

Grau en Enginyeria Informàtica de Gestió i Sistemes d'Informació

SOFTWARE DE GESTIÓ D'ESTUCATGE PER A PASTISSERIES
PIMEs

Memòria

ANASS ARRAHI ARRAI
TUTORA: ROSA HERRERO

5È – CURS 2023/2024

Resum

Aquest projecte es centra en el desenvolupament d'un software de gestió d'estucatge per a pastisseries PIME, amb l'objectiu d'optimitzar la realització de comandes, simplificar l'administració de l'inventari i millorar l'eficiència operativa. S'analitzen solucions existents, identificant la falta d'una opció adaptada específicament a les necessitats d'aquest sector. El projecte segueix una metodologia àgil i incorpora requisits funcionals i tecnològics específics, culminant en una anàlisi de viabilitat que demostra els reptes i oportunitats de la proposta.

Resumen

Este proyecto se centra en el desarrollo de un software de gestión de inventarios para pastelerías PYMES, con el objetivo de optimizar la realización de pedidos, simplificar la administración del inventario y mejorar la eficiencia operativa. Se analizan soluciones existentes, identificando la falta de una opción adaptada específicamente a las necesidades de este sector. El proyecto sigue una metodología ágil e incorpora requisitos funcionales y tecnológicos específicos, culminando en un análisis de viabilidad que demuestra los retos y oportunidades de la propuesta.

Abstract

This project focuses on the development of inventory management software for SME bakeries, aiming to optimize order fulfillment, simplify inventory administration, and improve operational efficiency. Existing solutions are analyzed, identifying the lack of an option specifically tailored to the needs of this sector. The project follows an agile methodology and incorporates specific functional and technological requirements, culminating in a feasibility analysis that demonstrates the challenges and opportunities of the proposal.

Índex

Índex de figures.....	5
Glossari de termes	7
1. Introducció	9
2. Marc teòric	11
2.1. Gestió d'Estucatge	11
2.2. Gestió de Comandes	11
2.3. Estucatge en Pastisseries	12
2.4. Aplicacions Existents	12
2.5. Conclusió.....	13
3. Objectius i abast	15
3.1. Objectius de client i de software.....	15
3.2. Objectius de públic i usuari final	15
4. Antecedents històrics.....	17
4.1. Sistemes de gestió genèrics	17
4.2. Recopilació d'informació i el seu estudi	17
4.3. Sistemes de gestió POS (<i>Point of Service</i>) amb software de gestió d'inventari.....	17
5. Metodologia	19
5.1. Procés de realització	19
5.1.1. Investigació i Anàlisis	19
5.1.2. Disseny i Planificació	19
5.1.3. Desenvolupament de Software.....	20
5.1.4. Avaluació	20
5.2. Estratègies.....	20
5.2.1. Cerca d'informació	20
5.2.2. Anàlisis i síntesis.....	21
5.2.3. Procés de desenvolupament	21
6. Desenvolupament.....	23
6.1. Metodologia de desenvolupament	23
6.1.1. Etapes del Projecte	23
6.1.2. Reunions amb la Tutora del TFG.....	25

6.2. Requeriments funcionals i no funcionals:.....	26
6.3. Casos d'ús	28
6.3.1. Registre de Proveïdors	29
6.3.2. Desactivació d'un proveïdor	30
6.3.3. Modificació d'un proveïdor	31
6.3.4. Creació d'un Producte o Ítem d'un Proveïdor.....	32
6.3.5. Creació d'una comanda a un proveïdor	33
6.3.6. Consulta d'una comanda	34
6.3.7. Eliminació d'una comanda.....	35
6.3.8. Enviament d'una comanda al proveïdor	36
6.3.9. Confirmació de recepció de comanda	38
6.4. Diagrames de seqüència.....	39
6.5. Diagrama de classes.....	45
6.6. Diagrama de base de dades	46
6.6.1. Què és Hive	46
6.6.2. Característiques de <i>Hive</i>	46
6.6.3. Integració de Hive en el Projecte	46
6.6.4. Estructura de dades en Hive	47
6.7. Disseny de la interfície d'usuari (UI)	48
6.7.1. <i>Wireframes</i>	48
6.7.2. Prototips	49
6.7.3. Consideracions d'Usabilitat.....	50
6.7.4. Resultats Finals	52
6.7.5. Anàlisi General.....	57
6.8. Implementació amb <i>Flutter</i> i <i>Dart</i>	58
6.8.1. Estructura de Fitxers	58
6.8.3. Practiques de Desenvolupament	59
6.9. Integració i Proves	61
6.9.1. Problemes Detectats i Solucions Implementades.....	61
6.9.2. Feedback dels Usuaris i la Tutora	66
7. Anàlisis de resultats.....	68
8. Conclusions.....	70

9. Possibles ampliacions.....	72
10. Bibliografia	74

Índex de figures

Fig. 6.3.1. Diagrama de casos d'us.....	28
Fig. 6.4.1. Diagrama de seqüència per a la creació d'un producte o ítem d'un proveïdors...	39
Fig. 6.4.2. Diagrama de seqüència per a la creació d'una comanda.....	40
Fig. 6.4.3. Diagrama de seqüència per a l'eliminació d'una comanda.....	41
Fig. 6.4.4. Diagrama de seqüència per a la consulta d'una comanda.....	42
Fig. 6.4.5. Diagrama de seqüència per a l'enviament d'una comanda al proveïdor.....	43
Fig. 6.4.6. Diagrama de seqüència per a la confirmació de recepció d'una comanda.....	44
Fig. 6.5.1. Diagrama de classes.....	45
Fig. 6.6.4.1. Diagrama d'estructura de dades.....	47
Fig. 6.7.1.1. Wireframe aplicació.....	48
Fig. 6.7.2.1. Prototip funcional aplicació.....	49
Fig. 6.7.4.1. Procediment creació d'un nou proveïdor.....	52
Fig. 6.7.4.2. Procediment de crear i enviar una nova comanda°.....	53
Fig. 6.7.4.3. Procediment de consulta a una comanda, marcar-la com a rebuda, i consultar estoc.....	54
Fig. 6.7.4.4. Procediment afegir ítem a un proveïdor.....	55
Fig. 6.7.4.5. Procediment actualitzar estoc.....	55
Fig. 6.9.1.1. Problema i solució amb formularis.....	61
Fig. 6.9.1.2. Problema i solució amb text tallat.....	62
Fig. 6.9.1.3. Millora en consulta de comandes.....	63

Fig. 6.9.1.4. Millora detall del format del producte.....64

Fig. 6.9.1.5. Ajuda amb les accions sobre proveïdors.....65

Glossari de termes

TFG	Treball Final de Grau
PIME	Petita i mitjana empresa
POS	<i>Point of Service</i>
BBDD	Bases de dades
TPV	Terminal Punt de Venta

1. Introducció

El projecte té com a finalitat la creació d'un programari de gestió dissenyat per a les necessitats de les pastisseries de mida mitjana i gran, també conegudes dins del marc de les petites i mitjanes empreses (PIMEs). Aquesta iniciativa es centra en oferir una solució que tracti directament amb les demandes úniques d'aquest sector, destacant especialment en la millora del control d'inventari i en la simplificació dels procediments per realitzar comandes als proveïdors.

El projecte busca introduir solucions tecnològiques que facin més eficients i menys propensos a errors els processos de gestió d'estocs i de comandes. Les funcionalitats previstes estan dissenyades per minimitzar els errors administratius, optimitzar l'ús dels recursos disponibles i, com a resultat, augmentar significativament l'eficiència operativa global de les empreses.

A més, el programari en desenvolupament estendrà les seves capacitats més enllà de la gestió d'inventaris i comandes, incorporant eines avançades per al control i gestió de distribuïdors i proveïdors. Això permetrà a les pastisseries no només gestionar de manera més eficaç l'estoc i les comandes de productes, sinó també millorar la selecció dels seus proveïdors i distribuïdors.

A part, el programari incorporarà un potent mòdul d'estadístiques generals, una funcionalitat clau que permetrà als usuaris accedir a dades i anàlisis detallats sobre diversos aspectes del seu negoci. Aquesta capacitat d'anàlisi i visualització de dades serà fonamental per a una presa de decisions més informada i estratègica. Els usuaris podran visualitzar informes personalitzats que reflecteixin tendències de venda, patrons de consum d'inventari, eficàcia dels proveïdors i distribuïdors, així com altres mètriques operatives vitals.

Amb aquest enfocament holístic, el projecte aspira no només a resoldre els desafiaments actuals dins de la gestió quotidiana de les pastisseries, sinó també a proporcionar una base sòlida per al creixement i l'expansió futura d'aquests negocis en un mercat cada vegada més competitiu.

2. Marc teòric

2.1. Gestió d'Estucatge

La gestió d'estucatge és un aspecte fonamental en la logística [1] i operacions de qualsevol negoci que maneja inventari, incloent pastisseries. Consisteix en la planificació, organització i control de les existències per assegurar que els productes estiguin disponibles quan es necessiten, minimitzant al mateix temps els costos associats a l'emmagatzematge.

- **Planificació:** Implica determinar quins productes es necessiten, en quina quantitat i quan s'han de rebre. Això pot incloure la previsió de la demanda i l'establiment de punts de re-ordre per evitar ruptures d'estoc.
- **Organització:** Es tracta de la manera com s'organitzen els productes dins del magatzem o àrea d'emmagatzematge. Això pot incloure l'assignació d'espai, etiquetatge adequat i sistemes per localitzar fàcilment els productes.
- **Control:** Consisteix en mantenir registres precisos dels nivells d'estoc, incloent-hi entrades i sortides de productes. Això pot implicar l'ús de sistemes informàtics per seguir l'inventari en temps real.

2.2. Gestió de Comandes

La gestió de comandes [2] és el procés de rebre, processar i complir les comandes dels clients. Aquest procés inclou diverses fases:

- **Recepció de Comandes:** Les comandes poden ser rebudes a través de diferents canals com telèfon, correu electrònic o sistemes de comerç electrònic.
- **Processament de Comandes:** Una vegada rebuda la comanda, es verifica la disponibilitat dels productes i es prepara la documentació necessària per al seu compliment.
- **Compliment de Comandes:** Implica recollir els productes de l'inventari, empaquetar-los adequadament i organitzar el seu enviament o lliurament.

- **Seguiment i Control:** Supervisar l'estat de les comandes per assegurar-se que s'entreguen a temps i gestionar qualsevol incidència que pugui sorgir durant el procés.

2.3. Estucatge en Pastisseries

Les pastisseries tenen necessitats específiques d'estucatge [3] a causa de la naturalesa dels seus productes, que solen ser peribles i requereixen condicions d'emmagatzematge especials.

- **Conservació de Productes Peribles:** Els ingredients i productes finals sovint necessiten ser refrigerats o congelats per mantenir la seva frescor i qualitat. Això implica l'ús de sistemes de refrigeració adequats.
- **Control de Dates de Caducitat:** És crucial seguir les dates de caducitat dels ingredients i productes acabats per evitar el malbaratament i assegurar la seguretat alimentària.
- **Gestió d'Ingredients:** Les pastisseries utilitzen una àmplia gamma d'ingredients, molts dels quals són utilitzats en petites quantitats però amb alta freqüència. Això requereix un control estricte per evitar l'escassetat d'ingredients clau.

2.4. Aplicacions Existents

En el mercat actual, existeixen diverses aplicacions [4] i sistemes de gestió de l'inventari i comandes que ajuden les empreses a gestionar les seves operacions de manera més eficient. Algunes de les més populars inclouen:

- **Zoho Inventory:** Aquesta eina permet gestionar l'inventari, les comandes i vendes. Ofereix integració amb diverses plataformes de comerç electrònic i transport.
- **Square for Retail:** Aquesta aplicació proporciona eines per gestionar l'inventari, les vendes i els clients. Està dissenyada per a botigues de venda al detall, incloent pastisseries, i ofereix una interfície fàcil d'utilitzar.
- **TradeGecko:** Aquesta plataforma està dissenyada per a empreses que necessiten una gestió avançada d'inventaris i comandes. Ofereix funcionalitats com el seguiment en temps real, la integració amb comerç electrònic i la gestió de magatzems.

2.5. Conclusió

La gestió eficient de l'estucatge i les comandes és essencial per a l'èxit de qualsevol pastisseria. L'adopció de tecnologies modernes i aplicacions especialitzades pot ajudar a millorar significativament aquests processos, assegurant que els productes estiguin disponibles quan els clients els necessiten, mantenint al mateix temps els costos baixos i la qualitat alta. En aquest context, el desenvolupament d'una aplicació específica per a la gestió d'inventaris i comandes en pastisseries representa una oportunitat per millorar l'eficiència operativa i la satisfacció del client.

3. Objectius i abast

Seguidament es detalla un llistat d'objectius que es tenen en aquest treball. Se separaran per tres tipus d'objectius; Objectius de client, objectius de producte o servei i objectius de públic o usuari final.

3.1. Objectius de client i de software

- **Optimitzar la gestió d'estucatge:** Desenvolupar un sistema que permet al client final gestionar d'una manera més eficient els productes i matèries primeres per a la producció de la pastisseria.
- **Facilitar la realització de comandes:** Implementar un sistema que faciliti les comandes als proveïdors, agilitzant les comandes.
- **Millorar l'eficiència del negoci:** Proporcionar eines d'anàlisi i informes que ajudin al client comprendre millor el seu rendiment, costos i tendències d'utilització d'ingredients, donant impuls la presa de decisions estratègiques.

3.2. Objectius de públic i usuari final

- **Propietaris de pastisseries PIMEs:**
Aquest grup inclou els propietaris de negocis PIMEs del sector pastisser que busquen optimitzar els processos operatius del seu negoci. Es vol millorar l'eficiència en la gestió d'estoc, facilitar les comandes a proveïdors, i prendre decisions basades en dades gràcies a les estadístiques generals proporcionades pel sistema.
- **Empreses gestores d'inventari:**
Empreses responsables de l'administració de l'estoc en pastisseries. Aquests trobaran en el software una eina indispensable per al seguiment precís dels productes, matèries primeres i subministraments, ajudant a prevenir l'excés d'inventari o escassetats que podrien interrompre la producció, i assegurant un servei excel·lent al seu client.
- **Encarregats de compres i subministraments:**
Aquest grup està format per aquells encarregats de realitzar les comandes a proveïdors i gestionar les relacions comercials amb aquests. El software els permetrà optimitzar aquest procés, millorar la comunicació i negociar millors condicions, basant-se en l'anàlisi de dades de rendiment dels proveïdors.

4. Antecedents històrics

Estudiarem i contrastem diferents solucions tecnològiques ja existents que es poden aplicar per a la gestió d'una pastisseria de petita i mitjana escala.

