



UNIVERSITAT DE
BARCELONA

Treball final de grau

GRAU DE MATEMÀTIQUES

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Desenvolupament d'una pàgina
web sobre reserva dolça

Autor: Biel Lopez Baos

Tutor: Xavier Baró Solé

Realitzat a: Departament de
Matemàtiques i Informàtica

Barcelona, Gener 2025

Índex

1	Introducció	1
1.1	Context i justificació del projecte	1
1.2	Descripció general de l'aplicació	1
1.3	Objectius del projecte:	2
2	Planificació	4
2.1	Diagrama de Gantt	4
2.2	Nova planificació basada en la realitat	4
2.3	Fases del projecte	7
2.3.1	Planificació inicial	7
2.3.2	Execució	7
2.3.3	Validació	8
3	Requeriments	9
3.1	Requisits Funcionals	9
3.1.1	Registre d'Usuaris:	9
3.1.2	Iniciar Sessió:	9
3.1.3	Gestió de Reserves:	10
3.1.4	Visualització d'Horaris:	10
3.1.5	Administració d'Usuaris i Permisos:	11
3.1.6	Gestió de Productes:	11
3.1.7	Gestió de Detalls de la Reserva:	12
3.1.8	Gestió de Clients:	12
3.1.9	Gestió de Dependents:	12
3.1.10	Canvi d'idioma:	13
3.1.11	Categoría Productes	13
3.1.12	Gestió Estadístiques	14
4	Tecnologies Utilitzades	15
4.0.1	Visual Studio Code (VSCode)	15
4.1	Front-End	15
4.1.1	Ionic	15
4.1.2	MUI (Material-UI)	16
4.1.3	CSS	16
4.1.4	React	17
4.1.5	React Router	18
4.1.6	React Calendar	18
4.1.7	React i18next	19
4.2	Back-End	19
4.2.1	FastAPI	19
4.2.2	Bcrypt	20
4.2.3	Fetch API	20
4.3	Postman	21
4.4	Base de Dades	21

4.4.1	Firestore	21
4.4.2	Firestore Admin SDK	22
4.4.3	Firestore Firestore	22
4.4.4	Google Cloud Storage	22
4.5	Deploy	23
4.5.1	Docker Compose	23
4.5.2	Render	24
4.6	Diagrama de Tecnologies	25
5	Model de dades ER	26
5.1	Distribució del Model de Base de Dades:	26
5.2	Entitats Principals:	26
5.2.1	BOTIGUES:	26
5.2.2	DEPENDENTES:	26
5.2.3	CLIENTS:	26
5.2.4	PRODUCTES:	27
5.2.5	PRODUCTES.RESERVA:	27
5.2.6	RESERVES:	27
5.2.7	OBRADOR:	28
5.3	Relacions:	28
6	Front-end	31
6.1	Explicació global de la vista	31
6.2	Explicació individual de les pantalles	32
6.2.1	Administrador Gestió dels productes	32
6.2.2	Obrador Gestió de les Reserves	32
6.2.3	Reserva	32
6.2.4	Administrador Gestió de les Reserves	32
6.2.5	Interfície de les botigues	32
6.2.6	Administració Gestió de les Botigues	33
6.2.7	Ver Detalles	33
6.3	Internacionalització i Canvi de Llenguatge	34
6.4	Pantallas Principales	35
6.4.1	Admin	35
6.4.2	Tab1	37
6.4.3	Obrador	39
7	Back-end	42
7.1	Explicació Global del Backend	42
7.2	Funcionalitats implementades al backend	42
7.3	Disseny i estructura del backend	43
8	Proves unitaries	44
8.1	FrontEnd	44
8.1.1	Sugerencias basadas en el feedback de los usuarios	44
8.1.2	Història 1: Anna, la dependenta multitask	44
8.1.3	Història 2: Jordi, el forner de l'obrador	45

8.1.4	Història 3: Laura, la clienta curiosa	45
8.2	Backend-Proves i Validació	46
8.2.1	Enfocament de les proves	46
8.2.2	Tipus de proves realitzades	46
8.2.3	Motivació darrere de les proves	47
8.2.4	Cobertura i anàlisi de proves amb Coverage	47
8.3	Cobertura per secció de main.py	48
8.4	Histograma de Cobertura	49
9	Deploy	50
9.1	Desplegament del Sistema	50
9.1.1	Desplegament del Backend	50
9.1.2	Desplegament del Frontend	50
9.1.3	Integració entre Backend i Frontend	51
9.1.4	Resultats del Desplegament	51
10	Resultats/conclusions/Treball futur	52
10.1	Conclusions	52
10.2	Treball a futur	53
10.2.1	Ampliar el sistema de notificacions en temps real	53
10.2.2	Millorar el sistema de reportes i estadístiques	53
10.2.3	Integració amb sistemes de pagament	53
10.2.4	Cobertura dels Tests	53
10.2.5	Gestió d'estoc	53
10.2.6	Evolució	54
10.2.7	Canvi d'Idioma	54
10.2.8	Optimització del funcionament dels filtres	54
10.2.9	Seguretat	54
11	ANEX	55

1 Introducció

1.1 Context i justificació del projecte

En l'entorn empresarial contemporani, la gestió eficient de les reserves s'ha convertit en un aspecte crític per l'èxit i la competitivitat de les empreses, especialment en el sector minorista i de serveis. La capacitat de coordinar i gestionar adequadament les reserves entre diferents punts de servei, com ara botigues físiques i centres de producció centralitzats, és fonamental per optimitzar l'operació i millorar l'experiència del client. En aquest context, el present treball de fi de grau es centra en el desenvolupament d'una solució tecnològica: una aplicació mòbil dissenyada per a la gestió integral de reserves entre botigues i un obrador centralitzat.

La proposta d'aquest projecte sorgeix com a resposta a la necessitat identificada en l'empresa del meu pare, que té aquest problema on busco millorar l'eficiència i la precisió en la gestió de reserves, així com de facilitar la comunicació i la coordinació entre els diferents punts de servei i el centre de producció. L'aplicació mòbil proposada té com a objectiu principal oferir una plataforma intuïtiva i fàcil d'utilitzar que permeti a les botigues realitzar i gestionar reserves de manera eficient, mentre que l'obrador centralitzat pot coordinar la producció i la distribució de productes d'acord amb la demanda i les reserves rebudes.

Per al desenvolupament d'aquesta aplicació, es proposa utilitzar el framework Ionic, que ofereix una combinació de tecnologies web estàndard (HTML, CSS, JavaScript) juntament amb tecnologies mòbils com AngularJS, React i Cordova, el que permet crear aplicacions mòbils multiplataforma amb un disseny i rendiment natiu. A més, s'utilitzarà Firebase com a plataforma de desenvolupament d'aplicacions mòbils i base de dades a la núvol, aprofitant les seves característiques de temps real i sincronització automàtica per garantir la disponibilitat i consistència de les dades en tots els dispositius i punts de servei.

El desenvolupament d'aquesta aplicació mòbil no només implica la implementació d'una solució tecnològica, sinó també la consideració d'aspectes com l'experiència de l'usuari, la seguretat de les dades i l'escalabilitat del sistema. Per tant, aquest treball de fi de grau abordarà tant els aspectes tècnics com els aspectes relacionats amb el disseny, la usabilitat i la viabilitat de la implementació en un entorn empresarial real.

1.2 Descripció general de l'aplicació

L'aplicació desenvolupada és una solució de gestió de reserves destinada a connectar i coordinar les operacions entre les diferents botigues d'una empresa i un obrador centralitzat. Aquesta aplicació mòbil, implementada amb les tecnologies Ionic, React i Firebase, ofereix una interfície d'usuari intuïtiva i amigable que facilita la realització i gestió de reserves, la comunicació interna entre botigues i l'obrador, així com l'administració de productes i usuaris.

L'aplicació es divideix en tres rols principals: botiga, obrador i administrador. Cada rol té accés a diferents funcionalitats segons les seves necessitats:

- **Rols de botiga:** Els usuaris de les botigues poden realitzar reserves de

productes a l'obrador, consultar l'estat de les seves reserves, modificar les quantitats i veure els detalls de les seves comandes en temps real. També poden gestionar la seva pròpia llista de productes disponibles per a la reserva.

- **Rols de l'obrador:** L'obrador pot visualitzar totes les reserves realitzades per les diferents botigues, actualitzar l'estat de les reserves (per exemple, en procés, completades, etc.), gestionar els productes disponibles i coordinar la distribució dels mateixos segons les necessitats de les botigues.
- **Rols d'administrador:** Els administradors tenen la capacitat de gestionar tots els aspectes de l'aplicació, incloent-hi l'administració d'usuaris, botigues, productes i dependents. A més, poden accedir a informes i estadístiques detallades sobre l'ús de l'aplicació, les reserves realitzades i la popularitat dels productes.

L'aplicació utilitza Firebase Firestore com a base de dades en temps real, la qual cosa permet la sincronització instantània de la informació entre totes les botigues i l'obrador. Aquesta característica assegura que les dades estiguin sempre actualitzades i disponibles, millorant l'eficiència i la comunicació interna de l'empresa.

A més, l'aplicació ofereix una gestió segura de l'autenticació d'usuaris mitjançant Firebase i la utilització de bcrypt per encriptar les contrasenyes. Això garanteix que només els usuaris autoritzats puguin accedir a la informació, i que les dades sensibles estiguin protegides en tot moment.

1.3 Objectius del projecte:

L'objectiu principal d'aquest projecte és desenvolupar una aplicació mòbil eficient i intuïtiva que permeti la gestió integral de reserves entre diverses botigues i un obrador centralitzat, amb la finalitat de millorar la coordinació i comunicació entre aquests punts de servei i optimitzar l'operació i distribució dels productes. Per aconseguir aquest objectiu general, es plantegen els següents objectius específics:

- **Facilitar la gestió de reserves:** Proporcionar a les botigues una interfície senzilla i fàcil d'usar per realitzar, modificar i gestionar les seves reserves de productes, assegurant un procés ràpid i intuïtiu que redueixi els errors i millori l'eficiència del procés de reserva.
- **Millorar la comunicació entre les botigues i l'obrador:** Implementar una plataforma que permeti la transmissió de dades en temps real, facilitant la coordinació entre les botigues i l'obrador centralitzat, per garantir la disponibilitat de productes d'acord amb les reserves i la demanda.
- **Proporcionar eines de gestió per a l'administració:** Desenvolupar un panell de control per als administradors que permeti la gestió d'usuaris, productes i botigues, així com la supervisió de les reserves realitzades i l'assignació de permisos a diferents rols dins del sistema.

- **Optimitzar la gestió de productes:** Permetre que l'obrador i les botigues puguin gestionar els productes disponibles per a la reserva, incloent-hi la possibilitat d'afegir, modificar i eliminar productes de forma fàcil i centralitzada.
- **Proporcionar informes i estadístiques:** Incloure funcionalitats que permetin als administradors obtenir informes detallats sobre les reserves, l'estat dels productes i l'ús de l'aplicació per prendre decisions informades que millorin l'eficiència de l'operació.

2 Planificació

La *figura 1* mostra el primer diagrama de Gantt, creat amb l'objectiu de planificar detalladament les tasques del meu Treball de Final de Grau. Aquest pla inicial estava estructurat en diferents fases, com ara:

- Planificació.
- Recollida de requeriments.
- Elecció de tecnologies.
- Desenvolupament del model de dades.
- Creació del *frontend* i *backend*.
- Proves unitàries.
- Redacció de la memòria.
- Preparació del debat final.

Cadascuna d'aquestes fases tenia unes dates inicials i finals assignades, amb l'objectiu d'assegurar un progrés constant i metòdic al llarg del projecte. No obstant això, per motius personals, no vaig poder complir estrictament aquesta planificació inicial, fet que va provocar que algunes tasques no es completessin dins dels terminis previstos.

2.1 Diagrama de Gantt

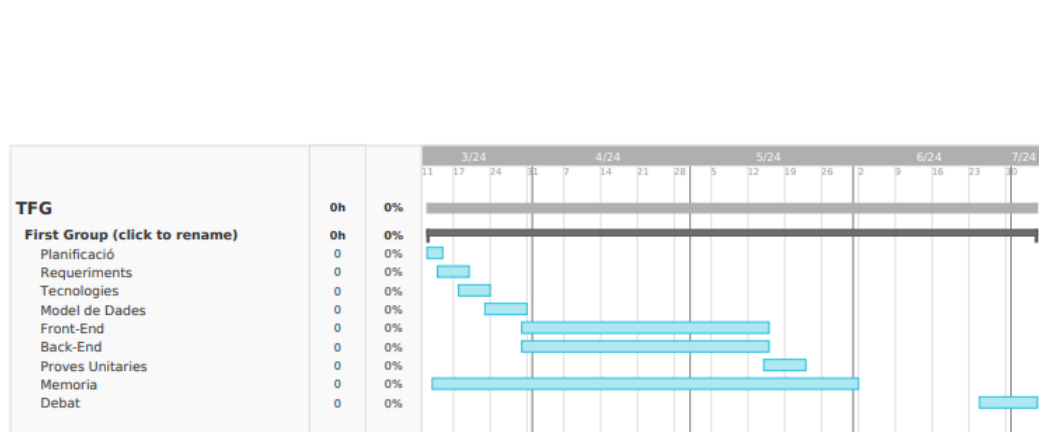


Figura 1: Diagrama de Gantt

2.2 Nova planificació basada en la realitat

Davant les circumstàncies que van sorgir durant el desenvolupament del projecte, vaig haver de replantejar la planificació inicial. Això es reflecteix en el segon diagrama de Gantt (*figura 2*, *figura 3*, *figura 4*), que ajusta les dates i prioritats de

les tasques en funció del temps disponible i les necessitats més urgents del projecte (dividit en tres imatges). Les modificacions principals inclouen:

- El desenvolupament del *backend* i del *frontend* es va convertir en una prioritat clau, ja que són el nucli funcional del sistema.
- Les proves unitàries, tot i ser inicialment planificades amb més profunditat, es van reduir en abast, implementant només les mínimes necessàries per garantir la funcionalitat crítica.
- La redacció de la memòria es va dur a terme paral·lelament amb altres tasques, adaptant-se al progrés general del projecte.

