

Aplicació web per a la gestió d'un magatzem logístic

Memòria de Projecte Final de Màster

Màster universitari de Desenvolupament de Llocs i Aplicacions Web

Autor: Tània Garcia Font

Consultor: Daniel Garcia Nebot

Professor: César Pablo Córcoles Briongos

3 de gener de 2022



Aquesta obra està subjecta a una llicència de [Reconeixement-
NoComercial-SenseObraDerivada 3.0 Espanya de Creative
Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Resum

En aquest projecte es desenvolupa una aplicació destinada al sector logístic amb la finalitat que sigui un sistema de gestió de magatzem, que al sector es coneix com a SGM o, en castellà, SGA (*Sistema de Gestión de Almacén*). Més concretament, es destinarà a la gestió de l'emmagatzematge de mercaderies de palet sencer. Aquest software té com a objectiu ser l'eina de treball del dia a dia d'un magatzem logístic. L'aplicació s'encarrega de contenir les dades dels diferents clients, les referències entrades de cada producte i les ubicacions on s'emmagatzemará la mercaderia. Mitjançant lectors de codis de barres es llegeixen les etiquetes dels palets que entren i surten del magatzem i s'introdueixen totes les seves dades al sistema; això permet saber, en temps real, la quantitat de palets que hi ha de cada referència i client, així com obtenir informació sobre els lots, les dates de caducitat, els SSCC, etc. Si és necessari, també es poden fer les lectures manualment.

Amb aquesta aplicació es podrà mantenir la traçabilitat del magatzem en temps real i es podrà donar un servei essencial als clients, ja que la traçabilitat del producte actualment és imprescindible en termes de qualitat i logística.

L'aplicació es desenvoluparà amb Angular com a *front-end*, amb Laravel com a *back-end* i amb MySQL per a la base de dades. A més, s'utilitzarà WebSockets per mantenir els usuaris informats dels canvis en temps real.

En conclusió, el resultat és una aplicació intuïtiva, fàcil d'utilitzar i d'instal·lar; a més, es pot fer servir en qualsevol dispositiu amb connexió a internet.

Paraules clau: sistema de gestió d'estocs, magatzem logístic, traçabilitat, aplicació web, software logístic, Laravel, Angular, MySQL, WebSockets, SGA.

Abstract

In this project, an application is developed for the logistics industry so that it is a Warehouse Management System (WMS). More specifically, the application will be used for the warehouse management of a full pallet system. The aim of this software is to become the day-to-day software for a logistic warehouse. The application will include client information, the input references of each product and the locations where the merchandise will be stored. Each pallet label is read through barcode readers, which allows us to know, in real time, the actual number of pallets for each reference and client, as well as information about production batches, expiration dates, SSCC codes, etc. If needed, it is also possible to read labels manually.

This application will allow to keep the traceability of a warehouse in real time and will provide customers with an essential service, since product traceability is currently essential in terms of quality and logistics.

The application will be developed using Angular as a front-end, Laravel as a back-end and MySQL for the database. In addition, WebSockets will be used to keep users up to date with changes in real time.

In conclusion, the result is an intuitive application that is easy to use and install. In addition, it can be used in any device with an Internet connection.

Keywords: warehouse management system, logistic warehouse, traceability, web application, logistic software, Laravel, Angular, MySQL, WebSockets, WMS.

Índex

1. Introducció	8
1.1 Context i justificació del Treball	8
1.2 Objectius del Treball	9
1.3 Enfocament i mètode seguit	9
1.4 Planificació del Treball	11
2. Arquitectura de l'aplicació i plataforma de desenvolupament	14
3. Diagrames UML	16
3.1 Diagrames de flux	16
3.2 Diagrama Base de dades	19
3.3 Usuaris	20
3.4 Casos/escenaris d'ús	21
4. Prototips	24
4.1 Wireframes	24
5. Usabilitat/UX	42
6. Tests	44
6.1 Resultats tests amb Karma (<i>ng test</i>):	45
6.2 Resultats test de cobertura:	48
7. Requisits i instruccions d'instal·lació i ús	49
7.1 Funcionament en xarxa local	49
7.2 Funcionament en xarxa externa	49
8. Projecció de futur	53
9. Anàlisi de mercat	54
9.1 Viabilitat	54
10. Conclusions	57
Annex 1. Lliurables del projecte	58
Annex 2. Llibreries/Codi extern utilitzat	59
Annex 3. Captures de pantalla	60
Annex 4. Guia d'usuari	61
Annex 5. Bibliografia	62

Figures i taules

Índex de figures

Figura 1: Planificació PAC1 Diagrama de Grantt.....	12
Figura 2: Planificació PAC2 Diagrama de Grantt.....	13
Figura 3 Planificació PAC3 Diagrama de Grantt.....	13
Figura 4: Planificació Lliurement Final Diagrama de Grantt.....	13
Figura 5: Diagrama de flux: Gestió d'usuaris.....	16
Figura 6: Diagrama de flux: Entrada de dades.....	17
Figura 7: Diagrama de flux: Entrada de palets.....	17
Figura 8: Diagrama de flux: Bloquejar palets.....	18
Figura 9: Diagrama de flux: Consulta estocs.....	18
Figura 10: Diagrama de flux: Planificació de sortides.....	19
Figura 11: Diagrama de flux: Sortida de palets.....	19
Figura 12: Diagrama de la Base de Dades.....	20
Figura 13: Wireframe Inici.....	24
Figura 14: Wireframe Regsitre ubicacions.....	24
Figura 15: Wireframe Regsitre articles.....	25
Figura 16: Wireframe Regsitre clients.....	25
Figura 17: Wireframe Ubicacions registrades.....	26
Figura 18: Wireframe Articles registrats.....	26
Figura 19: Wireframe Clients registrats.....	27
Figura 20: Wireframe Introducció de palets.....	27
Figura 21: Wireframe Consulta d'entrades per data.....	28
Figura 22: Wireframe Planificació de sortides.....	29
Figura 23: Wireframe Consulta pickings planificats.....	30
Figura 24: Wireframe Picking sortides.....	31
Figura 25: Wireframe Consulta de sortides per data.....	32
Figura 26: Wireframe Estoc SSCC per producte.....	33
Figura 27: Wireframe Estoc per albarà.....	34
Figura 28: Wireframe Estoc per client.....	35
Figura 29: Wireframe Estoc per producte.....	36
Figura 30: Wireframe Estoc per ubicació.....	37
Figura 31: Wireframe Estoc de lots per client.....	38
Figura 32: Wireframe Dades per SSCC.....	39
Figura 33: Wireframe Bloquejar i desbloquejar palets.....	40
Figura 34: Wireframe Consulta de palets bloquejats.....	40
Figura 35: Wireframe SignIn.....	41
Figura 36: Wireframe LogIn.....	41
Figura 37: Resultat testing unitari amb Karma (part 1).....	45
Figura 38: Resultat testing unitari amb Karma (part 2).....	46
Figura 39: Resultat testing unitari amb Karma (part 3).....	47
Figura 40: Resultat cobertura testing amb <i>code coverage</i>	48
Figura 41: Antena WIFI.....	55

Figura 42: Antena WIFI.....	56
Figura 43: Extracte codi font introduccio-palets2.component.ts.....	60

Índex de taules

Taula 1: Planificació PAC1	11
Taula 2: Planificació PAC2	11
Taula 3: Planificació PAC3	11
Taula 4: Planificació Lliurement Final	12
Taula 5: Cas/escenari d'ús 1	21
Taula 6: Cas/escenari d'ús 2	21
Taula 7: Cas/escenari d'ús 3	22
Taula 8: Cas/escenari d'ús 4	22
Taula 9: Cas/escenari d'ús 5	22
Taula 10: Cas/escenari d'ús 6.....	23

1. Introducció

1.1 Context i justificació del Treball

Actualment la logística i el transport han adquirit molta importància en el món on vivim. Els productes que consumim són molts i diferents, a més venen de tota part de món i requerim que ens arribin amb la màxima velocitat i qualitat de control.

Per fer possible tot aquest moviment de mercaderies s'ha desenvolupat la traçabilitat, que resulta imprescindible per poder controlar les mercaderies en tot moment, des del moment que surten de les fàbriques de producció fins que arriben al client.

Per poder portar a terme la traçabilitat es necessita una tecnologia específica capaç de reconèixer el producte com a peça única i ser capaç de situar-la en temps i espai en tot moment.

A les plantes de producció s'assigna un número específic a cada unitat única de producte, que és com una matrícula o carnet d'identitat que permetrà identificar aquesta unitat de producte allà on vagi. Aquest número específic juntament amb altres dades de la unitat de producte s'imprimeixen a les etiquetes que aniran enganxades en tot moment a la unitat de producte.

Per raons personals fa molts anys que tinc relació amb empreses del sector logístic, específicament en el món dels magatzems logístics, dits també magatzems reguladors, ja que permeten als seus clients poder tenir un punt on poder estocar i redistribuir els seus productes permetent-los així poder fer una regulació i millors organització dels seus estocs.

És per aquesta raó per la qual voldria encarar el TFM a crear una aplicació web encarada a poder gestionar un magatzem logístic i proporcionar-li una eina imprescindible per poder donar el millor servei als seus clients, aportant-los un seguiment en temps real de l'estoc dels seus productes, així com també un control de les entrades i sortides del producte amb totes les dades que se'n deriven. De portar a terme aquest procés se n'anomena portar la traçabilitat del producte.

En aquest cas m'interessa enfocar l'aplicació a gestionar unitats de producte paletitzat, així que es treballarà amb etiquetes de producte amb la nomenclatura GS1-128, que és la que s'utilitza per realitzar la distribució general de productes i és el més utilitzat a l'entorn logístic.

Actualment és pràcticament impensable que un magatzem o distribuïdor es dediqui a la logística sense tenir cap mena de software que s'encarregui de portar a terme la traçabilitat del producte en temps real.

1.2 Objectius del Treball

Pel que fa als objectius del treball, en general, la majoria estan encarats a ser capaç de donar una solució a les necessitats de la gestió d'estocs i la traçabilitat d'un magatzem logístic, tal i com ja he esmentat anteriorment.

1.2.1 Principals

- Crear una aplicació web totalment funcional.
- Aconseguir descodificar les dades obtingudes de la lectura dels codis de barres dels palets i emmagatzemar-les.
- Aplicar diferents mètodes per evitar errors de lectura durant l'entrada i la sortida dels palets.
- Ser capaç de mostrar la informació obtinguda de l'estoc i el moviment de palets en diferents formats.
- Mostrar diferents tipus i opcions d'interfície segons el tipus d'usuari autenticat.
- Tenir la possibilitat de seguir en temps real la traçabilitat del magatzem.
- Poder registrar clients, referències, ubicacions i usuaris a l'aplicació i obtenir-ne el detall.
- Que l'aplicació permeti la planificació de sortides amb referències diferents i quantitats de palets prefixades.
- Tenir la possibilitat de bloquejar i desbloquejar palets perquè no puguin sortir erròniament del magatzem.

1.2.2 Secundaris

- Considerar la possibilitat d'aplicar el format Progressive Web App a l'aplicació segons la disponibilitat.
- Considerar la possibilitat de poder exportar els llistats d'estoc i consulta de sortides i entrades a Excel.

1.3 Enfocament i mètode seguit

Aquesta aplicació està dirigida a magatzems amb gran quantitat de moviment de palets, per tant, ha de ser una aplicació fàcil d'utilitzar i efectiva, ja que l'empresa que la pugui utilitzar ha de confiar al 100% en aquest software i en la informació que li proporciona. És així tan pel que fa a la facturació, com a l'hora de poder resoldre dubtes als clients sobre l'origen o destí de la seva mercaderia o del seu estoc.

Per descodificar els codis de barres s'utilitzarà un codi lliure i es modificarà per tal de poder satisfer les necessitats del nostre software.

Com a exemple utilitzarem etiquetes de palets fictícies i clients ficticis amb referències de productes, però tenint en compte referents reals per poder fer la situació d'exemple el més real possible.

Tal i com he explicat, l'aplicació seria per a un magatzem logístic, així que ha de ser senzilla, ja que l'haurà d'utilitzar des del personal que és al despatx com els treballadors que hi ha al magatzem carregant i descarregant camions. Així que és molt important que el software sigui ràpid i fàcil d'utilitzar, perquè així no dificulti ni faci més lenta la feina al magatzem.

El mètode a seguir per realitzar el treball serà orientat al que he esmentat anteriorment, a la senzillesa però efectivitat i efectivitat del software.

Per començar s'ha de consensuar el tema del treball i l'abordatge amb el professor de l'assignatura.

S'ha de fer una recerca de bibliografia on es parli de diferents softwares orientats a la gestió de mercaderia i anotar els aspectes més importants a abordar i a tenir en compte.

S'ha de tenir en compte la possibilitat de realitzar el desenvolupament i finalització del projecte en els terminis acordats, així com també de les diferents entregues.

Per desenvolupar el projecte es crearan uns wireframes de les diferents parts de l'aplicació des d'on es partirà per crear l'estructura de l'aplicació. Seguidament es plantejarà una estructura per a la base de dades i es procedirà a la instal·lació de l'entorn de treball. Després s'iniciarà el procés de programació pròpiament dit, es farà una gestió d'errors i es comprovarà el correcte funcionament de l'aplicació. Finalment, es farà una comprovació d'assoliment dels objectius.

És important crear una documentació de tot el procés seguint les PAC a realitzar i acabar entregant-la en forma de memòria. També es crearà una presentació visual del projecte i unes conclusions del treball.

Durant tot el procés hi ha d'haver una connexió constant amb el professor per mantenir la comunicació i anar gestionant el projecte conjuntament.

La idea principal és realitzar el projecte utilitzant diferents eines estudiades i apreses durant el màster, així que he escollit Angular per treballar el FrontEnd, amb Bootstrap per poder donar una interacció a l'usuari senzilla. Pel BackEnd utilitzaré Laravel, i em puc plantejar utilitzar Factories per crear una bona quantitat de dades d'exemple. Per a la base de dades utilitzaré mySQL.

Per desplegar el projecte necessitaré Heroku pel BackEnd i Netlify pel FrontEnd penjant el projecte a GitHub.

També utilitzaré WebSockets des d'un servidor Node per crear alertes per avisar als usuaris connectats sobre els canvis que hi pugui haver mentre estan utilitzant l'aplicació en temps real.

1.4 Planificació del Treball

PAC1			
Tasca	Data inici	Data finalització	Duració
Definició i proposta del projecte	16/09/2021	22/09/2021	7
Recerca d'informació	19/09/2021	23/09/2021	5
Elaboració PAC1 i planificació del projecte	24/09/2021	28/09/2021	5
Entrega PAC1	28/09/2021	28/09/2021	1

Taula 1: Planificació PAC1

PAC2			
Tasca	Data inici	Data finalització	Duració
Estudi usabilitat, casos d'ús, anàlisi de mercat i viabilitat	29/09/2021	30/09/2021	2
Definició definitiva de les funcionalitats	01/10/2021	02/10/2021	2
Prototipat	03/10/2021	08/10/2021	6
Correcció PAC1	09/10/2021	09/10/2021	1
Definició dels principis de disseny i d'interacció	10/10/2021	15/10/2021	6
Disseny de la base de dades	16/10/2021	17/10/2021	2
Posada a punt de l'entorn de desenvolupament	18/10/2021	20/10/2021	3
Elaboració PAC2	21/10/2021	26/10/2021	6
Entrega PAC2	27/10/2021	27/10/2021	1

Taula 2: Planificació PAC2

PAC3			
Tasca	Data inici	Data finalització	Duració
Programació del BackEnd*	28/10/2021	28/11/2021	32
Programació del FrontEnd*	28/10/2021	28/11/2021	32
Correcció PAC2	08/11/2021	09/11/2021	2
Video del funcionament de la primera versió de la app	29/11/2021	29/11/2021	1
Elaboració PAC3	30/11/2021	04/12/2021	5
Entrega PAC3 i video	05/12/2021	05/12/2021	1

Taula 3: Planificació PAC3

Lliurement Final			
Tasca	Data inici	Data finalització	Duració
Finalització del desenvolupament del projecte	06/12/2021	19/12/2021	14
Correcció PAC3	20/12/2021	21/12/2021	2
Finalització de la memòria del projecte	19/12/2021	23/12/2021	5
Crear una presentació del projecte	27/12/2021	29/12/2021	3
Presentació en video del projecte	30/12/2021	31/12/2021	2
Completar l'informe d'autoavaluació	02/01/2022	02/01/2022	1
Entregar tots els documents i videos	03/01/2022	03/01/2022	1

Taula 4: Planificació Lliurement Final

*Desglossament de l'ordre de programació del BackEnd i el FrontEnd:

- Registre i detall de clients
- Registre i detall de referències
- Registre i detall d'usuaris
- Registre i detall d'ubicacions
- Entrada de palets
- Consulta d'entrades
- Estocs en diferents formats
- Planificació de sortides
- Sortida de palets
- Consulta de sortides
- Funció de bloquejar palets
- Detall de palets bloquejats
- Gestió d'usuaris i permisos
- Testing