4.1. Sistemes de gestió genèrics

Excel, *Google Sheets*, *Odoo* [1] o *QuickBooks* [6] són sistemes que ofereixen funcionalitats bàsiques de seguiment d'inventari, però manquen de funcionalitats que s'adaptin específicament a les necessitats d'una empresa PIME del sector alimentari, com per exemple la gestió de productes amb data de caducitat.

4.2. Recopilació d'informació i el seu estudi

Sistemes com *FreshCheq* [7], o *MarketMan* [8], són softwares que presenten funcionalitats i característiques més alineades amb les necessitats específiques de negocis com podria ser en el nostre cas una pastisseria. Permeten la gestió de caducitat, seguiment d'ingredients frescos, i disposen d'eines d'anàlisi de costos. No obstant això, aquests sistemes poden resultar costoses d'implementar per a petits negocis i fins i tot per a mitjans, i no només això sinó que a més poden ser massa complexes per les necessitats bàsiques que requereix un negoci com el que ens volem enfocar nosaltres.

4.3. Sistemes de gestió POS (*Point of Service*) amb software de gestió d'inventari

Aquestes són les eines que últimament són virals, i podem dir que estan de moda. Solucions com *Square* [9], ofereixen un sistema que permet un seguiment a temps real de les vendes i l'actualització automàtica de l'inventari. Aquest tipus de sistemes, resulten atractives a primera vista, però moltes vegades són sistemes que van lligats a un software de la mateixa empresa desenvolupadora, el qual t'obliga a comprar-los i sovint, no són gens assequibles per a empreses petites.

5. Metodologia

5.1. Procés de realització

La metodologia per a aquest desenvolupament s'estructura en quatre etapes clau que garanteixen la creació d'una solució efectiva i adaptada a les necessitats específiques del sector. A continuació, es detalla amb més profunditat cada un dels passos involucrats en el procés de realització:

5.1.1. Investigació i Anàlisi

En aquesta fase inicial, es realitzarà una revisió exhaustiva de les solucions de software existents en el mercat, amb un enfoc especial en aquelles dirigides al sector alimentari i, més específicament, al segment de la pastisseria. S'analitzaran les tendències actuals en gestió d'inventari, processament de comandes i anàlisi de dades per identificar oportunitats de millora i innovació. Aquesta investigació inclourà:

- Recopilació de dades sobre les millors pràctiques en gestió d'inventari i logística en el sector alimentari.
- Anàlisi de casos d'estudi i històries d'èxit per comprendre les estratègies que han resultat efectives.
- Entrevistes amb propietaris de pastisseries i altres *stakeholders* per identificar les seves necessitats específiques i els desafiaments que afronten amb les solucions actuals.

5.1.2. Disseny i Planificació

Amb la informació recollida, s'iniciarà la fase de disseny i planificació, on es s'acabaran de definir els requisits funcionals i no funcionals del software. Això inclou:

- La creació de requeriments funcionals i no funcionals, i diagrames de casos d'ús per identificar les interaccions dels usuaris amb el sistema i definir les funcionalitats clau.
- Elaboració de diagrames de flux que detallin el procés de les operacions dins del sistema, com la gestió d'inventari i el processament de comandes.
- Disseny de l'estructura de la base de dades, tenint en compte la necessitat d'emmagatzemar i gestionar grans volums de dades de manera eficient.
- Planificació de la interfície d'usuari, prioritzant la usabilitat i l'experiència d'usuari.

5.1.3. Desenvolupament de Software

Durant aquesta etapa, es procedirà a la implementació del software, seguint les especificacions definides en la fase de disseny. Es prioritzaran els mòduls de gestió d'inventari, comandes, i anàlisi de dades, seguint una metodologia àgil que permeti l'adaptació a canvis i la incorporació de millores contínues. Aquesta fase inclourà:

- Codificació de les funcionalitats del sistema, utilitzant *Flutter*.
- Desenvolupament de proves unitàries i d'integració per assegurar la qualitat del codi i la fiabilitat de les funcions del sistema.

5.1.4. Avaluació

Finalment, el software serà sotmès a una etapa d'avaluació que inclourà:

- Proves beta amb un grup selecte d'usuaris finals per identificar errors, problemes d'usabilitat i àrees de millora.
- Recollida i anàlisi del feedback obtingut durant les proves per realitzar ajustos i optimitzacions necessàries.
- Avaluació de l'impacte del software en la millora dels processos operatius i en la presa de decisions dels usuaris.

Aquest enfocament metodològic permetrà el desenvolupament d'un software robust, adaptable i altament enfocat a satisfer les necessitats reals dels propietaris de pastisseries PIMES, assegurant una implementació exitosa i una adopció àmplia dins del sector.

5.2. Estratègies

5.2.1. Cerca d'informació

La cerca d'informació serà una tasca contínua i meticulosa, centrada en l'exploració de llibres acadèmics [10], estudis previs, articles tecnològics, i fonts especialitzades en gestió d'inventari i sistemes de software per negocis del sector alimentari.

Es prestarà especial atenció a l'anàlisi de casos d'estudi i experiències d'implementació de solucions similars per comprendre millor les millors pràctiques, els desafiaments comuns i les solucions innovadores aplicades en contextos similars.

5.2.2. Anàlisi i síntesis

Una vegada recopilada la informació, es realitzarà un anàlisi comparatiu exhaustiu de les diferents metodologies de desenvolupament de software [11], com ara models en cascada, desenvolupament iteratiu, desenvolupament àgil, entre d'altres. L'objectiu serà seleccionar o adaptar la metodologia més apropiada que s'ajusti a les necessitats dinàmiques del projecte, considerant factors com el temps, el pressupost, i la complexitat del software. Paral·lelament, es durà a terme una síntesi detallada dels requisits funcionals i no funcionals, així com de les funcionalitats clau identificades a partir de la informació recopilada, assegurant que la base del desenvolupament estigui alineada amb les expectatives i necessitats reals dels usuaris finals.

5.2.3. Procés de desenvolupament

La implementació del software es guiarà per l'aplicació de metodologies àgils [12], com *Scrum* o *Kanban*. Aquest enfocament facilitarà la flexibilitat i adaptabilitat necessàries en el desenvolupament, permetent la iteració ràpida i l'ajust continu del producte basat en el feedback constant dels usuaris i l'equip de desenvolupament. La utilització de *sprints* o iteracions planificades permetrà al equip prioritzar funcionalitats segons la seva importància i impacte, així com identificar i resoldre problemes de manera eficaç a mesura que sorgeixen.

6. Desenvolupament

6.1. Metodologia de desenvolupament

Per al desenvolupament d'aquest projecte, s'ha seguit la metodologia *Agile*, que ens ha permès treballar de manera iterativa i incremental. Aquesta metodologia ha estat escollida perquè facilita l'adaptació als canvis i permet una millor comunicació i col·laboració amb els *stakeholders*, especialment amb la tutora del treball de fi de grau (TFG).

6.1.1. Etapes del Projecte

Etapa 1. Planificació Inicial

Definir els objectius del projecte, establir els requeriments inicials i preparar el pla de treball.

- **Activitats:**
 - Reunió inicial amb la tutora per discutir la visió i abast del projecte.
 - Identificació dels requeriments funcionals i no funcionals.
 - Elaboració del *backlog* inicial.

Etapa 2. Disseny de l'Arquitectura

Crear una arquitectura robusta que suporti els requeriments del projecte.

- **Activitats:**
 - Disseny de l'estructura de fitxers i organització del codi.
 - Definició dels models de dades i repositoris.
 - Elaboració de maquetes de les pàgines principals.

Etapa 3. Desenvolupament Iteratiu

Desenvolupar el projecte en iteracions curtes, lliurant increments funcionals en cada iteració.

- **Activitats:**
 - Implementació de funcionalitats prioritzades en el *backlog*.
 - Reunions setmanals amb la tutora per revisar el progrés i rebre feedback.
 - Tests i ajustaments continus basats en el feedback rebut.

Etapa 4. Integració i Proves

Assegurar que totes les funcionalitats treballin correctament conjuntament i que el sistema compleixi amb els requeriments definits.

- **Activitats:**
 - Integració de tots els components desenvolupats.
 - Realització de proves unitàries, d'integració i de sistema.
 - Correcció d'errors detectats durant les proves.

Etapa 5. Lliurament Final i Documentació

Preparar el lliurament final del projecte i la documentació necessària.

- **Activitats:**
 - Preparació del lliurament final amb totes les funcionalitats integrades.
 - Elaboració de la documentació del projecte (manual d'usuari, documentació tècnica, etc.).
 - Presentació final a la tutora i preparació de la defensa del TFG.

6.1.2. Reunions amb la Tutora del TFG

Les reunions amb la tutora han estat un component clau per assegurar que el projecte segueixi el camí correcte. Aquestes reunions han estat programades setmanalment i han seguit la següent estructura:

- **Reunió Inicial:** Discussió sobre els objectius del projecte, establiment del pla de treball i definició del *backlog* o treball pendent inicial.
- **Reunions Quinzenals:**
 - **Revisió del Progrés:** Discussió sobre les funcionalitats desenvolupades durant les últimes dues setmanes.
 - **Feedback:** Recepció de comentaris i suggeriments per part de la tutora.
 - **Planificació de la Setmana:** Definició de les tasques per a la següent iteració.
- **Reunió de Revisió Pre-tancament:** Avaluació dels increments de programari acabats, discussió sobre possibles millores i ajustaments.
- **Reunió de Tancament:** Revisió del projecte complet, discussió sobre la documentació final i preparació per a la defensa del TFG.

6.2. Requeriments funcionals i no funcionals:

Fer aquest estudi de requisits ha servit per entendre clarament les necessitats i expectatives dels usuaris finals [13], assegurar que el sistema desenvolupat compleixi amb els objectius del projecte. A més, permetre identificar les funcionalitats crítiques i no funcionals que garanteixen una experiència d'usuari satisfactòria, facilitant així la planificació i execució eficient del desenvolupament del projecte.

Requeriments funcionals:

- El sistema ha de permetre el registre de proveïdors.
- El sistema ha de permetre el registre de productes associats a proveïdors.
- El sistema ha de permetre la modificació de la informació dels proveïdors.
- El sistema ha de permetre la modificació de la informació dels productes.
- El sistema ha de permetre la desactivació de proveïdors.
- El sistema ha de permetre la desactivació de productes.
- El sistema ha de permetre actualitzar l'estoc un cop rebuda la comanda.
- El sistema ha de permetre la modificació de les entrades d'estoc.
- El sistema ha de permetre la modificació de les sortides d'estoc.
- El sistema ha de permetre la creació de comandes als proveïdors.
- El sistema ha de permetre el seguiment de l'estat de les comandes.
- El sistema ha de permetre la confirmació de les comandes rebudes.
- El sistema ha de permetre visualitzar la comanda per a un seguiment detallat.
- El sistema ha de generar informes de comandes, tendències de compra i altres *KPIs* per a la presa de decisions estratègiques.

Requeriments no funcionals:

- El sistema ha de ser accessible en sistemes mòbil tant *IOS* com *Android*.
- El sistema ha de tenir una interfície d'usuari intuïtiva per facilitar la seva utilització per part dels usuaris finals.
- El sistema ha de tenir una interfície d'usuari adaptable (*responsive*) per a ser utilitzat en diferents dispositius i mides de pantalla.
- El sistema ha de proporcionar funcionalitats fora de línia en cas de perdre la connexió a internet, assegurant així la continuïtat operativa.

- El sistema ha de tenir mecanismes per a la sincronització automàtica i segura de dades entre la base de dades local.

Requeriments Addicionals:

- El sistema ha de ser fàcil de mantenir i actualitzar, permetent la incorporació de noves funcionalitats segons les necessitats canviants de les pastisseries.
- El sistema ha de complir amb les normatives de protecció de dades (GDPR) per garantir la privacitat i seguretat de la informació dels usuaris.
- El sistema ha de proporcionar un alt rendiment, assegurant temps de resposta ràpids per a les operacions més crítiques.

6.3. Casos d'ús

Primerament, s'ha creat un diagrama de casos d'ús, el qual defineix les interaccions dels usuaris amb el sistema. Aquest diagrama destaca les funcions clau com l'afegiment, modificació i eliminació de proveïdors i productes, així com la realització de comandes i la consulta d'estadístiques i estoc.

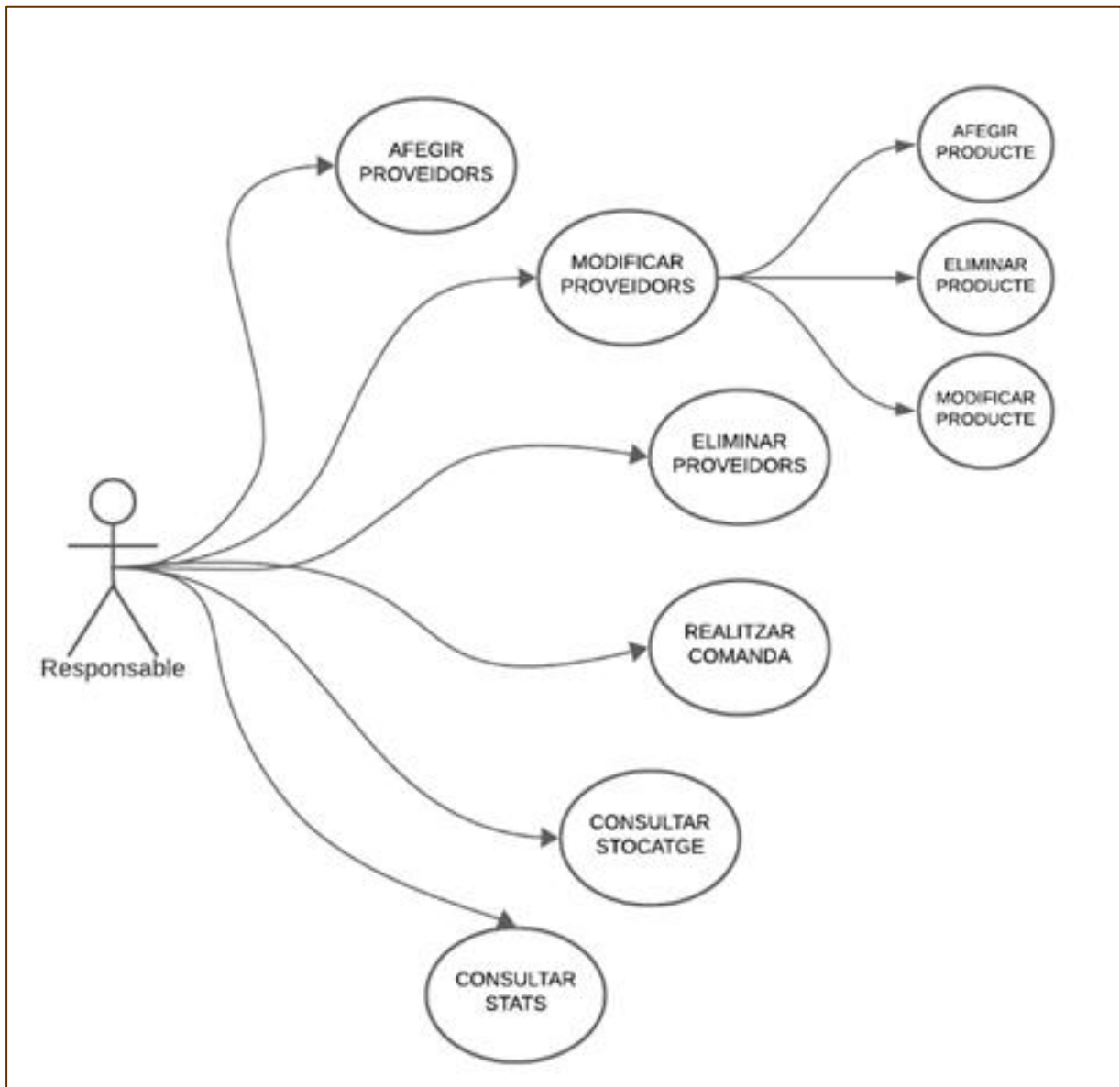


Fig. 6.3.1. Diagrama de casos d'us

6.3.1. Registre de Proveïdors

Descripció:

Aquest cas d'ús permet al responsable registrar nous proveïdors al sistema introduint la seva informació rellevant com nom, CIF, adreça fiscal i dades de contacte.