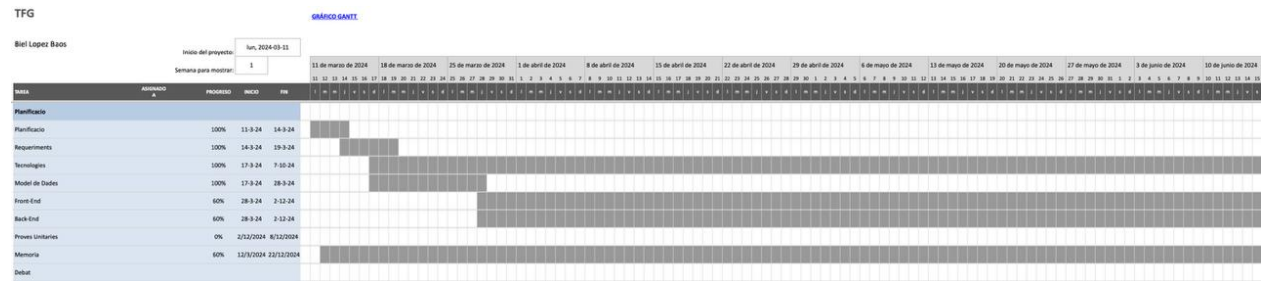


Figura 2: Diagrama Gantt 1

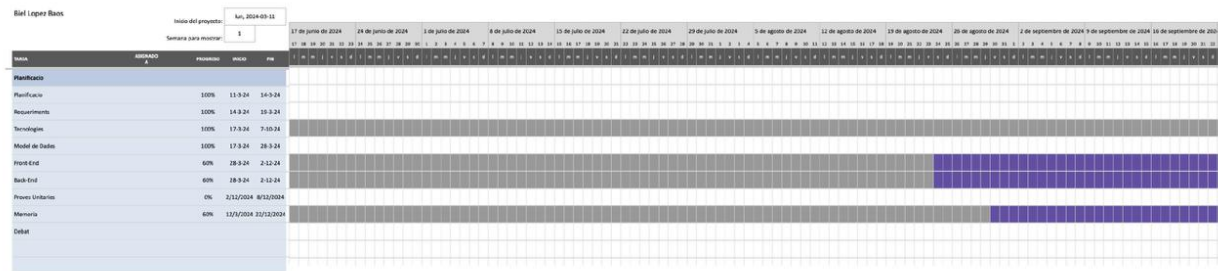


Figura 3: Diagrama Gantt 2

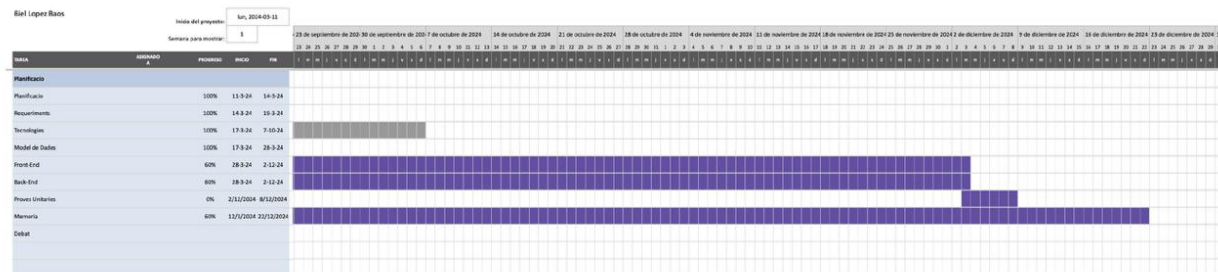


Figura 4: Diagrama Gantt 3

Aquesta reorganització va permetre enfocar els esforços en les parts essencials del projecte, assegurant-ne la conclusió dins dels terminis establerts per a la presentació final. Així, el segon diagrama de Gantt reflecteix aquesta realitat ajustada i posa en evidència la importància de la flexibilitat i adaptabilitat durant el desenvolupament.

2.3 Fases del projecte

2.3.1 Planificació inicial

Durant aquesta fase, es realitza una anàlisi detallada dels requisits del projecte i es defineixen els objectius i les funcionalitats principals de l'aplicació. Aquesta etapa és crucial per establir les bases del projecte i assegurar-se que totes les parts interessades tinguin una comprensió clara del que s'espera assolir. Aquesta fase inclou les següents tasques:

- **Anàlisi de requisits:** Identificació de les necessitats específiques de l'empresa i les funcionalitats que ha de tenir l'aplicació, com la gestió de reserves, la coordinació entre botigues i l'obra, i l'administració de productes i usuaris.
- **Definició de l'abast del projecte:** Delimitació de les funcionalitats que es desenvoluparan en l'aplicació i establiment de les limitacions del projecte.
- **Elaboració del model de dades ER:** Disseny del model de dades que defineix les entitats principals, les seves relacions i els atributs necessaris per a l'aplicació.
- **Elecció de les tecnologies:** Selecció de les tecnologies que s'utilitzaran per al desenvolupament, com Ionic, React, Firebase i FastAPI, basant-se en els requisits identificats.
- **Creació del diagrama de Gantt:** Elaboració d'un diagrama de Gantt que detalli la planificació temporal de cada fase del projecte, definint els terminis per a cadascuna de les tasques identificades.

2.3.2 Execució

En aquesta fase, es duu a terme el desenvolupament del projecte seguint la planificació establerta. Aquesta etapa s'organitza en diverses subfases per garantir un desenvolupament eficient i ordenat:

- **Disseny de l'arquitectura del sistema:** Creació de l'arquitectura de l'aplicació, incloent-hi la definició de la interacció entre el front-end i el back-end, així com la manera com es connectaran amb la base de dades de Firebase.
- **Desenvolupament del backend:** Implementació del backend amb FastAPI i integració amb Firebase per a la gestió d'autenticació, reserves, productes, usuaris i botigues. Creació de les diferents rutes per permetre la comunicació entre el client i el servidor i implementació de la lògica per a la gestió de dades.

- **Desenvolupament del frontend:** Creació de la interfície d'usuari utilitzant Ionic i React, assegurant que totes les funcionalitats previstes (gestió de reserves, productes, usuaris, etc.) estiguin disponibles i siguin fàcils d'utilitzar. En aquesta fase, també s'incorporen llibreries com MUI per al disseny d'elements visuals.
- **Integració de la funcionalitat de notifikacions:** Implementació d'un sistema de notifikacions en temps real utilitzant Firebase per informar entre usuaris dels canvis en els estats de les reserves.
- **Desenvolupament de les funcionalitats d'administració:** Creació d'un panell de control per als administradors per gestionar usuaris, productes, botigues i veure informes detallats sobre l'activitat del sistema.
- **Proves unitàries i d'integració:** Realització de proves unitàries per verificar que les diferents parts de l'aplicació funcionen correctament de manera individual. A continuació, es duen a terme proves d'integració per assegurar que totes les funcionalitats treballen correctament conjuntament.
- **Optimització i millores:** Anàlisi del rendiment de l'aplicació i optimització del codi per millorar la velocitat i l'eficiència del sistema. En aquesta etapa, es fan ajustaments a l'experiència d'usuari i a la interfície per assegurar la usabilitat de l'aplicació.

2.3.3 Validació

Aquesta és l'etapa final del projecte, en la qual es realitzen proves exhaustives per assegurar que l'aplicació compleix amb els requisits establerts i que està preparada per a la seva implementació en un entorn real. Les tasques d'aquesta fase inclouen:

- **Proves de validació i acceptació:** Realització de proves de validació amb usuaris reals per assegurar que totes les funcionalitats compleixen les expectatives i que l'aplicació ofereix una experiència d'usuari intuïtiva i eficient.
- **Resolució d'errors i ajustos finals:** Correcció de qualsevol error o problema detectat durant les proves de validació, així com realització d'ajustos finals per garantir el correcte funcionament de l'aplicació.
- **Preparació de la documentació tècnica:** Elaboració de la documentació tècnica del projecte, incloent-hi la descripció del sistema, el model de dades, les funcionalitats implementades, les tecnologies utilitzades i les proves realitzades.
- **Implementació i desplegament:** Preparació de l'aplicació per al seu llançament en un entorn de producció, incloent-hi la configuració del servidor, la càrrega de les dades inicials i el desplegament de l'aplicació en la plataforma final.
- **Entrega i presentació del projecte:** Preparació de la presentació final del projecte per a la seva avaluació, incloent-hi la redacció de l'informe final i la realització d'una demostració de l'aplicació desenvolupada.

Aquestes fases proporcionen una estructura clara per al desenvolupament del projecte, assegurant que cada etapa sigui abordada de manera eficient i sistemàtica, amb l'objectiu de lliurar una aplicació que compleixi amb els requisits establerts i ofereixi una solució efectiva per a la gestió de reserves en l'entorn de l'empresa.

3 Requeriments

3.1 Requisits Funcionals

3.1.1 Registre d'Usuaris:

Descripció: L'aplicació ha de permetre el registre de noves botigues, assignant a cada botiga un compte d'usuari únic que permeti l'accés a les seves funcionalitats.

Detalls:

- Cada nova botiga registrarà un compte d'usuari a l'aplicació amb un nom de botiga, número de telèfon i una contrasenya.
- El registre de la botiga es farà des del panell d'administració, on l'administrador afegirà manualment la botiga a través d'un formulari que recopilarà la informació bàsica com el nom, el número de telèfon i la contrasenya.
- La contrasenya de la botiga es xifra i s'emmagatzema de manera segura utilitzant un procés de hashing per garantir la seguretat de les dades.
- Cada botiga tindrà un identificador únic que permetrà la gestió de les seves reserves i la consulta de la informació associada a la mateixa.
- Una vegada registrades, les botigues poden accedir a l'aplicació amb les seves credencials per començar a gestionar les seves reserves i clients.

3.1.2 Iniciar Sessió:

Descripció: Els usuaris han de poder iniciar sessió al sistema mitjançant les seves credencials per accedir a les funcionalitats segons el seu rol (administrador, obrador, botiga).

Detalls:

- L'aplicació permet als usuaris iniciar sessió amb un identificador únic (ID de botiga, obrador o administrador) i una contrasenya segura que serà validada en el sistema.
- Les credencials s'emmagatzemen de manera segura a la base de dades centralitzada (Firebase Firestore) i es verifiquen cada vegada que un usuari intenta accedir al sistema.

- Segons el rol de l'usuari, es mostra una interfície adaptada amb les funcions específiques que li corresponen.
- En cas que les credencials siguin incorrectes, es mostrarà un missatge d'error a l'usuari, indicant-li que revisi la informació proporcionada.
- Després d'iniciar sessió correctament, l'usuari serà redirigit a la interfície adequada segons el seu rol, garantint una experiència d'usuari personalitzada.

3.1.3 Gestió de Reserves:

Descripció: Els usuaris autoritzats a les botigues i l'obraor han de poder gestionar les reserves de productes de manera eficient, permetent la creació, modificació i cancel·lació de les mateixes.

Detalls:

- Els usuaris poden crear noves reserves seleccionant el client, els productes que es volen reservar, la quantitat de cada producte i l'estat inicial de la reserva.
- Cada reserva contindrà informació com l'ID del client, la data i hora de la reserva, la llista de productes reservats, l'estat actual de la reserva (pendent, confirmada, en procés, completada o cancel·lada), i qualsevol nota addicional que sigui necessària.
- L'aplicació ofereix una interfície intuïtiva per visualitzar i gestionar les reserves existents, amb la possibilitat d'afegir o eliminar productes de la reserva segons les sol·licituds dels clients.
- Els usuaris poden filtrar les reserves per data, estat, o client, facilitant la gestió de les comandes segons les necessitats de la botiga.
- En cas de cancel·lació d'una reserva, els productes reservats es retiren automàticament del sistema i l'estat de la reserva es marca com "cancel·lada".

3.1.4 Visualització d'Horaris:

Descripció: Els usuaris han de poder veure les reserves de productes organitzades per data i botiga per facilitar la gestió del personal i les comandes.

Detalls:

- Els usuaris poden seleccionar una data específica a través d'un calendari interactiu que mostra totes les reserves programades per aquell dia, facilitant la planificació del treball.
- Els administradors poden filtrar les reserves per botiga, client o dependent, permetent obtenir una visió detallada de l'estat de les reserves a cada localitat.

- La informació de les reserves es mostra de manera clara i ordenada, incloent els noms dels clients, la llista de productes reservats, les quantitats i l'estat actual de la reserva.
- El sistema ofereix la possibilitat de canviar la visualització entre diferents modes (per dia, setmana o mes), oferint flexibilitat en la gestió dels horaris.

3.1.5 Administració d'Usuaris i Permisos:

Descripció: Els administradors han de tenir la capacitat de gestionar els usuaris i els seus permisos dins de l'aplicació per assegurar un ús correcte del sistema.

Detalls:

- Els administradors poden crear, editar i eliminar usuaris del sistema, incloent-hi dependents, altres administradors i botigues.
- Els administradors tenen la capacitat de modificar els permisos d'accés de cada usuari, assegurant que només puguin accedir a les funcionalitats que els corresponen segons el seu rol.
- També poden canviar les contrasenyes dels usuaris i actualitzar la informació de contacte de les botigues, garantint la seguretat i precisió de la informació del sistema.
- En casos de reorganització interna, els administradors poden transferir dependents d'una botiga a una altra, ajustant els permisos i accessos segons sigui necessari.

3.1.6 Gestió de Productes:

Descripció: Els administradors han de poder gestionar els productes disponibles per a la reserva a cada botiga de manera eficient.

Detalls:

- Els administradors poden afegir nous productes al sistema, introduint la informació com el nom, pes, i descripció del producte.
- També poden editar els productes existents per ajustar-los segons la disponibilitat o modificar-los si es produeixen canvis en les característiques del producte.
- Els administradors tenen la possibilitat d'eliminar productes que ja no estiguin disponibles o que hagin deixat de formar part de l'oferta de la botiga, assegurant que només els productes actuals es mostrin a les reserves.
- Els canvis realitzats en els productes es reflecteixen de manera immediata a totes les reserves i botigues que els utilitzen, mantenint la informació actualitzada.

3.1.7 Gestió de Detalls de la Reserva:

Descripció: Les dependents de les botigues han de poder veure i modificar els detalls de les reserves, mentre que l'obrador ha de poder visualitzar i imprimir les reserves.

Detalls:

- Les dependents poden consultar els detalls complets d'una reserva, incloent la informació del client, els productes reservats, la data, i l'estat actual de la reserva.
- També poden modificar la llista de productes de la reserva, afegint nous productes o canviant la quantitat dels ja existents segons les sol·licituds dels clients.
- El obrador té la capacitat d'imprimir els detalls de les reserves per facilitar la preparació i la gestió de les comandes a mesura que es realitzen, assegurant que tot el personal tingui la informació necessària.
- Els usuaris poden canviar l'estat de la reserva de pendent a confirmada, en procés o completada, oferint un seguiment de l'estat de les comandes en temps real.

3.1.8 Gestió de Clients:

Descripció: Els administradors i les dependents han de poder gestionar els clients que fan reserves, assegurant que la informació estigui actualitzada i sigui precisa.

Detalls:

- Els administradors poden afegir nous clients al sistema, introduint el seu nom complet i número de telèfon.
- Les dependents poden veure i seleccionar clients existents a l'hora de crear una nova reserva, garantint que totes les reserves estiguin associades a un client concret.
- Si un client repeteix la reserva, la seva informació es troba guardada al sistema, facilitant la gestió de noves comandes.

3.1.9 Gestió de Dependents:

Descripció: Els administradors han de poder gestionar les dependents associades a cada botiga, assegurant que hi hagi prou personal per gestionar les reserves.

Detalls:

- Els administradors poden afegir, eliminar o transferir dependents d'una botiga a una altra, ajustant els permisos segons sigui necessari.
- Cada dependent està associada a una única botiga i té accés complet a la gestió de les reserves d'aquesta botiga, incloent la possibilitat de modificar-les i afegir noves.
- La informació de les dependents es manté actualitzada en temps real, assegurant que l'administrador tingui una visió clara del personal disponible en cada botiga.

3.1.10 Canvi d'idioma:

Descripció: L'aplicació ha de permetre canviar l'idioma de la interfície d'usuari de manera dinàmica, per tal que els usuaris puguin seleccionar l'idioma preferit sense necessitat de recarregar l'aplicació.

