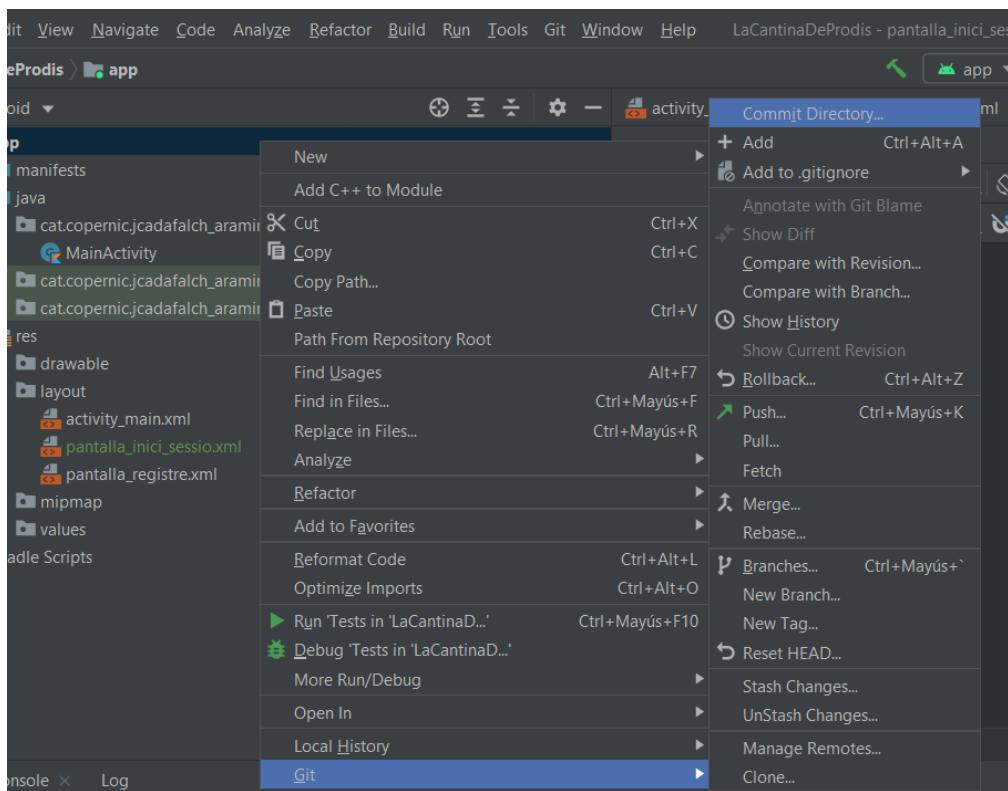
Un cop hem creat la branca i comencem a treballar sobre ella el més comú és crear un fixer del que necessitem.

La primera vegada que ho fem (el primer fitxer que creem) ens apareix el següent missatge. Seleccionem “Don’t ask again” i fem apretem “Add”. Amb això estem indicant que cada fitxer que creem es faci un Add.

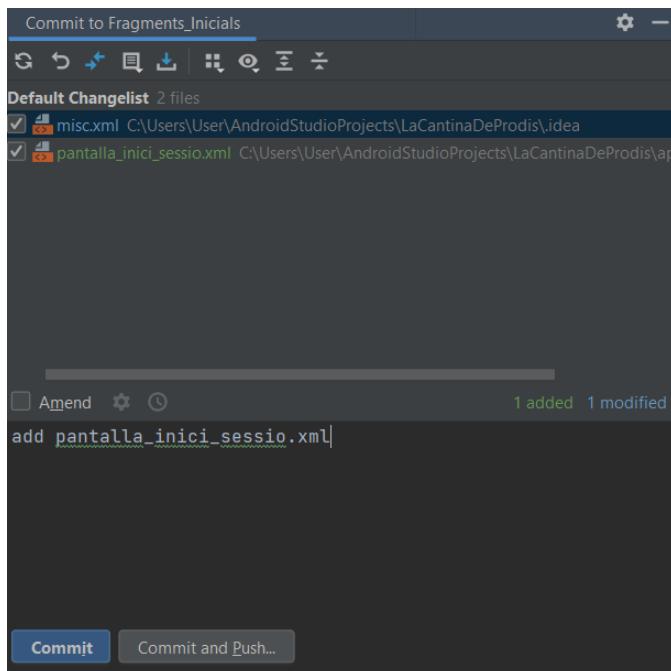


Un cop hem acabat per aquell dia, el més recomanable és sincronitzar el treball fet amb el repositori remot, per fer-ho fem un commit del directory arrel, és a dir de *Project* o *app*

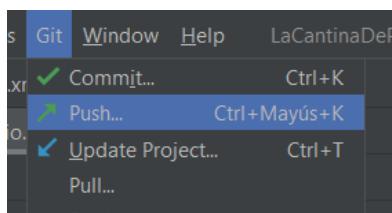
Seguim la ruta per anar a git (vista abans) i seleccionem *Commit Directory*:



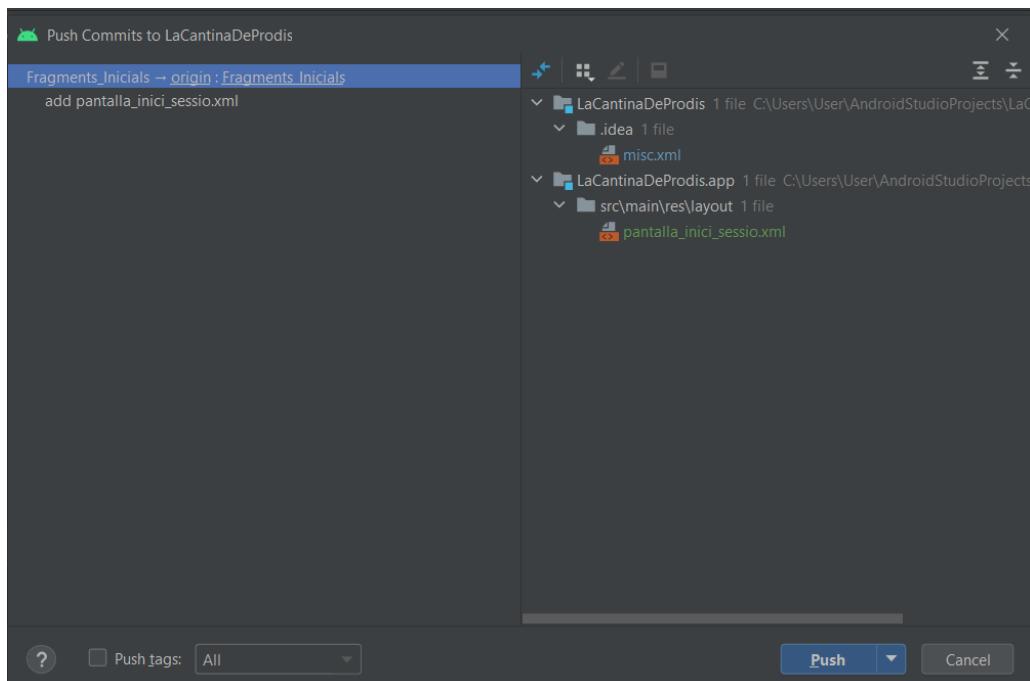
A continuació seleccionem els fitxers que volem incloure en el commit, afegim un missatge descriptiu (en el nostre cas hem afegir una nova pantalla) i pixem a commit.



Finalment per penjar-ho al git definitivament és tan senzill com fer un push.



Fem click a push i ja ho haurem pujat al git

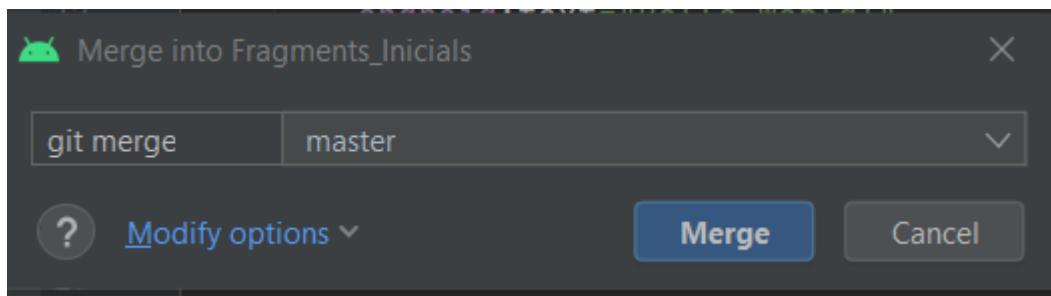


Per acabar, un cop hem acabat de treballar amb aquella branca que havíem creat al principi i enviar el contingut a la principal, només haurem de fer un merge.

Anant a git i seleccionant Merge:



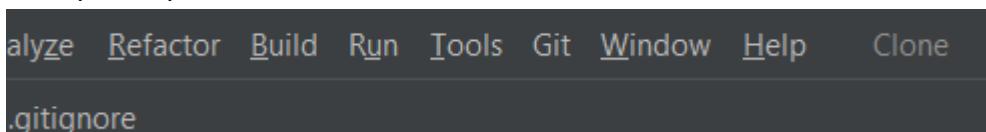
En el menú que s'acaba d'obrir seleccionar la branca en la qual volem fer un merge amb la branca master



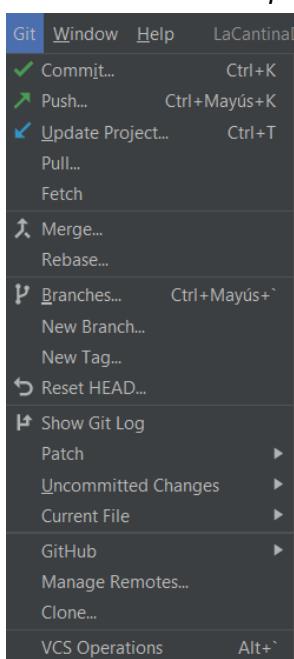
5.1.2.3 Historial de versions

L'historial de versions és un apartat dins del nostre projecte que ens permet veure totes les versions del projecte que hem anat pujant durant el temps. Per veure l'historial de versions hem d'anar a:

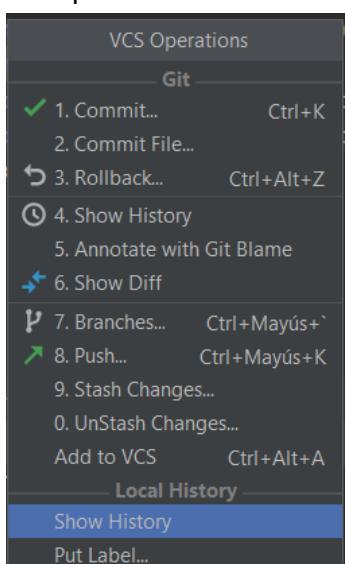
1. Anar al nostre projecte.
2. En la part superior anar al menú de Git.



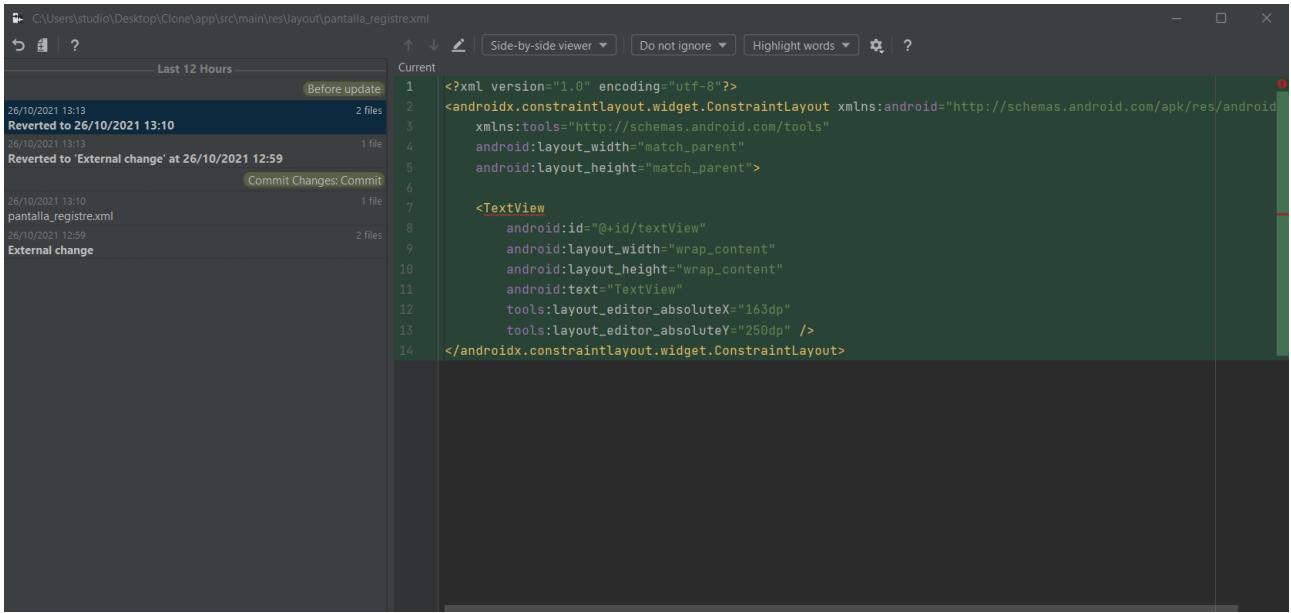
3. Seleccionem *VCS Operations*



4. En aquest nou menú seleccionem *Show History*.



5. S'obrirà un menu on sortiran tots els canvis fets en els diferents pushes.



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        tools:layout_editor_absoluteX="163dp"
        tools:layout_editor_absoluteY="250dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

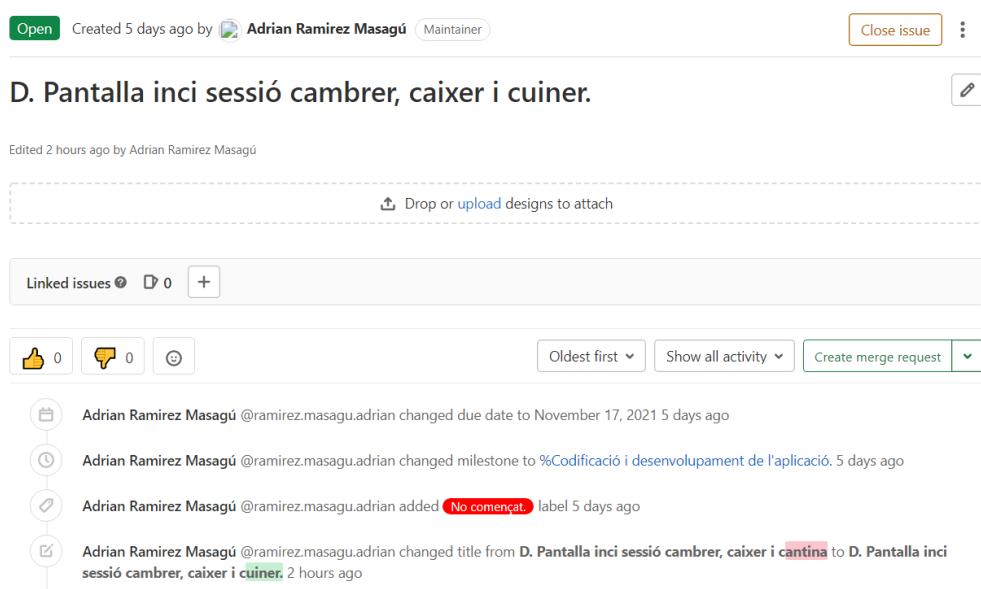
5.1.2.4 Issues

5.1.2.4.1 Seguiment dels issues

El seguiment dels issues creats es pot fer des de l'apartat d'issues del teu projecte. En aquest apartat surten tots els issues creats amb totes les característiques afegides específicament per cada issue (nom, id, milestone, etiquetes, data de venciment, etc...).

5.1.2.4.2 Tancament del issues

Per tancar un issue s'ha d'anar a l'issue que es vol tancar i en la part superior esquerra surt un botó on posa *Close issue*.



The screenshot shows a GitLab issue page for an issue titled "D. Pantalla inci sessió cambrer, caixer i cuiner.". The top bar includes status "Open", creation date "Created 5 days ago by Adrian Ramirez Masagú (Maintainer)", and a "Close issue" button. Below the title is a note "Edited 2 hours ago by Adrian Ramirez Masagú". A central area has a dashed box for attachments with "Drop or upload designs to attach". Below this is a "Linked issues" section with a "+ Add" button. At the bottom are social sharing icons (like, dislike, share) and a timeline of activity:

- Adrian Ramirez Masagú @ramirez.masagu.adrian changed due date to November 17, 2021 5 days ago
- Adrian Ramirez Masagú @ramirez.masagu.adrian changed milestone to %Codificació i desenvolupament de l'aplicació. 5 days ago
- Adrian Ramirez Masagú @ramirez.masagu.adrian added **No començat** label 5 days ago
- Adrian Ramirez Masagú @ramirez.masagu.adrian changed title from D. Pantalla inci sessió cambrer, caixer i cantina to D. Pantalla inci sessió cambrer, caixer i cuiner. 2 hours ago

5.1.2.4.3 Etiquetes

Les etiquetes dels issues són un petit text amb un color assignat que donen una informació específica introduïda per un dels integrants del projecte. En el nostre cas tenim tres etiquetes per informat l'estat del issue; "No començat" que és de color vermell, "En procés" que és de color taronja i "Acabat" que és de color verd.

Labels Edit

No començat. X

No començat. ▼

Assign labels X

Search

Acabat.

En procés.

No començat.

5.1.2.4.4 Assignees

En l'apartat d'assignees es poden indicar que integrant del projecte fa cada issue.

Assignee Edit

 Adrian Ramirez Masagú
@ramirez.masagu.adrian

5.1.2.4.5 Milestone

En l'apartat del milestone s'ha d'indicar el milestone que té l'issue. Un milestone és una manera de controlar un issue o un merge de manera que sigui un objectiu dins d'un període de temps.

Milestone Edit

Codificació i desenvolupament de
l'aplicació.

5.1.2.4.6 Due date

La due date és la data límit en la qual l'issue ha d'estar acabat.

Due date Edit
Dec 22, 2021 - remove due date

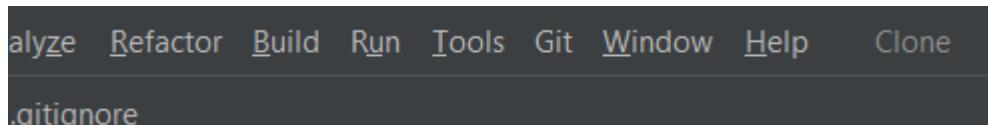
5.1.2.5 Merge requests

Les merge requests són la principal manera de treballar de manera col·laborativa amb el codi del projecte. Funcionen de la següent manera: a cada projecte hi ha una branca predefinida la qual conté la versió principal del codi, d'aquesta branca surten branques secundàries amb propostes de canvi pel codi principal. Una vegada creada aquesta branca secundària amb el codi a introduir a la branca principal per unificar el codi en una sola branca.

5.1.2.5.1 Com fer un merge request

Per fer un merge request hem de fer el següent:

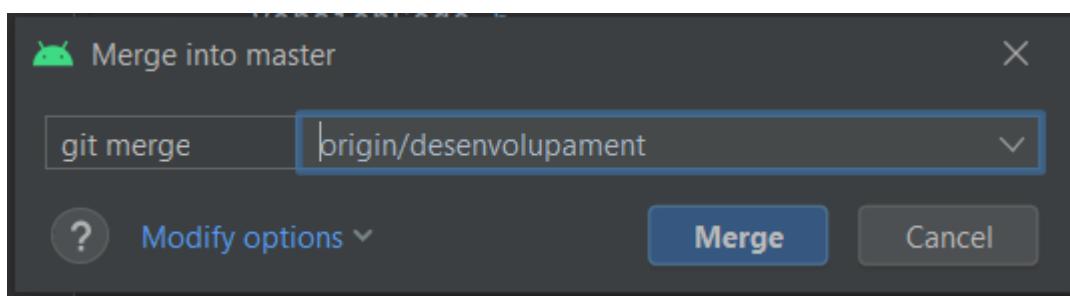
1. Anar al nostre projecte.
2. En la part superior anar al menú de Git.



3. En aquest menú seleccionar Merge.



4. En el menú que s'acaba d'obrir seleccionar la branca en la que volem fer un merge amb la branca master.

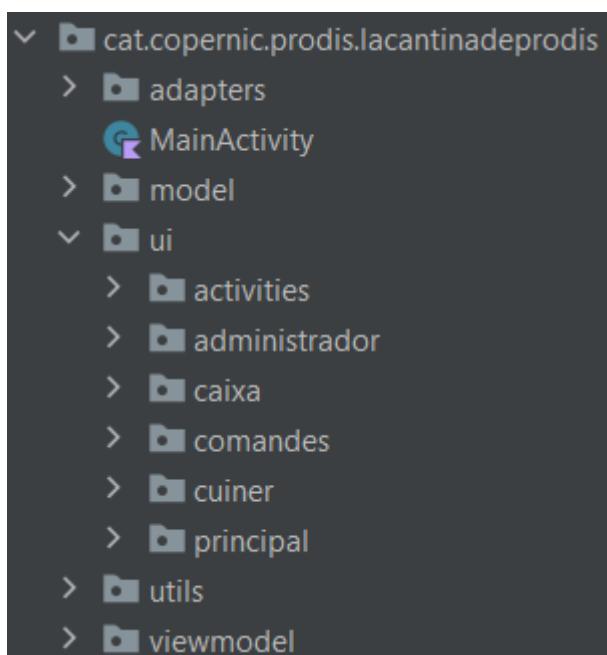


5.2 Estructura del projecte

5.2.1 Packages

Hem dividit el projecte de la següent manera:

- Carpeta general
- Adapters
- Model
- Activities
- Packages per als diferents tipus d'usuaris
 - Administrador
 - Caixa
 - Comandes
 - Cuiner
- Principal
- Utils
- Viewmodel



5.2.1.1 ViewModel & LiveData

```

class PantallaEdicioPerfilViewModel() : ViewModel() {
    lateinit var dni: String

    private val _nom = MutableLiveData<String>()
    val nom: LiveData<String>
        get() = _nom

    private val _cognom = MutableLiveData<String>()
    val cognom: LiveData<String>
        get() = _cognom

    init{
        var tascal: Job? = crearCorrutina()

        tascal?.let {}
    }

    fun crearCorrutina(): GlobalScope. launch(Dispatchers.Main) { this: CoroutineScope
        withContext(Dispatchers.IO){ this: CoroutineScope
            val db = Firebase Firestore.getInstance()

            val currentUser =
                FirebaseAuth.getInstance().currentUser

            db.collection( collectionPath: "users").get().addOnSuccessListener { result ->
                for (document in result) {
                    if (currentUser?.email.toString() == document.get("email").toString()) {
                        dni = document.id
                        db.collection( collectionPath: "users").document(dni).get()
                            .addOnSuccessListener { document ->
                                if (document != null) {
                                    _nom.value = document.get("username").toString()
                                    _cognom.value = document.get("usersurname").toString()
                                }
                            }
                    }
                }
            }
        }
    }
}

//Declrem el view model
viewModel = ViewModelProvider( owner: this)[PantallaEdicioPerfilViewModel::class.java]

viewModel.dni.observe( owner: this, Observer { it: String!
    dni = it
    //Cridem a la funció per agafar l'imatge del usuari
    agafarImatgeUsuari(dni)

})

//Fem que desde el view model s'observevi els canvis del camp de text del nom i del cognom
viewModel.nom.observe( owner: this, Observer { it: String!
    binding.editTxtNom.setText(it.toString())
})

viewModel.cognom.observe( owner: this, Observer { it: String!
    binding.editTxtCognom.setText(it.toString())
})

```

5.2.1.2 Coroutines

```

init{
    var tasca1: Job? = crearCorrutina()

    tasca1.let {}
}

fun crearCorrutina()= GlobalScope.launch(Dispatchers.Main) { this CoroutineScope
    withContext(Dispatchers.IO){ this CoroutineScope
        val db = FirebaseFirestore.getInstance()

        val currentUser =
            FirebaseAuth.getInstance().currentUser

        db.collection( collectionPath: "users").get().addOnSuccessListener { result ->
            for (document in result) {
                if (currentUser?.email?.toString() == document.get("email").toString()) {
                    dni = document.id
                    db.collection( collectionPath: "users").document(dni).get()
                        .addOnSuccessListener { document ->
                            if (document != null) {
                                _nom.value = document.get("username").toString()
                                _cognom.value = document.get("usersurname").toString()
                            }
                        }
                }
            }
        }
    }
}

```

5.2.1.3 Data Class

model

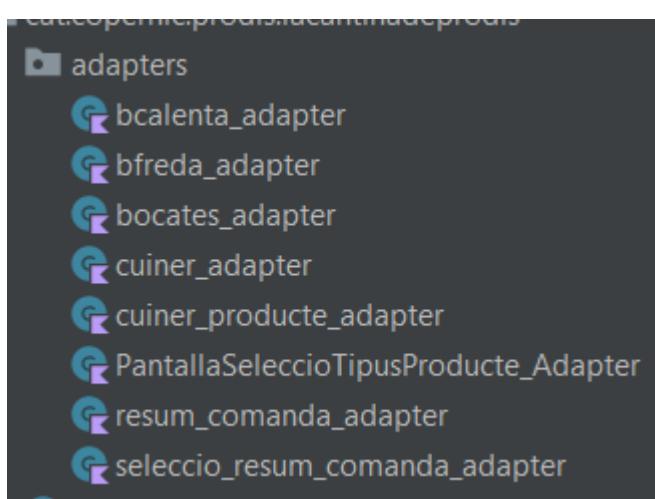
-  dataclass
-  dtclss_cuiner
-  dtclss_cuiner_producte
-  dtclss_products
-  dtclss_resum_comanda
-  dtclss_seleccio_resum_comanda

```

data class dtclss_products(
    var nom: String? = null,
    var idProducte: Long? = null,
    var nomid: String? = null,
    var img: String? = null,
    var preu: Int? = null,
    var tipus: String? = null,
    var visible: Boolean? = null
)

```

5.2.1.4 Adapter



```
class Resum_comanda_adapter(private val producte: ArrayList<dtclss_resum_comanda>) :  
    RecyclerView.Adapter<Resum_comanda_adapter.ViewHolder>() {  
  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
        val v = LayoutInflater.from(parent.context).inflate(R.layout.resum_comanda_card, parent, attachToRoot: false)  
        return ViewHolder(v)  
    }  
  
    override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
        val producte: dtclss_resum_comanda = producte[position]  
  
        holder.itemNom.text = producte.nomProducte  
        val preu = "${producte.preuProducte.toString()}€"  
        holder.itemPreu.text = preu  
    }  
  
    override fun getItemCount(): Int {  
        return producte.size  
    }  
  
    class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
        val itemNom: TextView = itemView.findViewById(R.id.itemProducte)  
        val itemPreu: TextView = itemView.findViewById(R.id.itemPreu)  
    }  
}
```

5.2.2 Layouts

Pantalla Principal:

En la pantalla principal ens trobem cinc botons amb els tipus d'usuari. En clicar en el tipus d'usuari amb el qual volem iniciar sessió, se'n obrirà l'inici de sessió per tal de poder accedir a l'apartat del nostre tipus d'usuari.



```

package cat.copernic.prodis.lacantinadeprodis.ui.principal

import ...

class PantallaPrincipal : Fragment() {

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?): View {
        val binding: FragmentPantallaPrincipalBinding = DataBindingUtil.inflate(
            inflater, R.layout.fragment_pantalla_principal, container, false
        )
        binding.btnPprincipalClient.setOnClickListener { view : View ->
            view.findNavController().navigate(PantallaPrincipalDirections.
                actionPantallaPrincipalToPantallaIniciSessioClientAdmin( usertype: "client"))
        }

        binding.btnPprincipalCambrer.setOnClickListener{ view : View ->
            view.findNavController().navigate(PantallaPrincipalDirections.
                actionPantallaPrincipalToPantallaIniciSessioClientAdmin( usertype: "cambrer"))
        }

        binding.btnPprincipalCaixer.setOnClickListener{ view : View ->
            view.findNavController().navigate(PantallaPrincipalDirections.
                actionPantallaPrincipalToPantallaIniciSessioClientAdmin( usertype: "caixer"))
        }

        binding.btnPprincipalCuiner.setOnClickListener{ view : View ->
            view.findNavController().navigate(PantallaPrincipalDirections.
                actionPantallaPrincipalToPantallaIniciSessioClientAdmin( usertype: "cuiner"))
        }

        binding.btnPprincipalAdministrador.setOnClickListener{ view : View ->
            view.findNavController().navigate(PantallaPrincipalDirections.
                actionPantallaPrincipalToPantallaIniciSessioClientAdmin( usertype: "admin"))
        }

        return binding.root
    }
}

```

Pantalla Principal Administrador:

