UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

OPTIMITZACIÓ DEL DESENVOLUPAMENT D'APLICACIONS E-COMMERCE: CREACIÓ D'UN CORE PERSONALITZABLE

LAURA CHAVARRIA SOLE

Director/a
MATÍAS SEBASTIÁN MARTÍNEZ MARTÍNEZ (Departament d'Enginyeria de Serveis i Sistemes d'Informació)

Titulació
Grau en Enginyeria Informàtica (Enginyeria del Software)

Memòria del treball de fi de grau

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

27/06/2024

Resum

Aquest treball de final de grau es desenvolupa en la modalitat A. El projecte té com a objectiu principal la creació d'un *core* personalitzable per a aplicacions mòbils d'e-commerce, centrant-se en el sector alimentari. Mitjançant l'ús de tecnologies com Ionic i Spring Boot, es pretén proporcionar una solució eficient i escalable per a empreses amb diversos clients, optimitzant així el temps de desenvolupament.

Resumen

Este trabajo de fin de grado se desarrolla en la modalidad A. El proyecto tiene como objetivo principal la creación de un *core* personalizable para aplicaciones móviles de comercio electrónico, centrándose en el sector alimentario. Mediante el uso de tecnologías como Ionic y Spring Boot, se pretende proporcionar una solución eficiente y escalable para empresas con diversos clientes, optimizando así el tiempo de desarrollo.

Abstract

This Final Degree Project is developed in modality A. The main objective of the project is to create a customizable core for e-commerce mobile applications, focusing on the food sector. By using technologies such as Ionic and Spring Boot, the aim is to provide an efficient and scalable solution for companies with multiple clients, thus optimizing development time.

Índex

1 Introducció **11**

1.1 Context 11

1.2 Descripció del problema 12

1.3 Solucions existents 12

1.4 Solució proposta 14

1.5 Objectius 15

1.6 Stakeholders 15

1.7 Riscs 16

1.8 Metodologia i rigor 18

2 Planificació inicial **21**

2.1 Descripció de les tasques 23

2.2 Recursos 27

2.3 Gestió de riscs 28

3 Gestió econòmica **30**

3.1 Pressupost inicial 30

3.2 Control de gestió 33

4 Especificació de requisits **35**

4.1 Requisits funcionals 35
4.2 Requisits no funcionals 41
4.3 Model conceptual de dades 43

5 Arquitectura **47**

5.1 Disseny de l'arquitectura del sistema 47
5.2 Exemples de diagrames de seqüència 50
5.3 Patrons de disseny 52
5.4 Bases de dades 53
5.5 Disseny de la interfície 65

6 Implementació **79**

6.1 Tecnologies, llenguatges utilitzats i eines de desenvolupament 79
6.2 Anàlisi d'alternatives 82
6.3 Aspectes rellevants del codi d'implementació 84

7 Proves **88**

7.1 Tests unitaris *Backend* 88
7.2 Tests de disseny i usabilitat 88
7.3 Integració contínua amb Jenkins 89

8 Execució del projecte **90**

8.1 Canvis en la planificació inicial 90
8.2 Afectacions dels canvis de la planificació inicial al pressupost 95

9 Aspectes legals **97**

9.1 Lleis aplicables al projecte 97

9.2 Llicències 98

10 Informe de sostenibilitat **99**

10.1 Dimensió econòmica 99

10.2 Dimensió ambiental 101

10.3 Dimensió social 103

11 Treball futur **105**

11.1 Implementació 105

11.2 Canvi a un model de llenguatge natural local 107

12 Conclusions **109**

12.1 Competències tècniques treballades 109

12.2 Relació del TFG amb el grau i l'especialitat 110

12.3 Conclusions personals 112

A Anexos **119**

A.1 Hores treballades per rols (Gestió econòmica) 119

A.2 Endpoints 121

A.3 Schemas 128

Índex de figures

1.1 Les 10 activitats amb major percentatge de volum de negoci del comerç electrònic. (CNMC) 11

1.2 Metodologia *Scrum*. (Scrum.org) 18

1.3 Taulell de treball *Kanban*. (Asana) 19

2.1 Planificació temporal de les fases i etapes del projecte (Font pròpia) 21

2.2 Planificació detallada de les fases i etapes del projecte. (Font pròpia) . . . 22

2.3 Descripció de les tasques de l'etapa d'*Inception* separades amb èpiques. (Font pròpia) 24

2.4 Descripció de les tasques de l'*Sprint 1* separades amb èpiques. (Font pròpia) 24

2.5 Descripció de les tasques de l'*Sprint 2* separades amb èpiques. (Font pròpia) 24

2.6 Descripció de les tasques de l'*Sprint 3* separades amb èpiques. (Font pròpia) 25

2.7 Descripció de les tasques de l'*Sprint 4* separades amb èpiques. (Font pròpia) 25

2.8 Descripció de les tasques de l'*Sprint 5* separades amb èpiques. (Font pròpia) 25

2.9 Diagrama de Gantt. (Font pròpia) 26

3.1 Costos dels recursos humans. (Font pròpia) 30

3.2 Costos de software. (Font pròpia) 31

3.3 Costos de hardware. (Font pròpia) 31

3.4 Costos genèrics. (Font pròpia) 32

3.5 Costos d'imprevistos. (Font pròpia) 32

3.6 Costos totals. (Font pròpia) 33

4.1 Model conceptual de dades - Gestió de productes. (Font pròpia) 43

4.2 Model conceptual de dades - Gestió de la cistella de la compra. (Font pròpia) 44

4.3 Model conceptual de dades - Gestió de favorits. (Font pròpia) 45

4.4 Model conceptual de dades - Gestió d'usuaris. (Font pròpia) 46

5.1 Diagrama d'arquitectura física. (Font pròpia) 48

5.2 Diagrama d'arquitectura lògica. (Font pròpia) 49

5.3 Diagrama de seqüència - Crida /api/payment/confirm/id. (Font pròpia) . . 50

5.4 Diagrama de seqüència - Crida /api/ai/recipe. (Font pròpia) 51

5.5 Patró Model-View-ViewModel. (Wikipedia) 52

5.6 Patró Model-View-Controller. (LinkedIn [38]) 53

5.7 External Design de la gestió de productes. (Font pròpia) 66

5.8 Disseny de la grid de productes. (Font pròpia) 67

5.9 Disseny de la pantalla del detall del producte. (Font pròpia) 68

5.10 External Design de la gestió de receptes. (Font pròpia) 68

5.11 Disseny de la pantalla de receptes. (Font pròpia) 69

5.12 External Design de la gestió de la cistella de la compra. (Font pròpia) . . . 70

5.13 Disseny de la pantalla de la cistella de la compra. (Font pròpia) 71

5.14 Disseny de la pantalla de pagament. (Font pròpia) 71

5.15 External Design de la gestió de favorits. (Font pròpia) 72

5.16 Disseny de la pantalla de favorits. (Font pròpia) 73

5.17 Disseny de la pantalla del detall d'una recepta. (Font pròpia) 73

5.18 External Design de l'autentificació. (Font pròpia) 74

5.19 Disseny de la pantalla inicial de l'aplicació. (Font pròpia) 75

5.20 Disseny de la pantalla d'inici de sessió. (Font pròpia) 76

5.21 Disseny de les pantalles de registre. (Font pròpia) 76

5.22 Disseny de la pantalla d'introducció. (Font pròpia) 77

5.23 Disseny de la pantalla del perfil de l'usuari. (Font pròpia) 78

6.1 Tecnologies, llenguatges utilitzats i eines de desenvolupament. (Font pròpia) 79

6.2 Procés implementat per a la cerca d'ingredients i productes a partir del nom d'una recepta. (Font pròpia) 86

8.1 Logotip creat per a l'aplicació del nostre projecte. (Font pròpia) 90

8.2 Canvis aplicats a l'spint 2. (Font pròpia) 91

8.3 Canvis aplicats a la distribució d'hores de les tasques de l'sprint 2. (Font pròpia) 91

8.4 Canvis aplicats a l'spint 3. (Font pròpia) 92

8.5 Canvis aplicats a l'spint 4. (Font pròpia) 92

8.6 Canvis aplicats a l'spint 5. (Font pròpia) 92

8.7 Canvis aplicats a la distribució d'hores de les tasques de l'sprint 4 i 5. (Font pròpia) 93

8.8 Canvis dels costos dels recursos humans. (Font pròpia) 96

8.9 Canvis dels costos de software. (Font pròpia) 96

11.1 Pantalla de benvinguda de l'aplicació creada a través del core desenvolupat. (Font pròpia) 107

A.1 Hores treballades per rols. (Font pròpia) 119

A.2 Modificacions d'hores treballades per rols. (Font pròpia) 120

Índex de taules

1.1 Comparació de les diferents solucions existents. *(Font pròpia)* 14

1.2 Relació dels diferents riscos amb el seu impacte potencial. *(Font pròpia)* . 17

2.1 Relació dels diferents riscos amb el seu impacte potencial. *(Font pròpia)* . 29

6.1 Comparació de models de llenguatge natural: API vs Local. (Font pròpia) 83

6.2 Comparació d'Avantatges i Desavantatges entre OpenAI ChatGPT i Ollama Llama3. (Font pròpia) 84

6.3 Taula comparativa entre els sistemes de pagament Redsys i Stripe. (Font pròpia) 87

Introducció

1.1 Context

En el marc actual del comerç electrònic, el sector dels hipermercats, supermercats i botigues d'alimentació destaca com una de les deu activitats amb més rellevància en termes de volum de negoci tal com es mostra a la Figura 1.1.

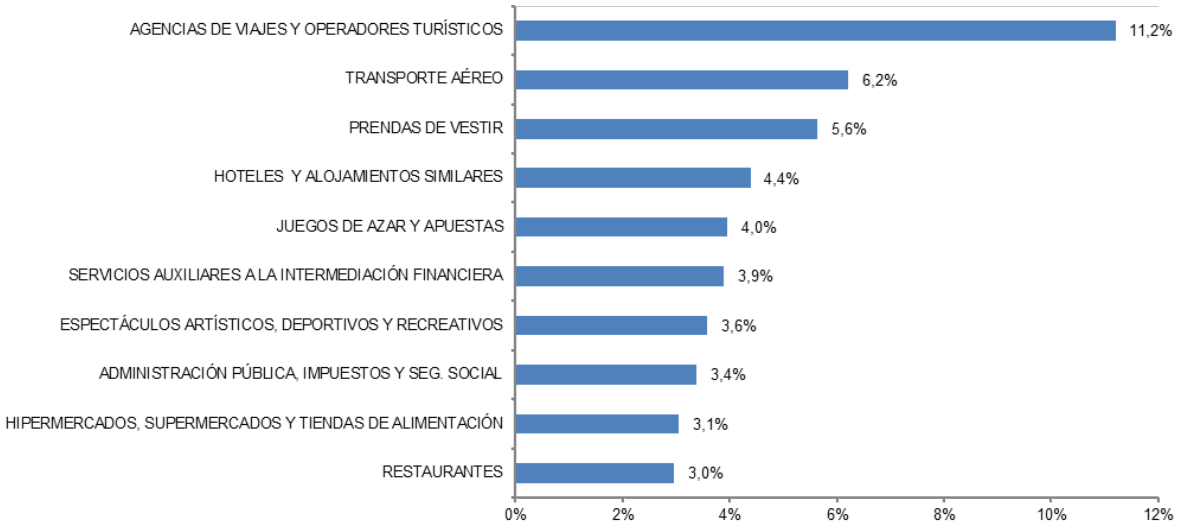

Figura 1.1: Les 10 activitats amb major percentatge de volum de negoci del comerç electrònic. (CNMC)

Segons les dades proporcionades per la Comissió Nacional dels Mercats i la Competència (CNMC) [1], en el segon trimestre del 2023, el volum d'aquest sector dins d'Espanya va assolir els 419.951.838,09 €, un 8.7% més que l'any anterior.

1.2 Descripció del problema

Segon Shakir C., autor de l'article '*The Impact of COVID-19 on mobile app usage and Development*' a la plataforma de *blogging* Medium [3], la pandèmia de la COVID-19 ha impactat significativament l'ús i el desenvolupament d'aplicacions mòbils, i les persones recorren cada vegada més a aplicacions mòbils per mantenir-se connectades, entretingudes i informades. La COVID ha accelerat l'adopció de tecnologies digitals i les aplicacions mòbils s'han convertit en una part integral de la nostra vida quotidiana.

Aquesta expansió del comerç electrònic, posa de manifest la importància de desenvolupar eines que permetin la creació àgil i eficient d'aplicacions. No obstant això, per a una empresa amb intenció d'augmentar significativament el volum de clients, el desenvolupament independent per a cada client d'una aplicació mòbil personalitzada des de zero li implica una duplicació d'esforços i recursos, augmentant així el cost econòmic i temporal.

Si ens basem en l'article *¿Cuánto tiempo se tarda en crear una aplicación?* [4], el temps necessari per desenvolupar una aplicació mòbil pot variar significativament segons diferents factors com la complexitat i funcionalitats requerides, els usuaris i dispositius objectius i altres factors, en aquest cas, com l'experiència dels desenvolupadors. Aquest procés de creació consta de 5 fases bàsiques: planificació, disseny, desenvolupament, proves i llançament. Si considerem que el projecte és de complexitat baixa, el temps de dedicació per finalitzar l'aplicació pot arribar a comportar unes 300-600 hores de treball (més de quatre mesos treballant a jornada completa), mentre que una aplicació mòbil de complexitat mitjana requereix entre unes 700 i 1200 hores. Finalment, un projecte d'alta complexitat podria suposar més de 1500 hores.

En resum, la creació d'una aplicació mòbil implica una inversió significativa de temps i recursos per a les empreses, és per això que cal optimitzar aquest procés per mantenir la competitivitat del mercat actual.

1.3 Solucions existents

En aquest apartat, es dues solucions Espanyoles existents a l'hora de desenvolupar aplicacions de forma simultania, ja siguin web o mòbil. És important destacar que aquestes empreses han estat seleccionades pel seu enfocament empresarial i la seva capacitat per oferir solucions personalitzades i escalables per a múltiples projectes simultàniament. Aquestes solucions no estan destinades a desenvolupadors individuals ni són de codi obert; més aviat, estan dissenyades per a empreses que necessiten desenvolupar i mantenir diversos projectes de manera eficient i efectiva. S'han seleccionat únicament aquestes empreses perquè no s'ha pogut obtenir gaire informació de la gestió interna d'altres empreses, es

pot saber que són consultores tecnològiques i que desenvolupen aplicacions per a diferents clients, però el que ens interessa és per al desenvolupament d'aquestes aplicacions utilitzin algun sistema que els permeti mantenir a tots els clients actualitzats sense un cost específic per a cada client. És a dir, han d'utilitzar un mateix projecte com a *core* per a diferents clients o alguna metodologia semblant.

1.3.1 Solucions Empresarials per al Desenvolupament d'Aplicacions

Les solucions empresarials seleccionades són aquelles que permeten a les empreses gestionar múltiples projectes de manera eficient, reutilitzant codi i estructures bàsiques per personalitzar aplicacions segons les necessitats de cada client.

Aktios

Aktios [6] és una empresa que ofereix un servei de desenvolupament de plataformes web E-commerce, entre altres serveis, utilitzant un *core* com a base de les aplicacions de tots els clients. D'aquesta manera, aquesta consultora només ha de modificar les configuracions específiques de cada client i els dissenys necessaris. Així, és capaç de mantenir simultàniament la gestió d'aplicacions web de molts clients com Consum, BM o Merza.

480 - Cuatroochenta

Cuatroochenta [7] és una altra consultora que entre els seus serveis ofereix el desenvolupament d'aplicacions mòbils. Aquesta empresa ha desenvolupat aplicacions com Mundo Consum, Hofmann o Aquarama. Tots aquests projectes parteixen d'unes configuracions bàsiques i una estructura de carpetes i fitxers d'un repositori de codi específic segons la tecnologia que s'empra. És a dir, aquesta empresa té per a cada tecnologia (Android, iOS, Flutter, ReactJS...) un projecte base per poder reaprofitar codi. Aquesta solució, a diferència de l'anterior, no està enfocada a un sector particular i tampoc té un desenvolupament extens de funcionalitats per reaprofitar, sinó que cada projecte és totalment diferent.

A continuació, per poder comparar d'una manera fàcil les diferents funcionalitats i les qualitats de cada solució juntament amb la solució proposta, s'ha creat la següent taula 1.1 que resumeix les principals característiques.

	Aktios	**Cuatroochenta**	**Solució**
Plataforma	Web	Mòbil	Mòbil (i adaptable a plataforma web)
Manteniment simple	Sí	No	Sí
Personalització	Sí	Sí	Sí
Eficiència	Sí	No	Sí
Funcionalitats amb IA	No	No	Sí
Codi obert	No	No	No

Taula 1.1: Comparació de les diferents solucions existents. *(Font pròpia)*

1.4 Solució proposta

El projecte "Optimització del desenvolupament d'aplicacions E-commerce: Creació d'un *core* personalitzable" emergeix com una solució estratègica per a empreses i consultores que han d'atendre múltiples clients amb necessitats similars en l'àmbit del comerç electrònic. La proposta se centra en la creació d'un *core* altament especialitzat i personalitzable per abordar la complexitat i les demandes creixents, en aquest cas, en centrarem especialment en el sector alimentar. A més a més, inclou de manera estratègica la integració d'intel·ligència artificial (IA) per oferir solucions avançades i adaptatives a les empreses en el sector del comerç electrònic.

Aquest *core* oferirà avantatges específics per optimitzar l'eficiència en el desenvolupament i garantir una gestió centralitzada de diferents clients a aquelles empreses que gestionen diverses iniciatives d'aplicacions de comerç electrònic simultàniament. Això implica una estructura modular que possibilita la configuració i personalització independent per a cada client, mantenint alhora una base comuna que redueix el temps de desenvolupament i les duplicacions d'esforços.

La solució proporcionarà eines de configuració centralitzades que permetin als gestors o administradors definir paràmetres globals que afectin a tots els clients. Al mateix temps, els desenvolupadors tindran la capacitat de personalitzar individualment cada instància de l'aplicació per satisfer les necessitats particulars de cada client. Això assegura una harmonia entre la coherència de la marca global i la flexibilitat individual.

1.5 Objectius

El projecte consta d'un objectiu clar que es divideix en subobjectius per tenir un major control de la solució a desenvolupar.

L'objectiu principal d'aquest projecte és crear un *core* que sigui personalitzable i configurable per poder optimitzar el temps de creació d'una aplicació mòbil *e-commerce*.

Per poder complir aquest objectiu, cal assolir uns subobjectius:

- Poder veure tots els productes.
- Poder fer una compra i fer-ne el seguiment.
- Poder obtenir un llistat de productes per ingredients a partir d'una recepta específica utilitzant IA.
- Poder afegir productes o receptes a preferits.