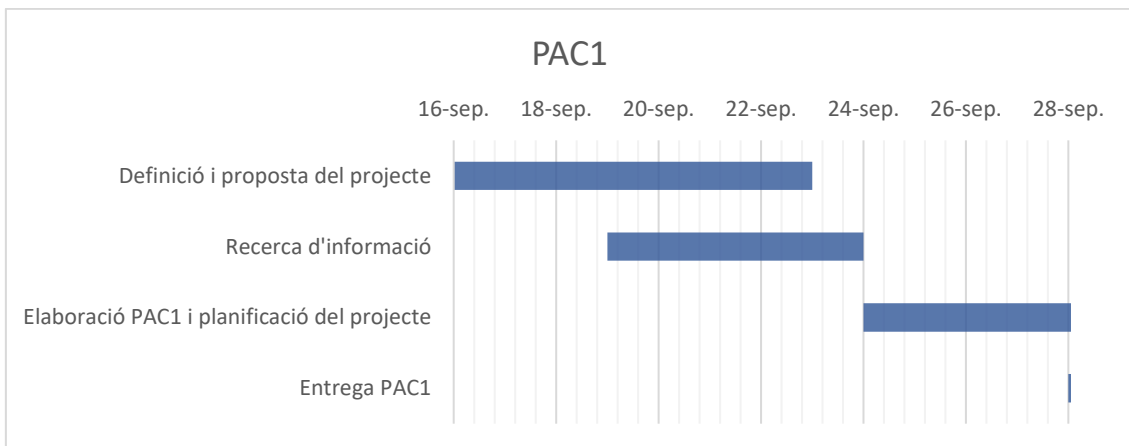


Figura 1: Planificació PAC1 Diagrama de Grantt

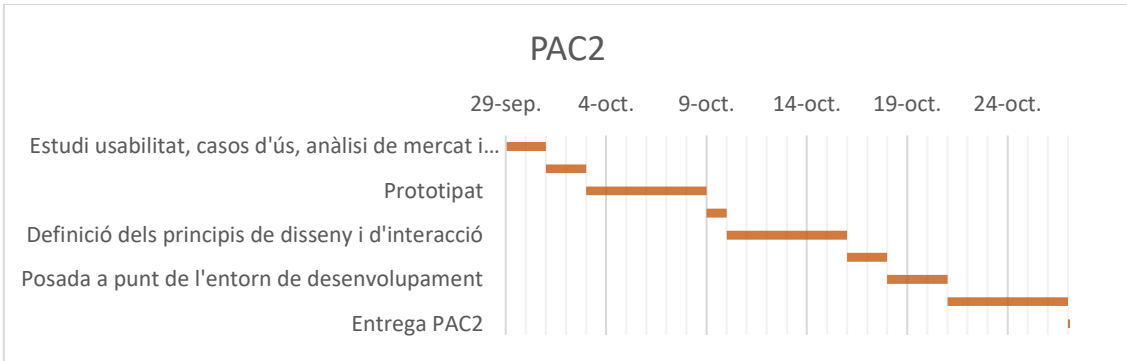


Figura 2: Planificació PAC2 Diagrama de Grantt

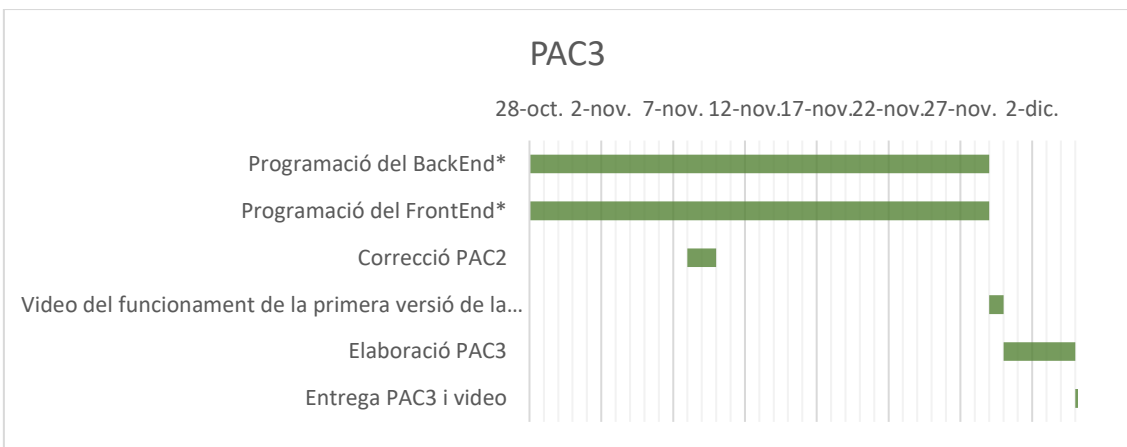


Figura 3 Planificació PAC3 Diagrama de Grantt

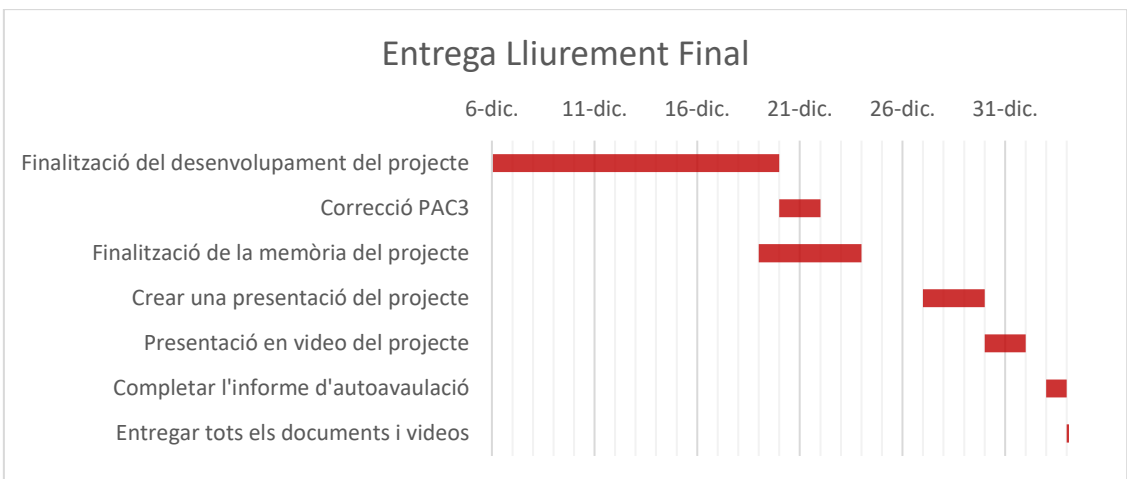


Figura 4: Planificació Lliurement Final Diagrama de Grantt

2. Arquitectura de l'aplicació i plataforma de desenvolupament

L'aplicació es distribueix en tres parts diferents: la base de dades, backend i frontend.

Pel que fa a la base de dades, s'utilitza el llenguatge MySQL, pel backend s'utilitza PHP i pel frontend s'utilitza HTML5 juntament amb Bootstrap, CSS3 i Typescript.

Com a plataforma de desenvolupament s'utilitza un ordinador de sobretaula de 16GB de memòria RAM, Windows 10 i un processador Intel Core i7 amb Windows 10 de 64bits amb 1Tbyte de memòria al disc dur. També s'utilitzen dos monitors, un de 24 polzades i un de 22.

Pel que fa al software de desenvolupament de l'aplicació, s'utilitzarà Laravel pel backend, Angular pel frontend i phpmyAdmin per la base de dades utilitzant el servidor local Xampp per desenvolupar l'aplicació localment en fase de proves. Per instal·lar l'aplicació formalment s'utilitzarà GIT i Netlify pel frontend i Heroku, pel backend de Laravel per gestionar la base de dades, i també pel servidor a Node dels WebSockets. Per gestionar la base de dades amb format MySQL desde Heroku hauré d'utilitzar ClearDB i MySQLWorkbench.

A Angular hi afegiré diferents característiques per poder optimitzar l'aplicació:

- Instal·lació de Redux.
- Instal·lació de Router Store
- Instal·lació de Redux dev-tools
- Instal·lació de Redux Effects
- Utilitzaré ReactiveForms pels formularis

A part d'aquestes característiques, també utilitzaré els diferents recursos externs:

- Bootstrap 5: Per aplicar estils a l'aplicació i fer-la responsive
- Fontawsome: Per utilitzar icones
- XLSX: Per exportar a Excel
- Ngx-print: Per imprimir directament des de l'aplicació
- Ng-bootstrap: En aquest cas per fer ús dels modals de bootstrap
- Gs1-barcode-parser: Llibreria externa per descodificar els codis de barres de format GS1-128
- Socket.io: Per crear el servei de WebSockets i crear les alertes en temps real
- Angular-input-focus: Per seleccionar automàticament el camp d'input necessari per llegir els codis de barres

- Service-worker: Per seguir el procés de l'aplicació i convertir-la a PWA(Progressive Web App)

Com a aspecte important de funcionament de l'aplicació, cal destacar que s'ha generat codi als components del projecte d'angular per tal d'adaptar l'aplicació a la utilització de la llibreria externa de descodificació Gs1-barcode-parser per tal d'adaptar-la a l'ús personal.

Així, l'aplicació també és capaç d'identificar els diferents dígitos dels codis de barres i pot identificar si l'etiqueta representa un palet amb la quantitat estàndard entrada a la referència o si, en canvi, es tracta d'un palet amb una quantitat diferent. L'aplicació és capaç d'identificar la quantitat i entrar-la al sistema.

A part d'això, també s'han utilitzat uns AuthGuards per assegurar que usuaris no autoritzats puguin accedir a parts concretes de l'aplicació.

3. Diagrames UML

3.1 Diagrames de flux

En els següents diagrames de flux es poden veure totes les possibilitats d'accions que té un usuari segons quina sigui la seva intenció i interacció amb el programa. Es contemplen les diferents possibilitats que hi ha i es donen solucions per a cada una d'elles.