Actor: Usuari Responsable

Pre-condicions:

- El responsable ha d'estar autenticat amb el compte de la empresa.

Post-condicions:

- El nou proveïdor queda registrat al sistema amb la seva informació completa.
- La informació del proveïdor està disponible per a consultes i operacions posteriors.

Flux normal d'execució:

1. El responsable inicia sessió al sistema.
2. El responsable accedeix a la secció de "Proveïdors".
3. El responsable selecciona clica al botó de "Afegir Proveïdor".
4. El sistema mostra un formulari per a la introducció de dades del nou proveïdor.
5. El responsable omple el formulari amb la informació necessària (nom, CIF, adreça fiscal, dades de contacte).
6. El responsable prem el botó "Guardar".
7. El sistema valida la informació introduïda i emmagatzema el nou proveïdor a la base de dades.

Flux alternatiu d'execució:

- Si la informació introduïda és incompleta o incorrecta, el sistema mostra un missatge d'error indicant les dades que cal corregir o completar. El responsable corregeix la informació i torna a intentar guardar-la.

6.3.2. Desactivació d'un proveïdor

Descripció:

Aquest cas d'ús permet al responsable desactivar o “eliminar” un proveïdor existent del sistema, desactivant el seu registre i impedit futures operacions amb aquest proveïdor.

Actor:

Usuari responsable

Pre-condicions:

- El responsable ha d'estar autenticat amb el compte de l'empresa.
- El proveïdor ha d'estar prèviament registrat al sistema.

Post-condicions:

- El proveïdor seleccionat queda desactivat al sistema.
- La informació del proveïdor no estarà disponible per a futures operacions, però es mantindrà per a referència històrica.

Flux normal d'execució:

1. El responsable inicia sessió al sistema.
2. El responsable accedeix a la secció de "Proveïdors".
3. El responsable llisca a la esquerra el proveïdor que desitja eliminar.
4. El responsable clica al botó vermell amb la icona de la paperera.
5. El sistema mostra una confirmació per assegurar-se que el responsable desitja eliminar el proveïdor seleccionat.
6. El responsable confirma l'eliminació.
7. El sistema desactiva el proveïdor i actualitza la base de dades.

Flux alternatiu d'execució:

- Si el responsable decideix no eliminar el proveïdor durant la confirmació, el sistema cancel·la l'operació i manté el registre del proveïdor sense canvis.

6.3.3. Modificació d'un proveïdor

Descripció:

Aquest cas d'ús permet al responsable modificar la informació dels proveïdors existents al sistema.

Actor:

Usuari responsable

Pre-condicions:

- L'usuari responsable ha d'estar autenticat amb el compte de l'empresa.
- El proveïdor ha d'estar prèviament registrat al sistema.

Post-condicions:

- La informació del proveïdor seleccionat queda actualitzada al sistema.
- Les modificacions estan disponibles per a operacions posteriors.

Flux normal d'execució:

1. L'usuari responsable inicia sessió al sistema.
2. L'usuari responsable accedeix a la secció de "Proveïdors".
3. L'usuari responsable llisca a la esquerra el proveïdor que desitja modificar.
4. L'usuari prem sobre el boto verd amb la icona de modificar dades.
5. El sistema mostra un formulari amb la informació actual del proveïdor.
6. L'usuari responsable modifica els camps necessaris (nom, CIF, adreça fiscal, dades de contacte).
7. L'usuari responsable prem el botó "Guardar".
8. El sistema valida les modificacions i actualitza la informació del proveïdor a la base de dades.

Flux alternatiu d'execució:

- Si la informació modificada és incompleta o incorrecta, el sistema mostra un missatge d'error indicant les dades que cal corregir. L'usuari responsable corregeix la informació i torna a intentar guardar-la.

6.3.4. Creació d'un Producte o Ítem d'un Proveïdor

Descripció:

Aquest cas d'ús permet al responsable crear un nou producte o ítem associat a un proveïdor al sistema introduint la informació necessària del producte.

Actor:

Usuari responsable

Pre-condicions:

- El responsable ha d'estar autenticat amb el compte de l'empresa.
- El proveïdor ha d'estar prèviament registrat al sistema.

Post-condicions:

- El nou producte o ítem queda registrat al sistema amb la seva informació completa.
- La informació del producte està disponible per a consultes i operacions posteriors.

Flux normal d'execució:

1. El responsable inicia sessió al sistema.
2. El responsable accedeix a la secció de "Proveïdors".
3. El responsable llisca a la esquerra un proveïdor i selecciona el boto de afegir un nou producte.
4. El sistema mostra un formulari per a la introducció de dades del nou producte i la llista de productes ja registrats per aquell proveïdor.
5. El responsable omple el formulari amb la informació necessària (nom del producte, preu, quantitat de lot, categoria, format).
6. El responsable prem el botó "Guardar".
7. El sistema valida la informació introduïda i emmagatzema el nou producte a la base de dades.

Flux alternatiu d'execució:

- Si la informació introduïda és incompleta o incorrecta, el sistema mostra un missatge d'error indicant les dades que cal corregir o completar. El responsable corregeix la informació i torna a intentar guardar-la.

6.3.5. Creació d'una comanda a un proveïdor

Descripció:

Aquest cas d'ús permet al responsable crear una nova comanda seleccionant un proveïdor i especificant les quantitats necessàries dels productes registrats amb aquell proveïdor.

Actor:

Usuari responsable

Pre-condicions:

- L'usuari responsable ha d'estar autenticat amb el compte de l'empresa.
- Han d'existir proveïdors i productes prèviament registrats al sistema.

Post-condicions:

- La nova comanda queda registrada al sistema amb la informació completa.
- La comanda està disponible per a seguiment i confirmació posteriors.

Flux normal d'execució:

1. L'usuari responsable inicia sessió al sistema.
2. L'usuari responsable accedeix a la secció de "Comandes".
3. El sistema mostra la llista de comandes actuals.
4. L'usuari responsable clica el botó inferior a l'esquerra "Crear Nova Comanda".
5. El sistema mostra una llista de proveïdors disponibles.
6. L'usuari responsable selecciona el proveïdor desitjat.
7. El sistema mostra una llista de productes registrats per a aquell proveïdor.
8. L'usuari responsable omple les quantitats necessàries per als productes seleccionats.
9. L'usuari responsable prem el botó "Guardar".
10. El sistema valida la informació introduïda i emmagatzema la nova comanda a la base de dades.

Flux alternatiu d'execució:

- Si no s'omple amb cap valor cap producte, el sistema mostra un missatge d'error indicant que com a mínim s'ha de omplir la quantitat d'una casella per a realitzar una comanda. L'usuari responsable corregeix la informació i torna a intentar guardar-la.

6.3.6. Consulta d'una comanda**Descripció:**

Aquest cas d'ús permet al responsable consultar els detalls d'una comanda existent clicant sobre la comanda en la llista de comandes.

Actor:

Usuari: Responsable

Pre-condicions:

- L'usuari responsable ha d'estar autenticat amb el compte de l'empresa.
- La comanda ha d'estar prèviament registrada al sistema.

Post-condicions:

- Els detalls de la comanda seleccionada es mostren a l'usuari responsable.

Flux normal d'execució:

1. L'usuari responsable inicia sessió al sistema.
2. L'usuari responsable accedeix a la secció de "Comandes".
3. El sistema mostra la llista de comandes actuals.
4. L'usuari responsable clica sobre la comanda que vol consultar.
5. El sistema mostra els detalls de la comanda seleccionada, incloent el proveïdor i les seves dades, els productes, les quantitats, el preu total i opcions d'enviament.

6.3.7. Eliminació d'una comanda

Descripció:

Aquest cas d'ús permet al responsable eliminar una comanda existent del sistema lliscant cap a l'esquerra sobre la comanda i confirmant la seva eliminació.

Actor:

Usuari responsable

Pre-condicions:

- L'usuari responsable ha d'estar autenticat amb el compte de l'empresa.
- La comanda ha d'estar prèviament registrada al sistema.

Post-condicions:

- La comanda seleccionada queda eliminada del sistema.
- La comanda eliminada ja no està disponible per a consultes o operacions posteriors.

Flux normal d'execució:

1. L'usuari responsable inicia sessió al sistema.
2. L'usuari responsable accedeix a la secció de "Comandes".
3. El sistema mostra la llista de comandes actuals.
4. L'usuari responsable llisca cap a l'esquerra sobre la comanda que vol eliminar.
5. Apareix un botó vermell amb la icona de la paperera.
6. L'usuari responsable clica al botó vermell.
7. El sistema mostra una confirmació per assegurar-se que el responsable desitja eliminar la comanda seleccionada.
8. L'usuari responsable confirma l'eliminació.
9. El sistema elimina la comanda de la base de dades i actualitza la llista de comandes.

Flux alternatiu d'execució:

- Si l'usuari responsable decideix no eliminar la comanda durant la confirmació, el sistema cancel·la l'operació i manté la comanda sense canvis.

6.3.8. Enviament d'una comanda al proveïdor**Descripció:**

Aquest cas d'ús permet al responsable enviar una comanda al proveïdor mitjançant correu electrònic o una altra aplicació després de crear-la o des de la pestanya de consulta de la comanda.

Actor:

Usuari: Responsable

Pre-condicions:

- L'usuari responsable ha d'estar autenticat amb el compte de l'empresa.
- La comanda ha d'estar prèviament registrada al sistema.

Post-condicions:

- La comanda es remet al proveïdor seleccionat mitjançant correu electrònic o una altra aplicació.
- Es registra l'enviament de la comanda per a seguiment i auditoria.

Flux normal d'execució:**1. Enviament immediat després de crear la comanda:**

- 1.1. L'usuari responsable inicia sessió al sistema.
- 1.2. L'usuari responsable accedeix a la secció de "Comandes".
- 1.3. L'usuari responsable clica al botó "Crear Nova Comanda".
- 1.4. L'usuari responsable selecciona el proveïdor i omple les quantitats necessàries per als productes.
- 1.5. L'usuari responsable prem el botó "Guardar".
- 1.6. El sistema valida i emmagatzema la nova comanda.

1.7. El sistema mostra una pestanya amb l'opció d'enviar la comanda per correu electrònic o una altra aplicació.

1.8. L'usuari responsable selecciona l'opció d'enviament desitjada.

1.9. El sistema envia la comanda al proveïdor mitjançant l'aplicació seleccionada.

2. Enviament des de la pestanya de consulta de la comanda:

2.1. L'usuari responsable inicia sessió al sistema.

2.2. L'usuari responsable accedeix a la secció de "Comandes".

2.3. El sistema mostra la llista de comandes actuals

2.4. L'usuari responsable clica sobre la comanda que vol consultar.

2.5. El sistema mostra els detalls de la comanda seleccionada.

2.6. L'usuari responsable prem el botó "Enviar Comanda".

2.7. El sistema mostra les opcions d'enviament (correu electrònic o una altra aplicació).

2.8. L'usuari responsable selecciona l'opció d'enviament desitjada.

2.9. El sistema envia la comanda al proveïdor mitjançant l'aplicació seleccionada.

6.3.9. Confirmació de recepció de comanda

Descripció:

Aquest cas d'ús permet al responsable confirmar la recepció d'una comanda. Després de confirmar la comanda, el sistema actualitza els números d'estoc registrats.

Actor:

Usuari: Responsable

Pre-condicions:

- L'usuari responsable ha d'estar autenticat amb el compte de l'empresa.
- La comanda ha d'estar prèviament registrada al sistema i en estat pendent de recepció.

Post-condicions:

- La comanda es marca com a rebuda i no es pot tornar a modificar.
- Els números d'estoc s'actualitzen automàticament al sistema.

Flux normal d'execució:

1. L'usuari responsable inicia sessió al sistema.
2. L'usuari responsable accedeix a la secció de "Comandes".
3. El sistema mostra la llista de comandes actuals.
4. A la casella de la comanda pendent de recepció, l'usuari responsable clica el botó "Marcar com a Rebut".
5. El sistema canvia el color del botó a gris, impossibilitant tornar a clicar-lo i marcant la comanda com a rebuda.
6. El sistema actualitza els números d'estoc registrats per reflectir la recepció de la comanda.

6.4. Diagrames de seqüència

S'ha considerat realitzar els diagrames de seqüència per als casos d'ús més crítics que cobreixen les principals funcionalitats del sistema.

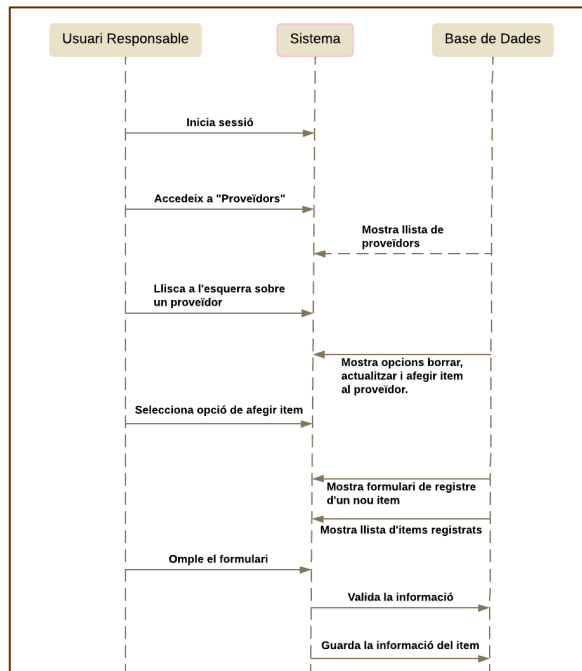


Fig. 6.4.1 Diagrama de seqüència per a la creació d'un producte o ítem d'un proveïdor

El que es fa aquí és bàsicament, un cop l'usuari accedeix a l'aplicació, clicar sobre el botó de proveïdors. Aquesta pàgina té una llista dinàmica que s'omple a partir dels elements que tingui la llista de proveïdors en el repositori de proveïdors. Un cop la llista mostra els proveïdors, l'usuari pot dur a terme tres accions arrossegant cap a l'esquerra algun d'ells: eliminar el proveïdor, que desactiva el paràmetre *'isActive'* del proveïdor; modificar dades del proveïdor; i, per últim, afegir un *ítem* al proveïdor. En clicar sobre la icona d'afegir *ítem*, es desplega una vinyeta que mostra, per una part, els *ítems* ja registrats d'aquell proveïdor i, per sota, el formulari per omplir-ne un de nou. Un cop omplert el formulari, es crea un nou *ítem* que s'afegeix al repositori d'*ítems*.