Pantalla principal de l'administrador, en aquesta pantalla l'administrador seleccionarà la tasca que vol realitzar, sigui afegir un usuari, modificar o eliminar un usuari, afegir un nou producte o administrar algun dels productes.



```

binding.btnAddAdministratorPrincipalNouUsuari.setOnClickListener {
    view?.findNavController()
        ?.navigate(
            PantallaAdministradorPrincipalDirections.
            actionPantallaAdministradorPrincipalToPantallaAdministradorNouUsuari(
                arrUserType
            )
        )
}

binding.btnAddAdministratorPrincipalEliminarModificarUsuari.setOnClickListener {
    db.collection( collectionPath: "users" ).get().addOnSuccessListener { result ->
        for (document in result) {
            if (document.id != "usertypes") {
                val user =
                    document.get("username").toString() + " " + document.get("usersurname")
                        .toString()
                arrUser.add(user)
                arrUserId.add(document.id)
            }
        }
        view?.findNavController()?.navigate(
            PantallaAdministradorPrincipalDirections.
            actionPantallaAdministradorPrincipalToPantallaAdministradorModificarUsuari(
                arrUser, arrUserId, arrUserType
            )
        )
    }
}

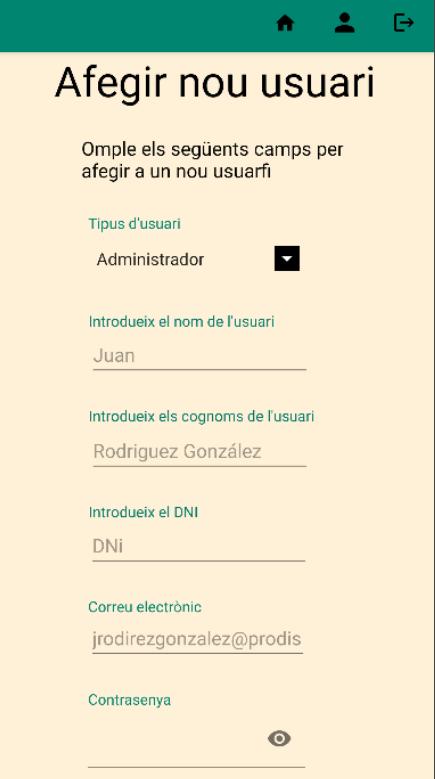
binding.btnAddAdministratorPrincipalNouProducte.setOnClickListener {
    view?.findNavController()
        ?.navigate(PantallaAdministradorPrincipalDirections.
            actionPantallaAdministradorPrincipalToPantallaAdministradorNouProducte())
}

binding.btnAddAdministratorPrincipalAdministrarProductes.setOnClickListener {
    view?.findNavController()
        ?.navigate(PantallaAdministradorPrincipalDirections.
            actionPantallaAdministradorPrincipalToPantallaAdministradorAdministrarProducte())
}

```

Pantalla Afegir Usuari:

En aquesta pantalla l'administrador pot afegir un nou usuari, primer indicarà el tipus d'usuari, posteriorment omplirà els camps i finalment clicarà en el botó "guardar".



```

val args = PantallaAdministradorNouUsuariArgs.fromBundle(requireArguments())
arrUserType = args.usertypeArr as ArrayList<String>

val context = this.requireContext()

spinner = binding.spnPAdministradorModificarUsuariTipusUsuari
adapter = ArrayAdapter(context, android.R.layout.simple_spinner_item, arrUserType)
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
spinner.adapter = adapter
spinner.onItemSelectedListener = this

binding.btnPNouUsuariGurardar.setOnClickListener { it: View! -
    var usertype = ""
    when (spinner.selectedItem.toString()) {
        "Administrador" -> usertype = "admin"
        "Caixer" -> usertype = "caixer"
        "Camarer" -> usertype = "camarer"
        "Cuiner" -> usertype = "cuiner"
        "Client Remot" -> usertype = "clientR"
        "Cuiner" -> usertype = "cuiner"
    }
    if (datavalids(
            nom: binding.dtTxtPAdministradorNouUsuariPersonName.text.toString() + "",
            cognom: binding.dtTxtPAdministradorNouUsuariPersonSurname.text.toString() + "",
            dni: binding.dtTxtPAdministradorNouUsuariDni.text.toString() + "",
            email: binding.dtTxtPAdministradorNouUsuariEmail2.text.toString() + "",
            password: binding.dtTxtPAdministradorNouUsuariPassword.text.toString() + "",
            usertype: usertype + ""))
    ) {
        newUser(
            nom: binding.dtTxtPAdministradorNouUsuariPersonName.text.toString() + "",
            cognom: binding.dtTxtPAdministradorNouUsuariPersonSurname.text.toString() + "",
            dni: binding.dtTxtPAdministradorNouUsuariDni.text.toString() + "",
            email: binding.dtTxtPAdministradorNouUsuariEmail2.text.toString() + "",
            password: binding.dtTxtPAdministradorNouUsuariPassword.text.toString() + "",
            usertype
        )
    }
}

```

Pantalla Modificar Usuari:

En aquesta pantalla l'administrador podrà modificar un usuari existent, o si ho creu oportú, eliminar-lo.



```

binding.btnAddPAdministradorModificarUsuariGuardar.setOnClickListener {
    var usertype = ""
    when (spinnerUserType.getSelectedItem.toString()) {
        "Administrador" -> usertype = "admin"
        "Caixer" -> usertype = "caixer"
        "Cambrer" -> usertype = "cambrer"
        "Client" -> usertype = "client"
        "Client Remot" -> usertype = "clientR"
        "Cuiner" -> usertype = "cuiner"
    }

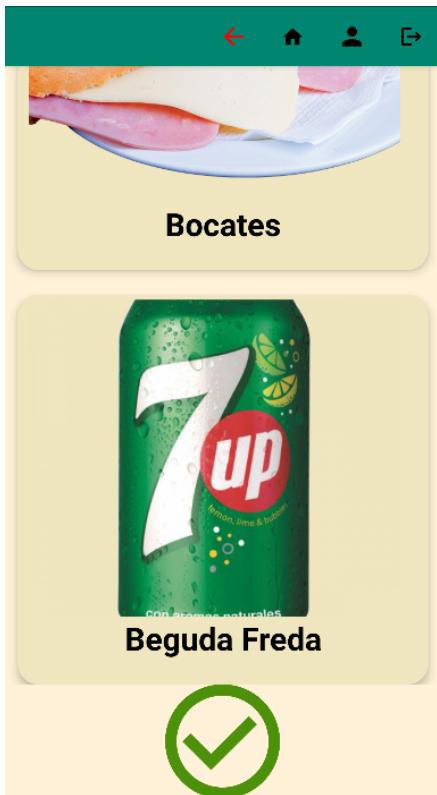
    saveUser(usertype)
}

binding.btnAddPAdministradorModificarUsuariEliminarUsuari.setOnClickListener {
    val user = binding.dtTxtPAdministradorModificarUsuariPersonName.text.toString() +
        " " +
        binding.dtTxtPAdministradorModificarUsuariPersonSurname.text.toString()
    val dni = binding.dtTxtPAdministradorModificarUsuariDni.text.toString()
    val builder = androidx.appcompat.app.AlertDialog.Builder(this.requireContext())
    builder.setTitle("AVIS!!!")
    builder.setMessage("Estas segur que vols eliminar a l'usuari: $user")
    builder.setPositiveButton("Acceptar") { _, _ ->
        deleteUser(dni)
    }
    builder.setNegativeButton("Cancelar", null)
    val dialog: androidx.appcompat.app.AlertDialog = builder.create()
    dialog.show()
}

```

Pantalla Selecció Tipus de Producte:

En aquesta pantalla tant client com cambrer seleccionaran la categoria del producte que volen afegir a la comanda.



```

recyclerView = binding.recyclerViewSeleccioProducte
recyclerView.setLayoutManager(this.context)
recyclerView.setHasFixedSize(true)

productelist = arrayListof()

db.collection( collectionPath: "products").document( documentPath: "categories")
    .get().addOnSuccessListener { document ->
        val stBocata = document.get("bocata").toString()
        val stCalenta = document.get("calenta").toString()
        val stFreda = document.get("freda").toString()
        val imBocata = "Bocata de pernil dolç amb formatge.png"
        val imCalenta = "Café.png"
        val imFreda = "7up.png"

        productelist.add(dataclass(stBocata, imBocata))
        productelist.add(dataclass(stFreda, imFreda))
        productelist.add(dataclass(stCalenta, imCalenta))

        Padaper = PantallaSeleccioTipusProducte_Adapter(
            productelist
        )
        recyclerView.adapter = Padaper
    }

binding.btnConfirmar.setOnClickListener { it: View?
    db.collection( collectionPath: "users").get().addOnSuccessListener { result ->
        for (document in result) {
            if (document.id != "usertypes") {
                val user =
                    document.get("username").toString() + " " + document.get("usersurname")
                    .toString()
                arrUser.add(user)
            }
        }
        arrUser.add("Extern")
        view?.findNavController()?.navigate(
            PantallaSeleccioTipusProdcteDirections.actionPantallaSeleccioTipusProducteToPan
                arrUser
        )
    }
}

```

Pantalla Selecció Bocates:

En aquesta pantalla tant client com cambrer seleccionaran el tipus de bocata que volen afegir a la comanda.



```
override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View {
    val binding: FragmentPantallaSeleccioBocataBinding = DataBindingUtil.inflate(
        inflater, R.layout.fragment_pantalla_seleccio_bocata, container, false
    )

    binding.backButton.setOnClickListener{ it: View!
        view?.findNavController()?.navigate(PantallaSeleccioBocataDirections.
        actionPantallaSeleccioBocataToPantallaSeleccioTipusProducte())
    }

    recyclerView = binding.rcyclrVwBocata
    recyclerView.setLayoutManager(LinearLayoutManager(this.context))
    recyclerView.setHasFixedSize(true)

    productesArrayList = arrayListOf()

    bocatesAdapter = bocates_adapter(productesArrayList)

    recyclerView.adapter = bocatesAdapter

    eventChangeListener()

    return binding.root
}
```

```
private fun eventChangeListener() {
    db.collection( collectionPath: "productes").addSnapshotListener(object : EventListener<QuerySnapshot>{
        @SuppressLint( ...value: "NotifyDataSetChanged")
        override fun onEvent(value: QuerySnapshot?, error: FirebaseFirestoreException?) {
            if (error != null){
                Log.e( tag: "Firestore Error", error.message.toString())
                return
            }
            db.collection( collectionPath: "productes").get().addOnSuccessListener { result -
                for (document in result){
                    if (document.id != "categories"){
                        if (document.get("tipus") == "Bocates" && document.get("visible") == true){
                            productesArrayList.add(document.toObject(dtclss_productes::class.java))
                        }
                    }
                }
                bocatesAdapter.notifyDataSetChanged()
            }
        }
    })
}
```

Pantalla Selecció Beguda Freda:

En aquesta pantalla tant client com cambrer seleccionaran el tipus de beguda freda que volen afegir a la comanda.



```
override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View {
    val binding: FragmentPantallaSeleccioBegudaFredaBinding = DataBindingUtil.inflate(
        inflater, R.layout.fragment_pantalla_seleccio_beguda_freda, container, attachToParent: false
    )

    binding.backButton.setOnClickListener { it: View ->
        view?.findNavController()?.navigate(PantallaSeleccioBegudaFredaDirections.actionPantallaSeleccioBegudaFredaToPantallaSeleccioTipusProducte())
    }

    recyclerView = binding.rcyclrvwBfreda
    recyclerView.layoutManager = LinearLayoutManager(this.context)
    recyclerView.setHasFixedSize(true)

    productesArrayList = arrayListOf()
    bfredaAdapter = bfreda_adapter(productesArrayList)

    recyclerView.adapter = bfredaAdapter
    eventChangeListener()

    return binding.root
}
```

```
private fun eventChangeListener() {
    db.collection( collectionPath: "productes").addSnapshotListener(object : EventListener<QuerySnapshot> {
        @SuppressLint( ...value: "NotifyDataSetChanged")
        override fun onEvent(value: QuerySnapshot?, error: FirebaseFirestoreException?) {
            if (error != null){
                Log.e( tag: "Firestore Error", error.message.toString())
                return
            }
            db.collection( collectionPath: "productes").get().addOnSuccessListener { result ->
                for (document in result){
                    if (document.id != "categories"){
                        if (document.get("tipus") == "Beguda freda" && document.get("visible") == true){
                            productesArrayList.add(document.toObject(dtclss_productes::class.java))
                        }
                    }
                }
                bfredaAdapter.notifyDataSetChanged()
            }
        }
    })
}
```

Pantalla Selecció Beguda Calenta:

En aquesta pantalla tant client com cambrer seleccionaran el tipus de beguda calenta que volen afegir a la comanda.



```
override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View {
    val binding: FragmentPantallaSeleccioBegudaCalentaBinding = DataBindingUtil.inflate(
        inflater, R.layout.fragment_pantalla_seleccio_beguda_calenta, container, false)
    binding.backButton.setOnClickListener { it: View ->
        view?.findNavController()?.navigate(PantallaSeleccioBegudaCalentaDirections.
            actionPantallaSeleccioBegudaCalentaToPantallaSeleccioTipusProducte())
    }

    recyclerView = binding.recyclerviewBcalenta
    recyclerView.setLayoutManager(this.context)
    recyclerView.setHasFixedSize(true)

    productesArrayList = arrayListOf()
    bcalentaAdapter = bcalenta_adapter(productesArrayList)

    recyclerView.adapter = bcalentaAdapter

    eventChangeListener()

    return binding.root
}
```

```
private fun eventChangeListener() {
    db.collection("productes").addSnapshotListener(object : EventListener<QuerySnapshot> {
        @SuppressLint("NotifyDataSetChanged")
        override fun onEvent(value: QuerySnapshot?, error: FirebaseFirestoreException?) {
            if (error != null){
                Log.e("Firestore Error", error.message.toString())
                return
            }
            db.collection("productes").get().addOnSuccessListener { result ->
                for (document in result){
                    if (document.id != "categories"){
                        if (document.get("tipus") == "Beguda Calenta" && document.get("visible") == true){
                            productesArrayList.add(document.toObject(dtclss_productes::class.java))
                        }
                    }
                }
                bcalentaAdapter.notifyDataSetChanged()
            }
        }
    })
}
```

Pantalla Nom Client Comanda:

En aquesta pantalla només hi accedirà el cambrer, serà en aquesta pantalla en la qual el cambrer seleccionarà el nom del client que està demanant la comanda, en cas que el client no estigués en el llistat, seleccionarà "Extern" i apareixerà un camp de text per introduir manualment el nom del client.



```

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View {
    binding = DataBindingUtil.inflate(
        inflater, R.layout.fragment_pantalla_seleccio_nom_client_comanda, container, false
    )
    val args = PantallaSeleccioNomClientComandaArgs.fromBundle(requireArguments())
    userArc = args.arrUser
    spinner = binding.spnNomClient

    val context = this.requireContext()

    adapter = ArrayAdapter(context, android.R.layout.simple_spinner_item, userArc)
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
    spinner.adapter = adapter
    spinner.onItemSelectedListener = this

    binding.btnTornaEnrerre.setOnClickListener { view: View? ->
        view?.findNavController()?.navigate(PantallaSeleccioNomClientComandaDirections.
            actionPantallaSeleccioNomClientComandaToPantallaSeleccioTipusProducte())
    }

    return binding.root
}

override fun onItemSelected(parent: AdapterView<*>?, view: View?, position: Int, id: Long) {
    if (position == userArc.size-1){
        binding.dtTxtNomUsuariExtern.visibility = View.VISIBLE
    }else{
        binding.dtTxtNomUsuariExtern.visibility = View.INVISIBLE
    }
}

```



Pantalla edició perfil

En aquesta pantalla l'usuari podrà editar la informació del seu perfil. Podrà editar: nom, cognom i contrasenya. També podrà canviar la seva foto de perfil i la mida de la lletra.



```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = DataBindingUtil.setContentView<FragmentPantallaEdicioPerfilBinding>(
        activity: this,
        R.layout.fragment_pantalla_edicio_perfil
    )

    var bundle = intent.extras
    dni = bundle?.getString(key: "dni").toString()

    agafarImatgeUsuari()

    binding.btnCambiarFoto.setOnClickListener { view: View ->
        triaCamGaleria()
    }

    db.collection(collectionPath: "users").document(dni).get()
        .addOnSuccessListener { document ->
            if (document != null) {
                binding.editTxtNom.setText(document.get("username").toString())
            }
        }

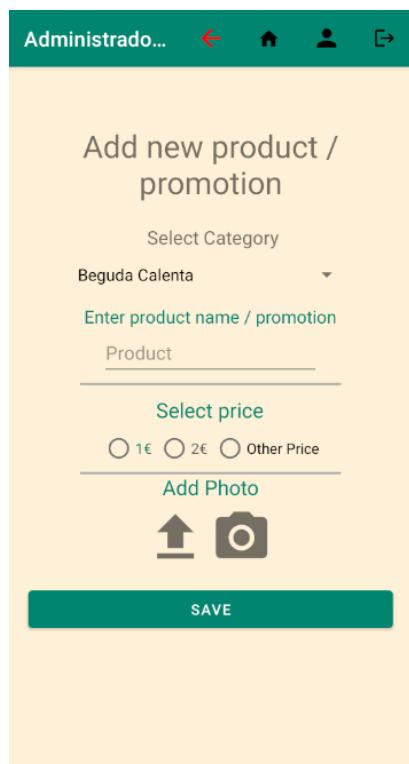
    db.collection(collectionPath: "users").document(dni).get()
        .addOnSuccessListener { document ->
            if (document != null) {
                binding.editTxtCognom.setText(document.get("usersurname").toString())
            }
        }

    binding.btnGuardar.setOnClickListener { view: View ->
        pujarImatge(view)
        if (dataisValidis(
            binding.editTxtNom.text.toString(),
            binding.editTxtCognom.text.toString()
        )) {
            db.collection(collectionPath: "users").document(dni).update(
                hashMapOf(
                    "username" to binding.editTxtNom.text.toString(),
                    "usersurname" to binding.editTxtCognom.text.toString(),
                ) as Map<String, Any>
            )
            val currentUserPass =
                FirebaseAuth.getInstance().currentUser

            currentUserPass?.updatePassword(binding.editTextContrassenya.text.toString())
                ?.addOnSuccessListener { it: Void -
                    println("Entra en successlistener")
                    db.collection(collectionPath: "users").document(dni).update(
                        hashMapOf(
                            "passwd" to binding.editTextContrassenya.text.toString(),
                        ) as Map<String, Any>
                    )
                }
            Toast.makeText(context: this, text: "Els canvis s'han fet amb èxit", Toast.LENGTH_SHORT).show()
        }
    }
}
```

Pantalla afegir nou producte

En aquesta pantalla es podrà afegir nous productes.



```

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View {
    binding = DataBindingUtil.inflate(
        inflater, R.layout.fragment_pantalla_administrador_nou_producte, container, false
    )

    storageRef = FirebaseStorage.getInstance().getReference()

    tipusProducte = ""

    val spinner: Spinner = binding.spinTipusProducte

    val context = this.requireContext()

    val args = PantallaAdministradorNouProductoArgs.fromBundle(requireArguments())
    arrayTipusProducte = args.arrayTipusProducte as ArrayList<String>

    adapter = ArrayAdapter(context, android.R.layout.simple_spinner_dropdown_item, arrayTipusProducte)

    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)

    spinner.adapter = adapter

    spinner.onItemSelectedListener = this

    binding.imgFotoCamera.setOnClickListener() { it: View! -->
        abrirCamera()
        binding.imgProducte.visibility = View.VISIBLE
    }
}

```

Pantalla administrar productes

En aquesta pantalla es podrà modificar les dades dels productes ja afegits en l'app.



```
//Iniciem el onCreateView
override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View {
    binding = DataBindingUtil.inflate(
        inflater,
        R.layout.fragment_pantalla_administrador_administrar_producto,
        container,
        attachToParent: false
    )
    //Fem que l'imatge del producto desapareixi
    binding.imgProducto2.isGone = true
    //Fem que l'el camp de text del numero decimal del preu del producto desapareixi
    binding.editTextNumberDecimal3.isGone = true
    //Declarem i inicialitzem una referencia al storage
    storageRef = FirebaseStorage.getInstance().getReference()
    //Declarem i inicialitzem el view model
    viewModel =
        ViewModelProvider(owner: this).get(PantallaAdministradorAdministrarProductoViewModel::class.java)

    //Cridem a la funció per carregar l'spinner de tipus de producto
    cargarSpinTipusProductos()
    //Cridem a la funció per carregar l'spinner de producto
    cargarSpinProductos()
    //Cridem a la funció per posar el preu
    setPreu()
    //Cridem a la funció per actualitzar la foto del producto
    actualizaFoto()
    //Cridem a la funció per indicar si el producto es visible o no
    productoEsVisible()
```

Pantalla selecció atributs bocata

En aquesta pantalla es podrà indicar com es voldrà el bocata seleccionat prèviament.



```

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View {
    val binding = FragmentPantallaSeleccioAtributsBocataBinding.inflate(
        inflater, R.layout.fragment_pantalla_seleccio_atributs_bocata, container,
        attachToRoot = false
    )

    val args = PantallaSeleccioAtributsBegudaArgs.fromBundle(requireArguments())
    idProducte = args.idProductes

    binding.btnTornaEnrerre.setOnClickListener { it: View? ->
        findNavController().popBackStack()
    }

    binding.btnConfirmar.setOnClickListener { view: View ->
        view.findNavController()
            .navigate(R.id.action_pantalla_seleccio_atributs_bocata_to_pantalla_seleccio_beguda)
    }
}

senseTomaquet = !binding.checkBoxSenseTomaquet.isChecked
perEmportar = binding.checkBoxPerEmportar.isChecked

val currentUser =
    FirebaseAuth.getInstance().currentUser

var dni: String

db.collection( collectionPath: "users" ).get().addOnSuccessListener { result ->
    for (document in result) {
        if (currentUser?.email.toString() == document.get("email").toString()) {
            dni = document.id
            db.collection( collectionPath: "comandes" ).get().addOnSuccessListener { result ->
                for (document in result) {
                    if (dni.equals(document.get("user"))) {
                        db.collection( collectionPath: "comandes" ).document(document.id)
                            .collection( collectionPath: "productes" ).document().set(
                                ...
                            )
                    }
                }
            }
        }
    }
}

```

Pantalla selecció atributs beguda

En aquesta pantalla es podrà indicar com es voldrà la beguda seleccionada prèviament. Els atributs seran els mateixos tant com si se selecciona una beguda calenta o una beguda freda.



```

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View {
    val binding = DataBindingUtil.inflate(
        inflater, R.layout.fragment_pantalla_seleccio_atributs_beguda, container, attachToParent: false
    )

    var args = PantallaSeleccioAtributsBegudaArgs.fromBundle(requireArguments())
    val currentUser =
        FirebaseAuth.getInstance().currentUser

    var dni: String

    var tasca1: Job? = setAtributs()

    idProducte = args.idProductes

    tasca1.let {

        binding.btnConfirmar.setOnClickListener { view: View ->
            setAtributs()

            db.collection( collectionPath: "users" ).get().addOnSuccessListener { result ->
                for (document in result) {
                    if (currentUser?.email.toString() == document.get("email").toString()) {
                        dni = document.id
                        db.collection( collectionPath: "comandes" ).get().addOnSuccessListener { result ->
                            for (document in result) {
                                if (dni.equals(document.get("user"))) {
                                    db.collection( collectionPath: "comandes" ).document(document.id)
                                        .collection( collectionPath: "productes" ).document().set(
                                            ...
                                        )
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Pantalla cuiner

En aquesta pantalla el cuiner podrà seleccionar de quina comanda voldrà veure el contingut.



```

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View {
    val binding: FragmentPantallaCuinerBinding = DataBindingUtil.inflate(
        inflater, R.layout.fragment_pantalla_cuiner, container, false
    )
    //declaració de tots els elements del recyclerView (recyclerView, adapter i dataclass)

    recyclerView = binding.recyclerviewCuiner

    //En funció de la orientació del dispositiu canviem la orientació del recyclerView
    val gridLayout = GridLayoutManager(this.context, spanCount: 2)
    when (resources.configuration.orientation) {
        Configuration.ORIENTATION_PORTRAIT -> {
            recyclerView.layoutManager = gridLayout
        }
        Configuration.ORIENTATION_LANDSCAPE -> {
            recyclerView.layoutManager =
                GridLayoutManager(this.context, spanCount: 2, RecyclerView.HORIZONTAL, reverseLayout: false)
        }
        Configuration.ORIENTATION_UNDEFINED -> {
            recyclerView.layoutManager =
                GridLayoutManager(this.context, spanCount: 2, RecyclerView.HORIZONTAL, reverseLayout: false)
        }
    }

    recyclerView.setHasFixedSize(true)

    comandesList = arrayListOf()
    cuinerAdapter = cuiner_adapter(comandesList)

    eventChangeListener()
}

```

Pantalla cuiner comanda

En aquesta pantalla el cuiner podrà veure el contingut de la comanda i marcar conforme la comanda està acabada mitjançant el botó en forma de tick.