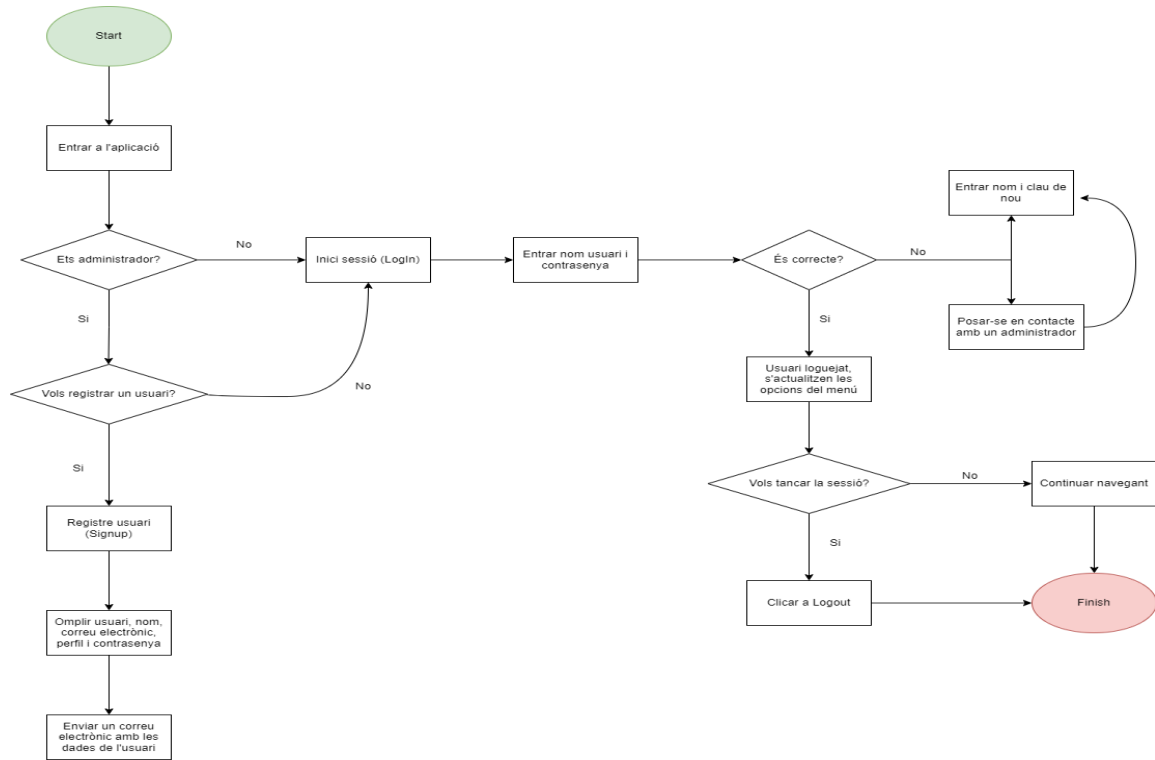


Figura 5: Diagrama de flux: Gestió d'usuaris

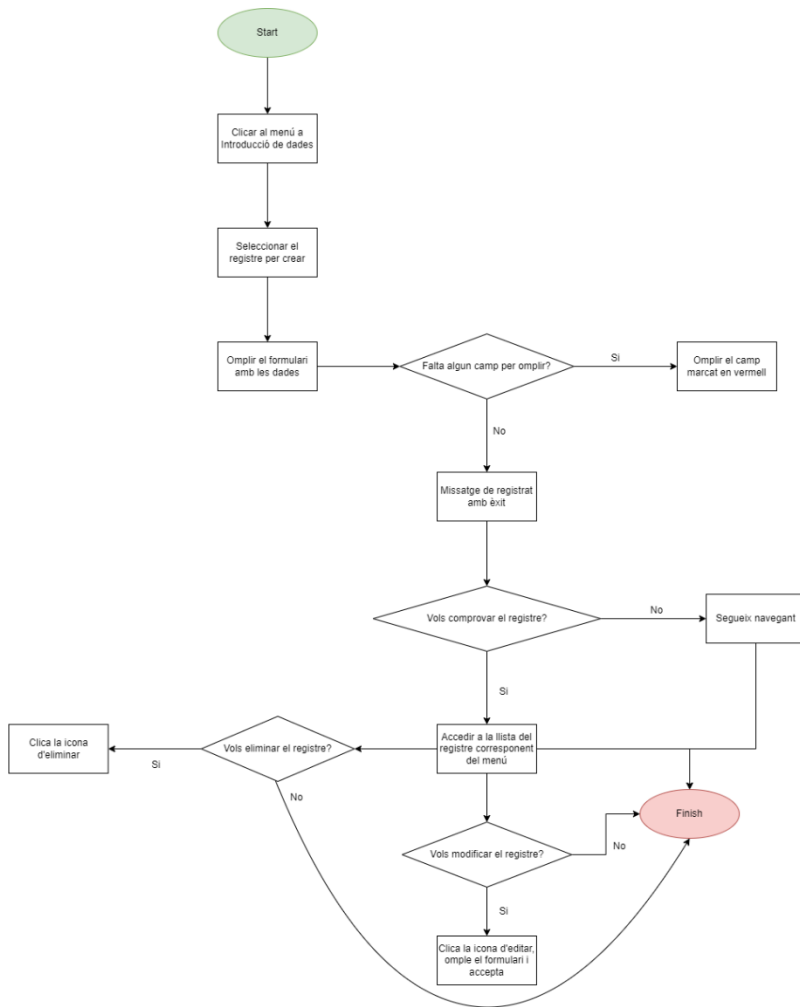


Figura 6: Diagrama de flux: Entrada de dades

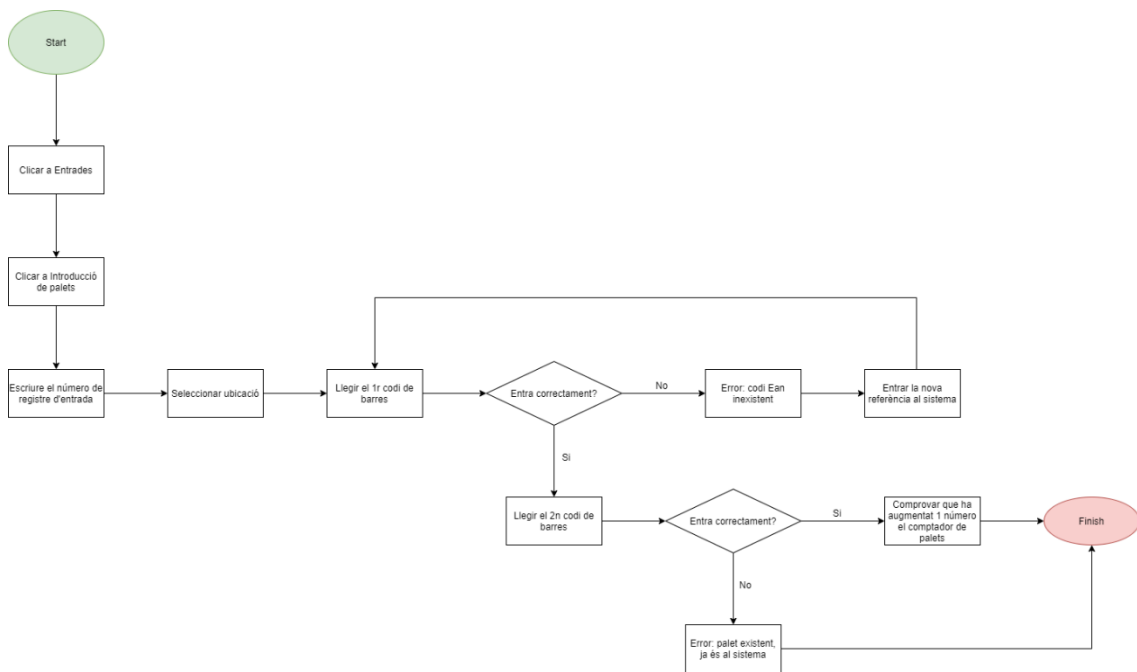


Figura 7: Diagrama de flux: Entrada de palets

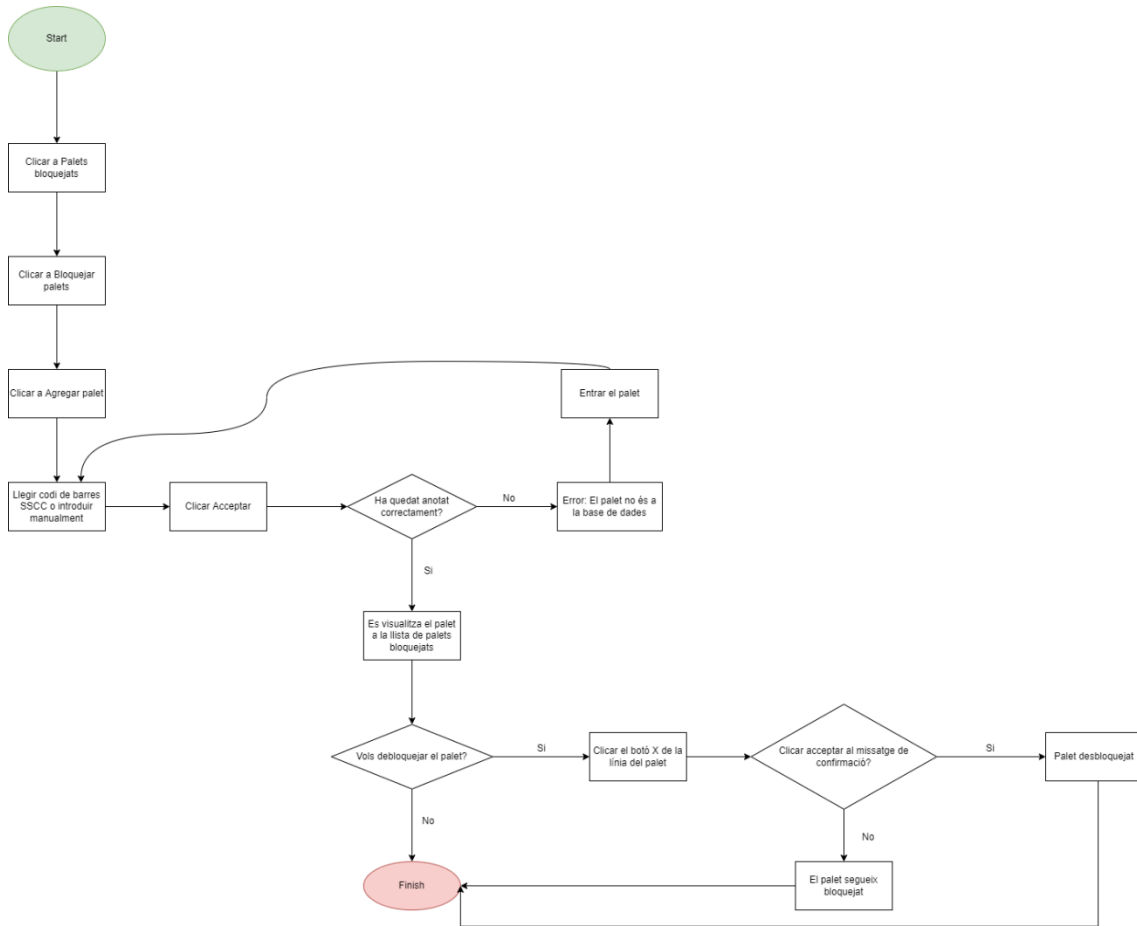


Figura 8: Diagrama de flux: Bloquejar palets

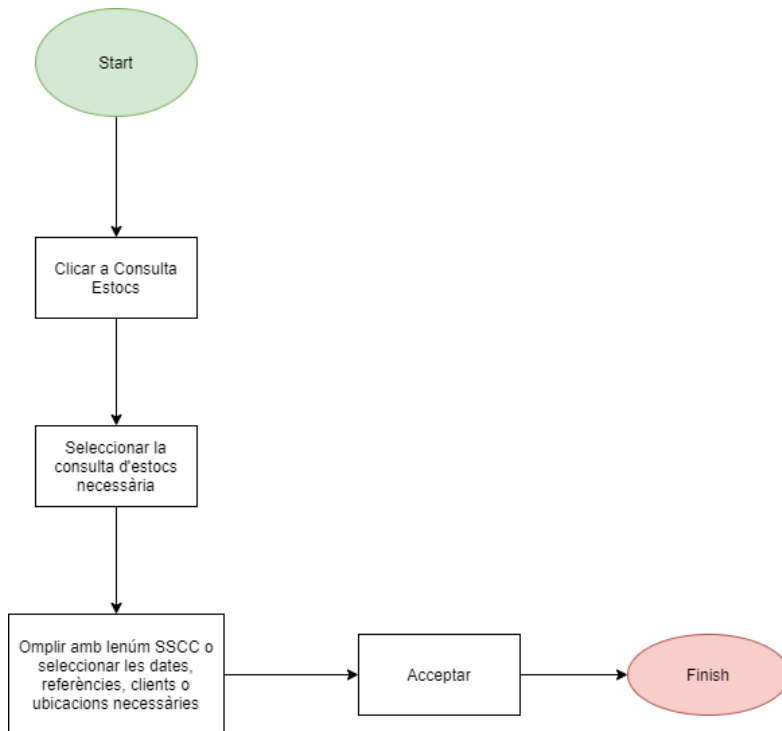


Figura 9: Diagrama de flux: Consulta estocs

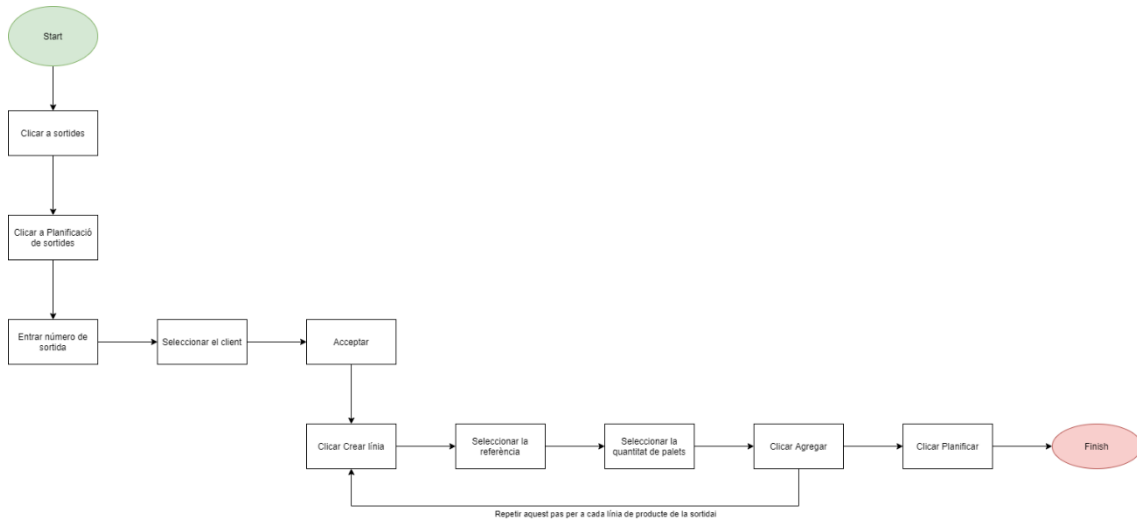


Figura 10: Diagrama de flux: Planificació de sortides

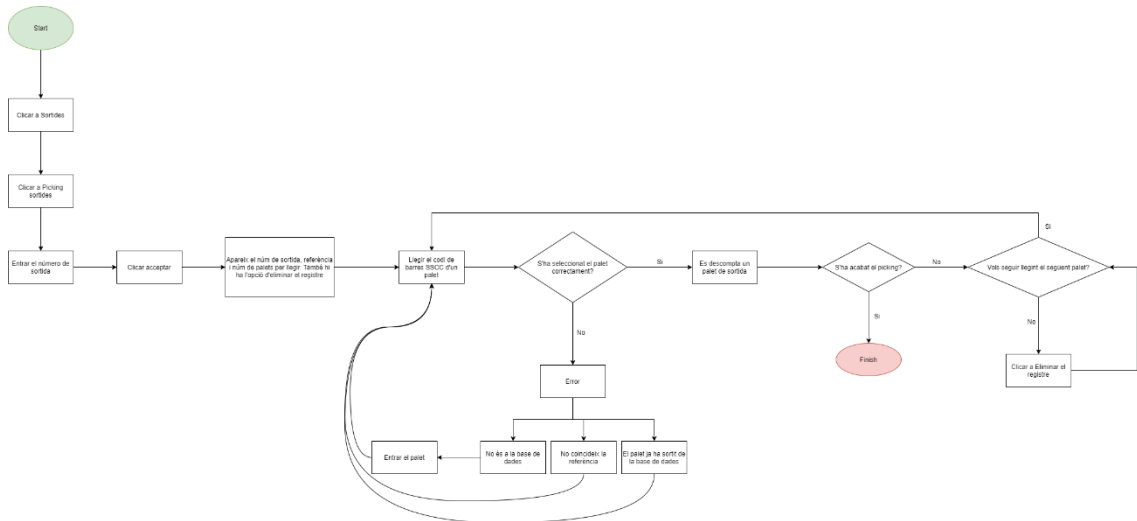


Figura 11: Diagrama de flux: Sortida de palets

3.2 Diagrama Base de dades

En el diagrama següent es mostra la distribució i connexions entre les diferents taules que componen la base de dades de l'aplicació. Hi ha una taula principal que és la que reuneix més connexions i contindrà l'element principal de l'aplicació, que en aquest cas seran els palets emmagatzemats al magatzem. La resta de taules tenen relacions més discretes i acabaran compostant l'entorn perquè estiguin tots els processos connectats.

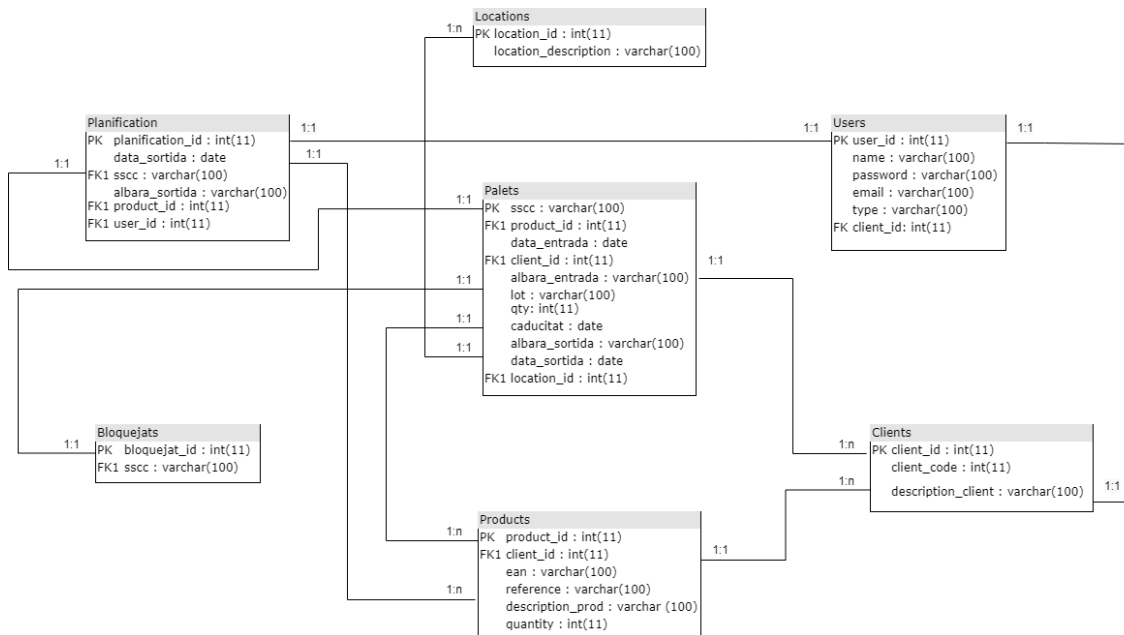


Figura 12: Diagrama de la Base de Dades

3.3 Usuaris

S'han definit tres rols d'usuari diferents que faran d'usuaris de l'aplicació amb diferents rols cadascun d'ells: administrador, magatzem i client.

3.3.1 Administrador

Aquest rol és el que tindran els responsables. Aquest rol s'encarregarà d'introduir les dades de les noves referències, clients o ubicacions, així com també de crear i donar els permisos als usuaris segons el seu rol. A part d'això també podrà planificar les sortides i bloquejar i desbloquejar palets. També podrà accedir als estocs i llegir d'entrada i sortida els palets. Aquest és l'usuari universal, pot accedir a totes les opcions.

3.3.2 Magatzem

Aquest rol és el que tindrà el personal de magatzem que s'encarregarà de la lectura dels palets a l'entrada i la sortida. També podrà consultar els estocs de producte i les seves ubicacions.

3.3.3 Client

Aquest rol és el que tindran els clients, i només podran consultar els seus estocs, els que pertanyin als seus productes.

3.4 Casos/escenaris d'ús

Segons els tres tipus d'usuaris hi ha diferents tipus d'escenaris d'ús segons la funcionalitat que es vulgui portar a terme. Es farà un exemple d'escenari diferent per a cada funció.

Cas/Escenari 1

Usuari	Administrador
Objectiu	Introduir dades per a una nova referència
Desenvolupament de la tasca	Escriure totes les dades de la referència, client o ubicació i acceptar el formulari

L'encarregada rep una mercaderia amb una referència nova que no és a la base de dades. Si és al magatzem pot utilitzar un dels terminals per accedir al formulari d'introducció de referències noves. Introdueix les dades llegint l'etiqueta del producte nou i accepta el formulari. Si, en canvi, vol utilitzar un ordinador, ho pot fer des del despatx fent abans una fotografia a l'etiqueta del producte per tenir presents les dades que ha d'introduir. També pot accedir directament a l'aplicació des del telèfon mòbil.

Taula 5: Cas/escenari d'ús 1

Cas/Escenari 2

Usuari	Administrador
Objectiu	Planificar una sortida
Desenvolupament de la tasca	Anotar el número de sortida, seleccionar el client i afegir les línies de producte i les quantitats que han de sortir

El personal d'administració rep un camioner que ve a buscar mercaderia. Anoten la comanda que ve a buscar el camioner i consulten la mercaderia que s'ha d'endur. Una de les persones del despatx accedeix a l'aplicació des de l'ordinador i accedeix a la planificació de sortides. Introdueix el número de comanda de sortida i selecciona el client. Afegeix línia per línia cada referència que compona la comanda seleccionant quina quantitat de producte s'endurà de cada referència. Avisa al personal de magatzem i els proporciona el número de comanda i la llista de producte perquè puguin efectuar la sortida.

Taula 6: Cas/escenari d'ús 2

Cas/Escenari 3

Usuari	Client
Objectiu	Consultar estoc per referència
Desenvolupament de la tasca	Seleccionar la consulta d'estoc per referència i seleccionar la referència

El Client vol saber quants palets li queden d'una de les referències i quines dates o lots. Selecciona l'opció del menú Estoc per referència i selecciona el client i la referència que necessita consultar. Li apareix un llistat amb totes les diferents dates, lots i quantitat de palets que té emmagatzemats d'aquesta referència. Pot exportar-les a excel i/o imprimir els registres.

Taula 7: Cas/escenari d'ús 3

Cas/Escenari 4

Usuari	Magatzem
Objectiu	Llegir els palets d'una entrada de producte
Desenvolupament de la tasca	Introduir el número d'entrada i llegir els codis de barres per entrar cada palet

Administració informa al personal de magatzem que hi ha un camió per descarregar i els dona el número de referència d'entrada. El personal de magatzem selecciona l'opció d'introduir palets, anota el número d'entrada i selecciona la ubicació on posaran la mercaderia. D'un a un can llegint els dos codis de barres de cada palet per introduir-ne les dades. S'asseguren que els palets han estat entrats correctament observant que el número de palets de l'entrada augmenta a la pantalla.

Taula 8: Cas/escenari d'ús 4

Cas/Escenari 5

Usuari	Administrador
Objectiu	Bloquejar palets
Desenvolupament de la tasca	Anotar el número de SSCC del palet a la pestanya de bloquejar palets

El client envia per correu electrònic una llista de palets que s'han de bloquejar a administració. Administració obre l'aplicació des de l'ordinador, selecciona la opció de bloquejar palets i hi agrega els palets amb els SSCC que hi ha a la llista que ha enviat el client. A partir d'ara, el palet apareix a la llista de palets bloquejats, i el sistema no deixarà llegir el palet en una sortida, marcarà un error conforme el palet està bloquejat.

Taula 9: Cas/escenari d'ús 5

Cas/Escenari 6

Usuari	Administrador
Objectiu	Consultar les entrades que hi ha hagut un dia concret
Desenvolupament de la tasca	Accedir a la consulta d'entrades, introduir les dates d'entrada i seleccionar cada número d'entrada per fer la consulta

Administració necessita saber quantes entrades s'han fet de producte un dia concret i quants palets de cada referència portava cada camió. El personal d'administració obre l'aplicació, selecciona l'opció de consulta d'entrades per data i selecciona l'interval de dies, en aquest cas selecciona la mateixa data en els dos camps perquè vol saber les entrades d'un dia concret. Apareix el llistat de totes les entrades del dia seleccionat i clica una per una cada entrada i s'obre un modal amb el detall amb les dades dels palets que s'han introduït a cada entrada.