Detalls:

- L'aplicació permet als usuaris canviar entre diferents llengües disponibles (català, castellà i anglès).
- El canvi d'idioma afecta totes les pàgines i seccions de l'aplicació, garantint que l'experiència de l'usuari sigui coherent en qualsevol idioma.
- Un cop l'usuari selecciona un idioma, aquest es manté actiu mentre l'usuari utilitza l'aplicació, sense necessitat de tornar a seleccionar-lo cada vegada.
- L'aplicació està preparada per afegir nous idiomes en el futur.
- Si alguna part de l'aplicació no està disponible en l'idioma seleccionat, es mostrarà en l'idioma predeterminat.

3.1.11 Categoria Productes

Descripció: L'aplicació ha de permetre gestionar les categories de productes, assignant categories específiques a cada producte per facilitar la seva organització i cerca en el sistema.

Detalls:

- Els administradors poden afegir noves categories de productes, introduint el nom de la categoria i una descripció opcional.
- Les categories es poden editar per modificar el nom o la descripció segons sigui necessari.
- Cada producte pot ser assignat a una o més categories, permetent una classificació més precisa.

- Els usuaris poden filtrar productes per categories en les interfícies de consulta i reserva, facilitant la selecció de productes específics segons les preferències dels clients.
- Les categories es mostren en les reserves, assegurant que els usuaris tenen una visió clara dels tipus de productes disponibles.
- Els canvis en les categories es reflecteixen immediatament en totes les reserves i productes associats.

3.1.12 Gestió Estadístiques

Descripció: Els administradors han de tenir accés a una secció de estadístiques per a visualitzar dades rellevants sobre les reserves, dependents, botigues i clients, facilitant la presa de decisions.

Detalls:

- La secció d'estadístiques ofereix diverses visualitzacions gràfiques (gràfics de barres i de sectors) per mostrar les dades de manera clara i accessible.
- Els administradors poden visualitzar estadístiques de les cinc categories més destacades:
 - **Top 5 Productes Més Reservats:** Mostra els productes amb el nombre més alt de reserves.
 - **Top 5 Dependents Actives:** Mostra les dependents que han gestionat el major nombre de reserves.
 - **Top 5 Botigues Actives:** Mostra les botigues amb més volum de reserves.
 - **Top 5 Clients Actius:** Mostra els clients que han fet més reserves.
- Els gràfics s'actualitzen en temps real per reflectir els canvis a les dades de reserves, clients i productes, assegurant que els administradors disposen de dades actualitzades.
- Els administradors poden seleccionar o deseleccionar categories per personalitzar les visualitzacions i centrar-se en la informació més rellevant en cada moment.

4 Tecnologies Utilitzades

En aquest apartat es descriuran les tecnologies utilitzades per al desenvolupament del projecte com podreu veure a la *figura 5*, tant a nivell de front-end com de back-end i base de dades. Cada tecnologia s'ha seleccionat en funció de les necessitats del projecte, tenint en compte aspectes com la facilitat d'ús, el rendiment, l'escalabilitat i la integració amb altres eines. A continuació es detallaran cadascuna d'aquestes tecnologies, explicant-ne les característiques principals, així com els avantatges i els inconvenients que comporta la seva utilització.

4.0.1 Visual Studio Code (VSCode)

Visual Studio Code és l'entorn de desenvolupament integrat (IDE) que s'ha utilitzat per escriure i depurar el codi del projecte [3].

Avantatges:

- Ofereix una gran quantitat d'extensions que milloren la productivitat i la integració amb les tecnologies utilitzades en el projecte.
- És lleuger, personalitzable i ideal per a projectes de desenvolupament web i mòbil.

Desavantatges:

- Pot consumir molts recursos si es carreguen moltes extensions simultàniament.

4.1 Front-End

4.1.1 Ionic

Ionic és un *framework* d'interfície d'usuari de codi obert orientat al desenvolupament d'aplicacions híbrides per a plataformes mòbils i web. Està construït sobre tecnologies web com HTML, CSS i JavaScript, i utilitza Angular, React o Vue per a la part lògica. En aquest projecte, s'ha utilitzat Ionic amb MUI (Material-UI), que proporciona components estilitzats basats en el disseny de Google Material Design [1].

Avantatges:

- **Multiplataforma:** Permet el desenvolupament d'aplicacions per a Android, iOS i web amb un sol codi, reduint el temps i els recursos necessaris per mantenir múltiples plataformes.
- **Integració amb *frameworks* moderns:** És compatible amb Angular, React i Vue, facilitant la integració amb altres tecnologies modernes.
- **Gran comunitat i ecosistema de *plugins*:** Disposa d'una gran comunitat de desenvolupadors i molts *plugins* per integrar funcionalitats natives dels dispositius mòbils, com la càmera o el GPS.

- **UI nativa:** Ofereix una experiència d'usuari nativa, amb components que imiten l'aspecte de les aplicacions natives de cada plataforma.

Desavantatges:

- **Rendiment:** Les aplicacions natives pures solen oferir millor rendiment en comparació amb les aplicacions híbrides.
- **Dependència del WebView:** Les aplicacions d'Ionic funcionen dins d'un WebView, cosa que pot ocasionar problemes de rendiment en dispositius més antics.

4.1.2 MUI (Material-UI)

Material-UI (MUI) és una llibreria de components per a React que implementa les pautes de disseny *Material Design* de Google. Proporciona una sèrie de components preestilitzats que faciliten la creació d'interfícies d'usuari consistents i visualment atractives [2].

Avantatges:

- **Disseny coherent:** Proporciona un conjunt de components estilitzats segons el sistema Material Design, garantint una experiència d'usuari visualment coherent i moderna.
- **Alta personalització:** Els components són altament personalitzables per adaptar-se a les necessitats específiques de cada projecte.
- **Documentació completa:** Disposa d'una documentació molt completa, cosa que facilita l'aprenentatge i implementació.

Desavantatges:

- **Curva d'aprenentatge:** Pot ser difícil d'aprendre per desenvolupadors que no estiguin familiaritzats amb React o Material Design.
- **Pes de la llibreria:** Pot afegir pes al paquet d'aplicació, cosa que afecta el rendiment.

4.1.3 CSS

CSS (Cascading Style Sheets) és el llenguatge utilitzat per descriure l'aparença visual d'una pàgina web, especificant com els elements HTML es mostren en pantalla. És un component fonamental en el disseny de la interfície d'usuari de qualsevol aplicació web [28].

Avantatges:

- **Separació de contingut i disseny:** Permet separar completament l'estil del contingut, facilitant així la gestió i actualització del disseny sense afectar la lògica de l'aplicació.
- **Estils reutilitzables:** Mitjançant classes i regles, es poden aplicar estils consistents a tota l'aplicació, reduint el codi duplicat.
- **Responsive Design:** CSS facilita la creació de dissenys adaptatius, que s'ajusten automàticament a la mida de la pantalla del dispositiu, cosa essencial per a aplicacions mòbils i web.
- **Compatibilitat amb totes les plataformes:** CSS és compatible amb tots els navegadors moderns, assegurant així una experiència consistent per a tots els usuaris.

Desavantatges:

- **Inconsistències entre navegadors:** Tot i la seva compatibilitat, certes propietats CSS poden comportar diferències d'implementació segons el navegador, cosa que requereix proves addicionals.
- **Complexitat en projectes grans:** Gestionar fulls d'estil en aplicacions grans pot ser complicat sense una estructura adequada, fet que pot derivar en estils solapats o conflictes.

4.1.4 React

React és una llibreria de JavaScript de codi obert dissenyada per construir interfícies d'usuari. Està mantinguda per Facebook i una comunitat de desenvolupadors independents. React permet desenvolupar components reutilitzables que gestionen la interfície de manera eficient [14].

Avantatges:

- **Componentització:** React està basat en components, cosa que permet modularitzar el codi i fer-lo més reutilitzable i mantenible.
- **DOM virtual:** Utilitza un DOM virtual per optimitzar les actualitzacions a la pàgina web, millorant el rendiment.
- **Gran ecosistema:** La llibreria té una gran comunitat i nombroses extensions i eines que faciliten el desenvolupament.
- **Unidireccionalitat de dades:** Les dades flueixen en una sola direcció, facilitant la gestió de l'estat de l'aplicació.

Desavantatges:

- **Curva d'aprenentatge:** Tot i ser potent, React pot requerir un cert temps d'aprenentatge per desenvolupar aplicacions complexes.
- **Configuració inicial:** React només proporciona la capa de vista, cosa que requereix la integració de llibreries addicionals per a una aplicació completa.

4.1.5 React Router

React Router és una llibreria de rutes per a aplicacions desenvolupades amb React que permet la navegació dinàmica i basada en components. Facilita la implementació de rutes i la gestió d'estats i paràmetres de navegació [35].

Avantatges:

- **Navegació dinàmica:** Permet canviar de pàgina sense recarregar tota l'aplicació.
- **Rutes dinàmiques:** Permet passar paràmetres i crear rutes dinàmiques en funció de l'estat de l'aplicació o l'usuari.
- **Modularitat:** Integra la navegació dins dels components, cosa que millora la modularitat del codi.

Desavantatges:

- **Complexitat en aplicacions grans:** En projectes més grans amb moltes rutes, la gestió pot esdevenir més complexa.

4.1.6 React Calendar

React Calendar és una llibreria que proporciona components de calendari per a React. En el projecte, s'utilitza per gestionar la selecció de dates en la funcionalitat de filtres [5].

Avantatges:

- **Fàcil integració:** Proporciona una interfície fàcil d'usar per mostrar i seleccionar dates.
- **Estil personalitzable:** És altament personalitzable gràcies a les opcions d'estil que proporciona, incloent-hi compatibilitat amb CSS.

Desavantatges:

- **Funcionalitats limitades:** Pot no ser suficient per a casos d'ús més complexos de calendari, que podrien requerir llibreries més completes.

4.1.7 React i18next

React i18next és una extensió de la llibreria i18next adaptada per a aplicacions desenvolupades amb React. En aquest projecte, s'ha utilitzat per implementar la funcionalitat de canvi d'idioma de manera dinàmica en tota l'aplicació. Això permet als usuaris seleccionar l'idioma de la interfície a través d'un selector, i tots els textos es tradueixen automàticament sense necessitat de recarregar la pàgina [24].

Avantatges:

- **Facilitat d'ús:** Proporciona una integració senzilla amb React, permetent l'ús de traduccions dins dels components de manera intuïtiva amb el hook `useTranslation`.
- **Modularitat:** Permet estructurar les traduccions en fitxers separats per a cada idioma, facilitant l'escalabilitat de l'aplicació i l'afegiment de nous idiomes en el futur.
- **Canvi d'idioma dinàmic:** Els usuaris poden canviar d'idioma sense necessitat de recarregar l'aplicació, mantenint una experiència d'usuari fluida i sense interrupcions.

Desavantatges:

- **Dependència de fitxers de traducció:** Cal gestionar adequadament els fitxers de traducció per evitar inconsistències o textos no traduïts.
- **Curva d'aprenentatge:** Encara que senzill, pot requerir un cert temps d'aprenentatge per familiaritzar-se amb la integració d'i18next i React i18next.

4.2 Back-End

4.2.1 FastAPI

FastAPI és un *framework* de backend per a Python, dissenyat per construir APIs de manera ràpida i eficient. Utilitza anotacions de tipus per a la validació automàtica de dades i genera documentació automàtica d'API mitjançant OpenAPI i JSON Schema [15].

Avantatges:

- **Rendiment:** FastAPI és conegut pel seu alt rendiment gràcies a l'ús d'asincio, que el fa molt adequat per a aplicacions amb moltes sol·licituds concurrents.
- **Tipat estàtic:** L'ús d'anotacions de tipus redueix errors i millora la qualitat del codi.
- **Documentació automàtica:** Genera automàticament documentació d'API utilitzant OpenAPI i Swagger UI.

Desavantatges:

- **Mida de la comunitat:** Tot i que està creixent, la comunitat de FastAPI és encara petita en comparació amb altres *frameworks* com Django.
- **Programació asíncrona:** Pot requerir una corba d'aprenentatge més pronunciada per a desenvolupadors que no estiguin familiaritzats amb la programació asíncrona.

4.2.2 Bcrypt

Bcrypt és una biblioteca de hash de contrasenyes que s'utilitza per encriptar les contrasenyes dels usuaris abans d'emmagatzemar-les a la base de dades, assegurant així la seguretat de les dades sensibles [4].

Avantatges:

- Proporciona un mètode segur i fiable per encriptar contrasenyes.
- És àmpliament reconegut i utilitzat en la indústria per la seva robustesa contra atacs de força bruta.

Desavantatges:

- El procés de hashing pot afegir certa càrrega de processament, tot i que és mínim en la majoria dels casos.

4.2.3 Fetch API

`fetch()` és una API nativa de JavaScript que s'utilitza per fer sol·licituds HTTP. En el codi, es fa servir per obtenir les dades de les reserves des d'un backend basat en FastAPI [13].

Avantatges:

- **Senzillesa:** És una manera fàcil i nativa de fer peticions HTTP sense necessitar llibreries addicionals.
- **Suport per promeses:** El fet que utilitzi promeses permet una gestió més eficient de les operacions asíncrones.

Desavantatges:

- **Compatibilitat amb navegadors antics:** Alguns navegadors antics poden no suportar l'API `fetch()` de manera nativa, tot i que això es pot solucionar amb polífills.

4.3 Postman

Postman és una eina utilitzada per al desenvolupament i la comprovació d'API, que em permet simular i realitzar peticions HTTP cap als endpoints del backend de l'aplicació. Aquesta eina m'ha estat de gran utilitat per verificar la funcionalitat i el comportament dels diferents endpoints en diferents condicions, permetent-me validar que els resultats obtinguts corresponen amb els esperats per a cada operació.

Principals funcionalitats utilitzades:

- He pogut enviar peticions GET, POST, PUT i DELETE, simulant les diferents operacions que es realitzen a l'aplicació, com la creació, actualització i eliminació de dades.
- Mitjançant Postman, he tingut la capacitat d'afegir headers i bodies personalitzats en les peticions, cosa que m'ha permès replicar amb precisió les peticions que l'aplicació envia al backend.
- L'eina també ofereix una vista detallada de les respostes, on puc observar el codi d'estat HTTP, el cos de la resposta i les capçaleres. Això ha estat especialment útil per detectar errors i ajustar el codi backend per assegurar una comunicació correcta amb el frontend.

4.4 Base de Dades

4.4.1 Firebase

Firebase és una plataforma de desenvolupament d'aplicacions mòbils i web creada per Google. Proporciona una base de dades en temps real, autenticació, hosting, notificaciones *push* i altres serveis que faciliten el desenvolupament d'aplicacions [26].

Avantatges:

- **Escalabilitat:** Està dissenyat per escalar fàcilment, ideal per a aplicacions amb un creixement ràpid.
- **Funcionalitats integrades:** Ofereix serveis com l'autenticació d'usuaris, hosting, analítiques i notificaciones *push*.
- **Actualització en temps real:** Permet la sincronització de dades en temps real, millorant l'experiència d'usuari.

Desavantatges:

- **Dependència de Google:** L'aplicació depèn completament de l'ecosistema de Google.
- **Limitacions en les consultes:** Firebase Realtime Database té limitacions en consultes complexes, ja que és una base de dades NoSQL.

4.4.2 Firebase Admin SDK

Firebase Admin SDK és una biblioteca que permet interactuar amb Firebase des del servidor. Al projecte, s'utilitza per gestionar la connexió amb la base de dades Firebase Firestore, així com per realitzar operacions d'autenticació i gestió de dades [27].