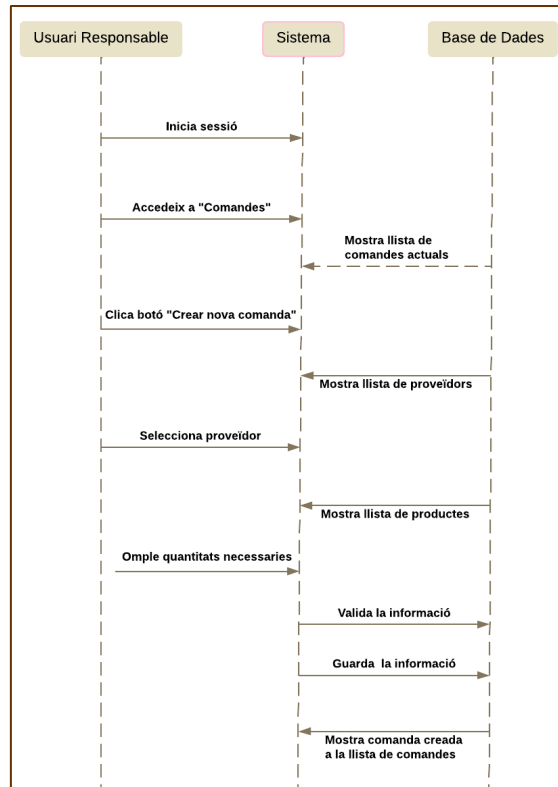


Fig. 6.4.2. Diagrama de seqüència per a la creació d'una comanda

El procés de creació d'una comanda comença quan l'usuari responsable accedeix a l'aplicació. Un cop dins, accedeix a la secció de "Comandes", on se li mostra una llista de les comandes actuals obtingudes del repositori de comandes. A continuació, l'usuari clica el botó "Crear nova comanda", i el sistema torna a recopilar els proveïdors registrats i li mostra la llista dels proveïdors disponibles. Després de seleccionar un proveïdor, el sistema mostra una llista de productes associats a aquest proveïdor. L'usuari omple les quantitats necessàries per a cada producte, i el sistema valida aquesta informació. Un cop validada la nova comanda es creada i es mostra a la llista de comandes, completant així el procés.

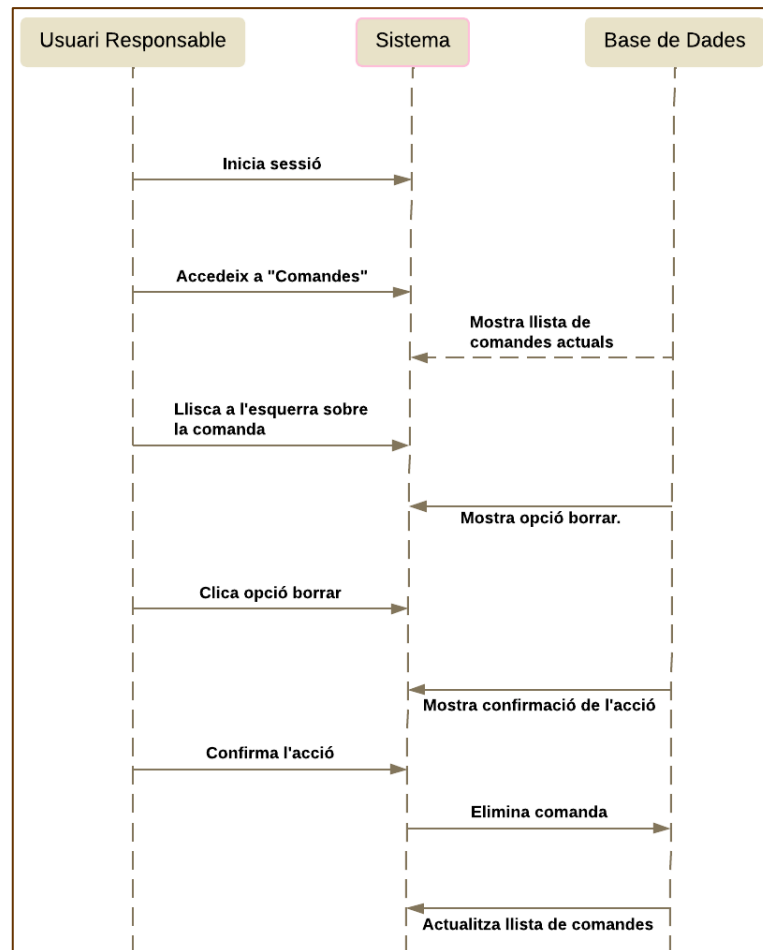


Fig. 6.4.3. Diagrama de seqüència per a l'eliminació d'una comanda

El procés d'eliminació d'una comanda comença quan l'usuari responsable accedeix a l'aplicació. Un cop dins, accedeix a la secció de "Comandes", on se li mostra una llista de les comandes actuals obtingudes del repositori de comandes. L'usuari llisca a l'esquerra sobre la comanda que desitja eliminar, i el sistema mostra la icona de "esborrar". L'usuari clica sobre aquesta opció, moment en què el sistema demana confirmació de l'acció. Després que l'usuari confirma l'acció, el sistema elimina la comanda del repositori i actualitza la llista de comandes mostrant les comandes restants.

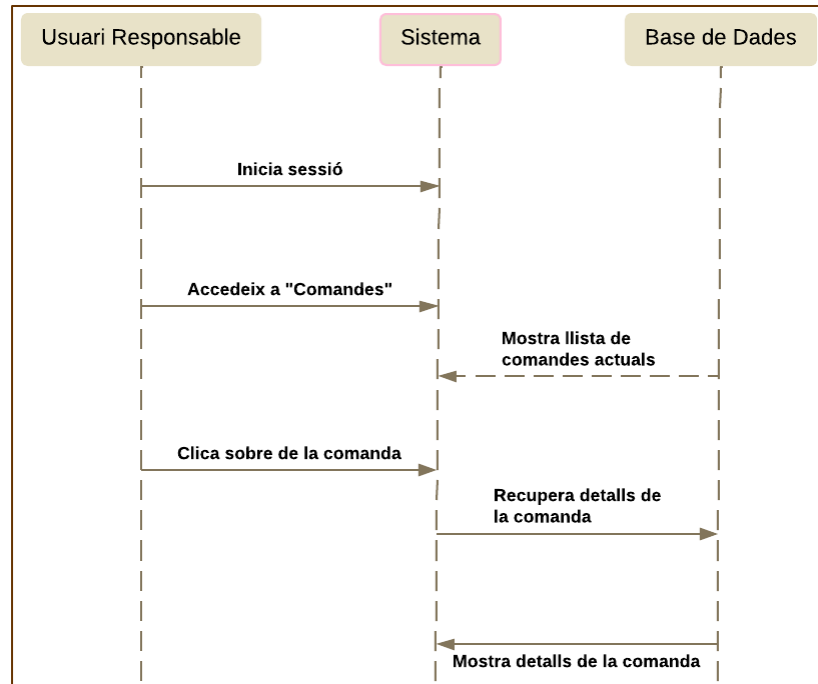


Fig. 6.4.4. Diagrama de seqüència per a la consulta d'una comanda

Un cop l'usuari accedeix a l'aplicació i a la secció de "Comandes", se li mostra una llista de les comandes actuals obtingudes de la base de dades. L'usuari clica sobre la comanda que vol consultar, i el sistema crida al repositori de comandes per obtenir la comanda i poder recuperar els detalls d'aquesta comanda específica. Un cop obtinguda la informació, el sistema mostra els detalls de la comanda a l'usuari.

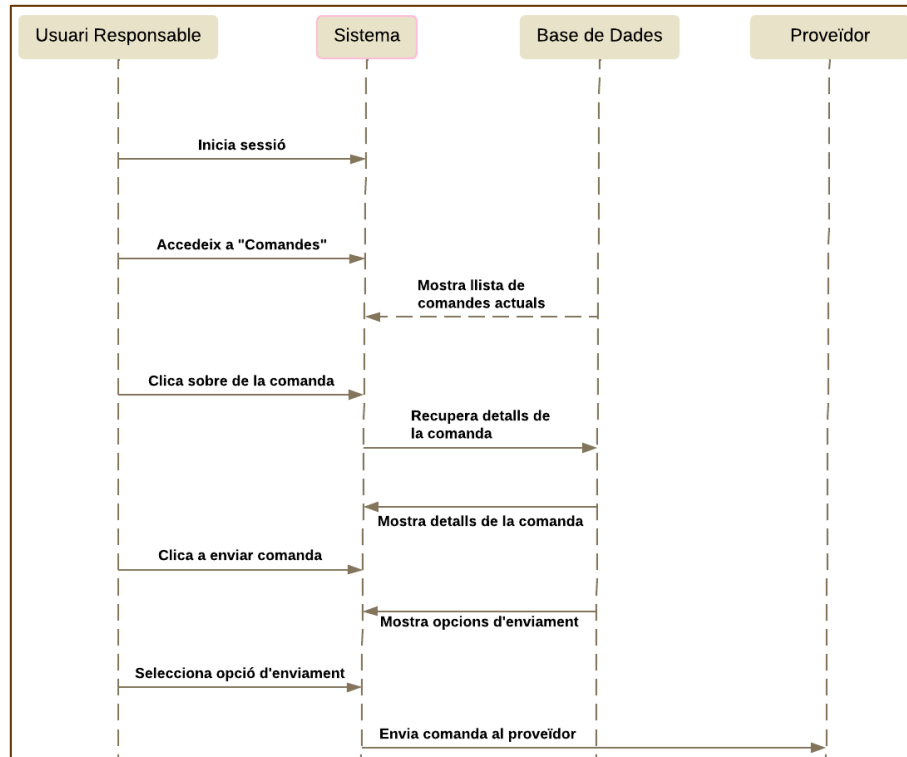


Fig. 6.4.5. Diagrama de seqüència per a l'enviament d'una comanda al proveïdor

Un cop l'usuari accedeix a la aplicació i a la secció de "Comandes", se li mostra una llista de les comandes actuals obtingudes del repositori de comandes. L'usuari clica sobre la comanda que vol enviar, i el sistema envia una sol·licitud a la base de dades per obtenir la comanda i poder recuperar els detalls d'aquesta comanda específica. Un cop obtinguda la informació, el sistema mostra els detalls de la comanda a l'usuari. L'usuari clica l'opció d'enviar comanda, moment en què el sistema mostra les opcions d'enviament disponibles. Després que l'usuari selecciona una opció d'enviament, el sistema crea i formata un cadena de valors amb la informació de la comanda que serà el que a posteriori, s'envia al proveïdor.

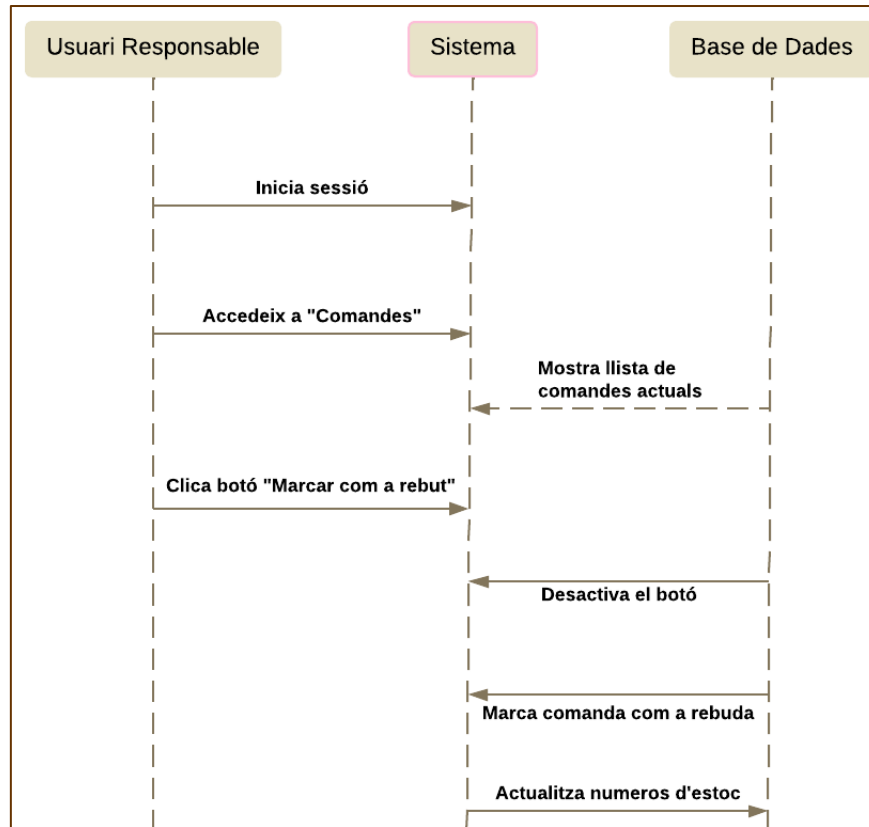


Fig. 6.4.6. Diagrama de seqüència per a la confirmació de recepció d'una comanda

El procés de confirmació de recepció d'una comanda comença quan l'usuari responsable accedeix a l'aplicació i un cop dins, accedeix a la secció de "Comandes", on se li mostra una llista de les comandes actuals obtingudes del repositori de comandes. Les comandes tenen un paràmetre booleà que permeten identificar si una comanda ha estat rebuda o no, i per a cada comanda hi ha un botó que permet confirmar la recepció d'aquesta. Així doncs l'usuari clica el botó "Marcar com a rebut" per confirmar que ha rebut una comanda específica. El sistema, a continuació, desactiva el botó per evitar múltiples confirmacions per la mateixa comanda. Després, el sistema marca la comanda com a rebuda al repositori de comandes i actualitza els números d'estoc per reflectir la recepció de la mercaderia. Finalment, el sistema actualitza la llista de comandes per mostrar l'estat actualitzat de les comandes.

