

Disseny i implementació d'una aplicació per realitzar compres de proximitat

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Autor: Marc aparicio Arbusà

Tutora: Carme Quer Bosor

Director: Alejandro Jiménez Ventura

Especialitat: Especialització en enginyeria del software

Resum

Amb l'aparició del COVID19 a principis de 2020 molts comerços de proximitat han hagut d'adaptar-se al context de l'aïllament social i mantenir el negoci utilitzant les xarxes per vendre els seus productes.

Un any després de l'inici de la pandèmia ens trobem que molts comerços no s'han pogut adaptar a aquest nou model de negoci i estan en procés de desaparició.

Per ajudar a aquest comerç s'ha plantejat una aplicació mòbil pensada perquè els propietaris d'aquests comerços puguin oferir els seus productes.

El document descriu el procés de creació de l'aplicació durant els vuit mesos de desenvolupament.

Resumen

Con la aparición del COVID 19 a principios de 2020 muchos comercios de proximidad han tenido que adaptarse al contexto de aislamiento social y mantener el negocio utilizando las redes sociales para vender sus productos.

Un año después del inicio de la pandemia nos encontramos que muchos de estos comercios no se han podido adaptar a este nuevo modelo de negocio y están en proceso de desaparición.

Para ayudar a estos comercios se ha planteado una aplicación móvil pensada para que los propietarios de estos comercios puedan ofrecer sus productos.

El documento describe el proceso de creación de la aplicación durante estos ocho meses de desarrollo.

Abstract

With the appearance of COVID 19 at the beginning of 2020, many local businesses have had to adapt to the global context of social isolation and maintain business using social networks to sell their product.

One year after the outbreak of the pandemic, we find that many of these businesses have not been able to adapt to this new business model and they are in the process of disappearing.

To help these businesses, a mobile application has been designed so that the owners of these businesses can offer their products.

This document describes the process of creating the application during this eight month of development.

ÍNDEX

1.Introducció	6
1.1 Introducció	6
1.2 Context	6
1.3 Glossari	7
1.4 Objectius	8
1.5 Parts interessades	8
1.5.1 Desenvolupador del projecte	8
1.5.2 Botiguers i propietaris de comerços	8
1.5.3 Els clients	8
1.5.4 Administradors	9
1.6 Anàlisis del mercat existent	9
1.6.1 Aplicació Lacompra	9
1.6.2 Aplicació TooGoodtogo	9
1.6.3 Pàgines d'associacions de comerciants	10
1.7 Possibles obstacles	10
1.8 Metodologia	11
1.8.1 Metodologia de treball	11
1.8.2 Mètode de validació	11
1.9 Organització de la resta de la memòria	12
2. Planificació inicial	14
2.1 Calendari	14
2.2 Descripció de les tasques	14
2.2.1 Gestió inicial del projecte	14
2.2.2 Cerca i comunicació amb una associació de comerciants	14
2.2.3 Creació d'entorn de treball	15
2.2.4 Especificació de requisits de l'aplicació	15
2.2.5 Desenvolupament del backend	15
2.2.6 Comprovació del funcionament del backend	16
2.2.7 Desenvolupament del frontend	16
2.2.8 Integració de les crides de l'API al frontend	16
2.2.9 Part final, redacció i defensa	17
2.3 Estimació de temps	17
2.4 Pla d'acció	18
2.5 Diagrames de Gantt	18
3. Gestió econòmica	19
3.1 Pressupost	19
3.2 Recursos humans	19
3.3 Recursos de hardware i software	20
3.4 Despeses generals	21
3.5 Despeses total	22
3.6 Control de gestió	22

3.7 Costos imprevistos	23
4. Especificació de requisits	24
4.1 Procés d'obtenció de requisits	24
4.2 Requisits funcionals	25
4.2.1 Llista de requisits funcionals	25
4.2.2 Diagrames de casos d'ús	26
4.2.3 Descripció dels casos d'ús	27
4.2.4 Model conceptual de les dades	36
4.3 Requisits no funcionals	37
4.3.1 Requisits de percepció	38
Taula 29: Requisit de percepció 1	38
4.3.2 Requisits d'usabilitat	38
Taula 30: Requisit d'usabilitat 1	38
4.3.3 Requisits de rendiment	38
Taula 31: Requisit de rendiment 1	38
Taula 32: Requisit de rendiment 2	39
4.3.4 Requisits de fiabilitat i disponibilitat	39
Taula 33: Requisit de fiabilitat i disponibilitat 1	39
Taula 34: Requisit de fiabilitat i disponibilitat 2	39
4.3.5 Seguretat	39
Taula 35: Requisit de seguretat 1	39
Taula 36: Requisit de seguretat 2	40
4.3.6 Requisits legals	40
Taula 37: Requisit legal 1	40
5. Arquitectura del sistema	41
5.1 Arquitectura física	41
5.1.1 Heroku	41
5.1.2 Firebase auth	41
5.1.3 Dispositiu Android	42
5.1.4 Diagrama de l'arquitectura física	42
5.2 Arquitectura lògica	43
5.2.1 Arquitectura lògica de l'aplicació mòbil	43
5.2.2 Arquitectura lògica de l'aplicació de serveis web	45
5.3 Patrons utilitzats	46
5.3.1 Patró de disseny MVC	46
5.3.2 Patró Singleton	46
5.3.3 Patró adaptador	47
5.3.4 Patró DTO (Data Transfer Object)	48
5.4 Exemples de diagrames de seqüència	49
5.4.1 Diagrama de seqüència de registre d'usuari	49
5.4.2 Diagrama de seqüència de mostrar informació de comanda	49
5.4.3 Diagrama de seqüència de sol·licitar una comanda	50
5.4.4 Diagrama de seqüència de modificar producte del comerç del botiguer	51

5.5 Bases de dades	51
5.5.1 Disseny lògic	52
5.5.2 Disseny físic	52
5.6 Disseny de la interfície	54
5.6.1 Finestra inicial / login i finestra de registre	55
5.6.2 Finestra principal	55
5.6.3 Finestra de comandes per recollir de l'usuari	56
5.6.4 Finestra del mapa de navegació	56
5.6.5 Finestra de cerca de comerços	57
5.6.6 Finestra d'informació de comerç	58
5.6.7 Finestra de confirmació de comanda	59
5.6.8 Finestra de perfil d'usuari	60
5.6.9 Finestra de modificació de dades de l'usuari i finestra de canvi de contrasenya	61
5.6.10 Finestra de comandes ja recollides de l'usuari	62
5.6.11 Finestra de gestió de comerç	63
5.6.12 Finestra de gestió de productes del comerç	64
5.6.13 Finestra d'informació de comanda	65
6. Implementació	66
6.1 Tecnologies i llenguatges emprats	66
6.1.1 Java	66
6.1.2 JSON	66
6.1.2 Hibernate JPA	66
6.2 Eines de desenvolupament	67
6.2.1 Spring framework	67
6.2.2 Android Studio	68
6.2.3 Postman	68
6.2.4 Github	68
6.3 Crides implementades	69
6.3.1 Crides a Productes	69
6.3.2 Crides a Comerços	71
6.3.3 Crides a Usuaris	73
6.3.4 Crides a Comandes	76
7. Proves	78
7.1 Proves a l'aplicació mòbil	78
7.2 Proves a l'aplicació de serveis web	78
8. Aspectes legals	79
8.1 Lleis aplicables al projecte	79
8.2 Llicències	79
9. Seguiment del projecte	80
9.1 canvis de planificació i pressupost	80
9.2 Execució real	80

10. Sostenibilitat	81
10.1 Econòmica	81
10.2 Social	81
10.3 Ambiental	81
11. Conclusions i treball futur	83
11.1 Competències tècniques i relació amb l'especialitat de software	83
11.2 Conclusions personals	84
11.3 Treball futur	84
11.3.1 Millorar la seguretat de les crides	84
11.3.2 Sistema de puntuació	84
11.3.3 Trobar un grup de comerciants per provar l'aplicació	84
Referències	86

1.Introducció

1.1 Introducció

Aquest projecte és un Treball de Fi de Grau (TFG) de l'especialitat d'enginyeria de software de la Facultat d'Informàtica de Barcelona (FIB) de la Universitat Politècnica de Catalunya (UPC).

En el projecte s'utilitzen els conceptes apresos al grau per desenvolupar una aplicació que ha de permetre realitzar compres a botigues i mercats de proximitat.

Es tracta d'un TFG en modalitat B que s'ha realitzat durant les pràctiques curriculars a Everis.

Quan vaig entrar a Everis[1] per a les pràctiques curriculars se'm va assignar a un projecte amb tasques relacionades amb l'especialitat però que no tenien l'entitat per a ser un projecte d'un TFG. Parlant d'aquest tema amb el director del TFG a l'empresa varem decidir que el millor era que realitzés un projecte fora de l'àmbit de Everis.

Per interès personal, vaig decidir fer un projecte centrat en ajudar el comerç de barri, per tal d'ajudar aquest tipus de comerç a millorar les seves possibilitats en el mercat online.

1.2 Context

El projecte s'ha realitzat en plena pandèmia global de Covid 19. L'aïllament social, la pèrdua de presencialitat i la crisi causada per la pandèmia global, ha fet que el volum de compres a través d'aplicacions i pàgines web, hagi augmentat molt, en detriment de les compres presencials.

En el moment de l'inici del projecte, ens trobavem en una situació on les aplicacions dedicades a realitzar compres a través de les xarxes eren propietats de grans empreses. El petit comerç es pot inscriure en aquestes aplicacions a canvi de comissions altes i que deixen poc o cap marge de benefici a aquest tipus de comerç.

Les botigues tradicionals han hagut de buscar maneres per a adaptar-se a les circumstàncies per evitar la desaparició. Algunes han optat per millorar la comunicació amb els clients facilitant la possibilitat de fer comandes per whatsapp, i lliuraments a domicili. Però molt d'aquest comerç petit encara no disposa de cap plataforma que els faciliti el poder vendre a través de les xarxes.

El projecte tracta d'ajudar al petit comerç en aquest context, permetent que tinguin una plataforma per entrar al món digital.

1.3 Glossari

1. **Riders:** Treballadors que es donen d'alta a plataformes digitals i treballen repartint paquets per la plataforma.
2. **API:** Acrònim de "Application programming interface", interfície que permet la comunicació entre dos sistemes, en aquest cas, la comunicació entre una aplicació mòbil i una base de dades.
3. **Android Studio:** Entorn de desenvolupament (IDE) d'aplicacions mòbils en Android creat i gestionat per Google.
4. **PostgreSQL:** Sistema per gestionar bases de dades relacionals.
5. **Postman:** Aplicació que ens permet realitzar proves d'una API, en concret el que ens permet és provar les crides HTTP de l'API.
6. **SQL:** Llenguatge de programació usat per definir i manipular bases de dades relacionals.
7. **Swagger:** Framework que s'utilitza per generar documentació sobre APIs, també permet provar les crides a aquesta.
8. **Backend:** S'anomena així a la part d'un sistema software que inclou la implementació de tota la lògica d'una aplicació.
9. **Frontend:** S'anomena així a la part d'un sistema software que inclou la implementació de a les parts visibles que veu l'usuari d'una aplicació de software.
10. **Spring Framework:** Framework de codi obert usat per realitzar aplicacions en JAVA.
11. **Hibernate:** Eina de mapeig que ens permet mapejar objectes en JAVA a un model relacional tradicional.
12. **JPA:** Acronim de "Java persistence API", és una eina de mapeig que facilita el mapeig entre les dades que es manipulen en les aplicacions mitjançant objectes en JAVA i les dades emmagatzemades en les bases de dades relacionals.
13. **Github:** Plataforma online que ens permet allotjar projectes i tenir un control de versions sobre ells.
14. **Barcelona Activa:** És un servei gratuït de l'ajuntament de Barcelona que inclou serveis pels habitants de la ciutat. Té un departament de software que s'encarrega de realitzar projectes per millorar la vida dels habitants de la ciutat.
15. **Firebase:** Plataforma digital de Google que ens ajuda a desenvolupar aplicacions web o mòbil. Firebase és una API que ens ajuda a automatitzar processos com el registre de l'usuari o l'inici de sessió d'aquest.

16. Kotlin: Llenguatge de programació pensat per funcionar amb màquines virtuals de Java i Android.

1.4 Objectius

L'objectiu general d'aquest TFG és crear una aplicació per permetre al petit comerç tenir una plataforma que el faci visible a Internet i que faciliti la venda dels seus productes també a través de la xarxa.

L'objectiu general es pot descomposar en els subobjectius següents:

- L'aplicació ha de facilitar la localització de comerços propers.
- L'aplicació ha de facilitar tota la informació i productes d'un comerç.
- L'aplicació ha de permetre realitzar comandes de productes.
- L'aplicació ha de tenir un espai personal on els propietaris d'un comerç puguin veure les comandes sol·licitades i es permeti modificar els productes oferts.

1.5 Parts interessades

En aquesta secció es descriu qui són les parts interessades en l'aplicació resultant del TFG.

1.5.1 Desenvolupador del projecte

El desenvolupador ha sigut la persona encarregada de desenvolupar una aplicació capaç de complir els objectius descrits anteriorment. Aquesta persona també ha sigut la persona encarregada de realitzar el testeig de l'aplicació.

També ha sigut la persona que s'ha posat en contacte amb comerciants i usuaris per trobar els requisits de l'aplicació. En aquest cas la persona encarregada de realitzar aquest rol, ha sigut la mateixa persona que ha realitzat el TFG.

1.5.2 Botiguers i propietaris de comerços

Els propietaris de comerços són els actors que tenen els productes i volen utilitzar l'aplicació per aconseguir possibles clients. Són els principals beneficiaris de l'aplicació. L'objectiu dels propietaris és el d'oferir els seus productes i contactar amb els clients rebent les seves comandes i fent el lliurament d'aquestes.

1.5.3 Els clients

Els clients són possibles compradors de productes dels comerços, són els principals actius de l'aplicació ja que sense ells l'aplicació no té cap valor. El seu objectiu és el de realitzar comandes als comerços i anar a recollir-los.

1.5.4 Administradors

Per tal de que aquesta aplicació pugui funcionar, cal que hi hagi alguna entitat que se'n faci càrrec, i que s'encarregui de que l'aplicació estigui en funcionament, i que certifiqui els comerços que s'apunten a la plataforma. També ha de ser una entitat que s'encarregui de que l'aplicació evolucioni, per tal d'adaptar-se als canvis que vagin sortint en el context del comerç de barri. En aquests moments no hi ha una entitat que es faci càrrec de l'aplicació resultat d'aquest projecte, però sembla que aquest paper el podria fer una associació de comerciants d'un barri o ciutat, o una junta de botigues d'un mercat.

1.6 Anàlisi del mercat existent

Per realitzar el projecte, s'ha analitzat les aplicacions que hi havien en el mercat a l'inici de la realització del projecte, les aplicacions analitzades són aplicacions que permeten realitzar compres directes a comerços o que permeten promocionar productes dels comerços.

1.6.1 Aplicació *Lacompra*

Lacompra[2] és una aplicació mòbil centrada a realitzar compres de proximitat a la ciutat de Girona. L'aplicació permet comprar productes alimentaris de botigues centrades a l'àrea metropolitana de Girona.

L'aplicació permet realitzar compres de menjar o beguda de diferents comerços de la ciutat, un cop realitzada la comanda un "rider" porta la comanda al domicili de l'usuari en un termini màxim de 24h.

El gran avantatge de l'aplicació és que et permet realitzar comandes incloent productes de diferents comerços, l'usuari rep tots els productes com una sola comanda independentment del comerç d'on provenguin els productes.

Un dels problemes de l'aplicació és que actualment només es pot comprar a comerços de la ciutat Girona. És a dir, una persona que viu en una altra ciutat, no pot usar aquesta mateixa aplicació per comprar en comerços de la seva ciutat.

Un altre problema de l'aplicació és l'abast del mercat, només se centra en el comerç alimentari. Els comerços que són d'un altre sector no estan inclosos en l'aplicació. Aquests comerços no es poden beneficiar de l'aplicació.

1.6.2 Aplicació *TooGoodtogo*

TooGoodToGo [3] és una aplicació per comprar packs de productes sobrants a comerços. Aquests productes són packs de productes que el comerç ha de llençar a la brossa al final del dia.

Aquests packs poden tenir una gran quantitat de productes de temàtiques molt diferents. Per exemple, poden incloure desde menjar a punt de caducar fins a flors que no s'han venut durant alguna celebració pasada.

L'avantatge és que el pack de productes està a un preu molt més reduït que el preu de mercat.

Un dels problemes de l'aplicació es que l'usuari no pot triar què conté el pack de productes, normalment són packs sorpresa on l'usuari només sap la temàtica. Això pot comportar que, per exemple, un pack de productes d'un forn només contingui pastes o dolços i no contingui pa, que pot ser el que interessa a la persona que l'ha demanat.

L'aplicació marca una franja horària concreta per anar a buscar el pack de productes. En cas que l'usuari no reculli els productes, l'usuari perd la oportunitat d'adquirir-los.

1.6.3 Pàgines d'associacions de comerciants

Les associacions de comerciants de Barcelona disposen d'una pàgina web amb tota la informació dels comerços que la formen. La intenció d'aquestes pàgines és la de promoure les compres a les botigues que hi formen part. La majoria d'aquestes pàgines tenen el mateix contingut.

Per realitzar l'anàlisi s'ha visitat les pàgines de tres associacions de comerciants de Barcelona, aquestes associacions són l'associació de comerciants de les *Corts de Sant Ramon i Maternitat*[4], l'associació de comerciants *Cor de les Corts*[5], l'*Eix comercial del carrer de Sants*[6] i l'*Associació de Comerciants de la Marina*[7].

El principal problema d'aquestes pàgines és la impossibilitat de realitzar comandes des de la mateixa pàgina web, les pàgines permeten ubicar les botigues membres en un mapa dinàmic. En el cas que un usuari vulgui realitzar una comanda, aquest usuari ha d'accedir a la pàgina pròpia del comerç, no es poden realitzar compres des de la mateixa pàgina de l'associació. Això també impossibilita la realització d'una comanda que inclogui productes de diferents comerços.

Per acabar, no abarca més que els comerços membres i no hi ha possibilitat de realització de comandes a domicili.

1.7 Possibles obstacles

- La limitació del temps és un gran obstacle. El projecte estava pensat per ser realitzat en un termini de 4 mesos (juny 2021), finalment es va decidir ampliar el termini d'entrega del projecte a octubre del 2021 degut a la falta de temps.
- El factor humà també pot ser un obstacle, en cas de no aconseguir botigues i associacions que puguin ajudar a definir requisits, i puguin valorar i donar opinions sobre el projecte. S'ha intentat mantenir el contacte fins al final del desenvolupament del projecte per aconseguir feedback constant. És molt important mantenir el contacte amb aquests grups ja que ens permet especificar requisits i comprovar el

seu correcte funcionament. I també perquè si l'aplicació resultant del TFG tingués continuïtat hi hauria d'haver una d'aquestes associacions que se'n fes càrrec.

En qualsevol moment la relació amb aquestes persones externes es podia trencar ja sigui per la pèrdua d'interès en el projecte o per interessos econòmics, poden dificultar l'execució del projecte.

- Un altre obstacle del projecte pot ser el tractament de les dades personals i la privacitat. S'ha de procurar mantenir el mínim d'informació sobre els clients i propietaris de botigues per assegurar la seva privacitat. S'ha d'evitar que tercers obtinguin aquesta informació per mètodes il·lícits.

1.8 Metodologia

1.8.1 Metodologia de treball

El projecte s'ha dividit en quatre parts, aquestes són: la fase inicial, la fase de planificació i especificació de requisits, la fase de desenvolupament i la fase final on s'ha realitzat la comprovació del correcte funcionament del projecte, l'escrit final de la memòria i la defensa d'aquest.

La fase inicial, és la fase encarregada de preparar el projecte i encarrilar-lo definint el context, l'abast, objectius, planificació, costos i sostenibilitat seguint el mòdul de GEP[8] de la FIB. També s'ha realitzat la cerca de contactes de associacions de comerciants i comerços i la creació d'un entorn de treball on es s'han posat a punt tots els materials tant de hardware com de software per a la realització del projecte.

En la segona fase, la fase de planificació o especificació de requisits, s'ha realitzat l'especificació dels requisits funcionals i no funcionals del projecte, el disseny lògic de la base de dades i les crides de l'API.

La fase de desenvolupament s'ha realitzat seguint una metodologia *agile*[9]. S'han anat implementant petites parts del sistema i s'ha anat provant el seu correcte funcionament. Les parts desenvolupades han sigut la implementació de la base de dades, l'API amb totes les seves crides, la implementació de l'aplicació mòbil amb *Android Studio*[10] i la integració final de les crides de l'API amb l'aplicació mòbil.

Finalment, s'ha finalitzat el projecte amb una última fase dedicada a comprovar el correcte funcionament de l'aplicació, el que s'ha fet és ensenyar l'aplicació a persones externes i deixar que aquestes provin les seves funcionalitats, en cas de trobar alguna errada, s'han realitzat canvis pertinents. També s'ha realitzat la finalització de l'escriptura de la memòria i la posterior defensa del projecte.

1.8.2 Mètode de validació

S'han realitzat reunions periòdiques amb el tutor del projecte i amb propietaris de comerços per rebre feedback i poder assegurar que s'està realitzant el projecte segons el previst.

Al final de cada tasca d'implementació s'ha realitzat una avaluació dels resultats per veure si s'han assolit tots els objectius plantejats i poder avançar a la següent tasca.

Per comprovar el funcionament de la base de dades i les crides de l'API implementada usant PostgreSQL i *Postman*[11].

El que s'ha fet és comprovar que la base de dades s'ha generat correctament i es podia tractar la informació allotjada al servidor extern on s'allotja la base de dades. Per visualitzar la informació i tractar-la s'ha usat PostgreSQL i SQL directament sobre la base de dades.

Per provar el funcionament de les crides de l'API s'ha usat *Postman*, el que s'ha fet és comprovar el funcionament de les crides, també s'ha generat un registre de totes les crides per optimitzar la seva utilització.

Les crides de l'API s'han documentat usant *Swagger*[12], les crides es poden veure a la següent direcció:

<https://backendproximitat.herokuapp.com/swagger-ui.html#/user-controller>

Per comprovar el correcte funcionament de l'aplicació s'ha deixat provar aquesta a persones alienes al projecte, s'ha rebut un feedback i s'han realitzat els canvis pertinents.