Avantatges:

- Permet realitzar operacions administratives com l'autenticació d'usuaris i la gestió de la base de dades des del servidor.
- Proporciona una integració senzilla amb les altres funcionalitats de Firebase, com Firestore i la gestió de fitxers.

Desavantatges:

- Dependència en l'ecosistema de Firebase i Google.
- Pot requerir un coneixement més avançat sobre Firebase i el seu funcionament.

4.4.3 Firebase Firestore

Firestore és la base de dades NoSQL de Firebase que s'utilitza en aquest projecte per emmagatzemar i sincronitzar les dades en temps real. Proporciona una arquitectura escalable que és ideal per a aplicacions mòbils i web [25].

Avantatges:

- Actualització de dades en temps real i sincronització automàtica amb els dispositius connectats.
- Escalabilitat i rendiment elevat per a aplicacions amb un gran volum de dades.

Desavantatges:

- Limitacions en les consultes complexes, ja que és un sistema NoSQL.
- Els costos poden augmentar ràpidament a mesura que es realitzen més operacions de lectura i s'utilitza més espai d'emmagatzematge.

4.4.4 Google Cloud Storage

Google Cloud Storage és un servei d'emmagatzematge d'arxius al núvol, dissenyat per a emmagatzematge segur, escalable i d'alt rendiment [17]. En aquest projecte, s'ha utilitzat per gestionar les imatges de l'aplicació, emmagatzemant-les en un *bucket* i accedint-hi mitjançant URLs firmades [23].

Avantatges:

- **Escalabilitat:** Permet emmagatzemar una gran quantitat d'arxius sense preocupar-se pel límit de capacitat [20].
- **Integració fàcil:** Ofereix una API robusta per integrar-se amb aplicacions, com el backend d'aquest projecte [16].
- **Seguretat:** Les URLs firmades asseguren que només els usuaris autoritzats puguin accedir als arxius [21].
- **Rendiment global:** Té punts de presència global que garanteixen temps de resposta ràpids independentment de la ubicació de l'usuari [18].

Desavantatges:

- **Cost:** Pot ser costós si es gestionen grans volums de dades o es fan moltes sol·licituds a les imatges [19].
- **Complexitat inicial:** La configuració de permisos i generació de credencials pot requerir temps si no es coneixen les eines [22].

4.5 Deploy

4.5.1 Docker Compose

Docker Compose és una eina que permet definir i gestionar aplicacions multi-*container* en un fitxer YAML [9]. En aquest projecte, s'ha utilitzat durant el desenvolupament per crear un entorn consistent que inclou el backend i altres serveis necessaris.

Avantatges:

- **Consistència:** Garanteix que l'entorn de desenvolupament sigui idèntic al de producció [6].
- **Facilitat d'ús:** Amb un únic fitxer `docker-compose.yml`, es poden definir i executar múltiples serveis [8].
- **Portabilitat:** Permet executar l'aplicació en qualsevol màquina amb Docker instal·lat [10].
- **Escalabilitat:** Facilita l'execució de múltiples instàncies de serveis per a proves de càrrega [12].

Desavantatges:

- **Curva d'aprenentatge:** Pot ser complex per a desenvolupadors sense experiència prèvia amb Docker [7].
- **Requisits de recursos:** Pot consumir molts recursos del sistema, especialment en màquines amb especificacions limitades [11].

4.5.2 Render

Render és una plataforma de desplegament que permet publicar aplicacions de backend, frontend i serveis estàtics amb configuracions senzilles [30]. En aquest projecte, s'ha utilitzat per desplegar el backend (com a *web service*) i el frontend (com a *static site*), garantint la integració completa del sistema.

Avantatges:

- **Facilitat d'ús:** Ofereix una interfície intuïtiva i configuració mínima per desplegar aplicacions [32].
- **Integració contínua:** Permet la integració amb GitHub per desplegar automàticament els canvis [31].
- **Escalabilitat automàtica:** Render ajusta els recursos automàticament en funció del tràfic [29].
- **Gratuït per projectes petits:** Ofereix un pla gratuït suficient per a desenvolupament i proves [34].

Desavantatges:

- **Limitacions del pla gratuït:** Els recursos del pla gratuït poden ser insuficients per a aplicacions amb alt tràfic [34].
- **Temps d'inactivitat en despertador:** En el pla gratuït, les aplicacions poden entrar en *sleep mode*, generant temps d'espera en reactivar-se [33].

4.6 Diagrama de Technologies

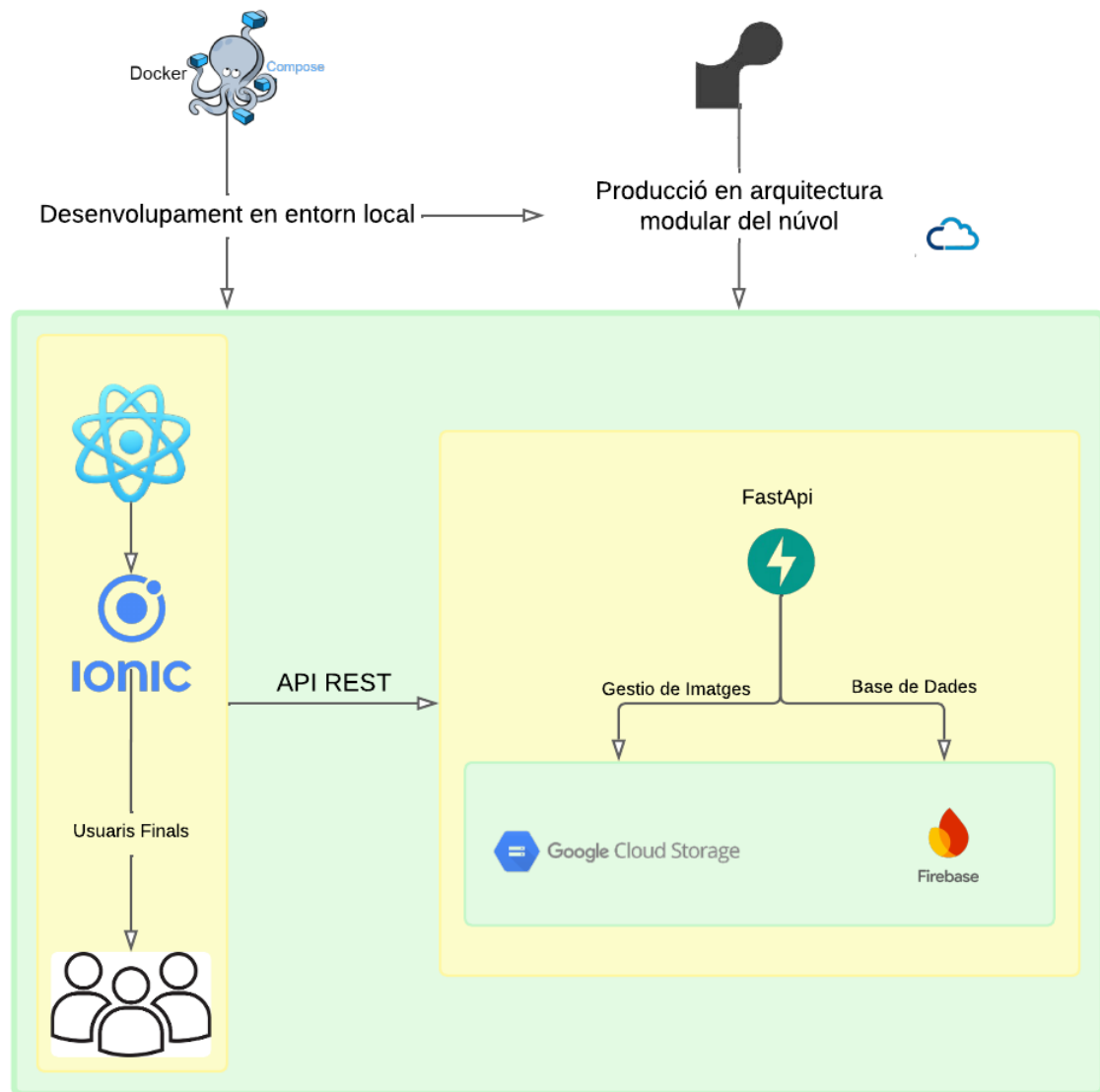


Figura 5: Diagrama de Technologies

5 Model de dades ER

5.1 Distribució del Model de Base de Dades:

En el context de l'aplicació mòbil de gestió de reserves entre botigues i un obrador centralitzat, he implementat un model de base de dades que permet emmagatzemar i gestionar eficientment la informació relacionada amb les botigues, les dependents, els productes, els clients i les reserves. La base de dades NoSQL de **Firebase Firestore** s'ha utilitzat per oferir una sincronització automàtica i en temps real entre els diferents punts de servei i dispositius connectats a l'aplicació.

5.2 Entitats Principals:

5.2.1 BOTIGUES:

Aquesta entitat emmagatzema la informació de les botigues registrades en el sistema. Cada botiga està associada a un conjunt de dependents i gestiona les seves pròpies reserves.

Atributs:

- **ID de Botiga:** Identificador únic per a cada botiga.
- **Nom de la Botiga:** Nom de la botiga.
- **Número de Telèfon:** Telèfon de contacte de la botiga.
- **Llista de Dependents:** Llista d'IDs de dependents associades a la botiga.
- **Llista de Reserves:** Llista d'IDs de reserves realitzades a la botiga.
- **Contrasenya:** Hash de la contrasenya d'accés per a la botiga.

5.2.2 DEPENDENTES:

Les dependents són empleades que treballen a les botigues i són responsables de gestionar les reserves dels clients.

Atributs:

- **ID de Dependenta:** Identificador únic per a cada dependenta.
- **Nom de la Dependenta:** Nom complet de la dependenta.
- **ID de Botiga:** Identificador de la botiga on treballa la dependenta.

5.2.3 CLIENTS:

Els clients són els usuaris que realitzen reserves a través de les botigues.

Atributs:

- **ID de Client:** Identificador únic per a cada client.
- **Nom del Client:** Nom complet del client.
- **Número de Telèfon:** Telèfon de contacte del client.

5.2.4 PRODUCTES:

Aquesta entitat emmagatzema la informació dels productes disponibles per a la reserva, incloent el nom ,el pes i el tipus de producte (categoria).

Atributs:

- **ID de Producte:** Identificador únic per a cada producte.
- **Nom del Producte:** Nom descriptiu del producte.
- **Pes:** Pes del producte.
- **Categoria:** Categoria del producte (Bollería, Pastissería, Fleca)

5.2.5 PRODUCTES_RESERVA:

Aquesta entitat defineix els productes que formen part d'una reserva, amb informació detallada sobre la quantitat, el pes i la descripció.

Atributs:

- **ID de ProducteReserva:** Identificador únic per a cada producte reservat.
- **ID de Producte:** Identificador del producte reservat.
- **ID de Reserva:** Identificador de la reserva a la qual pertany el producte.
- **Quantitat:** Quantitat reservada del producte.
- **Descripció:** Descripció addicional del producte en la reserva.
- **Pes:** Pes del producte reservat.

5.2.6 RESERVES:

Aquesta entitat conté tota la informació sobre les reserves realitzades pels clients, incloent els productes reservats, la data i l'estat de la reserva.

Atributs:

- **ID de Reserva:** Identificador únic per a cada reserva.
- **ID de Client:** Identificador del client que va realitzar la reserva.
- **ID de Dependenta:** Identificador de la dependenta associada a la reserva.
- **Data i Hora:** Data i hora en què es realitzarà la reserva.
- **Estat de la Reserva:** Estat actual de la reserva (pendent, confirmada, cancel·lada, etc.).
- **Llista de Productes Reservats:** Llista d'IDs dels productes inclosos en la reserva.

5.2.7 OBRADOR:

Aquesta entitat emmagatzema la informació de l'Obrador registrat en el sistema, el qual és responsable de la producció dels productes que seran distribuïts a les botigues.

Atributs:

- **ID de l'Obrador:** Identificador únic per a l'Obrador.
- **Nom de l'Obrador:** Nom oficial de l'Obrador.
- **Número de Telèfon:** Telèfon de contacte de l'Obrador per a comunicació amb administradors i botigues.
- **Contrasenya:** La contrasenya d'accés per a l'Obrador.
- **Productes Disponibles:** Llista dels productes que l'Obrador produeix i subministra a les botigues.

5.3 Relacions:

- **Botigues - Dependentes:** Relació un a molts, on una botiga pot tenir múltiples dependents.
- **Botigues - Reserves:** Relació un a molts, on una botiga pot tenir múltiples reserves associades.
- **Dependents - Reserves:** Relació un a molts, on una dependenta pot estar associada a múltiples reserves.
- **Clients - Reserves:** Relació un a molts, on un client pot realitzar múltiples reserves.
- **Productes - Productes Reserva:** Relació un a molts, on un producte pot formar part de múltiples productes reserva.

- **Reserves - Productes_Reserva:** Relació un a molts, on una reserva pot contenir múltiples productes reserva.

Aquest model de dades ER es proporciona una estructura clara i organitzada que facilita la gestió de les entitats principals del sistema i les seves interaccions. El model està optimitzat per treballar en un entorn de base de dades NoSQL, permetent un rendiment eficient i una fàcil escalabilitat.

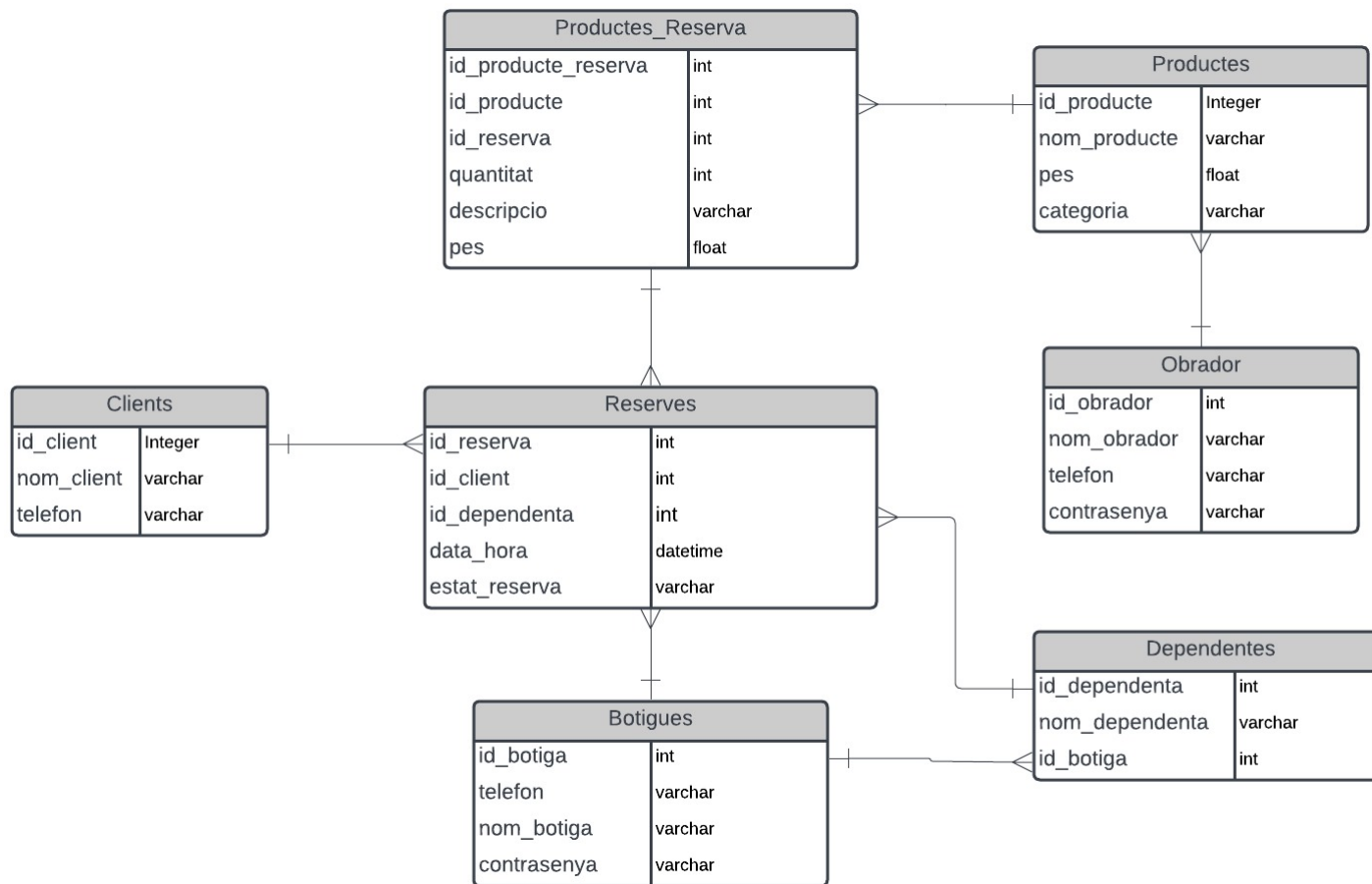


Figura 6: Diagrama Base de Dades

6 Front-end

6.1 Explicació global de la vista

He desenvolupat un conjunt de pantalles i funcionalitats per gestionar la meua aplicació d'administració de reserves i productes dins d'un sistema basat en Ionic i React. Aquestes pantalles em permeten gestionar productes, reserves i botigues de manera eficient i efectiva, incloent funcions per afegir, eliminar, modificar i visualitzar informació de les entitats que administro. He estructurat l'aplicació de tal manera que cadascuna de les pantalles compleix un rol específic, i totes estan interconnectades per proporcionar una experiència d'usuari coherent i integral.