Per altra part, també cal considerar certs objectius tècnics:

- Millorar les tècniques de desenvolupament mòbil amb les tecnologies Ionic [8] i Spring Boot [10].
- Adquirir coneixements sobre la integració de la intel·ligència artificial.
- Adquirir coneixements sobre la integració de sistemes de pagament.

1.6 Stakeholders

Director del TFG: Encarregat de dirigir i supervisar a l'estudiant durant el procés de desenvolupament del TFG oferint assessorament tècnic i metodològic per poder aconseguir els objectius plantejats.

Autor del TFG: Principal interessat a desenvolupar un projecte òptim que compleixi amb tots els requisits especificats.

Dissenyador: Dissenyador de la interfície de l'aplicació mòbil i l'experiència de l'usuari amb l'objectiu d'assolir un disseny que sigui intuïtiu, fàcil d'utilitzar i el més accessible possible.

Desenvolupador: Persona encarregada d'implementar totes les funcionalitats requerides per al projecte i documentar el projecte.

Tester: Grup de persones encarregades de fer proves de l'aplicació i explicar la seva experiència i donar les seves opinions per tal que el desenvolupador tingui un *feedback* sobre l'experiència de l'usuari. A més del mateix autor del TFG, que és l'encarregat de comprovar el correcte funcionament de totes les funcionalitats requerides.

Clients: Aquelles persones interessades a fer servir l'aplicació final. Principalment, els clients de qualsevol dels supermercats que faci ús de la nostra solució com a mètode per gestionar les seves comandes.

Competidors: Plataformes o serveis que ofereixen solucions a aquest problema i, per tant, es veuran afectades pel nou competidor.

Hipermercats, supermercats i botigues d'alimentació: Principals clients interessats a beneficiar-se d'una aplicació mòbil que compleixi amb els requisits principals del projecte.

1.7 Riscs

A l'hora de planificar i desenvolupar un projecte de *software*, cal considerar diferents situacions que poden alterar la planificació inicial del projecte. Per evitar un gran impacte, hem de conèixer els possibles riscos existents juntament amb la seva probabilitat i impacte. A continuació, apareix el llistat dels riscos considerats juntament amb la Taula 1.2 que expressa la relació de probabilitat i impacte amb cada risc avaluat.

1.7.1 Desviacions de la planificació

Existeix la probabilitat de possibles desviacions en la planificació inicial del projecte causades per la inexperiència de l'autor en certes tecnologies o per qualsevol problema tècnic. Hi ha una probabilitat mitjana que això succeeixi, ja que la planificació, ja es farà considerant tots aquests factors i amb funcionalitats extres per no carregar extremadament la planificació sabent els possibles riscs que existeixen. Tot i això, si la falta de coneixements en algun aspecte es desborda, llavors sí que podria afectar i causar un impacte alt.

1.7.2 Bugs

Els errors de codi poden sorgir durant el desenvolupament o després de la implementació, cosa que podria afectar la funcionalitat, l'estabilitat i la seguretat del sistema. La probabilitat d'ocurrència de *bugs* és alta, ja que cal considerar la naturalesa del desenvolupament de programari i la varietat de plataformes i dispositius. L'impacte dels *bugs* pot ser significatiu, causant interrupcions al servei o pèrdua de dades. De totes maneres, considerarem que els possibles *bugs* que apareixeran al nostre projecte seran *bugs* de menor impacte com ara errors de disseny (colors, alineació d'elements o altres aspectes estètics).

1.7.3 Estimació incorrecta del temps de les tasques

Un dels problemes que pot impactar greument és l'estimació incorrecta del temps estimat de desenvolupament de les tasques. Aquest projecte té una data d'entrega tancada i, per tant, encara que el desenvolupament s'expandís, no es podria retardar la presentació del treball. Això significa que algunes de les funcionalitats requerides inicialment no es podrien portar a terme i no quedaria finalitzat amb èxit el treball.

Tot i això, hi ha una probabilitat molt baixa que succeeixi aquest risc, ja que durant l'assignatura de GEP es fa especial èmfasi en la planificació del projecte.

1.7.4 Complexitat tècnica

El desenvolupament d'una aplicació mòbil e-commerce pot implicar l'ús de diferents tecnologies complexes que ens puguin dificultar el desenvolupament i augmentar el risc d'errors. Un exemple seria la integració de passarel·les de pagament com Redsys, aquesta tecnologia de sistemes de pagament comporta tenir un coneixement profund en protocols de seguretat i els processos d'autorització de les compres, a més a més de dur a terme moltes proves i garantir la seguretat i la confiança al client. La probabilitat que aparegui aquest risc és mitjana tenint en compte que necessitem un sistema de pagament com a mínim i l'impacte de no arribar aconseguir aquest objectiu seria molt alt.

Risc	Probabilitat	Impacte
Desviacions de la planificació	MITJANA	ALT
Bugs	ALTA	BAIX
Estimació incorrecta del temps de les tasques	BAIXA	ALT
Complexitat tècnica	MITJANA	ALT

Taula 1.2: Relació dels diferents riscos amb el seu impacte potencial. *(Font pròpia)*

1.8 Metodologia i rigor

1.8.1 Metodologia de treball

Per dur a terme aquest projecte de final de grau s'ha escollit una metodologia *Agile* [12] perquè es caracteritza per la seva flexibilitat i adaptabilitat als canvis.

Per aplicar aquesta metodologia, es combinaran dos marcs de gestió de projectes: *Scrum* i *Kanban*. De la metodologia *Scrum* [13](Figura 1.2) ens beneficiarem la planificació amb *sprints* per poder estimar i distribuir les tasques, de les reunions *Sprint planning, Sprint review i Sprint retrospective* i també del *Product Backlog* que ens permetrà tenir una llista prioritzada amb totes les tasques a realitzar per completar tots els objectius del projecte.

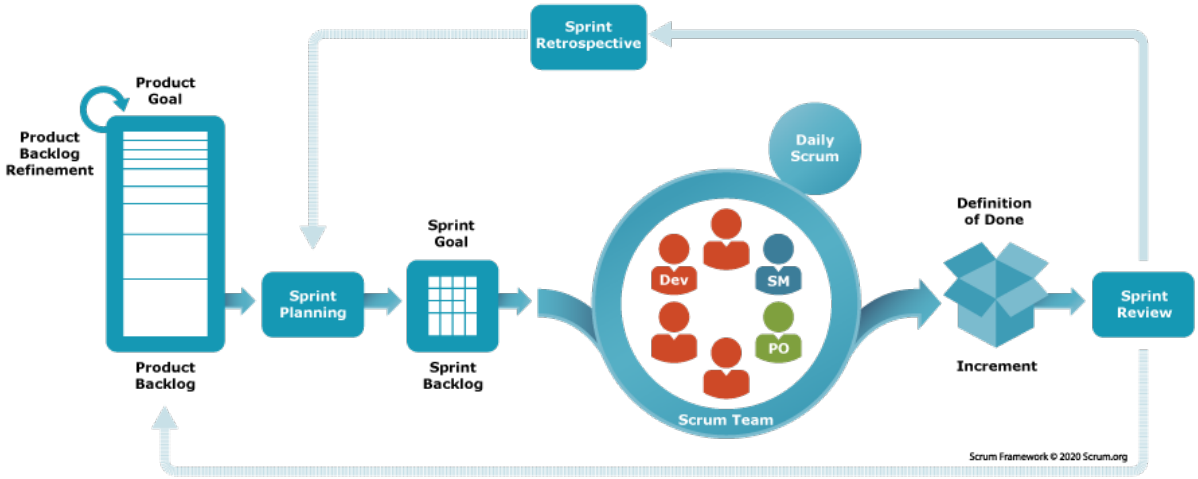

Figura 1.2: Metodologia *Scrum*. (Scrum.org)

D'altra banda, de la metodologia Kanban [14] utilitzarem el taulell (Figura 1.3), ja que ens permet tenir una visió del flux de treball i la càrrega de treball durant els *sprints*.

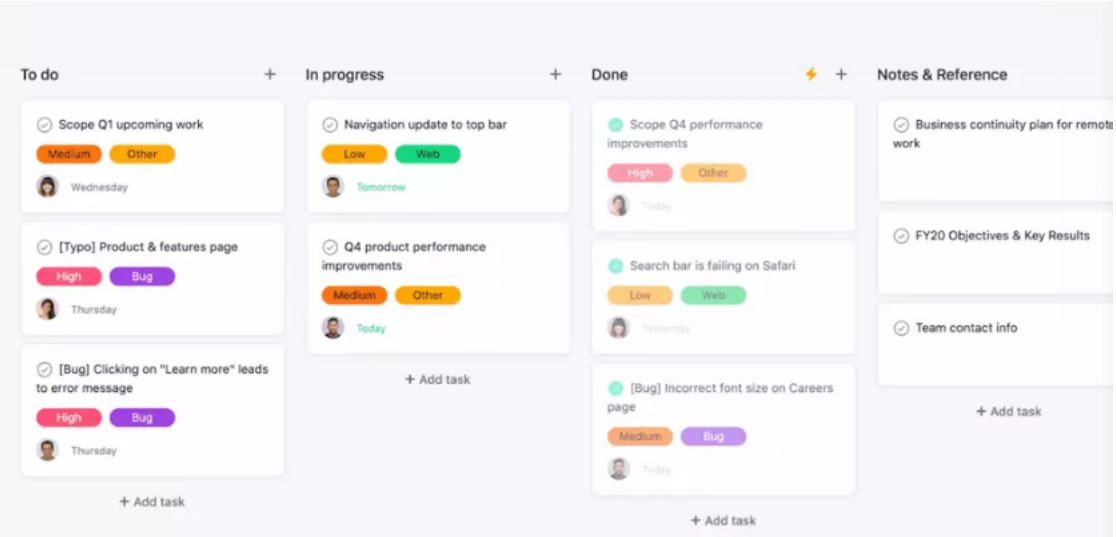

Figura 1.3: Taulell de treball *Kanban*. (Asana)

1.8.2 Eines de seguiment

DESENVOLUPAMENT

Jira

Jira [15] és un *software* que proporciona eines de planificació i seguiment per poder gestionar les èpiques, les tasques i el backlog seguint la metodologia àgil prèviament descrita.

GitHub

GitHub [16] és una plataforma que ofereix la possibilitat de crear i gestionar repositoris de codi utilitzant un sistema de control de versions.

Jenkins

Jenkins [17] és una eina àmpliament utilitzada per a la integració contínua (CI) i el lliurament continu (CD). Permet automatitzar el procés de construcció, test i desplegament de les aplicacions de manera eficient i reiterativa.

GitFlow

L'adopció de la metodologia GitFlow [18] proporciona una estructura clara per al desenvolupament col·laboratiu amb Git [19]. Defineix un conjunt de regles i branques per gestionar el cicle de vida del codi font d'una manera ordenada i eficient.

DOCUMENTACIÓ

Confluence

Confluence [20] és una eina que ofereix un espai de treball col·laboratiu per a la creació de documentació. Amb Confluence, es poden generar pàgines dinàmiques per documentar tots els aspectes d'un projecte o idea.

Overleaf

Overleaf [21] és una plataforma en línia que ofereix un editor LaTeX col·laboratiu per a la creació de documents tècnics i científics. Facilita la col·laboració en temps real entre múltiples autors i ofereix una gran varietat de plantilles i funcionalitats per a la creació de documents de gran qualitat.

Planificació inicial

En aquest apartat, es descriu detalladament la planificació per portar a terme aquest projecte per complir satisfactòriament tots els objectius abans de l'entrega final. Com les imatges incloses en aquest apartat són molt petites i no es poden llegir totes clarament, es pot accedir al pdf amb tota la informació fen clic aquí.

La planificació comença el dia 19 de febrer amb l'assignatura de GEP i finalitza el dia 23 de juny, just abans del període de la defensa del treball. A partir d'aquí s'ha començat a fer la planificació repartint les hores de feina requerides per la normativa de la FIB (540 hores) entre el període definit anteriorment.

A la figura següent 2.1, es poden veure les diferents etapes que tindrà cada fase amb la seva data d'inici i finalització.

Fase	Etapa	Data d'inici	Data de finalització
Gestió del projecte	Contextualització i abast	19-02-2024	27-02-2024
	Planificació temporal	28-02-2024	04-03-2024
	Gestió econòmica i sostenibilitat	05-03-2024	11-03-2024
	Document final	12-03-2024	18-03-2024
Desenvolupament	Inception	19-03-2024	24-03-2024
	Sprint 1	25-03-2024	07-04-2024
	Sprint 2	08-04-2024	21-04-2024
	Sprint 3	22-04-2024	05-05-2024
	Sprint 4	06-05-2024	19-05-2024
	Sprint 5	20-05-2024	02-06-2024
Finalització	Desenvolupament	03-06-2024	09-06-2024
	Documentació	10-06-2024	16-06-2024
	Presentació	17-06-2024	23-06-2024

Figura 2.1: Planificació temporal de les fases i etapes del projecte (Font pròpia)

Després d'estructurar i planificar les diferents etapes entre el període de temps que disposem per fer el treball, hem passat a desenvolupar més detalladament cadascuna d'aquestes etapes juntament amb les duracions, dependències i recursos. A la imatge que trobareu a continuació 2.2, podeu veure cada fase amb un color diferent i dintre de cada fase les diferents etapes. Algunes de les etapes del període de desenvolupament contenen èpiques (assenyalades amb un color gris clar) que ens permeten agrupar les tasques que pertanyen a una funcionalitat específica a desenvolupar per complir amb els objectius.

Codi	Tasca	Duració	Dependències	Recursos
	Gestió del projecte	72		
GP1	Contextualització i abast	24		Overleaf
GP2	Planificació temporal	16	GP1	Overleaf
GP3	Gestió econòmica i sostenibilitat	16	GP2	Overleaf
GP4	Document final	16	GP3	Overleaf
DES	Desenvolupament	372		
	Inception	34		
E	Preparació de l'entorn	18	-	-
D	Definició tècnica de les tasques i èpiques	16	-	-
	Sprint 1	70		
P	Gestió de productes	52	E,D	-
M	Menú i navegació	6	-	-
REV	Sprint Review	2	P, M	Overleaf, Confluence i Jira
RET	Sprint Retrospective	2	REV	Overleaf, Confluence, EasyRetro i Jira
DOC	Documentació	8	-	Overleaf
	Sprint 2	70		
C	Gestió de la cistella de la compra	48	P	-
GB	Gestió de bugs	10	P, M	Jira + altres
REV	Sprint Review	2	C, GB	Overleaf, Confluence i Jira
RET	Sprint Retrospective	2	REV	Overleaf, Confluence, EasyRetro i Jira
DOC	Documentació	8	-	Overleaf
	Sprint 3	70		
R	Gestió de receptes	48	P	-
GB	Gestió de bugs	10	C	Jira + altres
REV	Sprint Review	2	R, GB	Overleaf, Confluence i Jira
RET	Sprint Retrospective	2	REV	Overleaf, Confluence, EasyRetro i Jira
DOC	Documentació	8	-	Overleaf
	Sprint 4	64		
A	Autentificació	28	-	-
F	Gestió de favorits	14	P	-
GB	Gestió de bugs	10	R	Jira + altres
REV	Sprint Review	2	A, F, GB	Overleaf, Confluence i Jira
RET	Sprint Retrospective	2	REV	Overleaf, Confluence, EasyRetro i Jira
DOC	Documentació	8	-	Overleaf
	Sprint 5	64		
B	Búsqueda de productes	42	P	-
GB	Gestió de bugs	10	A, F	Jira + altres
REV	Sprint Review	2	P, GB	Overleaf, Confluence i Jira
RET	Sprint Retrospective	2	REV	Overleaf, Confluence, EasyRetro i Jira
DOC	Documentació	8	-	Overleaf
	Finalització	96		
	Desenvolupament	40		
T	Testing i refactoring	24	DES	Android Studio, XCode, mòbils, VSC, IntelliJ
APP	Desplegar aplicació	16	T	Mòbils
	Documentació	32		
GP5	Memòria final	32	GP4	Overleaf
	Presentació	24		
GP6	Presentació final	24	GP5	PowerPoint
	TOTAL	**540**		

Figura 2.2: Planificació detallada de les fases i etapes del projecte. (Font pròpia)

2.1 Descripció de les tasques

2.1.1 Documentació

GP1. Contextualització i abast: Definir el context, el problema actual i la solució proposta i l'abast del treball juntament amb els seus objectius i requisits.

GP2. Planificació temporal: Definir les tasques a fer, juntament amb les seves estimacions i dependències tenint en compte els riscos definits anteriorment.

GP3. Gestió econòmica i sostenibilitat: Fer un pressupost del projecte i analitzar-ne la sostenibilitat.

GP4. Document final: Unir els documents anteriors tenint en compte el *feedback* dels professors.

GP5. Memòria final: Acabar de redactar i revisar la memòria final del projecte.

GP6. Presentació final: Crear una presentació i fer un guió per a la defensa final.

REV. Sprint Review: Revisar totes les tasques desenvolupades durant l'sprint per comprovar que funcionen correctament. En cas de trobar algun error, s'obrirà un bug que s'avoratrà al següent sprint.

RET. Sprint retrospective: Analitzar la forma de treball que s'ha adquirit durant l'print i buscar punts a millorar.

DOC. Documentació: Redactar la documentació referent a les funcionalitats desenvolupades durant l'sprint.

2.1.2 Desenvolupament

Les tasques de desenvolupament s'han dividit en 2 èpiques de preparació que es realitzaran durant la fase *Inception* i 7 èpiques més, que estaran repartides entre els 5 sprints.

En la següent imatge 2.3, es poden veure detalladament les tasques pertanyents a cada una de les dues èpiques de l'etapa inicial de desenvolupament juntament amb la seva descripció, duració, dependències i recursos materials de software necessaris.