1.9 Organització de la resta de la memòria

La memòria s'organitza en els següents capítols:

- Capítol 1 Introducció: Capítol introductor on s'explica el context del projecte.
- Capítol 2 Planificació inicial: Capítol on s'explica les tasques que s'han realitzat en el projecte i planificació temporal d'aquest.
- Capítol 3 Gestió econòmica: Capítol on es descriu i s'argumenta el cost del projecte.
- Capítol 4 Especificació dels requisits: Capítol on es descriu els requisits funcionals i no funcionals del projecte, així com s'argumenta la seva obtenció i els casos d'ús corresponents.
- Capítol 5 Arquitectura del sistema: Capítol on es descriu i s'argumenta l'arquitectura física i lògica del sistema, els patrons utilitzats, els diagrames de seqüència i el disseny lògic i físic de la base de dades.
- Capítol 6 Implementació: Capítol on es descriu i s'argumenta els llenguatges utilitzats per la implementació del sistema, les eines utilitzades i les crides implementades a l'aplicació de serveis web.

- Capítol 7 Proves: Capítol on es descriu les proves realitzades per comprovar el correcte funcionament del sistema.
- Capítol 8 Aspectes legals del projecte: Capítol on es descriu les lleis aplicables al projecte i les llicències utilitzades durant la implementació d'aquest.
- Capítol 9 Seguiment del projecte: Capítol on es descriu el seguiment que s'ha realitzat durant la implementació del sistema.
- Capítol 10 Sostenibilitat: Capítol on es descriu la sostenibilitat econòmica, social i ambiental del projecte realitzat.
- Capítol 11 Conclusions i treball futur: Capítol final on es descriu i s'argumenta les conclusions finals del projecte i el treball futur que es realitzarà en aquest.

2. Planificació inicial

2.1 Calendari

L'estimació de la durada d'aquest projecte és aproximadament de vuit mesos. La data d'inici del projecte és el dia 22 de febrer i la data de finalització és la setmana del 25 a 28 d'octubre (és a dir, és la setmana establerta per realitzar presentacions de TFGs).

2.2 Descripció de les tasques

Les tasques que s'han realitzat seran les següents:

1. Gestió inicial del projecte
2. Cerca i comunicació amb una associació de comerciants
3. Creació d'entorn de treball
4. Especificació
5. Desenvolupament del *backend*
6. Comprovació del funcionament del *backend*
7. Desenvolupament del *frontend*
8. Integració de les crides de l'API al *frontend*
9. Part final, redacció i defensa

2.2.1 Gestió inicial del projecte

Primera tasca del projecte. Aquesta tasca s'ha realitzat mentre es cursava l'assignatura de GEP, s'ha fet és la gestió inicial del projecte.

En concret, el que s'ha fet és seguir les indicacions marcades pel mòdul de GEP, aquest mòdul es divideix en quatre parts, aquestes són: definir el context objectiu i abast del projecte, realitzar la planificació temporal, realitzar el càlcul del pressupost i integrar totes les parts en un document definitiu i realitzar millores amb el feedback rebut pels professors de GEP.

Té una durada estimada de 90 hores i no té cap dependència, s'ha realitzat conjuntament amb la tasca de cerca i comunicació amb una associació de comerciants (2.2.2) i la creació d'un entorn de treball (2.2.3).

2.2.2 Cerca i comunicació amb una associació de comerciants

Mentre es realitzava la tasca de gestió inicial del projecte, s'ha realitzat al mateix temps la tasca de búsqueda d'un contacte a alguna associació de comerciants de Barcelona.

Com ja s'ha dit anteriorment, l'objectiu del projecte és la realització d'una aplicació que permet-hi realitzar compres de proximitat. És interessant trobar el contacte d'aluna associació de comerciants per poder definir els requisits del sistema.

Aquesta tasca s'ha realitzat principalment al principi i al final del projecte, durant la part d'implementació només s'ha realitzat reunions informatives per informar de l'estat del projecte.

Té una durada estimada de 25 hores, al tractar-se d'una tasca inicial no té cap dependència, s'ha de realitzar conjuntament amb la tasca de gestió inicial 2.2.1 i la tasca de crear un entorn de treball 2.2.3

2.2.3 Creació d'entorn de treball

Tasca de *warm up*, aquesta etapa serveix per posar a punt i configurar les eines que s'utilitzen durant la realització del projecte.

Té una durada estimada de 25 hores, es realitzarà conjuntament amb la tasca de cerca i comunicació amb una associació de comerciants (2.2.2) i la gestió inicial del projecte (2.2.1).

2.2.4 Especificació de requisits de l'aplicació

Dins d'aquesta tasca, primer s'han detallat els detallar els requisits funcionals i no funcionals de l'aplicació a desenvolupar.

El que s'ha fet és definir els objectius de l'aplicació amb l'ajuda de comerços i associacions de comerciants, un cop es van obtenir els objectius es van descomposar en requisits funcionals i no funcionals.

Un cop obtingut els requisits funcionals es va procedir a definir els casos d'ús de l'aplicació a desenvolupar i la especificació del model conceptual de les dades necessàries per l'aplicació.

Té una durada estimada de 40 hores, com a dependència té la tasca de la gestió inicial del projecte (2.2.1) i la tasca de cerca i comunicació amb una associació de comerciants (2.2.2).

2.2.5 Desenvolupament del *backend*

La primera part implementada de l'aplicació ha sigut el *backend*, l'objectiu d'aquesta tasca ha sigut la de crear una base de dades i una api rest pel projecte, per fer-ho s'ha usat el model conceptual dissenyat a l'apartat anterior.

S'ha dividit la implementació del *backend* en diferents subtasques, aquestes tasques són:

- Implementació d'una base de dades usant *Spring framework* i *Hibernate*.
- Implementació de les crides de l'API REST amb *Spring* i *JPA*.
- Cerca d'un servidor web per publicar l'aplicació.
- Publicació de l'API i la base de dades al servidor web.

Té una durada estimada de 100 hores, per iniciar aquesta tasca s'ha d'haver realitzat la especificació dels requisits del projecte (2.2.4).

Un cop finalitzat la implementació del backend es realitzarà la comprovació del funcionament de les crides.

2.2.6 Comprovació del funcionament del *backend*

En aquesta part s'ha comprovat el funcionament de la base de dades i de l'API, en concret el que s'ha fet és accedir a la base de dades i comprovar el funcionament d'aquesta utilitzant comandes SQL.

També s'ha usat Postman per comprovar el funcionament de l'API, s'han provat totes les crides i s'ha vist que es pot accedir a la informació de la base de dades a través de les crides.

Es pot separar aquesta tasca en dues subtasques, aquestes són:

- Comprovació del funcionament de la base de dades utilitzant SQL.
- Comprovació del funcionament dels Endpoints de l'API.
- Correcció d'errors.

Té una durada estimada de 30h, per començar a realitzar aquesta tasca s'ha d'haver realitzat la tasca d'implementació del *backend* (2.2.5).

2.2.7 Desenvolupament del *frontend*

Un cop realitzat el *backend* i comprovat el seu correcte funcionament, es va procedir a realitzar la implementació del *frontend*. Per realitzar-ho s'han utilitzat els casos d'ús dissenyats en la tasca d'especificació del sistema.

L'objectiu de la tasca ha sigut la d'implementar una aplicació mòbil amb *Android estudio* que s'atisfagi aquets casos d'ús.

El desenvolupament del *frontend* està dividit en diferents subtasques, aquestes tasques corresponen a diferents parts de la interfície que estan relacionades.

- Implementació dels d'usuaris.
- Implementació dels comerços.
- Implementació de les comandes.
- Implementació d'un mapa per veure l'ubicació dels comerços.
- Comprovació del correcte funcionament de l'aplicació.

Té una durada estimada de 110 hores, per poder realitzar aquesta tasca s'ha d'haver realitzat les tasques de creació d'entorn de treball (2.2.3) i la tasca de la realització d'especificació dels requisits del projecte (2.2.4). L'inici d'aquesta tasca pot coincidir amb la tasca d'implementació del *backend* (2.2.5) i la tasca de comprovació del *backend* (2.2.6).

2.2.8 Integració de les crides de l'API al *frontend*

En aquest apartat s'han realitzat la integració de les crides de l'API al *frontend*, s'ha realitzat conjuntament amb la implementació del *frontend*.

Aquesta tasca s'ha dividit en diferents subtasques, aquestes són:

- Implementació de les crides d'usuari al *frontend*.
- Implementació i integració de les crides de comanda al *frontend*.
- Implementació i integració de les crides de comerç al *frontend*.

Té una durada estimada de 70 hores, per poder realitzar aquesta tasca s'ha d'haver realitzat la tasca d'implementació del *backend* (2.2.5) i la tasca de comprovació del *backend*(2.2.6). Aquesta tasca es realitza simultàniament amb la tasca de la implementació del *frontend* (2.2.7).

2.2.9 Part final, redacció i defensa

S'ha definit una tasca final per realitzar la prova de l'aplicació, la redacció final de la memòria i la preparació de la defensa.

El que s'ha fet és deixar que persones externes al projecte provim l'aplicació, d'aquestes proves tenen la funció de trobar errors i realitzar petites modificacions, així com comprovar que l'aplicació satisfà els requisits definits al principi de la realització del projecte.

El que s'ha fet és deixar que persones externes provin l'aplicació en una prova pilot prepara, aquestes persones han provat l'aplicació i han reportat errors que el desenvolupador ha solucionat.

També s'ha realitzar l'escriptura final de la memòria i la defensa del projecte.

Aquesta tasca només es pot realitzar si s'han completat tota la resta de tasques, té una durada estimada de 45 hores.

2.3 Estimació de temps

La planificació i l'estimació de les tasques és una previsió, hi haurà una certa flexibilitat en la realització de les tasques.

L'estimació total del temps es pot veure a la Taula 1.

Tasques del projecte	Hores
Gestió inicial del projecte	90
Cerca i comunicació amb una associació de comerciants	25
Creació d'entorn de treball	25
Especificació	40
Desenvolupament <i>backend</i>	100
Comprovació del funcionament del <i>backend</i>	30
Desenvolupament <i>frontend</i>	110
Integració de les crides de l'API al <i>frontend</i>	70
Part final, redacció i defensa	45
Total	540

Taula 1: Resum duració cada tasca

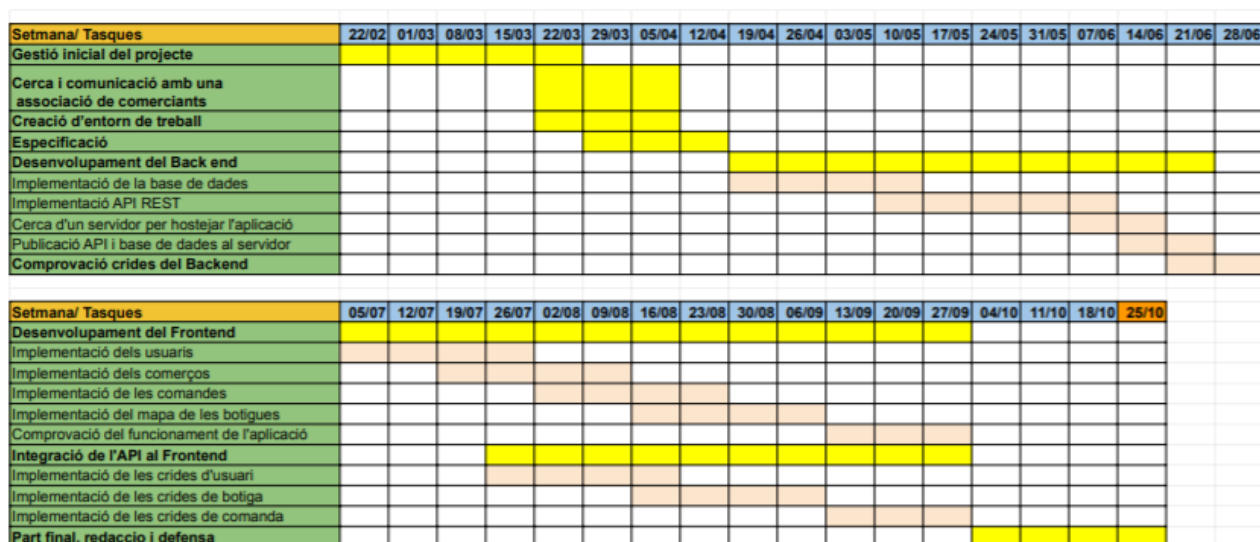
2.4 Pla d'acció

L'estimació d'hores és una previsió, durant la realització del projecte s'ha tingut certa flexibilitat horària. Pot ser que el còmput real de les hores variï de les hores estimades. A cada tasca s'ha afegit hores extres per solucionar possibles problemes que puguin sorgir.

S'ha deixat una setmana de marge abans d'entregar el projecte per tal de solucionar possibles errors que puguin sorgir.

Per a la realització del projecte és imprescindible tenir el suport d'una associació de comerciants de Barcelona, en cas que no sigui possible realitzar la prova pilot o no s'aconsegueixi tenir contacte amb cap associació, es realitzarà el projecte igualment i es realitzaran proves utilitzant comerços i productes no reals.

2.5 Diagrames de Gantt



Taula 2: Diagrama de Gantt del projecte

3. Gestió econòmica

3.1 Pressupost

Els costos del projecte estan directament relacionats amb els recursos descrits a continuació, és a dir, tant els recursos de hardware com de software com també els recursos humans.

Tot i tractar-se d'un projecte acadèmic, es realitzarà un estudi del pressupost del projecte com si es tractés d'un projecte real a realitzar per una empresa o particular.

3.2 Recursos humans

Aquest projecte el realitzarà principalment per una persona, només hi participarà una altra persona, externa al projecte, per ajudar a realitzar la part final de la prova pilot.

El desenvolupador del projecte serà la persona encarregada de realitzar tots els rols necessaris en un projecte de software, aquests rols són els de cap de projecte, analista o arquitecte, programador i tester.

La persona externa que ajudarà a l'última part de testeig i prova pilot, la seva funció serà la d'ajudar a integrar els comerços que vulguin realitzar la prova pilot a l'aplicació.

Aquesta persona tindrà el rol de tester i realitzarà unes 25 hores de treball.

Per poder estimar el pressupost necessari en recursos humans, s'assignarà un preu per hora a cadascun dels rols esmentats. Aquests preus són extrets del preu mitjà dels treballadors de *Everis*. Aquests els podem veure a la *Taula 3*.

Rol	Preu per hora
Cap de projecte	22
Analista o Arquitecte	18
Arquitecte	18
Programador	10
Tester	10

Taula 3: Cost per hora dels diferents rols

A continuació, passem a calcular el cost humà per a cada fase i rol a partir de l'assignació de temps. Es pot visualitzar a la *Taula 4*.

Fase	Dedicació (en hores)					Cost
	cap de projecte	analista	arquitecte	programador	tester	
Gestió inicial del projecte	90					1.980€
Cerca i comunicació amb una associació de comerciants	20	5				530€
Creació d'entorn de treball				25		250€
Especificació		20	20			720€
Desenvolupament del <i>backend</i>			40	60		1320€
Comprovació del funcionament del <i>backend</i>				10	20	300€
Desenvolupament del <i>frontend</i>				110		1100€
Integració de les crides al <i>frontend</i>			20	50		860€
Part final, redacció i defensa	15			10	20	660€
Total			540			7.720€

Taula 4: Estimació del cost per fase i rol.

També podem veure el cost que tindrà cada rol del projecte a la *Taula 5*.

Rol	Hores previstes	preu per hora	Cost estimat
Cap de projecte	125	22	2.750€
Analista	25	18	450€
Arquitecte	80	18	1440€
Programador	265	10	2.650€
Tester	40	10	400€
Total	540	-	7.720€

Taula 5: Estimació de costos per rol.

3.3 Recursos de hardware i software

En aquest apartat es detallaran el cost dels diferents recursos de hardware i software que s'utilitzaran durant el desenvolupament del projecte.

Per establir el cost d'amortització del cost dels materials emprats usem la següent fórmula:

Amortització = $\text{Preu} - (\text{Preu}/(\text{Vida útil}/(\text{Vida útil} - \text{dies treballats})))$.

Per realitzar el cost dels recursos de hardware i software només es tenen en compte els recursos que tenen un cost pel seu ús. Els recursos que no comporten un càrrec al projecte són *Spring Framework 5.3.4*, *Android Studio 4.1.2*, *Postman*, *swagger IU* i *Github*[13].

Aquests recursos disposen de llicències d'estudiants o són gratuïts.

S'ha calculat les hores d'utilització de l'ordinador portàtil serà de 540 hores, la duració completa del projecte.

El temps d'utilització del telèfon mòbil per realitzar les proves de l'aplicació serà de 65 hores, la duració completa del testeig de l'aplicació i la prova pilot.

Les hores d'utilització d'eines de *Microsoft 365*[14] són d'unes 180 h.

A la *taula 6* podem observar els costos d'aquests recursos.

Producte	Preu	temps d'ús	Vida útil	Amortització
Portatil Hp pavilion power 15	800€	540 hores	4 anys	12,32€
Xiaomi Redmi Note 9s	200€	65 hores	2 anys	0,74€
Microsoft Word i Excel	69€	180 hores	1 any	1,41€
Total	-	-	-	14,47€

Taula 6: Estimació de costos de recursos de hardware i software.

3.4 Despeses generals

En aquest apartat es tenen en compte altres despeses. Aquestes despeses són el consum energètic, el cost de la connexió a internet i el cost del transport.

El consum energètic que l'ordinador durant el transcurs de la realització del projecte, s'espera un ús de 450 hores de les 540 hores totals del projecte.

El cost de la connexió a internet amb 100 MB de baixada i 20 MB de pujada.

També s'inclouen les despeses del transport per realitzar els desplaçaments amb transport públic per realitzar reunions o la realització de la prova pilot.

S'utilitzarà el bitllet *T-casual* de 10 viatges. S'espera que es realitzin un mínim de 15 viatges.

A la *taula 7* es pot observar la estimació del preu de les despeses generals

Descripció	Preu	Quantitat	Cost estimat
Consum energètic	0.1€ al Kwh	450 hores	45€
Internet	40€ al mes	4 mesos	120€
Transport	11.35€	2	22,70€
Total	-	-	188€

Taula 7: Estimació de despeses generals.

3.5 Despeses total

En cas que hi hagi algun problema en els recursos esmentats anteriorment, o faci falta incorporar un nou recurs durant l'execució del projecte, s'estableix com a mesura de contingència un marge del 10% sobre el cost total del projecte.

El cost total del projecte es pot visualitzar a la *Taula 8*.

Recurs	Cost estimat
Recursos humans	7.720€
Recursos de hardware i software	14,47€
Despeses generals	187,70€
Cost parcial	7.922,17
Contingències	10%
Cost total	8.714,40

Taula 8: Cost total del projecte.

3.6 Control de gestió

L'estimació temporal del cost humà, l'encarregat de realitzar la seva gestió ha sigut l'autor del projecte. Aquest ha sigut l'encarregat de gestionar les hores dedicades al projecte, per tant, s'ha fet un control de les hores per no sobrepassar l'estimació realitzada.

Al final de cada tasca s'han contat les hores dedicades per veure possibles desviacions al programa establert, i si cal realitzar una modificació del pressupost.

Pel que respecte al cost de hardware, software i despeses generals s'ha fet un seguiment destinat a no superar els costos d'aquests apartats en cada iteració, si es sobrepassen els costos d'aquestes seccions es realitzarà una investigació per descobrir la font de l'excés i s'ha compensat en la següent iteració.

3.7 Costos imprevistos

Pel que fa a costos imprevistos, s'ha calculat segons els riscos que pot tenir aquest projecte.

El primer obstacle és la limitació del temps, en cas que s'hagi de destinar més hores al projecte segurament s'hagi de realitzar durant la fase de desenvolupament. El preu per hora és de 10€, aquest és el preu mitjà que un programador té a *Everis*. En el cas que s'hagi de realitzar més hores de les previstes, el preu seria de 500 € (50 hores extres).

En cas que hi hagués algun problema amb el contacte amb l'associació de comerciants (segon obstacle esmentat en l'apartat 2.2 Possibles obstacles) i no es pugui realitzar la prova pilot, llavors es provaria l'aplicació amb comerços no reals, això suposaria que la persona externa dedicada a ajudar a gestionar la prova pilot, no hauria de realitzar la seva part, reduint el nombre d'hores del projecte en 35.

Els altres obstacles esmentats no comporten cap altre augment del pressupost del projecte.

4. Especificació de requisits

4.1 Procés d'obtenció de requisits

Per obtenir els requisits funcionals del projecte es va entrar en contacte amb propietaris de comerços a Barcelona, en concret, l'Associació de Comerciants de la Marina[7]. Aquesta associació es va mostrar molt interessada en el projecte. Un cop trobat el contacte es va procedir a realitzar tres reunions per definir els requisits del projecte.

Les reunions es van realitzar de forma telemàtica amb la presidenta de l'associació, la secretaria de l'associació i la persona que havia fet d'intermediaria entre el desenvolupador del projecte i l'associació.

En un principi, l'associació es va mostrar interessada en el projecte i fins i tot es va interessar en realitzar una prova pilot un cop finalitzat el projecte.

Un dels punts que es van tractar, van ser les ofertes que havia rebut l'associació, en la fase inicial de l'aplicació, l'associació havia tingut ofertes per part de *Barcelona Activa*[15] i un centre de formació professional per realitzar un projecte semblant.

En aquestes reunions es van especificar quins requisits els interessa per a una aplicació com la que és objectiu d'aquest projecte:

- Es vol que els clients puguin realitzar comandes a un comerç.
- Es vol que les comandes no puguin incloure productes de diferents comerços.
- Es vol que l'aplicació permeti informar als clients de quina és la millor hora per anar a buscar una comanda a un comerç.
- Es vol que un client faci els pagaments de les comandes presencialment al recollir la comanda.
- Es vol que inicialment l'aplicació no permeti demanar que la comanda es reparteixi a domicili. Aquest requisit ve motivat perquè l'associació no pot assegurar que totes els comerços que la formen tinguin servei de repartiment. En cas que l'aplicació funcionés correctament aquest servei s'implementaria un cop l'associació trobes un servei extern de repartiment a domicili.
- Es vol una aplicació entenedora i fàcil d'usar. Aquest requisit ve motivat perquè gran part dels clients i propietaris potencials dels comerços seran gent amb pocs o nuls coneixements d'informàtica.
- Es vol que els propietaris puguin veure les comandes que s'han demanat a través de l'aplicació i que siguin els botiguers els encarregats de marcar una comanda com entregada.