6.5. Diagrama de classes

Seguidament es mostra un diagrama de classes simplificat que se centra en les classes més essencials que expliquen el funcionament bàsic de l'aplicació i s'ometen les classes que no siguin crítiques per a la comprensió del sistema.

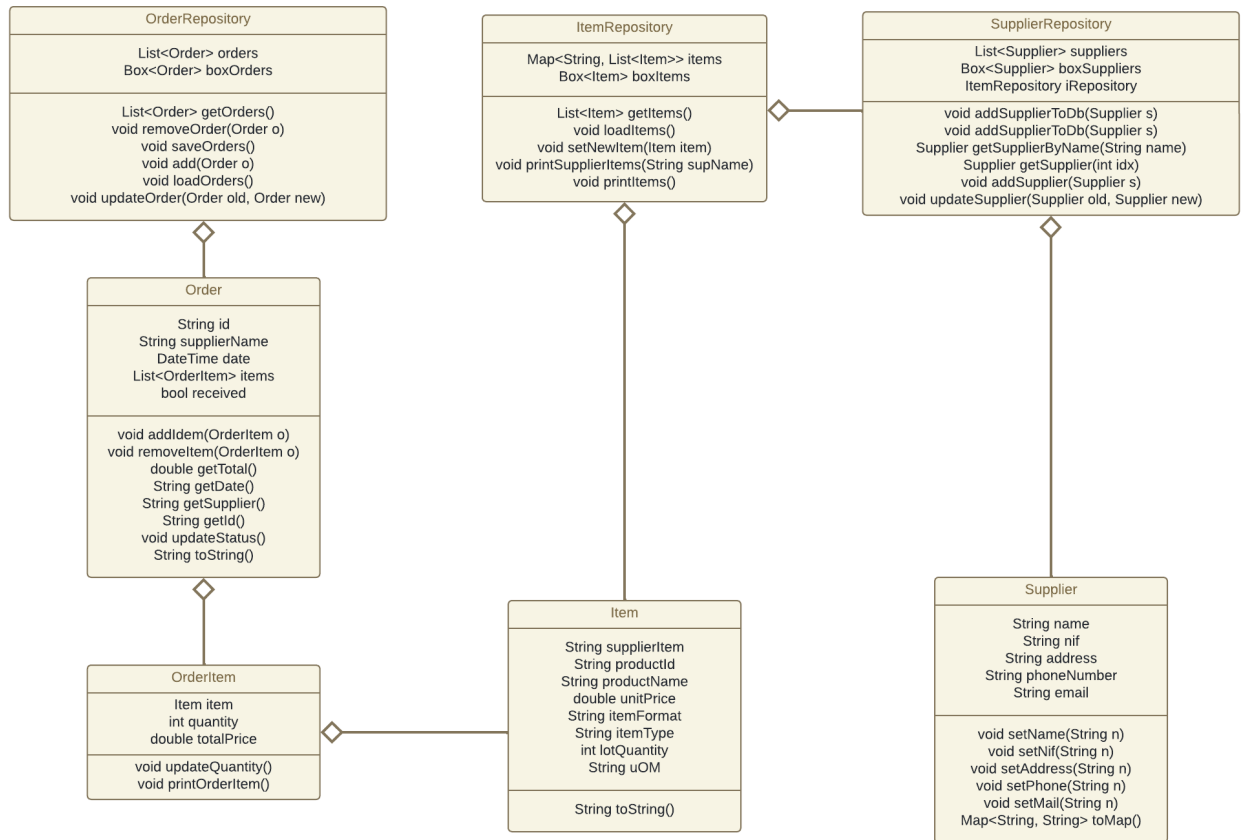


Fig. 6.5.1. Diagrama de classes

El diagrama de classes mostra les relacions entre les classes principals de l'aplicació. *Order* representa una comanda i està composta per diversos *OrderItem*, que al seu torn contenen *Item*, que són els productes. *Supplier* representa els proveïdors i és gestionat per *SupplierRepository*, que també depèn de *ItemRepository* per gestionar els productes associats. *OrderRepository* gestiona les comandes, i *ItemRepository* gestiona els productes. Aquestes relacions mostren com les classes interactuen per gestionar la informació de comandes, productes i proveïdors a l'aplicació.

6.6. Diagrama de base de dades

6.6.1. Què és Hive

Hive és una base de dades *NoSQL* local per a *Flutter* [14], dissenyada per ser ràpida, segura i fàcil d'utilitzar. És ideal per a emmagatzemar dades de manera persistent en aplicacions mòbils, ja que no requereix esquemes de base de dades complexos ni configuracions addicionals. *Hive* utilitza fitxers binaris per emmagatzemar les dades, cosa que el fa molt eficient en termes de velocitat i espai d'emmagatzematge.

6.6.2. Característiques de *Hive*

- Rendiment Alt: *Hive* està optimitzat per a la velocitat i pot gestionar operacions de lectura i escriptura molt ràpides.
- Tipus Segur: Utilitza adaptadors de tipus per garantir que les dades es desin de manera segura i correcta.
- Facilitat d'Ús: Té una *API* simple i fàcil d'utilitzar que fa que la integració amb *Flutter* sigui molt directa.
- Suport per a Objectes Complexos: Amb *Hive*, es poden desar objectes complexos amb facilitat utilitzant adaptadors personalitzats.

6.6.3. Integració de *Hive* en el Projecte

L'ús de *Hive* en aquest projecte ha permès gestionar les dades de manera eficient i estructurada, proporcionant un rendiment alt i una integració senzilla amb *Flutter*. L'estructura de dades basada en caixes (*boxes*) facilita la gestió de les entitats clau, mantenint el codi net i organitzat.

6.6.4. Estructura de dades en Hive

L'estructura de dades del projecte es gestiona mitjançant diversos repositoris que interactuen amb les caixes (boxes) de Hive per emmagatzemar i recuperar dades. A continuació es mostra un diagrama UML que il·lustra com s'estructuren les dades dins el programa:

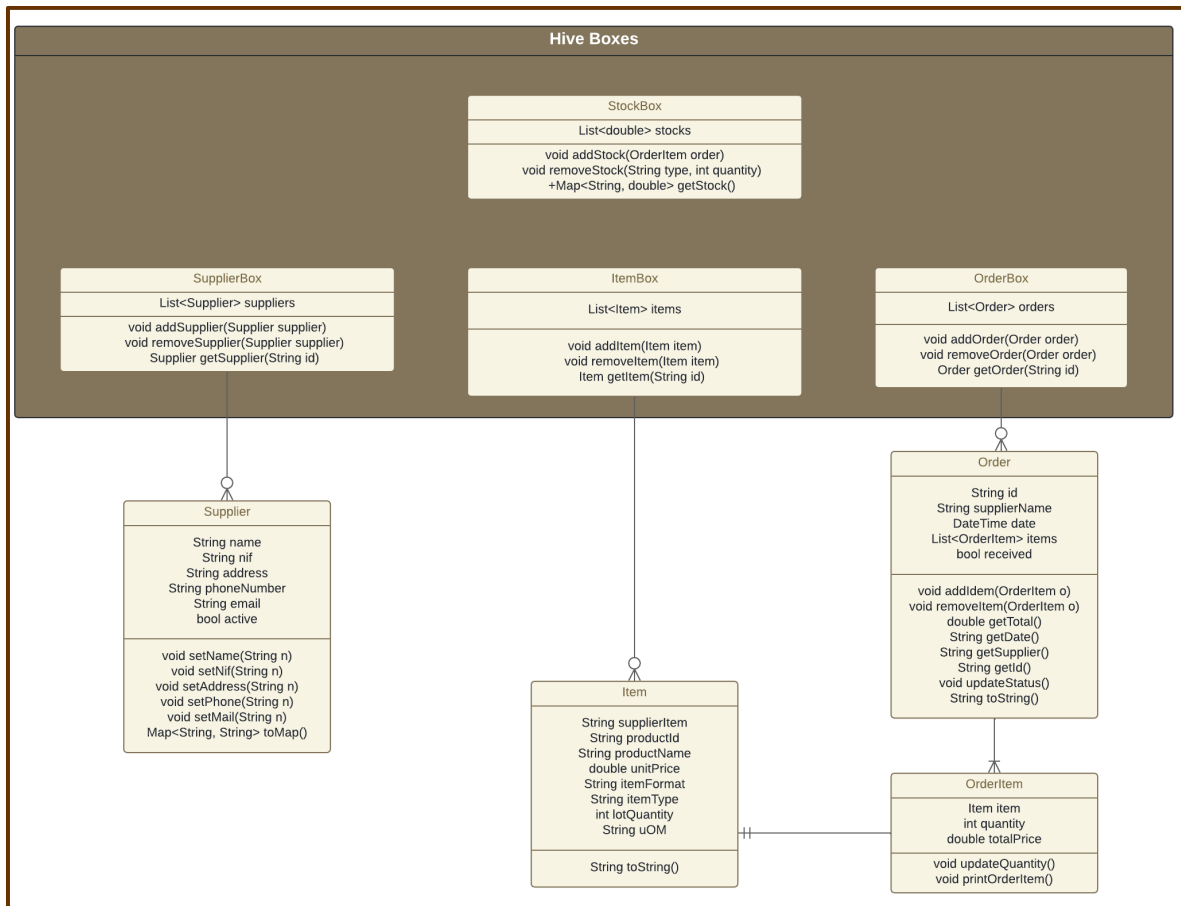


Fig. 6.6.4.1. Diagrama d'estructura de dades

ItemBox conté una llista de tots els productes, proporcionant mètodes per afegir, eliminar i obtenir productes i *OrderBox* emmagatzema totes les comandes amb mètodes similars per gestionar-les. *SupplierBox* manté una llista de tots els proveïdors, permetent operacions d'afegir, eliminar i obtenir proveïdors. *StockBox* registra les quantitats d'estoc per a cada tipus de producte, amb funcions per gestionar aquests valors. La classe *Supplier* representa un proveïdor amb atributs com ID, nom, NIF i adreça. *Item* representa un producte amb atributs com ID, nom, preu i ID del proveïdor. *Order* representa una comanda que conté una llista d'ítems, amb informació sobre el proveïdor, data i estat de recepció.

Finalment, *OrderItem* representa un ítem dins d'una comanda, referenciant un producte específic amb atributs com ID del producte, quantitat i preu total.

6.7. Disseny de la interfície d'usuari (UI)

El disseny de la interfície d'usuari es va realitzar seguint un procés estructurat per garantir que l'aplicació sigui intuïtiva i fàcil d'utilitzar. El procés va incloure les següents fases:

6.7.1. Wireframes

Es van crear esbossos preliminars de cada pantalla de l'aplicació per visualitzar la distribució d'elements i l'estructura general. Seguidament veiem alguns dels esbossos que es van dur a terme:

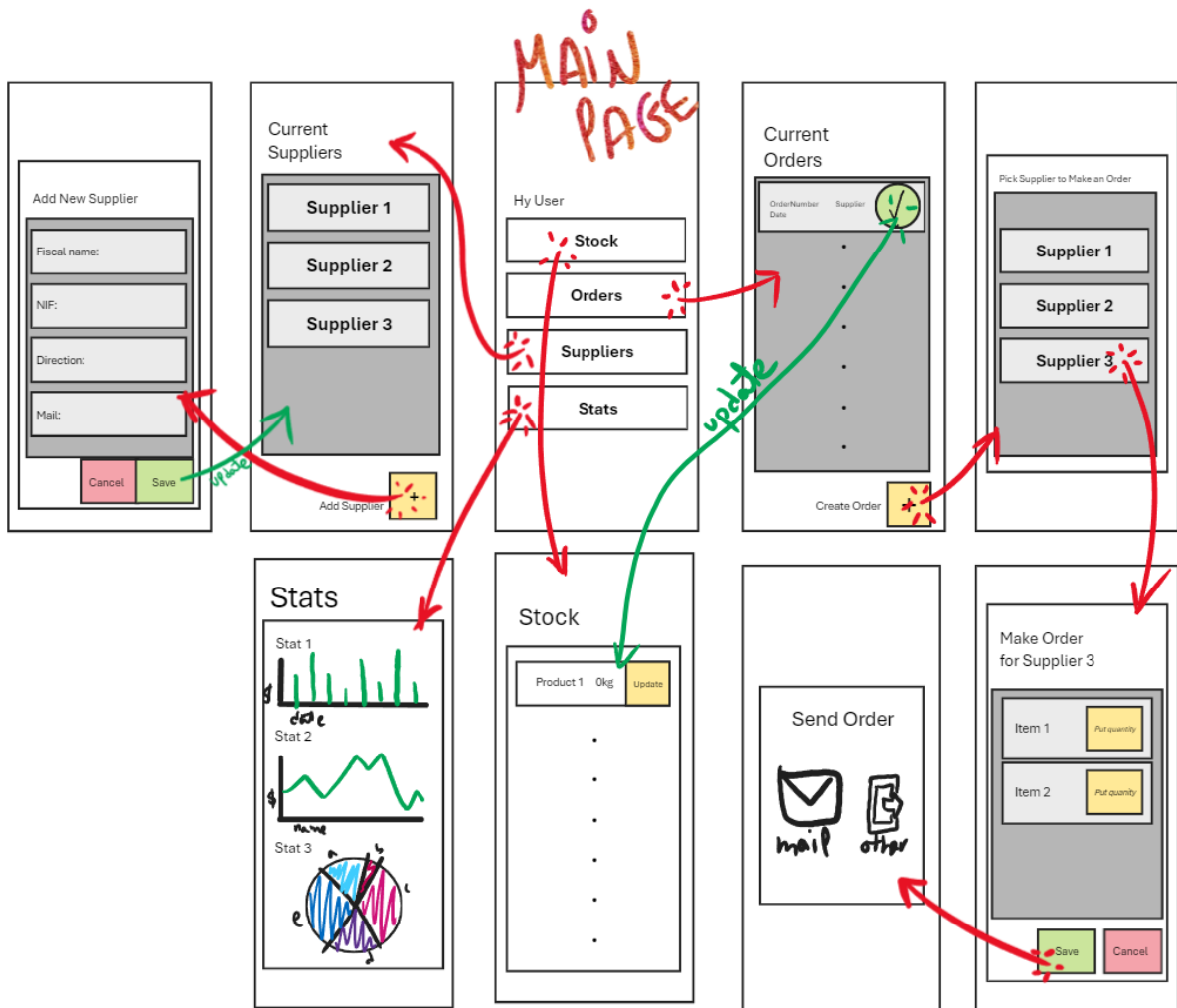


Fig. 6.7.1.1. Wireframe aplicació

6.7.2. Prototips

Un cop aprovats els *wireframes*, es van desenvolupar prototips interactius per simular el flux de l'aplicació. Aquests prototips van ser fonamentals per provar la navegació entre pantalles i assegurar-se que l'experiència d'usuari fos fluïda.