Codi	Tasca	Descripció	Duració	Dependències	Recursos
E	Preparació de l'entorn		18		
E1	[FRONT] - Creació projecte Ionic	Creació del projecte Ionic com a infraestructura del *frontend*.	2	-	VSC
E2	[FRONT] - Implementació "multilanguage" (i18n)	Integració de suport multillenguatge (i18n) a l'aplicació *frontend*.	2	E1	VSC
E3	[BACK] - Creació projecte Spring Boot	Creació del projecte Spring Boot com a infraestructura del servei de *backend*.	2	-	IntelliJ
E4	[BBDD] - Creació base de dades	Inicialització de la base de dades per a l'emmagatzematge de dades de l'aplicació.	2	-	MySQL Workbench, Heidi SQL
E5	[BACK] - Configuració CORS + Base de dades	Configuració del Cross-Origin Resource Sharing (CORS) i integració amb la base de dades al servidor.	2	E3,E4	IntelliJ
E6	[LIQUIBASE] - Integració liquibase	Implementació de Liquibase per a la gestió de canvis i versionat de la base de dades.	4	E3,E4	IntelliJ, Heidi SQL
E7	[CI] - Integració Jenkins	Configuració de Jenkins per a la integració contínua del codi font i la construcció automatitzada del projecte.	4	-	GitHub, Jenkins
D	Definició tècnica de les tasques i èpiques		16		
D1	Model conceptual de dades	Dissenyar un model que relacioni les diverses entitats i els seus atributs, proporcionant una comprensió clara de com s'organitzen les dades.	8	-	Draw.io
D2	Definició de les tasques i èpiques detallades	Especificació de les tasques i èpiques, detallant els requeriments i els criteris de finalització.	8	D1	Jira, Confluence, Overleaf

Figura 2.3: Descripció de les tasques de l'etapa d'*Inception* separades amb èpiques. (Font pròpia)

A continuació, també es poden veure diferents imatges 2.4 2.5 2.6 2.7 2.8 on es descriuen detalladament totes les èpiques i tasques que s'avortaran en cada sprint. Cada èpica pertany a una de les funcionalitats principals del treball.

Codi	Tasca	Descripció	Duració	Dependències	Recursos
P	Gestió de productes		52		
P1	[LIQUIBASE] - Creació de taules	Creació de les taules i atributs necessaris.	2	-	IntelliJ, Heidi SQL
P2	[BACK] - Product controller	Desenvolupament del controlador de productes, responsable de gestionar les operacions relacionades.	2	P1	IntelliJ
P3	[INFO] - Gestió d'imatges dels productes	Cerca d'informació per carregar, emmagatzemar i mostrar les imatges dels productes.	4	-	-
P4	[BACK] - Gestió d'imatges	Desenvolupament de funcionalitats relacionades amb la gestió d'imatges al backend.	12	P3	IntelliJ
P5	[BACK] - Endpoints	Creació de endpoints realacionats amb els productes per a la interacció amb l'aplicació.	20	P2,P4	IntelliJ, Postman
P6	[DISSENY UI/UX] - Disseny grid + detall	Disseny i desenvolupament de l'estructura que ha de tenir la pantalla de la grid de productes i la del detall del producte.	4	-	Figma
P7	[FRONT] - Grid de productes i detall del producte	Implementació del grid de productes i del detall de producte.	8	P6	VSC
M	Menú i navegació		6		
M1	[DISSENY UI/UX] - Disseny barra menú	Disseny i desenvolupament de l'estructura de la barra de menú de l'aplicació.	2	-	Figma
M2	[FRONT] - Barra de menú + navegació	Implementació de la barra de menú i navegació	4	M1	VSC

Figura 2.4: Descripció de les tasques de l'*Sprint 1* separades amb èpiques. (Font pròpia)

Codi	Tasca	Descripció	Duració	Dependències	Recursos
C	Gestió de la cistella de la compra		48		
C1	[LIQUIBASE] - Creació de taules	Creació de les taules i atributs necessaris.	2	-	IntelliJ, Heidi SQL
C2	[BACK] - Cart controller	Desenvolupament del controlador de la cistella de la compra, responsable de gestionar les operacions relacionades.	2	C1	IntelliJ
C3	[INFO] - Sistemes de pagament i gestió de la cistella	Cerca d'informació per poder realitzar el pagament de la compra.	2	-	-
C4	[BACK] - Sistema de pagament Redsys	Desenvolupament de funcionalitats relacionades amb el pagament de la compra.	16	C2, C3	IntelliJ, Postman
C5	[BACK] - Endpoints	Creació de endpoints realacionats amb la cistella de la compra per a la interacció amb l'aplicació.	12	C4	IntelliJ, Postman
C6	[DISSENY UI/UX] - Disseny cistella + pantalla pagament	Disseny i desenvolupament de l'estructura que ha de tenir la pantalla de la cistella de la compra i el pagament.	2	-	Figma
C7	[FRONT] - Cistella de la compra + pantalla pagament	Implementació de la cistella de la compra i la pantalla de pagament.	12	C6	VSC

Figura 2.5: Descripció de les tasques de l'*Sprint 2* separades amb èpiques. (Font pròpia)

Codi	Tasca	Descripció	Duració	Dependències	Recursos
R	Gestió de receptes		48		
R1	[INFO] - Integració IA	Cerca d'informació per poder integrar la IA al nostre sistema.	4	-	-
R2	[BACK] - Controller	Desenvolupament del controlador de les receptes, responsable de gestionar les operacions relacionades.	2	R1	IntelliJ
R3	[BACK] - Endpoints	Creació de endpoints realacionats amb la búsqueda de receptes per a la interacció amb l'aplicació.	24	R2	IntelliJ, Postman
R4	[DISSENY UI/UX] - Disseny pantalla receptes	Disseny i desenvolupament de l'estructura que ha de tenir la pantalla de receptes.	2	-	Figma
R5	[FRONT] - Receptes	Implementació de la pantalla de búsqueda de receptes	16	R4	VSC

Figura 2.6: Descripció de les tasques de l'*Sprint 3* separades amb èpiques. (Font pròpia)

Codi	Tasca	Descripció	Duració	Dependències	Recursos
A	Autentificació		28		
A1	[LIQUIBASE] - Creació de taules	Creació de les taules i atributs necessaris.	2	-	IntelliJ, Heidi SQL
A2	[INFO] - Registre + Inici de sessió	Cerca d'informació per poder fer el registre i l'inici de sessió.	2	-	-
A3	[BACK] - User controller	Desenvolupament del controlador dels usuaris, responsable de gestionar les operacions relacionades.	2	A1, A2	IntelliJ
A4	[BACK] - Endpoints	Creació de endpoints realacionats amb la gestió d'usuaris per a la interacció amb l'aplicació.	16	A3	IntelliJ, Postman
A5	[DISSENY UI/UX] - Disseny perfil usuari	Disseny i desenvolupament de l'estructura que ha de tenir la pantalla de perfil.	2	-	Figma
A6	[FRONT] - Perfil	Implementació de la pantalla de perfil de l'usuari.	4	A5	VSC
F	Gestió de favorits		14		
F1	[LIQUIBASE] - Creació de taules	Creació de les taules i atributs necessaris.	2	-	IntelliJ, Heidi SQL
F2	[BACK] - Endpoints	Creació de endpoints realacionats amb la gestió de favorits per a la interacció amb l'aplicació.	8	F1	IntelliJ, Postman
F3	[DISSENY UI/UX] - Disseny favorits	Disseny i desenvolupament de l'estructura que ha de tenir la pantalla de favorits.	2	-	Figma
F4	[FRONT] - Favorits	Implementació de la pantalla de favorits.	4	F3	VSC

Figura 2.7: Descripció de les tasques de l'*Sprint 4* separades amb èpiques. (Font pròpia)

Codi	Tasca	Descripció	Duració	Dependències	Recursos
B	Búsqueda de productes		42		
B1	[BACK] - Filtratge de productes	Creació de endpoints realacionats amb la gestió del filtratge de productes per a la interacció amb l'aplicació.	8	-	IntelliJ, Postman
B2	[BACK] - Recomanacions personalitzades	Creació de endpoints realacionats amb les recomanacions personalitzades per a la interacció amb l'aplicació.	16	-	IntelliJ, Postman
B3	[DISSENY UI/UX] - Disseny pantalla búsqueda	Disseny i desenvolupament de l'estructura que ha de tenir la pantalla de búsqueda.	2	-	Figma
B4	[FRONT] - Pantalla búsqueda	Implementació de la pantalla de búsqueda.	16	B3	VSC

Figura 2.8: Descripció de les tasques de l'*Sprint 5* separades amb èpiques. (Font pròpia)

Finalment, a la fase de 'Finalització', dintre de l'etapa de 'Desenvolupament' tenim dues tasques: 'Testing i refactoring' i 'Desplegar aplicació'. La primera tasca consiteix en acabar de revisar el funcionament de tota la app i comprovar que no hi hagi cap error en alguna funcionalitat o estil i, la segona tasca consisteix en desplegar l'aplicació per a Android i iOS i comprovar-ne el seu correcte funcionament.

A la pàgina següent, es pot veure el **diagrama de Gantt** 2.9 que representa tota la planificació del nostre projecte.

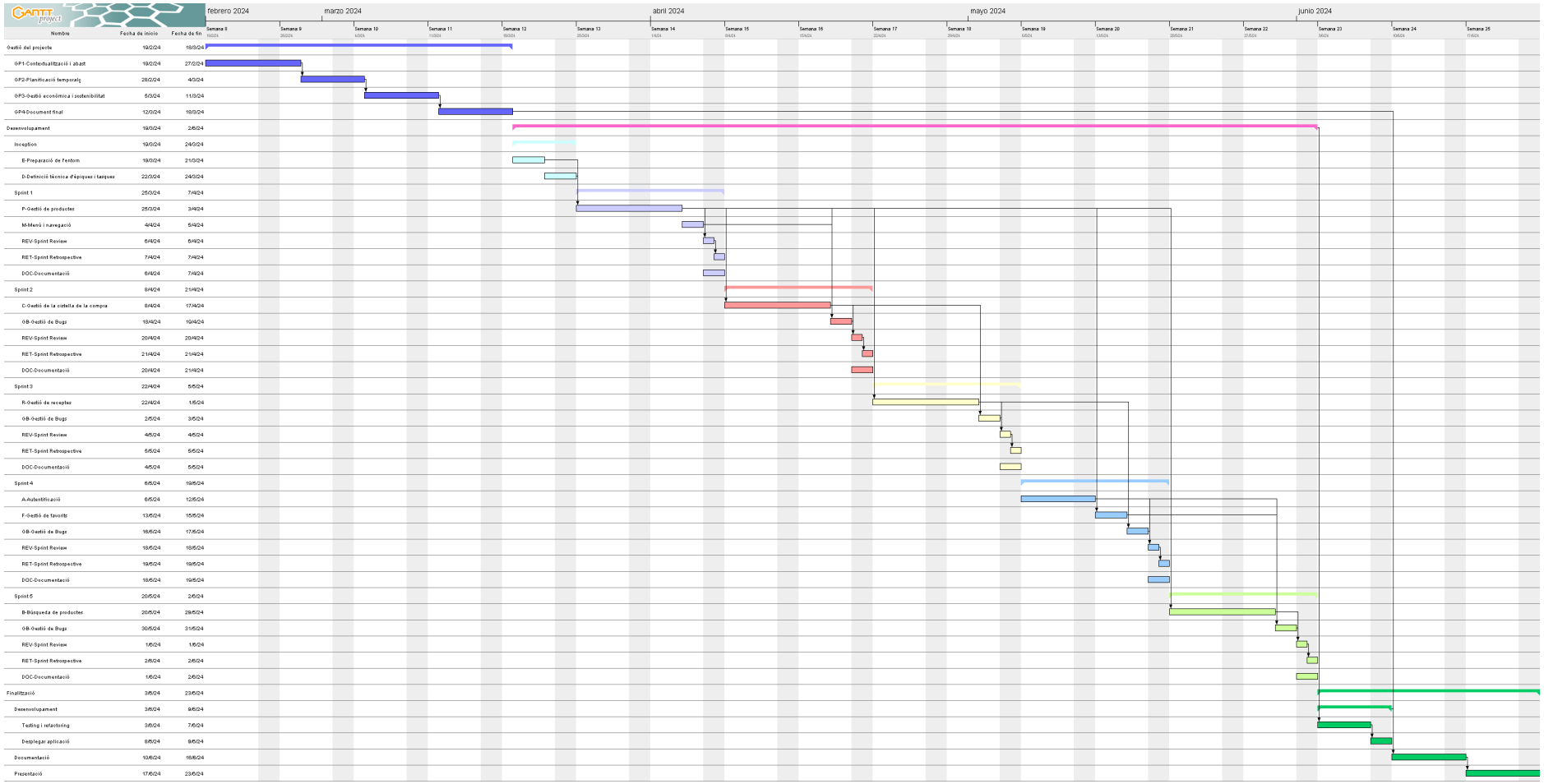

Figura 2.9: Diagrama de Gantt. (Font pròpia)

2.2 Recursos

2.2.1 Recursos humans

- **Director del TFG:** El seu rol és supervisar i ajudar l'estudiant a complir els seus objectius.
- **Autor del TFG:** El seu rol és planificar i desenvolupar la solució proposta. També s'encarregarà de testejar l'aplicació en l'àmbit tècnic.
- **Usuaris de test:** El rol d'aquestes persones serà testejar l'aplicació abans de desplegar-la i analitzar totes les seves opinions i sensasions per saber si es compleixen alguns dels requisits no funcionals.

2.2.2 Recursos materials

Hardware: Ordinador portàtil Windows i iOS, dispositius mòbils Android i iOS, teclat mecànic, ratolí i monitor.

Software:

- **Visual Studio Code:** IDE per desenvolupar el projecte *forntend*.
- **IntelliJ IDEA:** IDE per a desenvolupament el projecte *backend*.
- **MySQL Workbench:** Eina de disseny i administració de bases de dades MySQL.
- **HeidiSQL:** Eina que ens permet administrar bases de dades amb una interfície gràfica d'usuari.
- **Postman:** Plataforma de col·laboració per al desenvolupament d'API.
- **SourceTree:** Eina de control de versions per al projecte del *backend*.
- **Figma:** Eina de disseny d'interfícies d'usuari i prototipatge.
- **GanttProject:** Aplicació per realitzar un diagrama de Gantt.
- **Draw.io:** Eina de creació de diagrames en línia.
- **Android Studio:** Entorn de desenvolupament integrat per a aplicacions Android.
- **XCode:** Entorn de desenvolupament integrat per a aplicacions iOS i macOS.
- **Google Drive:** Eina d'emmagatzematge de documents per crear les taules de planificació del projecte.

A més a més de totes aquestes eines, també s'utilitzaran els recursos descrits en l'apartat d'eines de seguiment (Jira, GitHub, Jenkins, GitFlow, Confluence i Overleaf) 1.8.2.

Finalment, cal destacar també que es necessita un compte de desenvolupador tant per a publicar la app a Google Play com per a App Store.

Espai de treball:

Es necessita un espai de treball on poder treballar i fer les reunions. En aquest cas, en ser un projecte de final de grau, el treball estarà realitzat per una única persona i tota la feina es farà des de casa. Tot i això, es valorarà la necessitat d'un espai de treball per un equip 4 persones en una oficina de cotreball de Barcelona que compleixi amb els requisits necessaris.

2.3 Gestió de riscs

2.3.1 Desviacions de la planificació i estimacions incorrectes de les tasques

Aquests riscs poden tenir un impacte alt, ja que l'entrega del treball té una data fixa i en cas que ens desviem de la planificació, el projecte no es podrà acabar i no complirà amb tots els objectius marcats. Un gran factor que pot influenciar aquesta desviació és la inexperiència en certes tecnologies, és per això que, hem afegit algunes **tasques destinades únicament a buscar informació i tenir clar com implementar certes funcionalitats**. A més a més, aquesta **planificació** també ha estat **supervisada** per altres persones amb més experiència per revisar que les hores assignades siguin suficients.

2.3.2 Bugs

Aquest risc és molt probable que ocorri, és per aquest motiu que durant els sprints de desenvolupament s'ha planificat un període de **40 hores de gestió de bugs** i també s'ha creat una **tasca de *testing* i *refactoring*** al final de tot el desenvolupament per acabar de revisar qualsevol error que pugui aparèixer.

2.3.3 Complexitat tècnica

Hem considerat que aquest problema es pot presentar en unes èpiques concretes com ara 'Gestió de la cistella de la compra' o 'Gestió de receptes'. Tenint en compte això,

hem estructurat la planificació de tal forma que **totes dues èpiques es desenvolupen individualment en dos sprints diferents**, sense haver de desenvolupar cap altra èpica més. D'aquesta manera, el programador podrà estar centrat al 100% amb aquestes complexitats, no perdrà el focus i podrà dedicar més hores a desenvolupar-les.

2.3.4 Averia de l'ordinador

Aquest risc és poc probable però en cas que es produeixi, l'impacte pot ser molt alt ja que comportaria possibles pèrdues de dades, codi del projecte... Si això succeeix, s'hauria de portar el portàtil a un tècnic i s'haurien de calcular les desviacions produides en la planificació.

Risc	**Probabilitat**	**Impacte**
Desviacions de la planificació i estimació incorrecta del temps de les tasques	MITJANA/BAIXA (25%)	ALT
Bugs	ALTA (50%)	BAIX
Complexitat tècnica	MITJANA (30%)	ALT
Averia de l'ordinador	BAIXA (15%)	ALT

Taula 2.1: Relació dels diferents riscos amb el seu impacte potencial. *(Font pròpia)*

Gestió econòmica

En aquest apartat, es presenta el pressupost inicial del projecte on s'ha tingut en compte tant els costos de fer les tasques com els costos de lloguer, materials necessaris i imprevists. Tal com s'ha explicat a l'apartat anterior, hi ha imatges que són petites, així que per facilitar-ne la lectura es pot fer clic aquí.

3.1 Pressupost inicial

3.1.1 Costos humans

Per calcular aquests costos, primerament s'han identificat els rols necessaris per a la realització del projecte i a partir d'aquí, s'ha determinat el cost per hores de cadascun. Un cop coneguts els sous de cada rol, s'ha fet una previsió del nombre d'hores que han de dedicar a cada tasca i s'ha calculat el nombre d'hores totals (afegida als annexos A.1). Finalment, s'ha estimat el cost total dels recursos humans multiplicant el nombre d'hores dedicades pel cost total per hora 3.1.

COSTOS HUMANS	Sou brut / any	Sou brut / hora	Costos SS / any [18]	Costos SS / hora	Cost total / hora	Hores treballades	Cost total
Cap de projecte [14]	34.236,00 €	16,46 €	11.434,82 €	5,50 €	21,96 €	197	4.325,55 €
Junior Developer [15]	24.000,00 €	11,54 €	8.016,00 €	3,85 €	15,39 €	302	4.648,48 €
Junior QA [16]	22.768,00 €	10,95 €	7.604,51 €	3,66 €	14,60 €	25	365,05 €
Dissenyador UI/UX [17]	25.000,00 €	12,02 €	8.350,00 €	4,01 €	16,03 €	16	256,54 €
TOTAL							**9.595,62 €**

Figura 3.1: Costos dels recursos humans. (Font pròpia)

3.1.2 Costos de software

Tot i que per realitzar aquest projecte no utilitzem eines de desenvolupament que ens suposin un cost, sí que cal considerar el preu que costaria publicar l'aplicació a la plataforma Google Play i App Store 3.2.