- Es vol que els clients puguin veure en un mapa els comerços que ofereixen productes a través de l'aplicació.

4.2 Requisits funcionals

Un cop realitzades les reunions per descriure els requisits que l'associació considerava que eren els rellevants per a l'aplicació, es va procedir a definir un llistat de requisits funcionals que es podien incloure en la primera versió de l'aplicació objectiu del TFG.

Primer de tot diferenciem els usuaris que tenim a l'aplicació. Al sistema hi ha dos rols, el rol de client i el rol de botiguer, tots dos tenen funcionalitats compartides a l'aplicació, la diferència està en que un botiguer té algunes funcionalitats exclusives que el client no tindrà disponibles.

En el document ens referim a usuaris com al conjunt de botiguers i clients. També ens referim als dos rols per separat com a client i botiguer.

4.2.1 Llista de requisits funcionals

- L'aplicació ha de permetre a un nou usuari registrar-se a l'aplicació.
- L'aplicació ha de permetre a un usuari loguejar-se usant les seves credencials.
- L'aplicació ha de mostrar en un mapa els comerços adscrits a l'aplicació i permetre accedir a ells a clicant sobre el seu icona.
- L'aplicació ha de permetre buscar un comerç a través del seu nom.
- L'aplicació ha de mostrar la informació dels comerços inscrits i un llistat dels productes que tenen a la venda.
- L'aplicació ha de permetre que un usuari pugui realitzar comandes a un comerç.
- L'aplicació ha de permetre a un usuari veure les comandes que ha demanat i les comandes que ja ha recollit.
- L'aplicació ha de permetre als botiguers poder gestionar el seu comerç i modificar la informació del comerç que són propietaris.
- L'aplicació ha de permetre als botiguers modificar la informació dels productes que ofereix el seu comerç, i especialment el seu preu.
- L'aplicació ha de permetre als botiguers veure les comandes que han d'entregar i les comandes ja entregades del seu comerç.

- L'aplicació ha de permetre modificar la informació personal de l'usuari.

4.2.2 Diagrames de casos d'ús

Com s'ha comentat a l'apartat anterior, hi ha dos rols diferents que podran usar l'aplicació, el rol de client i el rol botiguer.

Tots els usuaris de l'aplicació tenen el rol d'usuari i només els usuaris que siguin propietaris d'un comerç tenen el rol de botiguer.

Els botiguers tindran disponibles les funcionalitats dels clients a més de funcionalitats que exclusives del seu rol.

En el diagrama 1 i 2 es pot observar les activitats que desenvolupen aquests rols.

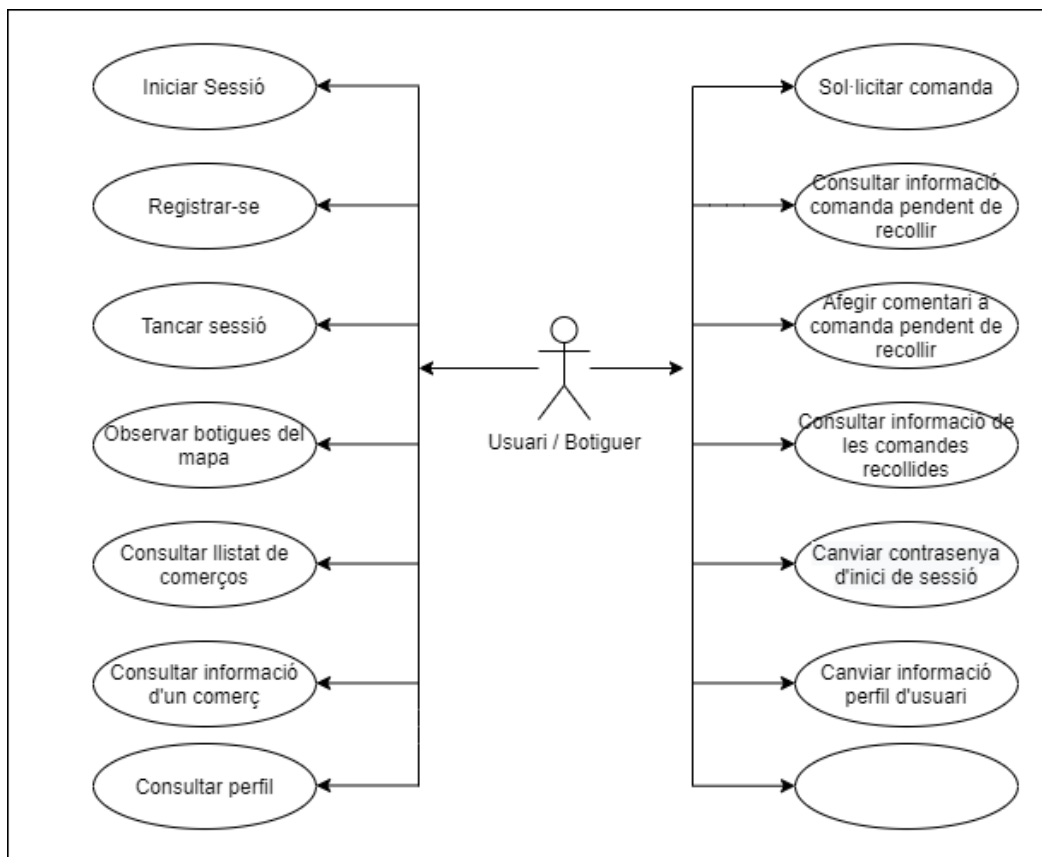


Diagrama 1: Casos d'ús dels rols d'usuari i botiguer

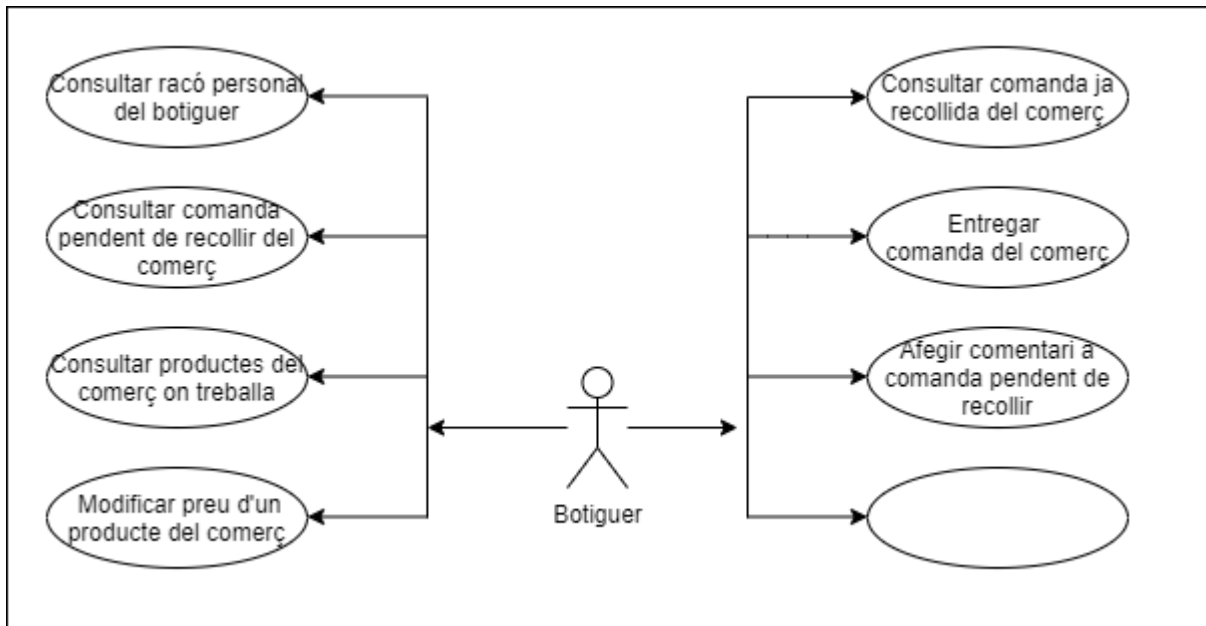


Diagrama 2: Casos d'ús del rol de botiguer

4.2.3 Descripció dels casos d'ús

4.2.3.1 Casos d'ús de clients i botiguers

Cas d'ús	Iniciar sessió
Actor	Usuari o botiguer
Precondicions	L'usuari no ha iniciat sessió.
Disparador	L'usuari vol iniciar sessió.
Procediment	<ol style="list-style-type: none"> 1. L'usuari introdueix les seves credencials 2. L'usuari prem el botó per continuar. 3. L'aplicació comprova que les dades són correctes <ol style="list-style-type: none"> a. Si són correctes es redirigeix a l'usuari a la pàgina principal de l'aplicació b. Si són incorrectes s'informa a l'usuari que ha introduït dades errònees.

Taula 9: Cas d'ús per iniciar sessió.

Cas d'ús	Registrar-se
Actor	Usuari o botiguer
Precondicions	L'usuari no està registrat a l'aplicació.
Disparador	L'usuari vol registrar-se a l'aplicació.
Procediment	<ol style="list-style-type: none"> 1. L'usuari prem el botó registrar. 2. L'usuari introdueix totes les seves dades. 3. L'usuari prem el botó per continuar. 4. L'aplicació comprova que les dades són correctes <ol style="list-style-type: none"> a. Si són correctes es redirigeix a l'usuari a la pàgina principal de l'aplicació i es registra l'usuari a la base de dades. b. Si són incorrectes s'informa a l'usuari que ha introduït dades errònees.

Taula 10: Cas d'ús per registrar-se.

Cas d'ús	Tancar sessió
Actor	Usuari o botiguer
Precondicions	L'usuari està registrat a l'aplicació.
Disparador	L'usuari vol tancar sessió a l'aplicació.
Procediment	<ol style="list-style-type: none"> 1. L'usuari prem els tres punts de dalt a la dreta desde qualsevol punt de l'aplicació. 2. L'usuari prem el botó "log out". 3. L'usuari és redirigit a la pantalla d'inici de l'aplicació.

Taula 11: Cas d'ús per tancar sessió

Cas d'ús	Observar botigues del mapa
Actor	Usuari o botiguer
Precondicions	L'usuari està registrat a l'aplicació.
Disparador	L'usuari vol observar els comerços propers a ell.
Procediment	<ol style="list-style-type: none"> 1. L'usuari prem el segon botó inferior del menú principal amb la icona de mapa. 2. L'usuari és redirigit a la finestra del mapa 3. L'usuari busca les botigues properes a ell. 4. Prem sobre elles si vol obtenir més informació.

Taula 12: Cas d'ús per observar les botigues del mapa

Cas d'ús	Consultar llistat de comerços de l'aplicació
Actor	Usuari o botiguer
Precondicions	L'usuari està registrat a l'aplicació.
Disparador	L'usuari vol consultar el llistat de comerços.
Procediment	<ol style="list-style-type: none"> 1. L'usuari prem el tercer botó inferior del menú principal amb la icona de lupa. 2. L'usuari és redirigit a la finestra amb el llistat de comerços 3. L'usuari pot filtrar els comerços per nom.

Taula 13: Cas d'ús per consultar el llistat de comerços de l'aplicació

Cas d'ús	Consultar informació d'un comerç
Actor	Usuari o botiguer
Precondicions	L'usuari està registrat a l'aplicació.
Disparador	L'usuari vol consultar el llistat de comerços.
Procediment	<ol style="list-style-type: none"> 1. L'usuari prem el tercer botó inferior del menú principal amb la icona de lupa. 2. L'usuari és redirigit a la finestra amb el llistat de comerços 3. L'usuari prem sobre el comerç que vol consultar. 4. Se li mostra a l'usuari tota la informació sobre el comerç que ha seleccionat

Taula 14: Cas d'ús per consultar la informació d'un comerç

Cas d'ús	Sol·licitar comanda
Actor	Usuari o botiguer
Precondicions	L'usuari està registrat a l'aplicació.
Disparador	L'usuari vol realitzar una comanda.
Procediment	<ol style="list-style-type: none"> 1. L'usuari prem el tercer botó inferior del menú principal amb la icona de lupa. 2. L'usuari és redirigit a la finestra amb el llistat de comerços 3. L'usuari prem sobre el comerç que vol consultar. 4. Se li mostra a l'usuari tota la informació sobre el comerços de l'aplicació. 5. L'usuari prem el botó "veure productes" 6. Se li mostra a l'usuari un llistat de productes 7. L'usuari selecciona els productes que vol realitzar la comanda. <ol style="list-style-type: none"> a. Si l'usuari prem el botó realitzar comanda però no ha seleccionat cap producte, s'informa l'usuari que no ha seleccionat cap producte. 8. L'usuari prem el botó "realitzar comanda" 9. Es redirigeix l'usuari a la finestra de confirmar comanda <ol style="list-style-type: none"> a. Si l'usuari està conforme amb la comanda, l'usuari prem el botó de confirmar i es genera la comanda. b. Si la comanda no complau a l'Usuari, l'usuari prem el botó de cancel·lar i es cancel·la la comanda.

Taula 15: Cas d'ús per sol·licitar una comanda

Cas d'ús	Consultar informació comanda pendent de recollir
Actor	Usuari o botiguer
Precondicions	L'usuari està registrat a l'aplicació.
Disparador	L'usuari vol veure la informació d'una comanda pendent de recollir.
Procediment	<ol style="list-style-type: none"> 1. L'usuari prem el primer botó inferior del menú principal. 2. L'usuari és redirigit a la finestra amb el llistat de comandes que té pendents de recollir. <ol style="list-style-type: none"> a. Si l'usuari no té cap comanda pendent per recollir, apareix un text informant-lo. b. Si l'usuari té comandes pendent de recollir, apareix un llistat amb la informació bàsica. 3. L'usuari prem sobre la comanda que vol consultar la informació. 4. L'usuari és redirigit a una finestra amb tota la informació de la comanda. 5. L'usuari pot prem el boto "veure productes" per veure quins productes té la comanda.

Taula 16: Cas d'ús per consultar la informació d'una comanda ja demanada

Cas d'ús	Afegir comentari a comanda pendent de recollir
Actor	Usuari o botiguer
Precondicions	L'usuari està registrat a l'aplicació.
Disparador	L'usuari vol afegir un comentari a una comanda pendent de recollir.
Procediment	<ol style="list-style-type: none"> 1. L'usuari prem el primer botó inferior del menú principal. 2. L'usuari és redirigit a la finestra amb el llistat de comandes que té pendents de recollir. <ol style="list-style-type: none"> a. Si l'usuari no té cap comanda pendent per recollir, apareix un text informant-lo. b. Si l'usuari té comandes pendent de recollir, apareix un llistat amb la informació bàsica. 3. L'usuari prem sobre la comanda que vol consultar la informació. 4. L'usuari és redirigit a una finestra amb tota la informació de la comanda. 5. L'usuari pot prem el botó "veure comentaris" per veure quins comentaris té la comanda. 6. L'usuari afegeix un text amb el comentari que vol afegir i prem el botó "modificar comentari" 7. S'afegeix el comentari a la comanda.

Taula 17: Cas d'ús per afegir un comentari a una comanda

Cas d'ús	Consultar informació comanda recollida
Actor	Usuari o botiguer
Precondicions	L'usuari vol consultar la informació d'una comanda recollida.
Disparador	L'usuari vol realitzar una comanda.
Procediment	<ol style="list-style-type: none"> 1. L'usuari prem l'últim botó inferior del menú principal. 2. L'usuari és redirigit a la finestra amb la informació del perfil de l'usuari 3. Se li mostra un llistat amb les comandes recollides de l'usuari. L'usuari prem sobre la comanda que vol consultar la informació. 4. L'usuari és redirigit a una finestra amb tota la informació de la comanda. 5. L'usuari pot preme el boto "veure productes" per veure quins productes té la comanda.

Taula 18: Cas d'ús per consultar la informació d'una comanda recollida

Cas d'ús	Consultar perfil
Actor	Usuari o botiguer
Precondicions	L'usuari està registrat a l'aplicació.
Disparador	L'usuari vol consultar la seva informació.
Procediment	<ol style="list-style-type: none"> 1. L'usuari prem el quart botó inferior del menú principal. 2. L'usuari és redirigit a la finestra del perfil on pot consultar la seva informació.

Taula 19: Cas d'ús per consultar el perfil d'usuari

Cas d'ús	Canviar contrasenya d'inici de sessió
Actor	Usuari o botiguer
Precondicions	L'usuari està registrat a l'aplicació.
Disparador	L'usuari vol canviar la contrasenya per accedir a l'aplicació.
Procediment	<ol style="list-style-type: none"> 3. L'usuari prem el quart botó inferior del menú principal. 1. L'usuari és redirigit a la finestra del perfil on pot consultar la seva informació. 2. L'usuari prem el botó "canviar contrasenya". 3. L'usuari és redirigit a la finestra per canviar la contrasenya 4. L'usuari introdueix la nova contrasenya i prem el botó "guardar" 5. Es valida la seguretat de la nova contrasenya. <ol style="list-style-type: none"> a. Si la contrasenya compleix els requisits de seguretat, l'usuari és redirigit a la finestra principal. b. Si la nova contrasenya no compleix els requisits s'informa a l'usuari i no es realitza cap canvi.

Taula 20: Cas d'ús per canviar la contrasenya d'inici de sessió

Cas d'ús	Canviar informació del perfil d'usuari
Actor	Usuari o botiguer
Precondicions	L'usuari està registrat a l'aplicació.
Disparador	L'usuari vol canviar la seva informació personal.
Procediment	<ol style="list-style-type: none"> 1. L'usuari prem el quart botó inferior del menú principal. 2. L'usuari és redirigit a la finestra del perfil on pot consultar la seva informació. 3. L'usuari prem el botó "modificar dades personals". 4. L'usuari és redirigit a la finestra per canviar les dades personals 5. L'usuari introdueix la informació que vol canviar i prem el botó "guardar" 6. Es modifica la informació de l'usuari i és redirigit a la finestra principal de l'aplicació

Taula 21: Cas d'ús per canviar la informació del perfil de l'usuari

4.2.3.1 Casos d'ús del rol botiguer

Cas d'ús	Consultar racó personal del botiguer
Actor	Botiguer
Precondicions	L'usuari està registrat a l'aplicació i és botiguer d'un comerç.
Disparador	El botiguer d'un comerç vol veure el racó personal de botiguer.
Procediment	<ol style="list-style-type: none"> 1. El botiguer prem l'últim botó inferior del menú principal. 2. El botiguer és redirigit a la finestra del perfil on pot consultar la seva informació. 3. El botiguer prem el botó "El meu comerç". 4. L'usuari és redirigit a l'espai personal del seu comerç.

Taula 22: Cas d'ús per consultar el racó del botiguer

Cas d'ús	Consultar comanda pendent de recollir del comerç
Actor	Botiguer
Precondicions	L'usuari està registrat a l'aplicació i és botiguer d'un comerç.
Disparador	El botiguer d'un comerç vol consultar una comanda pendent de recollir del comerç on és botiguer.
Procediment	<ol style="list-style-type: none"> 1. El botiguer prem l'últim botó inferior del menú principal. 2. El botiguer és redirigit a la finestra del perfil on pot consultar la seva informació. 3. El botiguer prem el botó "El meu comerç". 4. El botiguer és redirigit a l'espai personal del seu comerç. 5. El botiguer prem el botó "Comandes" 6. El botiguer es redirigir a una nova finestra on es mostren totes les comandes pendents de recollir amb el seu identificador. 7. El botiguer prem sobre la comanda que vol consultar. 8. El botiguer és redirigit a la finestra amb tota la informació de la comanda

Taula 23: Cas d'ús per consultar una comanda pendent de recollir del comerç

Cas d'ús	Consultar comanda ja recollida del comerç
Actor	Botiguer
Precondicions	L'usuari està registrat a l'aplicació i és botiguer d'un comerç.
Disparador	El botiguer d'un comerç vol consultar una comanda recollida del comerç on és botiguer.
Procediment	<ol style="list-style-type: none"> 1. El botiguer prem l'últim botó inferior del menú principal. 2. El botiguer és redirigit a la finestra del perfil on pot consultar la seva informació. 3. El botiguer prem el botó "El meu comerç". 4. El botiguer és redirigit a l'espai personal del seu comerç. 5. El botiguer prem el botó "Comandes entregades" 6. El botiguer es redirigit a una nova finestra on es mostren totes les comandes entregades amb el seu identificador. 7. El botiguer prem sobre la comanda que vol consultar. 8. El botiguer és redirigit a la finestra amb tota la informació de la comanda

Taula 24: Cas d'ús per consultar una comanda entregades del comerç

Cas d'ús	Entregar comanda pendent de recollir
Actor	Botiguer
Precondicions	L'usuari està registrat a l'aplicació i és botiguer d'un comerç.
Disparador	El botiguer vol marcar una comanda com entregada.
Procediment	<ol style="list-style-type: none"> 1. El botiguer prem l'últim botó inferior del menú principal. 2. El botiguer és redirigit a la finestra del perfil on pot consultar la seva informació. 3. El botiguer prem el botó "El meu comerç". 4. El botiguer és redirigit a l'espai personal del seu comerç. 5. El botiguer prem el botó "Comandes" 6. El botiguer es redirigit a una nova finestra on es mostren totes les comandes entregades amb el seu identificador. 7. El botiguer prem sobre la comanda que vol entregar. 8. El botiguer és redirigit a la finestra amb tota la informació de la comanda. 9. El botiguer prem el botó inferior "Entregar comanda". 10. La comanda canvia el seu estat a entregada.