```

recyclerView.setHasFixedSize(true)
producteList = arrayListOf()
cuinerProducteAdapter = cuiner_producte_adapter(producteList)
eventChangeListener()

binding.btnCuinat.setOnClickListener { it: View ->
    db.collection("comandes").document(documentId).get().addOnSuccessListener { dc ->
        if (dc.get("comandaPagada").toString() == "true"){
            db.collection("comandes").document(documentId).update(
                hashMapOf(
                    "preparat" to true,
                    "visible" to false
                ) as Map<String, Any>
            )
        }else{
            db.collection("comandes").document(documentId).update(
                hashMapOf(
                    "preparat" to true
                ) as Map<String, Any>
            )
        }
        view?.findNavController()?.navigate(PantallaCuinerComandesDirections.actionPantallaCuinerComandaToPantallaCuiner())
    }
}

binding.btnTornaEnerre.setOnClickListener { it: View ->
    view?.findNavController()?.navigate(PantallaCuinerComandesDirections.actionPantallaCuinerComandaToPantallaCuiner())
}

```

Pantalla caixer selecció resum comanda

En aquesta pantalla el caixer podrà seleccionar de quina comanda voldrà veure el contingut



```

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View {
    val binding: FragmentPantallaSeleccioResumComandaBinding = DataBindingUtil.inflate(
        inflater, R.layout.fragment_pantalla_seleccio_resum_comanda, container, false
    )

    recyclerView = binding.recyclerviewSeleccioComandes!!
    val gridLayout = GridLayoutManager(this.context, spanCount: 2)
    //recyclerView.layoutManager = gridLayout

    when (resources.configuration.orientation) {
        Configuration.ORIENTATION_PORTRAIT -> {
            recyclerView.layoutManager = gridLayout
        }
        Configuration.ORIENTATION_LANDSCAPE -> {
            recyclerView.layoutManager =
                GridLayoutManager(this.context, spanCount: 2, RecyclerView.HORIZONTAL,
                    true)
        }
        Configuration.ORIENTATION_UNDEFINED -> {
            recyclerView.layoutManager =
                GridLayoutManager(this.context, spanCount: 2, RecyclerView.HORIZONTAL,
                    true)
        }
    }

    comandalist = arrayListOf()
    recyclerView.setHasFixedSize(true)

    seleccioResumComandaAdapt = seleccio_resum_comanda_adapter(comandalist)

    recyclerView.adapter = seleccioResumComandaAdapt

    eventChangeListener()
}

```

Pantalla caixer resum comanda

En aquesta pantalla el caixer podrà veure el contingut de la comanda i marcar conforme està acabada mitjançant el botó en forma de tick



```

val args = PantallaResumComandesArgs.fromBundle(requireArguments())
comandaId = args.comandaId.toString()
username = args.username.toString()
documentoId = args.documentId.toString()
prevTotal = args.prevTotal

binding.txtUsername.text = username
binding.txtComanda.text = comandaId
binding.txtWPrevTotal.text = prevTotal.toString()

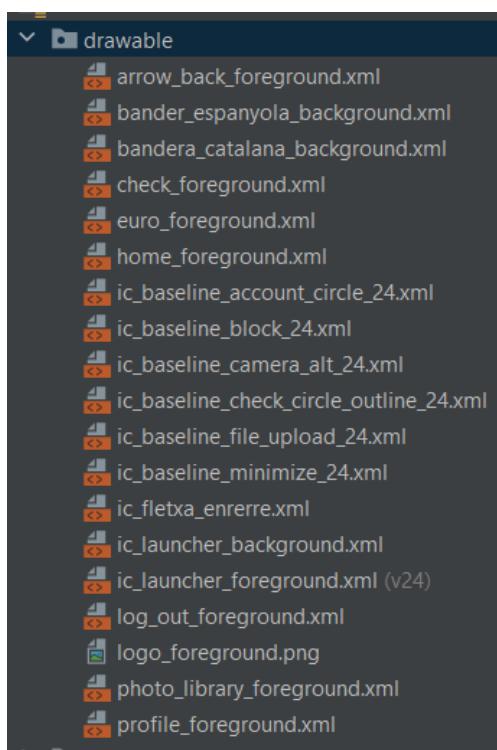
recyclerView = binding.recyclerviewResumComanda

recyclerView.layoutManager = LinearLayoutManager(this.context)
recyclerView.setHasFixedSize(true)
productelist = arrayListOf()
resumComandaAdapt = resum_comanda_adapter(productelist)
eventChangeListener()

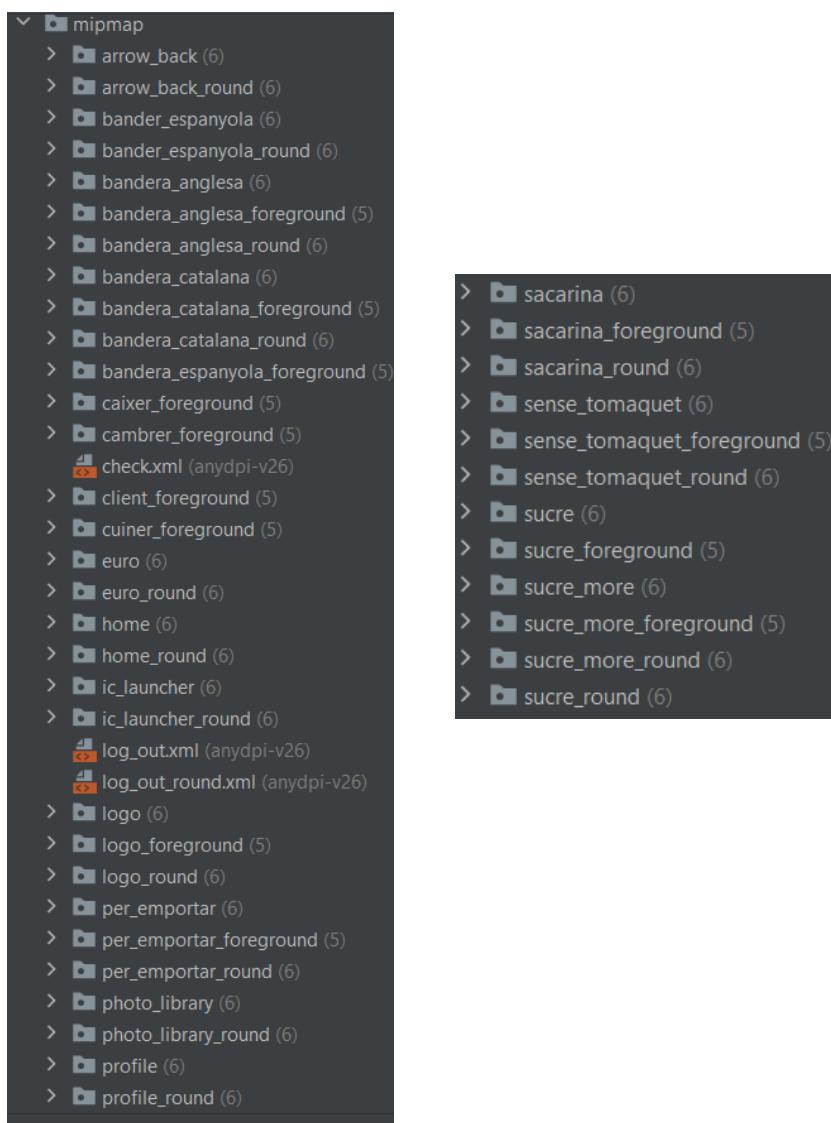
binding.btnPagat.setOnClickListener { view: View ->
    db.collection("comandes").document(documentoId).get().addOnSuccessListener {
        if (dc.get("preparat").toString() == "true") {
            db.collection("comandes").document(documentoId).update(
                hashMapOf(
                    "comandaPagada" to true,
                    "visible" to false
                ) as Map<String, Any>
            )
        } else {
            db.collection("comandes").document(documentoId).update(
                hashMapOf(
                    "comandaPagada" to true
                ) as Map<String, Any>
            )
        }
        view?.findNavController()?.navigate(PantallaResumComandesDirections.actionPantallaResumComandasToPantallaResumComandas)
    }
}

```

5.2.3 Drawables



5.2.4 Mipmap

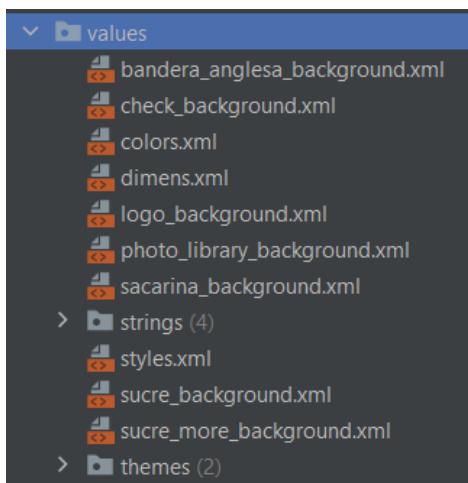


5.2.5 Menu

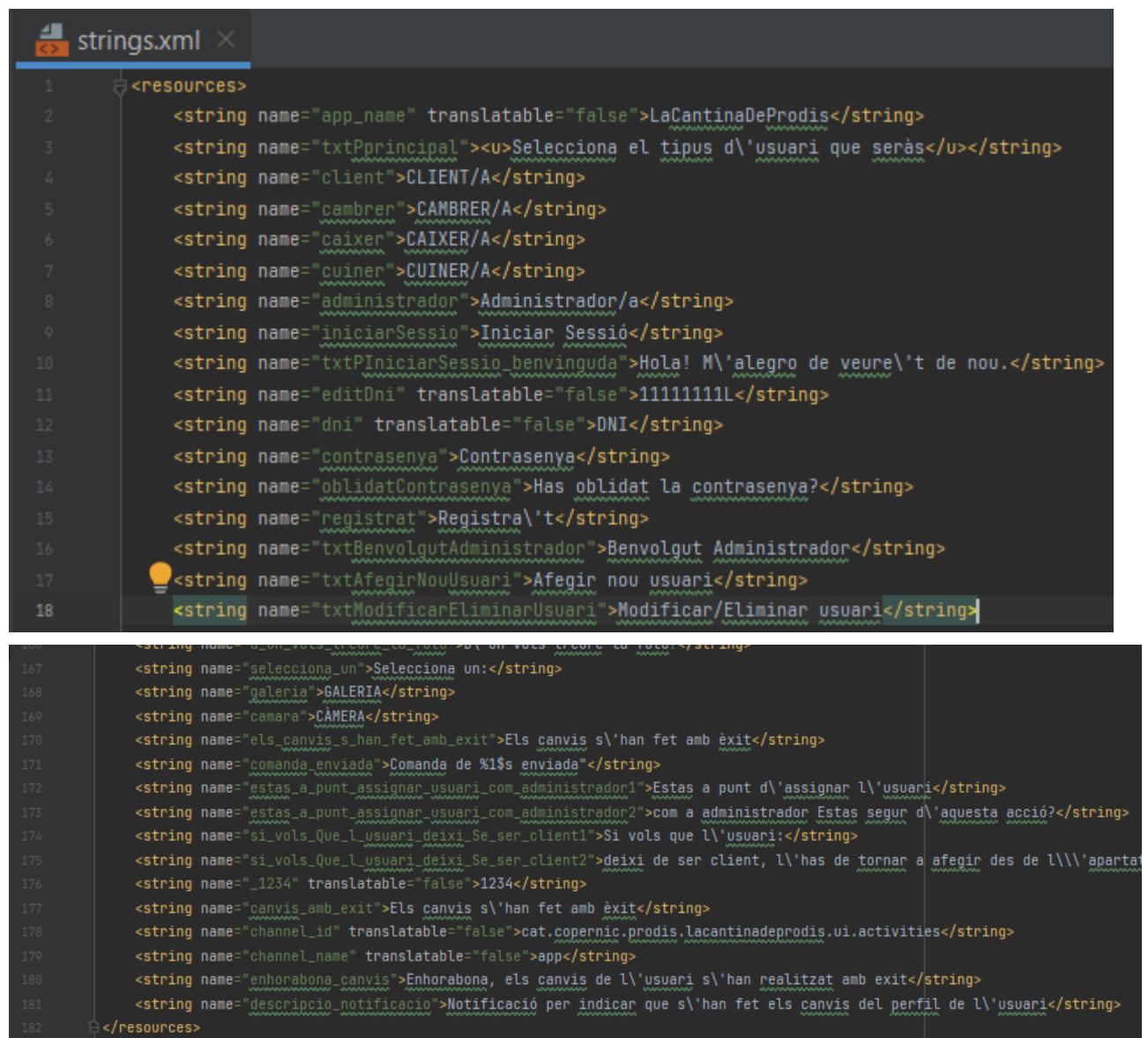
En el menú trobem la icona per tornar endarrere, icona per tornar a la pantalla principal d'aquell tipus d'usuari, icona per accedir al perfil de l'usuari i la icona per tancar sessió.



5.2.6 Values



5.2.6.1 Strings



```

<resources>
    <string name="app_name" translatable="false">LaCantinaDeProdis</string>
    <string name="txtPprincipal"><u>Selecciona el tipus d\'usuari que seràs</u></string>
    <string name="client">CLIENT/A</string>
    <string name="cambren">CAMBRER/A</string>
    <string name="caixer">CAIXER/A</string>
    <string name="cuiner">CUINER/A</string>
    <string name="administrador">Administrador/a</string>
    <string name="iniciarSessio">Iniciar Sessió</string>
    <string name="txtPINiciarSessio_benvinguda">Hola! M\'alegro de veure\'t de nou.</string>
    <string name="editDni" translatable="false">11111111L</string>
    <string name="dni" translatable="false">DNI</string>
    <string name="contrasenya">Contrasenya</string>
    <string name="oblidatContrasenya">Has oblidat la contrasenya?</string>
    <string name="registrat">Registra\'t</string>
    <string name="txtBenvolgutAdministrador">Benvolgut Administrador</string>
    <string name="txtAfegirNouUsuari">Afegir nou usuari</string>
    <string name="txtModificarEliminarUsuari">Modificar/Eliminar usuari</string>

    <string name="d_oh_vols_modificar_tu_perfil">D\'oh vols modificar tu perfil?</string>
    <string name="selecciona_un">Selecciona un:</string>
    <string name="galeria">GALERIA</string>
    <string name="camara">CAMERA</string>
    <string name="els_canvis_s_han_fet_amb_exit">Els canvis s\'han fet amb èxit</string>
    <string name="comanda_enviada">Comanda de %1$s enviada</string>
    <string name="estas_a_punt_assignar_usuari_com_administrador1">Estas a punt d\'assignar l\'usuari</string>
    <string name="estas_a_punt_assignar_usuari_com_administrador2">com a administrador Estas segur d\'aquesta acció?</string>
    <string name="si_vols_Quer_l_usuari_deixi_Se_ser_client1">Si vols que l\'usuari:</string>
    <string name="si_vols_Quer_l_usuari_deixi_Se_ser_client2">deixi de ser client, l\'has de tornar a afegir des de l\'apartat</string>
    <string name="1234" translatable="false">1234</string>
    <string name="canvis_amb_exit">Els canvis s\'han fet amb èxit</string>
    <string name="channel_id" translatable="false">cat.copernic.prodis.lacantinadeprodis.ui.activities</string>
    <string name="channel_name" translatable="false">app</string>
    <string name="enhorabona_canvis">Enhorabona, els canvis de l\'usuari s\'han realitzat amb exit</string>
    <string name="descripcio_notificacio">Notificació per indicar que s\'han fet els canvis del perfil de l\'usuari</string>
</resources>

```

5.2.6.3 Colors



```

<color name="purple_200">#FFBB86FC</color>
<color name="purple_500">#FF6200EE</color>
<color name="purple_700">#FF3700B3</color>
<color name="teal_200">#FF03DAC5</color>
<color name="teal_700">#FF018786</color>
<color name="black">#FF000000</color>
<color name="white">#FFFFFF</color>
    <color name="colorPrimary">#008570</color>
    <color name="colorPrimaryDark">#005745</color>
    <color name="colorLight">#4cb59e</color>
    <color name="colorFons">#fff1d7</color>

```

5.2.6.3 Styles

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorLight</item>
    </style>
    <style name="AppTheme.ButtonPrincipal">
        <item name="backgroundTint">@color/colorPrimary</item>
        <item name="android:textColor">@color/white</item>
        <item name="android:textSize">@dimen/txtTitol</item>
        <item name="fontFamily">sans-serif-medium</item>
    </style>
    <style name="AppTheme.ButtonAdmin">
        <item name="backgroundTint">@color/colorPrimary</item>
        <item name="android:textSize">@dimen/txtSubTitol</item>
        <item name="android:textColor">@color/white</item>
    </style>
    <style name="AppTheme.btnGuardar">
        <item name="backgroundTint">@color/colorPrimary</item>
        <item name="android:textSize">@dimen/txtPetit</item>
        <item name="android:textColor">@color/white</item>
    </style>
    <style name="AppTheme.btnContinuar">
        <item name="backgroundTint">@color/colorPrimary</item>
        <item name="android:textSize">@dimen/txtPetit</item>
        <item name="android:textColor">@color/white</item>
    </style>
    <style name="AppTheme.ImgUsuari">
        <item name="android:layout_width">120dp</item>
        <item name="android:layout_height">122dp</item>
    </style>
    <style name="AppTheme.TextView">
        <item name="android:layout_width">wrap_content</item>
        <item name="android:layout_height">wrap_content</item>
    </style>
</resources>
```

5.2.6.4 Dimensions

```
<resources>
    <dimen name="txtTitol">30sp</dimen>
    <dimen name="txtSubTitol">20sp</dimen>
    <dimen name="txtGran">18sp</dimen>
    <dimen name="txtMig">16sp</dimen>
    <dimen name="txtPetit">14sp</dimen>
    <dimen name="txtTitolTablet">40sp</dimen>
    <dimen name="txtSubTitolTablet">30sp</dimen>
    <dimen name="txtGranTablet">28sp</dimen>
    <dimen name="txtMigTablet">26sp</dimen>
    <dimen name="txtPetitTablet">24sp</dimen>
    <dimen name="centrat">60dp</dimen>
    <dimen name="marginLeft">160dp</dimen>
    <dimen name="marginRight">40dp</dimen>
    <dimen name="marginLeft2">40dp</dimen>
    <dimen name="tamanyFoto">140dp</dimen>
    <dimen name="tamanyFotoPetit">100dp</dimen>
    <dimen name="checkWidth">300dp</dimen>
    <dimen name="checkHeight">60dp</dimen>
    <dimen name="imgBossaWidth">240dp</dimen>
    <dimen name="imgBossaHeight">200dp</dimen>
    <dimen name="marginLeftAdminLandscape">160dp</dimen>
    <dimen name="paddingLeftPantallaPrincipal">440dp</dimen>
    <dimen name="paddingRightPantallaPrincipal">590dp</dimen>
    <dimen name="contrassenyaWidth">204dp</dimen>
    <dimen name="marginAtributsButtons">600dp</dimen>
    <dimen name="marginLandscape">160dp</dimen>
    <dimen name="adminButtonsWidth">370dp</dimen>
</resources>
```

5.2.7 Gradle

5.2.7.1 Gradle Project

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.
buildscript {
    repositories {
        google()
        mavenCentral()
    }
    dependencies {
        classpath "com.android.tools.build:gradle:7.0.3"
        classpath 'org.jetbrains.kotlin:kotlin-gradle-plugin:1.6.10'
        classpath "androidx.navigation:navigation-safe-args-gradle-plugin:2.3.5"
        classpath 'com.google.gms:google-services:4.3.10'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

5.2.7.1 Gradle Module

```
plugins {
    id 'com.android.application'
    id 'kotlin-android'
    id 'androidx.navigation.safeargs'
    id 'com.google.gms.google-services'
}

android {
    compileSdk 31

    defaultConfig {
        applicationId "cat.copernic.prodis.lacantinadeprodis"
        minSdk 21
        targetSdk 31
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    buildFeatures {
        dataBinding true
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = '1.8'
    }
}
```

```
dependencies {

    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.appcompat:appcompat:1.4.0'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.2'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'androidx.navigation:navigation-fragment-ktx:2.3.5'
    implementation 'androidx.navigation:navigation-ui-ktx:2.3.5'

    //ViewModel
    implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.4.0'

    //RecyclerView
    implementation 'androidx.recyclerview:recyclerview:1.2.1'

    // Firebase
    implementation platform('com.google.firebaseio:firebase-bom:29.0.0')
    implementation 'com.google.firebaseio:firebase-analytics-ktx'
    implementation 'com.google.firebaseio:firebase-auth-ktx'
    implementation 'com.google.firebaseio:firebase-firebase-store-ktx'
    implementation 'com.google.firebaseio:firebase-storage-ktx'
    implementation 'androidx.test:core-ktx:1.4.0'

    //Firestorage
    implementation 'com.github.bumptech.glide:glide:4.12.0'
    implementation 'androidx.gridlayout:gridlayout:1.0.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'

    //Coroutines
    implementation("org.jetbrains.kotlinx:kotlinx-coroutines-android:1.5.0")

    testImplementation 'junit:junit:|'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'

}
```

5.3 Codificació

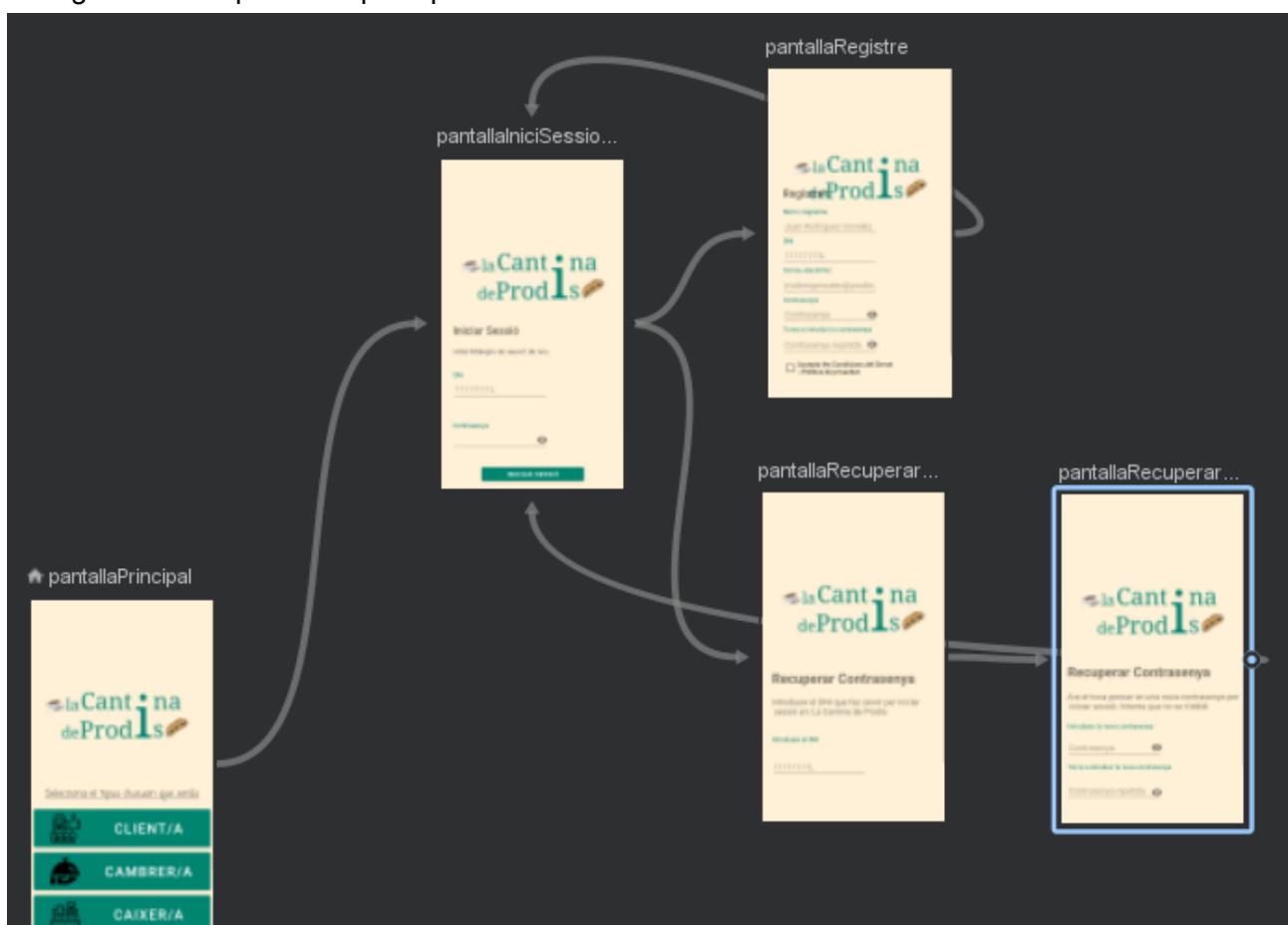
Enllaç al repositori GitLab del projecte:

<https://docs.google.com/document/d/1uEUfKwgfdMfoFg8n0v3Y8Hpus17H-zOXk9Cn216zkII/edit?usp=sharing>

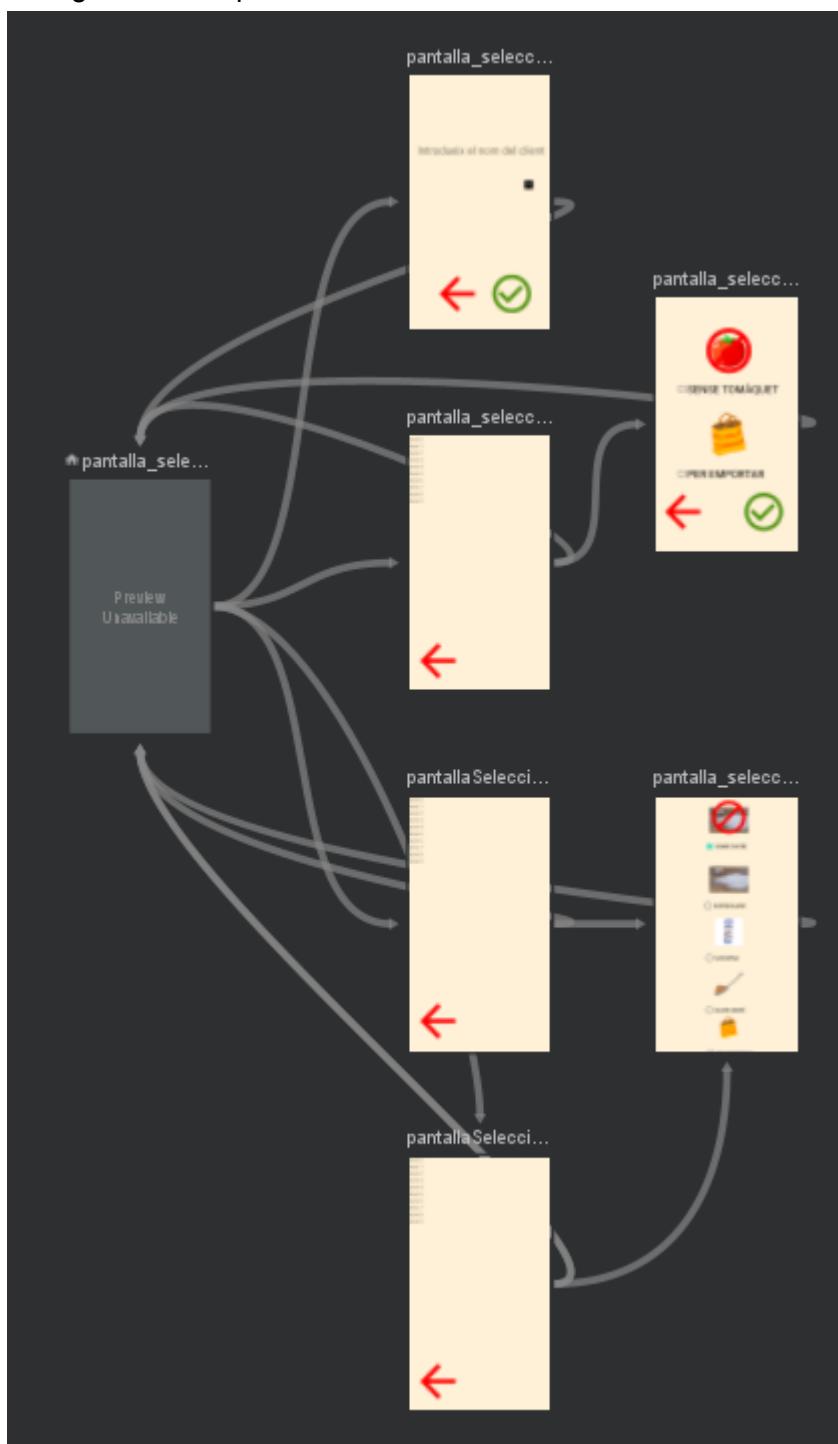
5.3.1 Navegabilitat

Per a la nostra codificació hem pensat a fer un fitxer de navegació i una activity per cada tipus d'usuari.

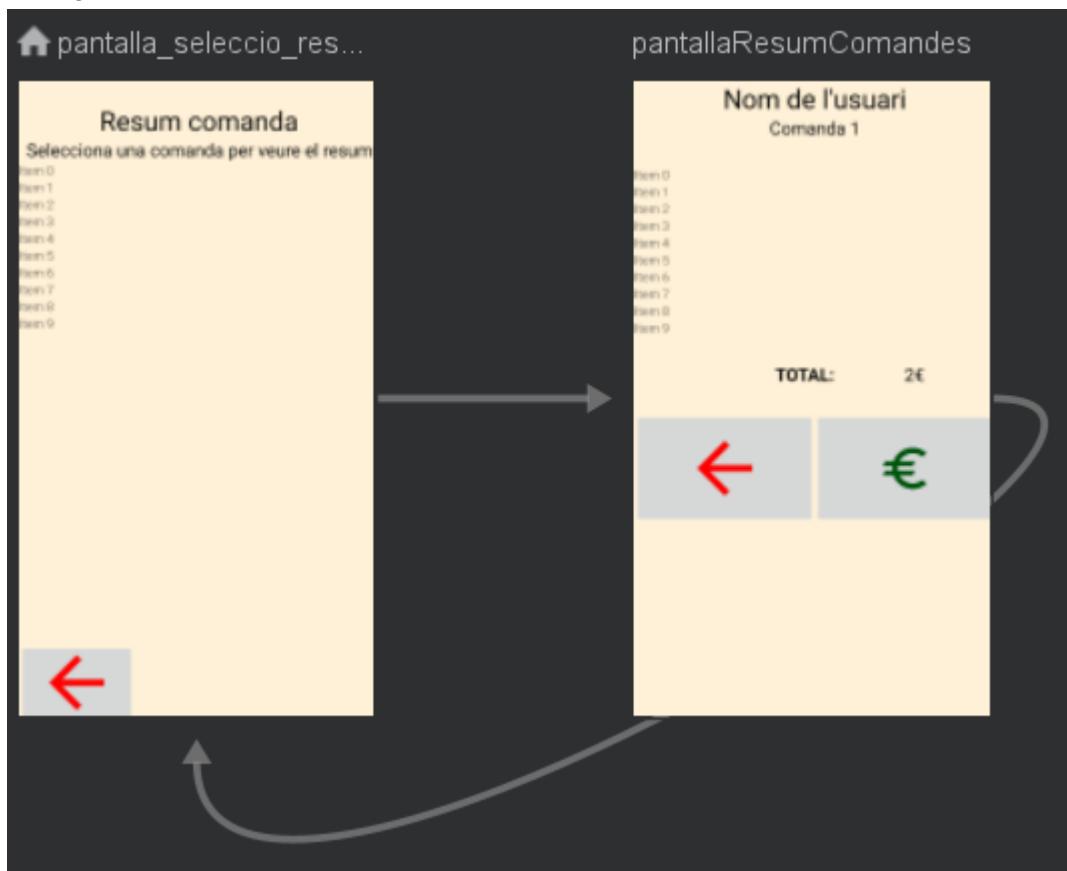
Navegació de les pantalles principals:



Navegació de les pantalles de client i cambrer:



Navegació de les pantalles de caixa:



Navegació de les pantalles d'administrador:



5.3.3 Base de dades Firebase

Firebase es tracta d'una plataforma mòbil creada per Google, la principal funció de la qual és desenvolupar i facilitar la creació d'apps d'elevada qualitat d'una forma ràpida, amb la finalitat que es pugui augmentar la base d'usuaris i guanyar més diners. La plataforma està pujada en el núvol i està disponible per a diferents plataformes com iOS, Android i web. Conté diverses funcions perquè qualsevol desenvolupador pugui combinar i adaptar la plataforma a mesura de les seves necessitats.

➤ **Authentication**

Firebase Authentication proporciona serveis de backend, SDK fàcils d'utilitzar i biblioteques de UI ja elaborades per autenticar als usuaris en la nostra aplicació. Admet l'autenticació mitjançant contrasenyes, números de telèfon, proveïdors d'identitat federada populars, com Google, Facebook, Twitter i molt més.