COSTOS SOFTWARE	Preu
Compte desenvolupador Google Play [19]	25,00 €
Compte desenvolupador App Store [19]	100,00 €
TOTAL	**125,00 €**

Figura 3.2: Costos de software. (Font pròpia)

3.1.3 Costos de hardware

Per altra banda, també necessitem calcular el pressupost destinat als ordinadors, mòbils i altres elements emprats per desenvolupar el projecte. Hisenda permet amortitzar aquests equips en 4 anys, per tant, s'ha aplicat aquesta fórmula per determinar els costos físics.

$$\text{Amortització}(€) = \frac{\text{Preu}(€) \cdot \text{Temps d'ús}}{\text{Vida útil} \cdot 224(\text{dies laborals/any}) \cdot 8(\text{h/dia})}$$

A la taula següent 3.3, podem veure l'amortització de cada aparell aplicant la fórmula anterior.

COSTOS MATERIALS	Preu	Vida útil	Temps d'ús	Amortització
Portàtil Windows - MSI Modern 14 i7/16GB/1TB SSD [20]	1.398,86 €	4	540	105,38 €
Portàtil iOS - MacBook Pro M3 [21]	2.029,00 €	4	24	6,79 €
Mòbil Android - Xiaomi Redmi 10A [22]	139,00 €	4	40	0,78 €
Mòbil iOS - Iphone 13 [23]	649,00 €	4	40	3,62 €
Teclat - KLIM Chroma Tecalado con cable [24]	29,97 €	4	540	2,26 €
Ratolí - VIVIJO Ratón Gaming con Cable [25]	19,99 €	4	540	1,51 €
Monitor - Monitor FHD ThinkVision E24-28 de 23,8" [26]	182,01 €	4	540	13,71 €
TOTAL				**134,05 €**

Figura 3.3: Costos de hardware. (Font pròpia)

3.1.4 Costos genèrics

A continuació, s'ha calculat el cost relacionat amb recursos involucrats en l'espai de treball 3.4. En aquest cas, en ser un projecte de final de grau, el treball estarà realitzat per una única persona que assumirà els 4 rols identificats i tota la feina es farà des de casa. Tot i això, s'ha valorat un càlcul en el supòsit que treballen un equip de 4 persones en una oficina de cotreball de Barcelona. En concret, en l'oficina d'Aurea co-working [35], que inclou servei d'Internet a alta velocitat, electricitat, calefacció i aire condicionat, accés a impressora i escàner, sala de reunions i accés les 24 hores del dia i 7 dies a la setmana.

Pla mensual	IVA	Cost total / mes	Mesos	Persones	Total
210,00 €	44,10 €	254,10 €	5	4	**5.082,00 €**

Figura 3.4: Costos genèrics. (Font pròpia)

3.1.5 Contingències

A més a més dels costos previstos, cal calcular el cost de contingència que s'ha de reservar per possibles imprevistos que no es poden anticipar. Aquest cost es calcula com un percentatge del valor total del pressupost. Aquest percentatge sol estar entre el 10 i el 20% en un projecte de software. En el nostre cas, s'ha decidit aplicar el 15%.

3.1.6 Costos d'imprevistos

Finalment, per acabar amb l'estimació de costos, cal calcular el preu que caldria destinar a possibles imprevistos que poden sorgir segons la seva probabilitat i impacte. Com ja s'ha comentat a l'apartat de gestió de riscos 2.3, s'han destinat diferents tasques a la gestió d'aquests possibles riscos com per exemple tasques de gestió de bugs o tasques de cerca d'informació per evitar possibles desviaments en tasques amb alta complexitat. Tot i això, a més a més d'aquestes hores ja incorporades dintre del diagrama de Gantt, també s'ha anticipat un càlcul del cost que podria suposar en cas que hi hagi algun imprevist no esperat.

A la següent taula 3.5, s'ha presentat la possibilitat d'haver d'augmentar el nombre d'hores destinades a la gestió de bugs, el nombre d'hores de cerca d'informació per a tasques d'alta complexitat o el simple fet d'haver d'augmentar recursos per haver estimat malament alguna tasca. També s'ha fet una previsió del cost que tindria el fet d'haver de reparar l'ordinador en cas de possible avaria.

Imprevist	Temps	Cost estimat	Probabilitat	Cost
Averia del portàtil	10	150,00 €	15	22,50 €
Bugs	20	307,85 €	50	153,92 €
Complexitat tècnica	30	461,77 €	30	138,53 €
Desviacions de la planificació i estimacions incorrectes de les tasques	30	461,77 €	25	115,44 €
TOTAL				**430,40 €**

Figura 3.5: Costos d'imprevistos. (Font pròpia)

3.1.7 Costos totals

El pressupost final 3.6 ve determinat per la suma de tots els costos anteriors.

COSTOS TOTALS	Cost
Costos humans (CPA)	9.595,62 €
Costos software	125,00 €
Costos materials	134,05 €
Costos genèrics	5.082,00 €
CG (software, materials, genèrics)	5.341,05 €
Total costos (CPA + CG)	14.936,67 €
Contingència (15%)	2.240,50 €
Total (CD + CI + Contingència)	17.177,17 €
Costos imprevistos	430,40 €
TOTAL	**17.607,57 €**

Figura 3.6: Costos totals. (Font pròpia)

3.2 Control de gestió

En aquesta secció, s'explica el control que es durà sobre les possibles desviacions en el pressupost final que es poden trobar al llarg del projecte. Per a això, caldrà anotar el temps real que ens ha representat desenvolupar cada tasca. D'aquesta manera, es pot saber la desviació que hi ha al pressupost i si s'està dins del que s'ha pressupostat en un inici. Per tal de tenir aquest control es defineixen les següents mètriques:

Desviació d'hores per tasca:

(Hores estimades - Hores reals) x Cost/hora estimat

Desviació en el cost dels recursos humans:

(Cost/hora estimat - Cost/hora real) x Hores reals

Desviació del cost de personal per activitat:

Cost estimat - Cost real

Desviació en les amortitzacions:

(Temps ús estimat - Temps ús real) x Cost/hora

Desviació total hores:

Hores totals estimades - Hores totals reals

Desviació total cost amortitzacions:

Cost amortització estimat - Cost amortització real

Desviació total pressupost:

Pressupost total estimat - Pressupost total real

Després de finalitzar el període de desenvolupament del projecte, es procedirà a calcular les desviacions per a avaluar el pressupost i decidir si és necessari utilitzar la contingència prevista al començament del projecte. Això permetrà un anàlisi exhaustiu del pressupost.

Especificació de requisits

4.1 Requisits funcionals

4.1.1 Descripció d'histories d'usuari

Gestió de productes

Història d'usuari	Gestió dels productes de la base de dades
Com	Administrador de les dades
Vull poder	Crear o eliminar un producte
Per	Poder gestionar més fàcilment el catàleg de productes existent.
Criteris d'acceptació	• Ha d'existir una crida a l'API que permeti crear un producte amb les dades bàsiques (preu, marca, imatge...) amb més d'un idioma a la vegada i amb tots els seus atributs nutricionals necessaris. • Ha d'existir una crida a l'API que permeti eliminar un producte i totes les seves referències.

Història d'usuari	Veure llistat de tots els productes
Com	Usuari de l'aplicació
Vull poder	Veure tots els productes disponibles
Per	Poder veure les dades de tots productes i facilitar la presa de decisions sobre els productes que desitjo adquirir.
Criteris d'acceptació	• Ha d'haver-hi una pantalla que permeti veure tots els productes disponibles. • Cada producte ha de mostrar com a mínim la seva imatge, el nom, la marca del producte i el preu. • S'han de poder ordenar els productes segons el nom i el preu. • En fer clic sobre un producte del llistat, s'ha de mostrar una vista detallada amb més informació sobre el producte (ingredients, valor nutricional, descripció del producte...) • Tant des del llistat de productes com des del detall del producte, s'han de poder realitzar les accions d'afegir el producte a la cistella com afegir-lo al llistat de preferits.

Gestió de la cistella de la compra

Història d'usuari	Fer una compra
Com	Usuari de l'aplicació
Vull poder	Afegir i eliminar productes a la meva cistella de la compra
Per	Fer la compra de manera eficient.
Criteris d'acceptació	• Selecció de productes: – L'usuari ha de poder seleccionar productes des del catàleg i afegir-los a la cistella amb la quantitat desitjada o eliminar-los. – S'ha de poder eliminar o modificar la quantitat de qualsevol element del llistat de productes afegits a la cistella de la compra. • Visualització de la cistella – Després de cada acció que modifiqui la cistella, la visualització de la cistella ha de reflectir immediatament els canvis. – Ha de mostrar-se el preu total de tots els productes de la cistella de la compra. • Procediment de pagament: – L'usuari ha de poder completar el pagament de la compra introduint les seves dades de la targeta de crèdit/dèbit. – Un cop feta la compra, la cistella de la compra de l'usuari ha de quedar buida.

Gestió de receptes

Història d'usuari	Obtenir un llistat de productes de cada ingredient necessari per cuinar una recepta
Com	Usuari de l'aplicació
Vull poder	Veure tots els productes necessaris per cuinar la recepta especificada
Per	Facilitar el procés de compra i poder afegir conjuntament tots els ingredients necessaris.
Criteris d'acceptació	• Ha d'haver-hi una pantalla que permeti introduir una recepta (ex: macarrons a la bolonyesa) i retorni un llistat de productes per cada ingredient necessari. • Cada producte ha de mostrar com a mínim la seva imatge, el nom i el preu. • S'ha de poder seleccionar o desseleccionar un producte segons si es vol afegir a la cistella de la compra o no. Per cada ingredient s'ha de poder seleccionar com a màxim un producte. • Ha de mostrar-se el preu total de tots els productes seleccionats. • Ha d'haver-hi un botó que permeti afegir tots els productes seleccionats a la cistella de la compra.

Gestió de favorits

Història d'usuari	Afegir o eliminar un producte de favorits
Com	Usuari de l'aplicació
Vull poder	Afegir o eliminar un producte al meu llistat de favorits
Per	Tenir un accés ràpid als productes que més m'agraden sense haver de buscar-los cada vegada
Criteris d'acceptació	• L'usuari ha de poder afegir/eliminar un producte als favorits mitjançant un botó tant des de la grid de productes com des del detall del producte com des del llistat de favorits. • Els productes afegits als favorits han d'aparèixer en un llistat a la secció de productes favorits de l'aplicació. • Quan un producte s'ha afegit als favorits, el botó canvia per poder eliminar-lo de favorits, i viceversa. • El llistat de favorits es guarda de manera persistent, de manera que l'usuari el pugui recuperar en iniciar sessió de nou.

Història d'usuari	Afegir o eliminar una recepta de favorits
Com	Usuari de l'aplicació
Vull poder	Afegir o eliminar una recepta del meu llistat de favorits
Per	Tenir un accés ràpid a les receptes que més m'agraden sense haver de buscar-les cada vegada
Criteris d'acceptació	• L'usuari pot afegir una recepta als favorits mitjançant un botó a la pàgina de la recepta. • La recepta s'ha de poder guardar amb el nom que l'usuari vulgui, però no poden haver-hi dues receptes amb el mateix nom. • L'usuari pot eliminar una recepta dels favorits mitjançant un botó des de la pàgina de favorits. • Les receptes afegides als favorits han d'aparèixer en un llistat a la secció de receptes favorites de l'aplicació. • El llistat de receptes favorites es guarda de manera persistent, de manera que l'usuari el pugui recuperar en iniciar sessió de nou.

Autentificació

Història d'usuari	Iniciar sessió a l'aplicació
Com	Usuari de l'aplicació
Vull poder	Iniciar sessió al meu compte de l'aplicació
Per	Accedir a les funcionalitats personalitzades del meu compte
Criteris d'acceptació	• L'usuari ha de poder introduir el seu correu electrònic i contrasenya en els camps proporcionats. – Si les credencials són correctes, l'usuari és redirigit a la pàgina principal de l'aplicació. – Si les credencials són incorrectes, es mostra un missatge d'error indicant que el correu electrònic o la contrasenya són incorrectes.

Història d'usuari	Registrar-se a l'aplicació
Com	Nou usuari de l'aplicació
Vull poder	Registrar-me creant un nou compte
Per	Accedir a les funcionalitats de l'aplicació
Criteris d'acceptació	• L'usuari ha d'introduir el seu correu electrònic, contrasenya, nom i cognoms (el mòbil i la data de naixement han de ser opcionals) per poder completar el procés de registre. • La contrasenya ha de tenir almenys 6 caràcters, incloent-hi una majúscula, una minúscula, un número i un caràcter especial. • Si l'usuari omple tots els camps obligatoris i compleix amb els requisits de la contrasenya, es pot crear un nou compte. • Si algun camp obligatori no està completat o la contrasenya no compleix amb els requisits, s'ha de mostrar un missatge d'error indicant què falta o què cal corregir. • Després de registrar-se correctament, l'usuari ha de ser redirigit a dintre de l'aplicació.

4.2 Requisits no funcionals

Per especificar els requisits no funcionals del nostre projecte ens basarem en la plantilla de Volere [11]. A continuació, trobem el llistat de requisits que ha de complir la nostra solució per garantitzar el correcte funcionament de l'aplicació.

Tipus de requisit	10a. Requisit d'aparença
Descripció	L'aspecte visual de l'aplicació ha de ser atractiu i coherent.
Criteri d'acceptació	Al 80% dels 30 usuaris enquestats els ha de semblar atractiu.

Tipus de requisit	11a. Requisit de facilitat d'ús
Descripció	L'usuari ha de poder utilitzar l'aplicació sense tenir cap coneixement sobre aquesta. Ha de ser intuïtiva.
Criteri d'acceptació	Al 80% dels 30 usuaris enquestats els ha de semblar atractiu.

Tipus de requisit	12a. Requisit de velocitat i latència
Descripció	El sistema ha de donar resposta a les interaccions en un curt període de temps.
Criteri d'acceptació	Les interaccions i les crides a la API no poden durar més de 5 segons.

Tipus de requisit	12d. Requisit de confiança i disponibilitat
Descripció	El sistema ha de garantir una disponibilitat elevada, amb temps de caiguda mínims, i ha de mantenir la integritat de les dades de manera fiable en tot moment.
Criteri d'acceptació	El sistema no pot deixar de funcionar més de 5 hores a la setmana.

Tipus de requisit	12g. Requisit d'escalabilitat
Descripció	El sistema ha de ser capaç d'adaptar-se i gestionar un augment significatiu en la demanda o el nombre d'usuaris sense veure afectat el seu rendiment o la seva disponibilitat.
Criteri d'acceptació	L'escalabilitat es provarà mitjançant proves de càrrega.

Tipus de requisit	14c. Requisit d'adaptabilitat
Descripció	L'aplicació ha d'estar disponible tant per a dispositius iOS com per a dispositius Android.
Criteri d'acceptació	L'aplicació es desenvoluparà amb Ionic, un *framework* multiplataforma per desenvolupar aplicacions híbrides.

4.3 Model conceptual de dades

4.3.1 Gestió de productes

A la figura 4.1 es pot veure el diagrama conceptual de dades que descriu les principals entitats i atributs que forment part de la gestió de productes.

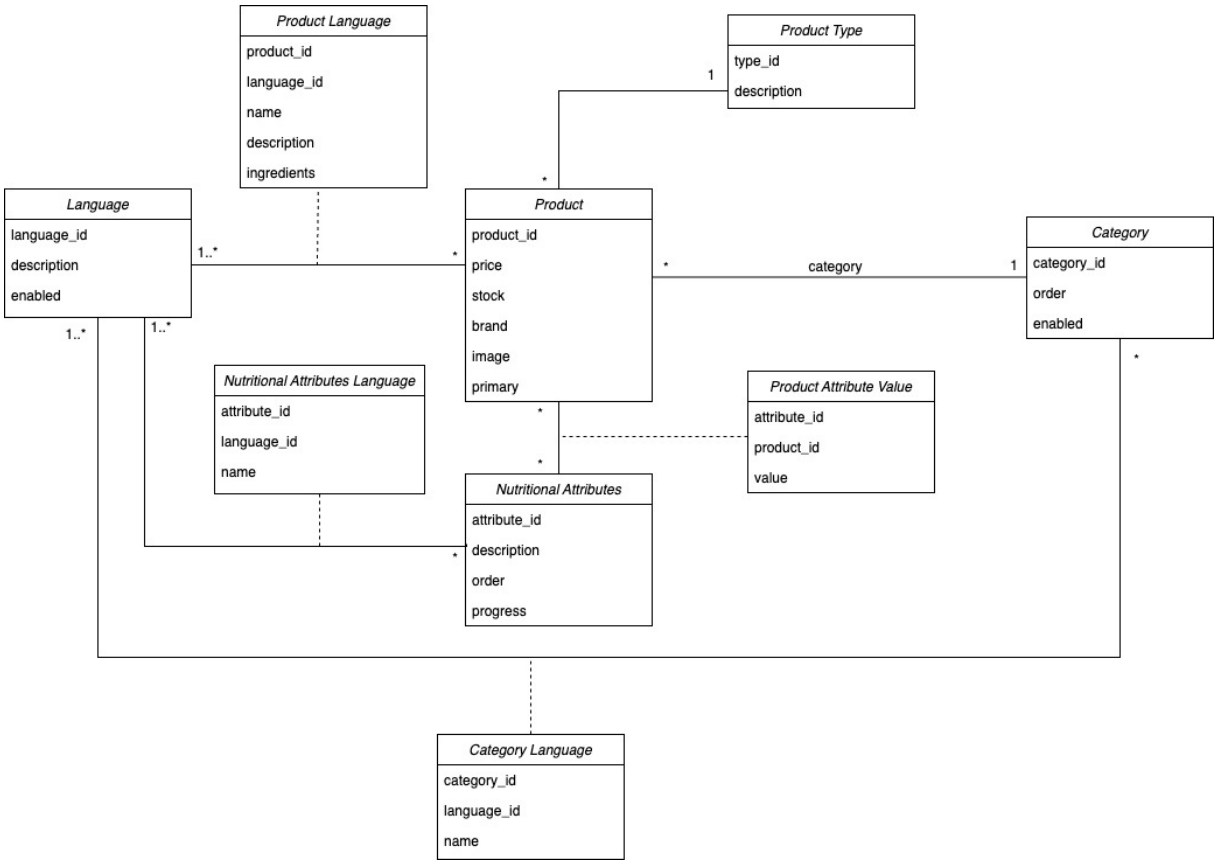

Figura 4.1: Model conceptual de dades - Gestió de productes. (Font pròpia)

Si observem la imatge anterior i ens focalitzem en l'entitat *Product*, podem veure que cada producte a més a més dels seus atributs, ha de tenir associada una categoria i un tipus. La categoria fa referència a la classificació alimentària dintre de la tenda (alimentació, beguda, cura personal, neteja...) i el tipus especifica quin tipus d'informació nutricional té el producte. Per altra banda, cada producte ha de tenir una associació amb la taula idioma que permet obtenir els atributs com el nom del producte, la descripció i els seus ingredients per cada idioma. D'aquesta manera es fa possible implementar una aplicació amb múltiples idiomes. Finalment, tenim l'entitat *Nutritional Attributes* la qual també té associada una relació amb la taula *Language* per la mateixa raó que els productes i també està associada amb els productes per poder declarar a cada producte el valor específic de cada atribut nutricional.