El codi està compost per diverses parts que cobreixen funcionalitats clau de la gestió de productes, reserves i botigues. Per exemple, la pantalla `ProductosAdmin` permet afegir i eliminar productes, mentre que la pantalla `PantallaObrador` és un punt central per visualitzar i gestionar les reserves en funció de diferents filtres. Altres pantalles, com `Reserva`, permeten veure els detalls d'una reserva específica i modificar-la si cal, mentre que `ReservasAdmin` permet a un administrador veure i gestionar totes les reserves d'una botiga. A més, la pantalla `TiendasAdmin` m'ofereix la possibilitat de gestionar botigues i dependents, i la pantalla `verDetalles` serveix per visualitzar de manera més detallada la informació d'una reserva seleccionada. Aquestes pantalles treballen de forma conjunta per formar un sistema integral i fàcil d'utilitzar per gestionar l'activitat de les botigues, productes i reserves.

6.2 Explicació individual de les pantalles

6.2.1 Administrador Gestió dels productes

Aquesta pantalla és l'encarregada de gestionar els productes que es venen a les botigues. He creat un espai on puc visualitzar tots els productes que hi ha disponibles i afegir-ne de nous, o bé eliminar-ne els existents. Quan obro la pantalla, faig una petició a l'API per obtenir la llista actualitzada de productes. A més, he incorporat una funcionalitat per cercar productes a la barra de cerca, que m'ajuda a trobar els productes més fàcilment. Els diàlegs modals que he creat em permeten afegir i eliminar productes de manera segura, confirmant l'acció abans de procedir, i això assegura que no es facin canvis accidentalment. Aquesta pantalla és clau per a la gestió de l'inventari de productes.

6.2.2 Obrador Gestió de les Reserves

En aquesta pantalla, he creat una interfície per a la gestió de les reserves que es fan a l'obrador. La informació de les reserves es mostra en llista, i es pot filtrar per data i per botiga. He integrat un calendari que permet seleccionar la data de les reserves que vull veure, i un filtre que em permet triar la botiga que m'interessa. Així mateix, la funcionalitat de cerca em facilita trobar les reserves d'un client específic. Aquesta pantalla és essencial per mantenir l'organització de les reserves diàries i per garantir que el procés de producció sigui eficient.

6.2.3 Reserva

He creat aquesta pantalla per veure els detalls d'una reserva específica. Aquí, se'm permet veure la informació del client, la dependent assignada, l'estat de la reserva, els productes reservats i altres dades rellevants. A més, puc activar el mode d'edició per modificar els detalls de la reserva si fos necessari. Això és útil si cal fer ajustos de darrera hora o si el client vol canviar algun aspecte de la seva comanda. Aquesta pantalla és crucial per assegurar que cada reserva es processi correctament i que els detalls estiguin sempre actualitzats.

6.2.4 Administrador Gestió de les Reserves

Aquesta pantalla em permet gestionar totes les reserves d'una botiga de manera centralitzada. Aquí puc veure una llista de totes les reserves realitzades, amb la possibilitat de cercar una reserva específica a través del camp de cerca. També he implementat un calendari i un sistema de filtratge per botiga, de manera que és fàcil trobar les reserves realitzades en un període de temps o botiga concreta. Això fa que la gestió de les reserves sigui més eficient i m'ajuda a mantenir un control precís de les comandes que han de ser processades.

6.2.5 Interfície de les botigues

La pantalla Tab1 serveix com una interfície d'accés a les reserves fetes en una botiga específica. A través d'ella, puc veure totes les reserves i filtrar-les per data o per

client, cosa que em facilita molt la cerca de la informació que necessito. A més, he inclòs la funcionalitat per veure els detalls de cadascuna de les reserves, amb la possibilitat de navegar a una pantalla de detall on es pot veure la informació completa de la reserva. Aquesta pantalla fa que el procés de gestió de reserves sigui molt més fluid i intuïtiu.

6.2.6 Administració Gestió de les Botigues

He desenvolupat la pantalla TiendasAdmin per gestionar les botigues dins del sistema. Aquí puc afegir, eliminar i modificar la informació de les botigues, així com canviar les seves contrasenyes. He inclòs una funcionalitat que em permet veure la llista de dependents de cada botiga, de manera que és més fàcil mantenir un registre de qui està treballant a cada local. Aquesta pantalla és fonamental per a l'administració global de les botigues i garanteix que la informació estigui sempre actualitzada i sigui fàcilment accessible.

6.2.7 Ver Detalles

La pantalla verDetalles és una de les més importants per a la gestió detallada d'una reserva. Aquí puc veure tota la informació relacionada amb una reserva específica, com ara el nom del client, els productes que ha demanat, el pes, la quantitat i altres dades importants. Aquesta pantalla proporciona una visió completa de la reserva i assegura que no es passi per alt cap detall important. També he afegit un botó per tornar enrere, cosa que facilita la navegació per l'aplicació i millora l'experiència d'usuari.

En conjunt, aquestes pantalles m'han permès crear un sistema d'administració de reserves i productes complet i eficient, que cobreix totes les necessitats de gestió que sorgeixen en el context d'una botiga o obrador. El codi està dissenyat per ser flexible i fàcil d'utilitzar, i l'ús de tecnologies com React, Ionic i Material-UI han estat fonamentals per aconseguir una experiència d'usuari atractiva i funcional.

6.3 Internacionalització i Canvi de Llenguatge

Per tal de millorar l'experiència d'usuari i adaptar l'aplicació a un públic més ampli, he implementat un sistema d'internacionalització (i18n) al front-end. Aquest sistema permet als usuaris canviar l'idioma de la interfície de manera dinàmica, sense necessitat de recarregar la pàgina, oferint una experiència fluida i personalitzada.

He estructurat aquesta funcionalitat a través de fitxers de traducció que contenen les cadenes de text en els idiomes disponibles: català, espanyol i anglès. Els usuaris poden canviar l'idioma seleccionant-lo des d'un menú desplegable situat a la barra de navegació de l'aplicació. Aquesta opció de canvi d'idioma està disponible en tot moment, permetent als usuaris adaptar la interfície a les seves preferències idiomàtiques amb facilitat.

Els fitxers de traducció s'han creat per a cada idioma, i cada cadena de text es pot traduir fàcilment afegint una entrada al fitxer corresponent. Això permet que l'aplicació sigui escalable en termes d'idiomes, ja que es poden afegir nous idiomes en qualsevol moment sense haver de modificar el codi font de l'aplicació.

A més de l'internacionalització de la interfície, aquesta funcionalitat també s'ha estès a altres aspectes de l'aplicació, com ara la generació d'informes en diferents idiomes. D'aquesta manera, tant la visualització de dades com la documentació generada es poden presentar en l'idioma seleccionat, millorant l'accessibilitat i l'experiència d'usuari.

Els beneficis d'aquesta implementació inclouen una major flexibilitat per als usuaris, una millor accessibilitat per a un públic més ampli i una adaptació fàcil i ràpida a nous idiomes segons les necessitats del sistema.

6.4 Pantallas Principales

6.4.1 Admin

Dissenyada per a l'usuari administrador, aquesta pantalla permet controlar totes les botigues, dependents i reserves.

Encapçalament de l'Administrador:

- Conté el nom de l'usuari o rol juntament amb un botó de tancament de sessió per una sortida segura.

Gestió de Botigues:

- Llista de totes les botigues del sistema, amb nom, telèfon i la llista de dependents assignades.
- **Afegir Botiga:** Permet afegir una nova botiga introduint el nom, el número de telèfon i la contrasenya d'accés.
- **Eliminar Botiga:** Permet eliminar una botiga de la base de dades.
- **Canviar Contrasenya de Botiga:** Opció per canviar la contrasenya d'accés d'una botiga específica.

Gestió de Dependentes:

- Llista de totes les dependents, permetent assignar-les a una botiga o eliminar-les.
- **Afegir Dependenta:** Permet afegir una nova dependenta i assignar-la a una botiga específica.
- **Assignar Dependenta a Botiga:** Opció per canviar la botiga a la que està assignada una dependenta.

Reservas i Productes:

- **Reserves:** L'administrador pot veure un resum de les reserves actives en cada botiga, amb detalls de productes i quantitats.
- **Productes:** L'administrador pot veure i gestionar l'inventari de productes disponibles en el sistema.

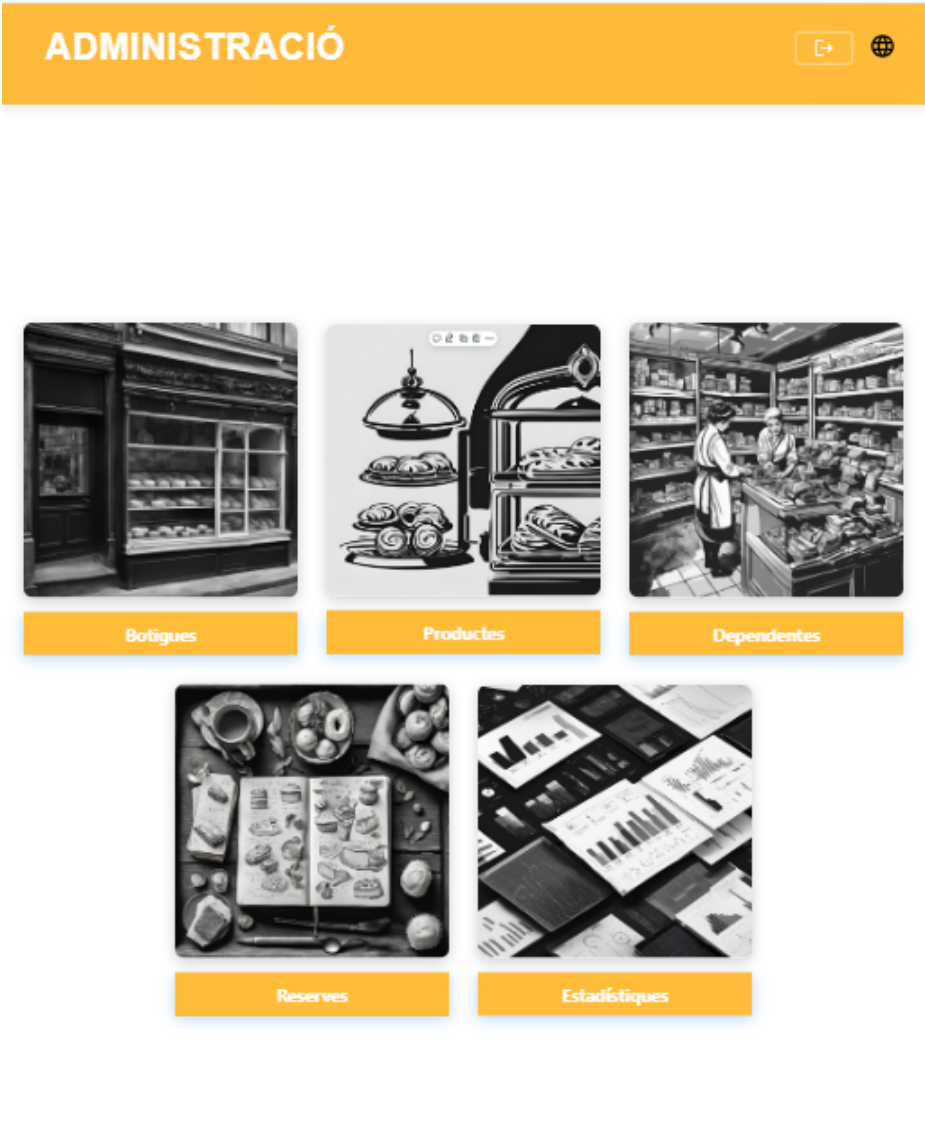


Figura 7: Pantalla del Administrador

6.4.2 Tab1

Aquesta pantalla està dissenyada per als empleats de cada botiga, on poden gestionar i consultar les reserves assignades a la botiga.

Encapçalament de la Botiga:

- Inclou el nom de la botiga (recuperat en iniciar sessió) a la part superior de la pantalla.
- Botó de tancament de sessió que redirigeix l'usuari a la pantalla de login i elimina les dades locals per seguretat.

Filtrat i Cerca de Reserves:

- Opcions de filtratge per seleccionar una data o buscar per client.
- **Filtre per Data:** Mostra un calendari. Seleccionant una data s'actualitza la llista de reserves.
- **Filtre per Categoria:** Permet filtrar per tipus de producte reservat, com ara "panaderia" o "patisserie".

Llistat de Reserves:

- Es mostren les reserves de la botiga amb informació del client, hora, dependència assignada i estat de la reserva.
- Menú de cada reserva per veure detalls o eliminar-la, permetent una gestió ràpida.
- **Detalls de la Reserva:** Mostra un resum detallat amb els productes reservats, quantitats i descripcions.
- **Eliminar Reserva:** Permet eliminar una reserva, eliminant també la informació de la base de dades.