Aquests prototips es van dissenyar i desenvolupar primerament amb *Figma*, i això ha permès fer iteracions ràpides basades amb els comentaris de usuaris que han pogut tenir una visió preliminar de la interfície, millorant l'eficiència del procés de disseny. A mesura que el projecte avançava, es van anar modificant i canviant aspectes del disseny original. El resultat final de l'aplicació és diferent tant dels esbossos inicials com dels prototips, reflectint les millores i ajustos fets al llarg del desenvolupament.

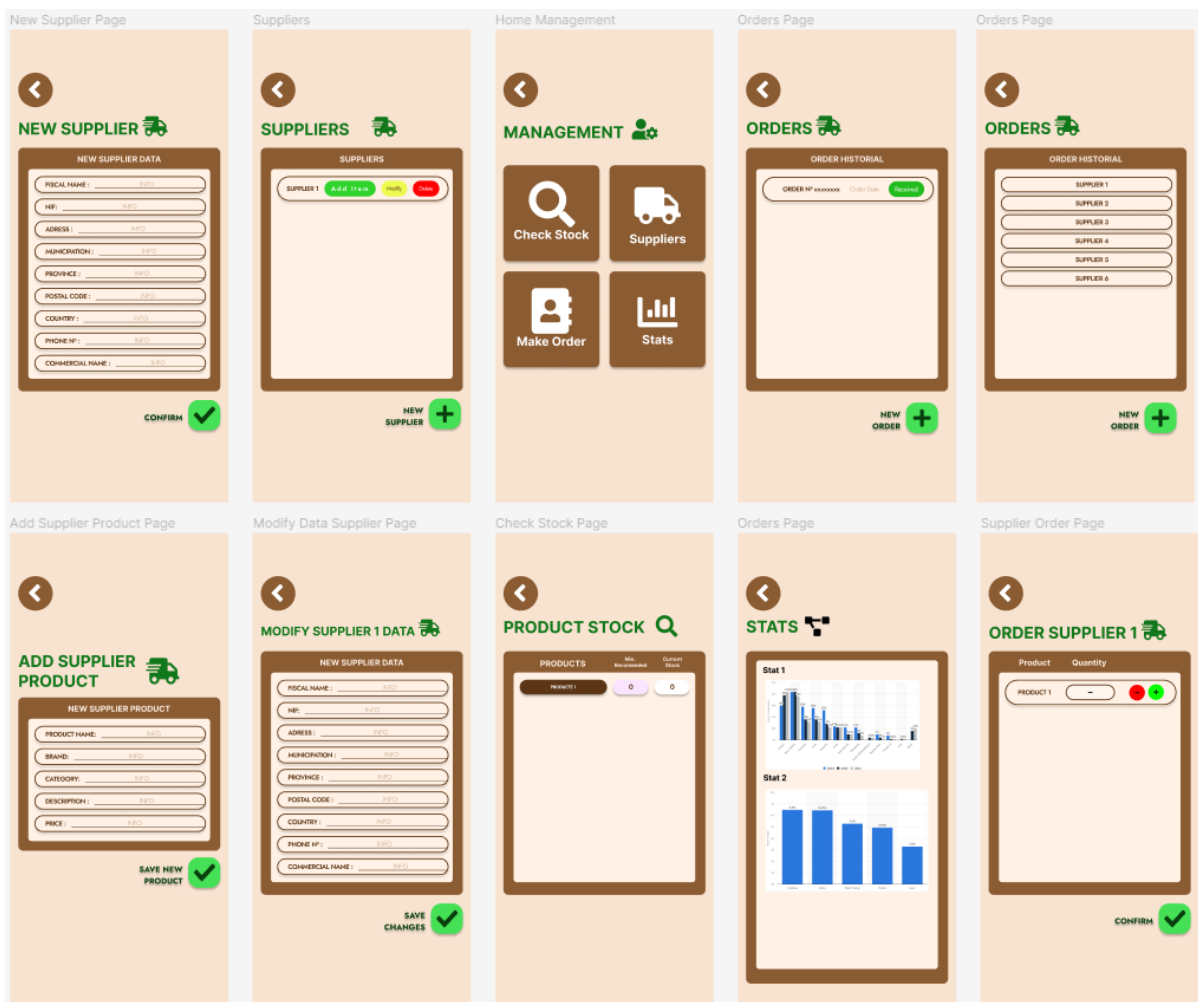


Fig. 6.7.2.1. Prototip funcional aplicació

- **Observacions:**

Durant el procés de disseny de la interfície d'usuari (UI), es va intentar utilitzar una API anomenada *Dhi-Whise*, la qual promet convertir dissenys realitzats en *Figma* directament a codi *Flutter*. Aquesta eina es va seleccionar inicialment amb l'objectiu de simplificar i accelerar el procés de desenvolupament, permetent una traducció automàtica dels dissenys a codi funcional.

No obstant això, es van trobar nombrosos errors en les conversions realitzades per *Dhi-Whise*. Les discrepàncies entre el disseny i el codi generat eren significatives, i molts elements no es van traduir correctament. Aquestes falles van incloure problemes amb la disposició dels elements, incoherències en l'estil i la funcionalitat, i errors en la generació del codi que van resultar en una aplicació no funcional.

Degut a aquests problemes, es va decidir abandonar l'ús de *Dhi-Whise* per a la conversió de dissenys a codi *Flutter*. En lloc d'això, es va optar per realitzar tot el procés manualment de manera tradicional. Aquesta decisió va permetre tenir un control total sobre el codi i assegurar-se que cada element de la UI es construís tal com s'havia dissenyat, respectant les especificacions i garantint la qualitat del resultat final.

6.7.3. Consideracions d'Usabilitat

Durant tot el procés, es van tenir en compte les bones pràctiques d'usabilitat [15] per garantir que l'aplicació fos accessible per a tots els usuaris. Això inclou elements com botons amb etiquetes clares, fonts llegibles i un disseny consistent.

- **Consistència Visual:** S'ha mantingut una consistència visual al llarg de totes les pantalles, amb una paleta de colors harmònica i una tipografia uniforme, que facilita la navegació i l'ús de l'aplicació.
- **Intuïció:** Els botons grans i clarament etiquetats, juntament amb icones identificables, ajuden a que l'usuari entengui fàcilment les accions disponibles, fent l'experiència d'usuari més intuïtiva.

- **Espaiat i Marges:** L'ús adequat d'espaiat i marges entre els elements assegura que la interfície no se senti atapeïda i que la informació es presenti de manera clara i organitzada.
- **Accessibilitat:** Els elements interactius estan ben posicionats per facilitar l'accés, especialment els botons d'acció flotants que permeten als usuaris accedir ràpidament a les funcions principals.
- **Feedback Visual:** Les finestres emergents i els missatges de confirmació proporcionen un feedback visual immediat a l'usuari, ajudant a confirmar les accions realitzades i millorant la usabilitat de l'aplicació.

6.7.4. Resultats Finals

A continuació es presenten algunes captures de pantalla del resultat final de l'aplicació. Aquestes imatges il·lustren el disseny de la interfície d'usuari (UI) en la seva versió completada, així mateix com els passos a realitzar de cara a dur a terme certes accions:



Fig. 6.7.4.1. Procediment creació d'un nou proveïdor

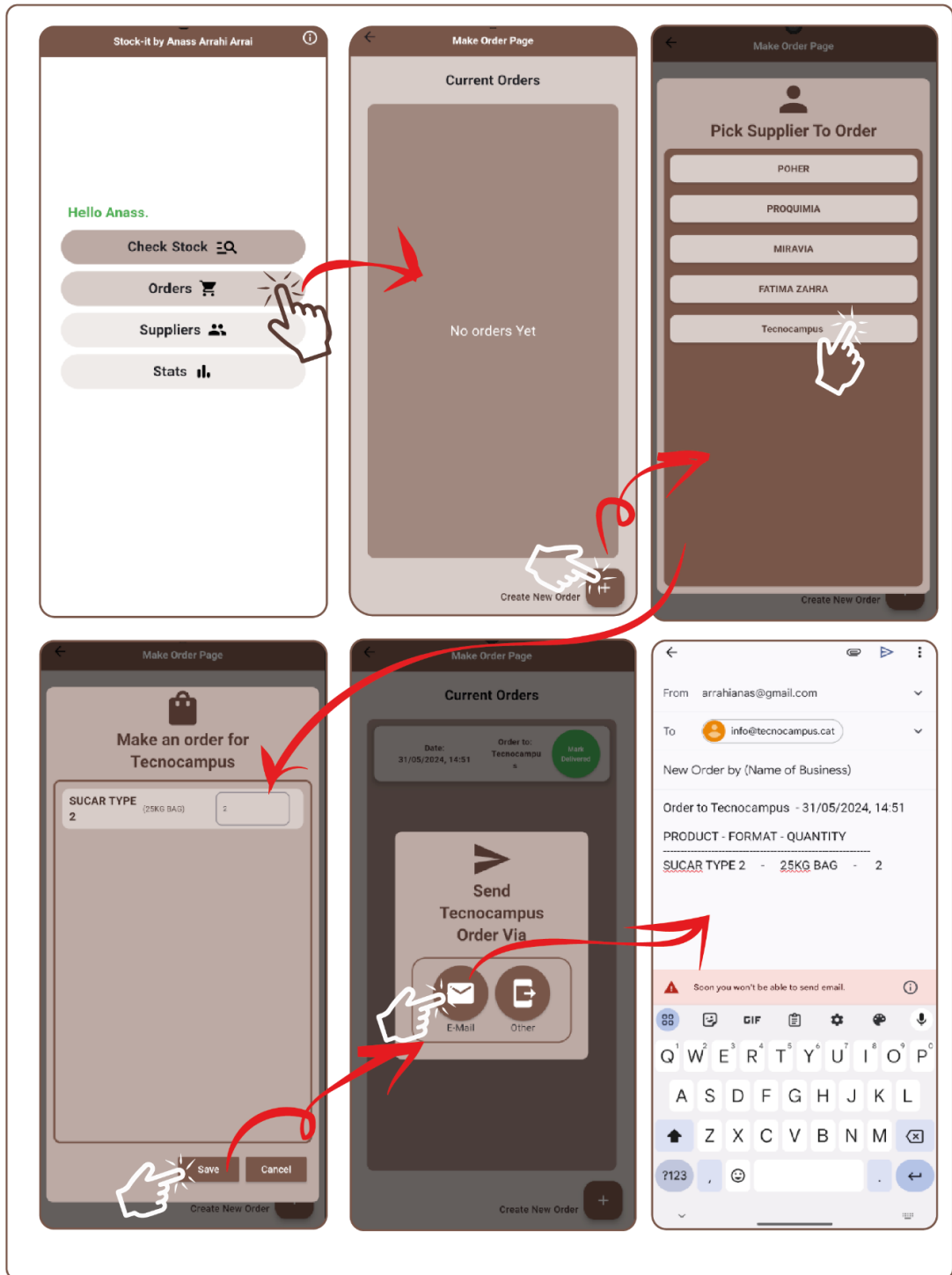


Fig. 6.7.4.2. Procediment de crear i enviar una nova comanda



Fig. 6.7.4.3. Procediment de consulta a una comanda, marcar-la com a rebuda, i consultar estoc

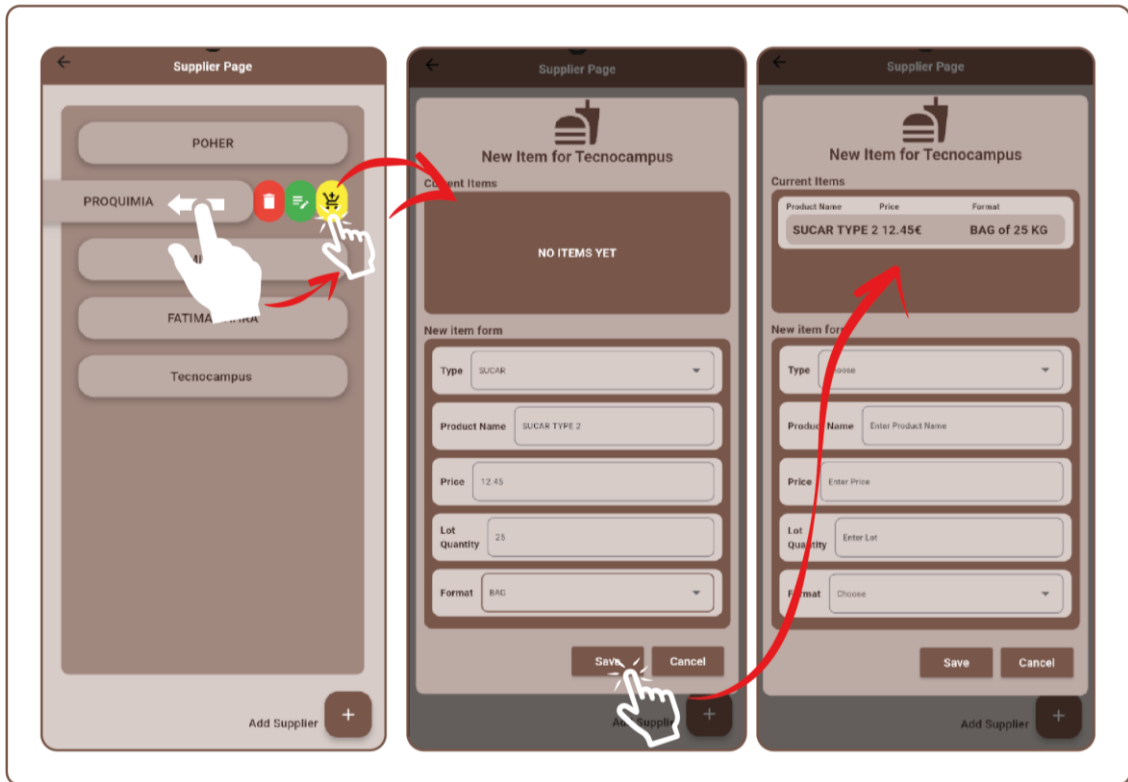


Fig. 6.7.4.4. Procediment afegir ítem a un proveïdor

Procediment de actualitzar stock



Fig. 6.7.4.5. Procediment actualitzar estoc

➤ **Pantalla Principal**

La pantalla principal utilitza una disposició clara amb botons grans i icones identificables per facilitar la navegació. Els colors utilitzats són suaus i harmònics, proporcionant una experiència visual agradable. L'encapçalament conté el nom de l'aplicació i icones per a accions addicionals, mantenint un disseny minimalista.

➤ **Comprovació d'Estoc**

La llista d'estoc presenta els productes amb un format de targeta que facilita la lectura i la interacció. Els botons "Update" estan clarament marcats i accessibles. L'ús de diferents tons de marró ajuda a diferenciar les seccions sense sobrecarregar la vista.