Taula 10: Cas/escenari d'ús 6

4. Prototips

4.1 Wireframes

4.1.1 Home

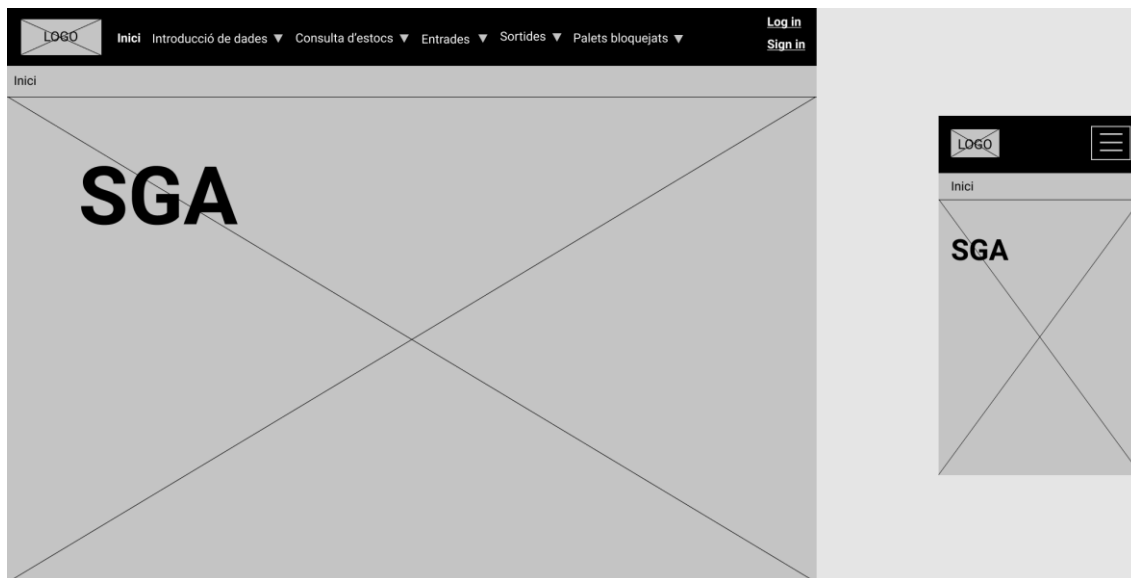


Figura 13: Wireframe Inici

4.1.2 Registres: *clients, articles, ubicacions*

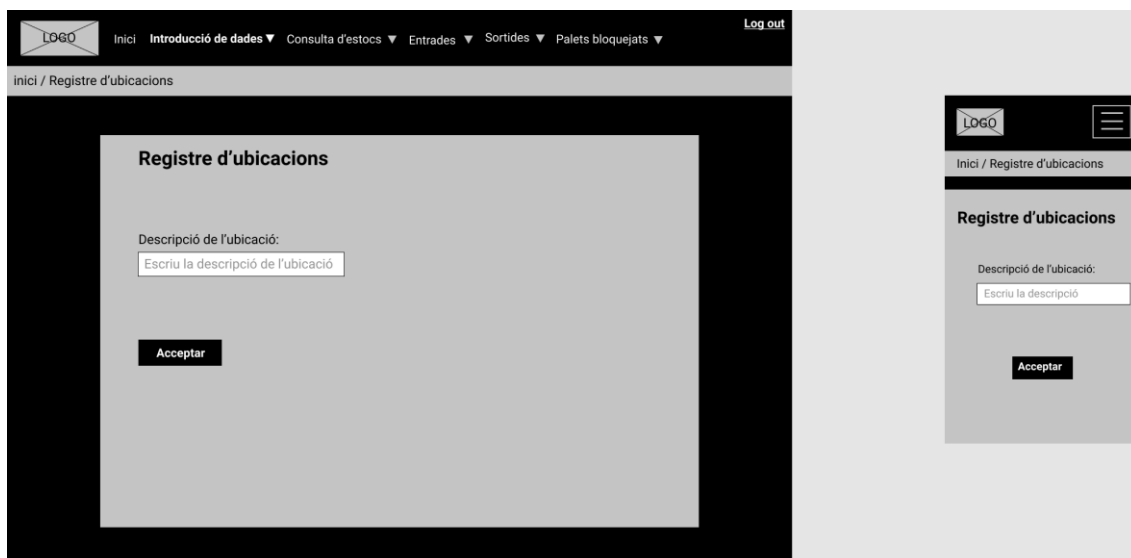


Figura 14: Wireframe Regsitre ubicacions

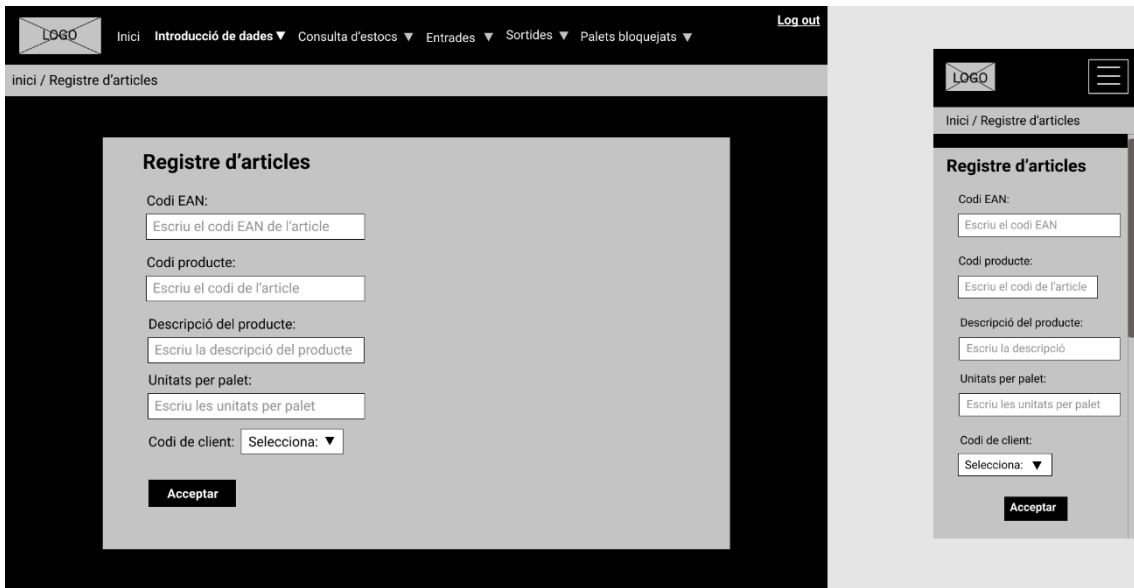


Figura 15: Wireframe Registre articles

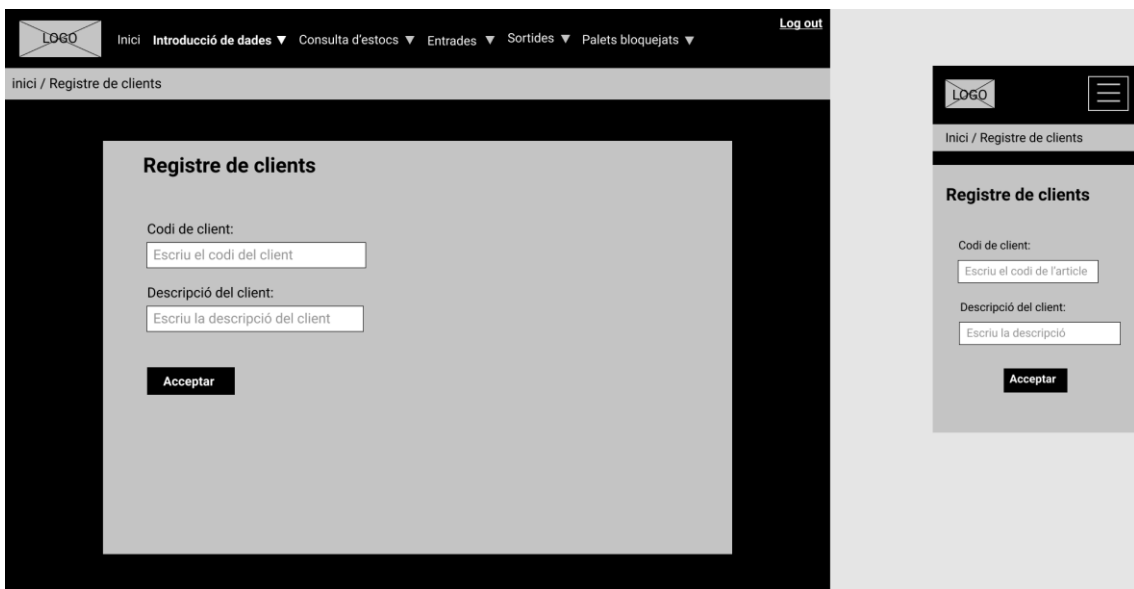


Figura 16: Wireframe Registre clients

4.1.3 Registres: clients, articles, ubicacions

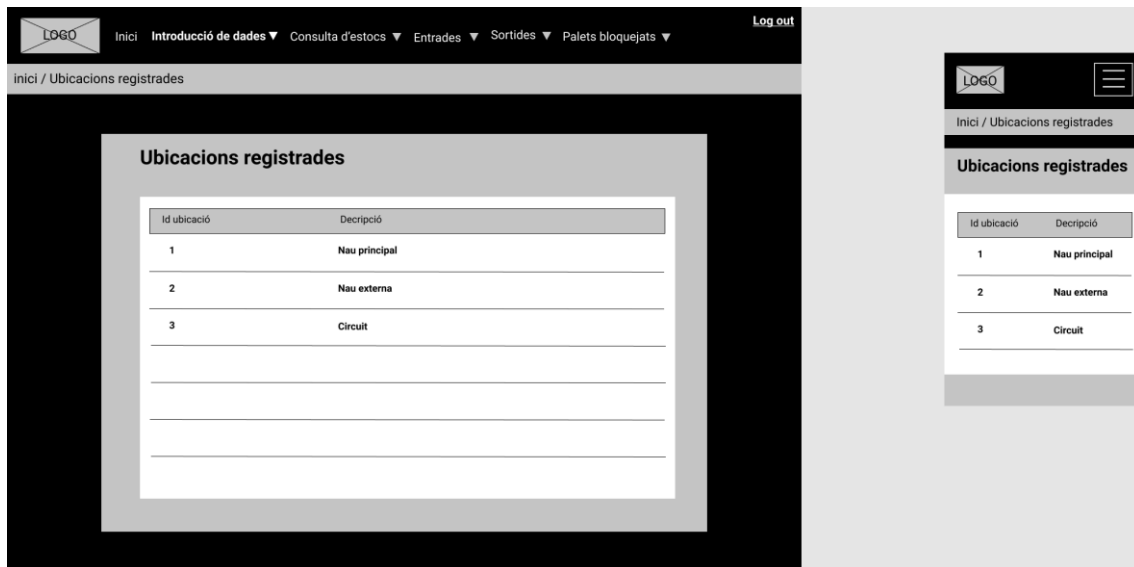


Figura 17: Wireframe Ubicacions registrades

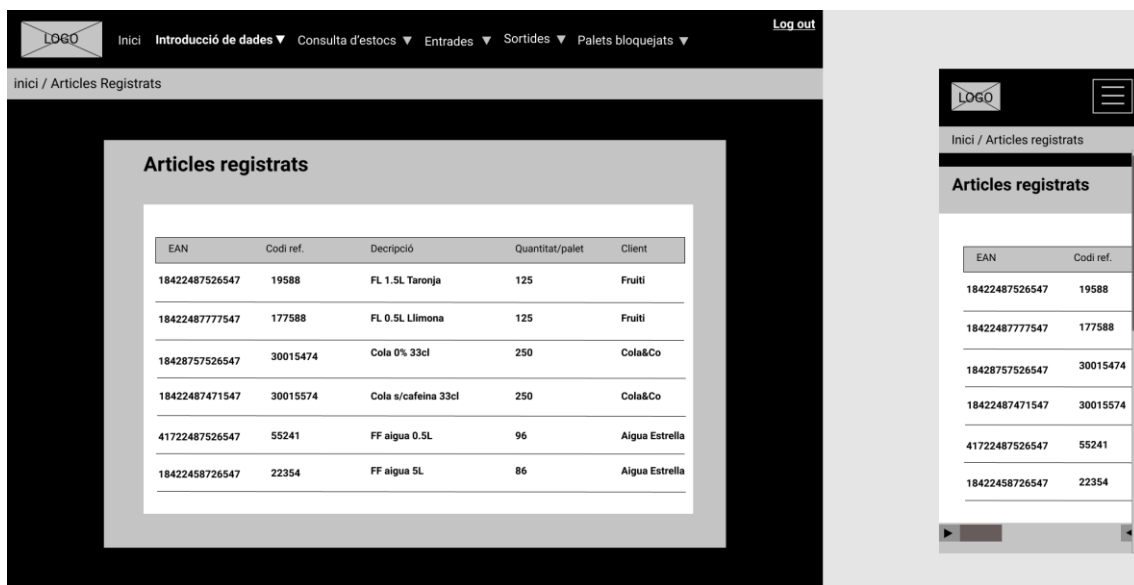


Figura 18: Wireframe Articles registrats

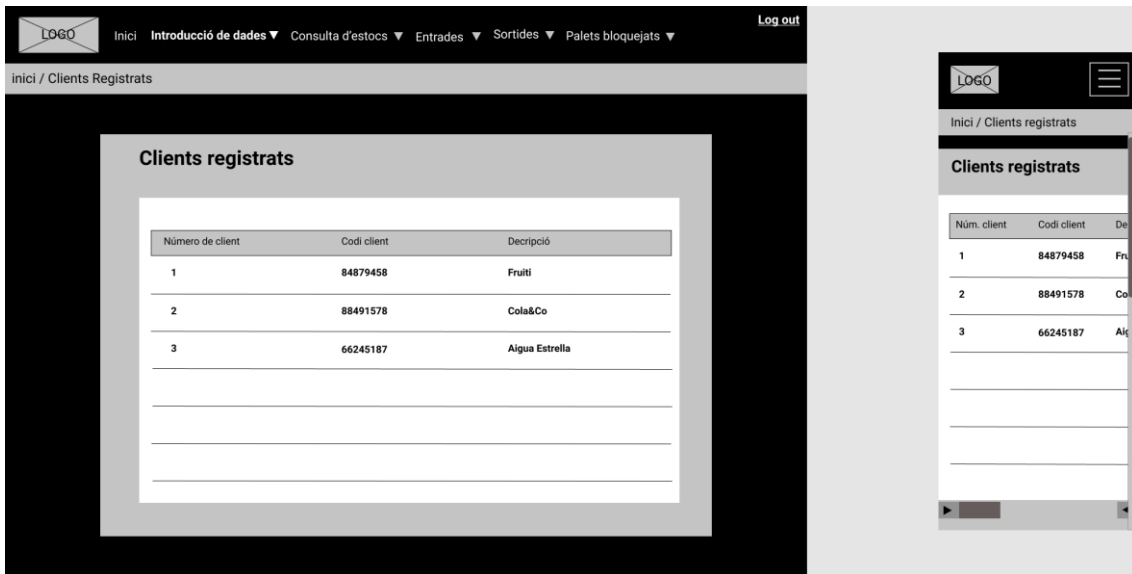


Figura 19: Wireframe Clients registrats

4.1.4 Introducció de palets

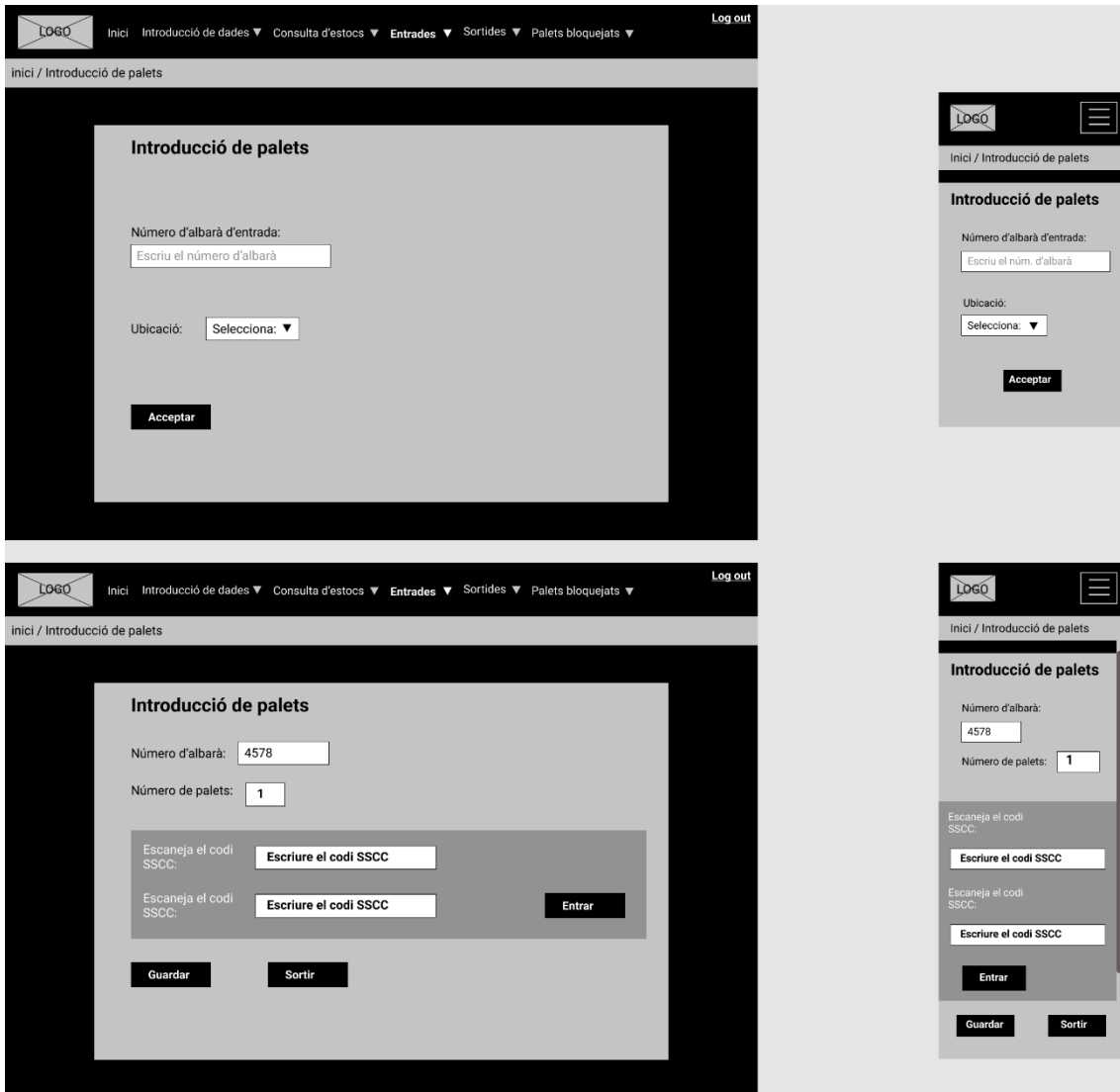


Figura 20: Wireframe Introducció de palets

4.1.5 Consulta d'entrades per data

The wireframe illustrates the 'Consulta d'entrades per data' (Query by date) interface in three stages:

- Selection Stage:** The user selects a start date ('Selecciona una d'inici: 00/00/0000') and an end date ('Selecciona una data final: 00/00/0000'). An 'Acceptar' button is present.
- Results Stage:** A table displays the query results. The total number of pallets is 130. A 'Tornar enrere' button is at the bottom.
- Details Stage:** A modal window titled 'SSCC palets entrada - 125' shows detailed information for a specific pallet, including SSCC, client, product, lot, and expiration date. It includes 'Exportar a Excel' and 'Imprimir' buttons.

Navigation and UI Elements: The top navigation bar includes 'LOGO', 'Inici', 'Introducció de dades', 'Consulta d'estocs', 'Entrades', 'Sortides', 'Palets bloquejats', and 'Log out'. The breadcrumb trail is 'inici / Consulta d'entrades per data'.

Table 1: Query Results

Número d'entrada	Data d'entrada	Client	Número de palets	Detalls
125	13/10/2021	Fruiti	25	Detalls
126	13/10/2021	Fruiti	25	Detalls
127	13/10/2021	Fruiti	25	Detalls
128	13/10/2021	Fruiti	25	Detalls
2	13/10/2021	Cola & Co	30	Detalls

Total núm. de palets: 130

Table 2: Detailed View (SSCC palets entrada - 125)

Núm. palet	Núm. entrada	Data entrada	SSCC	Client	Producte	Lot	Caducitat
1	125	13/10/2021	377101283928457845	Fruiti	FL 1.5L Taronja	922121	13/12/2022
2	125	13/10/2021	377101283928457478	Fruiti	FL 1.5L Taronja	922121	13/12/2022
3	125	13/10/2021	377101283928457985	Fruiti	FL 1.5L Taronja	922121	13/12/2022
4	125	13/10/2021	377101283928457334	Fruiti	FL 1.5L Taronja	922121	13/12/2022
5	125	13/10/2021	377101283928457744	Fruiti	FL 0.5L Llimona	932128	13/12/2022
6	125	13/10/2021	377101283928457745	Fruiti	FL 0.5L Llimona	932128	13/12/2022
7	125	13/10/2021	377101283928457766	Fruiti	FL 0.5L Llimona	932128	13/12/2022