4.3.2 Gestió de la cistella de la compra

A la figura 4.2 es pot veure el diagrama conceptual de dades que descriu les principals entitats i atributs que forment part de la gestió de la cistella de la compra.

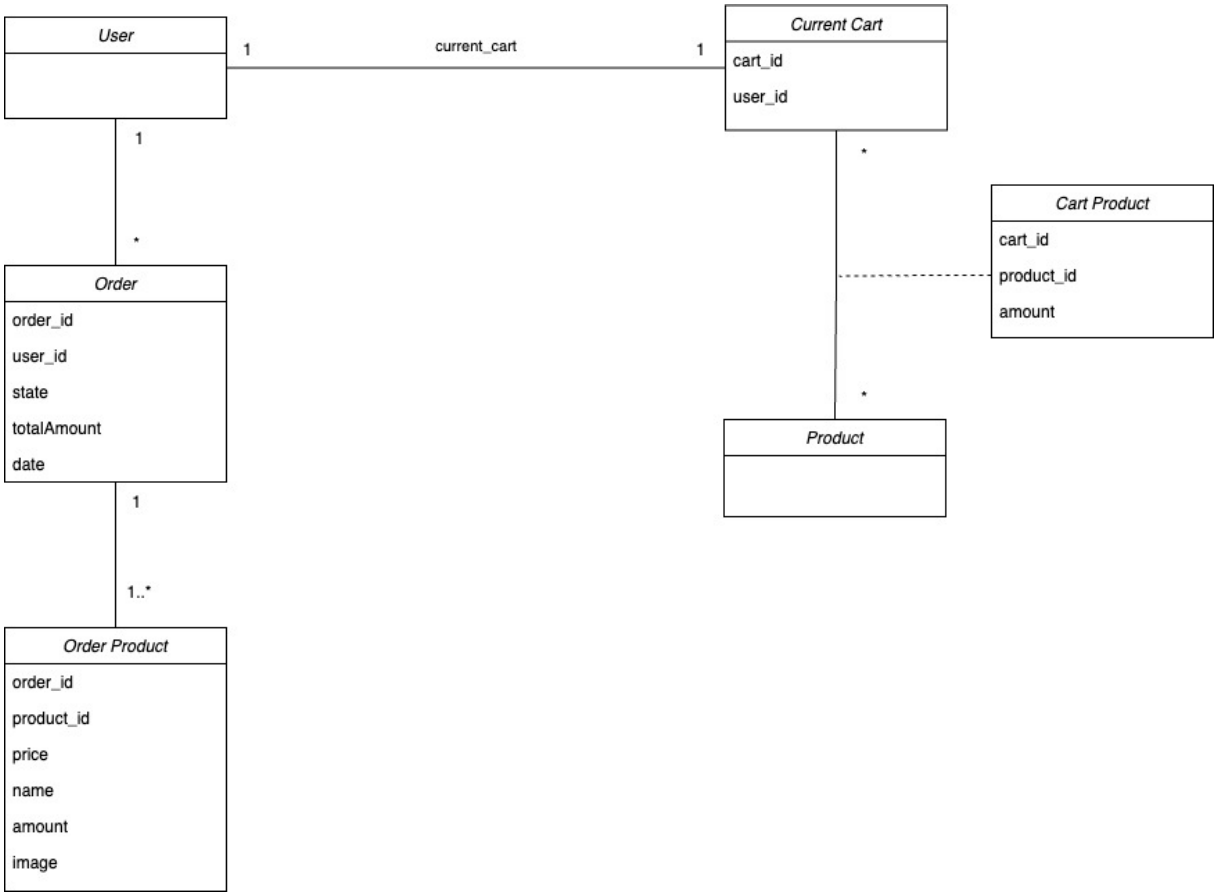

Figura 4.2: Model conceptual de dades - Gestió de la cistella de la compra. (Font pròpia)

Si ens fixem en la imatge, es pot veure clarament, que cada usuari té associat una única cistella de la compra. Aquesta cistella pot contenir múltiples *Cart Product* que representen els diferents productes que l'usuari ha afegit a la cistella de la compra. Aquestes entitats estan associades també a l'entitat *Product*. Per altra banda, un usuari pot tenir diferents comandes amb els atributs corresponents on cada comanda conté com a mínim un *Order Product*.

4.3.3 Gestió de receptes

Per a l'èpica de gestió de receptes, no s'ha creat cap taula específica per emmagatzemar les receptes o els seus ingredients a la base de dades. En lloc d'això, els ingredients de les receptes són obtinguts per intel·ligència artificial (IA). Això permet una major flexibilitat i eficiència en la gestió de receptes, ja que la IA pot adaptar-se dinàmicament a les preferències i necessitats dels usuaris en temps real.

4.3.4 Gestió de favorits

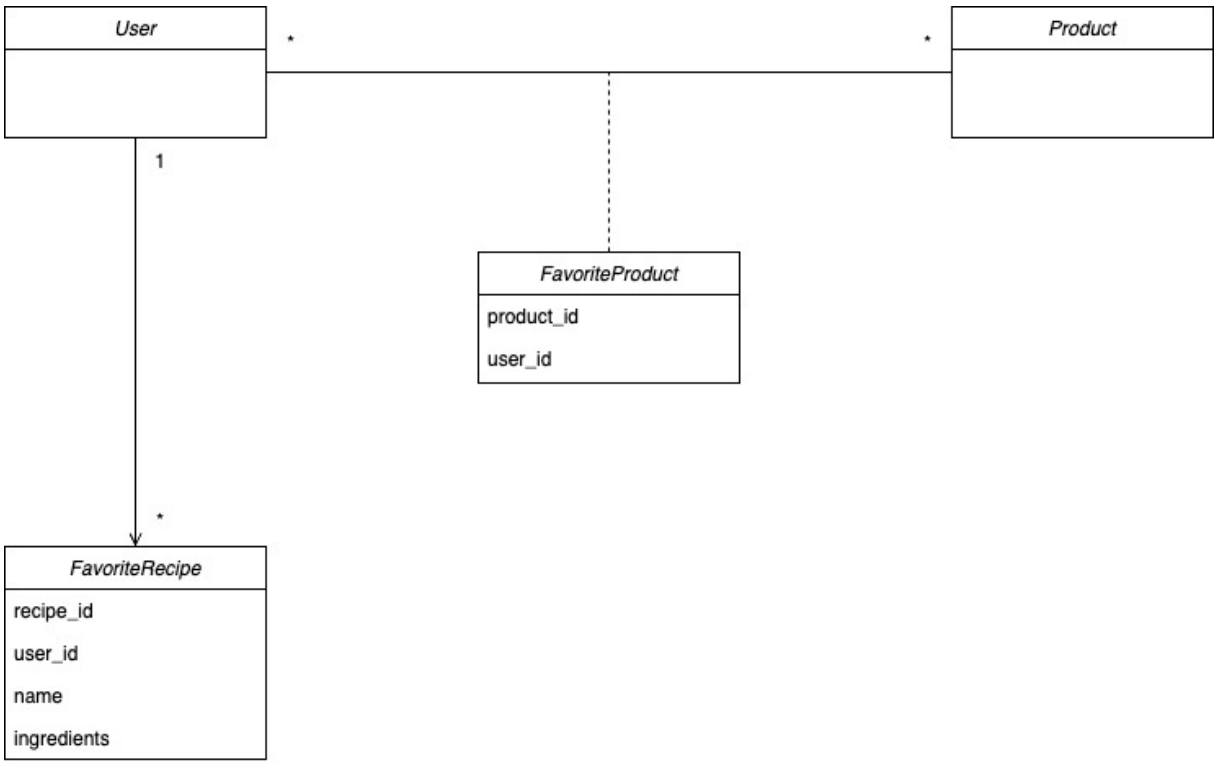

Figura 4.3: Model conceptual de dades - Gestió de favorits. (Font pròpia)

A la figura 4.3 es pot veure que els usuaris poden guardar tant productes a favorits com receptes. Per registrar els productes favorits d'un usuari s'ha creat la taula *FavoriteProduct* que emmagatzema l'identificador de l'usuari juntament amb el producte afegit a preferits. Per altra banda, per fer la gestió de les receptes favorites dels usuaris, s'ha creat la taula *FavoriteRecipe* que guarda l'identificador de l'usuari que ha guardat aquesta recepta, el nom amb el qual ha decidit guardar la recepta i els seus ingredients.

4.3.5 Gestió d'usuaris

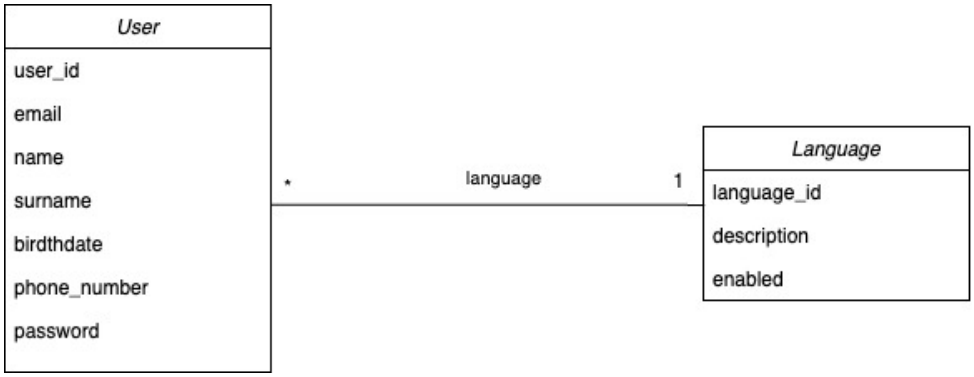

Figura 4.4: Model conceptual de dades - Gestió d'usuaris. (Font pròpia)

Finalment, per a la gestió d'usuaris tenim la taula *User* que s'ha estat mostrant als diagrames anteriors i detallada en aquest apartat 4.4. Aquesta taula conté un identificador, el correu amb el que l'usuari s'ha registrat, el nom i cognoms, la contrasenya encriptada i l'idioma fent referència a un dels idiomes existents a la taula *Language*. A més a més, si l'usuari ho registra, també es pot guardar el mòbil i la data de naixement. Aquesta taula és també refrenciada per altres taules ja mencionades anteriorment.

Arquitectura

5.1 Disseny de l'arquitectura del sistema

L'arquitectura del sistema segueix un model de 3 capes, cada una amb responsabilitats ben definides per a la presentació, la lògica de negoci i la gestió de dades. Aquesta estructura modular facilita el desenvolupament, manteniment i escalabilitat de l'aplicació.

Capa de presentació

La capa de presentació està formada per l'aplicació mòbil desenvolupada amb Ionic per a dispositius Android i iOS. Aquesta capa és la responsable de la interacció amb l'usuari, de la presentació de la informació de manera atractiva i funcional i de la comunicació amb l'API.

Capa de negoci

La capa de negoci és gestionada per l'aplicació Spring Boot, que s'encarrega del processament de les crides HTTP i la gestió de les dades. Aquesta capa inclou els següents components:

- **Controladors (Controllers):** Gestionen les sol·licituds HTTP entrants des de la capa de presentació, criden als serveis necessaris i tornen les respostes adequades.
- **Serveis (Services):** Implementen la lògica de negoci de l'aplicació. Aquí es processa la informació abans de ser emmagatzemada o després de ser recuperada de la base de dades.
- **Repositoris (Repositories):** S'encarreguen de la comunicació amb la base de dades, utilitzant JPA (Java Persistence API) per a la interacció amb MySQL.

Capa de dades

La capa de dades és responsable de l'emmagatzematge i la gestió de les dades de l'aplicació. En aquesta arquitectura, utilitzem MySQL com a sistema de gestió de bases de dades relacional.

Els components són:

- **Base de dades MySQL:** Emmagatzema les dades de l'aplicació a les taules corresponents.
- **Firebase Storage:** Utilitzat per emmagatzemar i recuperar imatges. Firebase Storage ofereix una solució escalable i segura per a la gestió d'arxius.
- **Stripe:** Gestió de dades de pagaments. Stripe proporciona una infraestructura per gestionar pagaments en línia de manera segura i fiable.

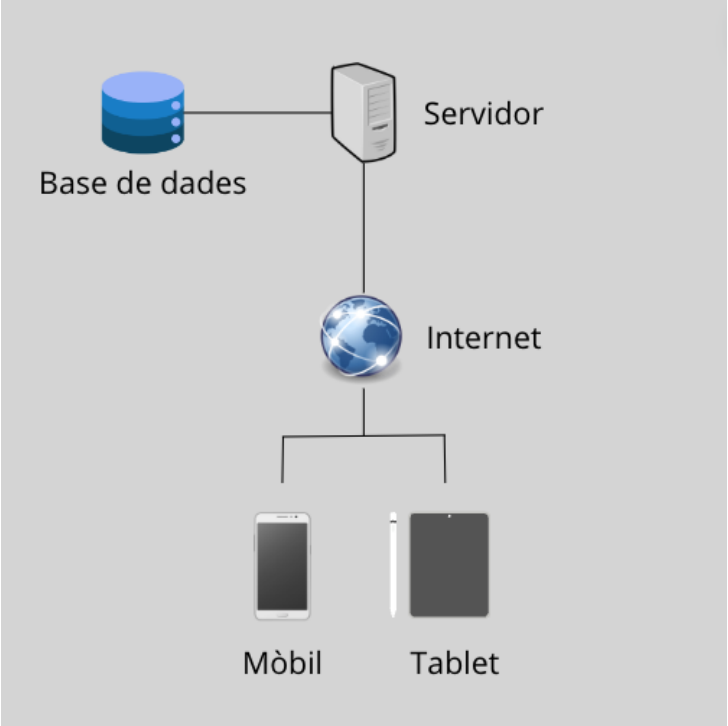

Figura 5.1: Diagrama d'arquitectura física. (Font pròpia)

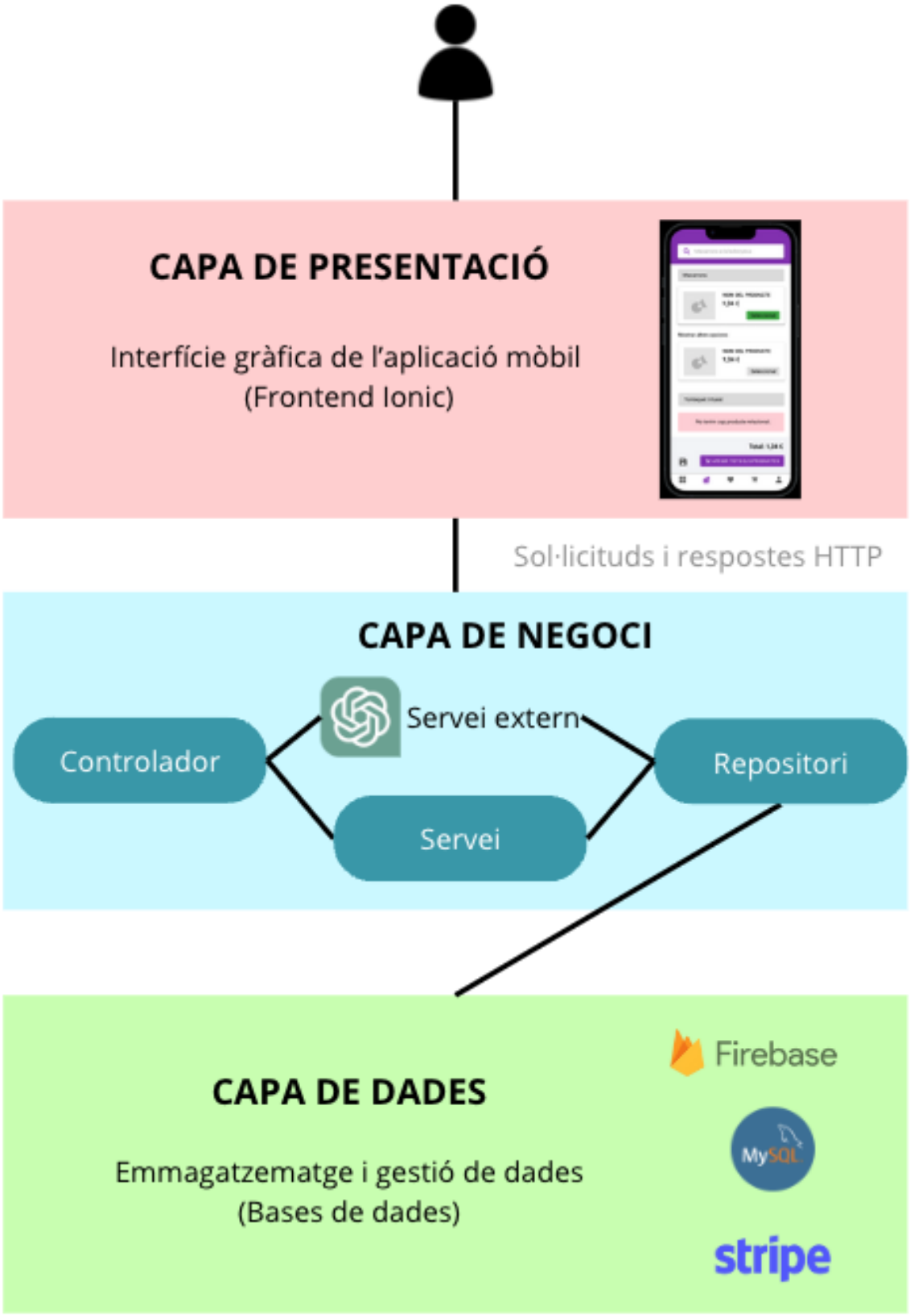

Figura 5.2: Diagrama d'arquitectura lògica. (Font pròpia)

L'elecció d'una arquitectura de 3 capes es basa en diversos avantatges clau:

- **Modularitat:** La separació en capes facilita el desenvolupament i manteniment del codi, permetent que cada capa es desenvolupi i s'actualitzi de manera independent.
- **Escalabilitat:** Cada capa pot escalar-se de manera independent segons les necessitats, millorant la capacitat de resposta i la disponibilitat del sistema.
- **Reutilització:** La lògica de negoci i els components de dades poden reutilitzar-se fàcilment en diferents aplicacions o interfícies de presentació.
- **Seguretat:** La separació de capes millora la seguretat del sistema, ja que cada capa pot implementar mesures de seguretat específiques i limitar l'accés als components subjacents.