➤ **Cloud Firestore**

Cloud Firestore és la base de dades més recent de Firebase per al desenvolupament d'apps per a dispositius mòbils. Aprofita el millor de Realtime Database amb un model de dades nou i més intuïtiu. Amb Cloud Firestore també es poden realitzar consultes més riques i ràpides, i l'escalament s'ajusta a un nivell més alt que Realtime Database.

➤ **Cloud Storage**

Cloud Storage per a Firebase és un servei d'emmagatzematge d'objectes potent, simple i rendible construït per a l'escalament de Google. Els SDK de Firebase per a Cloud Storage agreguen la seguretat de Google a les operacions de càrrega i descàrrega d'arxius de les nostres apps de Firebase, sense importar la qualitat de la xarxa.

Pots utilitzar els seus SDK per a emmagatzemar imatges, àudio, vídeo i altres tipus de contingut generat per l'usuari. En el servidor, pots usar les API de Google Cloud Storage per a accedir als mateixos arxius.

➤ **Cloud Messaging**

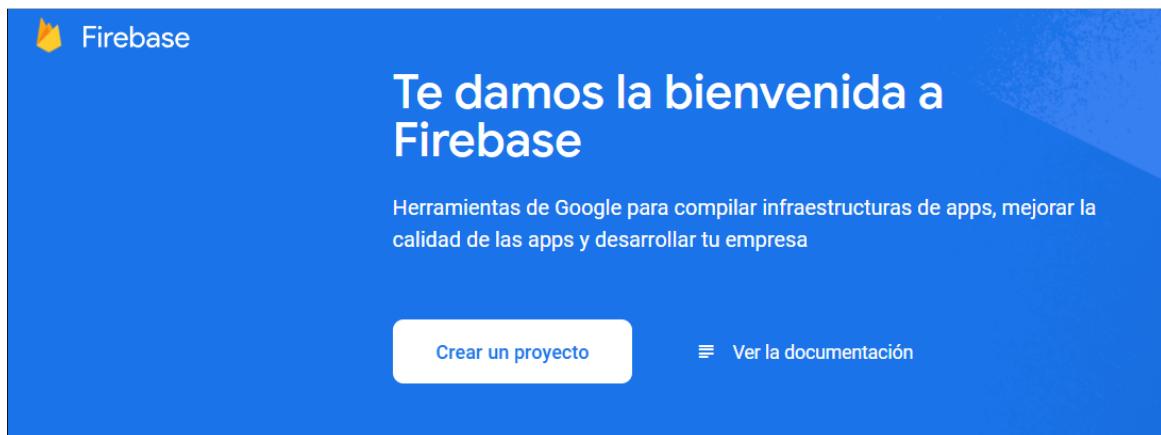
Firebase Cloud Messaging (FCM) és una solució de missatgeria multiplataforma que et permet enviar missatges de manera segura i gratuïta.

Amb FCM, pots notificar a una app client que un correu electrònic nou o que altres dades estan disponibles per a la sincronització. Pots trametre missatges de notificació per a tornar a atreure a més usuaris i augmentar la seva retenció. Per als casos d'ús de missatgeria instantània, un missatge pot transferir una càrrega útil de fins a 4.000 bytes a una app client.

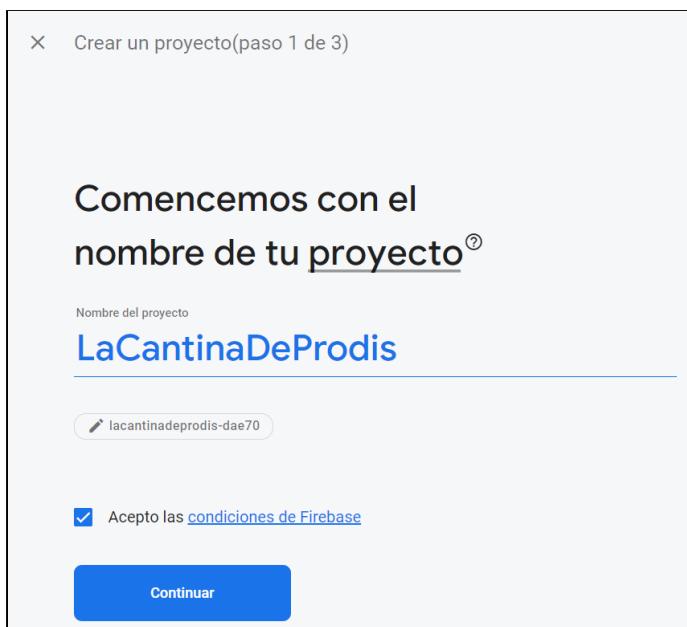
5.3.3.0 Creació d'un projecte Firebase

Per la creació de la base de dades amb Firebase vam introduir l'adreça web: <https://firebase.google.com/>.

En accedir a la pàgina web, ens trobem amb la següent capçalera, per crear el projecte va ser tan fàcil com fer clic en el botó “*Crear un proyecto*”.



El primer que ens vam trobar, va ser que ens demana el nom del projecte, en el nostre cas **LaCantinaDeProdis**, Firebase automàticament ens assigna un identificador únic de projecte. Finalment acceptem les condicions d'ús i cliquem en Continuar.

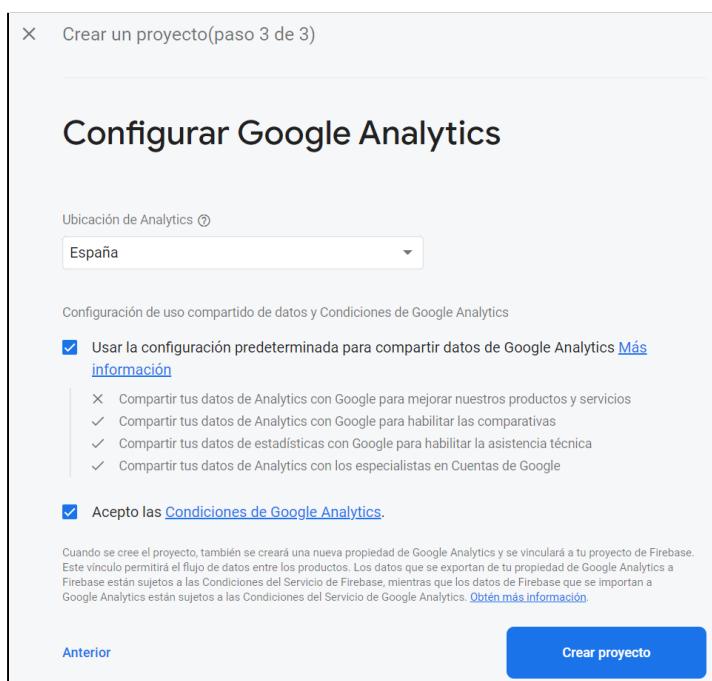
A screenshot of the first step of the Firebase project creation process. The title is "Crear un proyecto (paso 1 de 3)". The main instruction is "Comencemos con el nombre de tu proyecto". A text input field contains "LaCantinaDeProdis". Below it is a note: "Nombre del proyecto" followed by the generated ID "lacantinadeprodis-dae70". A checkbox is checked with the text "Acepto las condiciones de Firebase". A blue "Continuar" button is at the bottom.

A continuació habilitem el servei [Google Analytics](#) per tal de tenir coneixement de l'impacte de la nostra aplicació, nombre d'usuaris, connexions que es fan a la base de dades, etc. Un cop seleccionat, fem clic en *Continuar*



A l'habilitar Google Analytics, ens apareix la següent pantalla, on seleccionem la ubicació del servidor Analytics, en el nostre cas “*España*”.

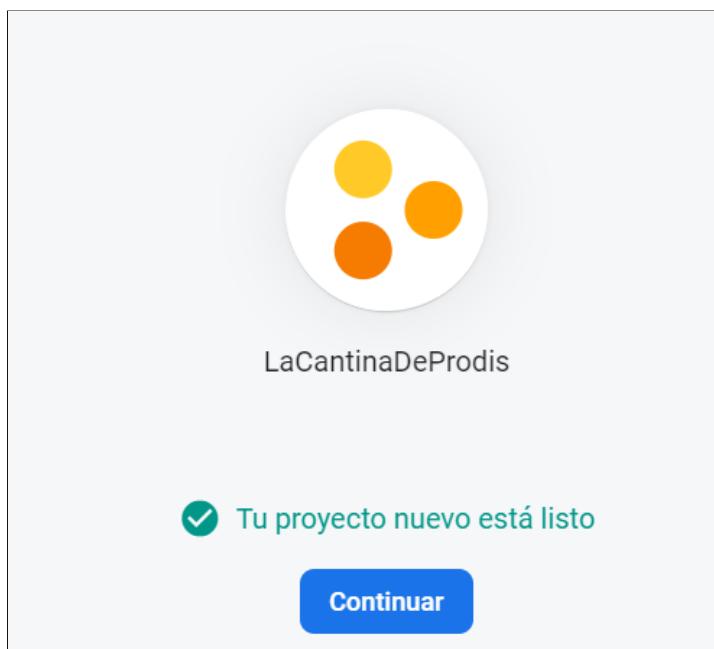
Seleccionem l'opció “*Usar la configuración predeterminada para compartir datos de Google Analytics*” i acceptem les condicions de Google Analytics. Finalment fem clic en “*Crear proyecto*”



Un cop hagim fet el pas anterior, es començà a crear el nostre projecte Firebase.



Un cop Firebase ja l'hagi creat, se'ns mostrarà el següent missatge notificant-nos que el projecte ja està llest. Per accedir a ell, fem clic en "*Continuar*".



Un cop fet el pas anterior se'ns mostra la següent pantalla amb el projecte ja creat.



Firebase

Descripción general del Proyecto

Compilación

Lanzamiento y supervisión

Analytics

Participación

Extensions

Spark

Ir a la documentación

Registrarse

Recibe notificaciones por correo electrónico sobre nuevas funciones de Firebase, investigaciones y eventos

LaCantinaDeProdis Plan Spark

Comienza por agregar Firebase a tu app

iOS Android JavaScript Cloud Functions

Agrega una app para comenzar

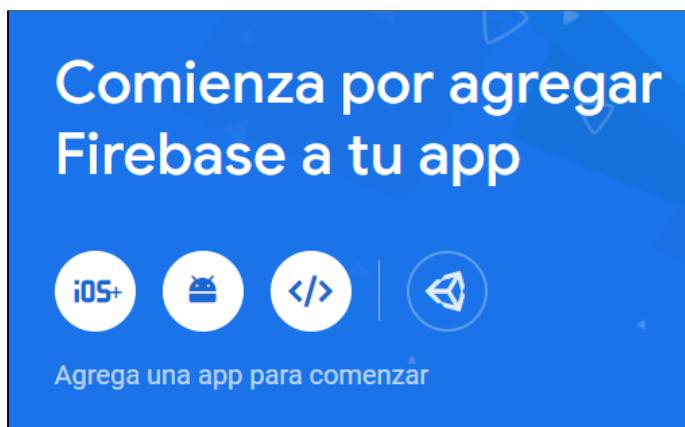
Almacena y sincroniza datos de app en milisegundos

A screenshot of the Firebase console dashboard for the project "LaCantinaDeProdis". The dashboard features a central banner with the text "Comienza por agregar Firebase a tu app" and icons for iOS, Android, JavaScript, and Cloud Functions. Below the banner, there's a call-to-action button "Agrega una app para comenzar" and a note about real-time synchronization. The left sidebar lists various Firebase services like Compilación, Lanzamiento y supervisión, Analytics, etc.

5.3.3.1 Vinculació del projecte amb la app

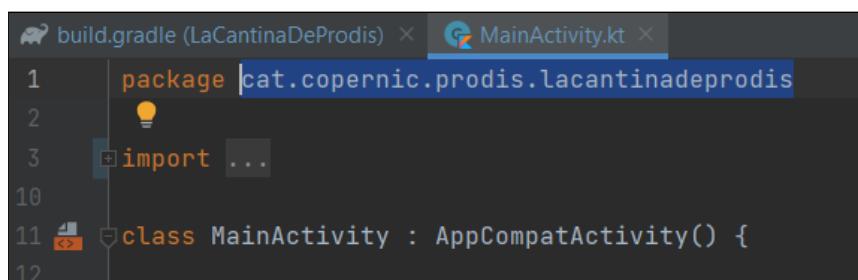
Ara és el moment d'afegir la nostra aplicació al projecte de Firebase.

En el nostre cas estem desenvolupant una aplicació Android, per tant, farem clic sobre la icona d'Android (la del mig).



A continuació registrarem la nostra app, primer de tot introduïm el nom del paquet de la nostra aplicació.

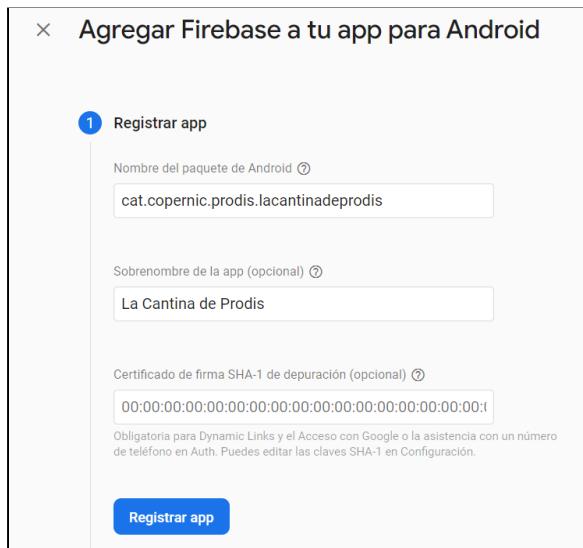
Anem a Android Studio ens posem dins de qualsevol fitxer, en el nostre cas el MainActivity i copiarem el nom del paquet (*Package*).



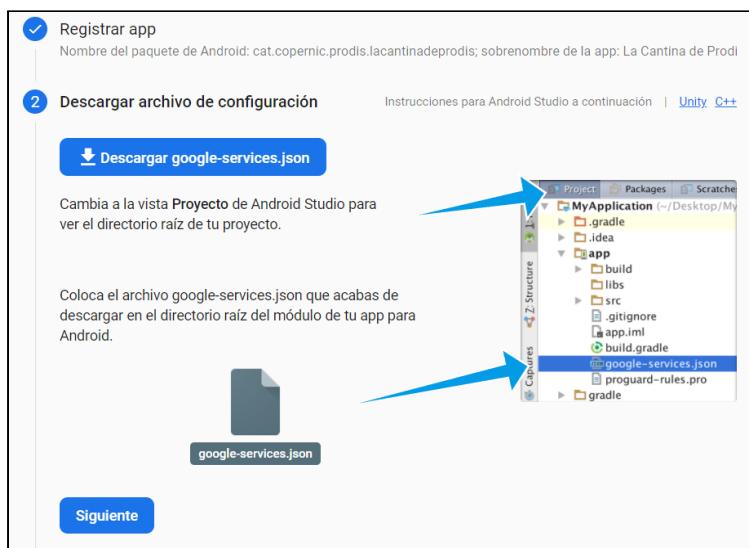
```
build.gradle (LaCantinaDeProdis) × MainActivity.kt ×
1 package cat.copernic.prodis.lacantinadeprodis
2
3 import ...
10
11 class MainActivity : AppCompatActivity() {
12 }
```

I l'enganyarem en el primer espai de text. Opcionalment podrem afegir un sobrenom, nosaltres hem decidit sí posar-li i li hem posat "La Cantina de Prodis".

El certificat SHA1 el deixarem en blanc i clicarem en "Registrar app".



Un cop registrada l'app, ens haurem de descarregar el fitxer "google-services.json"



I col·locar-lo la carpeta "app" del projecte.

LaCantinaDeProdis > app			
	Nombre	Fecha de modificación	Tipo
build	build	29/10/2021 9:44	Carpeta de archivos
libs	libs	26/10/2021 10:35	Carpeta de archivos
src	src	03/11/2021 12:39	Carpeta de archivos
.gitignore	.gitignore	26/10/2021 10:35	Documento de tex...
build.gradle	build.gradle	08/11/2021 13:01	Archivo de origen ...
google-services.json	google-services.json	10/11/2021 12:55	Archivo de origen ...
proguard-rules.pro	proguard-rules.pro	26/10/2021 10:35	Archivo PRO

Un cop descarregat el JSON hem d'introduir en el build.gradle(project) els següents requisits

```
buildscript {
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
    dependencies {
        ...
        // Add this line
        classpath 'com.google.gms:google-services:4.3.10'
    }
}
```

A continuació fem un "Sync Now" perquè s'apliquin els repositoris i les "classpath" noves.

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

Sync Now

```
1 // Top-level build file where you can add configuration options common to all sub-projects/modules.
2
3 buildscript {
4     repositories {
5         google()
6         mavenCentral()
7     }
8     dependencies {
9         classpath "com.android.tools.build:gradle:7.0.3"
10        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:1.5.31"
11        classpath 'com.google.gms:google-services:4.3.10'
12
13        // NOTE: Do not place your application dependencies here; they belong
14        // in the individual module build.gradle files
15    }
}
```

També haurem d'afegir nous plugins i implementacions en el build.gradle, però ara en el de l'app, el build.gradle(Module)

Archivo build.gradle de nivel de app (<project>/<app-module>/build.gradle):

```

apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'

dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:29.0.0')

    // Add the dependency for the Firebase SDK for Google Analytics
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation 'com.google.firebase:firebase-analytics-ktx'

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}

```

Si usas la BoM de Firebase para Android, tu app siempre utilizará versiones compatibles de la biblioteca de Firebase.
[Más información](#)

Introduïm els nous plugins

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

```

1   plugins {
2       id 'com.android.application'
3       id 'kotlin-android'
4       id 'com.google.gms.google-services'
5   }
6

```

Afegim les noves implementacions

```

37
38     dependencies {
39
40         implementation 'androidx.core:core-ktx:1.7.0'
41         implementation 'androidx.appcompat:appcompat:1.3.1'
42         implementation 'com.google.android.material:material:1.4.0'
43         implementation 'androidx.constraintlayout:constraintlayout:2.1.1'
44         implementation 'androidx.legacy:legacy-support-v4:1.0.0'
45         implementation 'androidx.navigation:navigation-fragment-ktx:2.3.5'
46         implementation 'androidx.navigation:navigation-ui-ktx:2.3.5'
47         implementation platform('com.google.firebase:firebase-bom:29.0.0')
48         implementation 'com.google.firebase:firebase-analytics-ktx'
49         testImplementation 'junit:junit:'
50         androidTestImplementation 'androidx.test.ext:junit:1.1.3'
51         androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
52
53     }

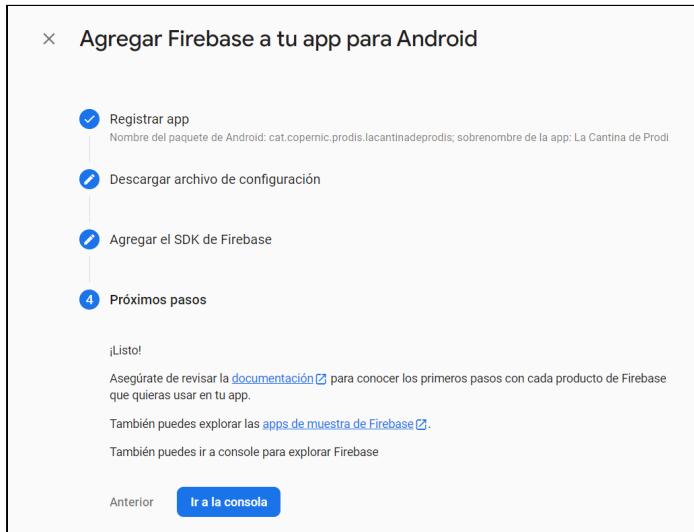
```

I igual que amb el build.gradle(Project), sincronitzem.

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

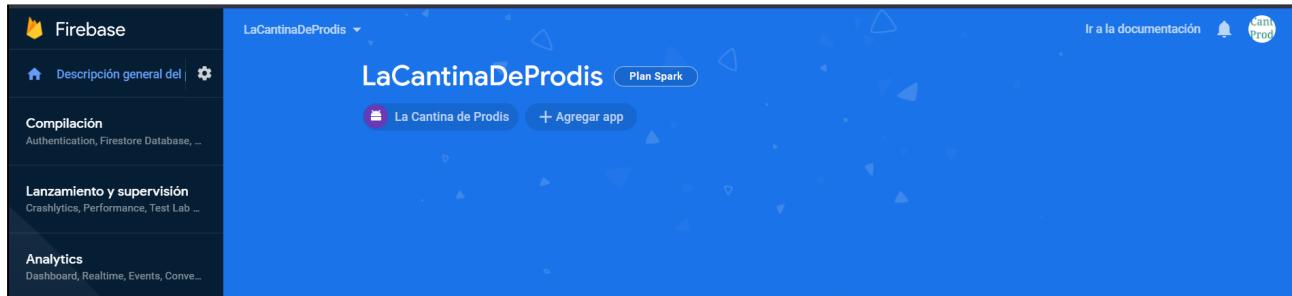
[Sync Now](#)

Ja hem acabat, un cop fets els passos anteriors, ja ho tindrem tot llist i Firebase ens dirà que ja podem anar a la consola.

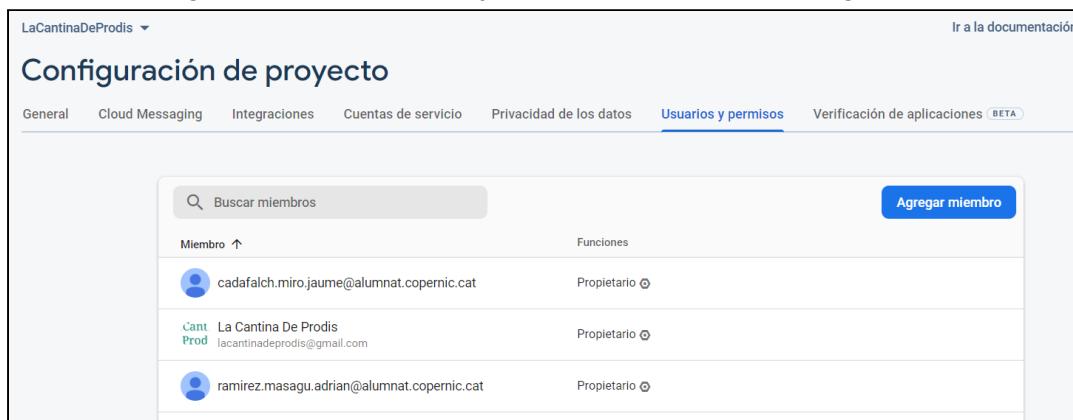


En accedir, veurem que dins el projecte LaCantinaDeProdis, tenim una aplicació anomenada:

La Cantina de Prodis.

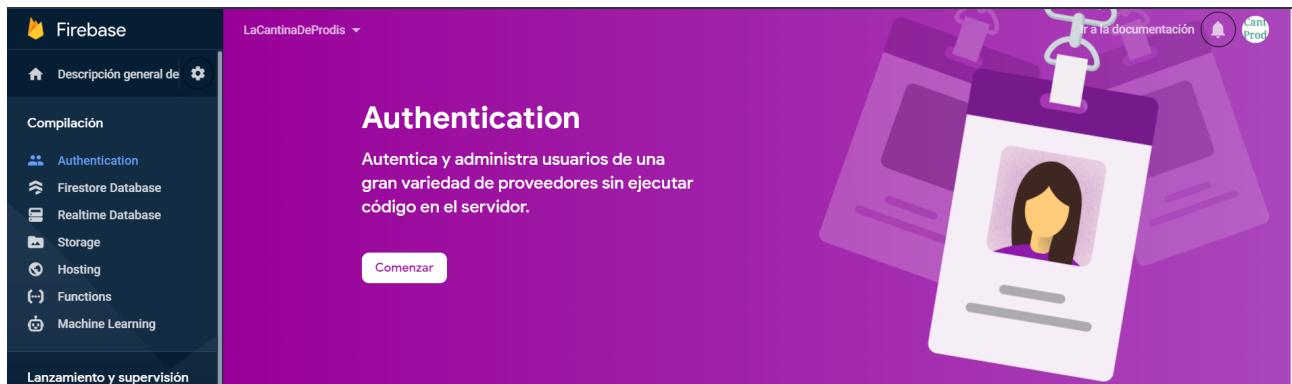


Finalment perquè tots els desenvolupadors puguin accedir i fer modificacions a la base de dades, els afegim com a propietaris, ja que ens interessa que tinguin tots els permisos.

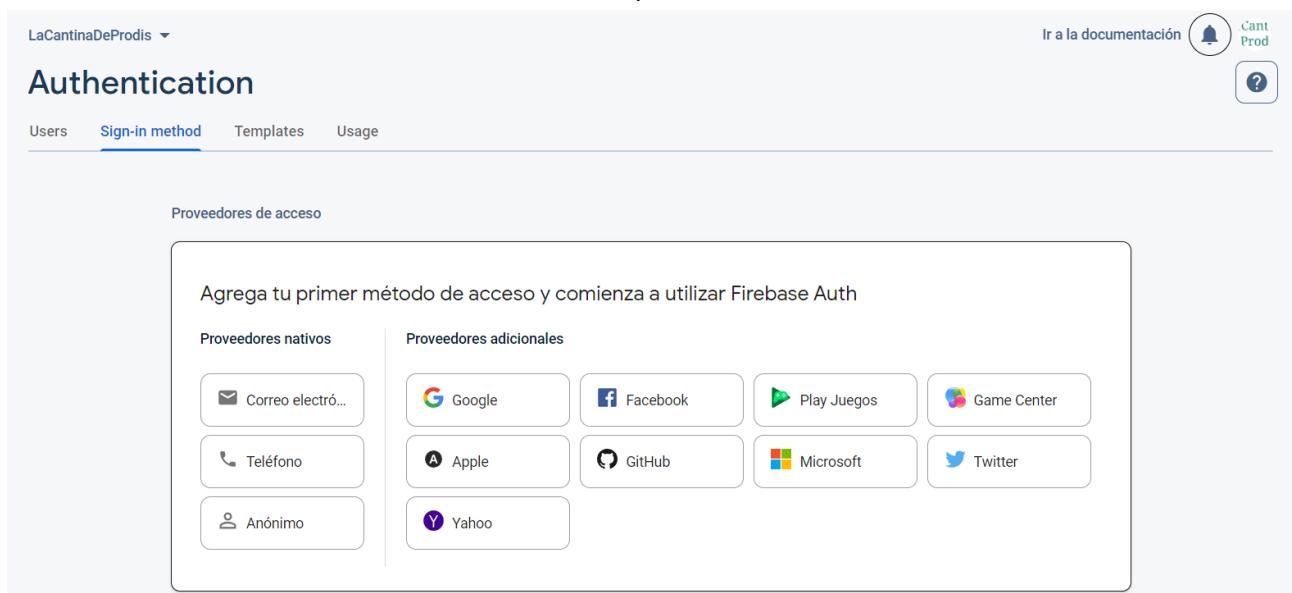


5.3.3.2 Autenticació amb Firebase Authentication

Per poder utilitzar Firebase Authentication, desplegarem “Compilación” i seleccionarem “Authentication”. I seleccionarem “Comenzar”

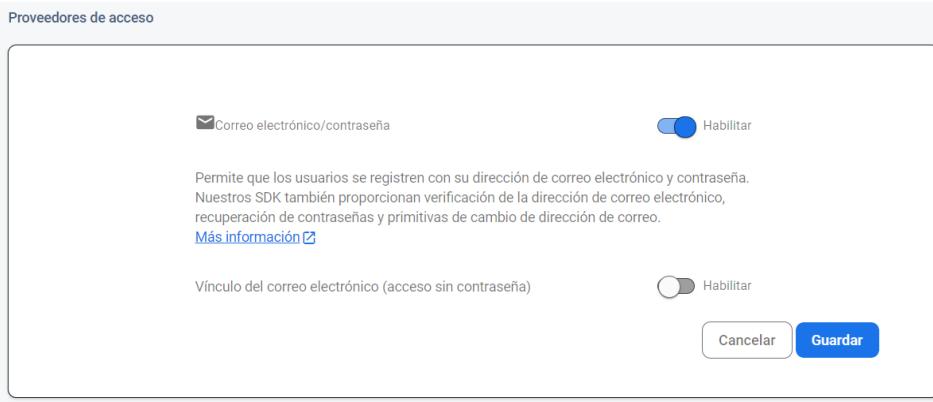


Se'ns mostraran tots els sistemes d'autenticació possibles

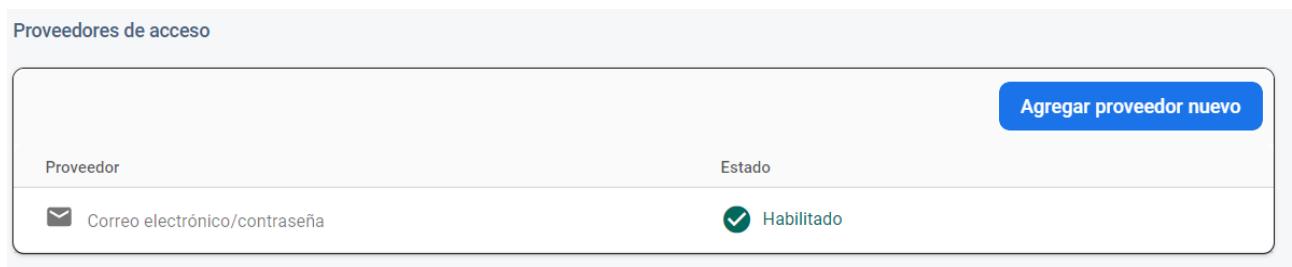


I seleccionarem el proveedor natiu “Correo electrónico”

En fer-ho habilitem l'opció “Correo electrónico/contraseña” i fem clic en Guardar



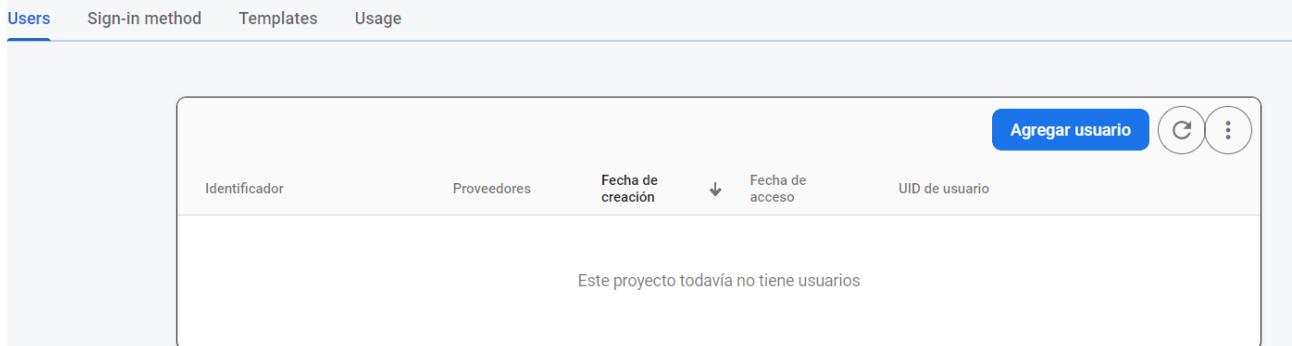
En fer-ho veurem que ja el tenim afegit



This screenshot shows the same 'Proveedores de acceso' section after saving the configuration. The 'Correo electrónico/contraseña' provider is now listed with a green checkmark indicating it is enabled ('Habilitado'). A blue 'Agregar proveedor nuevo' (Add new provider) button is visible in the top right corner.