Figura 21: Wireframe Consulta d'entrades per data

4.1.6 Planificació de sortides

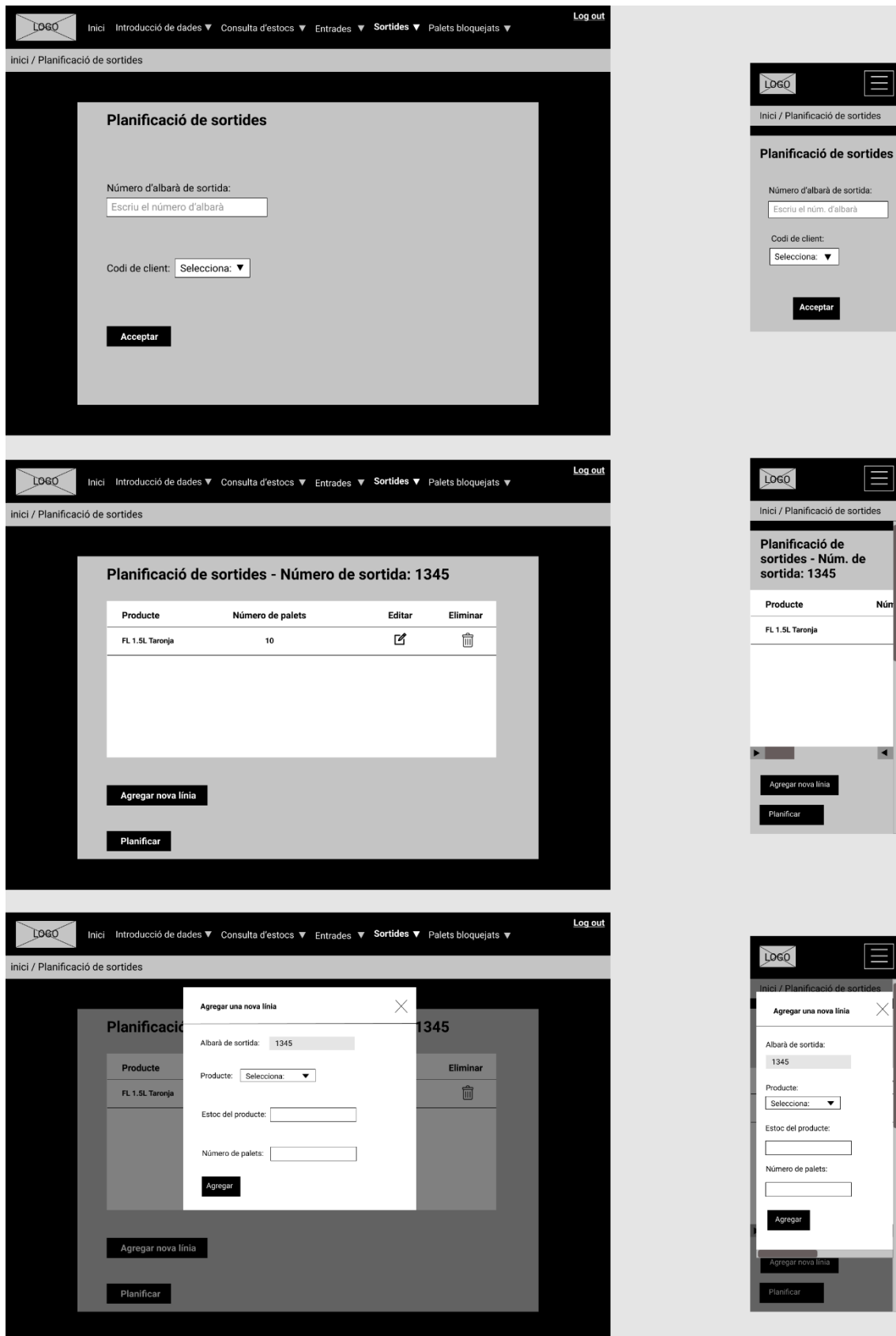


Figura 22: Wireframe Planificació de sortides

4.1.7 Consulta pickings planificats

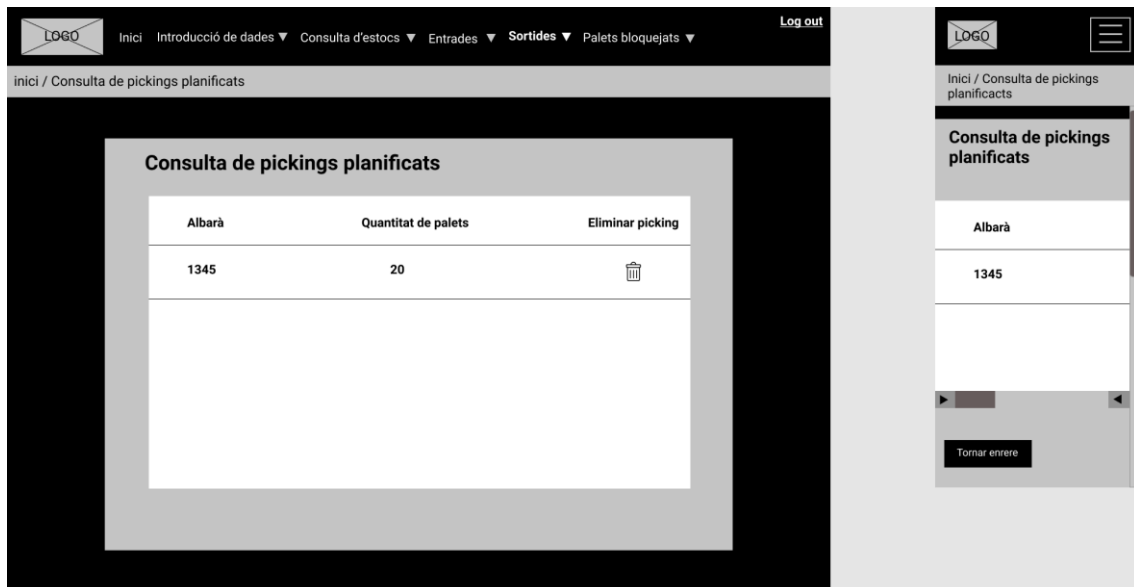


Figura 23: Wireframe Consulta pickings planificats

4.1.8 Picking (lectura) de sortides

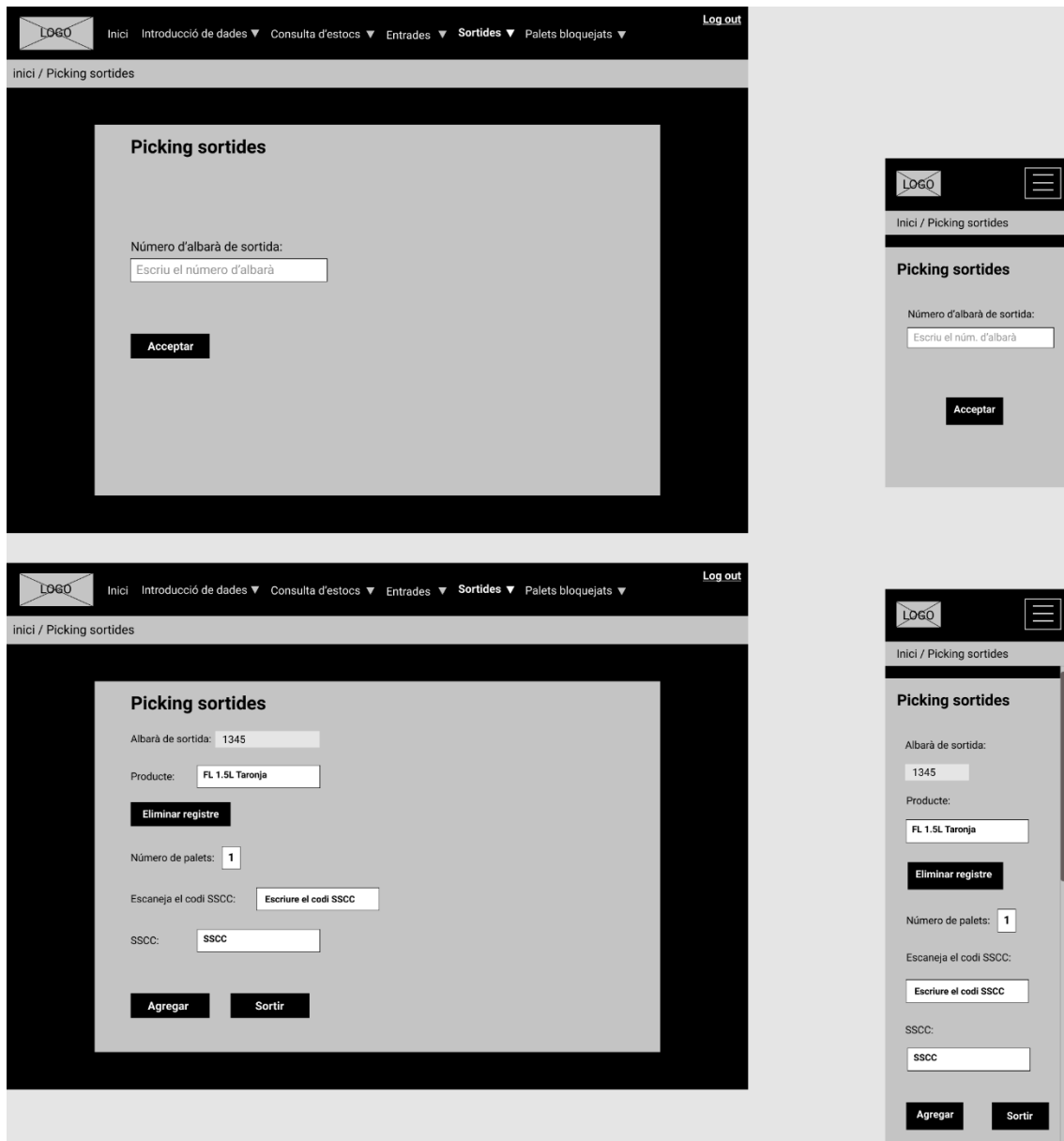


Figura 24: Wireframe Picking sortides

4.1.9 Consulta de sortides per data

The wireframe illustrates the 'Consulta de sortides per data' (Query by date) interface in three stages:

Stage 1: Selection of Date Range

The interface shows a header with a logo, navigation menu (Inici, Introducció de dades, Consulta d'estocs, Entrades, Sortides, Palets bloquejats), and a 'Log out' button. The main content area is titled 'Consulta de sortides per data' and contains two date selection fields:

- Selecció d'inici: 00/00/0000
- Selecció de data final: 00/00/0000

An 'Acceptar' button is located at the bottom.

Stage 2: Summary Table

The interface displays a table with the following data:

Número de sortida	Data de sortida	Client	Número de palets	Detalls
1345	20/10/2022	Fruiti	25	Detalls
1348	20/10/2022	Fruiti	10	Detalls
1350	20/10/2022	Fruiti	10	Detalls
1352	20/10/2022	Fruiti	25	Detalls
889	20/10/2022	Cola & Co	10	Detalls

Below the table, it shows 'Total núm. de palets: 75' and a 'Tornar enrere' button.

Stage 3: Detailed View of Pallet 1345

The interface shows a modal window titled 'SSCC palets sortides - 1345' with 'Exportar a Excel' and 'Imprimir' buttons. The table contains the following data:

Núm. palet	Núm. sortida	Data sortida	SSCC	Client	Producte	Lot	Caducitat
1	1345	20/10/2021	377101283928457845	Fruiti	FL 1.5L Taronja	922121	13/12/2022
2	1345	20/10/2021	377101283928457478	Fruiti	FL 1.5L Taronja	922121	13/12/2022
3	1345	20/10/2021	377101283928457985	Fruiti	FL 1.5L Taronja	922121	13/12/2022
4	1345	20/10/2021	377101283928457334	Fruiti	FL 1.5L Taronja	922121	13/12/2022
5	1345	20/10/2021	377101283928457744	Fruiti	FL 0.5L Llimona	932128	13/12/2022
6	1345	20/10/2021	377101283928457745	Fruiti	FL 0.5L Llimona	932128	13/12/2022
7	1345	20/10/2021	377101283928457766	Fruiti	FL 0.5L Llimona	932128	13/12/2022

A 'Tornar enrere' button is located at the bottom of the modal.