Aquest model d'arquitectura proporciona una base sòlida i flexible per al desenvolupament de l'aplicació, assegurant una experiència d'usuari òptima i una gestió eficient dels recursos del sistema.

5.2 Exemples de diagrames de seqüència

5.2.1 Exemple 1: Diagrama de seqüència de la crida /api/payment/confirm/id

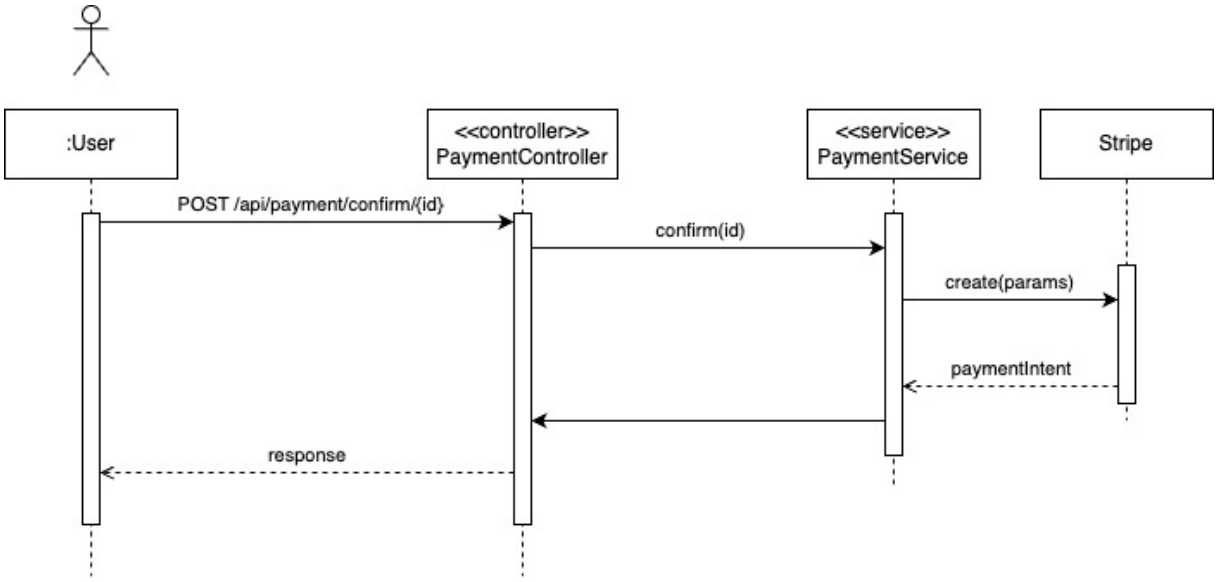

Figura 5.3: Diagrama de seqüència - Crida /api/payment/confirm/id. (Font pròpia)

La interacció s'inicia quan l'usuari envia una sol·licitud POST a l'endpoint /api/payment/confirm/id, on id és un identificador únic que correspon a un intent de pagament preexistent. Aquesta sol·licitud és rebuda pel *PaymentController*, que actua com a controlador en aquesta arquitectura. El *PaymentController*, després de rebre la sol·licitud, invoca al *PaymentService*, passant-li l'identificador del pagament (id) que necessita ser confirmat. A continuació, el servei interactua amb Stripe, una plataforma de pagaments externa, cridant la funció *create(params)*. Aquesta funció intenta confirmar un *PaymentIntent* basat en els paràmetres que han estat especificats, els quals inclouen detalls com l'import del pagament, la moneda, i altres metadades necessàries per completar la transacció. Un cop el *PaymentIntent* és modificat exitosament, *Stripe* retorna aquest objecte al servei. En rebre aquesta informació, el *PaymentService* processa el resultat i finalment retorna una resposta al *PaymentController*. Finalment, el controlador envia una resposta a l'usuari amb l'estat de la resposta i el *PaymentIntent* en format *JSON*.

5.2.2 Exemple 2: Diagrama de seqüència de la crida /api/ai/-recipe a la API

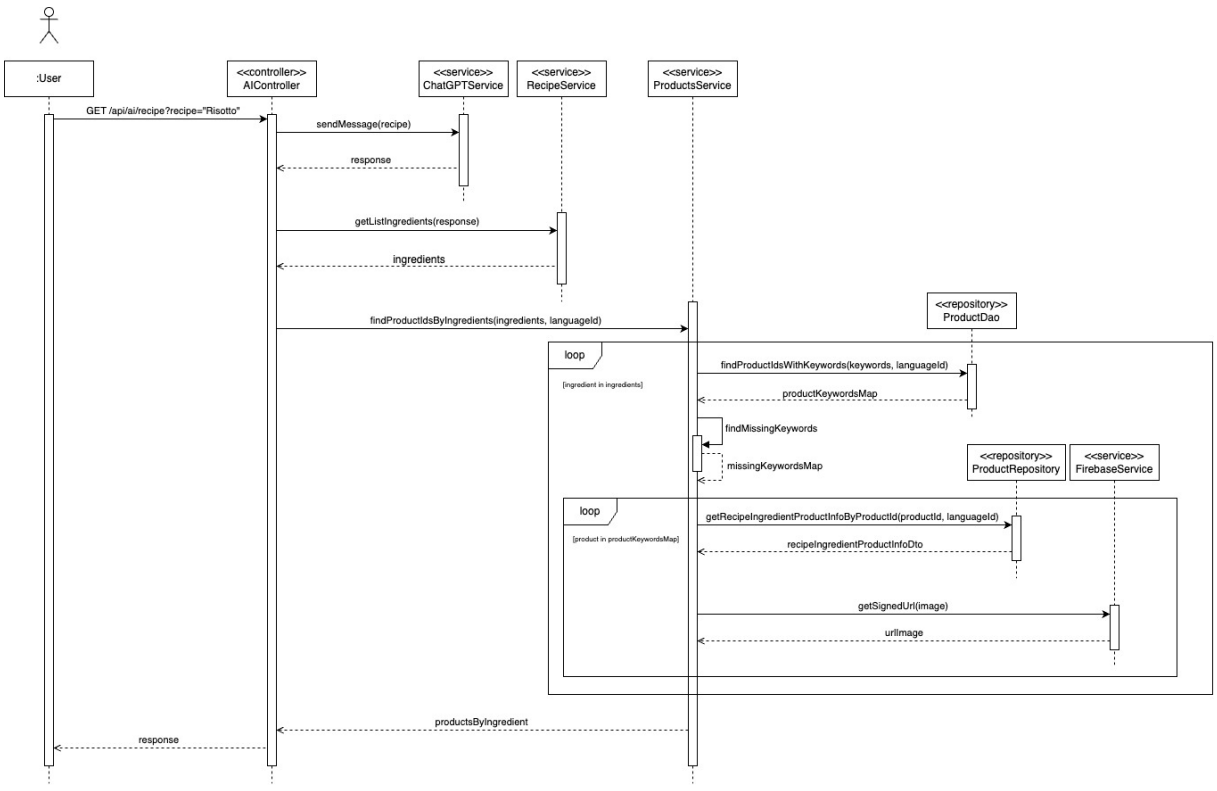

Figura 5.4: Diagrama de seqüència - Crida /api/ai/recipe. (Font pròpia)

El diagrama de seqüència de la imatge 5.4 mostra la interacció entre diferents components del nostre sistema de software dissenyat per processar i respondre a diferents sol·licituds mitjançant una API.

La interacció comença quan l'usuari envia una sol·licitud *GET* per obtenir els ingredients i els seus productes relacionats a partir del nom d'una recepta. En aquest cas, la informació enviada és "Risottoï s'envia a través d'un *query param*. El controlador rep la sol·licitud i procedeix a invocar al *ChatGPTService* passant-li el nom de la recepta sol·licitada. Aquest servei genera una resposta que conté els ingredients necessaris per cuinar-la. Amb les dades obtingudes per aquest servei d'intel·ligència artificial, es processa la resposta al nostre *RecipeService* per extraure la llista d'ingredients amb el format necessari.

A partir d'aquí, el *ProductSevice* s'encarrega d'identificar els productes disponibles que corresponen a cada ingredient. Aquesta búsqueda implica diferents subtasques que es veuen descrites al diagrama. Finalment, un cop obtinguda tota aquesta informació, el *ProductService* l'envia de volta al *RecipeService* i, per acabar, al usuari a través del controlador.

5.3 Patrons de disseny

5.3.1 Model-View-ViewModel (MVVM)

El patró Model-View-ViewModel (MVVM) [36] separa la lògica de negoci de la interfície de l'usuari, facilitant el manteniment i l'escalabilitat del projecte. Aquest patró és s'ha utilitzat per al desenvolupament de la part del front, on es fa servir Ionic i Angular.

- **Model:** Representa el model de dades i la lògica empresarial de l'aplicació. Angular i Ionic poden utilitzar una interfície per definir un model, ja que permeten definir l'estructura d'un objecte i assegurar que segueixi un cert format.
- **ViewModel:** Serveix com intermediari entre el *Model* i la *View*. Conté tota la lògica de presentació i és responsable de gestionar les dades que es mostren a la interfície de l'usuari i de processar les accions de l'usuari.
- **View:** Defineix com la informació i les funcionalitats es mostraran a l'usuari gràficament. Aquest component es connecta amb el ViewModel per obtenir dades i enviar-ne.

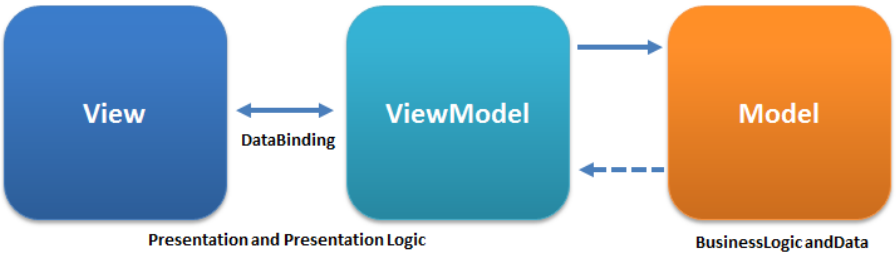

Figura 5.5: Patró Model-View-ViewModel. (Wikipedia)

5.3.2 Model-View-Controller (MVC)

El patró Model-View-Controller (MVC) [37] és un patró de disseny arquitectònic que separa una aplicació en tres components interconnectats. El desenvolupament del backend de l'aplicació s'ha implementat amb Spring Boot, que facilita la implementació d'aquest patró MVC.

- **Model:** Representa les dades i la lògica de negoci de l'aplicació. És l'encarregat de gestionar les dades i les normes que les regeixen. Spring Data JPA i Hibernate s'utilitzen habitualment per a la persistència de dades. (`@Entity i @Repository`)
- **View:** Representa la capa de presentació i s'encarrega de mostrar les dades a l'usuari. Gestiona els elements de la interfície de l'usuari i la representació. (Projecte frontend Ionic)
- **Controller:** Actua com intermediari entre el model i la vista. Rep l'entrada de l'usuari, la processa i gestiona el flux de dades entre el model i la vista. Els controladors Spring Boot gestionen les sol·licituds dels usuaris, gestionen la interacció entre el model i la vista i controlen el flux de l'aplicació. (`@Controller`)

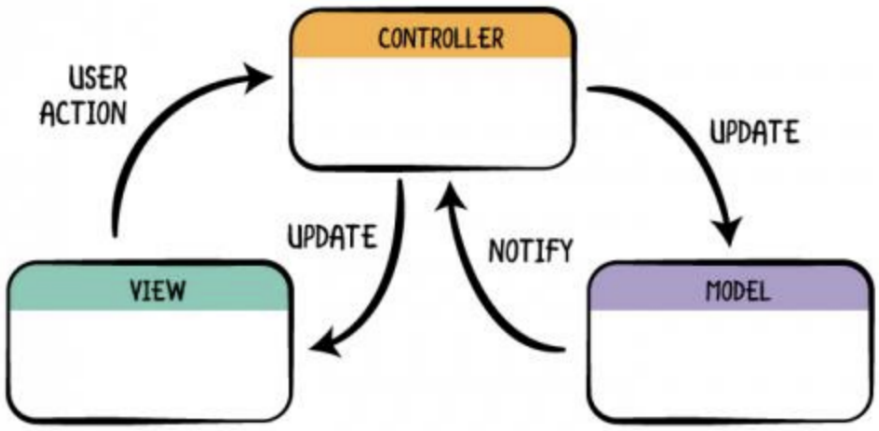

Figura 5.6: Patró Model-View-Controller. (LinkedIn [38])

5.4 Bases de dades

El disseny de la base de dades implica definir com s'emmagatzemarà i com s'accedirà a les dades dins del sistema. Aquest disseny a centra en optimitzar el rendiment de la base de dades i assegurar l'eficiència en l'accés a les dades.

Per desenvolupar aquest projecte, s'ha optat per una base de dades relacional, ja que aquest tipus suporta transaccions ACID (Atomicitat, Consistència, Aïllament i Durabilitat), oferint una gestió de transaccions robusta i de confiança. A més a més, utilitzen llenguatge SQL (Structured Query language) per al tractament i consulta de les dades, el qual permet realitzar consultes complexes que involucren diferents taules.

5.4.1 Gestió de l'esquema de la base de dades

En aquest projecte, la gestió de les taules i els canvis en l'esquema de la base de dades s'han realitzat utilitzant *Liquibase*. Liquibase és una eina de gestió de canvis de bases de dades de codi obert que permet tenir un control de versions dels canvis realitzats a l'esquema de la base de dades. Així es garanteix que tots els canvis s'apliquin de manera controlada i reversible, proporcionant una base sòlida per a l'escalabilitat i el manteniment.

CREACIÓ DE LES TAULES

```
CREATE TABLE category (
    category_id INT AUTO_INCREMENT PRIMARY KEY,
    order INT DEFAULT 0 NOT NULL,
    enabled TINYINT(1) DEFAULT 1 NOT NULL
);
```

Listing 5.1: Creació de la taula category

- **category_id:** És de tipus INT AUTO_INCREMENT. Aquest camp s'utilitza per assignar automàticament un valor únic a cada categoria registrada a la base de dades, eliminant la necessitat d'assignació manual i assegurant que cada categoria pugui ser identificada de manera única. Això és fonamental per a l'índex i accés ràpid a les dades.
- **order:** És de tipus INT amb un valor per defecte de 0. Aquest camp s'utilitza per determinar l'ordre de visualització de les categories. Un número més baix significa que la categoria apareixerà abans que altres amb números més alts. Això permet als administradors del sistema controlar com es presenten les categories als usuaris finals de manera dinàmica.
- **enabled:** És de tipus TINYINT(1) amb un valor per defecte de 1, el que indica que la categoria està habilitada per defecte. Aquest camp determina si la categoria ha de ser mostrada o ocultada en la interfície d'usuari. Un valor de 1 significa que la categoria és visible i accessible, mentre que un valor de 0 indicaria que està desactivada.

```
CREATE TABLE product (
    product_id INT AUTO_INCREMENT PRIMARY KEY,
    price DECIMAL(10,2) NOT NULL,
    stock INT NOT NULL,
    brand VARCHAR(100),
    image VARCHAR(255),
    category_id INT NOT NULL,
    type INT DEFAULT 1,
    primary TINYINT(1) DEFAULT 0 NOT NULL,
    FOREIGN KEY (category_id) REFERENCES category(category_id),
    FOREIGN KEY (type) REFERENCES product_type(type_id)
);
```

Listing 5.2: Creació de la taula product

- **product_id:** És de tipus INT AUTO_INCREMENT. Aquest camp s'utilitza per assignar automàticament un valor únic a cada producte registrat a la base de dades, eliminant la necessitat d'assignació manual i assegurant que cada producte pugui ser identificat de manera única. Això és fonamental per a l'índex i accés ràpid a les dades.
- **price:** És de tipus DECIMAL(10,2), el que significa que pot emmagatzemar números amb fins a deu dígits en total, incloent dos decimals. Això és crucial per assegurar que els preus es tractin amb precisió.
- **stock:** És de tipus INT i s'utilitza per registrar la quantitat disponible de cada producte.
- **brand:** És de tipus VARCHAR(100). Aquest camp opcional permet emmagatzemar el nom de la marca del producte fins a un màxim de 100 caràcters, facilitant la identificació i classificació de productes segons el seu fabricant o marca comercial.
- **image:** És de tipus VARCHAR(255). Aquest camp opcional s'utilitza per emmagatzemar el nom de la imatge.
- **category_id:** És de tipus INT i és obligatori (NOT NULL). Actua com una clau forana que referència a la taula category, assegurant que cada producte estigui vinculat a una categoria existent. Això facilita l'organització i cerca de productes per categories.
- **type:** És de tipus INT amb un valor per defecte de 1. Aquest camp també és una clau forana que referència a la taula *product_type*. Permet categoritzar productes en diferents tipus segons els valors de les dades del atributs nutricional.
- **primary:** És de tipus TINYINT(1) amb un valor predeterminat de 0 i és obligatori (NOT NULL). Aquest camp indica si el producte és el primari o no, la qual cosa ens serà útil a l'hora de buscar els productes per a fer una recepta.

```sql
CREATE TABLE language (
    language_id VARCHAR(2) PRIMARY KEY,
    description VARCHAR(50),
    enabled TINYINT(1) DEFAULT 0
);
```

Listing 5.3: Creació de la taula language

- **language_id:** És de tipus VARCHAR(2) i s'utilitza com a clau primària de la taula. Aquest identificador està dissenyat per emmagatzemar codis d'idioma ISO de dos lletres, com 'en' per anglès, 'es' per espanyol, etc. La decisió d'utilitzar un VARCHAR(2) per aquest camp permet un emmagatzematge eficient dels codis d'idioma, essencial per a operacions ràpides de cerca i filtratge basades en l'idioma.
- **description:** És de tipus VARCHAR(50) i proporciona una descripció breu o el nom complet de l'idioma representat. Aquest camp, limitat a 50 caràcters, permet una descripció suficient per reconèixer l'idioma sense consumir massa espai en la base de dades.
- **enabled:** És de tipus TINYINT(1) amb un valor per defecte de 0. Aquest camp determina si l'idioma està habilitat o no dins de la plataforma o l'aplicació. Un valor de 1 indica que l'idioma està activat, mentre que un 0 indica que està desactivat. Aquesta configuració permet als administradors gestionar quins idiomes són accessibles per als usuaris de manera dinàmica.

```sql
CREATE TABLE category_language (
    category_id INT,
    language_id VARCHAR(2),
    name VARCHAR(50) NOT NULL,
    PRIMARY KEY (category_id, language_id),
    FOREIGN KEY (category_id) REFERENCES category(category_id),
    FOREIGN KEY (language_id) REFERENCES language(language_id)
);
```

Listing 5.4: Creació de la taula category_language

Descripció dels atributs

- **category_id:** És una part de la clau primària i és una clau forana que referència a la taula *category*. Aquesta connexió assegura que cada idioma associat a una categoria estigui vinculat a una categoria existent en la base de dades.