Taula 25: Cas d'ús per entregar una comanda

Cas d'ús	Afegir comentari a comanda pendent de recollir
Actor	Botiguer
Precondicions	L'usuari està registrat a l'aplicació i és botiguer d'un comerç.
Disparador	El botiguer vol afegir un comentari a una comanda del comerç on treballa perquè ho vegi el client que l'ha sol·licitat.
Procediment	<ol style="list-style-type: none"> 1. El botiguer prem l'últim botó inferior del menú principal. 2. El botiguer és redirigit a la finestra del perfil on pot consultar la seva informació. 3. El botiguer prem el botó "El meu comerç". 4. El botiguer és redirigit a l'espai personal del seu comerç. 5. El botiguer prem el botó "Comandes" 6. El botiguer es redirigit a una nova finestra on es mostren totes les comandes entregades amb el seu identificador. 7. El botiguer prem sobre la comanda que vol entregar. 8. El botiguer és redirigit a la finestra amb tota la informació de la comanda. 9. El botiguer prem sobre el botó "Veure comentaris". 10. Se li mostra al botiguer els comentaris de la comanda. 11. El botiguer afegeix el comentari que vol posar i prem el botó "Modificar comentari". 12. El comentari és afegit a la comanda.

Taula 26: Cas d'ús per afegir un comentari a una comanda del comerç

Cas d'ús	Consultar producte del comerç
Actor	Botiguer
Precondicions	L'usuari està registrat a l'aplicació i és botiguer d'un comerç.
Disparador	El botiguer vol consultar els productes que ofereix el seu comerç.
Procediment	<ol style="list-style-type: none"> 1. El botiguer prem l'últim botó inferior del menú principal. 2. El botiguer és redirigit a la finestra del perfil on pot consultar la seva informació. 3. El botiguer prem el botó "El meu comerç". 4. El botiguer és redirigit a l'espai personal del seu comerç. 5. El botiguer prem el botó "Productes" 6. El botiguer es redirigit a una nova finestra on es mostren tots els productes que el comerç ofereix. 7. El botiguer prem sobre "Mostrar informació" del producte que vol consultar. 8. Se li mostra al botiguer la informació del comerç que vol consultar.

Taula 27: Cas d'ús per consultar producte del comerç

Cas d'ús	Modificar preu d'un producte del comerç
Actor	Botiguer
Precondicions	L'usuari està registrat a l'aplicació i és botiguer d'un comerç.
Disparador	El botiguer vol modificar el preu d'un producte que ofereix el comerç on treballa..
Procediment	<ol style="list-style-type: none"> 1. El botiguer prem l'últim botó inferior del menú principal. 2. El botiguer és redirigit a la finestra del perfil on pot consultar la seva informació. 3. El botiguer prem el botó "El meu comerç". 4. El botiguer és redirigit a l'espai personal del seu comerç. 5. El botiguer prem el botó "Productes" 6. El botiguer es redirigit a una nova finestra on es mostren tots els productes que el comerç ofereix. 7. El botiguer prem sobre "Mostrar informació" del producte que vol consultar. 8. Se li mostra al botiguer la informació del comerç que vol consultar. 9. El botiguer prem sobre el botó "Canviar preu". 10. El botiguer introdueix el nou preu al desplegable que li ha aparegut i prem el botó "Guardar" 11. El preu del producte canvia al valor introduït pel botiguer.

Taula 28: Cas d'ús per consultar producte del comerç

4.2.4 Model conceptual de les dades

Tenint en compte els requisits definits a l'apartat anterior, es procedeix a representar en un diagrama amb les classes d'objectes que existeixen en el context del projecte i les relacions que hi ha entre ells. Primer es descriu quins els objectes de cada classe que surt en el diagrama, i en la figura 26 es pot veure el diagrama.

- **Usuari:** La classe Usuari representa les persones que utilitzen l'aplicació. Per cada persona, l'aplicació necessita emmagatzemar les seves dades personals (email, nom, numero de telefon, icona) i el identificador (Id) de l'usuari a la base de dades de Firebase.
- **Comerç:** La classe Comerç representa les botigues que ofereixen productes a través de l'aplicació. Cada comerç té com a mínim un botiguer. Un comerç pot vendre molts productes. Les dades que interessa emmagatzemar d'un comerç són: nom, eslogan o descripció curta, descripció del comerç, telèfon, ubicació del comerç, imatge del comerç i icona i logotip del comerç.
- **Botiguer:** Els Botiguers són una subclasse dels Usuaris que gestionen un Comerç.
- **Producte:** La classe Producte representa els productes que es venen en Comerços que utilitzaran l'aplicació. Els atributs necessaris per a un producte són: nom, descripció, tipus de producte, preu i icona.

- **Comanda:** La classe Comanda representa les comandes que faran els Usuaris als Comerços que vendran a través de l'aplicació. Cada comanda la demana un un Usuari. Una Comanda es fa de productes d'un únic Comerç. Les dades que ens interessa emmagatzemar d'una comanda són: el seu estat, la data d'expedició, comentaris i el cost total de la comanda.
- **Producte quantitat:** En una Comanda es demanen de 1 a molts Productes. Per cada producte que es demana en una comanda interessa guardar la quantitat del producte que s'ha demanat a la comanda.

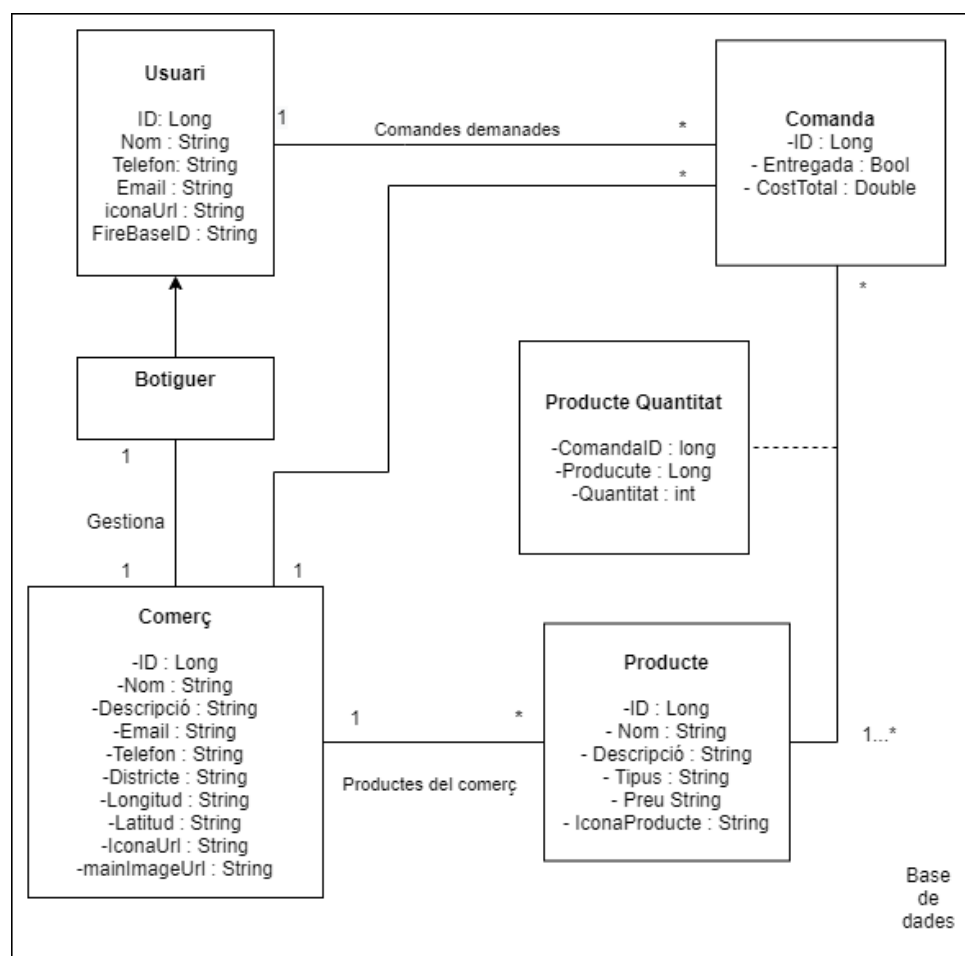


Diagrama 3: Model conceptual de les dades

4.3 Requisits no funcionals

Els requisits no funcionals són requisits que estableixen la qualitat que els projectes de software han de tenir, a continuació es llisten tots els requisits no funcionals que un sistema software ha de tenir.

4.3.1 Requisits de percepció

Descripció	La interfície ha de ser atractiva i agradable a la vista
Justificació	Una interfície atractiva i agradable a la vista convida als usuaris a utilitzar-la.
condició de satisfacció	El sistema ha de ser provat per persones que tinguin un perfil similar als usuaris que utilitzin l'aplicació. El sistema serà valorat amb la opinió d'aquets usuaris. Les valoracions han de ser positives.

Taula 29: Requisit de percepció 1

4.3.2 Requisits d'usabilitat

Descripció	La interfície ha de ser intuïtiva i fàcil d'utilitzar
Justificació	Una interfície poc intuïtiva i difícil d'utilitzar comporta una mala experiència pels usuaris causant una pèrdua d'aquests.
condició de satisfacció	L'aplicació ha de ser apte per què la utilitzin persones amb pocs coneixements informàtics.

Taula 30: Requisit d'usabilitat 1

4.3.3 Requisits de rendiment

Descripció	El temps de resposta al utilitzar crides de l'aplicació amb serveis web ha de ser inferior a 2 segons
Justificació	Els usuaris de l'aplicació volen disposar de la informació de l'aplicació en el menor temps possible.
condició de satisfacció	Les crides a l'aplicació de serveis web han de tenir un temps de resposta inferior a 2 segons

Taula 31: Requisit de rendiment 1

Descripció	Disponibilitat horària del sistema
Justificació	L'aplicació ha d'estar sempre disponible pels usuaris, s'han d'evitar caigudes del sistema.
condició de satisfacció	L'aplicació ha d'estar activa 24h al dia, 7 dies a la setmana.

Taula 32: Requisit de rendiment 2

4.3.4 Requisits de fiabilitat i disponibilitat

Descripció	Disponibilitat del sistema
Justificació	L'aplicació ha de ser disponible per la major part dels dispositius dels usuaris.
condició de satisfacció	L'aplicació ha d'estar disponible per almenys un 85% dels usuaris d'Android.

Taula 33: Requisit de fiabilitat i disponibilitat 1

Descripció	Fiabilitat del sistema
Justificació	L'aplicació no ha de contenir errors i en cas que aparegui un error ha de seguir funcionant sense impedir el funcionament per part dels usuaris.
condició de satisfacció	L'aplicació ha de tractar els possibles errors que puguin sorgir durant la seva execució.

Taula 34: Requisit de fiabilitat i disponibilitat 2

4.3.5 Seguretat

Descripció	Accés al sistema
Justificació	L'aplicació ha de controlar l'accés al sistema.
condició de satisfacció	Només els usuaris donats d'alta al sistema poden accedir a les seves funcionalitats.

Taula 35: Requisit de seguretat 1

Descripció	Privacitat
Justificació	L'aplicació ha de garantir que les dades dels usuaris no siguin visibles per altres usuaris.
condició de satisfacció	L'aplicació ha de complir la llei de protecció de dades i avisar a l'usuari si es realitza un canvi que pugui afectar la seva privacitat.

Taula 36: Requisit de seguretat 2

4.3.6 Requisits legals

Descripció	Disponibilitat del sistema
Justificació	L'aplicació ha de complir amb totes les lleis per no tenir cap problema de caràcter legal.
condició de satisfacció	El sistema ha de complir la llei orgànica de protecció de dades vigent

Taula 37: Requisit legal 1

5. Arquitectura del sistema

L'aplicació es pot dividir en una estructura clàssica de *backend* i *frontend*.

L'aplicació final implementada consisteix en una aplicació mòbil i una base de dades. Per accedir a les dades de la base de dades des de l'aplicació mòbil, el que s'ha fet és implementar un servei web amb diferents direccions URL per accedir a les dades..

L'aplicació utilitza *Firebase auth*[16], un sistema extern per realitzar el registre i login dels usuaris de l'aplicació, aquest sistema permet guardar la informació mínima necessària dels usuaris per realitzar aquestes accions. Concretament, la informació necessària de cada usuari són: La direcció email i la contrasenya.

5.1 Arquitectura física

L'aplicació desenvolupada està allotjada en tres sistemes diferents, aquests són: el *backend* de l'aplicació a la plataforma de serveis *Heroku*, el servei d'autorització i autenticació d'usuaris a *Firebase auth* i el Front End de l'aplicació en el dispositiu android dels usuaris.

5.1.1 Heroku

Heroku és una plataforma pensada per allotjar serveis web accessibles a través d'APIs, en el cas d'aquest projecte, el que s'ha fet és allotjar la base de dades i l'aplicació que conté el serveis web per accedir-hi.

La base de dades que s'utilitza en el projecte és una base de dades relacional emmagatzemada en un servidor *PostgreSQL*.

Un dels motius per escollir *Heroku* com a servei per allotjar la base de dades i el serveis web ha sigut l'experiència previa, ja que ja s'havia utilitzat aquest servei per allotjar aplicacions en les pràctiques d'una assignatura del Grau, i ha resultat relativament senzill tonrar a utilitzar-la.

L'altre factor important és la gratuïtat d'aquest servei, *Heroku* permet allotjar aplicacions de manera gratuïta si es tracta de projectes de petites dimensions.

5.1.2 Firebase auth

Firebase auth és un servei de Google de *Firebase* que ens permet mantenir la sessió de l'usuari durant l'execució d'un programa en un dispositiu Android.

En el cas d'aquest projecte, s'utilitza per automatitzar el registre d'un usuari, iniciar sessió i conseguir informació de l'usuari durant la sessió.

Un dels motius que ha fet que escollim Firebase han sigut les seves crides i la seva senzillesa. Firebase permet de manera molt senzilla mantenir la persistència de l'usuari a l'aplicació, està molt ben documentat, permet utilitzar totes les seves funcionalitats de manera gratuïta sempre que no es superin els 10.000 crides mensuals o el 1 GB de memòria per guardar informació.

5.1.3 Dispositiu Android

El *frontend* s'allotja en format aplicació a cada un dels dispositius Android dels usuaris de l'aplicació.

El Front End s'ha implementat en llenguatge *Java* apte per a dispositius android amb versió 5.0 *Lollipop* o posterior. Aquesta versió ens permet que l'aplicació sigui utilitzada per un 94% dels dispositius Android actuals.

5.1.4 Diagrama de l'arquitectura física

En el Diagrama 4 es pot visualitzar l'arquitectura física i com els diferents elements interactuen entre ells.

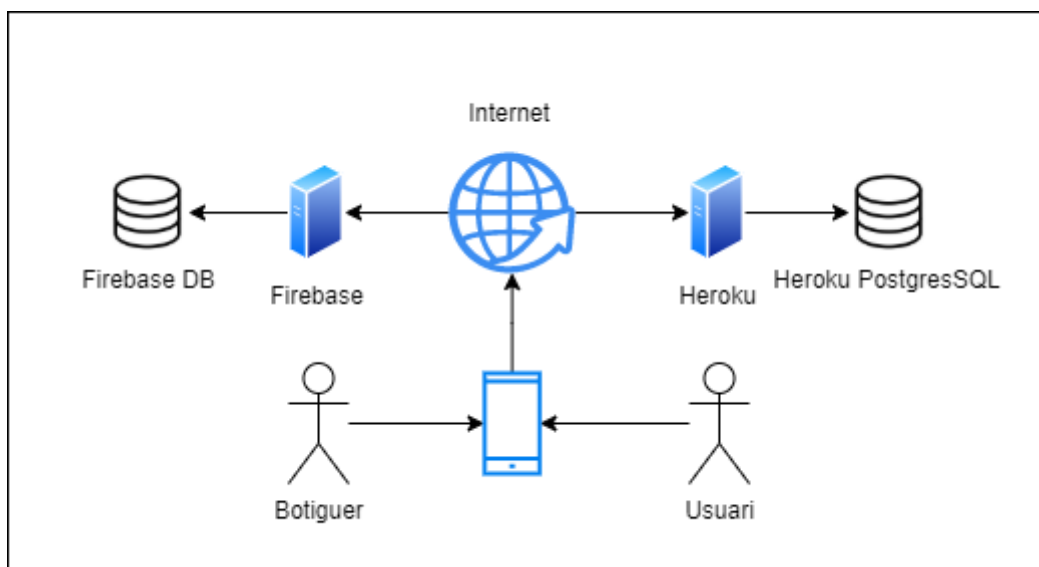


Diagrama 4: Representació gràfica de l'arquitectura física

5.2 Arquitectura lògica

Tant el Front End que és l'aplicació mòbil desenvolupada per a Android, com el *backend* que són els serveis web que donen accés a la base de dades situats en el servidor Heroku, segueixen el patró arquitectònic MVC.

En el diagrama 5 podem veure l'arquitectura lògica d'aquests dos components i com interactuen entre ells.

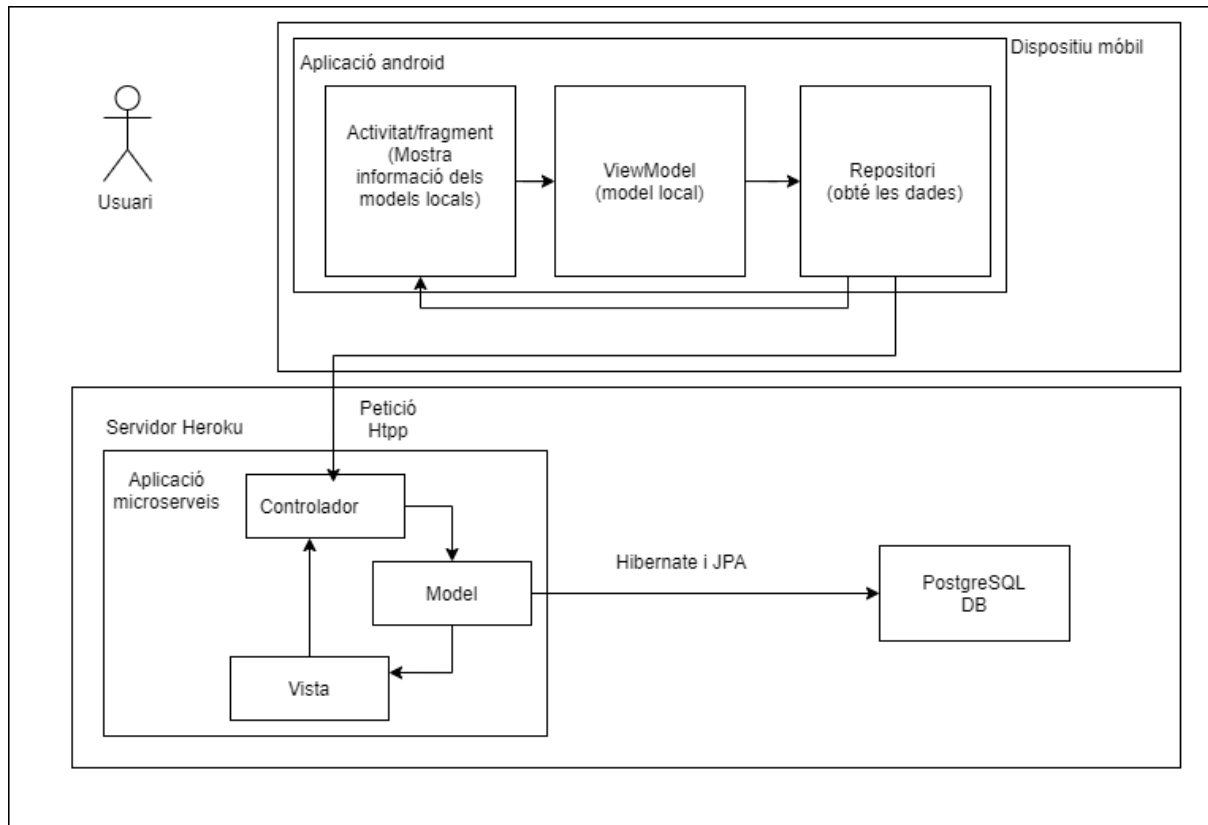


Diagrama 5: Arquitectura lògica del projecte

S'ha escollit el patró de disseny MVC ja que és un dels patrons de disseny més fàcils d'utilitzar per realitzar una aplicació que conté serveis web, un altre factor per escollir aquest model és la escalabilitat dels sistemes que utilitzen aquest model.

5.2.1 Arquitectura lògica de l'aplicació mòbil

L'arquitectura lògica de l'aplicació mòbil segueix una arquitectura de tres nivells. Tenim una capa de presentació, una capa de negoci i una capa de dades.

A la capa de presentació hi ha les pantalles amb que l'usuari interactua amb l'aplicació mòbil. Aquesta capa la conformen els *layouts* i les *carviews* del sistema.

A la capa de negoci hi ha els programes que s'executen les peticions que l'usuari fa a la capa de presentació, també s'encarrega d'obtenir i pasar la informació de la capa de dades

a la capa de presentació. També és l'encarregada de realitzar les crides a l'aplicació de serveis web.

Aquesta capa la conformen, les activitats, els fragments, els adaptadors, i les classes que realitzen les crides a l'aplicació de serveis web.

A la capa de dades hi tenim els models del sistema, aquests models guarden de forma local informació de les entitats del sistema. L'aplicació no té mai tota les entitats de la base de dades, es van realitzant crides a través de la capa de negoci per obtenir entitats de la base de dades.

Aquesta capa la conformen els models del sistema i els *DTOs* d'aquests.

En el diagrama 6 podem veure un exemple part de l'arquitectura lògica de l'aplicació de mòbil implementada.