Figura 25: Wireframe Consulta de sortides per data

4.1.10 Consulta Estocs: client, ubicació, producte, dades SSCC, SSCC per producte, albarà, lots

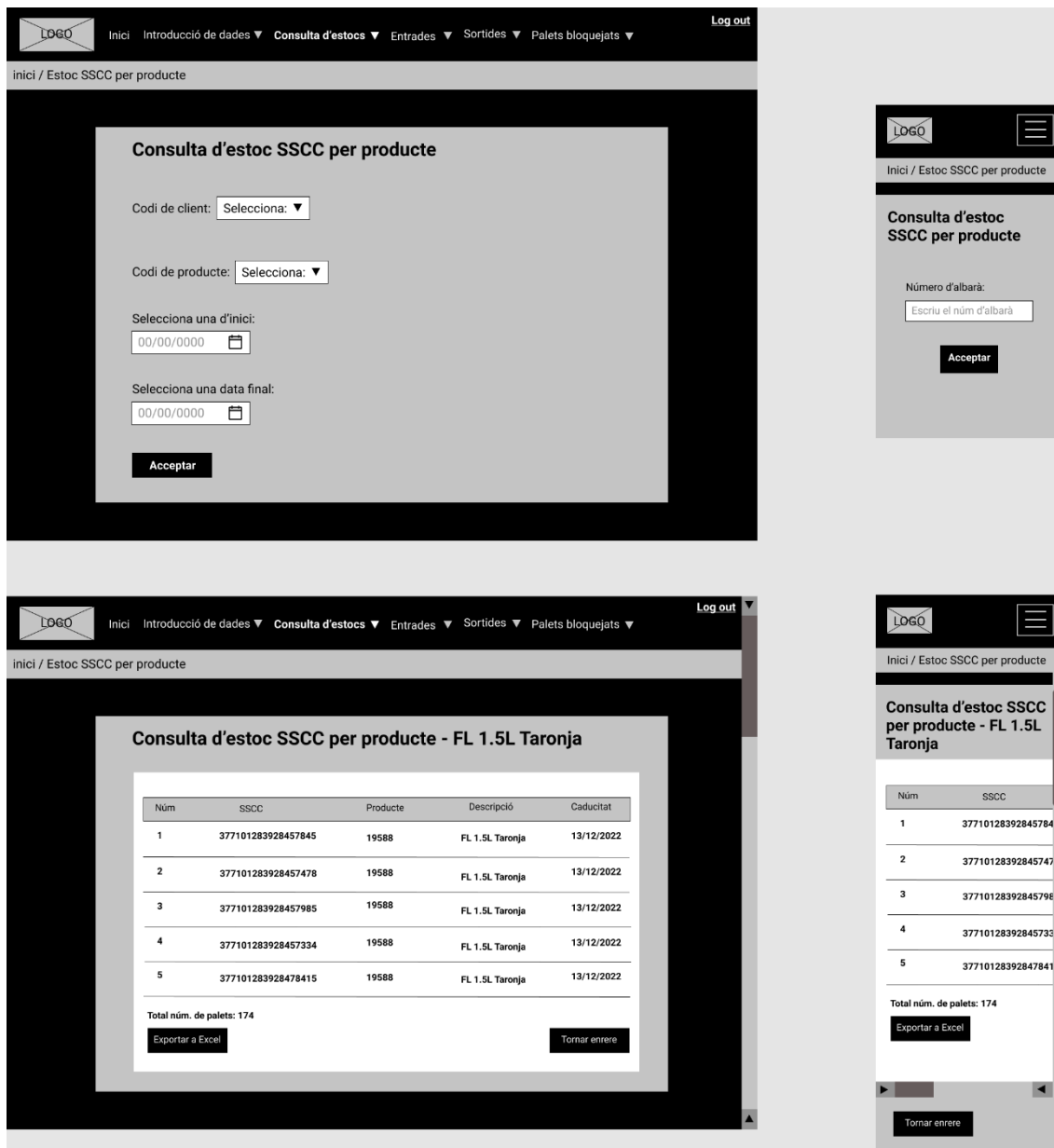


Figura 26: Wireframe Estoc SSCC per producte

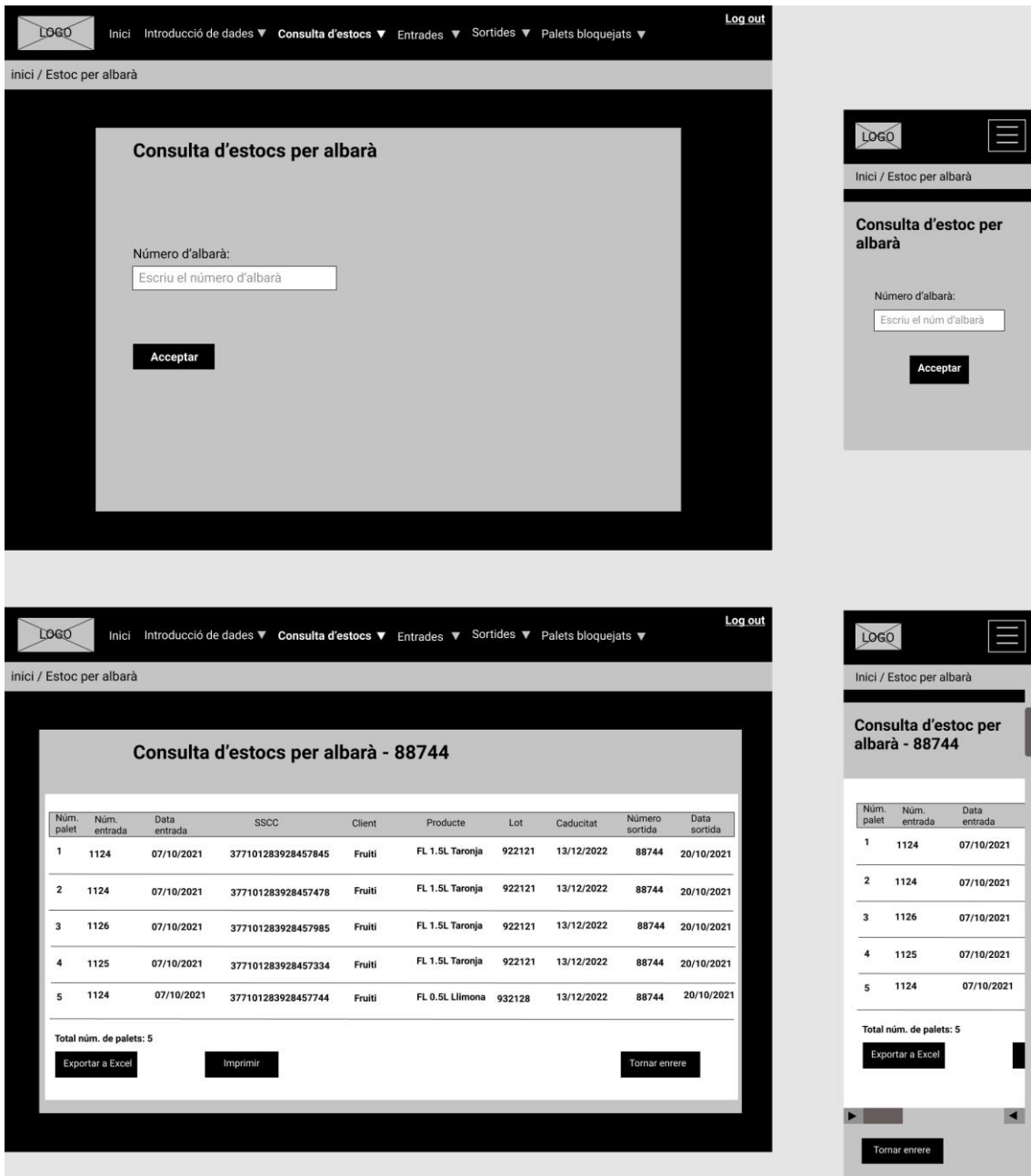


Figura 27: Wireframe Estoc per albarà

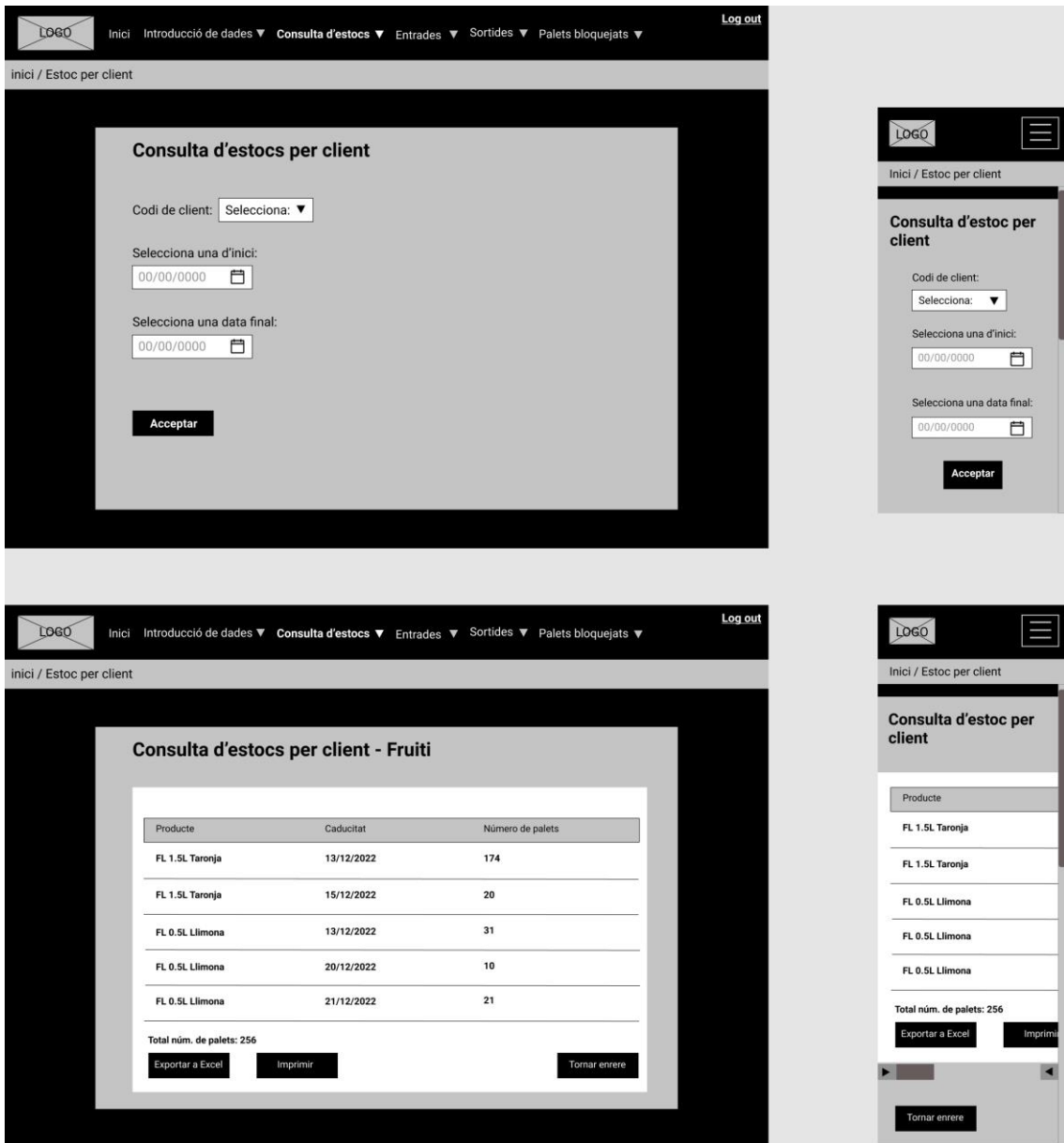


Figura 28: Wireframe Estoc per client

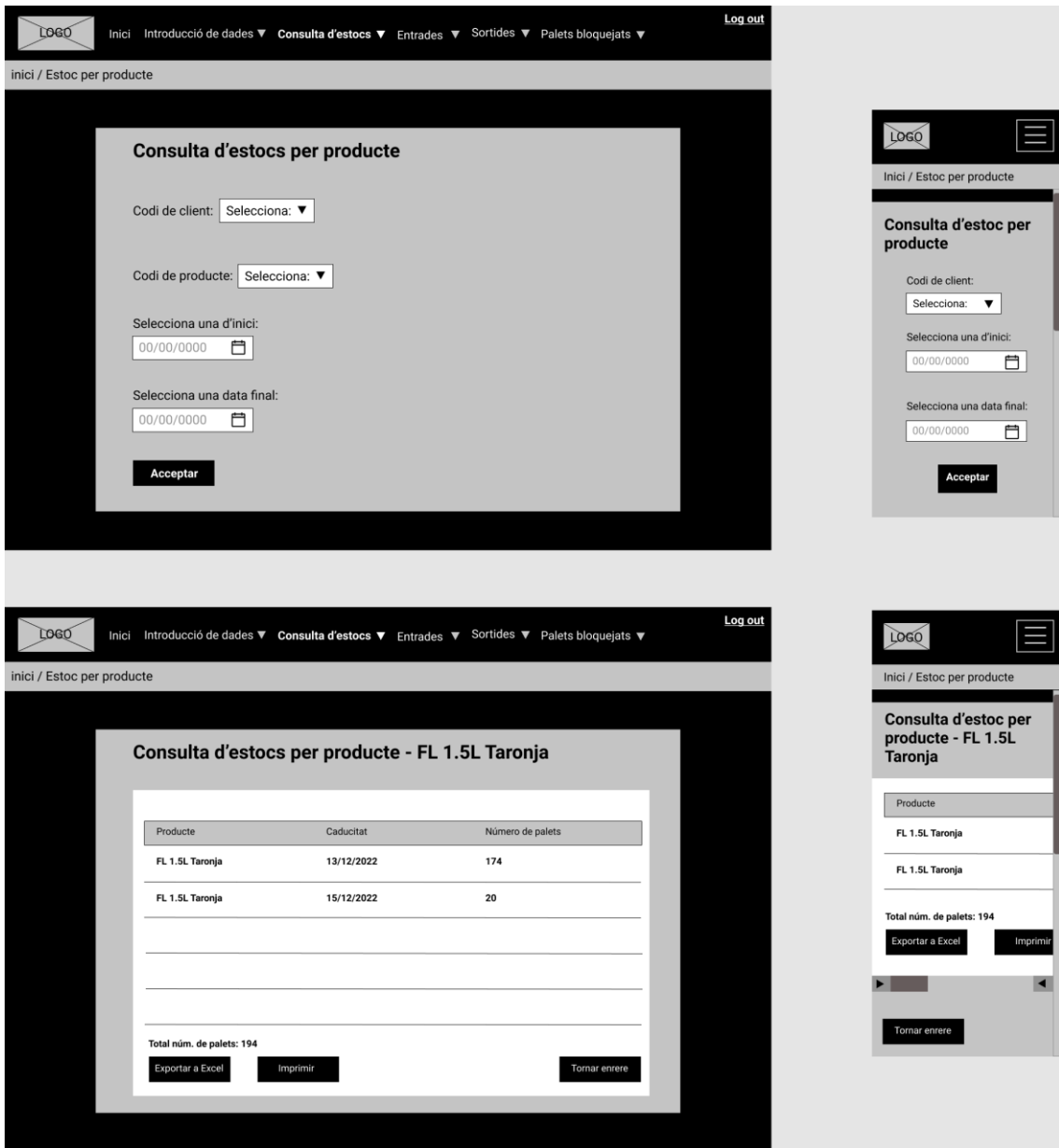


Figura 29: Wireframe Estoc per producte

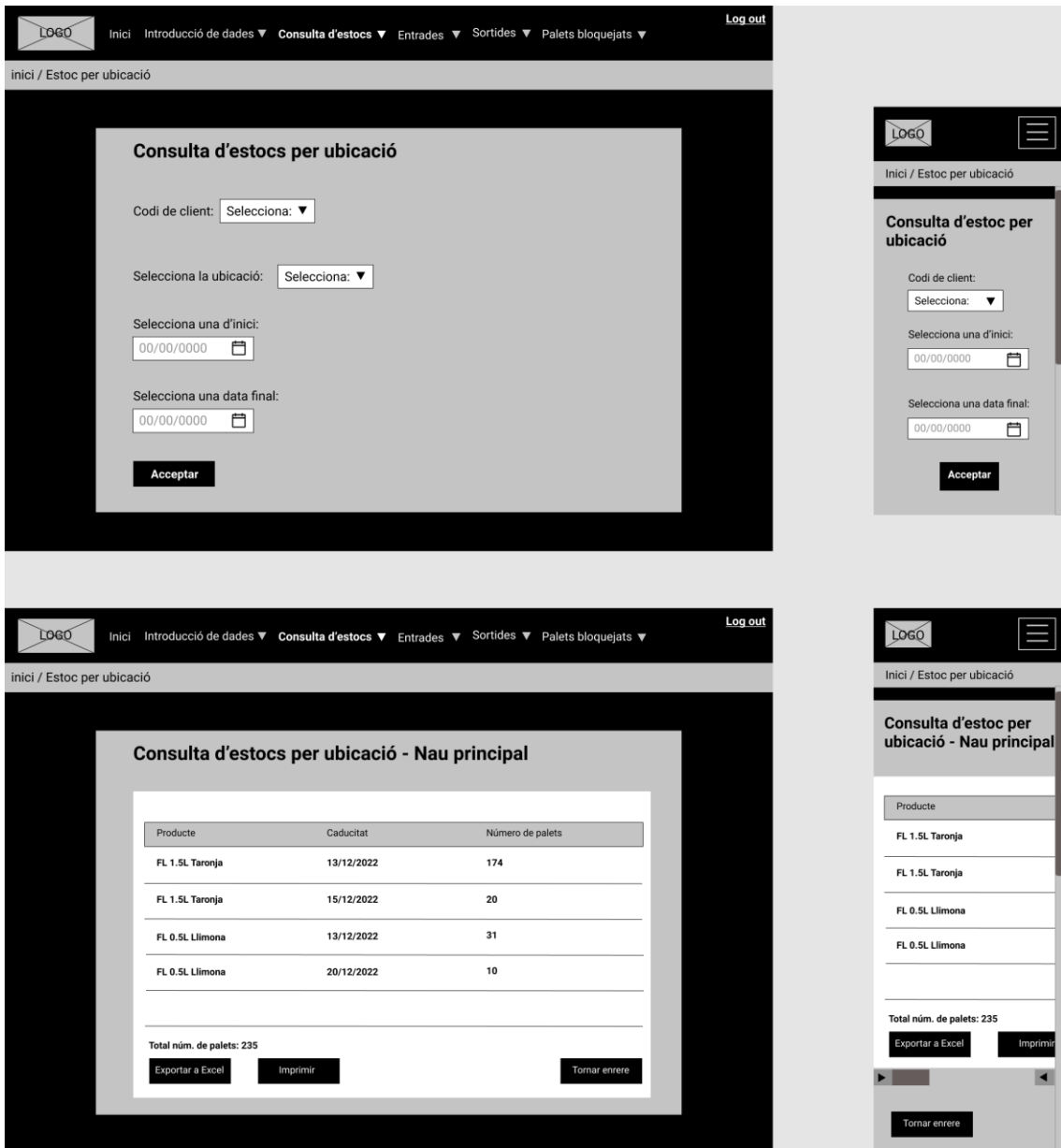


Figura 30: Wireframe Estoc per ubicació

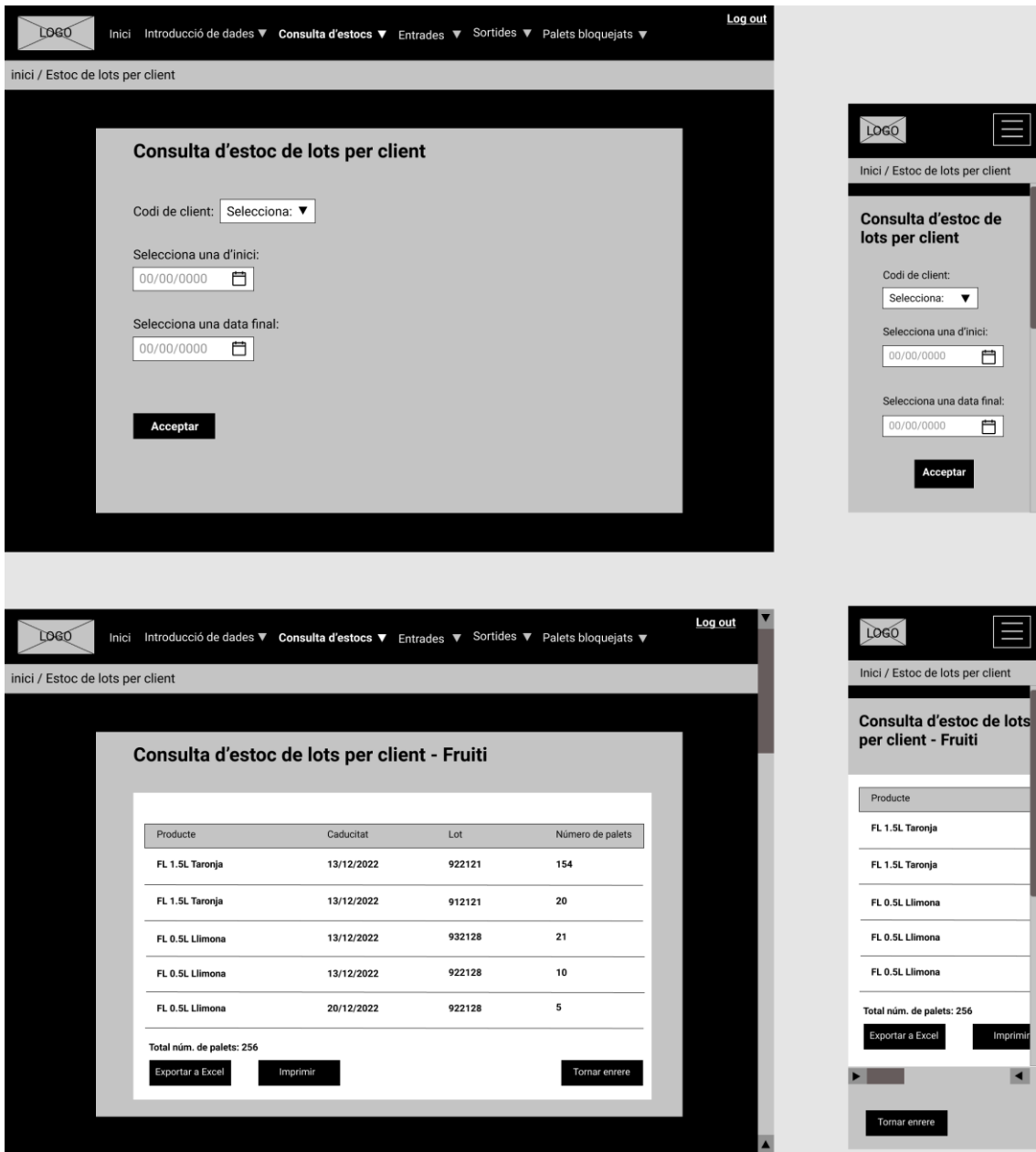


Figura 31: Wireframe Estoc de lots per client

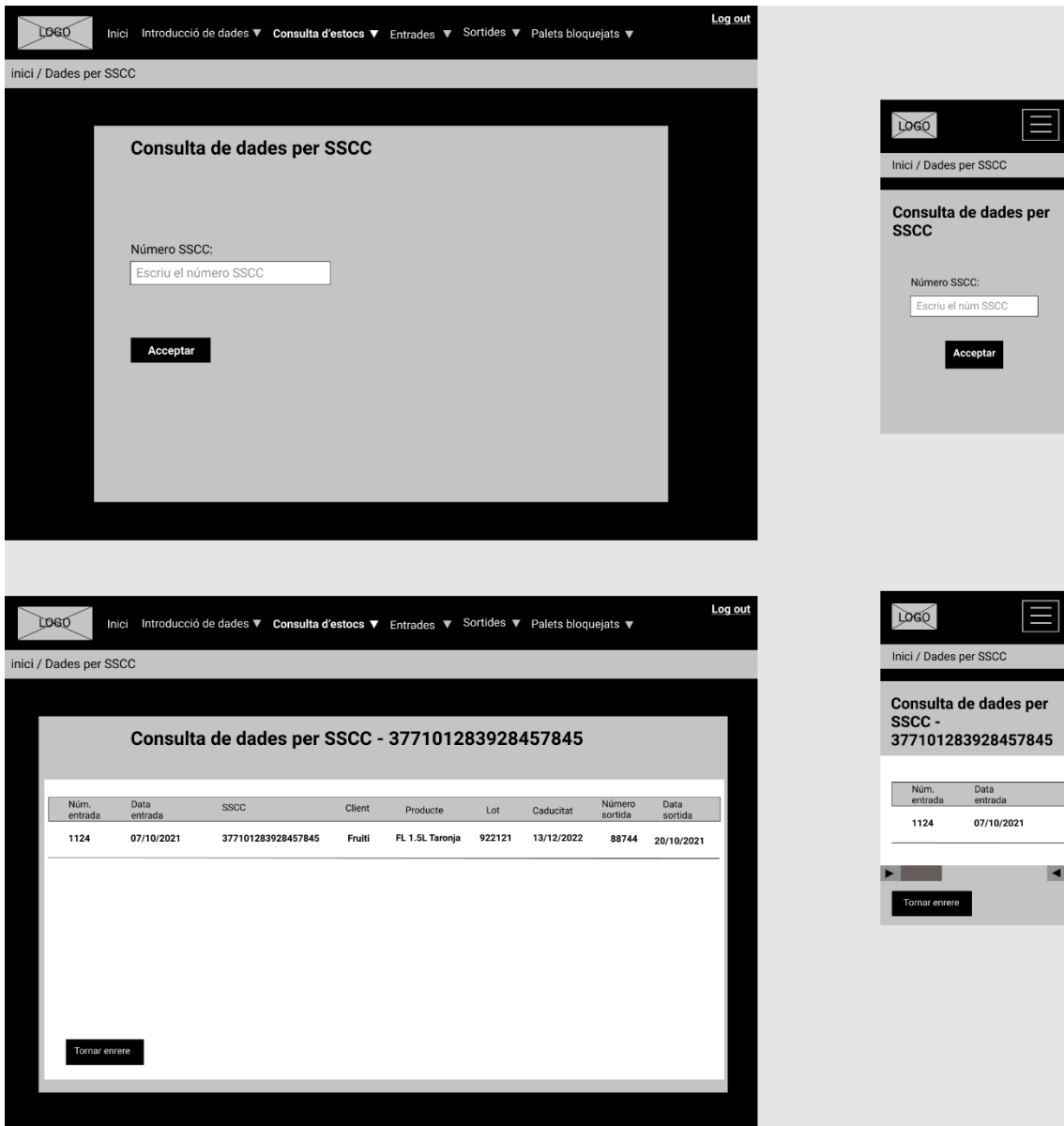


Figura 32: Wireframe Dades per SSCC

4.1.11 Bloquejar i desbloquejar palets

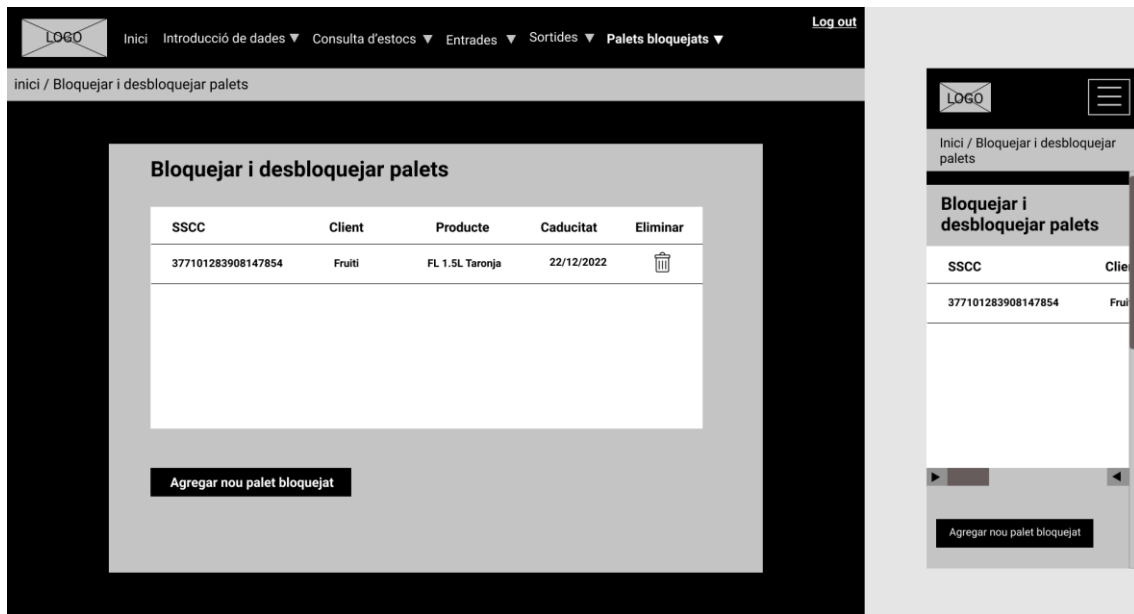


Figura 33: Wireframe Bloquejar i desbloquejar palets

4.1.12 Consulta de palets bloquejats

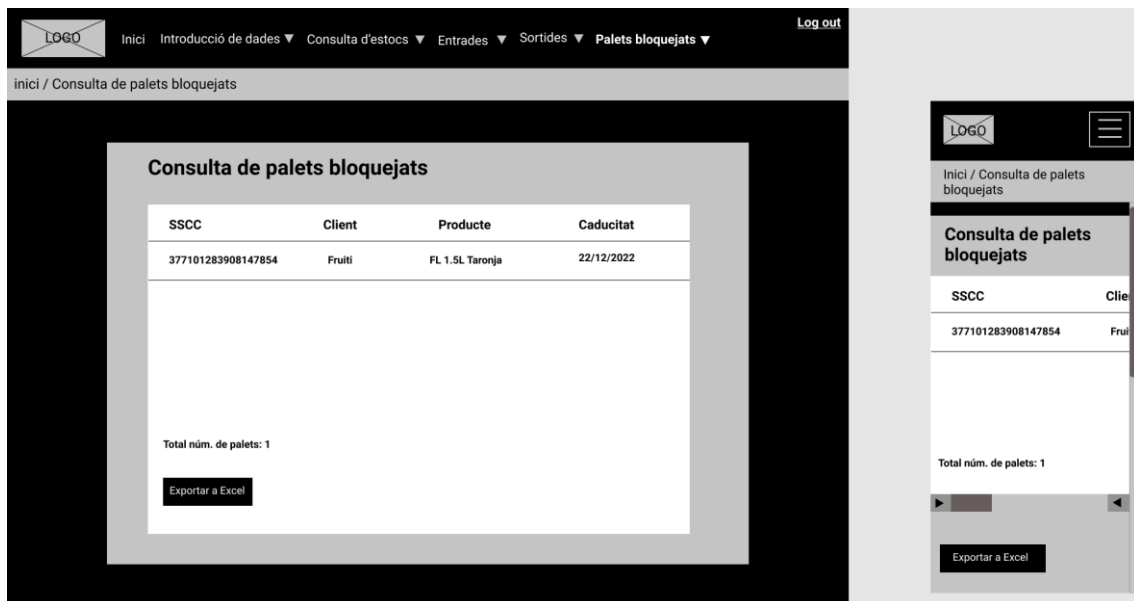


Figura 34: Wireframe Consulta de palets bloquejats

4.1.13 Registre i entrada usuaris (SignIn, LogIn)

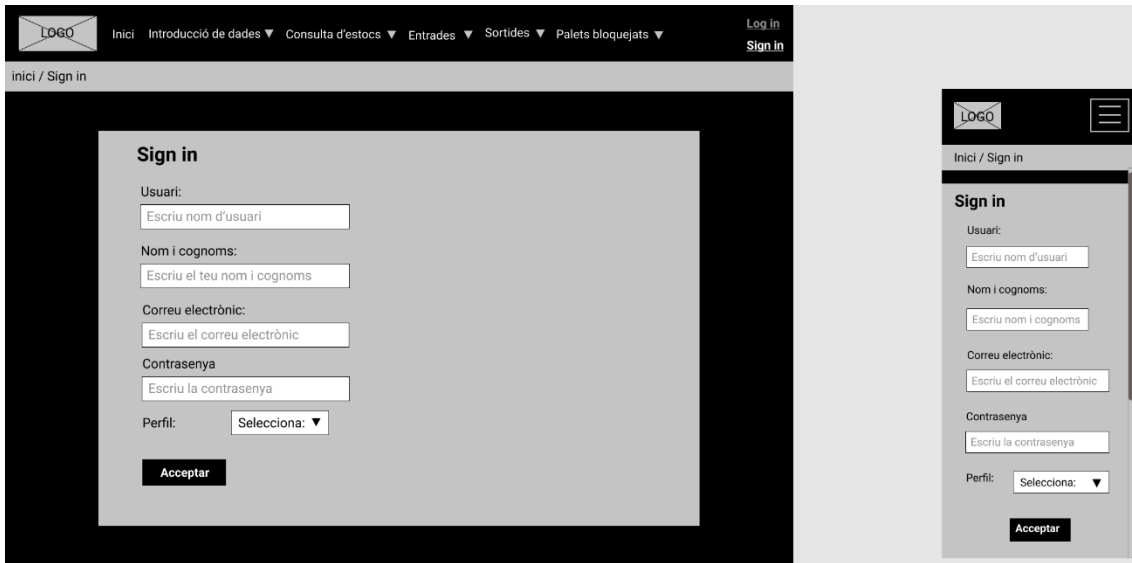


Figura 35: Wireframe SignIn

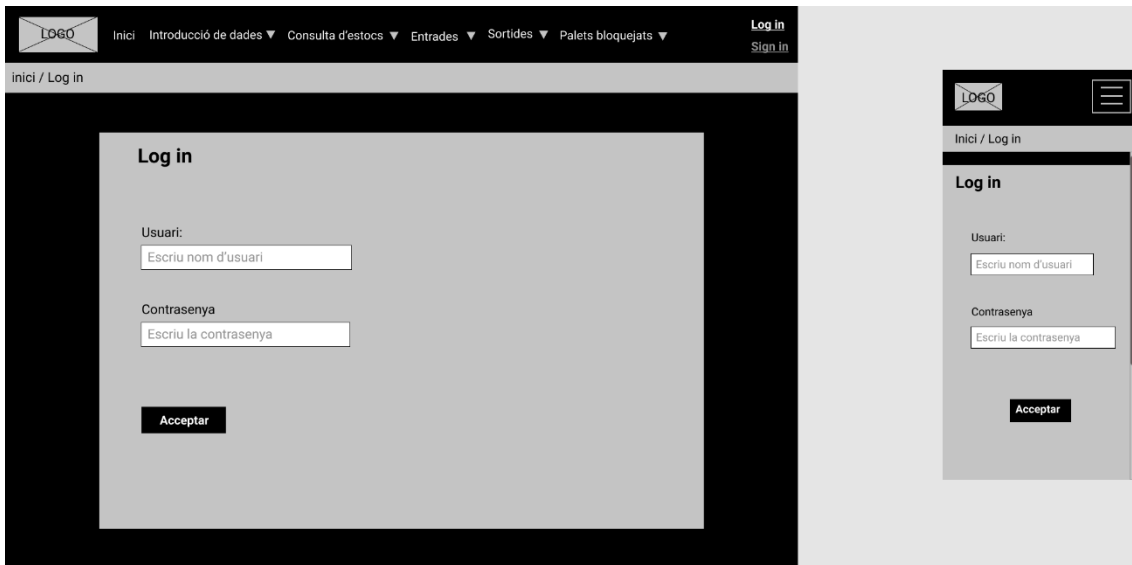


Figura 36: Wireframe LogIn

5. Usabilitat/UX

Tal i com s'ha detallat a la planificació del projecte, l'objectiu de l'aplicació és proporcionar una eina útil, fàcil d'utilitzar i intuïtiva per poder treballar d'una forma ràpida i efectiva. L'aplicació ha de poder ser utilitzada per diferents tipus d'usuaris, ja sigui per edats o per rols a l'empresa, així que és important que no tingui una gran corba d'aprenentatge i que detalli en cada cas avisos sobre errors o confirmacions de processos per facilitar la comprensió del procés que s'està duent a terme.