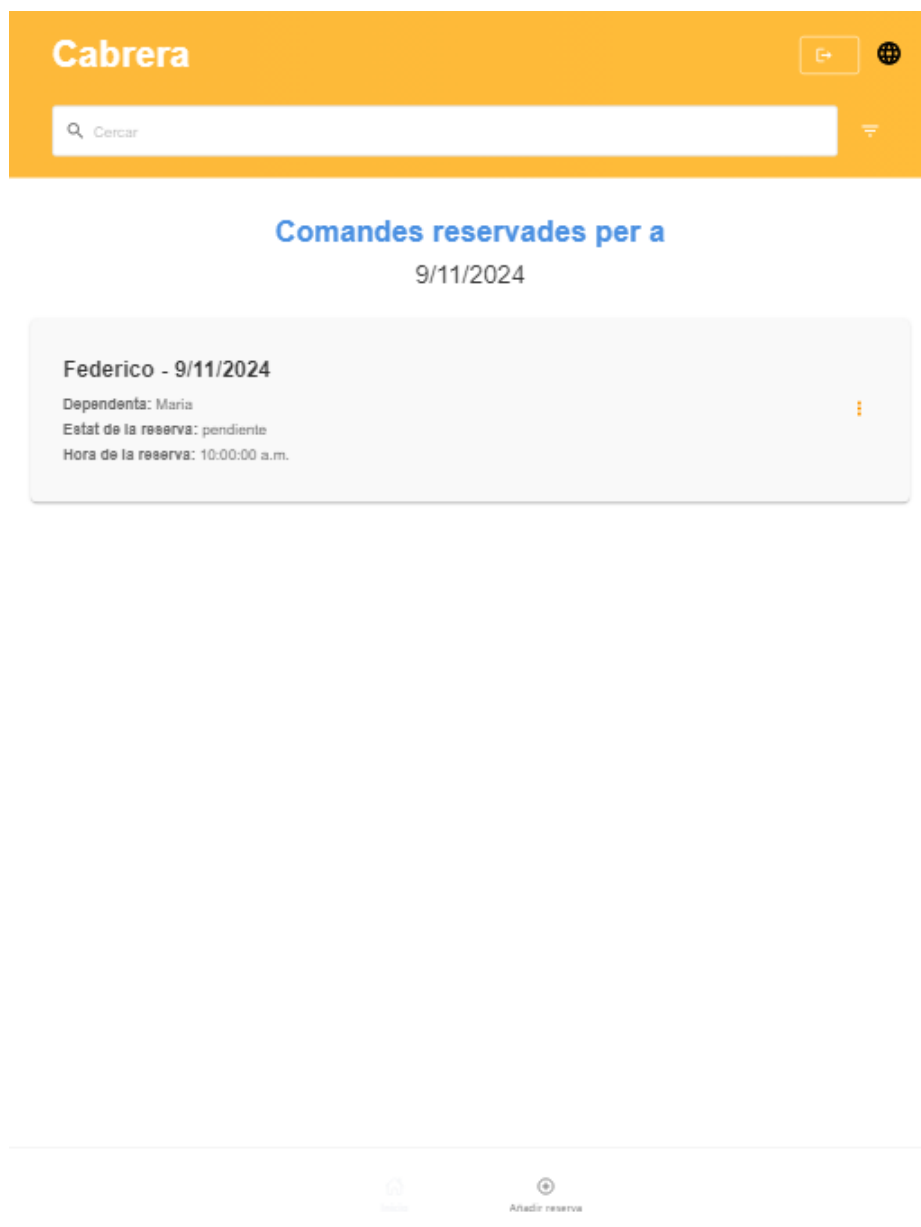


Figura 8: Pantalla del Administrador

6.4.3 Obrador

La pantalla d'Obrador gestiona la producció i l'stock de productes. Els empleats del obrador poden gestionar la demanda.

Encapçalament de l'Obrador:

- Conté el nom de l'obrador i un botó de tancament de sessió que redirigeix l'usuari a la pantalla de login.
- Selector d'idioma per adaptar-se a les preferències de l'usuari.

Secció de Productes:

- Mostra tots els productes disponibles amb el seu nom, categoria i pes per a una visualització ràpida de l'stock.
- **Afegir Producte:** Permet als empleats afegir productes nous, mantenint l'inventari actualitzat.
- **Eliminar Producte:** Cada producte té un botó per eliminar-lo del sistema.

Secció de Reserves:

- Es mostren totes les reserves actives que requereixen productes del obrador, actualitzant-se en temps real.
- **Detalls de la Reserva:** Mostra cada producte reservat i les seves quantitats, permetent una planificació precisa de la producció.

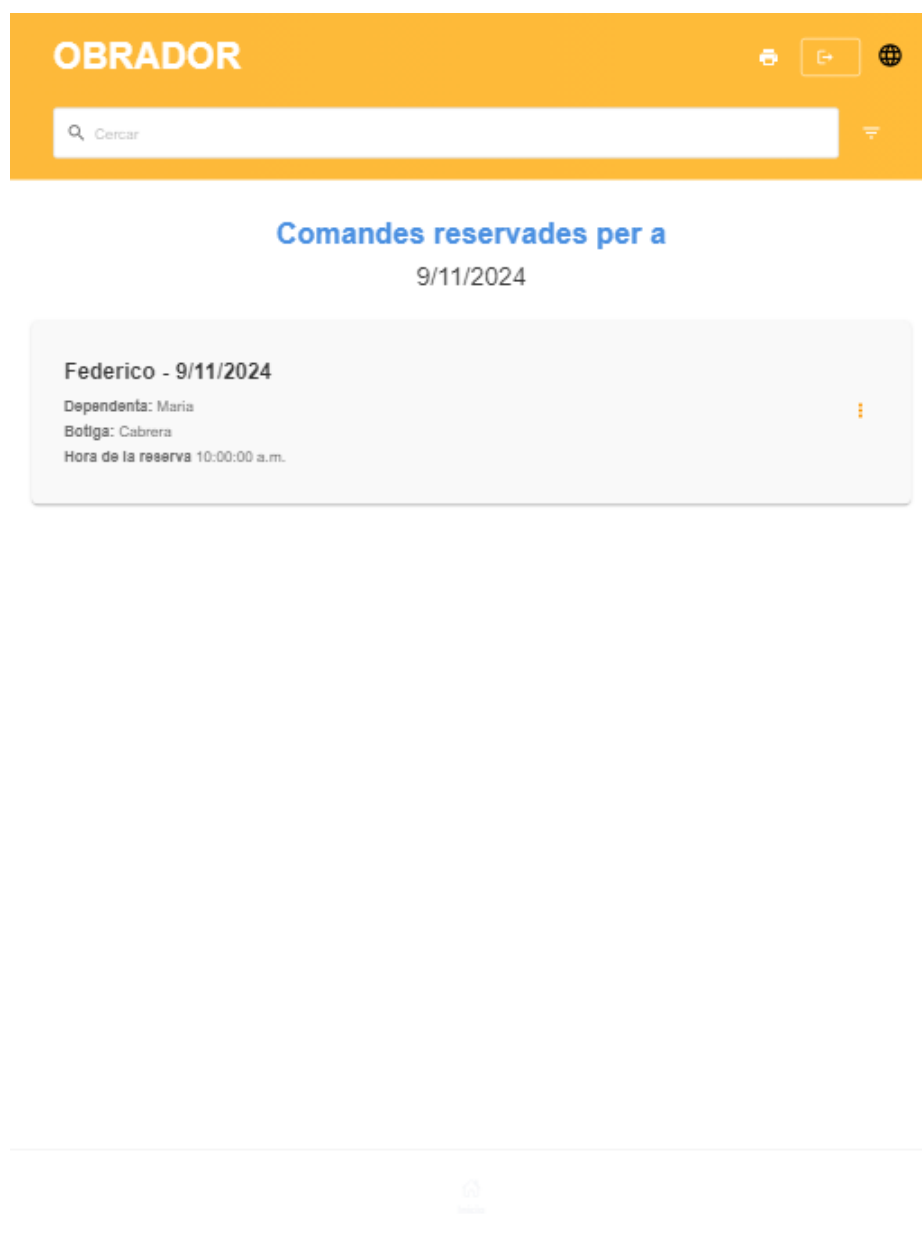


Figura 9: Pantalla del Administrador

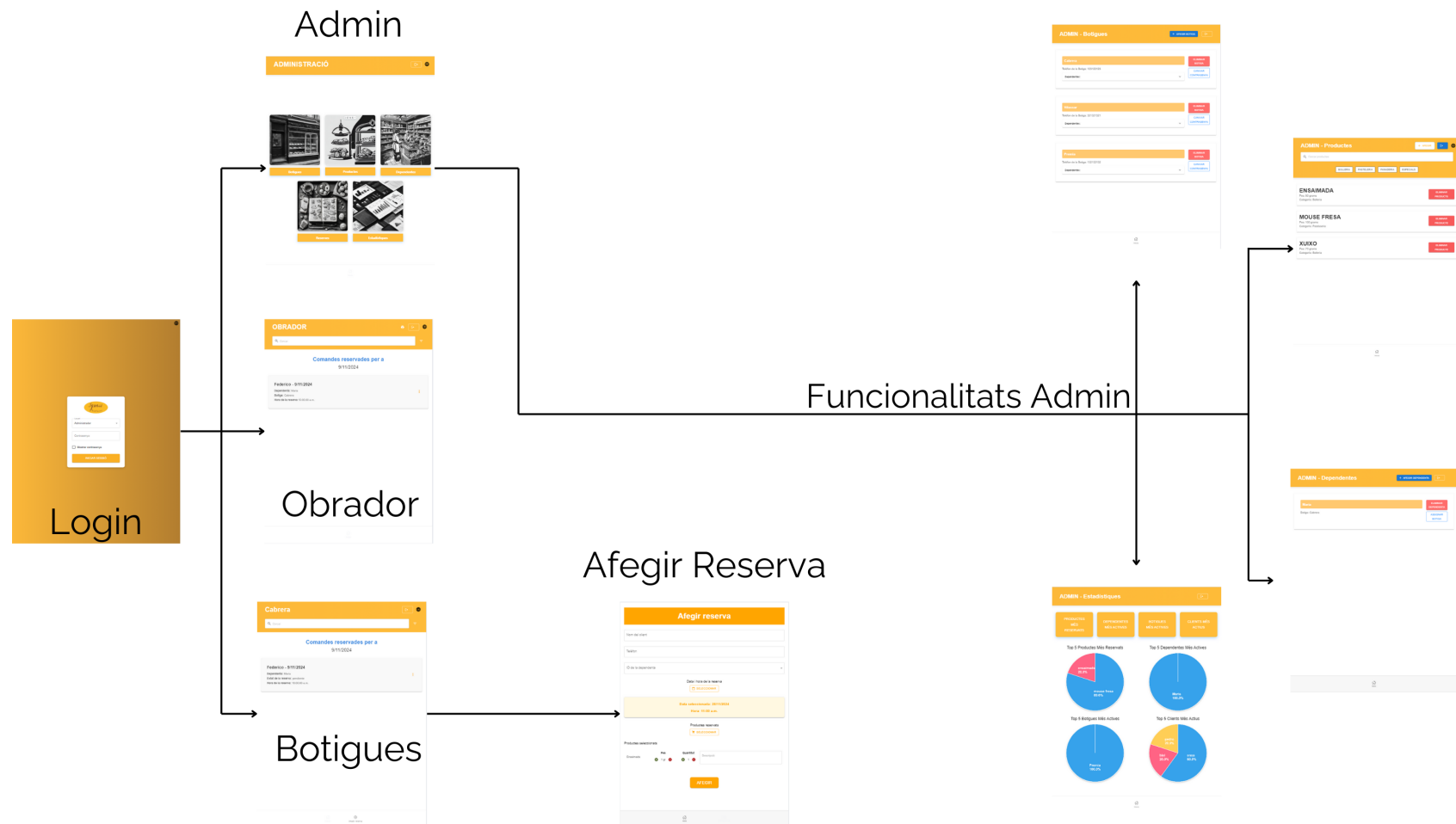


Figura 10: Diagrama de Navegació

7 Back-end

7.1 Explicació Global del Backend

He desenvolupat un backend utilitzant **FastAPI** per gestionar tota la lògica del servidor de la meua aplicació. Aquest backend és la columna vertebral de la funcionalitat de l'aplicació i permet la gestió d'entitats com les botigues, les dependents, els productes, els clients i les reserves, a més de proporcionar mecanismes d'autenticació per a l'accés segur a l'aplicació.

El sistema s'inicia amb la configuració de **Firebase** com a base de dades principal, utilitzant la seva integració amb **Firestore** per emmagatzemar i recuperar informació de manera eficient. He implementat diferents endpoints que permeten a l'aplicació interactuar amb la base de dades per gestionar les entitats esmentades anteriorment.

Les principals funcionalitats del backend inclouen la gestió d'autenticació, permetent als usuaris accedir a l'aplicació amb diferents rols (administrador, obrador o botiga); la gestió de botigues i dependents, on és possible afegir, eliminar i modificar les dades de les botigues i les seves dependents; la gestió de productes, que permet administrar els productes disponibles; la gestió de clients, que permet afegir, consultar i modificar els clients registrats, i finalment, la gestió de reserves, que és essencial per a l'operativa del negoci.

Aquest backend també incorpora un sistema de seguretat amb encriptació de contrasenyes mitjançant **bcrypt**, assegurant així que la informació sensible dels usuaris es manté protegida.

7.2 Funcionalitats implementades al backend

En el backend, m'he centrat en desenvolupar funcionalitats que cobreixin les necessitats específiques de l'aplicació, dividint-les en diversos mòduls per mantenir una estructura clara i modular:

- **Gestió d'autenticació i autorització:** He implementat sistemes d'inici de sessió i validació de credencials per a diferents tipus d'usuaris, incloent-hi administradors, dependents, clients i botigues. Aquestes funcionalitats asseguren que només els usuaris autoritzats poden accedir a la informació i realitzar operacions específiques.
- **Gestió de botigues:** Les APIs per a la gestió de botigues permeten afegir noves botigues, actualitzar-ne la informació, eliminar-les i recuperar dades relacionades, com llistes de dependents assignades o reserves associades.
- **Gestió de dependents:** He desenvolupat funcionalitats per a la creació, assignació, reassignació i eliminació de dependents. A més, aquestes APIs permeten gestionar la relació entre dependents i botigues, assegurant que aquestes es mantinguin consistents en tot moment.
- **Gestió de clients:** Les APIs associades als clients permeten crear-ne de nous, recuperar informació detallada dels clients registrats, actualitzar dades

si fos necessari i eliminar-los, garantint sempre la integritat de les relacions amb altres elements, com les reserves.

- **Gestió de reserves:** Un dels components més importants és la gestió de reserves, que inclou la creació, modificació, recuperació i eliminació de reserves. Aquest sistema assegura que les reserves estiguin correctament associades a clients, botigues i dependents, oferint una visió completa del procés.
- **Gestió de productes:** He implementat funcionalitats per administrar els productes disponibles, incloent-hi la seva creació, recuperació, eliminació i associació amb reserves específiques. Això permet gestionar tant la informació general dels productes com els detalls relacionats amb les comandes.
- **Estadístiques i anàlisis:** Per millorar l'experiència de l'usuari i proporcionar informació útil, he inclòs APIs que ofereixen estadístiques sobre les botigues més actives, els clients amb més reserves, les dependents més actives i els productes més reservats. Aquestes funcionalitats són essencials per optimitzar la gestió i prendre decisions informades.

7.3 Disseny i estructura del backend

El backend ha estat dissenyat tenint en compte l'escalabilitat i la modularitat, utilitzant una arquitectura basada en serveis per assegurar una separació clara de responsabilitats. Totes les operacions estan dissenyades per interactuar amb una base de dades centralitzada, assegurant que la informació es gestioni de manera coherent i que les operacions siguin ràpides i fiables.