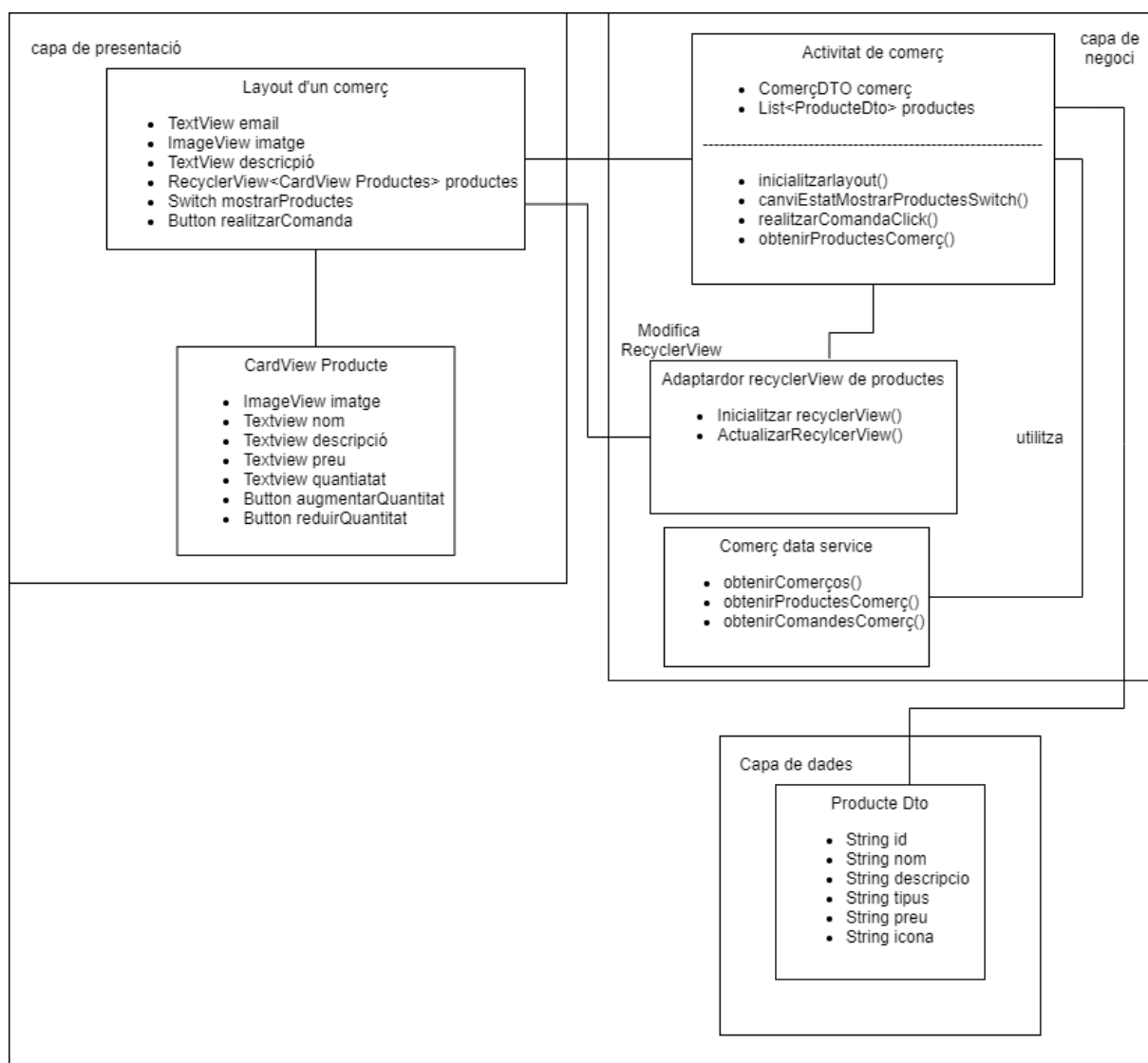


Diagrama 6: Exemple de l'arquitectura lògica d'un comerç del *frontend*

5.2.2 Arquitectura lògica de l'aplicació de de serveis web

L'aplicació que conté la base de dades i els serveis web la conformen dues capes, la capa de negoci i la capa de dades.

A la capa de negoci hi tenim els programes que reben les crides que arriben a l'aplicació i les interaccions que es fa amb la capa de dades. Aquesta capa la conformen els controladors, els serveis i els repositoris.

La capa de dades és la capa hi tenim les entitats i la base de dades del sistema. Es fa servir *Hibernate* i *JPA* per gestionar les crides a la base de dades.

L'aplicació de serveis web funciona de la següent manera:

- Els controladors reben les crides a l'aplicació, els controladors evalúan la crida, si aquesta és vàlida, aquest crida una de les funcions de el servei corresponent.
- El servei amb la informació rebuda del controlador interactua amb la capa de dades, per interactuar amb la capa de dades utilitza les funcions del repositori corresponent.
- El repositori s'encarrega de traduir les seqüències de JAVA en seqüències SQL per interactuar amb la base de dades. Aquest interactua amb la base de dades i retorna el resultat de la seqüència SQL al servei, si és una entitat del sistema, aquest retorna un objecte.
- El servei rep la informació del repositori. Si la crida ha de retorna un objecte, aquest crea un DTO a partir de la informació obtinguda del repositori. un cop creat el DTO, aquest l'envia al controlador.
- El controlador retorna com a resultat de la crida la informació obtinguda del servei.

En el diagrama 7 podem veure els elements que conformen el disseny lògic i com interactuen entre ells amb un exemple.

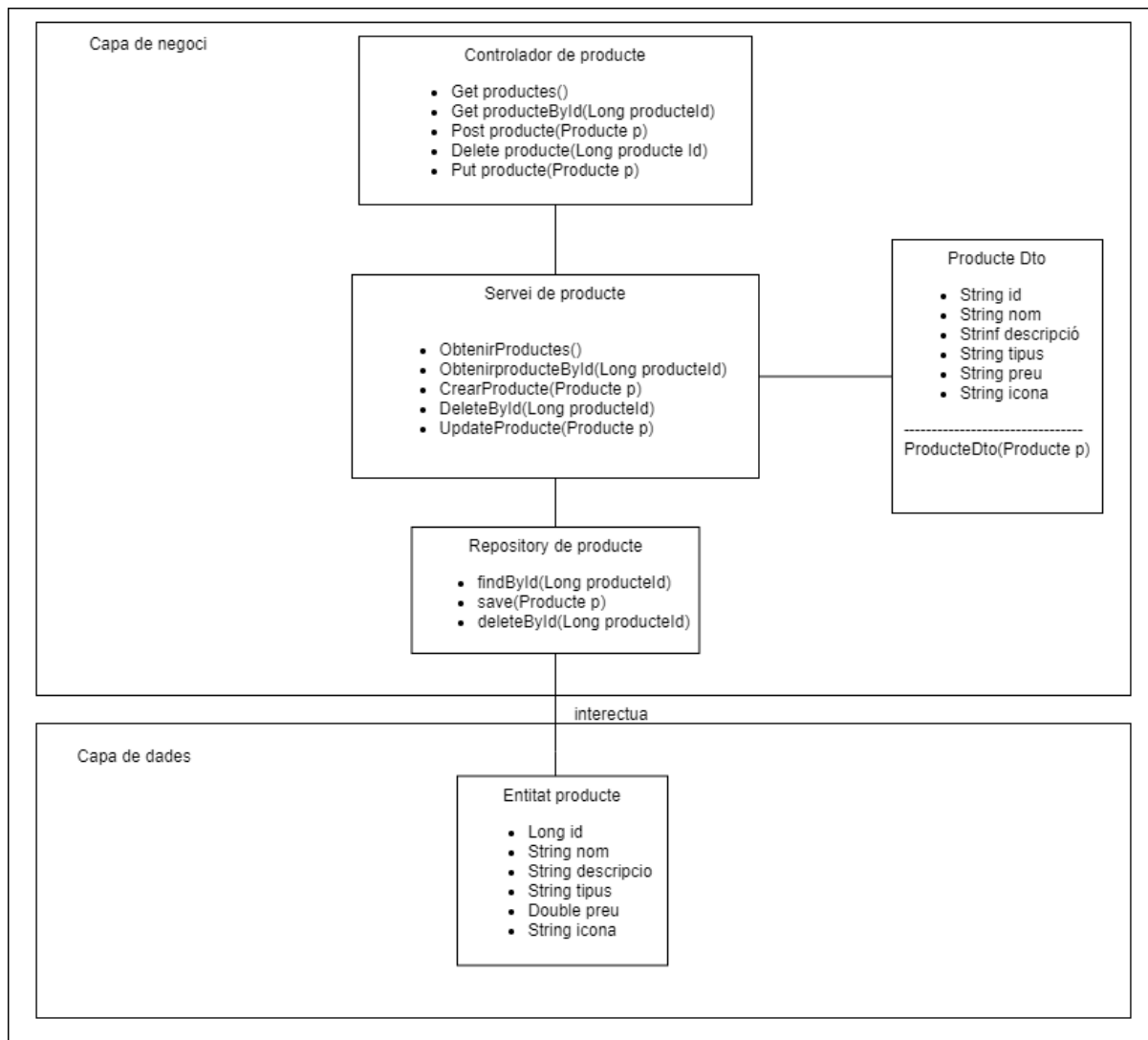


Diagrama 7: Exemple de l'arquitectura lògica de producte al *backend*

5.3 Patrons utilitzats

5.3.1 Patró de disseny MVC

Com ja s'ha comentat a l'apartat anterior, s'utilitza el patró de disseny MVC per desenvolupar l'aplicació de serveis web.

5.3.2 Patró Singleton

S'utilitza el patró *Singleton* en la implementació de l'aplicació mòbil.

Aquest patró s'utilitza per tenir una única instància d'un objecte en tot el projecte, en el cas de l'aplicació mòbil, el que tenim és una única instància de l'objecte que s'encarrega de fer les peticions a l'API.

El que ens permet aquest patró *Singleton* és posar totes les crides a una cua i que s'executin a mesura que van arribant.

La implementació de la classe singleton es pot veure al segment de Codi 1.

```
public class RequestQueueSingleton {
    private static RequestQueueSingleton instance;
    private RequestQueue requestQueue;
    private static Context ctx;

    private RequestQueueSingleton(Context context) {
        ctx = context;
        requestQueue = getRequestQueue();
    }

    public static synchronized RequestQueueSingleton getInstance(Context context) {
        // això permet només una instància en tota la aplicació del singleton
        if (instance == null) {
            instance = new RequestQueueSingleton(context);
        }
        return instance;
    }

    public RequestQueue getRequestQueue() {
        if (requestQueue == null) {
            // getApplicationContext() is key, it keeps you from leaking the
            // Activity or BroadcastReceiver if someone passes one in.
            requestQueue = Volley.newRequestQueue(ctx(getApplicationContext()));
        }
        return requestQueue;
    }

    public <T> void addToRequestQueue(Request<T> req) { getRequestQueue().add(req); }
}
```

Codi 1: Implementació del singleton usat a l'aplicació mòbil

5.3.3 Patró adaptador

El patró adaptador s'utilitza a l'aplicació de mòbil.

Aquest patró ens permet transformar una interfície en una altre, d'aquesta manera una classe que no pugui utilitzar aquesta interfície la pugui usar.

En l'aplicació mòbil aquest patró s'utilitza per utilitzar un *RecyclerView*.

Un *Recyclerview*, és una vista que ens permet visualitzar un llistat d'objectes, per exemple un llistat de comerços.

En aquest cas l'adaptador s'usa per crear i manipular un *Recyclerview* amb la informació d'una crida a un servei web, per exemple una crida que retorna un llistat de comerços.

5.3.4 Patró DTO (Data Transfer Object)

El patró DTO s'utilitza tant en el l'aplicació mòbil com a l'aplicació que utilitza els serveis web.

La funció del patró DTO és la de crear objectes a partir de les entitats que tenim en el nostre sistema. Ens permet separar les entitats de la base de dades dels objectes que es retornem al rebre una crida a un servei.

També ens permet no retornar tota la informació d'una entitat de la base de dades o crear un objecte DTO que contingui informació de diferents entitats.

En el codi 2 i 3 podem veure la diferència entre una entitat usuari amb les seves propietats i el DTO que retorna quan es realitza una crida a un dels serveis web per obtenir la informació d'un usuari.

```
public class User {  
  
    private long id;   
    private String nom;   
    private String telefon;   
    private String email;   
    private String iconaUrl;   
    private Boolean esBotiguer;   
    private String firebaseUID;   
  
    private Botiga botigaPropietari;   
  
    private List<Comanda> comandesUsuari = new ArrayList<Comanda>();   
}
```

Codi 2: implementació dels atributs de la entitat usuari

```
public class UserDto {  
  
    private String id;   
    private String nom;   
    private String telefon;   
    private String email;   
    private String iconaUrl;   
    private Boolean esbotiguer;   
  
    public UserDto(User user) {   
        this.id = String.valueOf(user.getId());   
        this.nom = user.getNom();   
        this.telefon = user.getTelefon();   
        this.email = user.getEmail();   
        this.esbotiguer = user.getEsBotiguer();   
        this.iconaUrl = user.getIconaUrl();   
    }   
}
```

Codi 3: Implementació dels atributs del Dto d'un usuari.

5.4 Exemples de diagrames de seqüència

5.4.1 Diagrama de seqüència de registre d'usuari

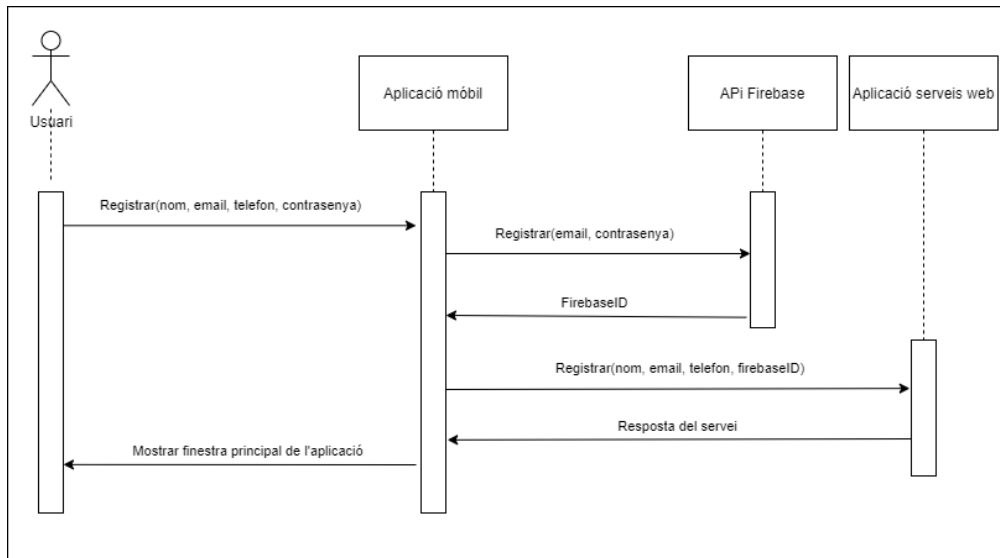


Diagrama 8: Diagrama de seqüència de registre d'usuari

5.4.2 Diagrama de seqüència de mostrar informació de comanda

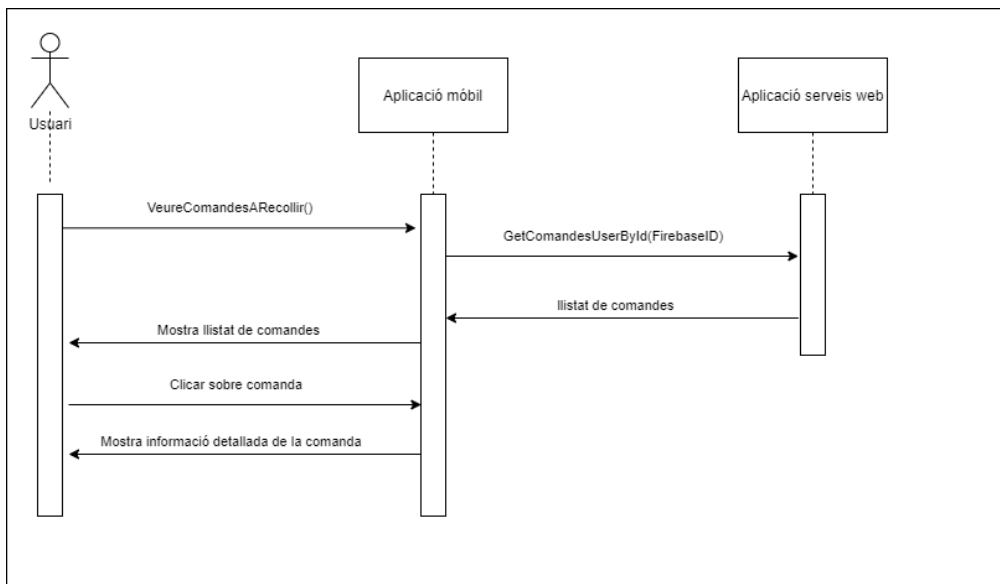


Diagrama 9: Diagrama de seqüència de mostrar informació de comanda

5.4.3 Diagrama de seqüència de sol·licitar una comanda

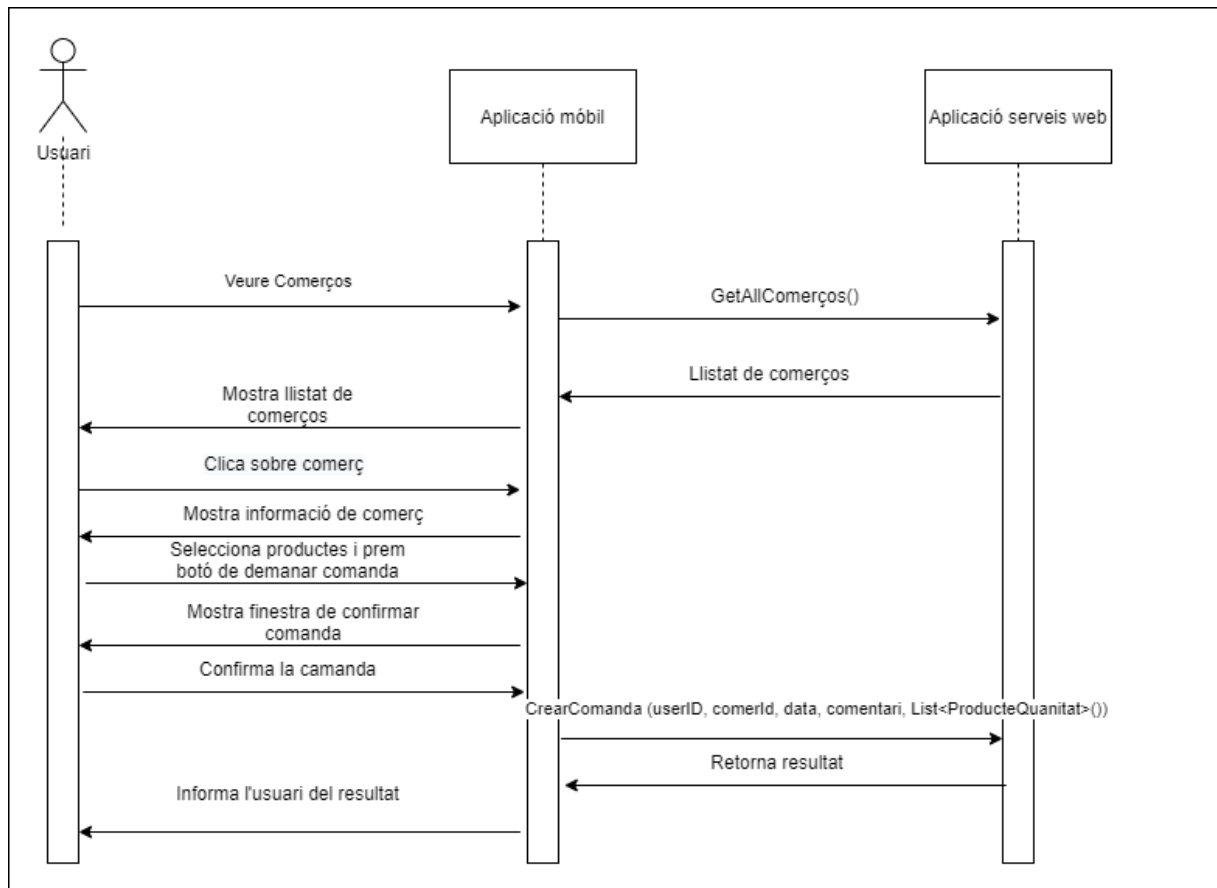


Diagrama 10: Diagrama de seqüència de sol·licitar una comanda

5.4.4 Diagrama de seqüència de modificar producte del comerç del botiguer

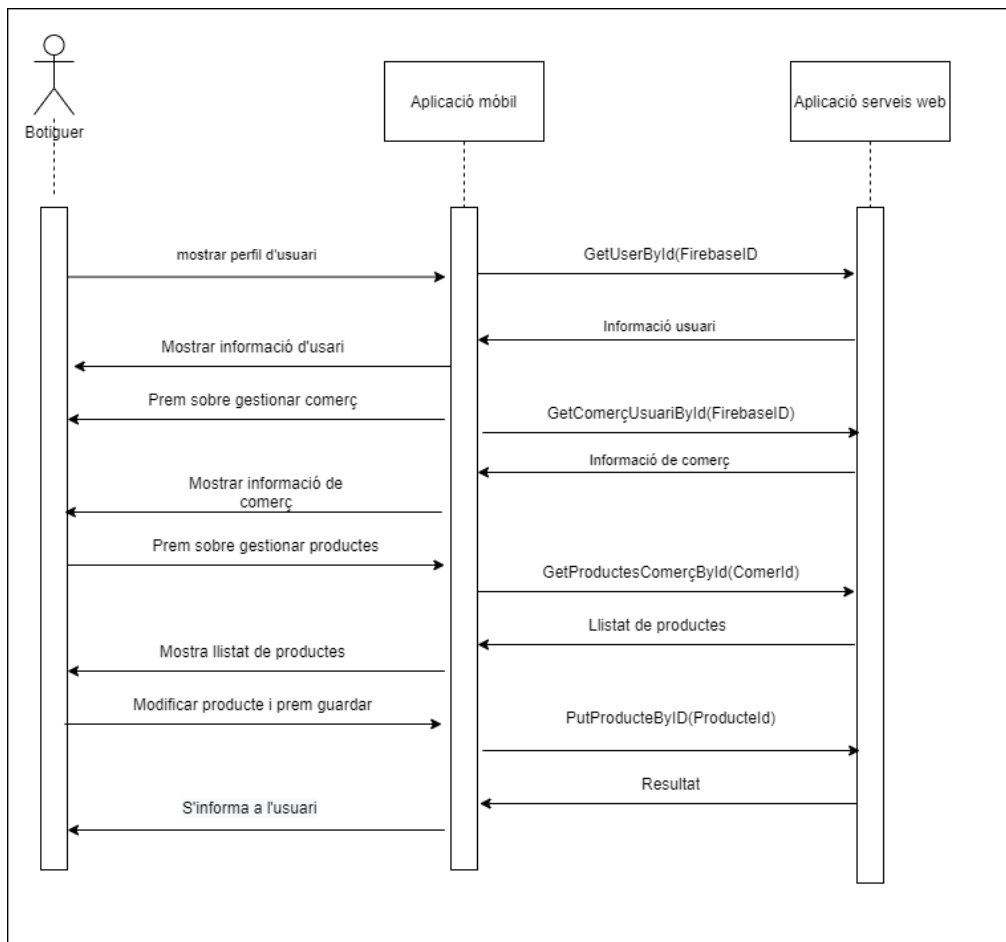


Diagrama 11: Diagrama de seqüència de modificar producte del comerç del botiguer

5.5 Bases de dades

El *backend* de l'aplicació té una base de dades relacional dins del servidor de Heroku. En concret, el que s'usa és PostgreSQL per gestionar aquesta base de dades.