En la secció "Users" aniran apareixent els usuaris registrats.

Authentication



This screenshot shows the 'Users' section of the Firebase Authentication console. At the top, there are tabs for 'Users' (which is selected), 'Sign-in method', 'Templates', and 'Usage'. Below the tabs is a table header with columns: 'Identificador', 'Proveedores', 'Fecha de creación', 'Fecha de acceso', and 'UID de usuario'. To the right of the table is a blue 'Agregar usuario' (Add user) button and two circular icons with 'C' and '⋮' respectively. The main table body contains the message 'Este proyecto todavía no tiene usuarios' (This project still has no users).

Finalment haurem d'afegir una dependència al nostre projecte android.

```
// Firebase
implementation platform('com.google.firebaseio:firebase-bom:29.0.0')
implementation 'com.google.firebaseio:firebase-analytics-ktx'
implementation 'com.google.firebaseio:firebase-auth-ktx'
```

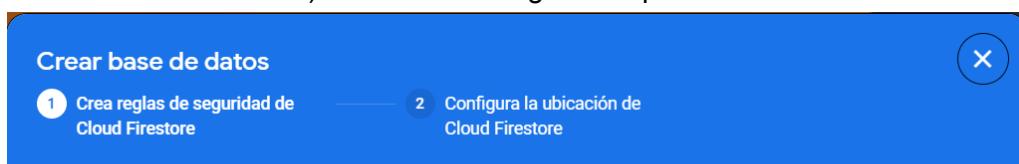
5.3.3.3 Creació base de dades Cloud Firestore

Per crear la nostra base de dades Firestore, desplegarem "Compilación" i seleccionarem "Firestore Database".

I seleccionarem: "Crear una base de datos"



Seleccionarem el mode de prova (per defecte la tindrem disponible durant 30 dies, però més endavant ho canviarem). Fem clic en “Siguiente” per continuar.



Después de definir la estructura de datos, debes crear reglas para protegerlos.

[Más información](#)

Iniciar en modo de producción
De forma predeterminada, tus datos son privados. El acceso de lectura/escritura de los clientes solo se otorgará como se indica en tus reglas de seguridad.

Comenzar en modo de prueba
Para permitir una configuración rápida, los datos se abren de forma predeterminada. Sin embargo, debes actualizar las reglas de seguridad dentro de 30 días a fin de habilitar el acceso de lectura/escritura a largo plazo para los clientes.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2021, 12, 15);
    }
  }
}
```

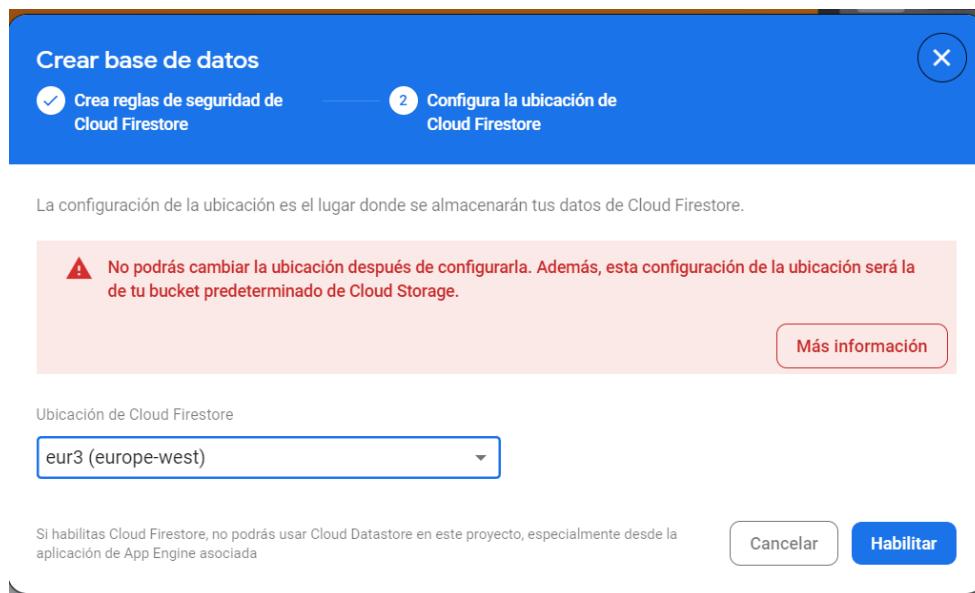
! Las reglas de seguridad predeterminadas del modo de prueba permiten que cualquier usuario con acceso a tu referencia de base de datos pueda ver, editar y borrar todos los datos durante los siguientes 30 días.

Si habilitas Cloud Firestore, no podrás usar Cloud Datastore en este proyecto, especialmente desde la aplicación de App Engine asociada

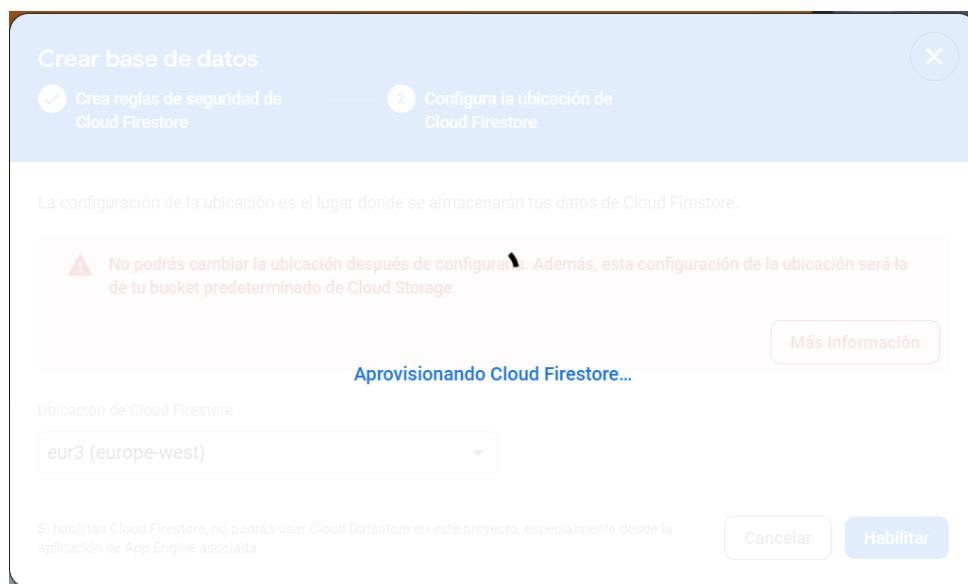
[Cancelar](#)

[Siguiente](#)

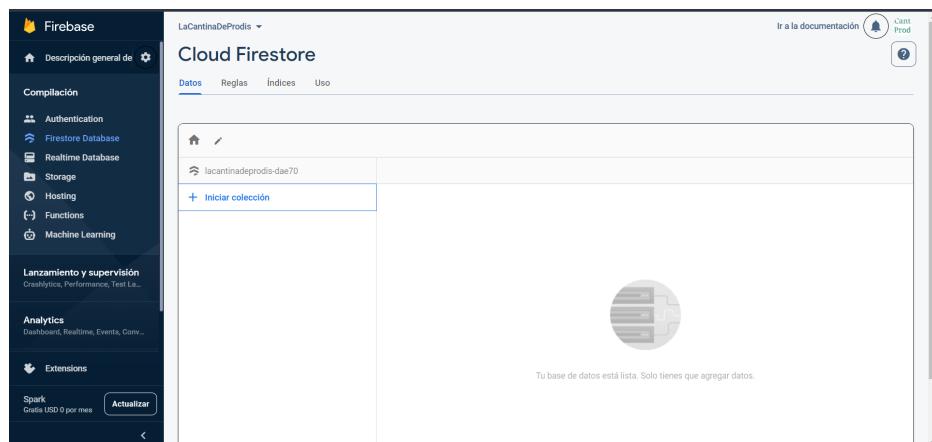
Seleccionem la ubicació de Cloud Firestore i fem clic en habilitar



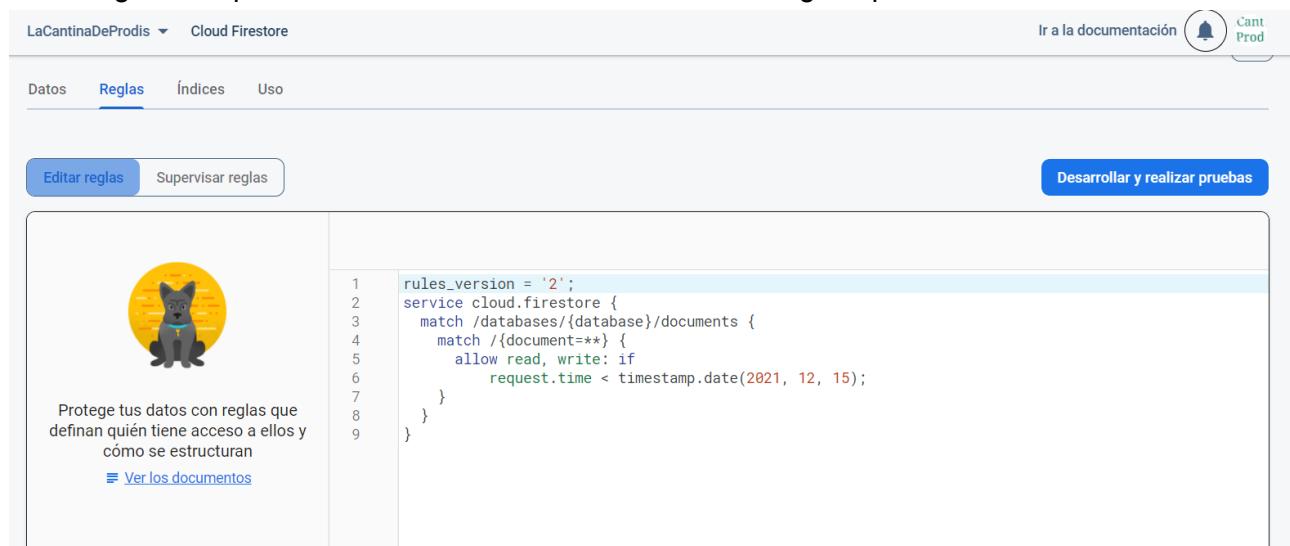
Ens esperem que es faci la creació i configuració de la nostra base de dades.



Un cop creada se'n mostrerà la nostra base de dades llena per a treballar amb ella.



Per allargar la disponibilitat de la base de dades, anem a "Reglas" per canviar la data límit.

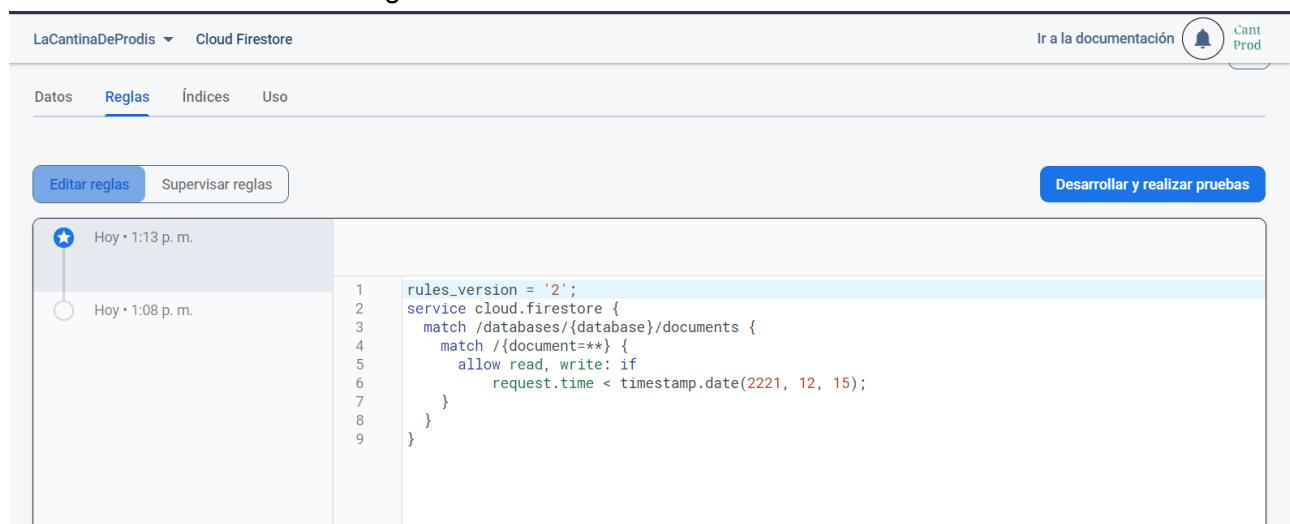


```

rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2021, 12, 15);
    }
  }
}

```

En el nostre cas ho hem allargat fins al 15-12-2221



```

rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2221, 12, 15);
    }
  }
}

```

Hoy • 1:13 p. m.

Hoy • 1:08 p. m.

Finalment afegim al nostre projecte Android lla dependència:
“com.google.firebaseio:firebase-firebase-ktx”

```
// Firebase
implementation platform('com.google.firebaseio:bom:29.0.0')
implementation 'com.google.firebaseio:firebase-analytics-ktx'
implementation 'com.google.firebaseio:firebase-auth-ktx'
implementation 'com.google.firebaseio:firebase-firebase-ktx'
```

5.3.3.4 Registre i accés a l'app

5.3.3.4.1 Pantalla Login

En el moment en què l'usuari fa clic en el botó d'iniciar sessió, es comprova que els camps DNI i contrasenya no estiguin buits i, en cas que ho estiguin apareix una alerta.

Si no estan buits es busca al Firestore el document que tingui d'id el DNI que l'usuari ha introduït, en cas que no existeix cap document, apareix una alerta.

Si es troba el document, s'agafa el camp "email" i es fa l'inici de sessió amb el valor del camp "email" i la contrasenya introduïda per l'usuari.



```

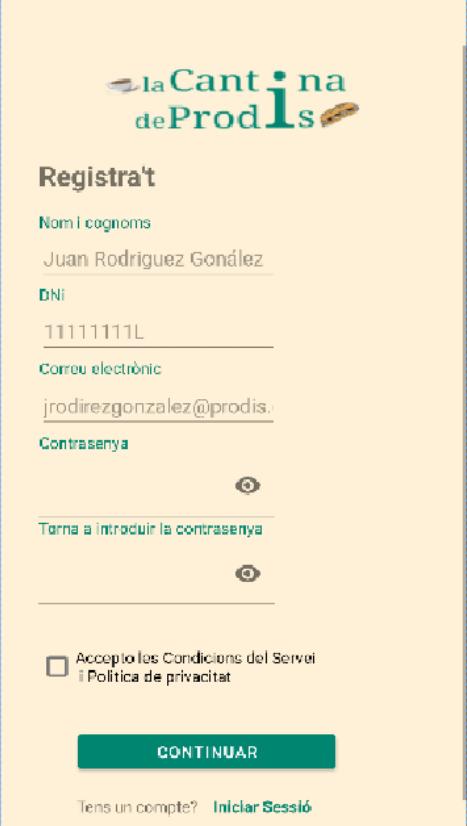
binding.btnPiniciarSessioClient.setOnClickListener { it: View!
    val dni = binding.dtTxtPIniciarSessioClientDni.text.toString().uppercase()
    var passwd = binding.dtTxtPIniciarSessioClientPassword.text.toString()
    var bool = false
    if (binding.dtTxtPIniciarSessioClientDni.text.isNotEmpty() &&
        binding.dtTxtPIniciarSessioClientPassword.text.isNotEmpty())
    ) {
        passwd += "prodis"
        db.collection( collectionPath: "users").get().addOnSuccessListener { result ->
            for (document in result) {
                if (document.id == dni) {
                    bool = true
                    auth.signInWithEmailAndPassword(
                        document.get("email").toString(),
                        passwd,
                    ).addOnCompleteListener { it: Task<AuthResult!>
                        if (it.isSuccessful) {
                            startActivity(usertype)
                        } else {
                            showAlert( message: "Error en inici de sessió")
                        }
                    }
                }
            }
            if (!bool){
                showAlert( message: "L'\u00e0usuari no est\u00e0 registrat")
            }
        }
    } else {
        Toast.makeText(this.context, text: "L'\u00e0usuari no existeix", Toast.LENGTH_SHORT).show()
    }
}

```

5.3.3.4.2 Pantalla Registre

En el moment en què l'usuari es vol registrar i clica el botó "Registra't", primer és crida una funció que comprova si les dades introduïdes són vàlides. En cas que no ho fossin, genera una alerta amb tots els errors.

Un cop les dades són vàlides, és crida a una altra funció que fa el registre l'usuari al Firebase Authentication i guarda tota la informació en un document (situat dins la col·lecció "users") que té com identificador el DNI de l'usuari. I finalment, es porta a l'usuari a la pantalla d'inici de sessió.



```

bdng.btnPregestre.setOnClickListener { view: View ->
    if (datavalids(
            bdng.dtTxtPRegistrePersonName.text.toString(),
            bdng.dtTxtPRegistreDni.text.toString(),
            bdng.dtTxtPRegistreEmail.text.toString(),
            bdng.dtTxtPRegistrePassword.text.toString(),
            bdng.dtTxtPRegistreRepeteixPassword.text.toString(),
            bdng.checkBox.isChecked
        )
    ) {
        makeregister(
            bdng.dtTxtPRegistrePersonName.text.toString(),
            bdng.dtTxtPRegistreDni.text.toString(),
            bdng.dtTxtPRegistreEmail.text.toString(),
            bdng.dtTxtPRegistrePassword.text.toString() | usertype
        )
        view.findNavController().navigate(
            PantallaRegistreDirections.actionPantallaRegistreToPantallaIniciSessioClientAdmin(
                usertype
            )
        )
    }
}

bdng.textPregistreIniciaSessio.setOnClickListener { view: View ->
    view.findNavController().navigate(
        PantallaRegistreDirections.actionPantallaRegistreToPantallaIniciSessioClientAdmin(
            usertype
        )
    )
    return bdng.root
}

```

5.3.3.4.3 Pantalla recuperar contrasenya I

En el moment que l'usuari vol recuperar la contrasenya, només ha d'introduir el seu DNI, es verificarà si l'usuari està registrat, en cas negatiu mostrarà una alerta.

Si l'usuari es troba registrat, es mostrarà la [següent pantalla](#)



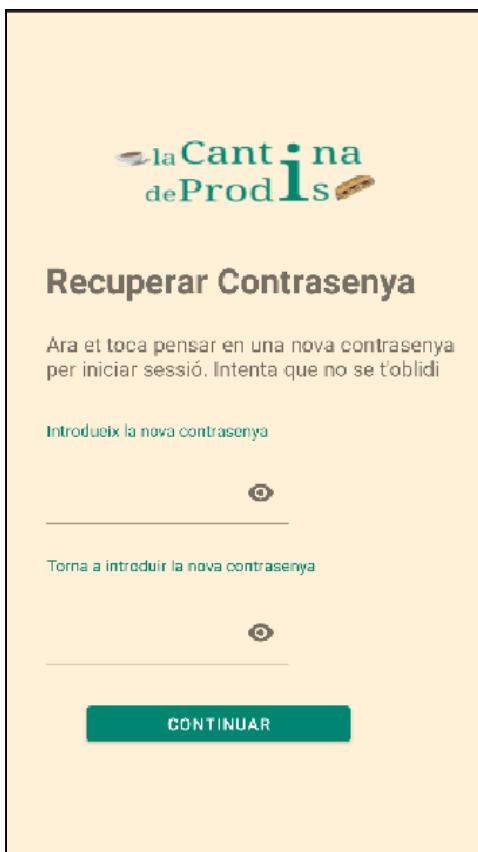
```
binding.btnPRecuperarContrasenya1Continuar.setOnClickListener { it: View! ->
    println(binding.dtTxtPRecuperarContrasenya1DNI.text.toString())
    if (binding.dtTxtPRecuperarContrasenya1DNI.text.toString().isNotEmpty()){
        db.collection( collectionPath: "users" ).get().addOnSuccessListener {result ->
            for (document in result){
                if (document.id == binding.dtTxtPRecuperarContrasenya1DNI.text.toString()) {
                    view?.findNavController()?.navigate(PantallaRecuperarContrasenya1Direction)
                }
            }
        }
    }else{
        Toast.makeText(this.context, text: "El camp DNI està buit", Toast.LENGTH_SHORT).show()
    }
}
```

5.3.3.4.4 Pantalla recuperar contrasenya II

En aquesta pantalla se li demana a l'usuari que introdueix una nova contrasenya i que la repeteixi.

Si els camps estan buits o les contrasenyes no coincideixen, es mostrerà una alerta.

Si els valors són correctes: internament, iniciarem sessió (agafant la contrasenya antiga, emmagatzemada al Firebase Firestore), agafarem l'usuari i li canviarem la contrasenya per la nova introduïda per l'usuari, posteriorment actualitzarem la contrasenya emmagatzemada al Firestore i tancarem la sessió per tal que l'usuari poguï iniciar sessió correctament.



```
private fun changePassword(dni: String, psswd: String, usertype: String) {
    var bool = false
    db.collection("users").get().addOnSuccessListener { result ->
        for (document in result) {
            if (document.id == dni) {
                bool = true
                auth.signInWithEmailAndPassword(
                    document.get("email").toString(),
                    document.get("password").toString()
                ).addOnCompleteListener { it: Task<AuthResult> ->
                    if (it.isSuccessful) {
                        val currentUser = auth.currentUser
                        currentUser?.updatePassword(psswd)?.addOnSuccessListener { it: Void! ->
                            println("SUCCESSFUL")
                            db.collection("users").document(dni).update(
                                hashMapOf(
                                    "password" to psswd
                                ) as Map<String, Any>
                            )
                            auth.signOut()
                            Toast.makeText(
                                this.context,
                                text: "S'ha canviat la contrasenya",
                                Toast.LENGTH_SHORT
                            ).show()
                        }
                    }
                }
            }
        }
    }
}
```

5.3.3.4.5 Tancar Sessió

Per tancar la sessió l'usuari únicament haurà de clicar sobre la icona de tancar sessió, situada en la appBar. En aquell moment es farà el LogOut i es tornarà a l'usuari a la pantalla d'iniciar sessió.

```
R.id.logOutBtn -> {
    FirebaseAuth.getInstance().signOut()
    finish()
    deleteComanda()
    true
}
```

5.3.3.5 Perfil

5.3.3.5.1 Pantalla mostrar perfil

En aquesta pantalla es podrà veure i modificar la informació de l'usuari.

Primer mirarà si les dades introduïdes per l'usuari són correctes. Si són correctes anirà a la base de dades i canviarà les dades antigues per les dades noves. També hi ha l'opció de canviar la foto de l'usuari.



```
binding.btnGuardar.setOnClickListener() { view: View ->
    if (dataisValids(
        binding.editTxtNom.text.toString(),
        binding.editTxtCognom.text.toString(),
        binding.editTextCorreu.text.toString()
    )) {
        db.collection( collectionPath: "users" ).document( dni ).update(
            hashMapOf(
                "username" to binding.editTxtNom.text.toString(),
                "usersusername" to binding.editTxtCognom.text.toString(),
            ) as Map<String, Any>
        )
        val currentUser =
            FirebaseAuth.getInstance().currentUser ?: return@setOnClickListener
        if (binding.editTextCorreu.text.toString() != currentUser.email) {
            currentUser
                .updateEmail( binding.editTextCorreu.text.toString() )
                .addOnSuccessListener { it: Void! -
                    db.collection( collectionPath: "users" ).document( dni ).update(
                        hashMapOf(
                            "email" to binding.editTextCorreu.text.toString()
                        ) as Map<String, Any>
                    )
                }
        }
    }
}
```

5.3.3.6 Inserció

5.3.3.6.1 Pantalla Formulari Afegir nou usuari

En la pantalla Afegir nou usuari, l'administrador pot afegir un nou usuari que no s'hagi registrat prèviament.

En aquesta pantalla, l'usuari omple uns camps en un formulari i un cop els camps introduïts són correctes, es fa una inserció a la base de dades.

```
private fun newUser(dni: String, username: String, usersurname: String) {
    db.collection( collectionPath: "users").document(dni).get().addOnSuccessListener { document ->
        if (!document.exists()) {
            db.collection( collectionPath: "users").document(dni).set(
                hashMapOf("username" to username, "usersurname" to usersurname, "dni" to dni)
            ).addOnCompleteListener { it: Task<Void> -
                if (it.isSuccessful) {
                    view?.findNavController()
                        ?.navigate(PantallaAdministradorNouUsuariDirections.actionPantallaAdministradorNouUsuariToPantallaAdministradorPrincipal())
                    Toast.makeText(
                        this.context,
                        text: "S'ha afegit correctament l'usuari $username $usersurname",
                        Toast.LENGTH_SHORT
                    ).show()
                } else {
                    showAlert( msg: "Error a l'hora de afegir el nou usuari")
                }
            }
        } else {
            showAlert( msg: "Aquest usuari ja existeix")
        }
    }
}
```

5.3.3.7 Consulta

L'app en tot moment està realitzant consultes a la base de dades, ja que quasi tot moment estem fent lectures a la BD.

```
db.collection( collectionPath: "comandes").document(documentId).collection( collectionPath: "productes").get().addOnSuccessListener
    for (document in result){
        db.collection( collectionPath: "productes").get().addOnSuccessListener { rs ->
            for (dc in rs){
                if (document.get("idProducte").toString() == dc.get("idProducte").toString()){
                    val nom = dc.get("nom").toString()
                    val preu = dc.get("preu") as Double
                    productList.add(dtclss_resum_comanda(nom, preu))
                }
            }
            recyclerView.adapter = resumComandaAdpt
        }
    }
}
```

5.3.3.8 Modificació

En el transcurs del procés de recuperar contrasenya, es realitza el canvi de contrasenya d'un usuari. Per fer el canvi de contrasenya i permetre que en un futur l'usuari la pugui canviar, la contrasenya es guarda a la base de dades i s'actualitza cada vegada que l'usuari canvia la contrasenya.