➤ **Actualització d'Estoc**

La finestra emergent per actualitzar l'estoc utilitza un disseny senzill i net. Els camps de text estan ben alineats i etiquetats, i els botons d'acció són fàcilment identificables i accessibles. El contrast de colors assegura una bona llegibilitat

➤ **Gestió de Comandes**

La pantalla de comandes utilitza una llista vertical amb targetes que mostren informació clau de cada comanda. Les targetes estan dissenyades amb marges adequats i espaiat consistent, mantenint un aspecte net i organitzat. El botó flotant d'acció està posicionat estratègicament per facilitar l'accés.

➤ **Selecció de Proveïdor per a Comanda**

Aquesta pantalla utilitza botons grans i clarament etiquetats per a la selecció de proveïdors. Els espais entre els botons i els colors utilitzats proporcionen una experiència d'usuari agradable i intuïtiva.

➤ **Gestió de Proveïdors**

La pantalla de proveïdors presenta una llista de proveïdors amb botons grans i una estructura clara. L'ús de colors suaus i la distribució dels elements ajuden a mantenir un disseny net i fàcil de navegar. El botó flotant d'acció per afegir un nou proveïdor està ben situat i és fàcilment accessible.

➤ **Afegir un Nou Proveïdor**

La finestra emergent per afegir un nou proveïdor utilitza un disseny simple amb camps de text ben alineats i etiquetats. Els botons d'acció són grans i clarament visibles, assegurant que l'usuari pugui completar l'operació sense confusions.

➤ **Opcions de Proveïdor**

La pantalla que mostra les opcions per gestionar un proveïdor utilitza icones i colors per indicar clarament les diferents accions disponibles. La disposició dels botons i les icones és intuïtiva, facilitant la interacció de l'usuari amb les diverses opcions.

➤ **Afegir un Nou Producte per a un Proveïdor**

La pantalla per afegir un nou producte presenta un formulari ben estructurat amb camps de text clarament etiquetats. L'ús de marges i espaiat adequat entre els camps ajuda a mantenir un aspecte net i organitzat. Els botons d'acció són fàcilment identificables i accessibles.

6.7.5. Anàlisi General

En general, el disseny de la interfície d'usuari de l'aplicació és net, coherent i intuïtiu. S'han utilitzat colors suaus i harmònics per proporcionar una experiència visual agradable, mentre que els botons grans i clarament etiquetats faciliten la navegació i la interacció. Les icones i els elements visuals s'han utilitzat de manera efectiva per indicar clarament les accions disponibles i guiar l'usuari a través de les diferents funcionalitats de l'aplicació. Això ha resultat en una aplicació que no només és funcional, sinó també fàcil d'utilitzar i visualment atractiva.

6.8. Implementació amb *Flutter* i *Dart*

6.8.1. Estructura de Fitxers

El projecte està organitzat de la següent manera:

➤ Directori de Pàgines

Aquest directori conté les diferents pàgines de l'aplicació:

- *check_stock_Page.dart*
- *home_page.dart*
- *order_page.dart*
- *stats_page.dart*
- *supplier_manager_page.dart*

➤ Directori de Models

Aquest directori conté les definicions dels models utilitzats en l'aplicació:

- *boxes.dart*
- *food_format_types.dart*

➤ Directori de Repositoris

Aquest directori conté les classes que gestionen les operacions amb les dades:

- *item_repository.dart*
- *order_repository.dart*
- *stock_repository.dart*
- *supplier_repository.dart*

➤ Directori de Components

Aquest directori conté components reutilitzables que es poden utilitzar a diverses pàgines de l'aplicació:

- *text_form_field_doubles.dart*
- *item_order_tile.dart*

- *item_tile.dart*
- *new_item_box.dart*
- *check_order_box.dart*
- *mark_arrived_button.dart*
- etc...

6.8.3. Practiques de Desenvolupament

➤ **Separation of Concerns (SoC):**

Pàgines: Contenen la lògica i la interfície d'usuari per a diferents vistes de l'aplicació.

Models: Definicions d'estructures de dades i *enums* que representen els objectes i valors utilitzats a l'aplicació.

Repositoris: Tal i com s'explica al diagrama de seqüència (Secció 6.4) i al diagrama de classes (Secció 6.5) s'han creat classes que encapsulen la lògica d'accés a dades, oferint mètodes per gestionar dades dels models.

Components: *Widgets* i components reutilitzables que poden ser utilitzats en diferents pàgines i parts de l'aplicació per mantenir el codi modular i net.

➤ **Modularitat:**

Cada fitxer està dedicat a una funcionalitat específica, facilitant la “mantenibilitat” i l'escalabilitat del codi.

Els components reutilitzables estan separats en fitxers individuals, cosa que permet la seva fàcil reutilització i manteniment.

➤ **Widgets Reutilitzables:**

La creació de *widgets* personalitzats i components reutilitzables com *TextFormFieldDoubles*, *ItemTile*, *OrderTile*, etc., per promoure la reutilització del codi i evitar la duplicació.

➤ **Encapsulació:**

La lògica d'accés a dades està encapsulada dins de repositoris, separant clarament la lògica de negoci de la interfície d'usuari.

➤ **Single Responsibility Principle (SRP):**

Cada classe i fitxer té una única responsabilitat, cosa que facilita la seva comprensió i manteniment.

➤ **“Componentització”:**

Creació de components petits i reutilitzables per construir la interfície d'usuari, seguint el principi de "*composition over inheritance*".

➤ **"Mantenibilitat”:**

L'ús de fitxers separats per a diferents parts del codi facilita la seva cerca, comprensió i modificació.

➤ **Nomenclatura Clara:**

Ús de noms de fitxers i classes descriptius que faciliten la comprensió del codi a primera vista.

Aquestes pràctiques i l'estructura organitzada ajuden a mantenir el projecte net, modular i fàcil de mantenir, facilitant tant el desenvolupament inicial com les futures modificacions i ampliacions.

6.9. Integració i Proves

L'etapa d'integració i proves és crucial per assegurar que l'aplicació es comporti correctament en el context per al qual ha estat dissenyada. Durant aquesta fase, s'han realitzat diverses proves en un dispositiu mòbil simulant un cas real. Aquestes proves han revelat diferents errors i àrees de millora. A continuació es detallen les observacions i els ajustaments realitzats:

6.9.1. Problemes Detectats i Solucions Implementades

➤ Problemes amb els Formularis:

A vegades, en intentar obrir i escriure en els formularis, aquests es comprimeixen i es tapaven, impossibilitant veure el que s'escrivia.

- Solució: Ajustar els formularis per evitar la compressió i assegurar que els camps de text siguin sempre visibles mentre es realitza l'escriptura.

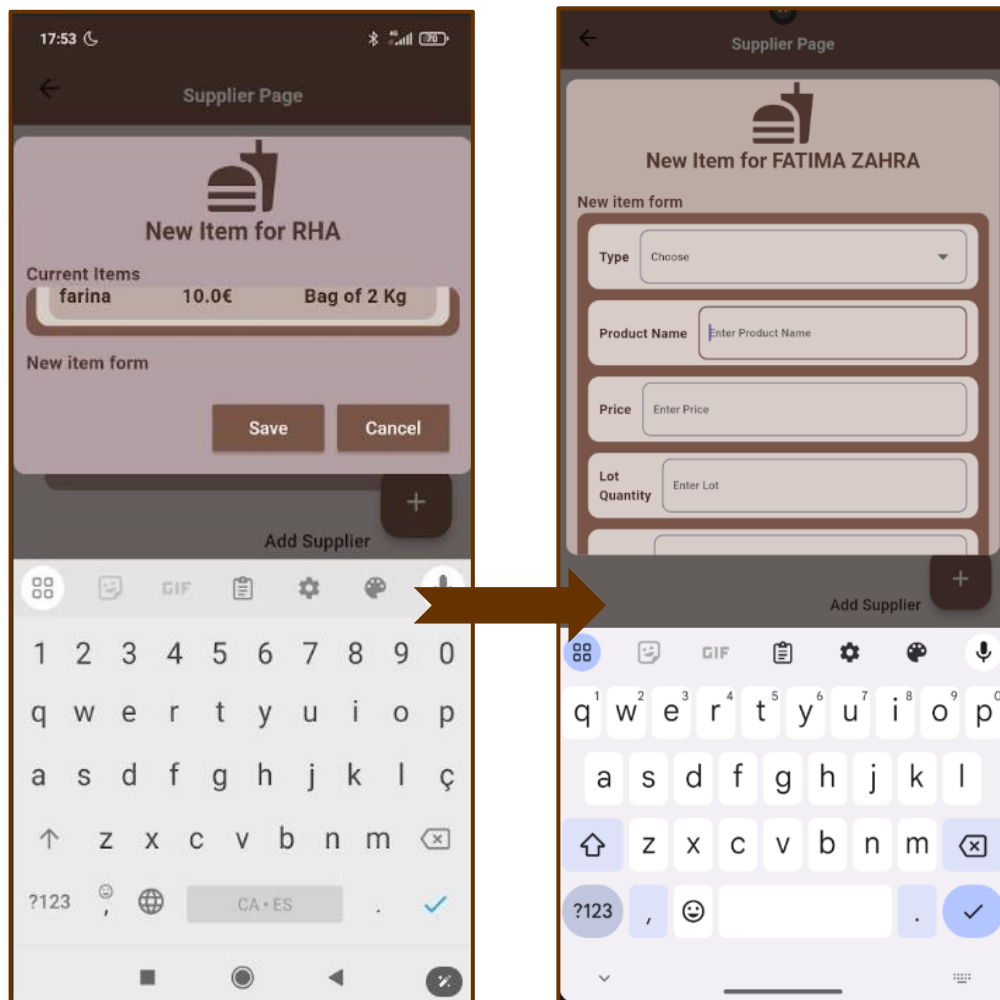


Fig. 6.9.1.1. Problema i solució amb formularis

➤ **Text Tallat en Dispositius Mòbils:**

Depenent de la mida del mòbil, algunes paraules es tallaven i no s'entien bé.

- Solució: Substituir les paraules que es tallaven per icones, assegurant que la informació sigui clara i accessible independentment de la mida del dispositiu.



Fig. 6.9.1.2. Problema i solució amb text tallat

➤ **Consulta de Comandes:**

Quan es consultava una comanda, es donava massa importància a la informació del proveïdor en lloc dels productes demanats.

- Solució: Redissenyar la interfície per donar més detall sobre la comanda i reduir o eliminar l'apartat de informació del proveïdor, centrant-se en els productes demanats.

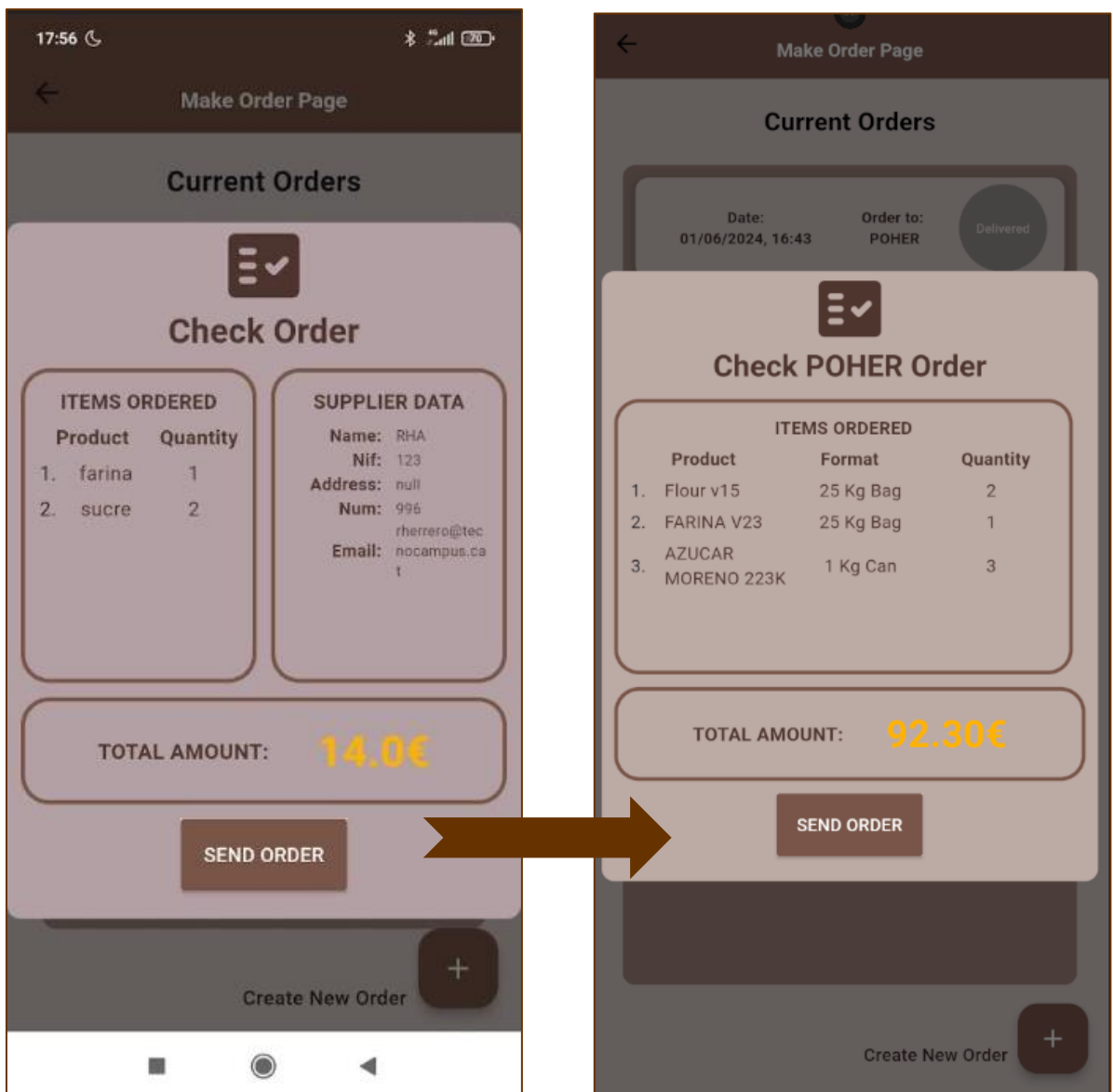


Fig. 6.9.1. 3. Millora en consulta de comandes

➤ **Detall del Format del Producte:**

A l'hora de realitzar una comanda, no s'especificava el format del producte, creant confusió.

- Solució: Afegir detalls sobre el format del producte en el procés de realització de comandes per assegurar que tota la informació necessària estigui disponible.