Hi ha dos mètodes per accedir a la base de dades, el primer és mitjançant sentències SQL desde la terminal de comandes de *Heroku*. El segon mètode és utilitzar les crides implementades dins del *backend*.

5.5.1 Disseny lògic

A continuació definim el disseny lògic de la base de dades.

- **Usuari** (idUsuari, nom, telèfon, email, iconaUrl, esBotiguer, idFirebase, idComerç)
 - nom Not null
 - email Not null
 - esBotiguer default false
 - IdFirebase unique, Not null
 - IdComerç referència **Comerç**

La taula usuari correspon a les classes Usuari i Botiguer de la base de dades. S'ha afegit un atribut esBotiguer que serà booleà i serà cert quan l'usuari sigui un botiguer.

Quan un usuari és botiguer, el idComerç indicarà quan és el Comerç del qual és propietari.

- **Comerç**(idComerç, nom, descripció, email, telèfon, districte, longitud, latitud, iconaUrl, imatgePrincipalUrl, idBotiguer)
 - nom Not null
 - email Not null
 - longitud not null
 - latitud not null
 - IdBotiguer referència **Usuari**
- **Comanda**(idComanda, entregada, costTotal, UserOwnerId, idComerç)
 - entregada default false
 - UserOwnerId referència **Usuari**, Not null
 - IdComerç referència **Comerç**, Not null
- **Producte**(idProducte, nom, descripció, tipus, preu, iconaProducte)
 - nom Not null
 - tipus Not null
 - preu Not null
- **ProducteQuantitat**(IdComanda, IdProducte, quantitat)
 - IdComnada referència **Comanda**,
 - IdProducte referència **Producte**
 - Quantitat Not null, quantitat > 0.

5.5.2 Disseny físic

Per realitzar la implementació de la base de dades s'utilitza *Hibernate* i *JPA*, *Hibernate* i *JPA* són dos frameworks que ens permet mapejar objectes Java a una base de dades relacional.

Gràcies a aquets framework les taules físiques de l'aplicació es generen automàticament al executar l'aplicació de serveis web en el servidor de Heroku.

Per a la generació de les taules és necessari definir les relacions que hi ha entre elles, les relacions que definim són les següents:

- Relació Usuari-Comanda (relació un a molts)
- Relació Comerç-Comanda (relació un a molts)
- Relació Producte-Quantitat (relació molts a molts)
- Relació Usuari-Comerç (relació un a un)
- Relació Comerç-Producte (relació un a molts)

Les seqüències SQL que s'obtenen del mapeig de les entitats del nostre sistema són les següents:

- Taula comerços: `create table botigues (id int8 generated by default as identity, descripcio varchar(255), districte varchar(255), email varchar(255) not null, icon varchar(255), latitud float8, longitud float8, main_image_url varchar(255), nom varchar(30) not null, telefon varchar(9) not null, primary key (id)).`
- Taula comandes: `create table comandes (id int8 generated by default as identity, comentari varchar(255), cost_total float8, data_expedicio varchar(255), entregada boolean, preparada boolean, botiga_id int8, user_id int8, primary key (id)).`
- Taula productes: `create table productes (id int8 generated by default as identity, descripcio varchar(255), icona varchar(255), nom varchar(30) not null, preu float8 not null, tipus varchar(30) not null, primary key (id)).`
- Taula users: `create table users (id int8 generated by default as identity, email varchar(255), botiguer boolean default false, fire_baseuid varchar(255), icona_url varchar(255), nom varchar(30) not null, telefon varchar(9), botiga_id int8, primary key (id)).`
- Taula relació producte_quantitat: `create table producte_quantitat (comanda_id int8 not null, producte_id int8 not null, quantitat int4, primary key (comanda_id, producte_id)).`
- Taula relació comerços - productes: `create table botigues_productes_botiga (botiga_id int8 not null, productes_botiga_id int8 not null)`

5.6 Disseny de la interfície

Respecte al disseny de la interfície, s'ha decidit de crear una interfície simple i senzilla. S'ha utilitzat *Android studio* per implementar l'aplicació mòbil.

En el diagrama 12 es pot veure la navegabilitat entre les finestres de l'aplicació.

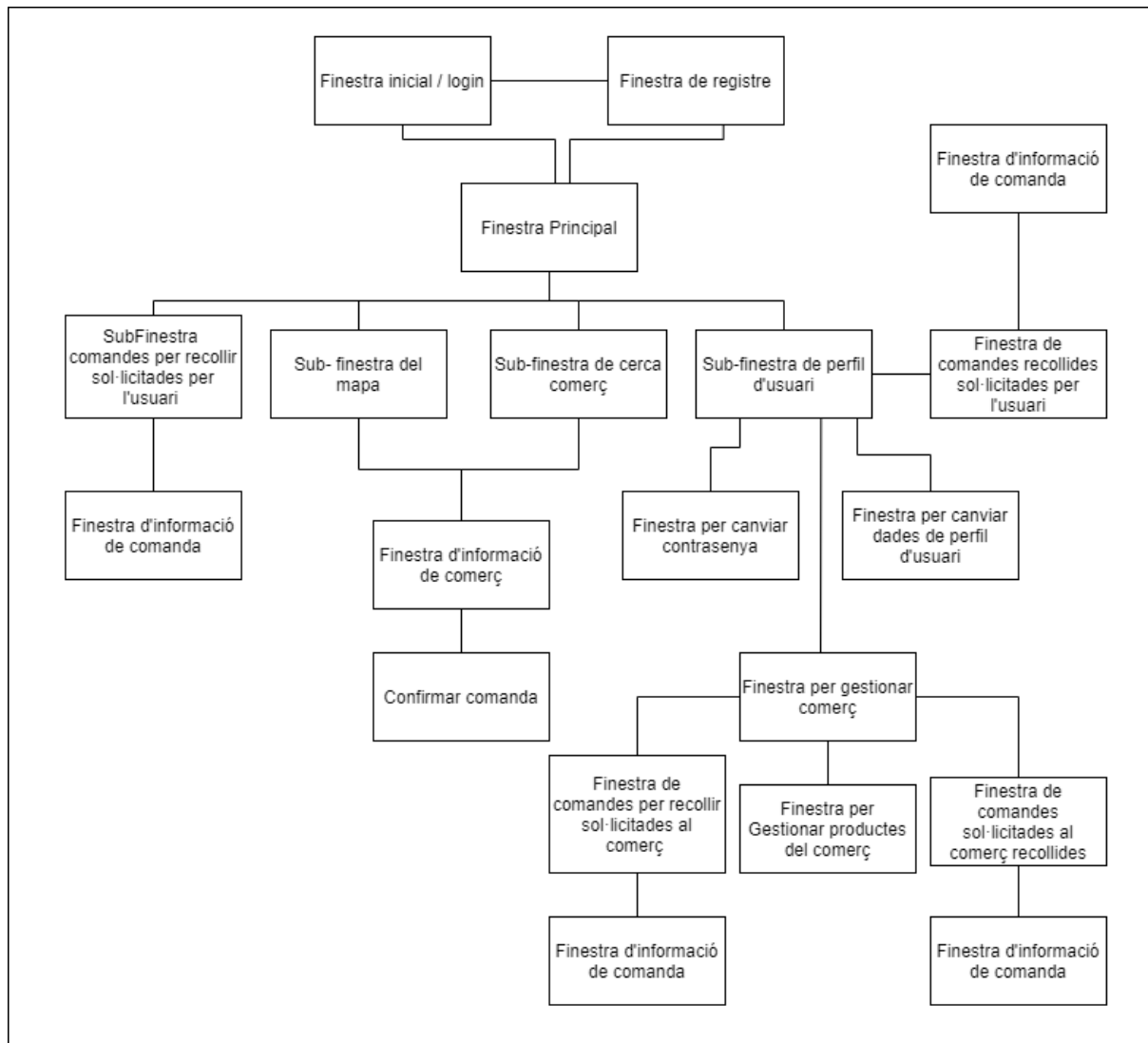
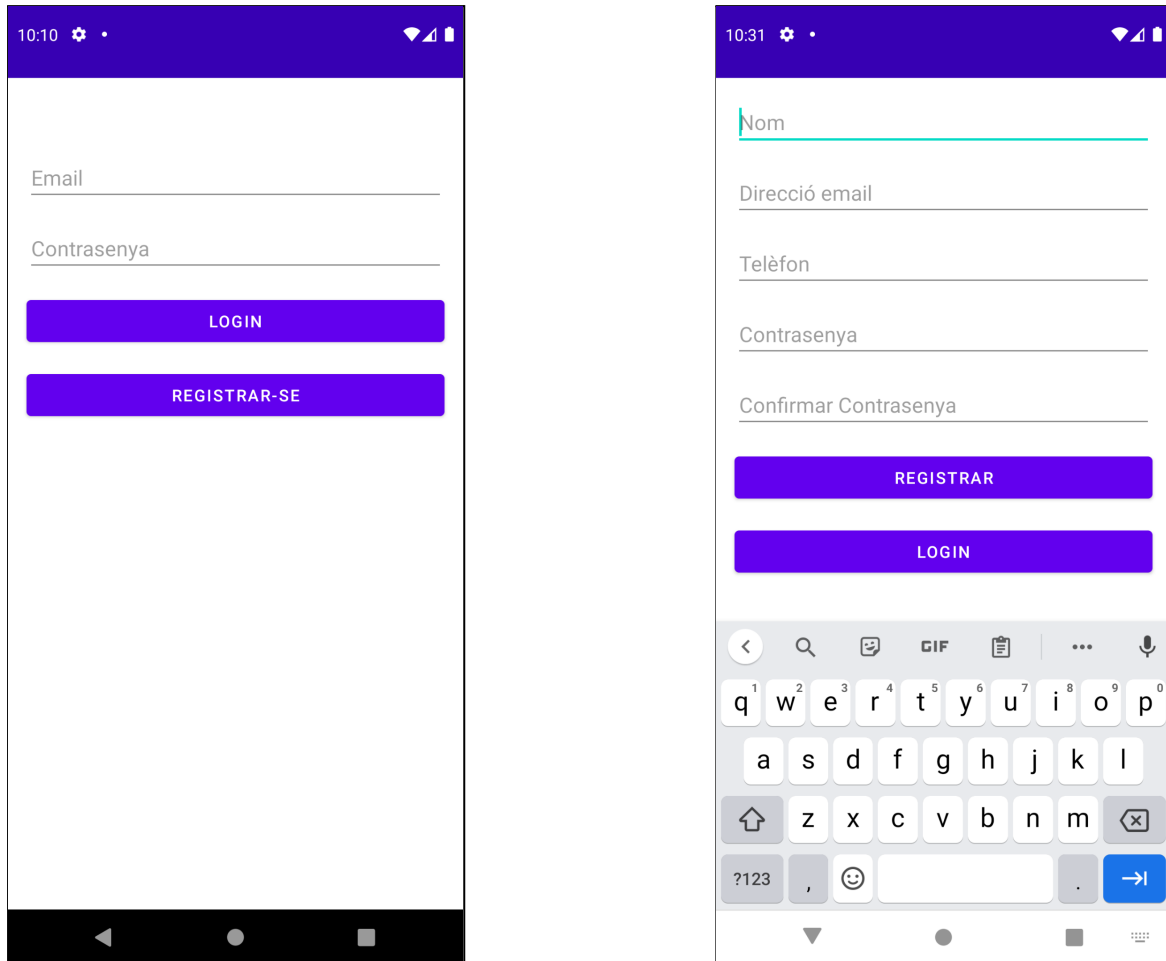


Diagrama 12: Navegabilitat entre finestres de l'aplicació mòbil

A continuació s'explicarà en detall que es pot fer a cada finestra i la seva navegabilitat.

5.6.1 Finestra inicial / login i finestra de registre

L'aplicació funciona de la següent forma, al accedir per primer cop s'accedeix a la finestra inicial, en aquesta l'usuari pot fer login a l'aplicació o passar a la finestra de registre si vol registrar-se a l'aplicació.



Imatge 1 i 2 : Finestra de registre i finestra de login

5.6.2 Finestra principal

Un cop l'usuari s'ha registrat o iniciat sessió, aquest és redirigit a la finestra principal. Aquesta finestra la formen quatre sub-finestres diferents aquestes són: la finestra de comandes per recollir, la finestra del mapa, la finestra del cercador de comerços i la finestra de perfil d'usuari. Es pot canviar de sub-finestra prement els botons inferiors.

5.6.3 Finestra de comandes per recollir de l'usuari

A la finestra de comandes per recollir tenim un llistat de les comandes pendents de recollir de l'usuari, si l'usuari prem sobre una comanda s'accedeix a la finestra d'informació de comanda on es veu la informació detallada de la comanda.



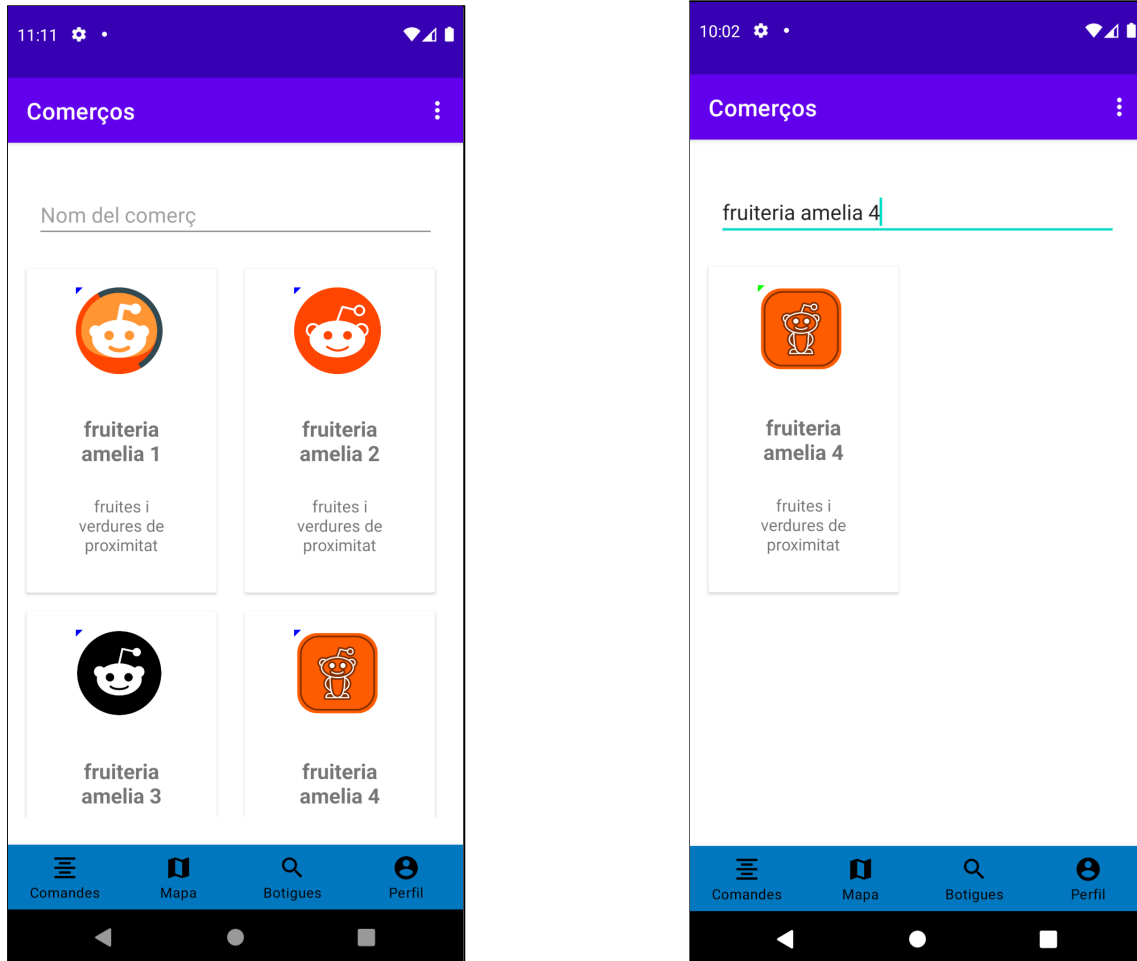
Imatge 3: Finestra de comandes per recollir de l'usuari amb menú de navegació inferior

5.6.4 Finestra del mapa de navegació

A la finestra del mapa tenim marcada al mapa la localització de l'usuari i els comerços que té aprop, si l'usuari prem sobre un comerç pot veure un resum de la informació del comerç i un link per accedir a la finestra del comerç.

5.6.5 Finestra de cerca de comerços

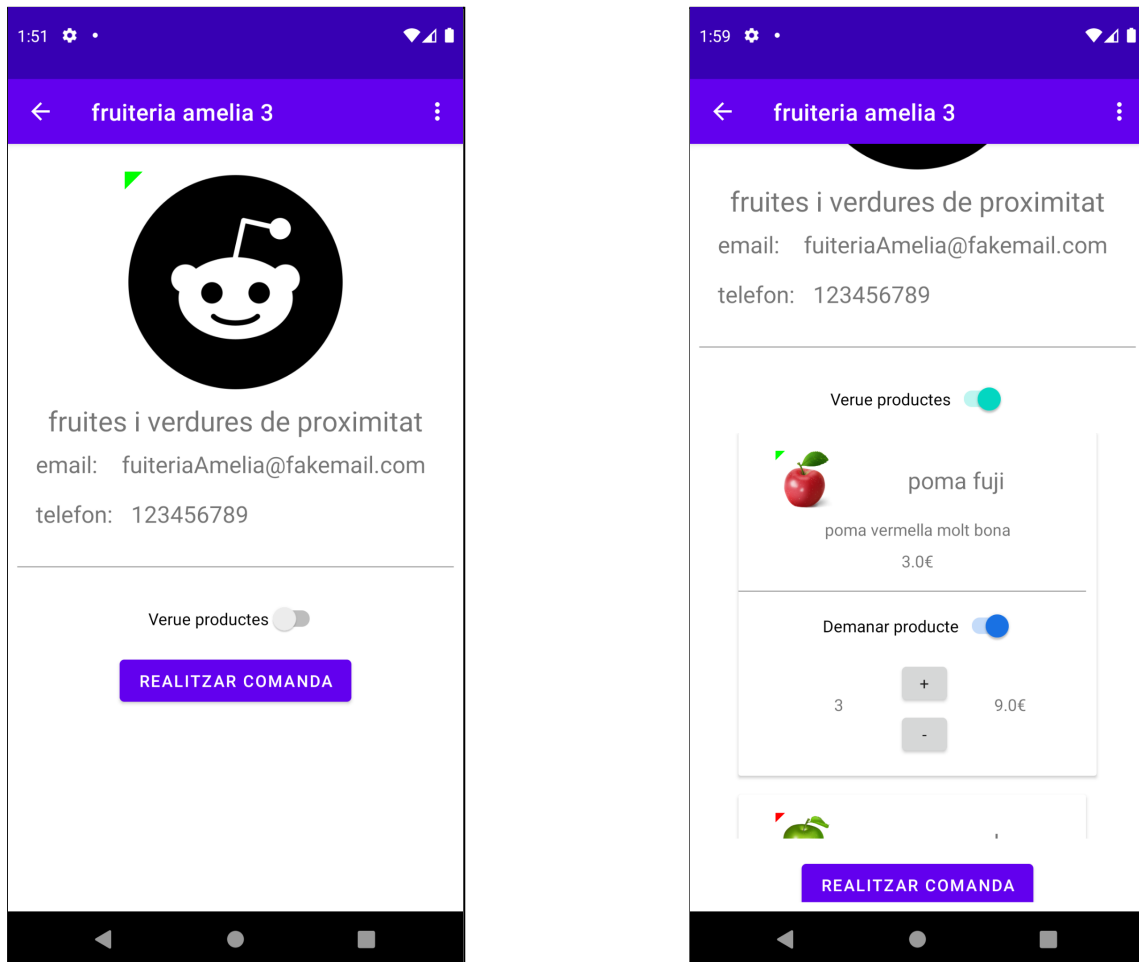
A la finestra de cerca comerç apareixen tots els comerços inscrits al sistema, l'usuari pot buscar un comerç i prémer sobre ell per veure la informació del comerç.