```
private fun newUser(dni: String, username: String, usersurname: String) {
    db.collection( collectionPath: "users").document(dni).get().addOnSuccessListener { document -
        if (!document.exists()) {
            db.collection( collectionPath: "users").document(dni).set(
                hashMapOf("username" to username, "usersurname" to usersurname, "dni" to dni)
            ).addOnCompleteListener { it: Task<Void!>
                if (it.isSuccessful) {
                    view?.findNavController()
                        ?.navigate(PantallaAdministradorNouUsuariDirections.actionPantallaAdministradorNouUsuariToPantallaAdministradorPrincipal())
                    Toast.makeText(
                        this.context,
                        text: "S'ha afegit correctament l'usuari $username $usersurname",
                        Toast.LENGTH_SHORT
                    ).show()
                } else {
                    showAlert( msg: "Error a l'hora de afegir el nou usuari")
                }
            }
        } else {
            showAlert( msg: "Aquest usuari ja existeix")
        }
    }
}
```

5.3.3.9 Eliminació

En la pantalla de modificar un usuari també s'ofereix l'opció d'eliminar aquell usuari. Per eliminar aquell usuari, igual que amb el canvi de contrasenya, sense que l'usuari s'adoni, iniciem sessió en el compte d'aquell usuari, obtenim la compte d'aquell usuari i eliminem el compte, finalment també eliminem totes les dades d'aquell usuari que tinguem emmagatzemades al Cloud Firestore.

```
private fun deleteUser(dni: String) {
    db.collection( collectionPath: "users").document(dni).get().addOnSuccessListener { result -
        auth.signInWithEmailAndPassword(
            result.get("email").toString(),
            result.get("password").toString()
        ).addOnCompleteListener { it: Task<AuthResult!>
            if (it.isSuccessful) {
                val currentUser = auth.currentUser
                currentUser?.delete()
                    ?.addOnCompleteListener { it: Task<Void!>
                        db.collection( collectionPath: "users").document(dni).delete()
                            .addOnSuccessListener { it: Void!
                                Toast.makeText(
                                    this.context,
                                    text: "Usuari eliminat correctament",
                                    Toast.LENGTH_SHORT
                                ).show()
                            }
                    }
            }
        }
}
```

5.3.4 Firebase Storage

5.3.4.1 Càrrega d'imatges

En aquest punt explicarem dues coses:

- Com agafar imatges del Firebase per posarles en un imageview
- Com agafar una imatge que hem fet nosaltres i pujarla al storage

5.3.4.1.1 Agafar imatges del Storage

```
//Aquesta funció agafarà l'imatge del usuari desde la base de dades
private fun agafarImatgeUsuari() {
    //Declarem la referencia del fire storage
    storageRef = FirebaseStorage.getInstance().getReference()

    //Cream una referencia a l'imatge del usuari
    var imgRef = Firebase.storage.reference.child("users/" + dni + ".jpg")

    //Agafem l'imatge i la posem en l'imatge del usuari
    imgRef.downloadUrl.addOnSuccessListener { Uri ->
        val imgUrl = Uri.toString()

        Glide.with( activity: this).load(imgUrl).into(binding.userIcon)
    }
}
```

Amb aquest codi aconseguirem agafar una imatge del nostre storage per posar de foto de perfil de l'usuari en la pantalla d'edició de perfil.

Fem una referencia al storage i li diem que agafi imatge amb el el nom del dni de l'usuari.

Aquesta imatge la carreguem al imageview de la foto de l'usuari

5.3.4.1.2 Afegir imatges al storage

Amb aquesta funció agafem una referencia al storage amb el dni de l'usuari i un bitmap de la foto que volem pujar. Amb aquestes dades podem fer que la imatge seleccionada s'afegeixi al storage.

```
fun pujarImatge(view: View) {
    // pujar imatge al Cloud Storage de Firebase
    // https://firebase.google.com/docs/storage/android/upload-files?hl=es

    // Cream una referencia amb el path i el nom de la imatge per pujar la imatge
    val pathReference = storageRef.child("users/" + dni + ".jpg")
    val bitmap =
        (binding.userIcon.drawable as BitmapDrawable).bitmap // agafem la imatge del imageView
    val baos = ByteArrayOutputStream() // declarem i inicialitzem un outputstream

    bitmap.compress(
        Bitmap.CompressFormat.JPEG,
        quality: 100,
        baos
    ) // convertim el bitmap en outputstream
    val data = baos.toByteArray() // convertim el outputstream en array de bytes.

    val uploadTask = pathReference.putBytes(data)
    uploadTask.addOnFailureListener { it: Exception
        Snackbar.make(view, getString(R.string.error_al_pujar), Snackbar.LENGTH_LONG).show()
        it.printStackTrace()
    }.addOnSuccessListener { it: UploadTask.TaskSnapshot!
        Snackbar.make(view, getString(R.string.exit_al_pujar), Snackbar.LENGTH_LONG).show()
    }
}
```

5.3.5 Dashboards - Llistats

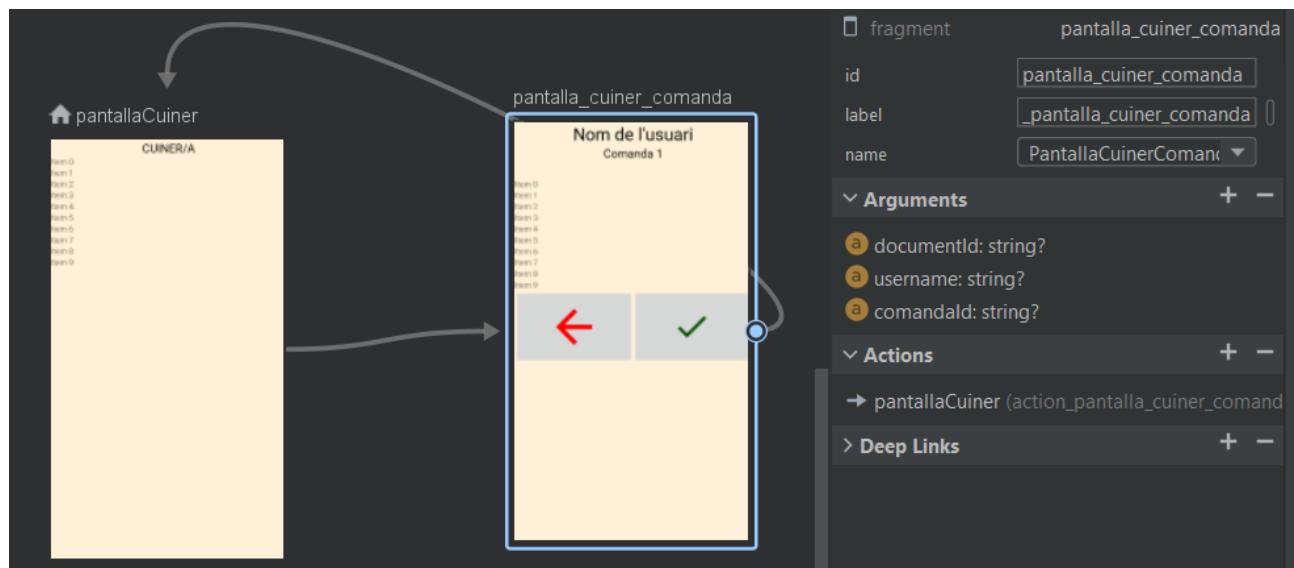
Dashboards:



Llistats:

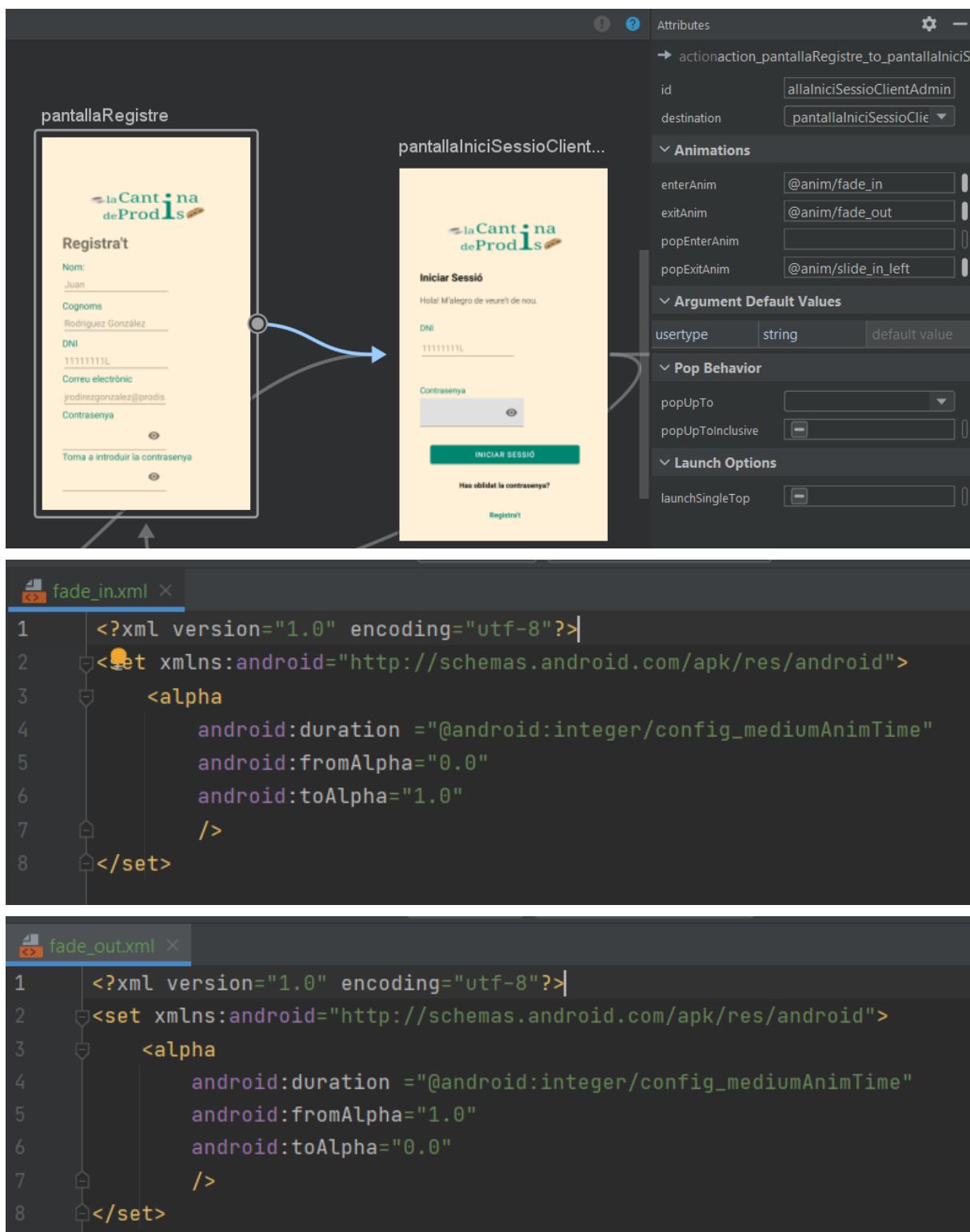
COCINERO/A	
Frederico Lorca	Usuari Temporal
Comanda 5	Comanda 8
Jaume Cadafalch Miró	Usuari Temporal
Comanda 3	Comanda 6
Adrià Tomas	Pepito PAlotes
Comanda 2	Comanda 5
Adrian Ramirez	Usuari Temporal
Comanda 6	Comanda 7
Usuari Temporal	
Comanda 9	

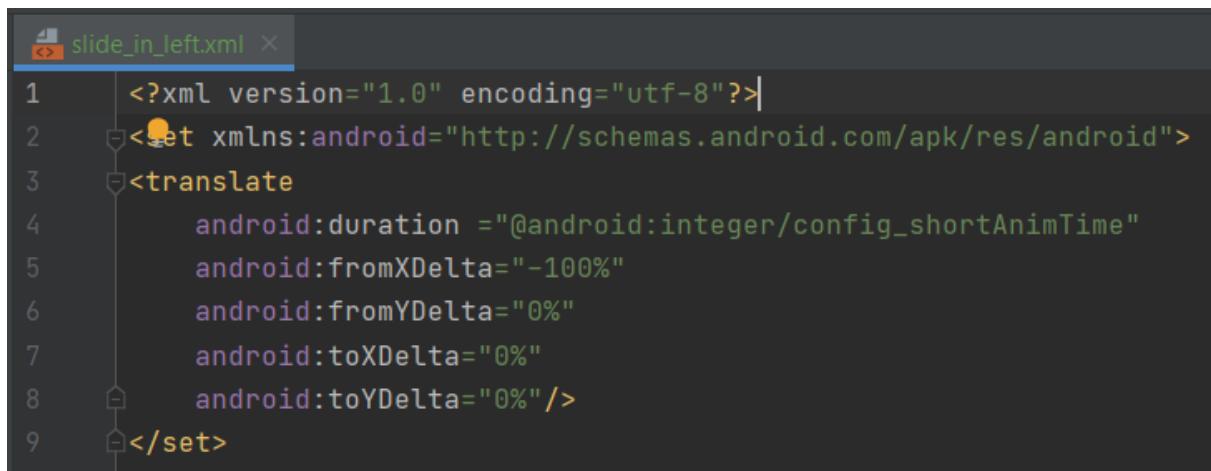
5.3.7 Pas de dades entre pantalles



5.3.8 Animacions

En el moment en què l'usuari canviï de pantalla farà l'efecte de fusió i acabarà apareixent la pantalla següent. I si l'usuari fa l'acció de tornar enrere, es farà l'animació de desplaçament de la pantalla anterior d'esquerra a dreta.





```
slide_in_left.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <set xmlns:android="http://schemas.android.com/apk/res/android">
3 <translate
4     android:duration="@android:integer/config_shortAnimTime"
5     android:fromXDelta="-100%"
6     android:fromYDelta="0%"
7     android:toXDelta="0%"
8     android:toYDelta="0%"/>
9 </set>
```

5.4 Notificacions

En la nostre aplicació enviarem una notificació cuan l'usuari faci canvis en el seu perfil.

```
//Cridem a la funció createChannel per crear un canal per poder enviar una notificació en el cas de que l'
createChannel(
    //Agadem del fitxer de strings un id i un nom per el nostre canal
    "cat.copernic.prodis.lacantinadeprodis.ui.activities",
    "app"
)

//Definim i inicialitzem una variable per passarli el NotificationManager
val notificationManager = ContextCompat.getSystemService(
    context: this,
    NotificationManager::class.java
) as NotificationManager

//Indiquem que notificationManager envii una notificació amb un text que agafara del fitxer de strings i el
notificationManager.sendNotification("Enhorabona, els canvis de l'usuari s'han realitzat amb ...", applicació)
```

```
Funció per crear el canal que tindrà un channelId i un channelName
private fun createChannel(channelId: String, channelName: String) {
    //Fem un if per comprobar si la versió del sdk es correcte
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        //Indiquem que el canal de la notificació sera de tipus NotificationChannel agafant els valors de channelId
        // notificació
        notificationChannel = NotificationChannel(
            channelId,
            channelName,
            NotificationManager.IMPORTANCE_LOW
        )

        //Indiquem que s'activi la llum del nostre dispositiu al rebre la notificació
        notificationChannel.enableLights( lights: true)
        //Indiquem el color de la llum del nostre dispositiu, en aquest cas serà blanc
        notificationChannel.lightColor = Color.WHITE
        //Indiquem que volem que el nostre dispositiu vibri al rebre la notificació
        notificationChannel.enableVibration( vibration: true)
        //Indiquem la descripció de la notificació
        notificationChannel.description = "Notificació per indicar que s'han fet els canvis del perfil"

        //Definim i inicialitzem una variable notificationManager que serà de tipus NotificationManager
        val notificationManager = getSystemService(
            NotificationManager::class.java
        )

        //Amb la variable que acabem de crear li indicarem que creï un canal amb els paràmetres que li hem indicat
        notificationManager.createNotificationChannel(notificationChannel)
    }
}
```

```
//Funció per el Notification manager que tindrà per paràmetres el missatge de la notificació i el context de l'app
private fun NotificationManager.sendNotification(
    messageBody: String,
    applicationContext: Context
) {

    //Builder per crear la notificació més tard
    val builder = NotificationCompat.Builder(
        applicationContext,
        "cat.copernic.prodis.lacantinadeprodis.ui.activities"
    )
    //Indiquem quin serà l'ícone que sortirà en la notificació
    .setSmallIcon(R.drawable.logo_foreground)
    //Indiquem quin serà el text principal de la notificació
    .setContentTitle(
        "Els canvis s'han fet amb èxit"
    )
    //Aquest serà el text de la notificació
    .setContentText(messageBody)

    //Creem la notificació amb un id i amb el builder que hem creat abans
    notify(NOTIFICATION_ID, builder.build())
}

}
```

LaCantinaDeProdis • ahora

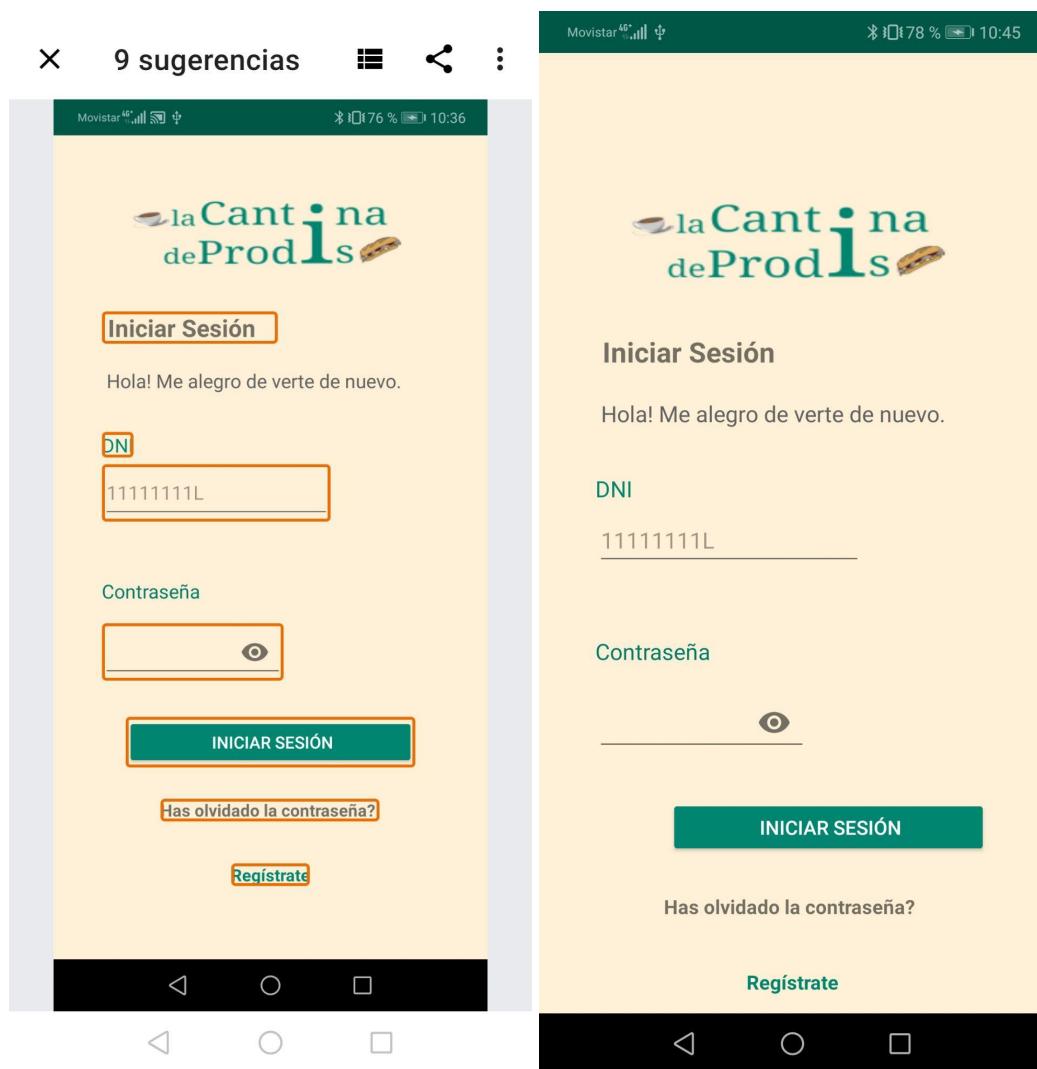
Los cambios se han hecho con éxito

Enhorabuena, los cambios del usuario se han realiz..

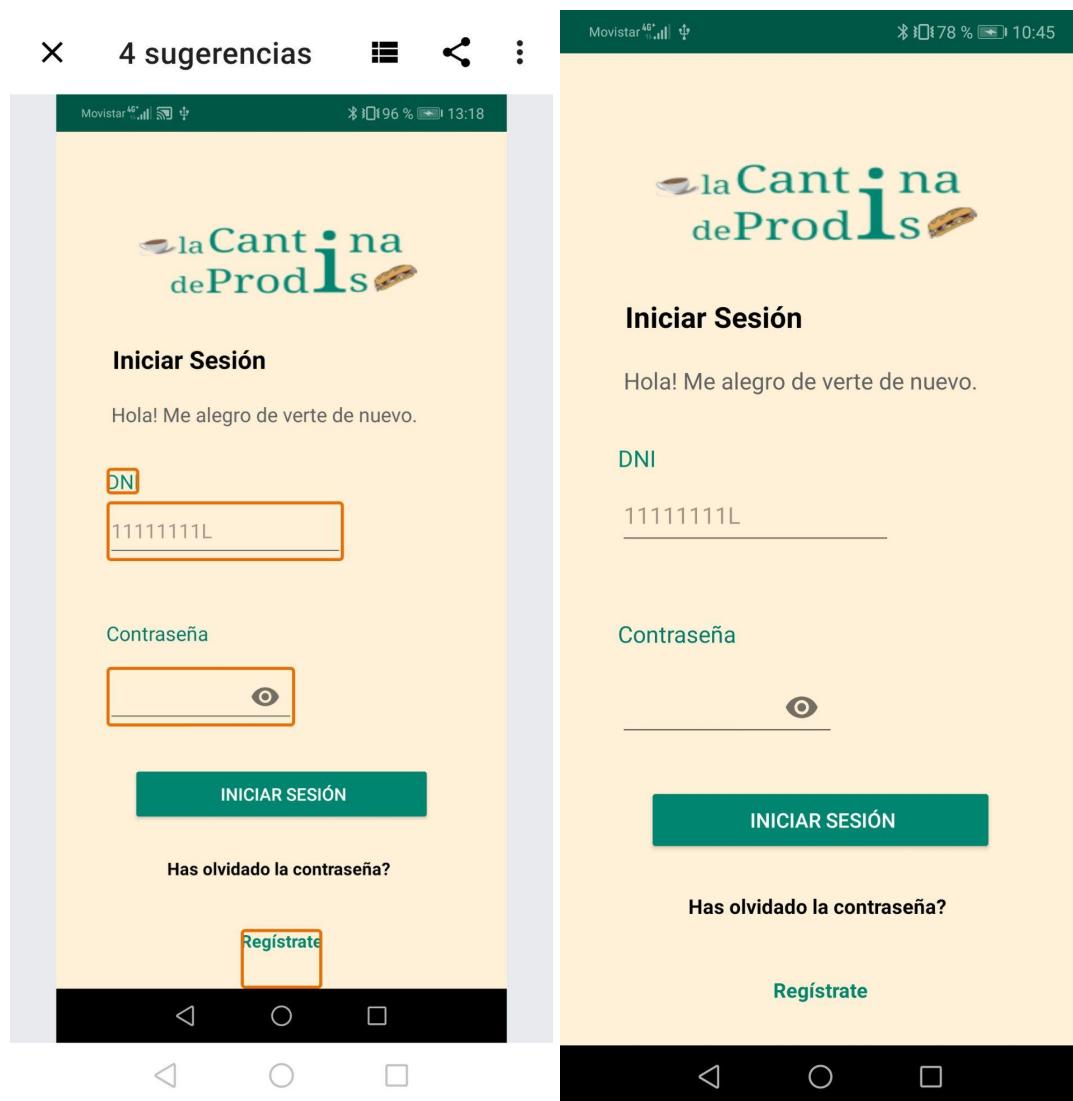
5.5 Accessibilitat

Per fer l'accessibilitat hem utilitzat l'aplicació de google [Test d'accessibilitat](#), la qual ens marcaba possibles millores a la nostre app. Aquí uns exemples:

En taronja marca els possibles errors que el test d'accessibilitat ha trobat, aquests errors normalment són; Relació de color insuficient i que l'objecte és massa petit per fer clic.



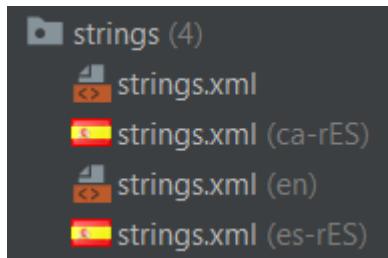
Aquest seria el resultat després de fer els canvis que el test ens ha recomanat:



Els canvis que no hem fet han sigut perquè els colors verds dels textos y els grisos dels "hints" no els volem canviar perquè ens agraden d'aquesta manera.

5.6 Internationalització

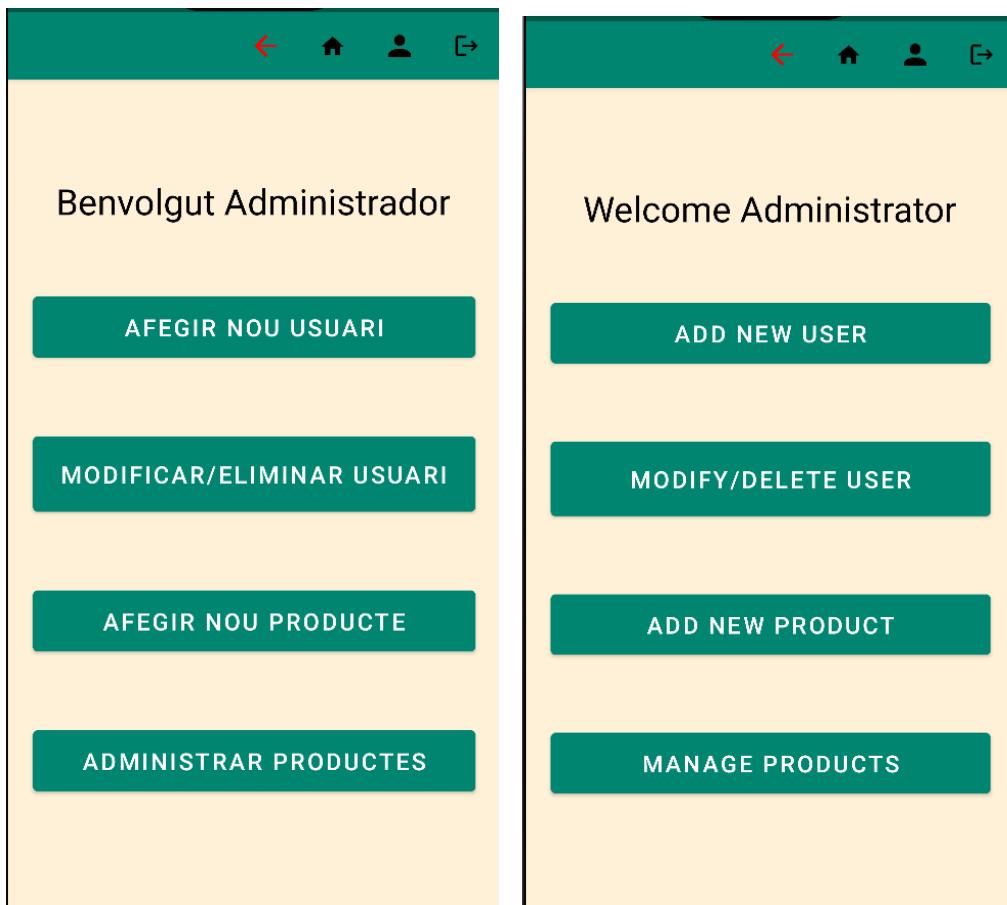
Els Strings estan disponibles en tres idiomes: Català, Castellà i Anglès



```
ca-rES\strings.xml ×
3     <string name="txtPprincipal"><u>Selecciona el tipus d\'usuari que seràs</u></string>
4     <string name="client">CLIENT/A</string>
5     <string name="cambrer">CAMBRER/A</string>
6     <string name="caixer">CAIXER/A</string>
7     <string name="cuiner">CUINER/A</string>
8     <string name="administrador">Administrador/a</string>
9     <string name="iniciarSessio">Iniciar Sessió</string>
10    <string name="txtPIniciarSessio_benvinguda">Hola! M\'alegro de veure\'t de nou.</string>
```

```
en\strings.xml ×
3     <string name="txtPprincipal"><u>Select the user type you will be</u></string>
4     <string name="client">CUSTOMER</string>
5     <string name="cambrer">WAITER</string>
6     <string name="caixer">CASHIER</string>
7     <string name="cuiner">CHEF</string>
8     <string name="administrador">Administrator</string>
9     <string name="iniciarSessio">Log in</string>
10    <string name="txtPIniciarSessio_benvinguda">Hello! Glad to see you again.</string>
```