A més, he implementat mecanismes per gestionar errors i excepcions de manera que el sistema pugui respondre adequadament a situacions inesperades, proporcionant missatges clars i facilitant la resolució de problemes.

8 Proves unitaries

8.1 FrontEnd

8.1.1 Sugerecias basadas en el feedback de los usuarios

Durant l'ús de l'aplicació, s'han recopilat comentaris i observacions de diversos usuaris. Aquestes històries descriuen casos concrets d'interacció amb l'aplicació i les millores suggerides.

8.1.2 Història 1: Anna, la dependenta multitask

Context: L'Anna treballa com a dependenta en una de les botigues que utilitzen l'aplicació. La seva feina principal és gestionar les reserves dels clients i coordinar-se amb l'obrador per assegurar que les comandes estiguin preparades a temps.

Experiència: L'Anna utilitza l'aplicació diàriament per comprovar les reserves que han fet els clients i confirmar la disponibilitat dels productes. Tot i que troba que l'aplicació és funcional, considera que algunes funcionalitats podrien ser més intuïtives i ràpides de fer servir, especialment durant hores de màxima activitat.

Suggeriments de millora:

- **Accés ràpid a les reserves d'avui:** L'Anna proposa incloure una secció destacada a la pantalla principal amb un llistat de les reserves del dia, ordenades per hora, perquè no hagi de buscar manualment entre totes les reserves.
- **Filtre per estat de reserva:** Li agradaria tenir un filtre senzill per veure només les reserves pendents de preparació, ja que sovint necessita accedir-hi ràpidament.
- **Gestió de l'estoc en temps real:** L'Anna suggereix que l'aplicació mostri una notificació si algun producte reservat està esgotat o té estoc baix, per poder avisar els clients immediatament.
- **Confirmació automàtica:** Quan marca una reserva com a "recollida", proposa que l'aplicació generi automàticament un rebut per al client, evitant haver de fer-ho manualment.
- **Millora en la visibilitat de l'horari:** També considera que seria útil tenir un calendari interactiu on pugui veure les seves hores de treball i els dies que té assignats.

Feedback adicional: L'Anna troba que el procés per assignar reserves a altres dependents, en cas que ella no pugui gestionar-les, és una mica complicat. Li agradaria un botó d'assignació ràpida que permetés transferir la reserva a una altra persona amb un sol clic.

8.1.3 Història 2: Jordi, el forner de l'obrador

Context: En Jordi treballa a l'obrador i utilitza l'aplicació principalment per gestionar les reserves diàries i planificar la producció. Tot i ser pràctic i eficient, no té gaire experiència amb eines digitals.

Experiència: A la pantalla de l'obrador, en Jordi aprecia l'ús del calendari per filtrar les reserves, però troba problemes quan cerca una reserva específica a causa de la poca visibilitat del camp de cerca. A més, quan obre una reserva, li costa tornar al llistat principal.

Suggeriments de millora:

- **Fer més visible la barra de cerca:** Mantenir la barra de cerca fixa a la part superior perquè sigui accessible en tot moment.
- **Botó "Enrere" més clar:** Afegir un botó destacat per tornar al llistat principal de reserves.
- **Ressaltar les reserves urgents:** Marcar les reserves urgents amb una icona especial o un color diferent per prioritzar-les.
- **Vista simplificada:** Oferir una opció per mostrar una vista bàsica amb només el nom del client i l'horari, sense detalls addicionals.

8.1.4 Història 3: Laura, la clienta curiosa

Context: La Laura és una clienta jove i familiaritzada amb les tecnologies. Utilitza l'aplicació per fer reserves de productes a la seva botiga local i valora molt l'estètica i la facilitat d'ús.

Experiència: La Laura troba l'aplicació funcional, però poc moderna. Té problemes per localitzar ràpidament el selector d'idioma i creu que la interfície podria ser més interactiva i visual.

Suggeriments de millora:

- **Disseny més visual:** Incloure imatges dels productes al formulari de reserves per fer-lo més atractiu.
- **Ubicació del selector d'idioma:** Reubicar el selector a la part superior dreta perquè sigui més accessible.
- **Animacions:** Afegir petites animacions en completar accions, com ara una marca de verificació verda en confirmar una reserva.
- **Mode fosc:** Implementar un mode fosc per millorar l'experiència en condicions de poca llum.

8.2 Backend-Proves i Validació

En aquesta secció es descriu l'enfocament adoptat per garantir la qualitat, funcionalitat i robustesa del sistema desenvolupat. Al llarg del procés, s'han implementat i executat diverses proves per verificar que les diferents funcionalitats compleixen amb els requisits especificats i operen correctament sota diverses condicions. Aquest procés és essencial per assegurar l'estabilitat de l'aplicació i la satisfacció dels usuaris finals.

8.2.1 Enfocament de les proves

L'objectiu principal de les proves realitzades va ser cobrir tant els aspectes funcionals com els casos límit del sistema. Per a això, es van dissenyar proves que validen les principals operacions de l'aplicació, com la gestió d'usuaris, dependents, botigues, clients i reserves, entre altres elements clau. Aquest enfocament es va fonamentar a assegurar que cada component del sistema funcioni correctament tant de forma aïllada com en conjunt.

A més, les proves van considerar tant els casos estàndard, on les dades proporcionades són correctes i completes, com els casos excepcionals, com intents d'accés amb credencials incorrectes, manipulació de dades inexistents i escenaris de duplicació de registres. Això va permetre validar no només el comportament esperat del sistema, sinó també la seva capacitat per gestionar errors i proporcionar respostes adequades en situacions anòmales.

8.2.2 Tipus de proves realitzades

Al llarg del desenvolupament, es van implementar diversos tipus de proves amb la finalitat de validar el sistema en diferents nivells i escenaris:

- **Proves de creació:** Aquestes proves van verificar que els recursos principals del sistema, com usuaris, dependents, botigues, clients i reserves, es poden crear correctament amb les dades proporcionades. Es van dissenyar per garantir que el sistema permeti la creació de nous elements de manera robusta i consistent.
- **Proves de recuperació:** Aquestes proves van assegurar que el sistema pot recuperar informació específica emmagatzemada a la base de dades, com llistes d'usuaris, dependents assignades a una botiga o reserves realitzades. Aquest tipus de proves és clau per garantir la integritat i accessibilitat de les dades.
- **Proves d'eliminació:** Es van implementar proves per confirmar que els recursos creats poden ser eliminats correctament. Aquestes proves també van validar que el sistema respon de manera adequada quan s'intenta eliminar elements que no existeixen.
- **Proves de duplicació:** Es van realitzar proves específiques per garantir que el sistema no permeti la creació de recursos duplicats, com clients amb el mateix nom i número de telèfon o dependents amb la mateixa informació

assignades a una botiga. Aquest tipus de validació és fonamental per mantenir la consistència de les dades.

- **Proves d'actualització:** Aquestes proves van comprovar que els recursos existents poden ser modificats, com l'actualització de contrasenyes o la reassignació de dependents a diferents botigues. Això va assegurar que el sistema sigui dinàmic i adaptable als canvis en les dades.
- **Proves d'autenticació:** Es van implementar proves per validar el procés d'inici de sessió i verificació de credencials, assegurant que els usuaris es poden autenticar correctament i que es gestionen adequadament els casos de credencials incorrectes.
- **Proves d'integritat:** Aquestes proves van verificar que les relacions entre els diferents recursos, com l'assignació de dependents a botigues o la relació entre clients i reserves, es mantenen consistents fins i tot després de realitzar operacions de creació, eliminació o actualització.
- **Proves de casos límit i errors:** Es van incloure proves dissenyades per avaluar com el sistema gestiona entrades incorrectes, dades incompletes o escenaris no previstos, com l'intent d'accedir a recursos que no existeixen. Aquestes proves són essencials per garantir la robustesa del sistema.
- **Proves de rendiment:** Encara que de manera limitada, es va validar que el sistema sigui capaç de gestionar múltiples operacions simultànies i que mantingui un rendiment acceptable sota càrrega. Això és crucial per assegurar una experiència fluida per a l'usuari.

8.2.3 Motivació darrere de les proves

El propòsit d'implementar aquesta varietat de proves és garantir que el sistema desenvolupat compleixi amb alts estàndards de qualitat i fiabilitat. Atès que el sistema gestiona dades sensibles i processos importants com la gestió de reserves i dependents en botigues, qualsevol error o inconsistència podria impactar negativament tant els usuaris com l'operació del sistema. Per tant, es va buscar prevenir errors abans que arribessin a l'entorn de producció.

La validació exhaustiva del sistema permet assegurar que cada mòdul compleix amb la seva funcionalitat específica, que la interacció entre els mòduls és coherent i que el sistema, en general, és resistent a fallades. A més, aquestes proves faciliten la identificació i correcció primerenca de possibles errors al codi, la qual cosa resulta més eficient i menys costós que fer-ho en fases posteriors.

8.2.4 Cobertura i anàlisi de proves amb Coverage

Per assegurar que les proves implementades cobreixen adequadament les funcionalitats del sistema, s'ha utilitzat l'eina de mesura de cobertura de codi Coverage. Aquesta eina permet identificar quines parts del codi han estat executades durant l'execució dels tests i quines no, proporcionant una visió clara del nivell de cobertura i ajudant a detectar àrees del codi que necessiten més proves.

El procés de mesura de cobertura s’ha dut a terme executant totes les proves dissenyades per al sistema i generant informes detallats. Aquests informes han permès quantificar el percentatge de línies del codi que estan cobertes pels tests, així com analitzar mòduls o funcionalitats específiques que podrien no estar adequadament validades.

8.3 Cobertura per secció de `main.py`

A continuació, es presenta una anàlisi gràfica de la cobertura del codi del fitxer `main.py`, organitzada per seccions funcionals. Aquest gràfic té com a objectiu reflectir l’abast dels tests realitzats sobre cadascuna de les funcionalitats principals de l’aplicació, destacant les àrees que necessiten millores en termes de cobertura.

Les seccions principals analitzades són:

- **Autenticació:** Inclou les funcionalitats per gestionar l’inici de sessió i la validació d’usuaris. Presenta una cobertura del 80%, amb algunes línies que podrien beneficiar-se de tests addicionals.
- **Gestió de botigues:** Comprèn operacions com la creació, consulta i actualització de botigues. Aquesta secció té una cobertura del 75%, sent una de les àrees amb més línies no testejades.
- **Gestió de dependents:** Enfocada a la gestió del personal de les botigues, aquesta secció té una cobertura del 80%, mostrant un bon nivell de validació.
- **Gestió de clients:** Englobant l’administració dels clients i les seves dades, aquesta secció té una cobertura destacable del 92%, essent una de les més completes en termes de proves.
- **Gestió de reserves:** Inclou funcionalitats per afegir, modificar i eliminar reserves. Aquesta secció és la més extensa del fitxer i presenta una cobertura del 71%, indicant que hi ha espai per optimitzar els tests.

Aquest gràfic posa en evidència l’esforç dedicat a assegurar la qualitat del codi a través de tests automatitzats, alhora que identifica oportunitats clares per millorar la robustesa del sistema.

8.4 Histograma de Cobertura

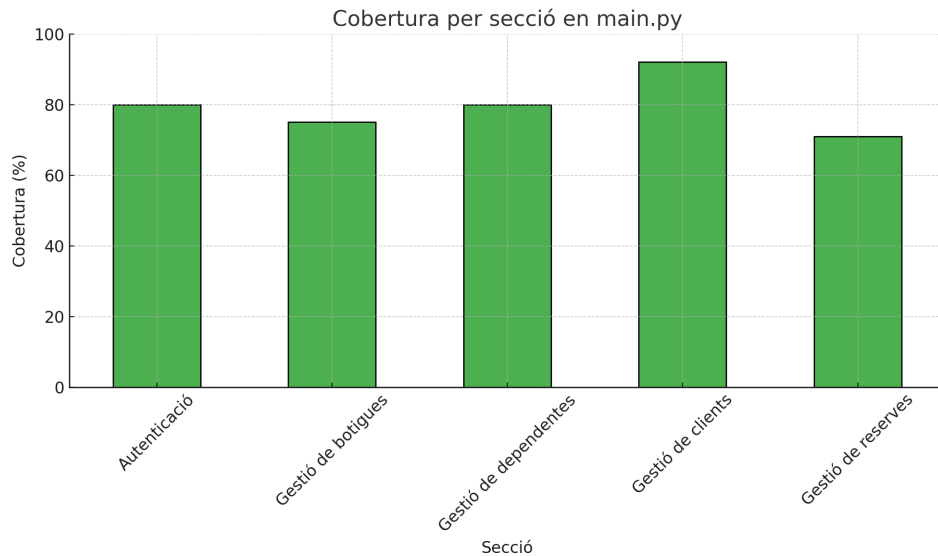


Figura 11: Histograma de Cobertura

Limitacions i millores identificades: Tot i que el sistema assoleix un alt nivell de cobertura global (al voltant del 83 percent del codi total), s'han identificat algunes àrees amb marge de millora, com la cobertura de casos menys freqüents o rutes específiques del codi que podrien no ser executades habitualment. Per exemple:

- S'han detectat algunes funcions auxiliars o camins alternatius en el codi que no han estat completament provats.
- L'anàlisi de cobertura ha revelat que certes funcionalitats relacionades amb casos límit o condicions excepcionals requereixen proves addicionals.

Importància dels resultats obtinguts: Aquest procés d'avaluació amb Coverage ha estat fonamental per garantir que les proves implementades no només cobreixin les funcionalitats més evidents, sinó també les situacions menys freqüents que podrien impactar en la robustesa del sistema. Els resultats obtinguts han confirmat que:

- El sistema és resistent i està preparat per a operar de manera fiable en entorns reals.
- Les possibles àrees de millora han estat clarament identificades i prioritzades per a iteracions futures.
- L'ús de Coverage ha facilitat la detecció i prevenció d'errors que podrien haver passat desapercebuts sense aquest anàlisi sistemàtic.

9 Deploy

9.1 Desplegament del Sistema

Per al desplegament del meu projecte, vaig utilitzar la plataforma **Render**, que em va permetre allotjar tant el backend com el frontend de manera gratuïta i eficient. A continuació, detallo els passos que vaig seguir per desplegar ambdós components i integrar-los correctament.

9.1.1 Desplegament del Backend

El backend, desenvolupat amb **FastAPI**, es va configurar i desplegar com un *Web Service* a Render. Els passos que vaig seguir van ser els següents:

- Vaig crear un nou servei a Render, vinculant-lo al repositori de **GitHub** que contenia el codi del backend.
- Vaig configurar les variables d'entorn necessàries, com les credencials per a Firebase i Google Cloud Storage, directament a la plataforma Render. Això em va permetre mantenir aquestes credencials segures i evitar incloure-les al codi font.
- Vaig especificar la comanda d'inici per executar el backend utilitzant **Uvicorn**, assegurant-me que el servidor de FastAPI arranqués correctament en l'entorn de producció.
- Vaig realitzar proves exhaustives dels endpoints utilitzant eines com **Postman** per assegurar-me que el backend funcionava correctament en producció.

A més, vaig configurar les polítiques de **CORS (Cross-Origin Resource Sharing)** perquè el backend acceptés sol·licituds únicament des del domini del frontend desplegat, augmentant així la seguretat de l'aplicació.