Imatge 4 i 5: Finestra de cerca de comerç i exemple del funcionament del cercador

5.6.6 Finestra d'informació de comerç

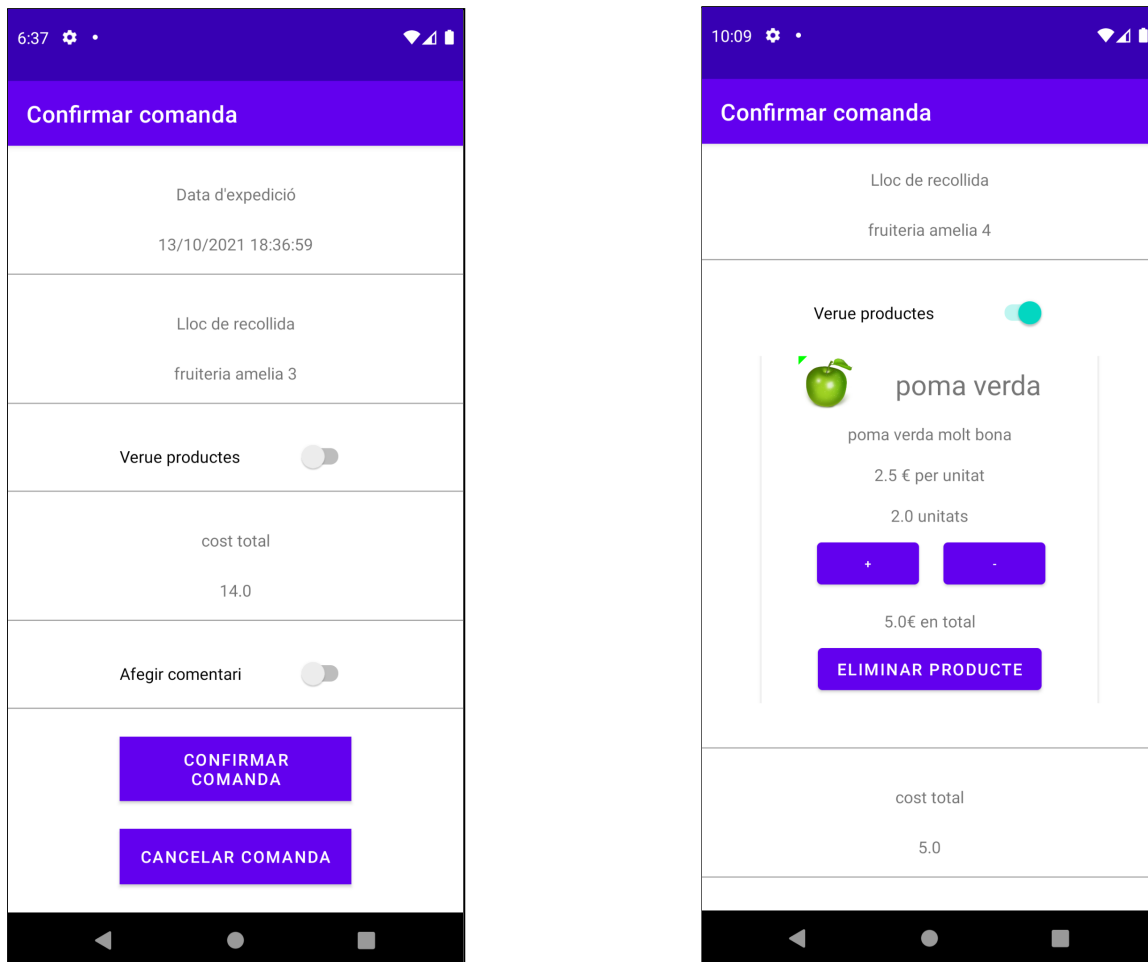
A la finestra d'informació de comerç tenim la informació detallada del comerç i els seus productes, l'usuari pot realitzar una comanda desde aquesta finestra. Un cop demanada una comanda és redirigit a la finestra de confirmació de comanda.



Imatge 6 i 7 : Finestra d'un comerç i finestra d'un comerç mostrant un producte

5.6.7 Finestra de confirmació de comanda

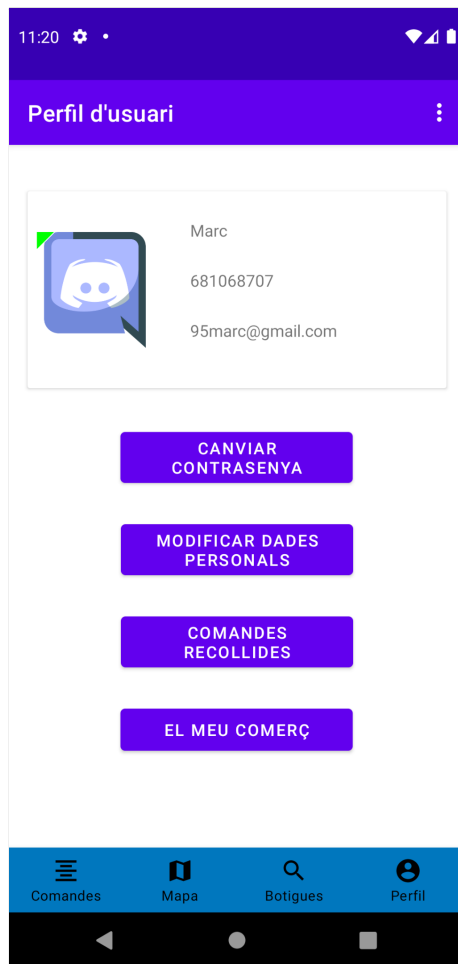
La finestra de confirmació de comanda és la finestra final per sol·licitar una comanda, en ella l'usuari pot canviar la quantitat de productes demanats i afegir comentaris a la comanda. L'usuari pot confirmar la comanda o cancel·lar.



Imatge 8 i 9: Finestra de confirmar comanda i finestra de confirmar comanda mostrant els productes

5.6.8 Finestra de perfil d'usuari

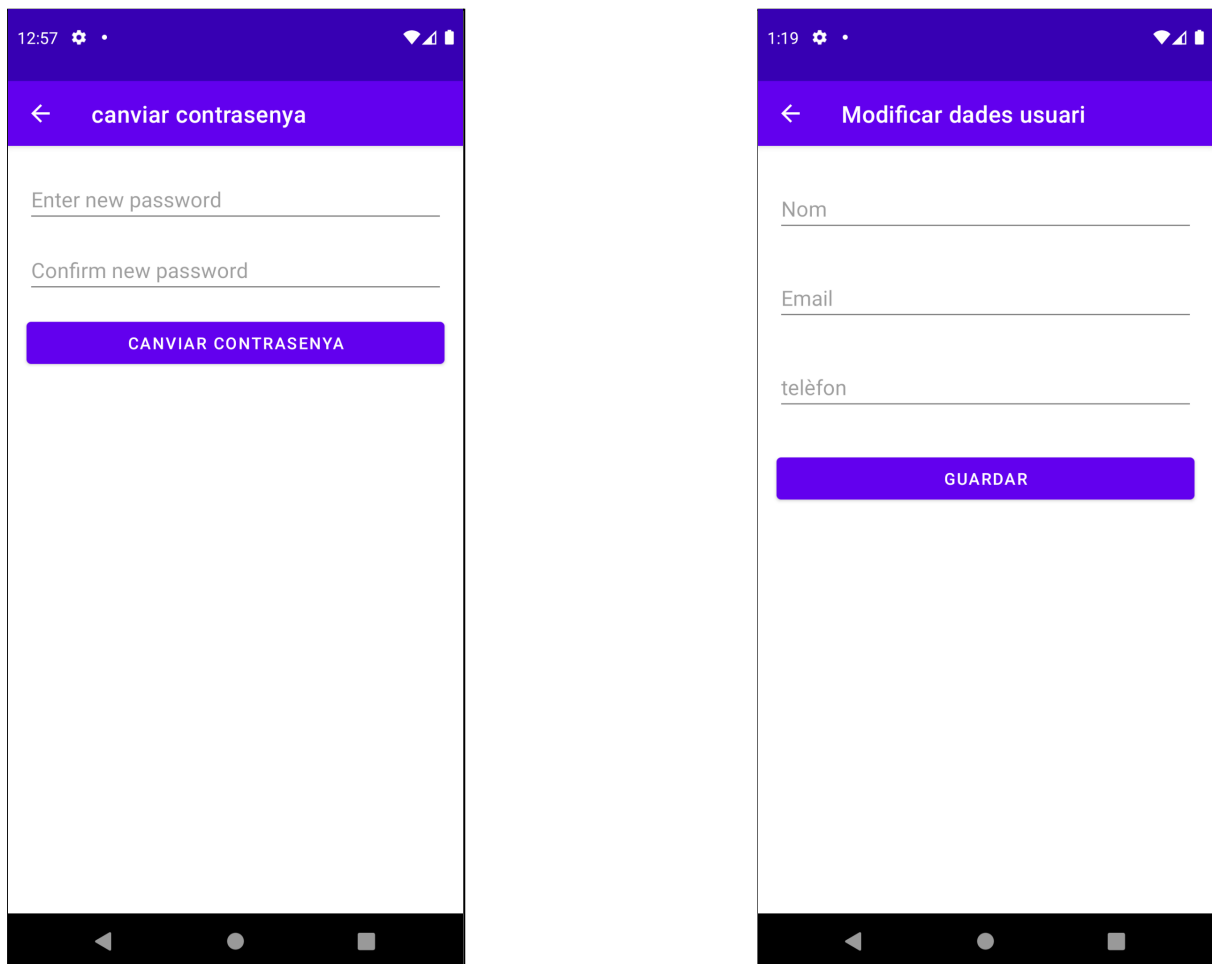
A la finestra de perfil hi ha la informació personal de l'usuari, desde aquesta es pot accedir a la finestra de modificar les dades d'usuari, la finestra de modificar la contrasenya de l'usuari, la finestra de comandes ja recollides de l'usuari i la finestra de gestió del comerç de l'usuari. Aquesta última finestra només és accessible si l'usuari és propietari d'un comerç.



Imatge 10: Sub-finestra principal de perfil d'usuari

5.6.9 Finestra de modificació de dades de l'usuari i finestra de canvi de contrasenya

A les finestres de modificació de dades d'usuari i contrasenya, l'usuari pot canviar les seves dades personals i la contrasenya per iniciar sessió a l'aplicació



Imatge 11 i 12: Finestra de canvi de contrasenya i finestra de modificació de dades d'usuari

5.6.10 Finestra de comandes ja recollides de l'usuari

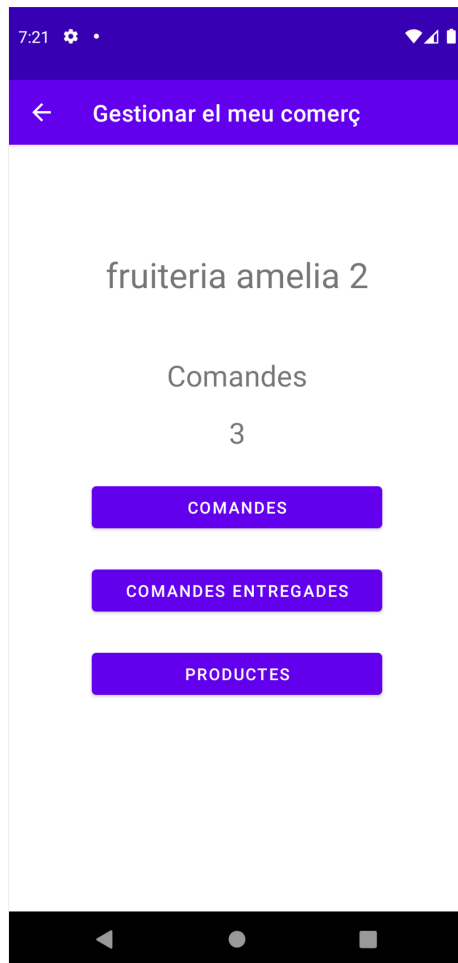
A la finestra de comandes ja recollides hi ha un llistat de les comandes ja recollides de l'usuari, si l'usuari prem sobre una de les comandes és redirigit a la finestra de informació de comanda amb tota la informació de l'usuari.



Imatge 13: Exemple de finestra amb un llistat de comandes

5.6.11 Finestra de gestió de comerç

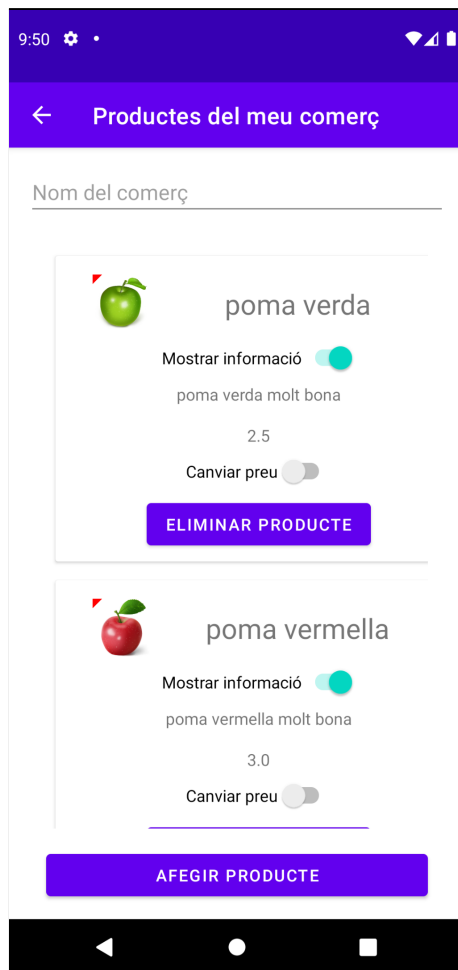
A la finestra de gestió del comerç de l'usuari tenim tota la informació del comerç que és propietari l'usuari, aquí podem veure les comandes que queden pendents de recollir. El propietari pot accedir al llistat de comandes pendents de recollir i les comandes ja recollides, si prem en una comanda és redirigit a la finestra de informació de comanda. Si la comanda està per recollir el propietari té la opció de marcar una comanda com a rellida. També pot accedir a la finestra de gestió de productes del comerç



Imatge 14: Finestra per gestionar el comerç d'un usuari

5.6.12 Finestra de gestió de productes del comerç

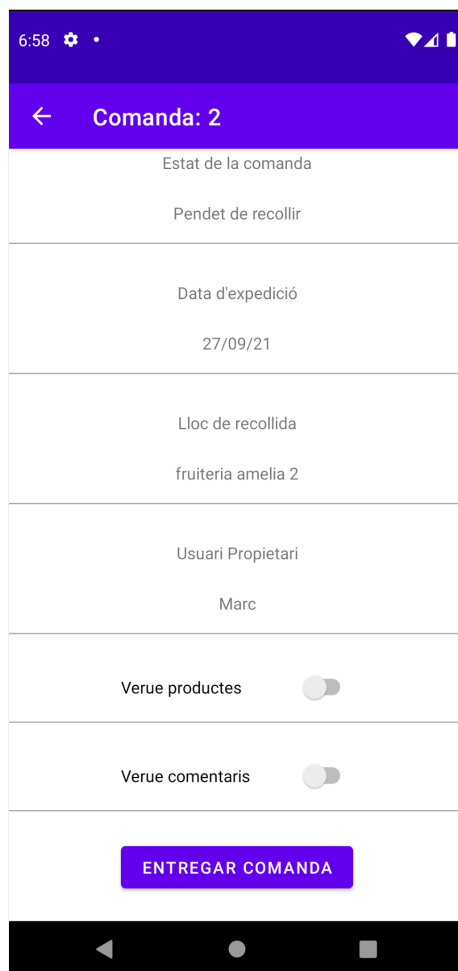
En aquesta finestra el propietari d'un comerç pot gestionar els productes del seu comerç, en concret el que pot fer és canviar el preu de productes, eliminar un producte del seu comerç i afegir un nou producte al comerç



Imatge 15: Finestra per gestionar productes d'un comerç

5.6.13 Finestra d'informació de comanda

En aquesta finestra es pot veure la informació de la comanda, també es pot afegir o canviar el comentari de la comanda per que ho pugui veure el propietari d'un comerç o l'usuari que ha sol·licitat la comanda. Si s'accedeix desde la finestra de gestió del comerç de l'usuari i la comanda està pendent de recollir, el propietari pot marcar la comanda com a recollida si l'usuari ha vingut a buscar-la.



Imatge 16: Finestra d'informació d'una comanda

6. Implementació

6.1 Tecnologies i llenguatges emprats

6.1.1 Java

El llenguatge utilitzat en la implementació tant de l'aplicació mòbil com de l'aplicació de serveis web és el JAVA.

S'ha usat el JAVA ja que és un llenguatge amb que ja havia realitzat algun projecte, coneixia i tenia molt per la mà, les eines de desenvolupament també han influït en escollir aquest llenguatge ja que tant *spring framework* com *Android studio* utilitzen aquest llenguatge.

6.1.2 JSON

Les aplicacions implementades utilitzen *JSON* per intercanvia dades.

JSON(JavaScript Object Notation) és un format de text que permet l'intercanvi de dades entre dos sistemes a través d'internet. Té una sintaxi molt senzilla formada per la combinació d'una clau-valor. Al codi 4 es pot veure un exemple del resultat *JSON* a la crida de l'*API* per retorna els productes del sistema.

```
{
  "id": 1,
  "nom": "poma",
  "descripcio": "poma acida",
  "tipus": "fruita",
  "preu": 5.0,
  "iconaProducte": "https://cdn.icon-icons.com/icons2/3007/PNG/128/discord_logo_icon_188422.png"
},
{
  "id": 2,
  "nom": "llimona",
  "descripcio": "llimona groga",
  "tipus": "fruita",
  "preu": 7.0,
  "iconaProducte": "https://cdn.icon-icons.com/icons2/3007/PNG/128/discord_logo_icon_188422.png"
}
```

Codi 4 : Resultat de la crida per retornar tots els productes del sistema

6.1.2 Hibernate JPA

Per gestionar la base de dades s'utilitza el framework de hibernate JPA

JPA (Java Persistence API) és una especificació per relacionar els objectes JAVA a una base de dades relacional.

El que ens permet *JPA* és mapejar una taula amb un classe i cada columna de la taula per un atribut de la classe.

Per realitzar el mapeig de les entitats s'utilitzen anotacions en la implementació.

En el exemple del codi 5 es pot veure com s'especifiquen aquestes anotacions, les anotacions de *JPA* són les implementacions que comencen amb una "@".

```
@Entity
@Table(name="productes")
public class Producte {

    @Id
    @Column(name="id")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private long id;

    @Column(name="nom", nullable=false, length=30)
    private String nom;

    @Column(name="descripcio")
    private String descripcio;

    @Column(name="tipus", nullable=false, length=30)
    private String tipus;

    @Column(name="preu", nullable=false)
    private Double preu;

    @Column(name="icona")
    private String iconaProducte;
}
```

Codi 5: Exemple d'implementació d'anotacions *JPA*

JPA només és una especificació, el framework que s'encarrega de traduir aquestes especificacions seqüències entenedores per una base de dades és *Hibernate*.

Hibernate és el framework encarregat de realitzar aquesta traducció. Aquest frameworks són coneguts com a ORM (*Object-Relational Mapping*).

6.2 Eines de desenvolupament

6.2.1 Spring framework

Per realitzar la base de dades i l'aplicació de serveis web s'ha usat *Spring framework*.

Spring framework és un framework de desenvolupament d'aplicacions Java, el que permet Spring és crear un codi d'alt rendiment i reutilitzable. Ens permet agilitzar i automatitzar molts processos. Permet la integració de *GitHub* i conté plantilles per agilitza la implemetació.

Els motius per escollir Spring com a eina de desenvolupament del *backend*, ha sigut la experiència prèvia amb aquest i la recomanació per part de la tutora d'utilitzar-lo.

És un framework molt simple i senzill d'utilitza i permet utilitzar el patró de disseny MVC i utilitzar els frameworks de *JPA* i *Hibernate*.

6.2.2 Android Studio

Per la implementació de l'aplicació mòbil s'ha usat el *IDE* de *Android Studio*.

Android Studio és un IDE creat per *Google* pensat per desenvolupar aplicacions *Android*. Està basat en *IntelliJ IDEA* i és gratuït. *Android Studio* inclou totes les eines necessàries per desenvolupar aplicacions *Android* en llenguatge *JAVA* o *Kotlin*

S'ha utilitzat aquest *Android Studio* per a la implementació de l'aplicació mòbil ja que és un IDE molt fàcil d'utilitzar, té un gran ventall d'eines que ajuden al desenvolupador a crear la seva pròpia aplicació. Un altre motiu per utilitzar-lo ha sigut la experiència prèvia amb aquest IDE. També cal remarcar que existeix una gran quantitat de documentació relacionada amb aquest IDE.

6.2.3 Postman

S'ha usat *Postman* per crear i provar totes les crides de l'aplicació de serveis web.

Postman és una eina que ens permet crear crides a una API. Les crides es poden guardar en col·leccions i compartir a altres usuaris de *Postman*.

A la codi 7 es pot veure un exemple d'una crida a l'aplicació de serveis web guardada en una col·lecció de *Postman*.

```
{
  "id": 1,
  "nom": "poma",
  "descripcio": "poma acida",
  "tipus": "fruita",
  "preu": 5.0,
  "iconaProducte": "https://cdn.icon-icons.com/icons2/3007/PNG/128/discord_logo_icon_188422.png"
},
{
  "id": 2,
  "nom": "llimona",
  "descripcio": "llimona groga",
  "tipus": "fruita",
  "preu": 7.0,
  "iconaProducte": "https://cdn.icon-icons.com/icons2/3007/PNG/128/discord_logo_icon_188422.png"
}
```

Codi 8: Exemple de resultat de la crida per obtenir els productes del sistema

Els motius per utilitzar *Postman* han sigut la experiència prèvia amb aquesta eina, i la facilitat d'ús que té per provar i guardar crides a APIs.

6.2.4 Github

S'ha usat *Github* per guardar i mantenir un control de versions sobre el *backend* i el *frontend*.

Les adreces URL dels repositoris són:

Backend: <https://github.com/marcazu/DProximitat>

Frontend: <https://github.com/marcazu/FrontDProximitat>

S'ha escollit *GitHub* com a eina de control de versions perquè ja que ja l'havia utilitzat anteriorment. Una altre motiu per escollir *GitHub* ha sigut la possibilitat de ser usat directament desde *Spring framework* i *Android studio*.

6.3 Crides implementades

A continuació es descriuen les crides més importants de l'aplicació de serveis wen. Algunes inclouen un exemple del JSON que retornen o del JSON que se li passa al ser cridades.