```
es-rES\strings.xml ×
3     <string name="txtPprincipal"><u>Selecciona el tipo de usuario que serás</u></string>
4     <string name="client">CLIENTE/A</string>
5     <string name="cambrer">CAMARERO/A</string>
6     <string name="caixer">CAJERO/A</string>
7     <string name="administrador">" Administrador/a" </string>
8     <string name="iniciarSessio">" Iniciar Sesión" </string>
9     <string name="txtPIniciarSessio_benvinguda">" Hola! Me alegra de verte de nuevo." </string>
```



5.7 Disseny alternatius per varis tipus de pantalla

Per a la nostre app hem dissenyat 3 layouts diferents per a cada fitxer xml:

- Vertical telèfon
- Horitzontal telèfon
- Versió tablet

```
> fragment_pantalla_administrador_administrar_producte (3)
> fragment_pantalla_administrador_modificar_usuari (3)
> fragment_pantalla_administrador_nou_producte (3)
> fragment_pantalla_administrador_nou_usuari (3)
> fragment_pantalla_administrador_principal (3)
> fragment_pantalla_cuiner (2)
> fragment_pantalla_cuiner_comanda (3)
> fragment_pantalla_edicio_perfil (3)
> fragment_pantalla_inici_sessio_client_admin (3)
> fragment_pantalla_principal (3)
> fragment_pantalla_recuperar_contrasenya1 (3)
> fragment_pantalla_recuperar_contrasenya2 (3)
> fragment_pantalla_registre (3)
> fragment_pantalla_resum_comandes (3)
> fragment_pantalla_seleccio_atributs_beguda (3)
> fragment_pantalla_seleccio_atributs_bocata (3)
> fragment_pantalla_seleccio_beguda_calenta (3)
> fragment_pantalla_seleccio_beguda_freda (3)
> fragment_pantalla_seleccio_bocata (3)
> fragment_pantalla_seleccio_nom_client_comanda (3)
> fragment_pantalla_seleccio_resum_comanda (3)
> fragment_pantalla_seleccio_tipus_producte (3)
```

5.7.1 Versió vertical per a telèfon

Aquesta és la versió que fins ara hem mostrat.

Modificar/Eliminar usuari

Selecciona l'usuari que vols modificar

Nom de l'usuari
Juan

Cognoms de l'usuari
Rodriguez González

DN
11111111L

Correu electrònic
jrodriguezgonzalez@prodis.cat

Contraseña

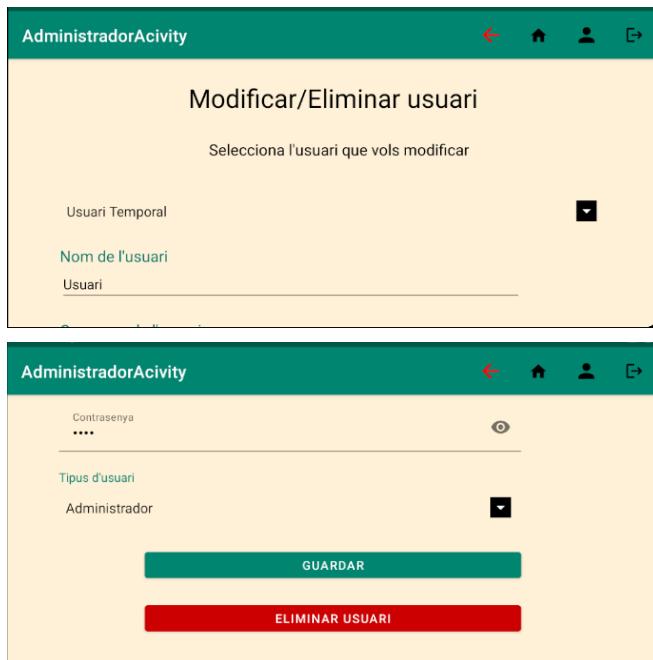
Típus d'usuari

GUARDAR

ELIMINAR USUARI

5.7.2 Versió horitzontal per a telèfon

Es una versió molt similar a l'anterior però modificant els margins i els tamanys dels botons.



AdministratorActivity

Modificar/Eliminar usuari

Selecciona l'usuari que vols modificar

Usuari Temporal

Nom de l'usuari

Usuari

AdministratorActivity

Contrassenya

Tipus d'usuari

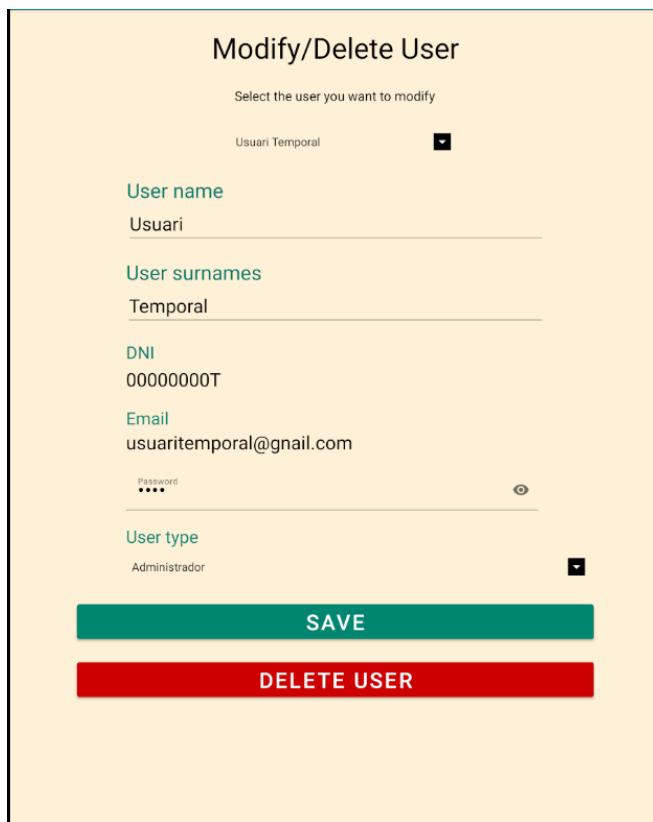
Administrador

GUARDAR

ELIMINAR USUARI

5.7.2 Versió per a tablet

És una versió molt semblant a la vertical de telèfon però amb lletres i marges més grans.



Modify/Delete User

Select the user you want to modify

Usuari Temporal

User name

Usuari

User surnames

Temporal

DNI

00000000T

Email

usuaritemporal@gmail.com

Password

User type

Administrador

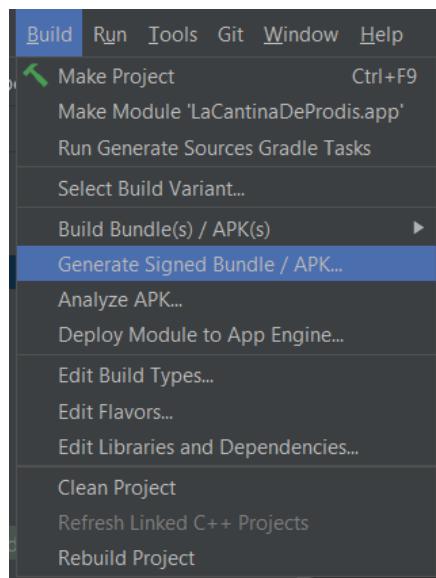
SAVE

DELETE USER

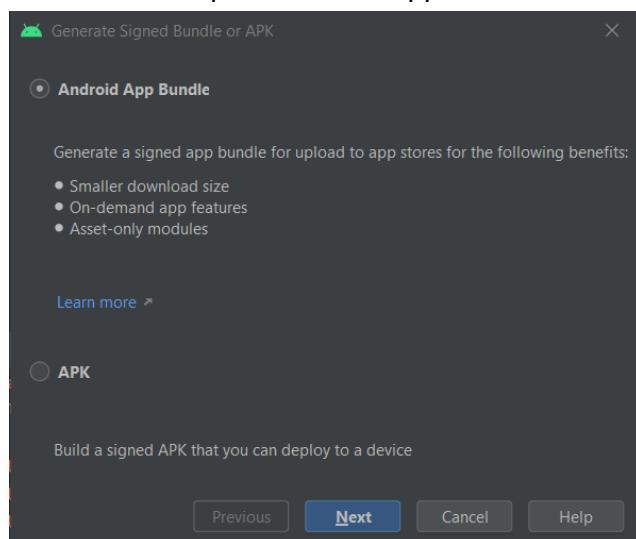
5.8 Generació APK i publicar a google play

5.8.1 Generació APK

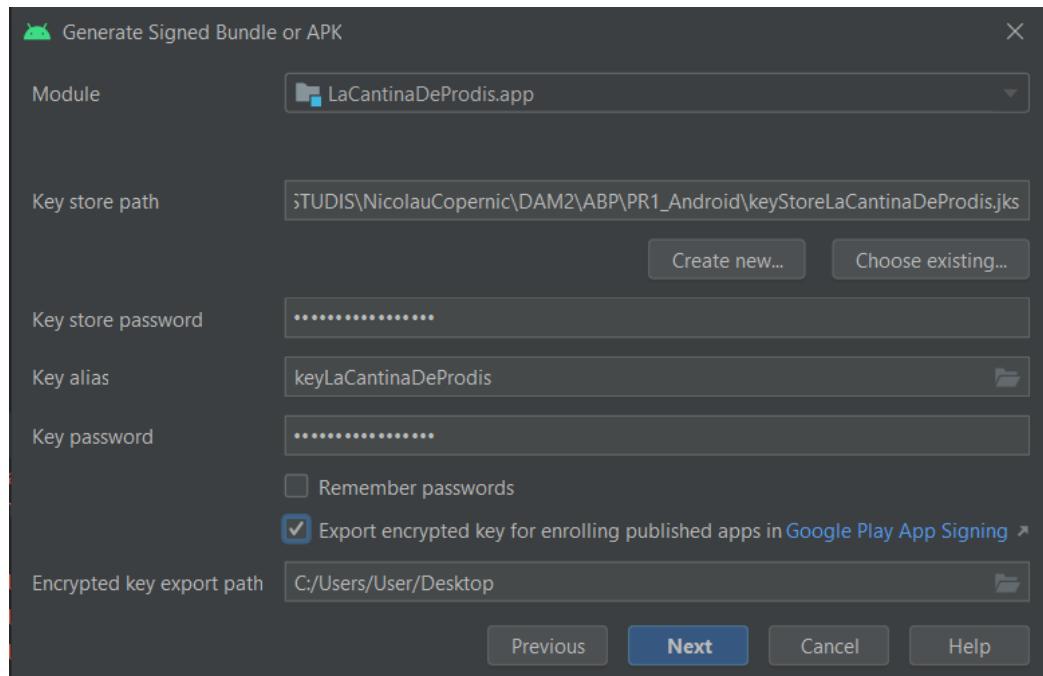
Per generar el App Bundle anem la pestanya Build i seleccionem l'opció “*Generate Signed Bundle / APK*”



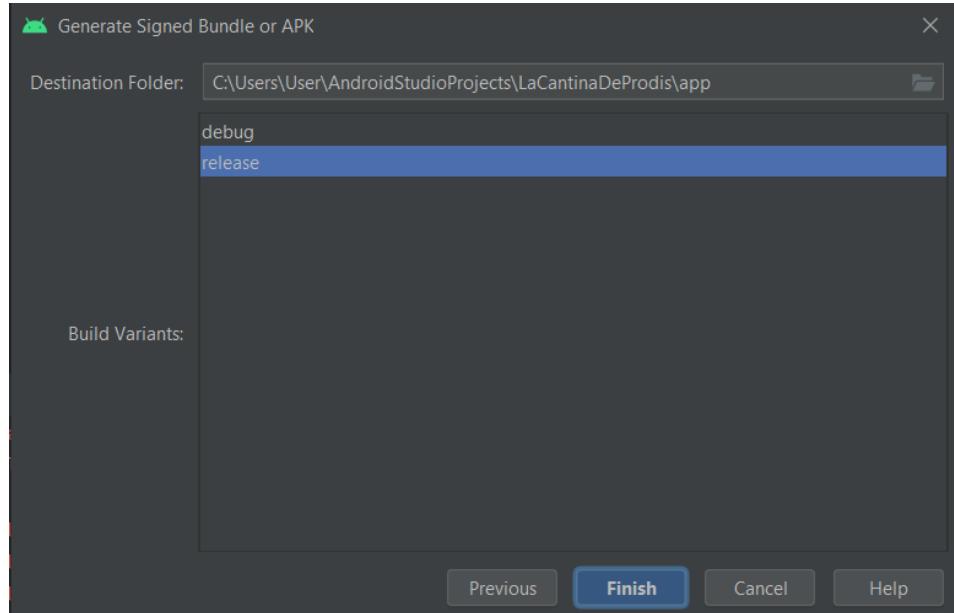
Seleccionem l'opció “*Android App Bundle*”



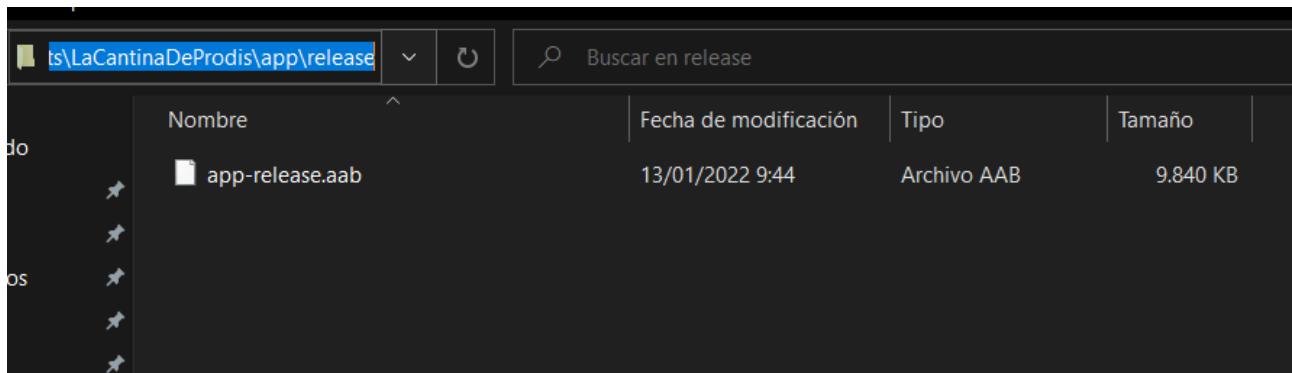
Creem una keyStore (nosaltres ja la tenim creada), introduim la “key store password”, la “key alias” i la “key password”



Indiquem la carpeta destí(nosaltres hem deixat la de per defecte) i en la Build Variants seleccionem l'opció *release*



Un cop haguem clicat en “Finish”, Android Studio generarà el fitxer que necessitem i el col·locarà en la carpeta que hem indicat abans.



5.8.2 Publicar a Google play

Per publicar la nostra aplicació a Google Play, hem d'accedir a Google Play Console.

Un cop dins, farem clic en “*Crea una aplicació*”

Totes les aplicacions

[Crea una aplicació](#)

Consulta totes les aplicacions i els jocs als quals tens accés des del teu compte de desenvolupador

I, a partir d'aquí, és on comença el procés de publicar la nostra aplicació.

Començarem indicant el nom de l'aplicació, l'idioma predeterminat, indicant si es tracta d'un joc o una aplicació, i també indiquem si la nostre aplicació serà gratuita o de pagament.

També acceptem les *Polítiques del programa per a desenvolupadors* i la *Legislació sobre exportació dels Estats Units*.

Crea una aplicació

Details de l'aplicació

Nom de l'aplicació

LaCantinaDeProdis

17/30

Així és com es mostrarà l'aplicació a Google Play

Idioma predeterminat

català (ca)

Aplicació o joc

Pots canviar-ho més endavant a la configuració de la botiga

Aplicació

Joc

Gratuita o de pagament

Pots editar aquesta informació més endavant a la pàgina Aplicació de pagament

Gratis

Declaracions

Polítiques del Programa per a desenvolupadors

Confirma que l'aplicació compleix les polítiques del Programa per a desenvolupadors
L'aplicació compleix les [polítiques del Programa per a desenvolupadors](#). Consulta [aquests consells sobre com pots crear descripcions d'aplicacions que compleixin les polítiques](#) per evitar errors habituals que poden provocar que se suspendin les aplicacions. Si la teva aplicació o la teva fitxa de Play Store compleix els requisits per enviar avisos anticipats a l'equip de revisió d'aplicacions de Google Play, contacta amb nosaltres abans de publicar el contingut.

Legislació sobre exportació dels Estats Units

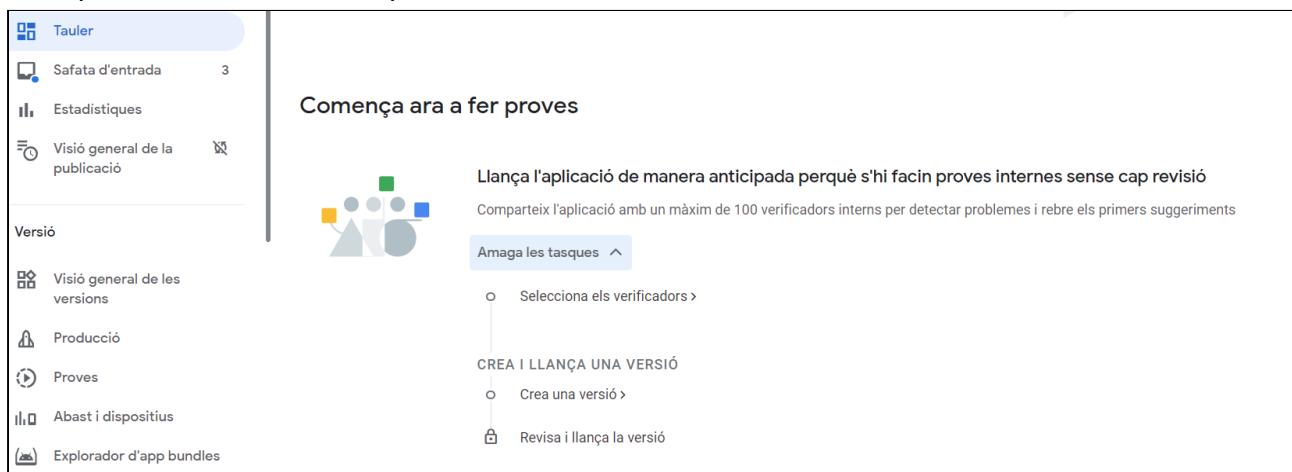
Accepta la legislació sobre exportació dels Estats Units
Accepto que la meva aplicació de programari està subjecta a la legislació sobre exportació dels Estats Units, siguin quina sigui la meva ubicació o la meva nacionalitat. Accepto el compliment d'aquesta legislació, inclos qualsevol requisit per al programari amb funcions d'criptació. D'aquesta manera, certifico que la meva aplicació està autoritzada per a l'exportació dels Estats Units d'acord amb aquesta legislació. [Més informació](#)

[Cancel·la](#)

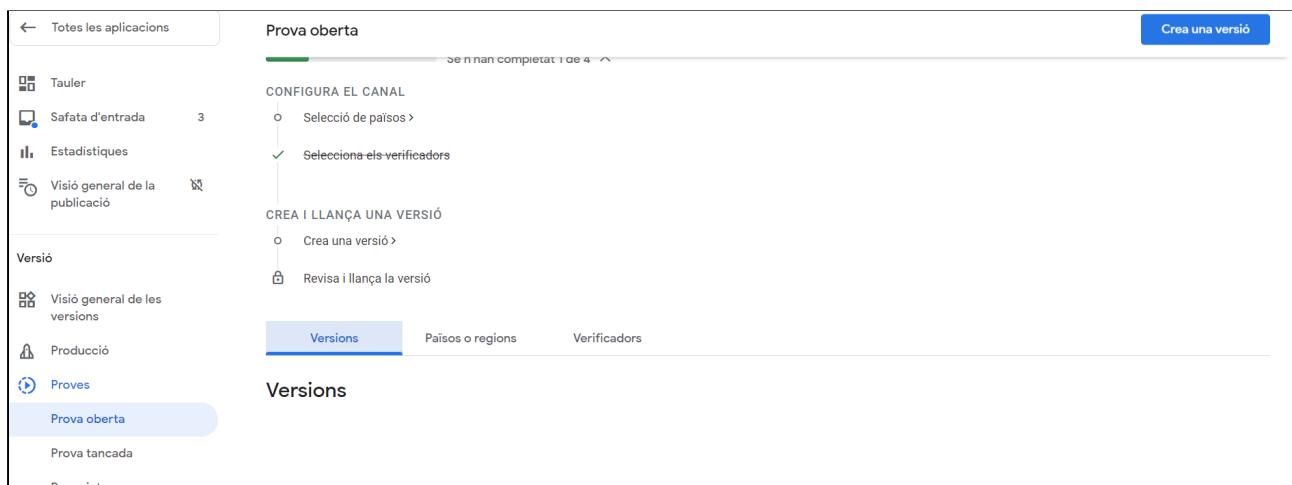
[Crea una aplicació](#)

Un cop hem fet clic en “Crea una aplicació”, se’ns mostra la següent pantalla.

En la pantalla seleccionem l’opció “Selecciona els verificadors”.



En seleccionar l’opció, ens porta a l’apartat de prova interna, com que nosaltres volem publicar una aplicació en versió “Beta” i volem que tothom la provi, ens anirem a l’apartat de “Prova oberta”.



Un cop dins veiem que tenim diferents tasques pendents de fer, comencem per seleccionar desde quins països es podran descarregar la nostra aplicació.

Prova oberta

Configura el canal de prova oberta

Inactiu

Se n'han completat 2 de 4

CONFIGURA EL CANAL

- Selecció de països
- Selecciona els verificadors

CREA I LLANÇA UNA VERSIÓ

- Crea una versió >
- Revisa i llança la versió

Versions **Països o regions** Verificadors

Països o regions 

Sincronitza els països o les regions amb el canal de producció 

Nosaltres seleccionem tots els països o regions

Afegeix països o regions a Prova oberta

If you sell subscriptions, make sure you set prices for the countries / regions you want to add.

Cerca països o regions

País o regió

Albània

Alemanya

Algèria

Angola

Antigua i Barbuda

Argentina

S'han seleccionat 176 països o regions i la resta del món

Descarta **Afegeix països o regions**

Vols afegir països o regions? 

L'aplicació estarà disponible a Google Play per a verificadors aptes de 176 països o regions nous i de la resta del món

Cancel·la **Afegeix**

Un cop hem seleccionat els països, és l'hora de crear una versió de la nostra aplicació.

Per fer-ho seleccionarem l'opció “*Crea una versió*”

Prova oberta

[Crea una versió](#)

Crea i gestiona versions de prova oberta perquè l'aplicació estigui disponible per als verificadors. Qualsevol usuari es pot unir a les proves a Google Play. [Més informació](#)

Configura el canal de prova oberta

Inactiu

 Se n'han completat 2 de 4 ^

CONFIGURA EL CANAL

-  Selecció de països
-  Selecciona els verificadors

CREA I LLANÇA UNA VERSIÓ

 Crea una versió >

 Revisa i llança la versió

Un cop dins ens demana que pengem els “app bundles”, aquí penjarem el fitxer que hem generat en l'apartat [5.8.1 Generació APK](#)

App bundles



Deixa anar aquí els app bundles que vulguis penjar

 Penja  Afegeix de la biblioteca

app-release.aab



Tipus de fitxer	Versió	Nivells d'API	SDK objectiu	Dissenys de pantalla	ABI	Funcions obligatòries
App bundle	1 (1.0)	21 i nivells superiors	31	4	Tot	2  

Un cop el fitxer s'ha penjat, indiquem el nom de la versió i opcionalment les notes de la versió.

En acabar fem clic en l'opció “Desa”.

Detalls de la versió

Nom de la versió *	v1
	2/50 Serveix per identificar aquesta versió i no es mostra als usuaris de Google Play. El nom sugerit es basa en el primer app bundle o APK d'aquesta versió, però el pots editar.
Notes de la versió	Copia d'una versió anterior
<pre><ca> Primera versió de l'aplicació </ca></pre>	
Notes de la versió introduïdes per a 1 idioma	
Informa els usuaris del contingut de la versió. Introdueix les notes de la versió en cada idioma dins de les etiquetes d'idioma.	

Descarta els canvis **Desa** [Revisa la versió](#)

En fer-ho se'n habilita el botó de “*Revisa la versió*”

Crea una versió de proves obertes

Tipus de fitxer	Versió	Nivells d'API	SDK objectiu	Dissenys de pantalla	ABI	Funció obligatòria
App bundle	1 (1.0)	21 i nivells superiors	31	4	Tot	2  

Detalls de la versió

Nom de la versió *	v1
	2/50 Serveix per identificar aquesta versió i no es mostra als usuaris de Google Play. El nom sugerit es basa en el primer app bundle o APK d'aquesta versió, però el pots editar.
Notes de la versió	Copia d'una versió anterior
<pre><ca> Primera versió de l'aplicació </ca></pre>	

 S'han desat els canvis. Ja pots revisar la versió abans de fer-ne el llançament.

Descarta els canvis Desa **Revisa la versió**

En clicar-hi sens mostrà la següent pantalla indicant que tenim errors que hem de solucionar.

Un d'ells és indicar si la nostra aplicació fa rastreig de contactes amb COVID-19, i els altres es solucionen anant a la secció “*Tauler*” i acabant de completar les tasques pendents.

Crea una versió de proves obertes

 Hem trobat alguns problemes a la teva versió



Prepara la versió

2

Revisa i llança la versió

Errors, advertiments i missatges

 2 errors

[Mostra'n més](#) ▾



1 advertiment

[Mostra'n més](#) ▾

→ Indiquem que la nostra aplicació, no ofereix serveis de rastreig de contactes amb COVID-19

← Contingut de l'aplicació

Aplicacions de rastreig de contactes amb COVID-19 i d'estat relatiu a la malaltia

Per ajudar-nos a saber si la teva aplicació és de rastreig de contactes o d'estat relatiu a la COVID-19, selecciona totes les afirmacions següents que s'hi puguin aplicar.

- La meva aplicació és de rastreig de contactes amb COVID-19 i està disponible públicament
Per exemple, una aplicació que rastreja o monitora les persones infectades o exposades amb finalitats de combatre o mitigar la COVID-19
- La meva aplicació és d'estat relatiu a la COVID-19 i està disponible públicament
Per exemple, una aplicació que verifica l'estat de contagi actual, la situació de vacunació o l'historial de contagis d'una persona amb l'objectiu de determinar si és apta per viatjar o accedir a espais públics. [Més informació](#)
- La meva aplicació no ofereix públicament serveis de rastreig de contactes ni d'estat relatiu a la COVID-19

→ En el tauler trobem totes les tasques pendents de fer

Tauler

Comença a configurar la teva aplicació. [Mostra'n més](#)

Comença ara a fer proves



Llança l'aplicació de manera anticipada perquè s'hi facin proves internes sense cap revisió

Comparteix l'aplicació amb un màxim de 100 verificadors interns per detectar problemes i rebre els primers suggeriments

[Mostra les tasques](#) ▾

Configura la teva aplicació



Proporciona informació sobre l'aplicació i configura la fitxa de Play Store

Indica'n quin és el contingut de la teva aplicació i gestiona com s'organitza i es presenta l'aplicació a Google Play

 Se n'han completat 1 de 9 ^

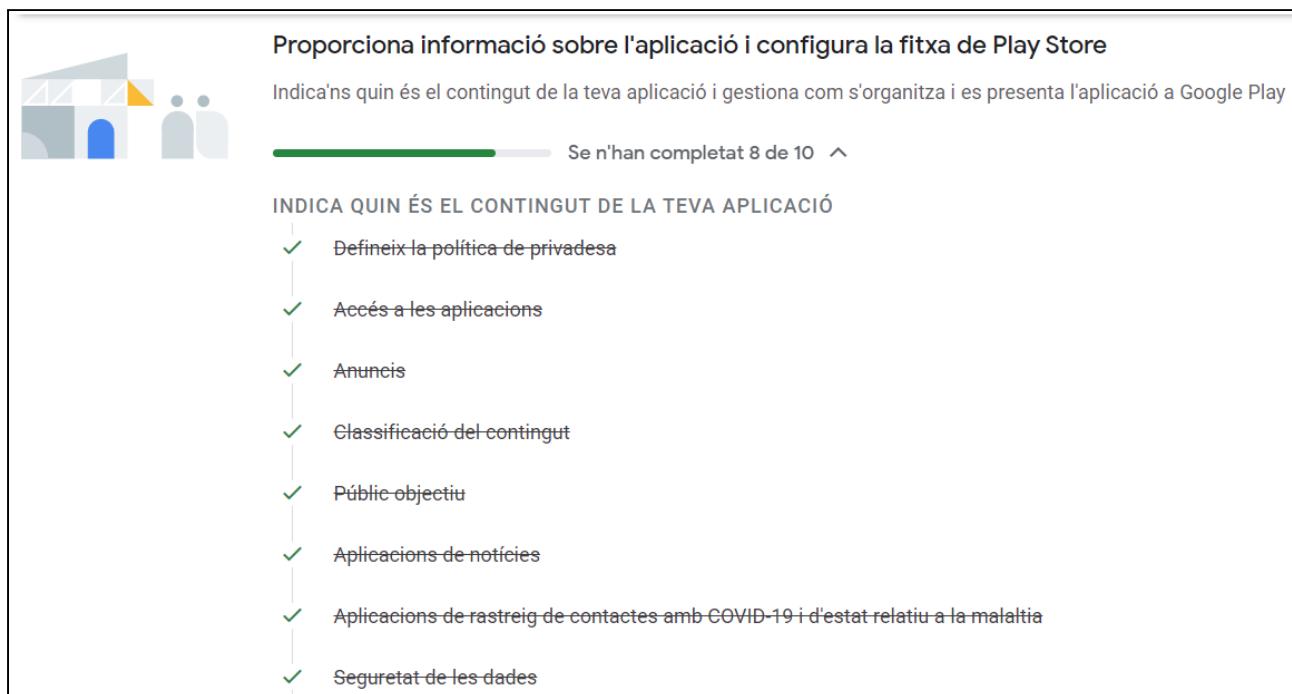
INDICA QUIN ÉS EL CONTINGUT DE LA TEVA APLICACIÓ

- Accés a les aplicacions >

Ens trobem un llistat de tasques pendents de fer.

<div style="background-color: #f0f0f0; padding: 5px;"> <p> Tauler</p> <p> Safata d'entrada 3</p> <p> Estadístiques</p> <p> Visió general de la publicació</p> <hr/> <p>Versió</p> <p> Visió general de les versions</p> <p> Producció</p> <p> Proves</p> <ul style="list-style-type: none"> Prova oberta Prova tancada </div>	<div style="background-color: #f0f0f0; padding: 5px;"> <p></p> <p>Proporciona informació sobre l'aplicació i configura la fitxa de Play Store</p> <p>Indica'n quin és el contingut de la teva aplicació i gestiona com s'organitza i es presenta l'aplicació a Google Play</p> <p> Se n'han completat 1 de 9 ^</p> <p>INDICA QUIN ÉS EL CONTINGUT DE LA TEVA APLICACIÓ</p> <ul style="list-style-type: none"> <input type="radio"/> Accés a les aplicacions > <input type="radio"/> Anuncis > <input type="radio"/> Classificació del contingut > <input type="radio"/> Públic objectiu > <input type="radio"/> Aplicacions de notícies > <input checked="" type="radio"/> Aplicacions de rastreig de contactes amb COVID-19 i d'estat relatiu a la malaltia <input type="radio"/> Seguretat de les dades > </div>
--	---

Realitzem les tasques que ens demanen



Proporciona informació sobre l'aplicació i configura la fitxa de Play Store

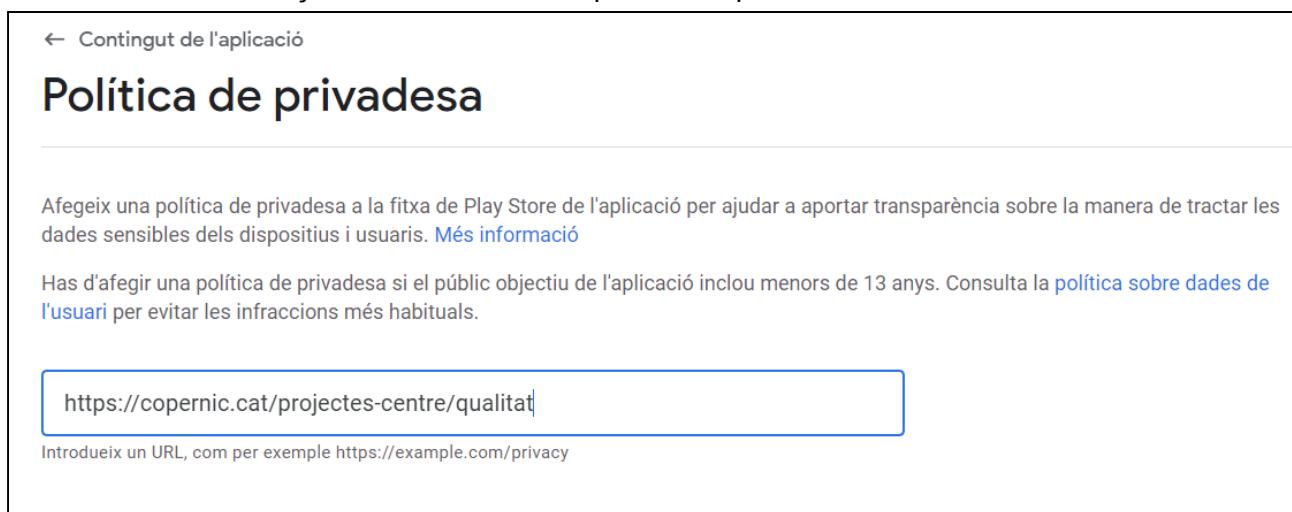
Indica'n quin és el contingut de la teva aplicació i gestiona com s'organitza i es presenta l'aplicació a Google Play

Se n'han completat 8 de 10 ^

INDICA QUIN ÉS EL CONTINGUT DE LA TEVA APLICACIÓ

- Defineix la política de privadesa
- Accés a les aplicacions
- Anuncis
- Classificació del contingut
- Públic objectiu
- Aplicacions de notícies
- Aplicacions de rastreig de contactes amb COVID-19 i d'estat relatiu a la malaltia
- Seguretat de les dades

A més declarem l'enllaç on es troba la nostra política de privacitat



← Contingut de l'aplicació

Política de privadesa

Afegeix una política de privadesa a la fitxa de Play Store de l'aplicació per ajudar a aportar transparència sobre la manera de tractar les dades sensibles dels dispositius i usuaris. [Més informació](#)

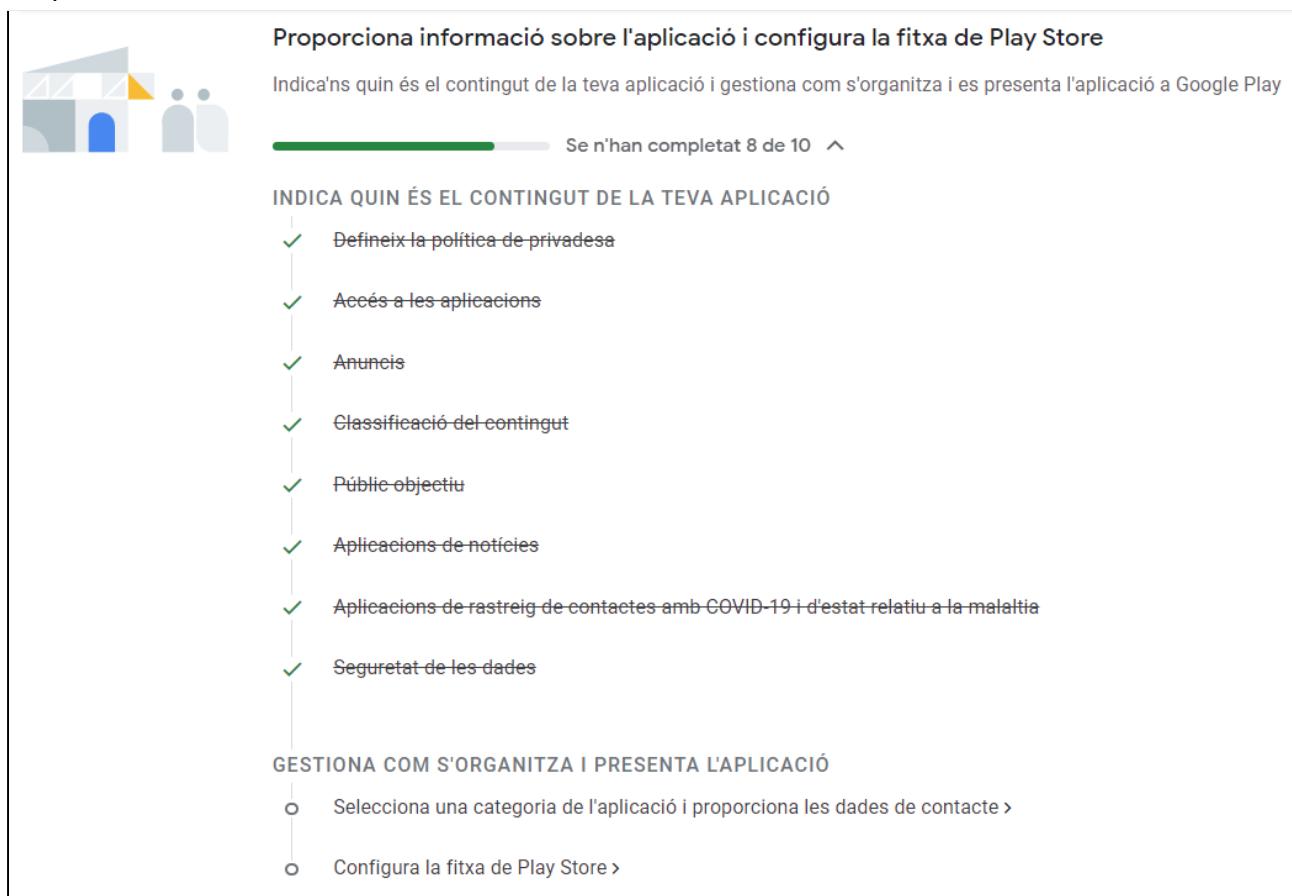
Has d'afegeir una política de privadesa si el públic objectiu de l'aplicació inclou menors de 13 anys. Consulta la [política sobre dades de l'usuari](#) per evitar les infraccions més habituals.

https://copernic.cat/projectes-centre/qualitat

Introdueix un URL, com per exemple https://example.com/privacy

Continuem omplint les dades que ens demanen:

Un cop hem acabat de fer les tasques anterior, ens trobem que hem de realitzar dues tasques més.



Proporciona informació sobre l'aplicació i configura la fitxa de Play Store

Indica'n quin és el contingut de la teva aplicació i gestiona com s'organitza i es presenta l'aplicació a Google Play

Se n'han completat 8 de 10 ^

INDICA QUIN ÉS EL CONTINGUT DE LA TEVA APLICACIÓ

- Defineix la política de privadesa
- Accés a les aplicacions
- Anuncis
- Classificació del contingut
- Públic objectiu
- Aplicacions de notícies
- Aplicacions de rastreig de contactes amb COVID-19 i d'estat relatiu a la malaltia
- Seguretat de les dades

GESTIONA COM S'ORGANITZA I PRESENTA L'APLICACIÓ

- Selecciona una categoria de l'aplicació i proporciona les dades de contacte >
- Configura la fitxa de Play Store >

Una d'elles és indicar si es tracta d'un joc o una aplicació, categoria i etiquetes.

Configuració de la botiga

Gestiona com s'organitza l'aplicació a Google Play i de quina manera els usuaris poden contactar amb tu

*: Camps obligatoris.

Categoría de l'aplicació

Tria el tipus d'aplicació, la categoria i les etiquetes que descriguin millor el contingut o la funció principal de l'aplicació. Aquesta informació ajuda els usuaris a descobrir aplicacions a Google Play.

Aplicació o joc *	Aplicació
Categoría *	Menjar i begudes
Etiquetes	Restaurants
Gestiona les etiquetes	

L'altre tasca és configurar la fitxa del Google Play:

Indicant la adreça electrònica i opcionalment el número de telèfon i el lloc web.

Informació de contacte de la fitxa de Play Store

Aquesta informació es mostra als usuaris a Google Play

Adreça electrònica *

lacantinadeprodis@gmail.com

Número de telèfon

Lloc web

http://www.lacantina.cat/

També ens demana que indiquem el nom de l'aplicació, una breu descripció i una descripció completa.

Fitxa principal de Play Store

Predeterminat: català (ca) Gestiona les traduccions ▾

Detalls de l'aplicació

Comprova la [política de metadades](#) i les [indicacions del Centre d'ajuda](#) per evitar problemes habituals amb la fitxa de Play Store. Revisa totes les [polítiques del programa](#) abans d'enviar l'aplicació.

Si compleixes els requisits per [proporcionar avisos anticipats](#) a l'equip de revisió d'applications, contacta amb nosaltres abans de publicar la teva fitxa de Play Store.

Nom de l'aplicació *

LaCantinaDeProdis

Així és com es mostrerà l'aplicació a Google Play

17/30

Descripció breu *

LaCantinaDeProdis és una aplicació destinada a gestió de una cantina.

Una descripció breu de l'aplicació. Els usuaris la poden desplegar per veure la descripció sencera. 69/80

Descripció completa *

LaCantinaDeProdis és una aplicació enfocada a la gestió de la cantina de Prodis Terrassa (una entitat de Terrassa d'iniciativa social sense ànim de lucre destinada a l'assistència i la promoció integral de persones adultes amb discapacitat intel·lectual, malaltia mental o paràlisi cerebral).

Entre les seves funcionalitats, trobem: realitzar comandes (ja sigui a nivell cambrer, com a nivell client), poder veure el contingut de les comandes i poder veure el preu total de la comanda.

486/4000

També ens demana que pujem la icona de l'aplicació, una imatge destacada

Icona de l'aplicació *



[⬆ Substueix](#)

La icona de l'aplicació ha de ser un fitxer PNG o JPEG transparent que ocupa com a màxim 1 MB i mesuri 512 x 512 pixels. A més, ha de complir les nostres [especificacions de disseny](#) i la nostra [política de metadades](#)

Imatge destacada *

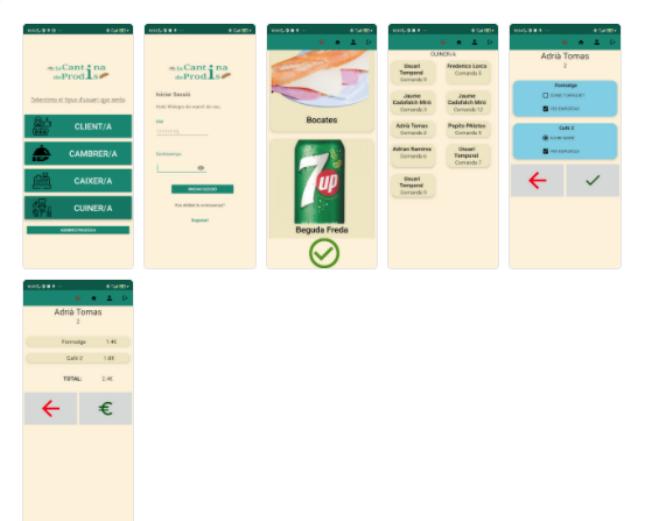


[⬆ Substueix](#)

També ens demana que pugem captures de pantalla d'algunes de les pantalles.

Telèfon

Captures de pantalla d'un telèfon *



6/8

[⬆ Penja](#)

Penja de 2 a 8 captures de pantalla del telèfon. Aquestes captures han de ser fitxers PNG o JPEG que ocupin com a màxim 8 MB cadascun, tinguin una relació d'aspecte de 16:9 o 9:16 i els costats dels quals mesurin entre 320 i 3840 pixels.

Un cop hem solucionat els errors ja podem donar-li a l'opció:

Crea una versió de proves obertes

⚠ Hem trobat alguns problemes a la teva versió

✓ Prepara la versió — 2 Revisa i llança la versió [Descarta la versió](#)

Errors, advertiments i missatges

⚠ 1 avvertiment [Mostra'n més](#) ▾

App bundles nous

Tipus de fitxer	Versió	Nivells d'API	SDK objectiu	Dissenys de pantalla	ABI	Funcions obligatòries
App bundle	1 (1.0)	21 i nivells superiors	31	4	Tot	2 →

Desa Revisa la versió

En clicar en “*Revisa la versió*” ens porta al Tauler on ens trobem que l'aplicació és troba en revisió.

Tauler

Comença a configurar la teva aplicació. [Mostra'n més](#)



LaCantinaDeProdis
 cat.copernic.prodis.lacantinadeprodis • [Mostra a Google Play](#)

Estat de l'actualització

🕒 En revisió • [Ves a Visió general de la publicació](#)

Producció
Inactiu

[Mostra els canals de prova](#) ▾

Un cop hagi acabat la revisió, ja tindrem l'aplicació publicada.



LaCantinaDeProdis

Institut Nicolau Copernic Menjar i begudes

 PEGI 3

 Aquesta aplicació està disponible per al teu dispositiu

Pots compartir aquest contingut amb la teva família.
[Més informació sobre la biblioteca familiar](#)

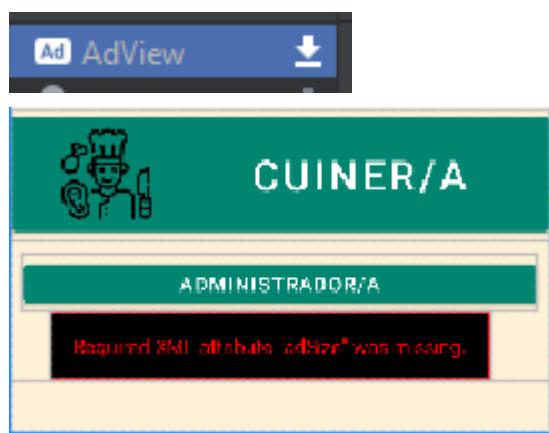
Instal·lada

5.9 Hipotètica monetització de l'app

En el nostre cas per monetitzar la nostra app utilitzaríem AdMob, un recurs de Google que ens permet guanyar diners amb anuncis. En l'Android Studio trobarem aquest recurs en l'apartat de Google en la paleta per construir la nostra vista. Aquest element es diu AdView i es pot programar per fer el que nosaltres vulguem després de veure un vídeo. Podem programar:

- Quan carregui
- Quan doni error al carregar
- Quan s'obre
- Al fer clic
- Al tancar-se

En pantalla també es podria posar un banner on sortiran els anuncis.



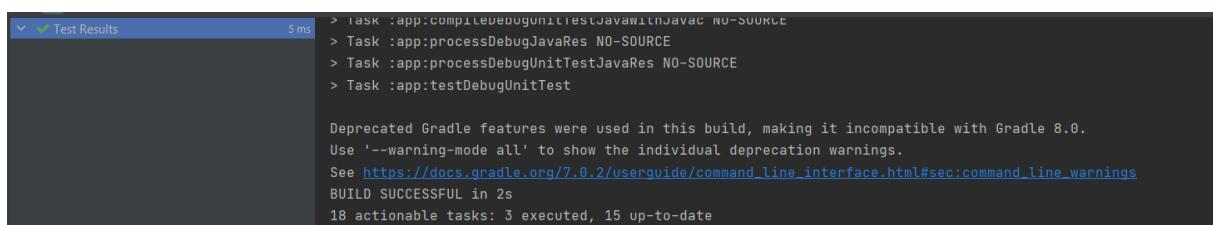
6 Testeig

6.1 Testeig proves unitàries

Per fe les proves hem utilitzat dos mètodes diferents, un per comprovar que el format de l'email sigui correcte i un altre per comprovar que el format del DNI també sigui correcte. En les següents captures es pot veure com primer es fa un testeig amb un correu que té un format correcte, donant "true" el testeig. En el següent test passem un correu amb un format incorrecte i esperem un "false", donant un testeig correcte.