- **language_id:** És de tipus VARCHAR(2) i està dissenyat per emmagatzemar codis d'idioma ISO de dos lletres. És l'altra part de la clau primària i també una clau forana que referència a la taula *language*, assegurant que els noms de les categories s'associïn amb un idioma suportat per la base de dades.
- **name:** És de tipus VARCHAR(50) i conté el nom de la categoria en l'idioma especificat. El camp és obligatori (NOT NULL), reflectint la necessitat que cada entrada de categoria tingui un nom associat en cada idioma. La longitud limitada a 50 caràcters permet una gestió eficient dels noms de categories dins de la base de dades.

```
CREATE TABLE product_type (
    type_id INT AUTO_INCREMENT PRIMARY KEY,
    description VARCHAR(200)
);
```

Listing 5.5: Creació de la taula product_type

Descripció dels atributs

- **type_id:** És de tipus INT i utilitza la propietat AUTO_INCREMENT per generar automàticament un identificador únic per a cada tipus de producte introduït a la taula. Com a clau primària, assegura que cada entrada a la taula sigui única i fàcilment referenciable.
- **description:** És de tipus VARCHAR(200) i proporciona un espai per a una descripció textual del tipus de producte. Aquesta descripció pot incloure detalls sobre les característiques que defineixen el grup de productes dins d'aquest tipus. Està pensat per separar segons la mesura que s'utilitza a l'hora de contrastar els atributs nutricionals.

```
CREATE TABLE product_language (
    product_id INT,
    language_id VARCHAR(2),
    name VARCHAR(50),
    description VARCHAR(1500),
    ingredients VARCHAR(10000),
    PRIMARY KEY (product_id, language_id),
    FOREIGN KEY (product_id) REFERENCES product(product_id),
    FOREIGN KEY (language_id) REFERENCES language(language_id)
);
```

Listing 5.6: Creació de la taula product_language

Descripció dels atributs

- **product_id:** És una part de la clau primària i és una clau forana que referència a la taula *product*. Aquesta connexió assegura que cada idioma associat a un producte estigui lligat a un producte existent en la base de dades.
- **language_id:** És de tipus VARCHAR(2) i està dissenyat per emmagatzemar codis d'idioma ISO de dos lletres. És l'altra part de la clau primària i també una clau forana que referència a la taula *language*, assegurant que les descripcions dels productes s'associïn amb un idioma suportat per la base de dades.
- **name:** És de tipus VARCHAR(50) i conté el nom del producte en l'idioma especificat. La longitud limitada a 50 caràcters permet emmagatzemar noms de producte de manera eficient sense consumir massa espai en la base de dades.
- **description:** És de tipus VARCHAR(1500) i proporciona una descripció detallada del producte en l'idioma especificat. La longitud de 1500 caràcters permet incloure informació suficient per explicar el producte sense ser excessivament llarg.
- **ingredients:** És de tipus VARCHAR(10000) i està destinat a llistar els ingredients del producte, si és aplicable. La longitud extensa de 10000 caràcters es dissenya per acomodar llistes d'ingredients complexes i extenses per a productes alimentaris.

```sql
CREATE TABLE nutritional_attributes (
    attribute_id VARCHAR(100) PRIMARY KEY,
    description VARCHAR(200),
    order INT DEFAULT 1 NOT NULL,
    progress TINYINT(1) DEFAULT 1 NOT NULL
);
```

Listing 5.7: Creació de la taula nutritional_attributes

Descripció dels atributs

- **attribute_id:** És la clau primària de la taula i està dissenyada per emmagatzemar identificadors únics per a cada atribut nutricional. És de tipus VARCHAR(100), el que permet una flexibilitat considerable en la codificació d'identificadors, incloent números, lletres i símbols. El format esperat és, per exemple, dietary_fiber.
- **description:** És de tipus VARCHAR(200) i proporciona una descripció del atribut nutricional. Aquest camp ofereix suficient espai per explicar detalladament l'atribut.

- **order:** És de tipus INT i té un valor per defecte d'1. Aquest camp s'utilitza per determinar l'ordre en què els atributs nutricionals haurien de ser mostrats o processats. És obligatori (NOT NULL), assegurant que sempre hi haurà un valor per a cada entrada.
- **progress:** És de tipus TINYINT(1) i també té un valor per defecte d'1. Aquest camp indica si la barra de progrés mostrada a la pantalla del detall del producte està activa (1) o no (0). És també un camp obligatori (NOT NULL), reflectint la necessitat de mantenir un registre actiu de l'estat de cada atribut.

```
CREATE TABLE nutritional_attributes_language (
    attribute_id VARCHAR(100),
    language_id VARCHAR(2),
    name VARCHAR(100) NOT NULL,
    PRIMARY KEY (attribute_id, language_id),
    FOREIGN KEY (attribute_id) REFERENCES nutritional_attributes(
   attribute_id),
    FOREIGN KEY (language_id) REFERENCES language(language_id)
);
```

Listing 5.8: Creació de la taula nutritional_attributes_language

Descripció dels atributs

- **attribute_id:** És de tipus VARCHAR(100) i serveix com a clau forana que referència l'identificador únic dels atributs nutricionals a la taula *nutritional_attributes*. Aquesta connexió assegura que cada traducció està vinculada a un atribut nutricional específic.
- **language_id:** És de tipus VARCHAR(2), dissenyat per emmagatzemar codis d'idioma ISO de dos lletres. És l'altra part de la clau primària i també una clau forana que referència a la taula *language*, garantint que cada entrada de nom d'atribut nutricional està associada a un idioma específic.
- **name:** És de tipus VARCHAR(100) i és un camp obligatori (NOT NULL). Aquest camp conté el nom de l'atribut nutricional en l'idioma específic indicat per *language_id*. La longitud de 100 caràcters permet una flexibilitat suficient per a descripcions completes i clares.

```sql
CREATE TABLE product_attribute_value (
    product_id INT,
    attribute_id VARCHAR(100),
    value VARCHAR(20) NOT NULL,
    PRIMARY KEY (product_id, attribute_id),
    FOREIGN KEY (product_id) REFERENCES product(product_id),
    FOREIGN KEY (attribute_id) REFERENCES nutritional_attributes(
   attribute_id)
);
```

Listing 5.9: Creació de la taula product_attribute_value

Descripció dels atributs

- **product_id:** És de tipus INT i actua com a clau forana que referència l'identificador únic dels productes a la taula *product*. Aquesta connexió assegura que cada valor d'atribut estigui associat a un producte específic.
- **attribute_id:** És de tipus VARCHAR(100) i serveix com a clau forana que referència l'identificador únic dels atributs nutricionals a la taula *nutritional_attributes*. Aquesta configuració permet vincular valors específics amb els atributs corresponents, assegurant la integritat de les dades nutricionals associades amb cada producte. També forma part de la clau primària.
- **value:** És de tipus VARCHAR(20) i és un camp obligatori (NOT NULL). Aquest camp conté el valor associat amb l'atribut nutricional per un producte donat. La limitació a 20 caràcters permet emmagatzemar una varietat de valors nutricionals amb la seva mesura corresponent (ex: 1672 kJ / 400kcal)

```sql
CREATE TABLE current_cart (
    cart_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT UNIQUE,
    FOREIGN KEY (user_id) REFERENCES user(user_id)
);
```

Listing 5.10: Creació de la taula current_cart

Descripció dels atributs

- **cart_id:** És de tipus INT i utilitza la propietat AUTO_INCREMENT per generar automàticament un identificador únic per a cada cistella de la compra. Aquest és la clau primària de la taula, garantint que cada entrada sigui única i facilment referenciable.

- **user_id:** És de tipus INT i està designat com a UNIQUE, això significa que cada usuari pot estar associat a una sola cistella de la compra en qualsevol moment. Aquest camp actua també com una clau forana que referència l'identificador únic dels usuaris en la taula *user* (actualment aquesta taula s'ha creat unicament amb l'identificador), establint una relació directa entre l'usuari i la seva cistella de la compra. Aquesta configuració assegura que no hi hagi duplicats i facilita la gestió de la integritat de les dades.

```
CREATE TABLE cart_product (
    cart_id INT,
    product_id INT,
    amount INT,
    PRIMARY KEY (cart_id, product_id),
    FOREIGN KEY (cart_id) REFERENCES current_cart(cart_id),
    FOREIGN KEY (product_id) REFERENCES product(product_id)
);
```

Listing 5.11: Creació de la taula cart_product

Descripció dels atributs

- **cart_id:** És de tipus INT i actua com a part de la clau primària juntament amb *product_id*. Aquest camp també és una clau forana que referència l'identificador *cart_id* de la taula *current_cart*, vinculant cada producte especificat en aquesta taula a una cistella específica.
- **product_id:** És de tipus INT i forma la segona part de la clau primària. Aquest camp també és una clau forana que s'enllaça amb el *product_id* de la taula *product*.
- **amount:** És de tipus INT i indica la quantitat de cada producte present en la cistella de la compra de cada usuari.

```
CREATE TABLE order (
    order_id INT PRIMARY KEY,
    user_id INT NOT NULL,
    state VARCHAR(50) NOT NULL,
    total_amount DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES user(user_id)
);
```

Listing 5.12: Creació de la taula order

Descripció dels atributs

- **order_id:** És de tipus INT AUTO_INCREMENT i serveix com la clau primària de la taula. Aquest camp identifica de manera única cada comanda feta per un usuari, facilitant el seguiment i la gestió de les comandes individualment dins del sistema.
- **user_id:** És de tipus INT i és un camp obligatori (NOT NULL). Actua com a clau forana que enllaça cada comanda a un usuari específic de la taula *user*. Aquesta configuració assegura que cada comanda està associada directament amb un usuari registrat, permetent al sistema seguir quines comandes pertanyen a quins usuaris.
- **state:** És de tipus VARCHAR(50) i també és un camp obligatori (NOT NULL). Aquest camp descriu l'estat actual de la comanda (per exemple, "paid", "send"...). La claredat en aquest camp ajuda a gestionar el flux de processament de les comandes dins de l'empresa.
- **total_amount:** És de tipus DECIMAL(10,2) i es requereix que sigui no nul. Representa el cost total de la comanda en termes monetaris.
- **date:** És de tipus DATE i és un camp obligatori (NOT NULL). Aquesta columna registra la data en què es realitza o processa cada comanda. L'ús del tipus DATE assegura que la data sigui emmagatzemada en un format consistent i reconegut per la base de dades.

```
CREATE TABLE order_product (
    order_id INT,
    product_id INT,
    price DECIMAL(10,2) NOT NULL,
    name VARCHAR(50) NOT NULL,
    image VARCHAR(255),
    amount INT NOT NULL,
    PRIMARY KEY (order_id, product_id),
    FOREIGN KEY (order_id) REFERENCES `order`(order_id),
    FOREIGN KEY (product_id) REFERENCES product(product_id)
);
```

Listing 5.13: Creació de la taula order_product

Descripció dels atributs

- **order_id:** És de tipus INT i forma part de la clau primària junt amb *product_id*. Aquest camp actua com una clau forana que enllaça cada entrada de producte amb una comanda específica a la taula *order*, garantint que cada producte listat està associat amb una comanda existent.

- **product_id:** És de tipus INT i és l'altra component de la clau primària. També és una clau forana que referència el *product_id* de la taula *product*, vinculant el producte específic amb una entrada a aquesta taula.
- **price:** És de tipus DECIMAL(10,2) i és un camp obligatori (NOT NULL). Aquest camp indica el preu del producte al moment de la comanda, assegurant que queden registrats els preus exactes en el moment de cada transacció.
- **name:** És de tipus VARCHAR(50) i és un camp obligatori (NOT NULL). Conté el nom del producte tal com apareix en la comanda.
- **image:** És de tipus VARCHAR(255) i és opcional. Aquest camp pot contenir el nom de la imatge del producte.
- **amount:** És de tipus INT i és un camp obligatori (NOT NULL). Representa la quantitat del producte, essencial per a la gestió d'inventari i logística de entrega.

```
CREATE TABLE favorite_products (
      user_id INT NOT NULL,
      product_id INT NOT NULL,
      PRIMARY KEY (user_id, product_id),
      FOREIGN KEY (product_id) REFERENCES product(product_id),
      FOREIGN KEY (user_id) REFERENCES user(user_id)
);
```

Listing 5.14: Creació de la taula favorite_products

Descripció dels atributs

- **user_id:** És de tipus INT i és un camp obligatori (NOT NULL). Aquest camp és una clau forana que referència el *user_id* de la taula *user*, assegurant que cada producte favorit està associat amb un usuari existent.
- **product_id:** És de tipus INT i és un camp obligatori (NOT NULL). També és una clau forana, en aquest cas referència el *product_id* de la taula *product*, establint la relació entre l'usuari i els productes favorits.

```sql
CREATE TABLE favorite_recipes (
    recipe_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    name VARCHAR(255) NOT NULL,
    ingredients VARCHAR(10000) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES user(user_id),
    UNIQUE (user_id, name)
);
```

Listing 5.15: Creació de la taula favorite_recipes

Descripció dels atributs

- **recipe_id:** És de tipus INT AUTO_INCREMENT i serveix com a clau primària de la taula. Aquest camp identifica de manera única cada recepta favorita afegida per un usuari.
- **user_id:** És de tipus INT i és un camp obligatori (NOT NULL). Actua com una clau forana que enllaça cada recepta amb un usuari específic de la taula *user*, assegurant que cada recepta favorita està associada directament amb un usuari registrat.
- **name:** És de tipus VARCHAR(255) i és un camp obligatori (NOT NULL). Conté el nom de la recepta favorita, i aquest camp combinat amb *user_id* ha de ser únic per usuari. És a dir, cada usuari no pot tenir guardada més d'una recepta amb el mateix nom.
- **ingredients:** És de tipus VARCHAR(10000) i és un camp obligatori (NOT NULL). Conté els ingredients necessaris per a la recepta, permetent als usuaris guardar i consultar fàcilment els detalls de les seves receptes favorites.

```sql
CREATE TABLE user (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(255) NOT NULL,
    name VARCHAR(100) NOT NULL,
    surname VARCHAR(100) NOT NULL,
    birthday DATE,
    phone_number VARCHAR(20),
    password VARCHAR(255),
    language_id VARCHAR(2) DEFAULT 'es',
    FOREIGN KEY (language_id) REFERENCES language(language_id)
);
```

Listing 5.16: Creació de la taula user

Descripció dels atributs

- **user_id:** És de tipus INT AUTO_INCREMENT i serveix com la clau primària de la taula. Aquest camp identifica de manera única cada usuari registrat a la base de dades.
- **email:** És de tipus VARCHAR(255) i és un camp obligatori (NOT NULL). Aquest camp emmagatzema l'adreça de correu electrònic amb la que l'usuari s'ha registat.
- **name:** És de tipus VARCHAR(100) i és un camp obligatori (NOT NULL). Conté el nom de l'usuari.
- **surname:** És de tipus VARCHAR(100) i és un camp obligatori (NOT NULL). Conté els cognoms de l'usuari.
- **birthday:** És de tipus DATE i és opcional. Aquest camp emmagatzema la data de naixement de l'usuari per si en un futur és vol utilitzar per fer un regal o una promoció a l'usuari el dia del seu aniversari. Actualment, aquest camp s'utilitza únicament com a informació adicional al perfil de l'usuari.
- **phone_number:** És de tipus VARCHAR(20) i és opcional. Aquest camp emmagatzema el número de telèfon de l'usuari. Per aquest projecte, el número de telèfon s'utilitza únicament com a informació adicional de l'usuari però en un futur pot ser utilitzat per poder-se comunicar amb l'usuari o enviar promocions,ofertes,regals...
- **password:** És de tipus VARCHAR(255). Aquest camp emmagatzema la contrasenya de l'usuari de forma segura.
- **language_id:** És de tipus VARCHAR(2) i té un valor per defecte de 'es'. Actua com una clau forana que enllaça amb la taula *language*, indicant l'idioma preferit de l'usuari per a la interfície i comunicacions.

5.5 Disseny de la interfície

En aquest apartat, es presenten les pantalles que formen la interfície de l'aplicació creada i el seu disseny i també com es va d'una pantalla a l'altra. Com moltes imatges tenen una mida reduïda per no ocupar tant espai al document, s'adjunta a continuació un enllaç per poder accedir al Figma i un enllaç al document pdf amb tots els diagrames creats amb Draw.io.

- Clica aquí per accedir al Figma.
- Clica aquí per veure el document amb tots els diagrames.

GESTIÓ DE PRODUCTES

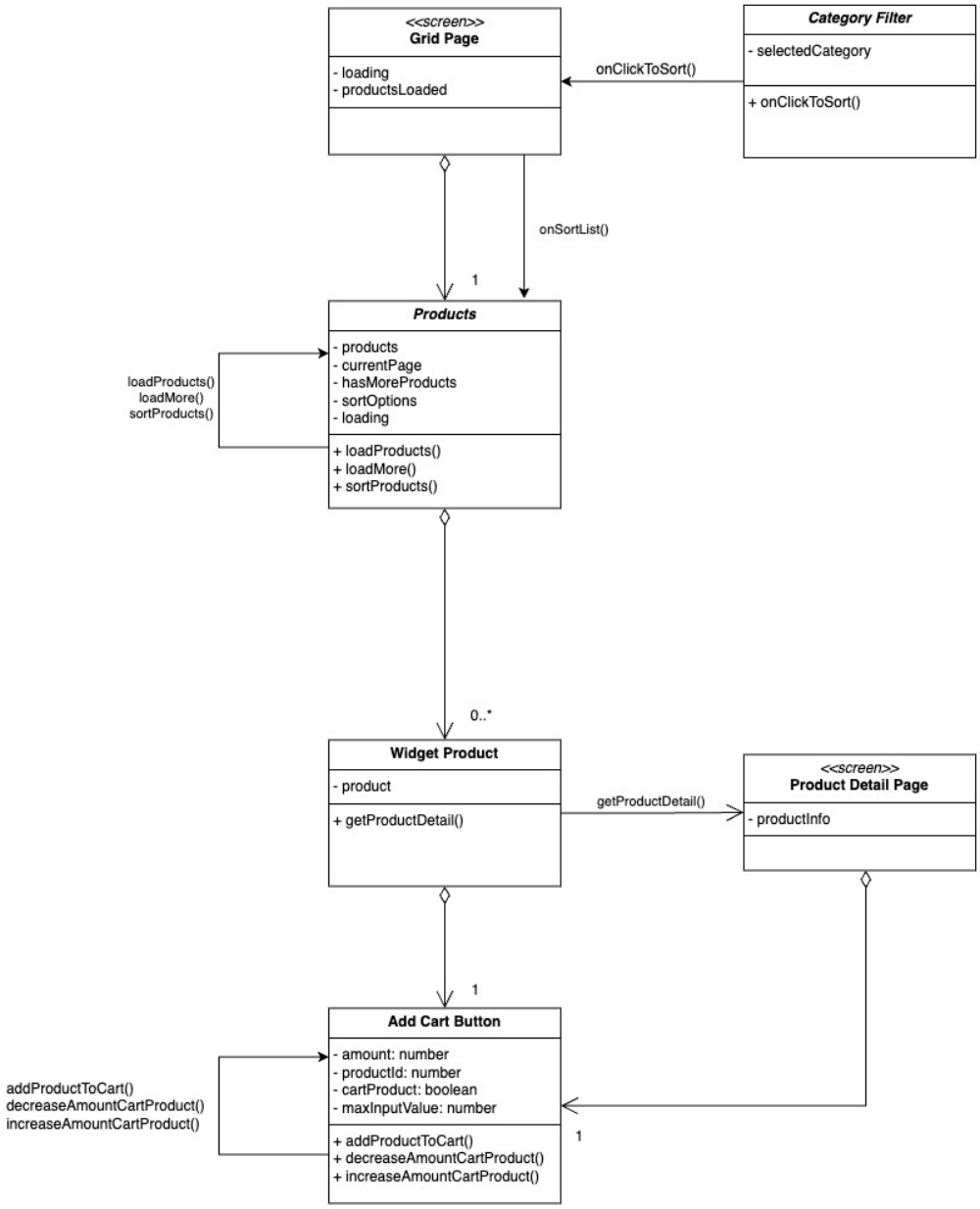

Figura 5.7: External Design de la gestió de productes. (Font pròpia)

La *Grid Page* és la pantalla principal on es mostren els productes en una disposició de graella. Aquesta pàgina té dos atributs principals, loading i productsLoaded, que indiquen l'estat de càrrega dels productes. La pàgina permet als usuaris ordenar els productes a través del mètode onClickToSort(), que es comunica amb el component *Category Filter*.