6.3.1 Crides a Productes

Nom	Obtenir producte
mètode	GET
Descripció	Obté la informació d'un producte concret
url	https://backendproximitat.herokuapp.com/productes/{id} on id és l'identificador d'un producte
Exemple de resultat	<pre>{ "id": 1, "nom": "poma", "descripcio": "poma acida", "tipus": "fruita", "preu": 5.0, "iconaProducte": "https://cdn.icon-icons.com/icons2/3007/PNG/128/discord_logo_icon_188422.png" }</pre>

Taula 38: Crida per obtenir un producte

Nom	Crear producte
mètode	POST
Descripció	Crea un producte al sistema
url	https://backendproximitat.herokuapp.com/productes
Exemple de input	<pre>{ "nom": "llimona", "descripcio": "llimona groga", "tipus": "fruita", "preu": 7, "iconaProducte": "https://cdn.icon-icons.com/icons2/3007/PNG/128/discord_logo_icon_188422.png" }</pre>

Taula 39: Crida per crear un producte

Nom	Modificar producte
mètode	PUT
Descripció	Modifica un producte concret
url	https://backendproximitat.herokuapp.com/productes
Exemple de input	<pre>{ "id": "1", "nom": "canviant producte", "descripcio": "holi", "tipus": "tipus modificat", "preu": 3.52 }</pre>

Taula 40: Crida per modificar un producte

Nom	Eliminar producte
mètode	DEL
Descripció	Elimina un producte concret del sistema
url	https://backendproximitat.herokuapp.com/productes

Taula 41: Crida per eliminar un producte

6.3.2 Crides a Comerços

Nom	Obtenir comerços
mètode	GET
Descripció	Obté la informació de tots els comerços del sistema
url	https://backendproximitat.herokuapp.com/botigues
Exemple de resultat	<pre>[{ "id": "1", "nom": "fruiteria amelia 1", "descripcio": "fruites i verdures de proximitat", "email": "fuiteriaAmelia@fakemail.com", "telefon": "123456789", "districte": "sants", "longitud": "12344.0", "latitud": "12345.0", "iconUrl": "https://cdn.icon-icons.com/icons2/3007/PNG/512/reddit_logo_icon_188423.png", "mainImageUrl": "https://imgur.com/a/Gyx8yHd", "numComandes": "1" }]</pre>

Taula 42: Crida per obtenir els comerços del sistema

Nom	Crear comerç
mètode	POST
Descripció	Crea un comerç al sistema
url	https://backendproximitat.herokuapp.com/botigues
Exemple de input	<pre>{ "nom": "fruiteria amelia 1", "descripcio": "fruites i verdures de proximitat", "email": "fuiteriaAmelia@fakemail.com", "districte": "sants", "telefon": "123456789", "longitud": 12344, "latitud": 12345, "iconUrl": "https://cdn.icon-icons.com/icons2/3007/PNG/512/reddit_logo_icon_188423.png", "mainImageUrl": "https://imgur.com/a/Gyx8yHd" }</pre>

Taula 43: Crida per crear un comerç

Nom	Obtenir productes d'un comerç
mètode	GET
Descripció	Obté tots els productes d'un comerç
url	https://backendproximitat.herokuapp.com/botigues
Exemple de resultat	<pre>[{ "id": "1", "nom": "poma", "descripcio": "poma acida", "tipus": "fruita", "preu": "5.0", "iconaProducte": "https://cdn.icon-icons.com/icons2/3007/PNG/128/discord_logo_icon_188422.png", "quantitat": "0" }, { "id": "2", "nom": "llimona", "descripcio": "llimona groga", "tipus": "fruita", "preu": "7.0", "iconaProducte": "https://cdn.icon-icons.com/icons2/3007/PNG/128/discord_logo_icon_188422.png", "quantitat": "0" }]</pre>

Taula 44: Crida per obtenir els productes d'un comerç

Nom	Obtenir comandes d'un comerç
mètode	GET
Descripció	Obté totes les comandes d'un comerç
url	https://backendproximitat.herokuapp.com/botigues
Exemple de input	<pre>[{ "id": "1", "entregada": false, "costTotal": "0", "botigaName": "fruiteria amelia 1", "propietari": "Marc", "dataExpedicio": "27/09/21", "comentari": "" }]</pre>

Taula 45: Crida per obtenir les comandes d'un comerç

Nom	Insertar producte al comerç
mètode	PUT
Descripció	Inserta un producte existent a un comerç existent
url	https://backendproximitat.herokuapp.com/botiques/{idComerç}/addProducte/{idProducte} on idComerç és l'identificador d'un comerç on idProducte és l'identificador d'un producte

Taula 46: Crida per insertar un producte a un comerç

6.3.3 Crides a Usuaris

Nom	Obtenir un usuari concret
mètode	GET
Descripció	Obté un usuari a partir del seu identificador a firebase
url	https://backendproximitat.herokuapp.com/users/firebase/{firebaseId} on firebaseId és l'identificador de firebase de l'usuari
Exemple de resultat	<pre>{ "id": "2", "nom": "Marc", "telefon": "123456789", "email": "95marc@fakemail.com", "iconaUrl": "soc una foto", "esbotiguer": false }</pre>

Taula 47: Crida per obtenir un usuari concret

Nom	Crear usuari
mètode	POST
Descripció	Crea un usuari al sistema
url	https://backendproximitat.herokuapp.com/users
Exemple de input	<pre>{ "nom": "Marc", "telefon": "123456789", "email": "95marc@fakemail.com", "iconaUrl": "", "firebaseUid": "RAXkvZ5HnTRGinSIknHOUCowsVq1" }</pre>

Taula 48: Crida per crear un usuari al sistema

Nom	Obtenir comandes d'usuari
mètode	GET
Descripció	Obté les comandes d'un usuari a partir del seu identificador a firebase
url	https://backendproximitat.herokuapp.com/users/firebase/{firebaseId}/comandes on firebaseId és l'identificador de firebase de l'usuari
Exemple de resultat	[<pre> { "id": "1", "entregada": false, "costTotal": "0", "botigaName": "fruiteria amelia 1", "propietari": "Marc", "dataExpedicio": "27/09/21", "comentari": "" }]</pre>

Taula 49: Crida per obtenir les comandes d'un usuari

Nom	Obtenir comerç propietat de l'usuari
mètode	GET
Descripció	Obté el comerç que és propietat d'un usuari a partir del seu id de firebase
url	https://backendproximitat.herokuapp.com/users/firebase/{firebaseId}/botigues on firebaseId és l'identificador de firebase de l'usuari
Exemple de resultat	{ <pre> "id": "1", "nom": "fruiteria amelia 1", "descripcio": "fruites i verdures de proximitat", "email": "fuiteriaAmelia@fakemail.com", "telefon": "123456789", "districte": "sants", "longitud": "12344.0", "latitud": "12345.0", "iconUrl": "https://cdn.icon-icons.com/icons2/3007/PNG/512/reddit_logo_icon_188423.png", "mainImageUrl": "https://imgur.com/a/Gyx8yHd", "numComandes": "1" }</pre>

Taula 50: Crida per obtenir el comerç que és propietat de l'usuari

Nom	Afegir usuari com a propietari d'un comerç
mètode	PUT
Descripció	Afegeix un botiguer com a propietari d'un comerç
url	https://backendproximitat.herokuapp.com/users/{idUsuari}/addBotiga/{idComerç} on idUsuari és l'identificador d'un usuari on idComerç és l'identificador d'un comerç

Taula 51: Crida per afegir un propietari com a propietari d'un comerç

Nom	Modificar un usuari
mètode	PUT
Descripció	Modifica la informació d'un usuari
url	https://backendproximitat.herokuapp.com/users
Exemple de input	<pre>{ "id": 1, "nom": "Arnau", "telefon": "9876321", "iconaUrl": "", "email": "ArnauEmail@fake.com" }</pre>

Taula 52: Crida per modificar un usuari

Nom	Eliminar usuari
mètode	DEL
Descripció	Elimina un usuari concret del sistema
url	https://backendproximitat.herokuapp.com/users

Taula 53: Crida per eliminar un usuari

6.3.4 Crides a Comandes

Nom	Obtenir Comandes
mètode	GET
Descripció	Obté les comandes del sistema
url	https://backenndproximitat.herokuapp.com/comandes
Exemple de resultat	<pre>[{ "id": "1", "entregada": false, "costTotal": "0", "botigaName": "fruiteria amelia 1", "propietari": "Marc", "dataExpedicio": "27/09/21", "comentari": "" }]</pre>

Taula 54: Crida per obtenir les comandes del sistema

Nom	Crear comandes
mètode	POST
Descripció	Crea una comanda al sistema
url	https://backenndproximitat.herokuapp.com/comandes
Exemple de input	<pre>{ "userID": "RAXkvZ5HnTRGinSlknHOUcowsVq1", "botigaID": "1", "dataExpedicio": "10/10/2021", "comentari": "", "producteQuantitat": [{ "productid": "1", "quantitat": "20" }, { "productid": "2", "quantitat": "40" }] }</pre>

Taula 55: Crida per crear una comanda al sistema

Nom	Obtenir productes d'una comanda
mètode	GET
Descripció	Obté els productes d'una comanda
url	https://backenndproximitat.herokuapp.com/comandes/{idComanda}/productes on idComanda és l'identificador d'una comanda
Exemple de resultat	<pre>[{ "id": "1", "nom": "poma", "descripcio": "poma acida", "tipus": "fruita", "preu": "5.0", "iconaProducte": "https://cdn.icon-icons.com/icons2/3007/PNG/128/discord_logo_icon_188422.png", "quantitat": "20" }, { "id": "2", "nom": "llimona", "descripcio": "llimona groga", "tipus": "fruita", "preu": "7.0", "iconaProducte": "https://cdn.icon-icons.com/icons2/3007/PNG/128/discord_logo_icon_188422.png", "quantitat": "30" }]</pre>

Taula 56: Crida per obtenir els productes d'una comanda

7. Proves

7.1 Proves a l'aplicació mòbil

L'aplicació s'ha anat provant a mesura que s'anaven implementant petites parts.

Per provar l'aplicació s'ha usat l'emulador que incorpora el IDE d'*Android studio*, aquest emulador permet un gran ventall de configuracions, en el cas d'aquest projecte s'ha configurat l'emulador per simular els telèfons *PIXEL 4* i *PIXEL 4 XL*.

El primer dispositiu simulat s'ha escollit perquè té una resolució idèntica al dispositiu que té el desenvolupador del projecte.

El segon dispositiu s'ha utilitzat per comprovar el funcionament de les finestres en un dispositiu de dimensions que el primer.

Un cop realitzada la major part de la implementació de l'aplicació, aquesta es va deixar provar a persones externes al projecte.

Amb l'opinió d'aquestes persones es van realitzar canvis en la interfície per millorar la seva usabilitat.

7.2 Proves a l'aplicació de serveis web

Les proves sobre les crides de l'aplicació de serveis web es van fer de forma local i remota. Per realitzar les proves s'ha usat *Postman*.

Un cop implementades un cert nombre de crides sobre un mateix sector de l'aplicació s'han realitzat proves sobre aquestes forma local. Per fer-ho s'ha desplegat l'aplicació i es configura el *Postman* per fer crides a la direcció local <http://localhost:8080>

Un cop comprovat el correcte funcionament, es desplega l'aplicació al servidor de *Heroku* i es realitzen les proves d'aquestes crides sobre la direcció del servidor.

8.Aspectes legals

8.1 Lleis aplicables al projecte

El projecte tractar amb informació personal dels usuaris, s'ha de poder garantir la protecció de les dades personals. Espanya i la unió europea tenen un conjunt de lleis que el projecte ha de seguir, aquestes lleis són les següents:

- Llei orgànica 3/2018 de protecció de dades personal i garantia dels drets digitals[17].
- Reglament general de protecció de dades (RGPD) de la unió europea[18].

Actualment no cal complir aquestes lleis ja que no es guarda informació personal real. En un futur si es vol desplegar en un entorn real, s'ha de complir que:

- Els usuaris no poden obtenir informació d'altres usuaris.
- S'ha de protegir la informació dels usuaris d'apropiacions indegudes.

8.2 Llicències

Per realitzar la implementació del projecte s'ha demanat llicències gratuïtes de programari de software, aquestes llicències són les següents:

- Llicència de Spring framework.
- Llicència de "Free"[19] de Heroku. Aquesta inclou:
 - l'ús de l'aplicació per un màxim de 1000 hores al mes.
 - Desplegament desde Github.
 - Domini personalitzat
 - Pla "Hobby dev" de PostgreSQL que permet un màxim de 10.000 files a la base de dades.

Cal remarcar que aquesta llicència només permet l'ús d'aplicacions no comercials. En el cas de desplegar l'aplicació a un entorn real, seria necessari canviar la llicència a llicència de producció, aquesta llicència té un cost de 25\$ al mes.

- Llicència d'Android studio 2020.3.1
- Llicència d'estudiant de Github.
- Llicència de Postman.

9. Seguiment del projecte

9.1 canvis de planificació i pressupost

En un principi el projecte estava pensat per ser finalitzat en un període de 4 mesos, l'entrega final estava pensada pel mes de Juny de 2021. Malauradament no es va poder finalitzar en aquest termini i es va optar per ampliar el termini de l'entrega fins al mes d'octubre de 2021.

Finalment el projecte ha tingut una durada de 8 mesos

També s'havia plantejat de realitzar una prova pilot a l'Associació de Comerciants de la *Marina*, malauradament es va perdre el contacte i no es va poder realitzar aquesta prova.

9.2 Execució real

En general s'ha realitzat les hores plantejades al principi del projecte.

En un principi s'esperava realitzar l'aplicació de serveis web prèviament abans de començar la implementació de l'aplicació mòbil.

L'aplicació de serveis web estava acabada i penjada al servidor de *Heroku* al mes de maig. Malauradament, al implementar l'aplicació mòbil es va veure que es necessitaven crear noves crides específiques i una reestructuració de com es tractava la comanda. Això va causar que al mes de Juny es tornés a implementar tota la part de la comanda.

Malgrat aquest impediment, aquesta aplicació estava acabada i penjada al servidor de *Heroku* a finals del mes de juny.

Finalment a finals del mes de setembre es va acabar la implementació de l'aplicació mòbil.

10. Sostenibilitat

10.1 Econòmica

Pel que fa a la gestió econòmica del projecte, hi ha una extensa avaluació dels costos del projecte, tant els costos materials com els humans.

No s'ha contemplat la viabilitat econòmica respecte a altres projectes de les mateixes característiques, això es deu al fet que és un projecte d'àmbit acadèmic destinat a ajudar al comerç de proximitat.

Respecte al temps dedicat al desenvolupament de les tasques, és l'idoni per la realització d'un projecte d'aquestes característiques, a més s'ha previst un cost de contingència en cas que hi hagi un imprevist en el cost de projecte.

10.2 Social

Aquest projecte té un gran impacte social, la seva intenció és la de millorar la viabilitat econòmica dels petits comerços.

El projecte es desenvolupa en un context de pandèmia global on el petit comerç ha patit una gran recessió a causa de la impossibilitat de realitzar compres físiques i a la manca de clientela.

Actualment aquest comerç està evolucionant a un comerç electrònic que les grans empreses i multinacionals estan dirigint (*Amazon, Globo, Ali Express...*), el petit comerç ha d'adaptar-se a les seves exigències o en alguns casos desaparèixer.

Per això crec que aquest projecte té una gran repercussió en la dimensió social, ja que permetrà als seus usuaris continuar utilitzant el comerç de proximitat que sempre ha utilitzat.

Quant a l'impacte que té el projecte al desenvolupador del projecte, s'espera que aquest ajudi més a valorar el comerç de proximitat i permeti no dependre tant de grans empreses dedicades al comerç electrònic.

10.3 Ambiental

El recurs més utilitzat en aquest projecte serà la utilització d'un ordinador portàtil, el qual tindrà una despesa energètica. Aquest consum energètic comporta un efecte negatiu pel medi ambient (producció d'electricitat i el seu impacte al medi en la seva generació).

Durant la vida útil del projecte suposarà un cost d'energia a cause del manteniment dels servidors d'internet. Aquest consum energètic es tindrà un impacte petit i per tant podem considerar que és menyspreable.

Els riscos que poden perjudicar l'aspecte ambiental són la necessitat de comprar un altre ordinador portàtil en cas que aquest deixes de funcionar. Suposaria un gran impacte ambiental, ja que la producció d'aparells electrònics suposa un gran impacte ambiental, malgrat això no s'espera un gran risc ambiental, ja que el material és relativament nou i no s'espera que pugui fallar.

11. Conclusions i treball futur

11.1 Competències tècniques i relació amb l'especialitat de software

- **CES1.1:** Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics [En profunditat].

Aquest s'ha centrat en desenvolupar un sistema complex. El sistema el componen una aplicació mòbil i una aplicació amb serveis web. Alguns aspectes a destacar de la complexitat del sistema són: La creació d'una base de dades per tota el sistema desenvolupat, la implementació de serveis web per accedir a la base de dades i la implementació d'una interfície gràfica per dispositius *Android* i la integració de les crides del servei web.

- **CES1.5:** Especificar, dissenyar, implementar i avaluar bases de dades[Bastant].

El sistema està compost per una base de dades pròpia les dades de les entitats que formen el sistema, és una base de dades relacional gestionada per PostgreSQL. La base de dades s'ha especificat i dissenyat a partir del model conceptual de les dades del sistema (*Diagrama 3*). La implementació de la base de dades s'ha fet a través de *Hibernate* i *Jpa*.

- **CES1.6:** Administrar bases de dades[Una mica].

Per administrar la base de dades s'ha usat seqüències SQL en el terminal de comandes del servidor que allotja la base de dades. També s'ha creat una aplicació de serveis web amb crides que permeten interactuar amb la base de dades a través d'objectes sense necessitat d'utilitzar seqüències SQL.

- **CES1.7:** Controlar la qualitat i dissenyar proves en la producció de software[Una mica].

-

Per verificar el correcte funcionament del sistema, s'ha realitzat proves de la base de dades i l'aplicació de serveis web a nivell local abans de penjar el sistema al servidor extern de *Heroku*. També s'ha realitzat proves a la interfície mòbil per comprovar el funcionament dels seus components

- **CES2.1:** Definir i gestionar els requisits d'un sistema software[Bastant].

Abans de realitzar la implementació dels components del projecte, s'han definit els requisits funcionals i no funcionals del sistema. Per especificar els requisits del projecte, es va entrar en contacte amb una associació de comerciants de la Marina.

11.2 Conclusions personals

A nivell personal puc dir que tinc una valoració bastant positiva del projecte.

Tot i que s'ha aconseguit desenvolupar la major part del projecte, considero que encara queden alguns aspectes que caldria canviar. Per exemple, un dels punts on caldria realitzar canvis és a l'aplicació mòbil. Es podria millorar la interfície i realitzant algun canvi a nivell de codi intern.

El projecte m'ha ajudat a consolidar molts dels coneixements apresos durant el grau d'enginyeria informàtica de la FIB. Sens dubte, la realització d'aquest projecte ha sigut l'experiència més rellevant que he tingut al llarg de la meva vida acadèmica.

El projecte també m'ha servit per adquirir un gran coneixement en la implementació d'aplicacions mòbil Android i aplicacions de serveis web.

11.3 Treball futur

11.3.1 Millorar la seguretat de les crides

A l'aplicació s'utilitzen dades personals, a falta de temps, no s'ha pogut implementar cap mesura de seguretat a les crides de l'aplicació de serveis web.

Un dels sistemes que es poden implementar és la identificació usant JWT.

JWT és un acrònim de JSON Web Token, es tracta d'utilitzar Tokens per permetre l'accés a les crides de l'aplicació de serveis web. L'objectiu seria només deixar utilitzar les crides als usuaris que estan registrats a l'aplicació

11.3.2 Sistema de puntuació

Una funcionalitat que es podria incorporar a l'aplicació és un sistema de puntuació i comentaris.

L'objectiu seria permetre els clients puntuar comerços i productes, també seria la d'afegir una sistema de comentaris on es pugui deixar opinions sobre aquets. Els propietaris de comerços tindrien visibilitat de les puntuacions i les opinions dels clients.

Per afegir aquesta funcionalitat caldria canviar el model conceptual del sistema. També caldria canviar al lògica de l'aplicació mòbil i afegir noves crides a l'aplicació de serveis web.

11.3.3 Trobar un grup de comerciants per provar l'aplicació

Una part del treball que no s'ha pogut realitzar és la prova del projecte a un conjunt de comerços reals.

En un futur s'ha de trobat una associació de comerciants disposada a realitzar una prova pilot de l'aplicació en els comerços que integren l'associació

Desde el punto de vista de l'aplicació seria interessant realitzar aquesta prova pilot per obtenir *feedback* del funcionament d'aquesta i realitzar els canvis que es vegin oportuns.

Referències

1. La compra. App de mòbil.
<https://play.google.com/store/apps/details?id=com.interactiu.lacompra.app&hl=es&gl=US>
2. Everis. Consultoria internacional. <https://www.everis.com/spain/es/home-spain>
3. Togoodtogo. Aplicació per comprar menjant sobrant a comerços.
<https://www.toogoodtogo.es>
4. Associació de comerciants de les *Corts de Sant Ramon i Maternitat*. Associació de comerciants <https://www.les-corts.com/es>
5. Associació de comerciants cor de les Corts. Associació de comerciants.
<https://www.corlescorts.com>
6. Eix comercial de Sants. Associació de comerciants.
<https://www.eixsantslescorts.com>
7. Associació de comerciants de la marina. Associació de comerciants.
<http://www.comerciantslamarina.cat>
8. Mòdul de GEP. Guia per fer la gestió de projectes.
<https://www.fib.upc.edu/sites/fib/files/documents/estudis/guia-gep-raco-fib.pdf>
9. Agile. Metodologia de treball.
<https://www.santaluciaimpulsa.es/metodologia-agile-que-es-para-que-sirve/>
10. Android studio. Ide per desenvolupar aplicacions Android.
https://developer.android.com/studio?hl=es&gclid=Cj0KCQjwtrSLBhCLARIsACh6Rmgrn4OvqgsbGs0SylwV-2zdkVcpqlc34rdngl8OZf9j_4dc9YDZwQaAi9iEALw_wcB&gclid=aw.ds
11. Postman. Eina per provar crides d'un servei web. <https://www.postman.com>
12. Swagger. Eina per documentar serveis web. <https://swagger.io>
13. Github. Sistema per realitzar control de versions del codi. <https://github.com>
14. Microsoft 365. Conjunt de programes informàtics de *microsoft*.
<https://www.microsoft.com/es-es/microsoft-365>
15. Barcelona activa. Agència de l'ajuntament de Barcelona.
<https://www.barcelonactiva.cat>

16. Firebase auth. API de Google que ofereix serveis com registrar usuaris.
<https://firebase.google.com/products/auth>
17. Llei orgànica 3/2018 de protecció de dades personal i garantia dels drets digitals. Llei espanyola per protegir la informació dels usuaris a internet.
<https://apdcat.gencat.cat/web/.content/01-autoritat/normativa/documentos/LOPDGDD-cat-modificada.pdf>
18. Reglament general de protecció de dades (RGPD). Reglament per la protecció de dades a la unió europea. https://apdcat.gencat.cat/ca/drets_i_obligacions/rgpd
19. Pressupostos de Heroku. Llistat de les llicències disponibles a Heroku.
<https://www.heroku.com/pricing>