9.1.2 Desplegament del Frontend

El frontend, desenvolupat amb **Ionic** i **React**, es va configurar i desplegar com un *Static Site* a Render. El procés que vaig seguir inclou:

- Vaig generar la versió de producció del projecte utilitzant **Vite**. Això va crear un directori de sortida (**dist**) amb tots els fitxers estàtics necessaris per al desplegament.
- Vaig configurar un nou lloc estàtic a Render, vinculant-lo al repositori corresponent de **GitHub**.
- Vaig especificar el directori **dist** com la carpeta que conté els fitxers a servir en l'entorn de producció.
- Vaig afegir un fitxer **static.json** amb regles específiques de redirecció perquè totes les rutes del frontend fossin gestionades correctament per **index.html**, evitant errors en recarregar les pàgines.

- Vaig validar el desplegament accedint al domini públic proporcionat per Render i comprovant que el frontend funcionés correctament.

9.1.3 Integració entre Backend i Frontend

Per integrar el backend i el frontend, vaig realitzar les següents configuracions:

- Vaig utilitzar variables d'entorn al frontend per emmagatzemar la URL base del backend. Això va permetre una fàcil separació entre l'entorn de desenvolupament i el de producció.
- Vaig ajustar les polítiques de CORS al backend per permetre únicament les sol·licituds provinents del domini del frontend.
- Vaig provar totes les interaccions entre el backend i el frontend, incloent operacions clau com l'inici de sessió, la càrrega d'imatges des de Google Cloud Storage i les operacions CRUD de l'aplicació.

9.1.4 Resultats del Desplegament

El resultat final va ser un sistema completament funcional, amb el backend allotjat com un servei dinàmic a Render i el frontend com un lloc estàtic. Ambdós components van quedar correctament integrats, cosa que va permetre una experiència d'usuari fluida i segura. El desplegament em va proporcionar una solució robusta i accessible, que compleix amb els requisits del projecte i està preparada per ser utilitzada en un entorn real.

10 Resultats/conclusions/Treball futur

10.1 Conclusions

El desenvolupament d'aquesta aplicació ha estat un procés complex i enriquidor, que ha permès integrar múltiples tecnologies i paradigmes per abordar les necessitats específiques del projecte. Al llarg del treball s'ha consolidat un sistema que combina un *frontend* modern i dinàmic basat en Ionic i React amb una arquitectura *backend* robusta i escalable utilitzant FastAPI, Firebase i Firestore.

Entre els assoliments més destacats del projecte, cal mencionar:

- **Complexitat tècnica i solucions implementades:** S'ha dissenyat i implementat funcionalitats clau com l'autenticació segura, la gestió eficient de dades amb Firestore i la implementació d'anàlisis avançats d'activitat. A més, s'han integrat eines modernes com Docker Compose per garantir un desplegament eficient i l'escalabilitat del sistema.
- **Desenvolupament incremental i adaptatiu:** S'ha evidenciat la importància de l'adaptabilitat en un entorn de desenvolupament iteratiu. Moltes funcionalitats i millores han sorgit de manera progressiva en resposta als reptes trobats i al feedback obtingut durant les primeres fases del projecte.
- **Resultats funcionals i experiència d'usuari:** L'aplicació final ha assolit un nivell de maduresa que permet als usuaris i administradors interactuar amb un sistema fluid, fiable i ben estructurat, oferint funcionalitats essencials com la gestió de botigues, reserves i anàlisis personalitzats.

Tot i això, també s'han identificat diversos reptes durant el desenvolupament:

- La corba d'aprenentatge associada a la integració de diverses tecnologies noves i el temps dedicat a la resolució de problemes específics.
- La necessitat d'ajustar constantment els requisits inicials per reflectir de manera més realista les necessitats de l'usuari final i les limitacions tecnològiques.

Finalment, aquest projecte no només ha complert amb els objectius inicials, sinó que també ha assentat les bases per a futures expansions i millores. Algunes direccions futures inclouen l'optimització del rendiment, la incorporació de noves funcionalitats basades en intel·ligència artificial, i la possibilitat d'extendre l'aplicació cap a altres sectors.

En resum, el treball realitzat ha estat un exemple clar de com la planificació, la flexibilitat i l'esforç continu poden convergir en el desenvolupament d'un producte tecnològic de gran valor. A més, aquest projecte ha proporcionat una experiència formativa clau que contribuirà, sens dubte, al meu desenvolupament professional en l'àmbit de l'enginyeria informàtica.

10.2 Treball a futur

Tot i que l'aplicació desenvolupada cobreix la major part de les necessitats actuals de l'empresa, hi ha diverses millores que es podrien implementar en el futur per ampliar les funcionalitats i millorar-ne l'eficiència.

10.2.1 Ampliar el sistema de notificacions en temps real

Actualment, la plataforma ofereix la sincronització en temps real de dades gràcies a Firebase, però una millora significativa seria l'implementació d'un sistema de notificacions push que alerti els usuaris de les actualitzacions més importants. Per exemple, els clients podrien rebre notificacions quan una reserva estigui preparada, i les dependents podrien ser notificades quan es facin noves comandes o s'actualitzin les existents.

10.2.2 Millorar el sistema de reportes i estadístiques

Tot i que l'aplicació ja ofereix estadístiques bàsiques sobre les reserves i l'estat de les comandes, una futura millora seria implementar un sistema més complet d'anàlisi de dades. Això podria incloure informes sobre els productes més venuts, hores punta de reserves, anàlisi de la demanda en diferents botigues i recomanacions basades en tendències de consum. Aquesta informació ajudaria a l'empresa a prendre decisions estratègiques basades en dades concretes.

10.2.3 Integració amb sistemes de pagament

Una possible ampliació de l'aplicació seria la integració de mètodes de pagament en línia per permetre que els clients puguin pagar les seves reserves directament des de l'aplicació. Això simplificaria el procés de compra i oferiria més comoditat tant als clients com a l'empresa.

10.2.4 Cobertura dels Tests

Tot i que s'han implementat tests bàsics per validar les funcionalitats principals de l'aplicació, una millora a considerar seria l'ampliació de la cobertura dels tests, incloent proves més detallades sobre lògica de negoci, seguretat i rendiment. Això garantiria una major robustesa de l'aplicació i facilitaria la detecció de possibles errors abans de desplegar actualitzacions.

10.2.5 Gestió d'estoc

Un altre punt a desenvolupar seria la incorporació d'un sistema de gestió d'estoc per controlar la disponibilitat dels productes en cada botiga. Això permetria automatitzar el seguiment de l'inventari, generar alertes quan el nivell d'estoc sigui baix i facilitar la gestió logística de l'empresa.

10.2.6 Evolució

Actualment, l'aplicació està dissenyada per ser utilitzada internament per l'empresa i el seu personal, però un objectiu a futur seria fer-la accessible als clients finals. Això permetria que els clients puguin realitzar reserves directament des de l'aplicació, consultar l'estat de les seves comandes i gestionar els seus comptes personals. A més, es podria integrar amb una pàgina web per oferir una experiència unificada i centralitzada per a tots els usuaris.

10.2.7 Canvi d'Idioma

Tot i que l'aplicació permet actualment canviar l'idioma de la interfície gràcies a la seva integració amb un sistema de traducció, els noms dels productes i altres dades emmagatzemades a la base de dades romanen en un sol idioma. Una millora futura seria implementar un sistema multilingüe per als productes i dades dinàmiques. Aquest sistema permetria als usuaris visualitzar els noms dels productes, categories i altres dades en l'idioma seleccionat, millorant significativament l'experiència d'usuari per a clients amb diferents preferències lingüístiques. Aquest canvi faria l'aplicació més accessible i adaptable a un públic més divers, especialment en cas d'expansió de l'empresa a altres regions o mercats internacionals.

10.2.8 Optimització del funcionament dels filtres

Tot i que els filtres actuals de l'aplicació permeten una selecció bàsica de dades segons criteris com botiga, data o categoria, el sistema encara presenta limitacions en la combinació de múltiples filtres de manera simultània. Un treball a futur seria garantir que totes les possibles combinacions de filtres funcionin de manera òptima i coherent, independentment de l'ordre en què es seleccionin o apliquin. Aquesta millora inclouria la revisió de la lògica dels filtres al backend i una interfície més intuïtiva al frontend per mostrar els resultats filtrats en temps real. A més, es podrien implementar eines de depuració per identificar i resoldre possibles conflictes entre filtres, millorant així la precisió i la fiabilitat de la funcionalitat per als usuaris.

10.2.9 Seguretat

Amb l'objectiu de garantir la seguretat de les dades i protegir la privacitat dels usuaris, seria important implementar mesures avançades de seguretat. Això inclou la incorporació de sistemes de xifratge robust tant per a la transmissió com per a l'emmagatzematge de dades, autenticació de dos factors per als comptes d'usuari i mecanismes de detecció d'activitats sospitoses. A més, es podrien establir auditories periòdiques de seguretat i simulacions d'atacs per identificar vulnerabilitats potencials. Aquestes millores serien essencials especialment en cas d'ampliar l'accés de l'aplicació als clients finals i gestionar dades sensibles, com ara informació de pagament o perfils personals.

11 ANEX

Històries d'Usuari

A continuació, es mostren les històries d'usuari que es van definir al començament del projecte per tenir una idea clara de com iniciar el desenvolupament. Aquestes històries han servit de guia inicial i, en alguns casos, s'han mantingut fidels al plantejament original, mentre que en altres s'han hagut d'actualitzar i modificar per adaptar-les a les necessitats reals del projecte.

- **HU01: Registre d'usuaris**
 - **Com:** Administrador.
 - **Vull:** Registrar noves botigues i assignar-los comptes d'usuari úniques.
 - **Per:** Permetre que gestionin les seves pròpies reserves i dades de manera independent.
- **HU02: Inici de sessió**
 - **Com:**Usuari del sistema (Administrador, Dependenta o Botiga).
 - **Vull:** Iniciar sessió amb les meves credencials.
 - **Per:** Accedir a les funcionalitats específiques segons el meu rol.
- **HU03: Gestió de reserves**
 - **Com:**Usuari autoritzat (Dependenta o Administrador).
 - **Vull:** Crear, modificar i cancel·lar reserves.
 - **Per:** Gestionar les necessitats dels clients i assegurar que les comandes estiguin organitzades.
- **HU04: Visualització d'horaris**
 - **Com:** Dependenta o Administrador.
 - **Vull:** Veure un calendari interactiu amb les reserves organitzades per data i botiga.
 - **Per:** Planificar eficientment la feina diària.
- **HU05: Gestió de productes**
 - **Com:** Administrador.
 - **Vull:** Afegir, modificar i eliminar productes del sistema.
 - **Per:** Mantenir actualitzada l'oferta disponible per a les reserves.
- **HU06: Gestió de clients**
 - **Com:** Administrador o Dependenta.
 - **Vull:** Registrar i gestionar informació dels clients.

- **Per:** Associar reserves i oferir un servei personalitzat.
- **HU07: Gestió de dependents**
 - **Com:** Administrador.
 - **Vull:** Assignar dependents a botigues i gestionar els seus rols.
 - **Per:** Garantir que cada botiga tingui prou personal.
- **HU08: Estadístiques del sistema**
 - **Com:** Administrador.
 - **Vull:** Visualitzar estadístiques sobre reserves, clients i dependents.
 - **Per:** Prendre decisions informades basades en dades.
- **HU09: Canvi d'idioma**
 - **Com:** Usuari del sistema.
 - **Vull:** Canviar l'idioma de la interfície.
 - **Per:** Utilitzar l'aplicació en el meu idioma preferit.
- **HU10: Gestió de categories de productes**
 - **Com:** Administrador.
 - **Vull:** Classificar els productes en categories.
 - **Per:** Facilitar la cerca i organització dels productes.

Referències

- [1] Ionic framework. <https://ionicframework.com/>, 2024. Accés el 14-Oct-2024.
- [2] Material-ui (mui) for react. <https://mui.com/>, 2024. Accés el 14-Oct-2024.
- [3] Visual studio code. <https://code.visualstudio.com/>, 2024. Accés el 14-Oct-2024.
- [4] bcrypt. bcrypt documentation. <https://pypi.org/project/bcrypt/>, 2024. Accés el 14-Oct-2024.
- [5] React Calendar. React calendar - a flexible calendar component for react. <https://www.npmjs.com/package/react-calendar>, 2024. Accés el 14-Oct-2024.
- [6] Docker Team. Docker compose documentation, 2025. Últim accés: Gener 2025.
- [7] Docker Team. Docker compose learning curve, 2025. Últim accés: Gener 2025.
- [8] Docker Team. Docker compose yaml file reference, 2025. Últim accés: Gener 2025.
- [9] Docker Team. Overview of docker compose, 2025. Últim accés: Gener 2025.
- [10] Docker Team. Portability with docker compose, 2025. Últim accés: Gener 2025.
- [11] Docker Team. Resource usage in docker compose, 2025. Últim accés: Gener 2025.
- [12] Docker Team. Scaling applications with docker compose, 2025. Últim accés: Gener 2025.
- [13] MDN Web Docs. Fetch api. https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API, 2024. Accés el 14-Oct-2024.
- [14] Inc. Facebook. React - a javascript library for building user interfaces. <https://reactjs.org/>, 2013. Accés el 14-Oct-2024.
- [15] FastAPI. Fastapi documentation. <https://fastapi.tiangolo.com/>, 2024. Accés el 14-Oct-2024.
- [16] Google Cloud Team. Google cloud storage api documentation, 2025. Últim accés: Gener 2025.
- [17] Google Cloud Team. Overview of google cloud storage, 2025. Últim accés: Gener 2025.
- [18] Google Cloud Team. Performance of google cloud storage, 2025. Últim accés: Gener 2025.

- [19] Google Cloud Team. Pricing of google cloud storage, 2025. Últim accés: Gener 2025.
- [20] Google Cloud Team. Scalability of google cloud storage, 2025. Últim accés: Gener 2025.
- [21] Google Cloud Team. Security features in google cloud storage, 2025. Últim accés: Gener 2025.
- [22] Google Cloud Team. Setup and permissions in google cloud storage, 2025. Últim accés: Gener 2025.
- [23] Google Cloud Team. Signed urls in google cloud storage, 2025. Últim accés: Gener 2025.
- [24] i18next. React i18next documentation. <https://react.i18next.com/>, 2024. Accés el 14-Oct-2024.
- [25] Google LLC. Cloud firestore. <https://firebase.google.com/docs/firestore>, 2024. Accés el 14-Oct-2024.
- [26] Google LLC. Firebase. <https://firebase.google.com/>, 2024. Accés el 14-Oct-2024.
- [27] Google LLC. Firebase admin sdk. <https://firebase.google.com/docs/admin>, 2024. Accés el 14-Oct-2024.
- [28] Eric A. Meyer. *Cascading Style Sheets: The Definitive Guide*. O'Reilly Media, 4th edition, 2017. Accés el 14-Oct-2024.
- [29] Render Team. How render handles scalability, 2025. Últim accés: Gener 2025.
- [30] Render Team. Render: Cloud hosting for developers, 2025. Últim accés: Gener 2025.
- [31] Render Team. Render documentation, 2025. Últim accés: Gener 2025.
- [32] Render Team. Render features, 2025. Últim accés: Gener 2025.
- [33] Render Team. Render free tier limitations, 2025. Últim accés: Gener 2025.
- [34] Render Team. Render pricing plans, 2025. Últim accés: Gener 2025.
- [35] React Training. React router - declarative routing for react.js. <https://reactrouter.com/>, 2014. Accés el 14-Oct-2024.