Per mantenir una bona experiència d'usuari a nivell de UX s'intentarà seguir els principis d'usabilitat detallats per **Jackob Nielsen**.

En primer lloc es tindrà en compte la visibilitat de l'estat del sistema, per tan, l'usuari ha d'estar constantment informat dels processos que porta a terme durant l'ús de l'aplicació. És important que si, per exemple, l'usuari elimina una línia de producte a la planificació d'una sortida, el sistema li mostri un petit missatge d'informació indicant que la línia de producte s'ha eliminat satisfactòriament. En el cas dels errors, també és important indicar a l'usuari quan hi ha algun error en algun procés. Aquests avisos i informacions permetran que l'usuari no es pugui sentir desorientat o confús sobre els processos realitzats durant l'ús de l'aplicació.

En segon lloc, hi ha la relació entre el sistema i el món real, que significa que és important que l'aplicació sigui intuïtiva per l'usuari, que sigui fàcil d'entendre. En aquest cas, per exemple, s'utilitzarà una icona on hi apareix una X o la icona d'una paperera on si la cliques procedeix a iniciar el procés per eliminar la línia de producte agregada a la planificació de sortides. Habitualment és una icona que els usuaris relacionen amb l'acció d'eliminar.

El mateix exemple es podria utilitzar per referir-se al tercer principi d'usabilitat, que és el de control i llibertat de l'usuari, on per exemple, pren importància la possibilitat de l'usuari de canviar d'opinió o esmenar algun error que hagi pogut tenir, en aquest cas agregar una línia de producte que no volia.

En quart lloc, hi ha la consistència i els estàndards, que fan referència a utilitzar elements visuals que siguin familiars per l'usuari i no el portin a confusions. Per exemple, a l'aplicació es mantindrà una estructura de les pàgines homogènia i semblant per a totes les pàgines per no confondre l'usuari.

En cinquè lloc, trobem el principi de prevenció d'errors, pel qual evitarem que l'usuari pugui cometre errors intentant prevenir-los abans. Per exemple, a l'aplicació, quan es planifiquen les sortides, el formulari automàticament calcula els palets que queden d'una referència concreta i

no deixa que l'usuari pugui seleccionar una quantitat més gran de la que hi ha al magatzem en aquell moment.

En sisè lloc, hi ha el principi de reconèixer abans de recordar, pel qual es facilita l'experiència de l'usuari perquè li dona la possibilitat de no haver de recordar algunes dades memoritzant-les. En el cas de l'aplicació, es dona la facilitat a l'usuari d'utilitzar la descripció de les referències a més del número, ja que és més fàcil de seleccionar en un desplegable una referència per la seva descripció que no pas pel número de referència.

El setè principi és el de flexibilitat i eficiència d'ús, que en aquest cas busca donar facilitat d'ús a persones menys experimentades i més eficiència a les que poden ser-ho més. En el cas de l'aplicació, es dona la possibilitat de buscar les dades dels estocs i els moviments de diferents formes i amb diferents filtres segons la conveniència de cada usuari i la informació que tingui cadascun d'ells. Es pot buscar informació per referència, client, SSCC, número d'albarà, etc.

En vuitè lloc es valora el principi de disseny estètic i minimalista, ja he comentat anteriorment que és un dels grans objectius de l'aplicació, que el disseny sigui senzill i concís però útil, sense necessitat de components o dissenys innecessaris o que puguin despistar a l'usuari.

En novè lloc, hi ha el principi d'ajudar als usuaris a reconèixer, diagnosticar i corregir els errors. En aquest cas es destaca la importància que els errors que puguin aparèixer a l'aplicació siguin clars i entenedors per l'usuari perquè pugui entendre quin problema hi ha i de quina manera el pot arreglar. En aquest cas a l'aplicació els errors estaran detallats de forma clara i entenedora.

Finalment, en desè lloc, hi ha l'ajuda i documentació, en aquest cas es destaca la importància que l'usuari sàpiga el que ha de fer i com ha de procedir en cada cas. L'aplicació en tot moment dona informació detallada a l'usuari dels passos a seguir en cada cas i, per exemple, als camps dels formularis es detalla amb placeholders la informació que hi ha d'haver a cada camp.

6. Tests

Pel que fa als tests i proves realitzades per posar a prova el treball respecte a la seva funcionalitat, rendiment i utilitat, he realitzat una sèrie de proves seguint diferents procediments utilitzant una eina facilitada per Angular per testejar la part del FrontEnd.

En aquest cas, he realitzat tota mena de testos unitaris als diferents components i serveis que componen l'aplicació i que estan desenvolupats a Angular. Per crear aquests testos unitaris he fet ús de les eines que facilita Angular per realitzar aquesta tasca, que és el framework *Jasmine* i *Karma test runner*.

Amb Jasmine, es creen diferents tests unitaris, ja que té varies funcionalitats per crear diferents tipus de tests. En canvi, Karma és una executador de tasques pels tests i permet automatitzar algunes tasques.

Tal i com he explicat, he creat diferents testos unitaris per provar petites unitats de codi per separat per corroborar-ne el bon funcionament, això fa que millori el disseny del codi, perquè es detecten petits errors de disseny a l'hora de crear les funcions per exemple. També permet la refactorització, ja que amb els tests unitaris ens assegurem que el codi funciona correctament i és més senzill afegir-hi altres funcionalitats sense preocupar-se d'afegir errors al codi. Finalment, l'aplicació d'aquests testos ens facilita el fet d'afegir noves funcionalitats sense trencar altres parts del codi.

Mitjançant aquestes eines he pogut millorar el codi en molts aspectes i redefinir algunes de les funcions o metodologies emprades per crear el codi, millorant-ne així el seu contingut i funcionalitat.

6.1 Resultats tests amb Karma (*ng test*):

Karma v 6.3.4 - connected; test: complete;

DEBUG

Chrome 96.0.4664.110 (Windows 10) is idle

3.7.1

Options



167 specs, 0 failures, randomized with seed 71477

finished in 0.942s

```
WebSocketService
  • should be created

ConsultaBloquejatsComponent
  • should create

BloquejarPaletsComponent
  • Ha de retornar el formulari com a vàlid
  • Ha de retornar el formulari com a no vàlid
  • should create

ClientsRegistratsComponent
  • Ha de retornar el formulari com a no vàlid
  • Ha de retornar el formulari com a vàlid
  • should create

BloquejatsService
  • should return expected bloquejats (called once)
  • should return expected bloquejats (called multiple times)
  • should call getBloquejat and return the appropriate Bloquejat
  • should be OK returning no bloquejats
  • should turn 404 into a user-friendly error
  • should call consultaBloquejatsEdit and the API should return the bloquejats
  • should call consultaBloquejats and the API should return the bloquejats from the client
  • should be created
  • should call deleteBloquejat and return the bloquejat that was deleted from the API
  • should call getBloquejats and return the array of bloquejats

ConsultaScccProducteComponent
  • Ha de retornar el formulari com a vàlid
  • should create
  • Ha de retornar el formulari com a no vàlid

SignInComponent
  • Ha de retornar el formulari com a no vàlid
  • Ha de retornar el formulari com a vàlid
  • should create

UsersComponent
  • should create
  • Ha de retornar el formulari com a no vàlid
  • Ha de retornar el formulari com a vàlid

LoginService
  • should call getUsers and return the array of users
  • should be created

AuthGuard
  • should be created

RegistreProductesComponent
  • Ha de retornar el formulari com a no vàlid
  • should create
  • Ha de retornar el formulari com a vàlid

ConsultaSortidesComponent
  • Ha de retornar el formulari com a no vàlid
  • Ha de retornar el formulari com a vàlid
  • should create

LoginComponent
  • Emetre bSubmitted false a l'inici
  • Ha de retornar el formulari com a vàlid
  • Mètode checkLogin()
  • should create
  • Ha de mostrar la icona de l'ull com a tancat
  • Ha de retornar el formulari com a no vàlid
  • Ha de mostrar la icona de l'ull com a obert
```

Figura 37: Resultat testing unitari amb Karma (part 1)

ProductesService

- should turn 404 into a user-friendly error
- should return expected Productes (called multiple times)
- should call getProductes and return the array of Productes
- should call getProducte and return the appropriate Producte
- should call updateProducte and return the updated Producte from the API
- should call deleteProducte and return the Producte that was deleted from the API
- should call getClientProductes and return the getClientProductes that was deleted from the API
- should be OK returning no Productes
- should return expected Productes (called once)
- should be created

PlanificacionsService

- should call consultaPlanificacions and the API should return the consultaPlanificacions
- should call deleteLinePlanificacions and the API should return the DeleteLinePlanificacions
- should call getPlanificacions and return the appropriate Planificacions
- should call deletePlanificacions and return the Planificacions that was deleted from the API
- should call deleteEntirePlanificacions and the API should return the deleteEntirePlanificacions that was added
- should be created
- should return expected Planificacions (called once)
- should return expected Planificacions (called multiple times)
- should call getPlanificacions and return the array of Planificacions
- should be OK returning no Planificacions
- should call comptador and the API should return the comptador
- should turn 404 into a user-friendly error
- should call addPlanificacions and the API should return the Planificacions that was added

SortidesComponent

- Ha de retornar el formulari de sortida com a vàlid
- Ha de retornar el formulari inicial com a vàlid
- Ha de retornar el formulari inicial com a no vàlid
- should create
- Ha de retornar el formulari de sortida com a no vàlid

LocationsRegistradesComponent

- Ha de retornar el formulari com a no vàlid
- Ha de retornar el formulari com a vàlid
- should create

EstocClientComponent

- Ha de retornar el formulari com a vàlid
- should create
- Ha de retornar el formulari com a no vàlid

ProductesRegistratsComponent

- Ha de retornar el formulari com a vàlid
- should create
- Ha de retornar el formulari com a no vàlid

PaletsService

- should call consultaEntrada and the API should return the consultaEntrada
- should call consultaSortida and the API should return the consultaSortida
- should call contador and the API should return the contador
- should call estocClient and the API should return the estocClient
- should call sortidesPal and the API should return the sortidesPal
- should call consultaSSCC and the API should return the consultaSSCC
- should call palResta and the API should return the palResta
- should call consultaSsccProduct and the API should return the consultaSsccProduct
- should call estocAlbara and the API should return the estocAlbara
- should turn 404 into a user-friendly error
- should call estocLot and the API should return the estocLot
- should call getPalets and return the array of palets
- should be OK returning no Palets
- should call getPalet and return the appropriate Palet
- should be created
- should call entradesPal and the API should return the entradesPal
- should return expected palets (called multiple times)

Figura 38: Resultat testing unitari amb Karma (part 2)

```

ConsultaPlanificatsComponent
  • should create

UserService
  • should return expected Users (called once)
  • should call deleteUser and return the User that was deleted from the API
  • should return expected Users (called multiple times)
  • should call updateUser and return the updated User from the API
  • should turn 404 into a user-friendly error
  • should call getUser and return the appropriate User
  • should call getUsers and return the array of Users
  • should be OK returning no Users
  • should be created

IntroduccioPaletsComponent
  • Ha de retornar el formulari com a vàlid
  • should create
  • Ha de retornar el formulari com a no vàlid

ConsultaEntradesComponent
  • Ha de retornar el formulari com a vàlid
  • Ha de retornar el formulari com a no vàlid
  • should create

HomeComponent
  • should create

EstocProducteComponent
  • Ha de retornar el formulari com a vàlid
  • Ha de retornar el formulari com a no vàlid
  • should create

PlanificacioComponent
  • Ha de retornar el formulari inicial com a vàlid
  • Ha de retornar el formulari inicial com a no vàlid
  • should create
  • Ha de retornar el formulari de planificació com a no vàlid
  • Ha de retornar el formulari de planificació com a vàlid

IntroduccioPalets2Component
  • should create
  • Ha de retornar el formulari com a vàlid
  • Ha de retornar el formulari com a no vàlid

LocationsService
  • should return expected Locations (called once)
  • should turn 404 into a user-friendly error
  • should be OK returning no Locations
  • should be created
  • should call getLocations and return the array of locations
  • should return expected Locations (called multiple times)
  • should call updateLocation and return the updated Location from the API
  • should call getLocation and return the appropriate Location
  • should call deleteLocation and return the Location that was deleted from the API

EstocLotComponent
  • Ha de retornar el formulari com a vàlid
  • should create
  • Ha de retornar el formulari com a no vàlid

HeaderComponent
  • should create

AppComponent
  • should have as title 'sga-app'
  • should create the app

ClientsService
  • should call updateClient and return the updated client from the API
  • should be created
  • should call deleteClient and return the client that was deleted from the API
  • should call getClients and return the array of clients
  • should call getClient and return the appropriate Client
  • should return expected Clients (called multiple times)
  • should turn 404 into a user-friendly error
  • should be OK returning no Clients
  • should return expected Clients (called once)

RegistreLocationsComponent
  • Ha de retornar el formulari com a vàlid
  • Ha de retornar el formulari com a no vàlid
  • should create

ConsultaSsccComponent
  • Ha de retornar el formulari com a no vàlid
  • should create
  • Ha de retornar el formulari com a vàlid

EstocUbicacioComponent
  • Ha de retornar el formulari com a vàlid
  • should create
  • Ha de retornar el formulari com a no vàlid

RegistreClientsComponent
  • Ha de retornar el formulari com a vàlid
  • Ha de retornar el formulari com a no vàlid
  • should create

EstocAlbaraComponent
  • Ha de retornar el formulari com a vàlid
  • should create
  • Ha de retornar el formulari com a no vàlid

```

Figura 39: Resultat testing unitari amb Karma (part 3)

A part de l'execució de tests amb Karma, també he realitzat el test de cobertura de l'aplicació. Aquest test permet saber fins a quin punt els testos realitzats cobreixen les diferents parts de l'aplicació, i et permet saber si els testos creats cobreixen de forma suficient les diferents parts del codi de l'aplicació. En aquest cas, gran part del codi no està testejada perquè he utilitzat Redux per gestionar tota la part del comportament i la gestió de dades de l'aplicació. La utilització de Redux ja és prou consistent per no haver-hi la necessitat imperativa de crear tants testos, ja que se suposa que si funciona, el codi és prou correcte, ja que el que fa és gestionar l'estat de l'aplicació i la seva funció és precisament la d'evitar errors i facilitar la gestió dels diferents estats de l'aplicació. És per això que considero que el resultat d'aplicar el test de cobertura és suficientment bo, tenint en compte que gran part del codi no el testejo perquè pertany a la funcionalitat de Redux. Els test de cobertura els he realitzat amb l'eina que ofereix Angular anomenada *code coverage*.

6.2 Resultats test de cobertura:

```
TOTAL: 167 SUCCESS
===== Coverage summary =====
Statements : 54.39% ( 940/1728 )
Branches   : 15.38% ( 66/429 )
Functions  : 38.99% ( 232/595 )
Lines      : 51.08% ( 803/1572 )
=====
```

Figura 40: Resultat cobertura testing amb *code coverage*

Tal i com he comentat anteriorment, amb una cobertura de més del 50% de testing en línies de codi i declaracions em considero bastant satisfeta tenint en compte la utilització de Redux i la falta d'experiència en el testing.

Pel que fa al testing de la part del backend (Laravel), no l'he portat a terme perquè durant el màster no n'hem fet, i tot i haver-me documentat pel meu compte amb la documentació oficial i altres fonts d'informació, no he tingut temps necessari d'assolir els coneixements per poder fer un testing correcte del backend. El timing per realitzar el TFM no m'ha permès poder dedicar més temps a aquest aspecte.

7. Requisits i instruccions d'instal·lació i ús

Des del primer moment he descrit aquesta aplicació com a un projecte destinat a les empreses per facilitar les tasques diàries derivades del funcionament d'un magatzem. L'objectiu principal de l'aplicació és facilitar el dia a dia en un entorn de treball, així que l'objectiu principal a part de la funcionalitat de l'aplicació, és també que sigui fàcil d'utilitzar i instal·lar.

Per instal·lar i utilitzar aquesta aplicació no fan falta uns requisits tècnics molt complicats. Hi ha dues formes d'implantar l'aplicació, i serà decisió de l'usuari o empresa triar quina li és més adequada.

7.1 Funcionament en xarxa local

D'una banda hi ha la possibilitat d'instal·lar l'aplicació en local, és a dir, s'instal·la l'aplicació en un servidor local i s'hi pot accedir des de qualsevol punt sempre i quan hi hagi a l'abast la ip local on hi ha instal·lada l'aplicació.

En aquest cas, els clients no podrien connectar-se a l'aplicació, ja que no tindrien a l'abast la connexió local. Tot i així, hi ha empreses a les quals els pot interessar aquesta opció, ja que permetria abaratir costos evitant la contractació de serveis addicionals per tenir hostings en línia. Pel que fa a la instal·lació, s'hauria d'instal·lar un servidor local com Xampp i disposar d'un dispositiu, que podria ser un ordinador, que fes de servidor i seria on s'instal·laria el servidor local físicament i es connectaria a la ip local perquè qualsevol dispositiu connectat a aquella xarxa d'internet pogués connectar-se a l'aplicació.

7.2 Funcionament en xarxa externa

D'altra banda, hi ha la opció d'instal·lar l'aplicació en un servidor web extern. Aquesta opció és bastant més recomanable i té més avantatges, ja que permet utilitzar totes les funcionalitats de l'aplicació i no fa necessari que es destini un dispositiu a fer de servidor físic de les dades de l'aplicació. A més, permet que qualsevol persona pugui connectar-s'hi des de qualsevol punt si disposa d'un dispositiu amb connexió a internet.

Em centraré més específicament a descriure la instal·lació i ús de l'aplicació en aquesta segona opció, ja que crec que és la més adequada per al projecte.

Personalment, per desenvolupar aquest projecte pel TFM he utilitzat recursos gratuïts per publicar l'aplicació a la xarxa. Per a un client real seria més apropiat disposar d'un hosting de pagament per poder gaudir de més avantatges, seguretat i emmagatzematge.

Tal i com he explicat anteriorment, he creat la part del FrontEnd creant un projecte d'Angular, que he exportat a Git i l'he pujat a la xarxa utilitzant Netlify.

Pel que fa al BackEnd, per a la API de gestió de la base de dades, he creat un projecte amb Laravel, l'he exportat a Git i l'he penjat a Heroku. He fet el mateix amb els Websockets, però enlloc de crear un projecte amb Laravel, l'he creat amb Node.

Per a connectar la part del FrontEnd amb el BackEnd, només he hagut de redireccionar les decretes amb les direccions als projectes penjats a la xarxa externa, enlloc de les rutes locals.