```
@Test
fun checkEmailFormat() {
    val util = NoCaractersEspecial()
    val txtEmail = "hola@gmail.com"
    val resultat = util.checkEmailFormat(txtEmail)
    Assert.assertEquals( expected: true, resultat)
}

@Test
fun checkEmailFormat2() {
    val util = NoCaractersEspecial()
    val txtEmail = "hola@gmailcom"
    val resultat = util.checkEmailFormat(txtEmail)
    Assert.assertEquals( expected: false, resultat)
}
```



The screenshot shows the 'Test Results' window in Android Studio. It displays the command-line output of the build process. The output includes:

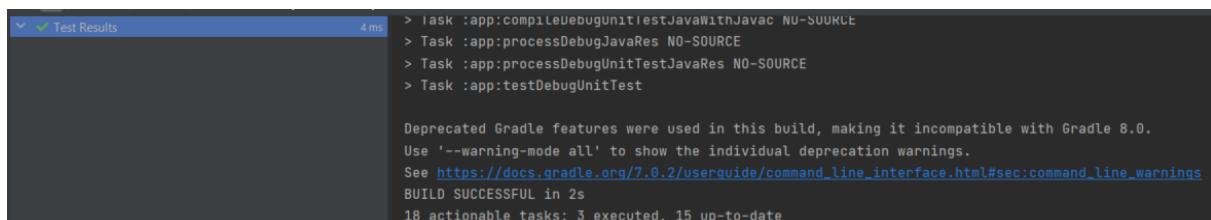
```
> Task :app:compileDebugUnitTestJavaWithJavac NO-SOURCE
> Task :app:processDebugJavaRes NO-SOURCE
> Task :app:processDebugUnitTestJavaRes NO-SOURCE
> Task :app:testDebugUnitTest

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/7.0.2/userguide/command\_line\_interface.html#sec:command\_line\_warnings
BUILD SUCCESSFUL in 2s
18 actionable tasks: 3 executed, 15 up-to-date
```

Fem el mateix que hem fet amb els testos del email amb el format del DNI

```
@Test
fun checkDni() {
    val util = NoCaracterEspecial()
    val dni = "45934842R"
    val resultat = util.checkDni(dni)
    Assert.assertEquals( expected: true, resultat)
}

@Test
fun checkDni2() {
    val util = NoCaracterEspecial()
    val dni = "26483492P"
    val resultat = util.checkDni(dni)
    Assert.assertEquals( expected: false, resultat)
}
```



6.2 Test amb usuaris

Hem fet el testeig amb 4 grups de persones; joves/adults amb experiència amb tecnologies, gent gran amb experiència amb tecnologies i gent gran sense experiència amb tecnologies.

Persona 1 → Jove amb experiència amb tecnologies:

- Cap problema
 - Intuitiva
 - Visualment agradable

Persona 2 → Persona gran amb experiència amb tecnologies:

- Ha necessitat ajuda per crear un compte
- Ha realitzat una comanda correctament

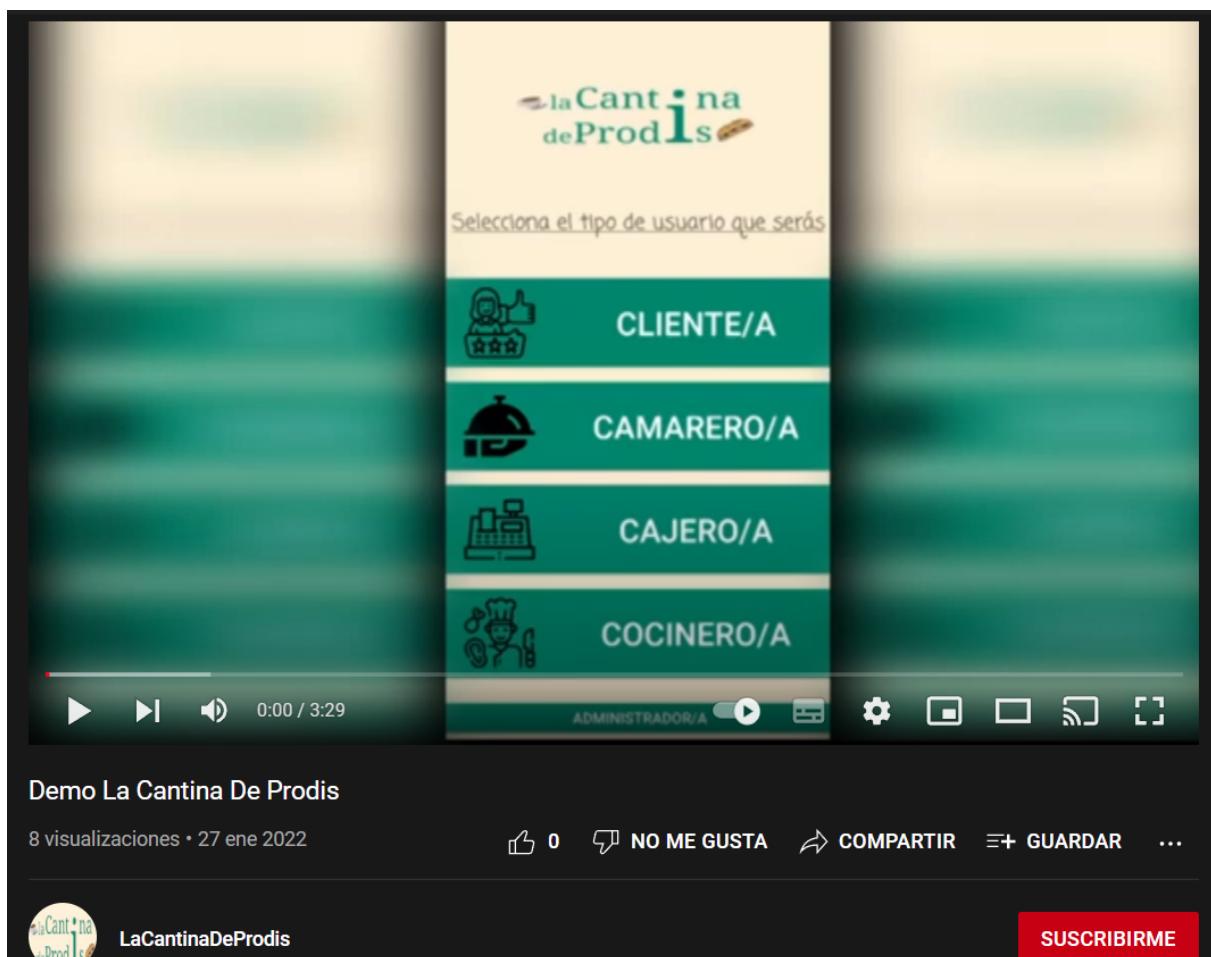
Persona 3 → Persona gran sense experiència amb tecnologies:

- Ha necessitat ajuda per crear un compte
- Ha necessitat ajuda per iniciar sessió
- Ha necessitat ajuda per fer una comanda

Persona 4 → Persona adulta amb experiència amb tecnologies:

- Fàcil ús de l'app

7 Aplicació final



8 Línes Futures

Possibles noves funcions

Per falta de temps hi ha funcionalitats que no han entrat en l'app i que volíem implementar, com per exemple; poder generar un arxiu PDF o CSV amb la comptabilitat de la caixa. També estem oberts a escoltar propostes noves i implementar-les.

Corregir bugs

Som conscients que no som perfectes i possiblement hi ha bugs que no hem trobat o que tampoc han sortit durant el testeig, per això, estem disposats a arreglar qualsevol bug que surtin en el futur.

9 Conclusió

En conclusió, hem après un nou llenguatge de programació a la vegada que l'hem estat implementant en aquest projecte. A més a més També volem recalcar que aquest projecte ha sigut una experiència molt interessant, ja que també hem estat treballant amb una metodologia molt diferent com és l'ABP (aprenentatge basat en projectes). Estem contents amb el resultat final i també estem molt contents per haver treballat per una entitat com Prodis. Per un altre costat, sí que ens hagués agradat implementar més funcionalitats com poder generar un registre de la caixa.

Webgrafia

Informació sobre Trello:

<https://www.getapp.es/software/90700/trello>

Informació sobre Team Gantt:

<https://jorgesaitz.com/blog/valoracion-de-teamgantt/>

<https://dhtrust.org/top-10/revision-de-teamgantt-pcmag/>

<https://www.getapp.es/software/103873/teamgantt>

Informació sobre Gitlab

<https://www.getapp.es/software/112635/gitlab>

Informació sobre Git:

<https://git-scm.com>

<https://openwebinars.net/blog/que-es-git-y-para-que-sirve/>

<https://www.atlassian.com/es/git/tutorials/using-branches>

<https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase>

<https://www.atlassian.com/es/git/tutorials/using-branches/git-merge>

<https://www.atlassian.com/es/git/tutorials/syncing/git-pull#:~:text=El%20comando%20git%20pull%20se,local%20para%20reflejar%20ese%20contenido.&text=El%20comando%20git%20pull%20es.fetch%20seguido%20de%20git%20merge%20.>

<https://www.freecodecamp.org/espanol/news/el-comando-git-push-explicado/#:~:text=El%20comando%20git%20push%20te,tus%20cambios%20al%20repositorio%20local.>

<https://www.atlassian.com/git/tutorials/saving-changes/git-commit>

<https://www.gitkraken.com/learn/git/git-clone#:~:text=Git%20clone%20is%20used%20to,checkouts%20an%20initial%20branch%20locally.>

<https://git-scm.com/docs/git-revert#:~:text=Note%3A%20git%20revert%20is%20used,particularly%20the%20--hard%20option.>

Informació sobre merge request:

https://docs.gitlab.com/ee/user/project/merge_requests/creating_merge_requests.html

https://docs.gitlab.com/ee/user/project/merge_requests/getting_started.html

<https://view.genial.ly/61e12ceeb5ddd60dcc35024b/presentation-lacantinadeprodis>