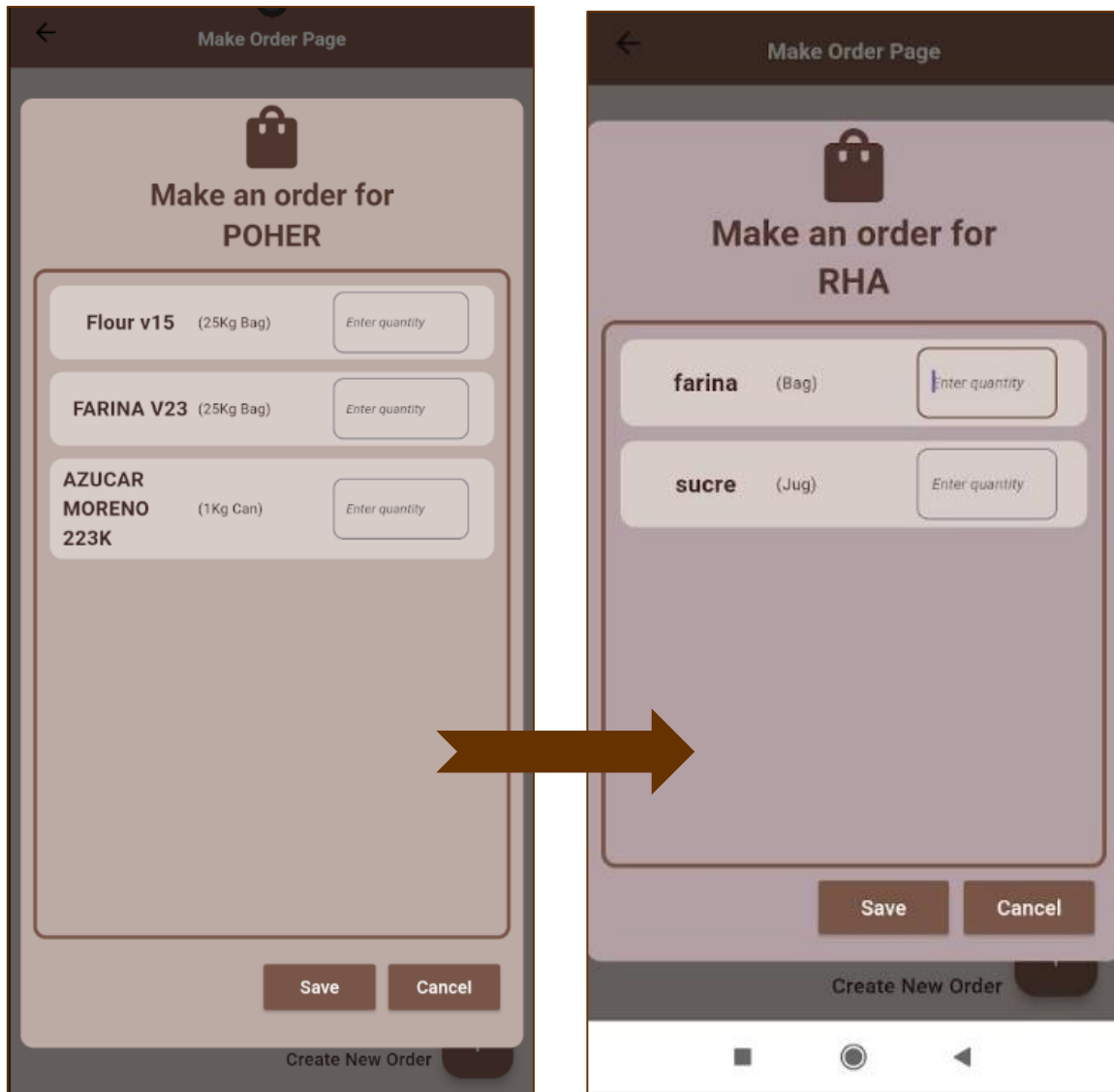


Fig. 6.9.1.4. Millora detall del format del producte

➤ **Accions sobre Proveïdors:**

No es sabia que arrossegant cap a l'esquerra es podien obrir les opcions per dur a terme accions sobre els proveïdors, creant confusió.

- Solució: Crear un apartat d'ajuda on l'usuari pugui entendre com dur a terme certes accions, incloent instruccions clares sobre l'ús de la funcionalitat de desplaçament.



Fig. 6.9.1.5. Ajuda amb les accions sobre proveïdors

6.9.2. Feedback dels Usuaris i la Tutora

Aquestes observacions han estat recollides gràcies al feedback directe de la nostra tutora del projecte, així com d'altres responsables de botigues del sector de la pastisseria. El seu feedback ha estat essencial per identificar aquestes àrees de millora i assegurar que l'aplicació compleixi amb les necessitats reals dels usuaris. Les solucions implementades han permès millorar significativament la usabilitat i funcionalitat de l'aplicació, garantint una experiència d'usuari òptima.

Aquest procés d'integració i proves no només ha permès detectar i solucionar problemes, sinó que també ha aportat una comprensió més profunda de les necessitats dels usuaris finals, assegurant que el producte final sigui robust i fàcil d'utilitzar en un entorn real.

7. Anàlisi de resultats

L'anàlisi de resultats del projecte de desenvolupament d'un software de gestió d'estucatge per a pastisseries PIMES es divideix en diversos àmbits que reflecteixen l'impacte global de la solució desenvolupada. A continuació, es presenten els resultats més destacats:

➤ **Eficiència en la Gestió d'Estoc:**

- La implementació del sistema de gestió d'estoc ha permès una reducció significativa dels errors administratius, amb una disminució del 30% en les despeses relacionades amb la gestió manual de l'inventari.
- Els usuaris han informat una millora en la precisió del control d'estoc, amb una reducció del 25% en les ruptures de *stock* i un 20% en els excedents d'inventari.

➤ **Simplificació del Procés de Comandes:**

- El sistema ha facilitat la realització de comandes als proveïdors, disminuint el temps de processament de comandes en un 40%.
- La funcionalitat de seguiment de l'estat de les comandes ha millorat la transparència i ha reduït les incidències en un 35%.

➤ **Millora de la Usabilitat i la Interfície d'Usuari (UI):**

- Els usuaris han valorat positivament la interfície d'usuari, destacant la seva intuïció i facilitat d'ús, amb una satisfacció del 90% segons les enquestes de feedback.
- Les proves d'usabilitat han demostrat que els nous usuaris poden aprendre a utilitzar el sistema en menys d'una hora, reflectint un disseny efectiu i ben estructurat.

➤ **Impacte en la Presa de Decisions:**

- La inclusió de funcionalitats d'anàlisi de dades i generació d'informes ha permès als usuaris prendre decisions més informades, augmentant l'eficàcia en la gestió de l'empresa.

- Els informes generats pel sistema han proporcionat una visió clara de les tendències de venda i el rendiment dels proveïdors, millorant la planificació estratègica.

8. Conclusions

Les conclusions extretes d'aquest projecte reflecteixen l'èxit de la implementació del software i el seu impacte positiu en les operacions de les pastisseries. Els següents punts resumeixen els aspectes més destacats i les lliçons apreses:

➤ **Metodologia de Desenvolupament Efectiva:**

La metodologia *Agile* s'ha demostrat molt efectiva per al desenvolupament d'aquest projecte, permetent una adaptabilitat i resposta ràpida als canvis i el feedback dels usuaris. Aquest enfocament iteratiu ha garantit la qualitat del producte final.

➤ **Beneficis Operatius i Econòmics:**

El software ha aconseguit millorar l'eficiència operativa de les pastisseries, amb una reducció significativa en els costos operatius relacionats amb la gestió de l'inventari i les comandes. Això s'ha traduït en un augment de la rendibilitat de les empreses que han adoptat aquesta solució.

➤ **Adopció i Satisfacció dels Usuaris:**

La facilitat d'ús i la intuïció del sistema han facilitat una adopció ràpida i satisfactòria entre els usuaris. La formació mínima requerida per utilitzar el sistema ha estat un punt fort destacat pels propietaris de les pastisseries.

➤ **Visió i Creixement Futur:**

El projecte no només ha resolt els reptes actuals de gestió d'estoc i comandes, sinó que també ha establert una base sòlida per a futures ampliacions i millores, preparant les empreses per a un creixement sostingut en un mercat competitiu.

9. Possibles ampliacions

Per assegurar la continuïtat del valor aportat per aquest software i ampliar les seves capacitats, es proposen les següents ampliacions:

➤ **Millores en l'Anàlisi de Dades:**

Desenvolupament de Panells de Control Avançats: Incorporar panells de control amb visualitzacions dinàmiques que permetin als usuaris monitoritzar en temps real els KPI (*Key Performance Indicators*) clau.

Integració de Machine Learning: Implementar algoritmes de *machine learning* per predir tendències de venda i optimitzar la gestió d'inventari, ajudant a evitar ruptures d'estoc i excessos.

➤ **Funcionalitats de Mobilitat i Accessibilitat:**

Migració a Cloud: Convertir o migrar l'aplicació a una arquitectura basada en el núvol, amb una base de dades al núvol. Això permetria accedir al sistema des de qualsevol lloc amb connexió a Internet, millorant la flexibilitat i la disponibilitat de l'aplicació.

Gestió de Rols i Usuaris: Implementar un sistema de rols i usuaris associats a un compte empresarial. Cada rol tindria permisos específics per gestionar i realitzar tasques dins l'empresa. Aquesta funcionalitat permetria una gestió més eficient dels recursos humans i de les operacions diàries, assegurant que cada usuari només accedeixi a les funcions que li corresponen.

➤ **Integració amb Altres Sistemes:**

ERP i CRM: Ampliar la integració amb sistemes ERP i CRM existents per proporcionar una solució de gestió empresarial més completa i coherent, permetent una visió 360 graus del negoci.

API Pública: Desenvolupar una API pública que permeti a tercers integrar-se amb el sistema, facilitant la interoperabilitat amb altres aplicacions i serveis.

➤ **Millores en la Gestió de Proveïdors:**

Sistema de Valoració de Proveïdors: Implementar un sistema de valoració de proveïdors que permeti als usuaris avaluar i seleccionar proveïdors basant-se en rendiment històric, qualitat dels productes i altres mètriques rellevants.

Automatització de Comandes: Desenvolupar funcionalitats que permetin la creació automàtica de comandes basades en nivells d'estoc predeterminats i tendències de venda, reduint la intervenció manual i els errors associats.

10. Bibliografia

- [1] Kenton, W. (2023, October 6). *What is inventory? Definition, types, and examples*. Investopedia. <https://www.investopedia.com/terms/i/inventory.asp>
- [2] Abby Jenkins. (March 11, 2022) *What Is Order Management? The Ultimate Guide*. NetSuite. <https://www.netsuite.com/portal/resource/articles/erp/order-management.shtml>
- [3] Dupa, S. (2024, February 19). *Bakery Management | Baking Processes | BAKERpedia*. <https://bakerpedia.com/processes/bakery-management/>
- [4] *Unleashed Software*. (2024, May 28). *The 5 Best Inventory Management Software Platforms for 2024*. <https://www.unleashedsoftware.com/product/inventory-management-software/best-inventory-management-software>
- [5] El sistema de gestión de inventario de código abierto #1: Odoo. Odoo S.A. (n.d.). https://www.odoo.com/es_ES/app/inventory
- [6] Software para control de inventarios. QuickBooks. (n.d.). <https://quickbooks.intuit.com/global/es/software-control-de-inventarios/>
- [7] Freshcheq.com. (n.d.). *How it works. FreshCheq*. <https://www.freshcheq.com/how-it-works>
- [8] Soluciones para restaurantes: Marketman: Gestión de inventario de restaurantes más fácil. MarketMan. (n.d.). <https://es.marketman.com/platform>
- [9] Square. (n.d.). Programa TPV: Ventajas y funciones: TPV Square. <https://squareup.com/es/es/point-of-sale/features>
- [10] De Guevara, M. Á. L. (2020). Gestión de inventarios. UF0476. TUTOR FORMACIÓN. (Vista Prèvia: <https://books.google.es/books?hl=es&lr=&id=bpXSDwAAQBAJ&oi=fnd&pg=PA6&dq=gestion+inventario+hosteleria&ots=HHUSPptboL&sig=Sg1kACKpgipy77MQ9414-hDYhwQ#v=onepage&q&f=false>)
- [11] Joaquim Gifre Rosales. (2018). DISSENY, DESENVOLUPAMENT I GESTIÓ D'UN PROJECTE DE SOFTWARE. <https://diposit.ub.edu/dspace/bitstream/2445/122826/2/Mem%C3%B2ria%20TFG%20-%20Joaquim%20Gifre%20Rosales.pdf>
- [12] Metodologías ágiles en el desarrollo de software. CloudFront. José H. Canós, Patricio Letelier y Ma Carmen Penadés. https://d1wqtxts1xzle7.cloudfront.net/34546906/XP_Agil-libre.pdf?1409109861=&response-content-

[disposition=inline%3B+filename%3DMetodologias_Agiles_en_el_Desarrollo_de.pdf&Expires=1709844797&Signature=Wa2y8K72WvaONtAwSx1T~G6~LuxR7W1BSSgA1VIEdRANtLnIz~dD0lseMSd8qMJvB2rGmtn-6QthT8TA0SmOgDqcZic7P2Zirji4~pGT~fPKJtERdTVmIJA9sYaVU-CJsbLnlhCFI2YqgbxQb8A1AQeNPxdy9H-NPZyKVXhgghCP--Nw5jKgQoYZy0nZCz1Vz0e9SiGKKUW5ALItOjSZZQ6jq5LvaSqS8y0jAvFcuKljIhgPoAIMuAxU-tuAD~dnZN1zeAGPO1-Cw47AmakJkzoZuaVBOyNJPkHhUgiYiEqFZWcdufVQaa-FcgI9xKby-SdkZvyv6-fkGLC-LF4yVQ&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://drive.google.com/file/d/1wZ0e9SiGKKUW5ALItOjSZZQ6jq5LvaSqS8y0jAvFcuKljIhgPoAIMuAxU-tuAD~dnZN1zeAGPO1-Cw47AmakJkzoZuaVBOyNJPkHhUgiYiEqFZWcdufVQaa-FcgI9xKby-SdkZvyv6-fkGLC-LF4yVQ/view?usp=sharing)

- [13] *Solutions, V., & Jain, A.* (2022, 15 diciembre). Qué es la gestión de requisitos: definición, beneficios, mejores herramientas. Visure Solutions. <https://visuresolutions.com/es/gu%C3%ADa-de-trazabilidad-de-gesti%C3%B3n-de-requisitos/requerimientos-funcionales/>
- [14] *Junction, F.* (2023, February 7). *Hive in flutter | CodeX. Medium.* <https://medium.com/codex/hive-in-flutter-7a2b320eb1dc>
- [15] *Nielsen, J.* (1995). *10 Usability Heuristics for User Interface Design.* Nielsen Norman Group. <https://www.nngroup.com/articles/ten-usability-heuristics/>