Per a la base de dades he exportat les dades utilitzant ClearDB MySQL de Heroku i he instal·lat MySQLWorkbench per gestionar-la.

Per crear les diferents dades fictícies de la base de dades, m'he ajudat dels Model Factories de Laravel, que permeten crear registres de prova.

Els requisits per instal·lar i utilitzar l'aplicació són mínims, tal i com ja he explicat anteriorment. No es necessita cap software ni hardware específic més enllà de connexió a internet i un navegador, ja que permet el seu funcionament en dispositius diferents i és compatible amb qualsevol sistema operatiu. És indiferent que es visualitzi en un smartphone, tablet, pc o terminal, o que sigui per a sistemes operatius tipus Android o iOS. S'adapta a qualsevol dispositiu sense la necessitat de tenir diferents versions.

Com a usuari, per a utilitzar l'aplicació només s'ha d'accedir a l'enllaç de Netlify on hi ha allotjada l'aplicació: <https://sga-frontend.netlify.app> .

Una vegada oberta l'aplicació mitjançant l'enllaç en un navegador web, hi ha la possibilitat d'utilitzar directament l'aplicació des del navegador, o bé es pot descarregar l'aplicació a l'escriptori del dispositiu, ja que l'aplicació és una PWA (Progressive Web App), i es pot crear un accés directe a l'aplicació per on accedir-hi com a una aplicació d'escriptori.

Per aconseguir col·locar la icona de l'aplicació a l'escriptori només s'ha de crear un accés directe des de la barra d'eines del navegador i seleccionar la opció d'obrir l'aplicació com a finestra.

Per tal d'utilitzar l'aplicació no és necessari tenir una formació o coneixements específics, ja que el funcionament és molt intuïtiu i pràctic, així que no fa falta tenir cap formació més enllà de conèixer les peculiaritats del funcionament del magatzem. Fàcilment es pot elaborar un petit informe o protocol, especificant els diversos passos a l'hora de realitzar l'entrada i sortida de

producte o el registre de dades, perquè els treballadors puguin tenir una guia a seguir en els primers contactes amb l'aplicació.

Hi ha diferents usuaris registrats actualment per entrar a l'aplicació:

- Usuari Administrador:

Usuari: tgarciafo@uoc.edu

Clau: 12345678

- Usuari Client:

Usuari: simba@uoc.edu

Clau: 12345678

- Usuari Magatzem:

Usuari: aida@uoc.edu

Clau: 12345678

Per poder entrar un palet a l'aplicació deixo diferents mostres d'etiquetes a la carpeta del projecte per poder fer proves. Si no es disposa d'un lector de codis de barres, es pot entrar directament cada codi numèric que apareix a sota de la imatge del codi de barres sense introduir els parèntesis entre números. Cada sèrie de número de codi de barres va a un camp diferent de la introducció de palets una vegada hem entrat el número d'entrada i la ubicació, per això n'hi ha dos.

Una vegada introduït el codi de barres, automàticament el sistema el reconeix, l'entra al formulari i augmenta el número de palets llegits pel número de sortida. El mateix passa amb la sortida de palets, però en aquest cas, per a la sortida només haurem de llegir el codi inferior, que és el que correspon al codi SSCC del palet.

Abans de realitzar la sortida de palets l'hem de planificar des de l'apartat de planificació, que és on seleccionarem el número de sortida, el client i finalment afegirem les diferents línies de producte que volem llegir per número de sortida.

La resta de funcionament de l'aplicació es basa en l'entrada de registres a través de formularis (usuaris, clients, referències i ubicacions), en el bloqueig i desbloqueig de palets a través dels codis SSCC com en el cas de la sortida de palets i en la consulta de dades dels estocs segons diferents criteris.

8. Projecció de futur

Pel que fa a possibles millores per a l'aplicació, una bona opció seria implementar el servei de picking. En el sector, el servei de picking es refereix al fet de crear nous palets a partir d'unitats de diferents palets sencers.

En aquest cas es dividirien els palets sencers en unitats independents i es crearien nous palets a partir d'aquestes unitats, quedant registrades al sistema les diferents dades referents a les unitats de cada palet sencer que compon el nou palet, al que se li atorgaria un nou número SSCC creat pel programa tenint en compte el sistema GS1-128 i els seus protocols per poder seguir la normativa vigent. Amb aquest servei també s'haurien de poder crear codis nous i imprimir les etiquetes amb les noves dades del palet nou creat seguint les normatives oficials.

Una altra millora podria ser afegir un servei de comptabilitat pel negoci tenint en compte els moviments de palets registrats i el seu estocatge. A partir dels moviments i les mitjanes mensuals d'estocatge es pot crear un servei de comptabilitat fent els càlculs automàticament amb les diferents variables com preu/palet carregat/descarregat, preu/mitjana mensual, etc.

També crec que hi ha un altre punt a millorar pel que fa a la gestió de nous usuaris. Per a la realització del treball he preferit seguir un mètode entenedor, clar i senzill, però si tingués més temps intentaria crear algun tipus de gestió utilitzant la mateixa aplicació per enviar els correus electrònics als usuaris nous i poder tenir un sistema de record de la contrasenya i restabliment.

En general, aquests serien els aspectes més importants que m'agradaria millorar si tingués més temps per millorar l'aplicació.

9. Anàlisi de mercat

Aquest projecte té com a objectiu crear una aplicació de gestió de magatzem, el que es coneix com a software SGA en el sector. Tal i com he esmentat anteriorment, actualment hi ha molta demanda de software d'aquest tipus, ja que cada vegada és més important i indispensable tenir una bona traçabilitat de les mercaderies. Per a un magatzem és un requisit pràcticament obligatori tenir un software de gestió d'estocs, ja que la majoria de clients ho requeriran, sobretot en el món de la logística de l'alimentació.

Aquests tipus de software són molt útils per poder localitzar i bloquejar, si és necessari, palets concrets, per això els departament de qualitat el posen com a requisit obligatori, ja que en moltes auditories els demanen aquest tipus d'informació per seguretat.

Actualment hi ha diferents tipus d'empreses que es dediquen a crear software dedicat a aquest objectiu, i la majoria tenen unes funcionalitats bastant semblants. Algunes estan més orientades a la creació de Picking manual, com és l'exemple de crear palets a partir de diferents palets de diferents referències, això fa que aquella etiqueta tingui informació dels palets originals dels quals es compon cada element del palet nou resultant. Aquesta possibilitat brinda a l'usuari la tranquil·litat de poder rastrejar cada peça del palet per separat. D'altres tenen sistemes de gestió de sortides integrant també el seguiment del transport fins al destí.

Però aquesta aplicació específicament és per a mercaderies de palet sencer, amb reconeixement de lot i caducitat i cobrint la mercaderia des del moment que entra al magatzem fins que en surt. Tal i com he descrit, hi ha diferents tipus d'aplicacions de gestió d'estocs depenent del tipus de magatzem, i en general, les empreses donen diferents opcions de software depenent de les necessitats del client.

Els preus d'aquestes aplicacions poden ser molt variats segons les prestacions que tinguin, però en general van dels 10000€ als 30000€.

Actualment, també hi ha empreses que ofereixen la possibilitat de llogar aquest servei, i es fa pagar una quota al client mensualment.

9.1 Viabilitat

9.1.1 Viabilitat tècnica

Pel que fa a la viabilitat tècnica, tenint en compte els elements tècnics que he esmentat a l'apartat de "Arquitectura de l'aplicació i plataforma de desenvolupament", no considero que sigui una aplicació amb una part tècnica molt complicada o difícil d'instal·lar, ja que no requereix grans estructures per posar-se en funcionament.

A l'hora de posar en funcionament l'aplicació, l'empresa té dues possibilitats:

Una opció seria instal·lar-la en un servidor local, destinant un ordinador com a servidor i instal·lar l'aplicació mitjançant Xampp. En aquest cas, no hi hauria cap mena de cost però l'aplicació només es podria utilitzar in-situ, ja que no estaria penjada a la xarxa.

L'altra opció seria penjar-la en un servidor web on poder allotjar l'aplicació i la base de dades, però en aquest cas hi hauria el cost de pagar una quantitat per poder allotjar-la. Aquesta opció permetria que qualsevol usuari des de qualsevol lloc pogués connectar-se a l'aplicació i utilitzar-la.

Cal dir, que si l'empresa que l'utilitza té per objectiu que els clients puguin accedir a l'aplicació per poder consultar personalment les dades dels seus productes al magatzem, la opció de l'allotjament local no seria una bona opció. En cas contrari, seria una bona opció per no tenir costos addicionals.

A l'hora d'utilitzar l'aplicació cal destacar que es pot fer servir tan a ordinadors, com en tauletes, smartphones o qualsevol terminal android, és a dir, a qualsevol dispositiu que tingui accés a internet. Cal destacar que és important que l'empresa disposi d'un terminal de lectura de codis de barres amb navegador. Un exemple de terminal per poder utilitzar l'aplicació seria un terminal de l'estil del Honeywell EDA61K. Aquest terminal és capaç de llegir codis de barres del tipus SG1 i a més pot accedir a l'aplicació.



Figura 41: Antena WIFI

El que també és important és destacar que el magatzem necessita disposar de xarxa wifi que arribi a totes les instal·lacions. Si és necessari es pot disposar d'antenes per poder estendre la xarxa wifi a tota la superfície del magatzem on ha d'actuar el terminal per poder llegir els palets.

Un exemple d'antena wifi seria un model com les antenes Ubiquiti UniFi AC Mesh Punt d'accés WIFI com a punt d'accés de tipus Mesh, que fa que ens puguem connectar de forma que sempre tinguem la millor senyal possible, selecciona quin és el dispositiu que et pot donar millor qualitat de connexió.

L'antena seria la següent:



Figura 42: Antena WIFI

9.1.2 Viabilitat econòmica

Pel que fa a la viabilitat econòmica, s'ha de tenir en compte les diferents opcions de desenvolupament tècnic i el cost que té el plantejament i el desenvolupament del projecte.

En aquest cas, ja que l'aplicació està dissenyada perquè els clients hi puguin accedir, tindrà en compte el cost dels components necessaris per allotjar l'aplicació a un servidor web.

Els costos a tenir en compte serien els següents (mostra d'exemple):

- Terminal de lectura de codis de barres: 1057.69€
- Hosting amb base de dades MySQL, espai web per 10Gb i tràfic il·limitat Arsys: 6.90€/mes
- Domini .com Arsys: 10€/any
- Hores de treball programador, instal·lació i posada en marxa: 300h x 30€/h = 9000€

10. Conclusions

Valorant la planificació inicial del projecte i els objectius principals i secundaris, es pot concloure que en general s'han pogut complir les expectatives prefixades inicialment. S'ha pogut crear una aplicació totalment funcional, aconseguint poder treballar amb els codis oficials GS1-128 decodificant-los i gestionant-ne la informació. S'han pogut crear les diferents funcionalitats proposades, així com també crear diferents perfils d'usuaris i d'interacció amb l'aplicació. Les diferents funcionalitats proposades als objectius han pogut ser creades, així com també s'ha pogut aplicar el format Progressive Web App i s'ha aconseguit exportar les dades a Excel.

Cal remarcar que en els wireframes s'havia contemplat la possibilitat de col·locar breadcrumbs, però finalment s'ha desestimat perquè no s'ha cregut necessari tenint en compte l'estructura de l'aplicació.

Amb la realització del TFM s'han pogut posar en pràctica la major part de coneixements adquirits durant el màster, des del disseny d'interfícies, fins a la utilització de llenguatges com HTML, CSS, typescript, com la utilització de diferents frameworks com Angular, Laravel o la gestió de bases de dades. També s'han pogut posar en pràctica els coneixements de testing, que en aquest aspecte cal dir que els coneixements adquirits durant el màster eren mínims, així que s'ha hagut de buscar molta bibliografia i informació. Cal dir que el testing realitzat és bastant limitat, i en aquest aspecte es podria millorar molt. S'ha optat per fer només testing d'Angular, ja que durant el màster no s'ha fet cap mena de referència al testing amb Laravel, i per la falta de temps no s'ha pogut entrar en matèria.

Per realitzar el treball s'hi ha realitzat moltes hores de treball, fet que ha permès poder arribar a tots els objectius marcats des de bon principi. Tot i així, hi ha molts aspectes millorables tal i com s'ha descrit a la part de projecció de futur.

El procés de treball ha estat progressiu i constant durant tota l'assignatura, i la interacció amb el professor ha estat habitual per tal de mantenir un bon desenvolupament del projecte i una bona comunicació.

Personalment, es considera que el treball ha aportat molts aspectes positius a l'alumna, tan d'assoliment de coneixements adquirits durant el màster com de coneixements nous adquirits per la creació de funcionalitats com la creació d'alertes en temps real mitjançant els WebSockets, com la publicació de l'aplicació a la xarxa mitjançant Heroku i penjant la base de dades utilitzant MySQLWorkbench.

En conclusió, el projecte ha estat tot un repte per a l'alumna i en general la percepció és de satisfacció per l'assoliment dels objectius i dels coneixements adquirits.

Annex 1. Lliurables del projecte

- PAC_FINAL_mem_GarciaFont_Tania.pdf
- PAC_FINAL_prj_GarciaFont_Tania.zip (conté dues carpetes 1 local i 1 servidor extern):
 - backEndLaravel.zip (codis font API Laravel)
 - frontEnd.zip (codis font app Angular)
 - webSockets.zip (codis font servidor WebSockets)
 - codis_exemples.pdf (exemples codis de barres per fer proves)
 - DB_sga.sql (base de dades)
- PAC_FINAL_vid_GarciaFont_Tania.mp4 (presentació-defensa TFM)
- PAC_FINAL_prs_GarciaFont_Tania.mp4 (presentació públic general)

Annex 2. Llibreries/Codi extern utilitzat

Tal i com he comentat anteriorment, he utilitzat **GS1-barcode-parser** com a llibreria externa per poder descodificar els codis de barres en format GS1-128. He modificat part del codi per poder adaptar-lo a l'aplicació.

També he utilitzat una altra llibreria externa per imprimir dades des de l'aplicació utilitzant la llibreria **ngx-print**.

Finalment, també he utilitzat una llibreria externa per poder exportar algunes taules a Excel amb la llibreria **xlsx**.

Anteriorment ja he comentat que he instal·lat diferents tipus de packages per a diverses funcionalitats de l'aplicació (Bootstrap, FontAwesome, ng-bootstrap, ngx-socket-io i angular-input-focus).

Annex 3. Captures de pantalla

Aquest codi és l'adaptació al component de Laravel per poder utilitzar la llibreria externa GS1-barcodeParser per introduir les dades dels palets segons les necessitats de l'aplicació. Vaig adaptar el codi perquè es pogués utilitzar en Typescript a Angular:

```
interpreteBarcode(){
  "use strict";

  try{

    let answer= parseBarcode(this.codi());

    return answer.parsedCodeItems.forEach(this.basedades, this);

  } catch (e){
    console.log(e);
  }
}

interpreteBarcode2(){
  "use strict";

  try{

    let answer= parseBarcode(this.codi());

    answer.parsedCodeItems.forEach(this.basedades, this);

    return this.buildForm();

  } catch (e){
    console.log(e);
  }
}

basedades(element: any, index: any,array: any){
  const ai=element.ai;
  let data=element.data;

  if(ai=='10'){
    this.lot.setValue(data);
  } else if ((ai=='01') || (ai=='02')){
    this.ean.setValue(data);
    this.getProduct();
  } else if (ai=='02'){
    this.ean.setValue(data);
    this.getProduct();
  } else if (ai=='00'){
    this.sccc.setValue(data);
  } else if (ai=='15'){
    data=data.toLocaleDateString('es-ES');
    data=data.split("/").reverse().join("-");
    this.caducitat.setValue(data);
  } else if (ai == '37') {
    this.qty.setValue(data);
  }
}

codi(){
  if(this.barcode2.value != ''){
    return this.barcode2.value;
  } else if (this.barcode.value != ''){
    return this.barcode.value;
  }
}
```

Figura 43: Extracte codi font introduccio-palets2.component.ts

Annex 4. Guia d'usuari

Pel que fa a la instal·lació de l'aplicació en un hosting extern, ja ho he detallat específicament a l'apartat 7 de la memòria sobre Requisits i instruccions d'instal·lació i ús.

A continuació, detallo la instal·lació de l'aplicació en un entorn local:

A la carpeta del projecte hi ha tres carpetes, una pel FrontEnd i dues pel BackEnd (base de dades Laravel i l'API Socket.IO pels WebSockets).

És necessari tenir un servidor local instal·lat, com xampp, i a la carpeta xampp->htdocs hi afegim les tres carpetes que hi ha a dins les de frontend i backend. Iniciem Xampp i a phpmyAdmin hi importem la base de dades que adjunto a la carpeta del projecte.

El frontEnd és en Angular, per tan s'ha d'entrar a l'arrel del projecte des d'un terminal i escriure `npm install`, es crearà l'arxiu `node_modules` i després escriure al terminal `ng serve`. Ja tindrem el Frontend funcionant a `localhost:4200`.

Per instal·lar el backend de la base de dades, entrem al projecte Laravel, i al terminal hi anotem `php artisan serve` i s'inicia la connexió.

Per instal·lar l'API Socket.io, entrem al projecte sga-sockets de l'API, i al terminal hi anotem `npm install`. En aquest cas instal·larà el projecte, i després per a connectar el servidor només haurem d'anotar a la consola `npm run serve` i s'inicia la connexió.

A partir d'aquí ja es pot utilitzar l'aplicació accedint a la ruta: <http://localhost:4200>.

Annex 5. Bibliografia

Laravel. Documentació oficial de Laravel: <https://laravel.com/docs/8.x>

Angular. Documentació oficial de Angular: <https://angular.io/docs>

Socket.io. Documentació oficial de Socket.io: <https://socket.io/docs/v4/>

Bootstrap. Documentació oficial de Bootstrap: <https://getbootstrap.com/docs/5.1/getting-started/introduction/>

Heroku. Documentació oficial de Heroku: <https://devcenter.heroku.com/categories/reference>

Netlify. Documentació oficial de Netlify: <https://docs.netlify.com/>

ClearDB MySQL. Documentació oficial ClearDB MySQL: <https://devcenter.heroku.com/articles/cleardb>

Guia iniciació a la codificació GS1-128. Obtinguda a partir de: <https://www.gs1es.org/wp-content/uploads/2016/07/Inicio-a-la-codificacion-GS1-128.pdf>

Llibreria Externa GS1-barcode-parser. Obtinguda a partir de: <https://www.npmjs.com/package/gs1-barcode-parser>

GS1-barcode-parser. Documentació descodificador de codis de barres: <https://github.com/hprange/BarcodeParser#readme>

Faker Factory Laravel. Documentació obtinguda a partir de: https://github.com/fzaninotto/Faker#fakerprovideren_uscompany

Utilitzar WebSockets a Heroku amb Node.js. Documentació obtinguda a partir de: <https://devcenter.heroku.com/articles/node-websockets>

Ngx-print. Obtinguda a partir de: <https://www.npmjs.com/package/ngx-print>

Xlsx. Obtinguda a partir de: <https://www.npmjs.com/package/xlsx>

@ng-bootstrap/ng-bootstrap. Obtinguda a partir de: <https://www.npmjs.com/package/@ng-bootstrap/ng-bootstrap>

FontAwesome. Obtinguda a partir de: <https://fontawesome.com/v5.15/how-to-use/on-the-web/setup/using-package-managers>

Redux Angular. Documentació oficial NgRx: <https://ngrx.io/docs>

Angular-input-focus. Obtinguda a partir de: <https://www.npmjs.com/package/angular-input-focus>

Ngx-socket-io. Obtinguda a partir de: <https://www.npmjs.com/package/ngx-socket-io>

Node.js. Documentació oficial Node.js: <https://nodejs.org/es/>

Express. Documentació Express: <https://expressjs.com/es/api.html>

Unsplash. Imatges de llicència lliure: <https://unsplash.com/>

PWA. Documentació per a Progressive Web App: <https://web.dev/progressive-web-apps/>

WebSockets. Informació WebSockets a Javascript.info: <https://es.javascript.info/websocket>