Aquest filtre permet als usuaris seleccionar un ordre específic per ordenar els productes que es mostren a la graella. El filtre té un atribut, selectedCategory, i un mètode, onClickToSort(), que activa l'ordenació dels productes. L'atribut selectedCategory canviarà en funció de la categoria filtrada. El component *Products* és la responsable de gestionar la càrrega, l'ordenació i la paginació dels productes que es mostren a la graella. Aquesta classe conté diversos atributs com products, currentPage, hasMoreProducts, sortOptions i loading, que mantenen l'estat dels productes carregats i la configuració d'ordenació. Els mètodes loadProducts(), loadMore() i sortProducts() permeten carregar els productes inicials, carregar més productes a mesura que l'usuari navega i ordenar els productes segons els criteris seleccionats. Cada producte individual és representat pel component *Widget Product*. Aquest element té un atribut, product, que conté la informació del producte, i un mètode, getProductDetail(), que redirecciona a l'usuari a la *Product Detail Page*, que té un atribut, productInfo, que conté la informació detallada del producte. Per facilitar la compra, la interfície inclou el component *Add Cart Button*. Aquest botó permet als usuaris afegir productes a la cistella de la compra, augmentar o disminuir la quantitat d'un producte al carret. Té diversos atributs, com amount, productId, cartProduct i maxInputValue, que gestionen la quantitat de producte. Els mètodes addProductToCart(), decreaseAmountCartProduct() i increaseAmountCartProduct() proporcionen la funcionalitat necessària per gestionar la cistella de la compra.

Pantalla de la grid de productes

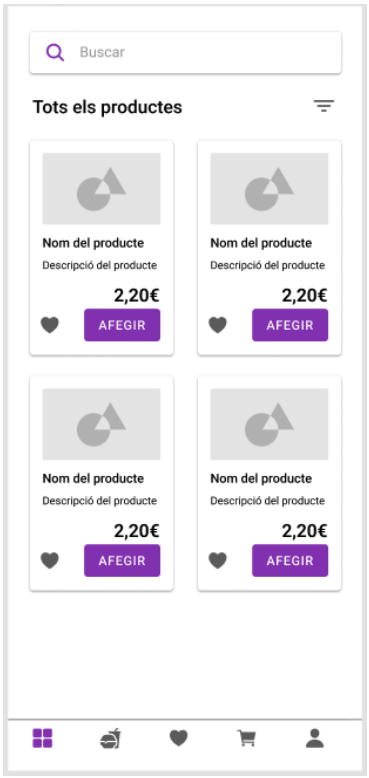

Figura 5.8: Disseny de la grid de productes. (Font pròpia)

Detall del producte

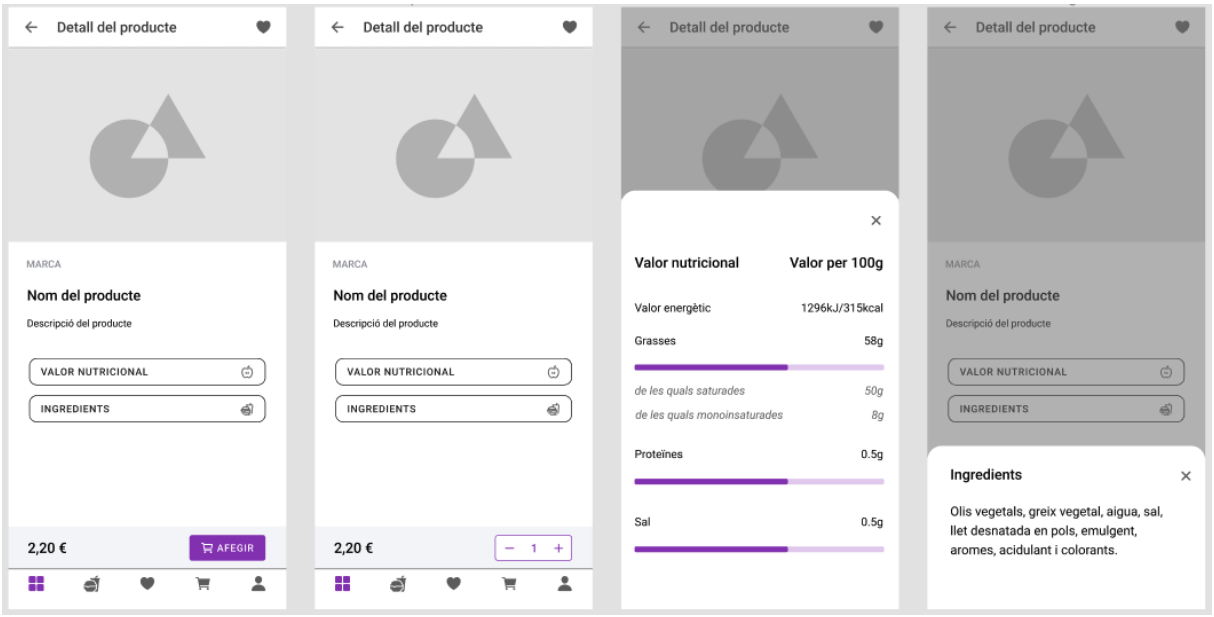

Figura 5.9: Disseny de la pantalla del detall del producte. (Font pròpia)

GESTIÓ DE RECEPTES

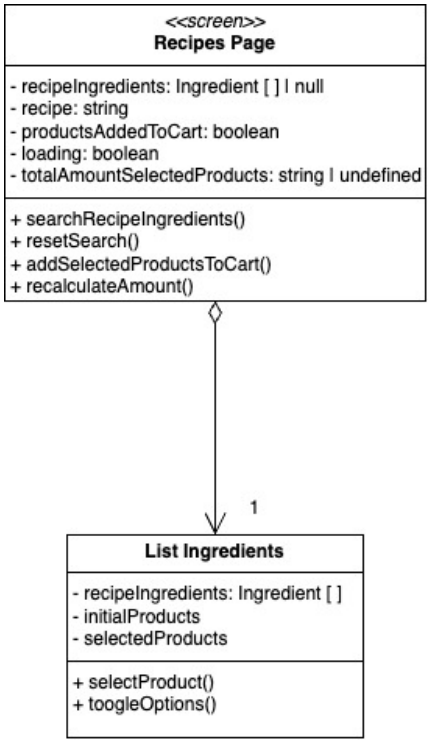

Figura 5.10: External Design de la gestió de receptes. (Font pròpia)

La *Recipes Page* és la pantalla principal on els usuaris poden interactuar amb les receptes i els seus ingredients. Aquesta pantalla té diversos atributs que gestionen l'estat de la

pàgina: recipeIngredients, que és una llista d'ingredients, recipe, que guarda el nom de la recepta, productsAddedToCart, un indicador booleà que mostra si els productes han estat afegits al carret, loading, que indica si la pàgina està carregant, i totalAmountSelectedProducts, que guarda l'import total dels productes seleccionats. Els mètodes associats a aquesta classe inclouen searchRecipeIngredients(), per buscar els ingredients de la recepta, resetSearch(), per reiniciar la cerca, addSelectedProductsToCart(), per afegir els productes seleccionats al carret, i recalculateAmount(), per recalcular l'import total dels productes seleccionats. Dins de la *Recipes Page* hi ha una relació amb el component *List Ingredients*, que es representa com una llista d'ingredients disponibles per a la recepta. Aquest component conté els atributs recipeIngredients, que és una llista d'ingredients, initialProducts, que guarda els productes inicials del llistat de recipeIngredients, i selectedProducts, que manté els productes seleccionats per l'usuari. Els mètodes associats amb aquesta classe són selectProduct(), que permet als usuaris seleccionar un producte, i toggleOptions(), que canvia les opció de veure les alternatives dels productes o no.

Pantalla de receptes

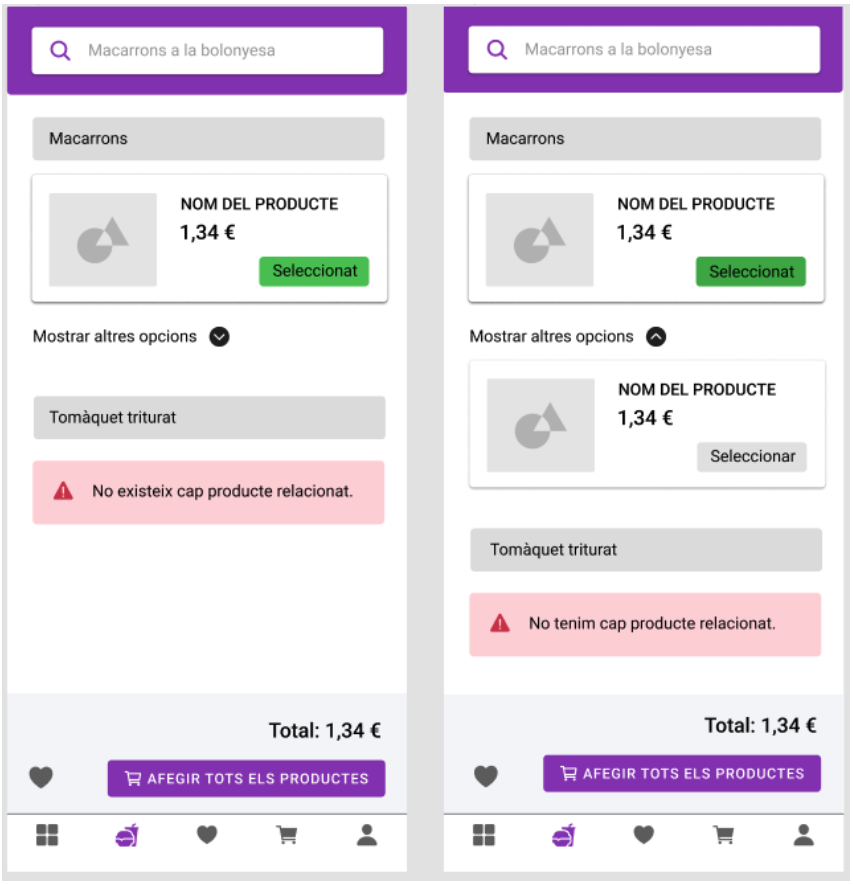

Figura 5.11: Disseny de la pantalla de receptes. (Font pròpia)

GESTIÓ DE LA CISTELLA DE LA COMPRA

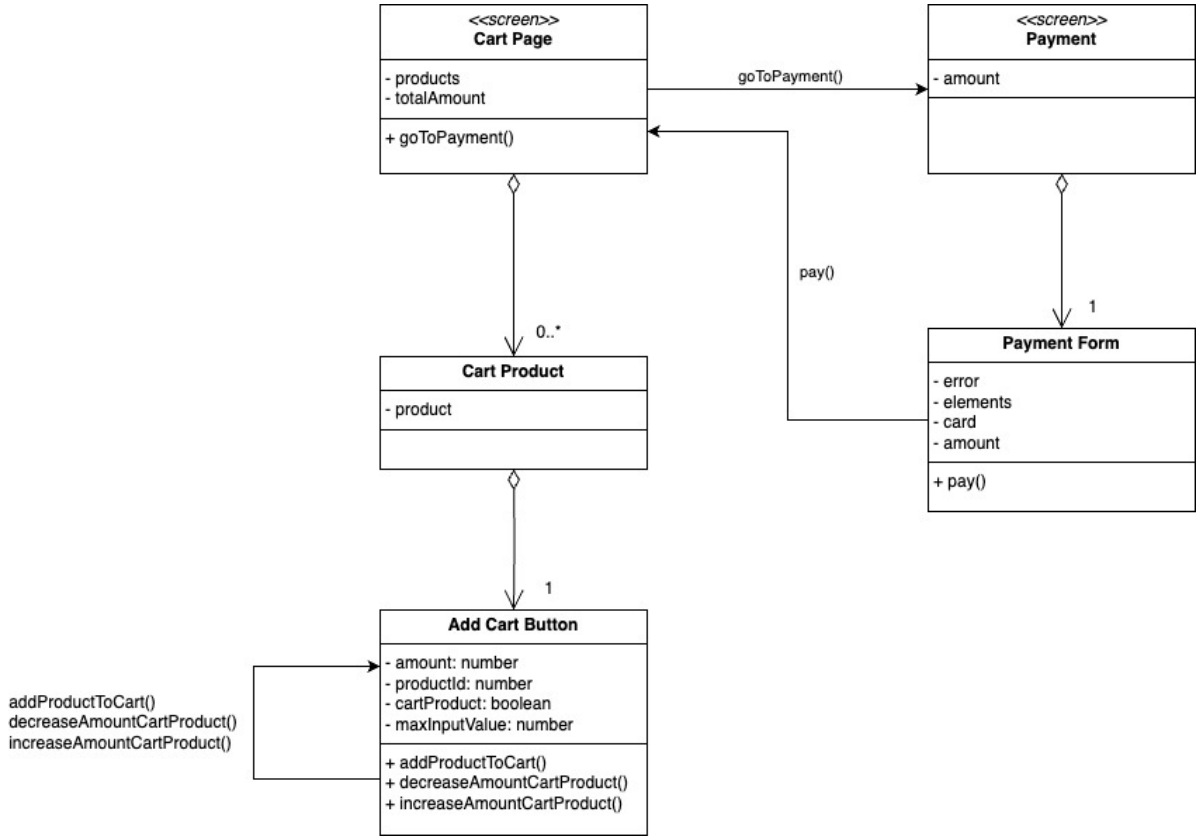

Figura 5.12: External Design de la gestió de la cistella de la compra. (Font pròpia)

La *Cart Page* és la pantalla principal on es mostren els productes afegits al carret de la compra. Aquesta pàgina conté dos atributs principals: products, que és una llista de productes de la cistella, i totalAmount, que representa l'import total dels productes afegits. La pàgina de la cistella de la compra permet als usuaris procedir al pagament mitjançant el mètode goToPayment(), que redirigeix a la pantalla de pagament. El *Cart Product* representa cada producte individual dins del llistat de la cistella. Aquest component conté l'atribut product, que emmagatzema la informació del producte. Els productes de la cistella es gestionen a través del component *Add Cart Button*, que permet als usuaris afegir productes a la cistella, així com augmentar o disminuir la quantitat de productes. Aquesta classe inclou diversos atributs: amount, que indica la quantitat del producte, productId, que identifica el producte, cartProduct, un indicador booleà que indica si prové del component *Cart Produt* o no i maxInputValue, que determina la quantitat màxima que es pot afegir. Els mètodes associats són addProductToCart(), decreaseAmountCartProduct() i increaseAmountCartProduct(). La *Payment Page* és la pantalla on els usuaris introdueixen la informació necessària per completar la compra. Aquesta pàgina té un atribut amount, que representa l'import a pagar, i inclou el mètode pay() per processar el pagament. Dins de la pàgina de pagament, es troba el component *Payment Form*, que gestiona la informació del pagament. Aquesta classe té diversos atributs: error, que emmagatzema

els errors de pagament, elements, que inclou els elements del formulari, card, que guarda la informació de la targeta, i amount, que representa l'import total a pagar. El mètode pay() s'encarrega de processar el pagament utilitzant la informació proporcionada.

Pantalla de la cistella de la compra

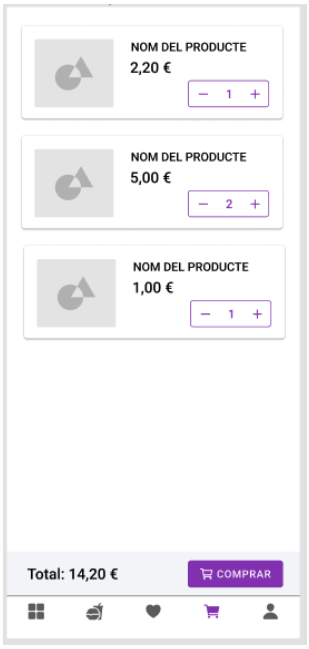

Figura 5.13: Disseny de la pantalla de la cistella de la compra. (Font pròpia)

Pagament

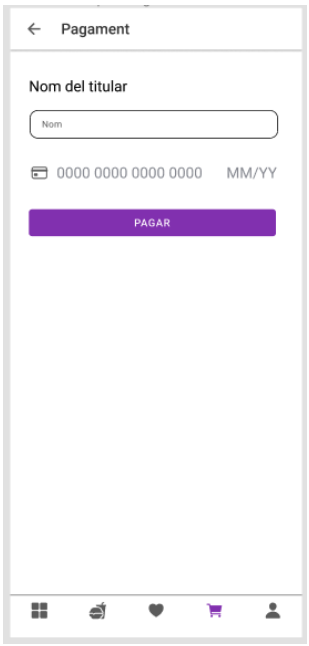

Figura 5.14: Disseny de la pantalla de pagament. (Font pròpia)

GESTIÓ DE FAVORITS

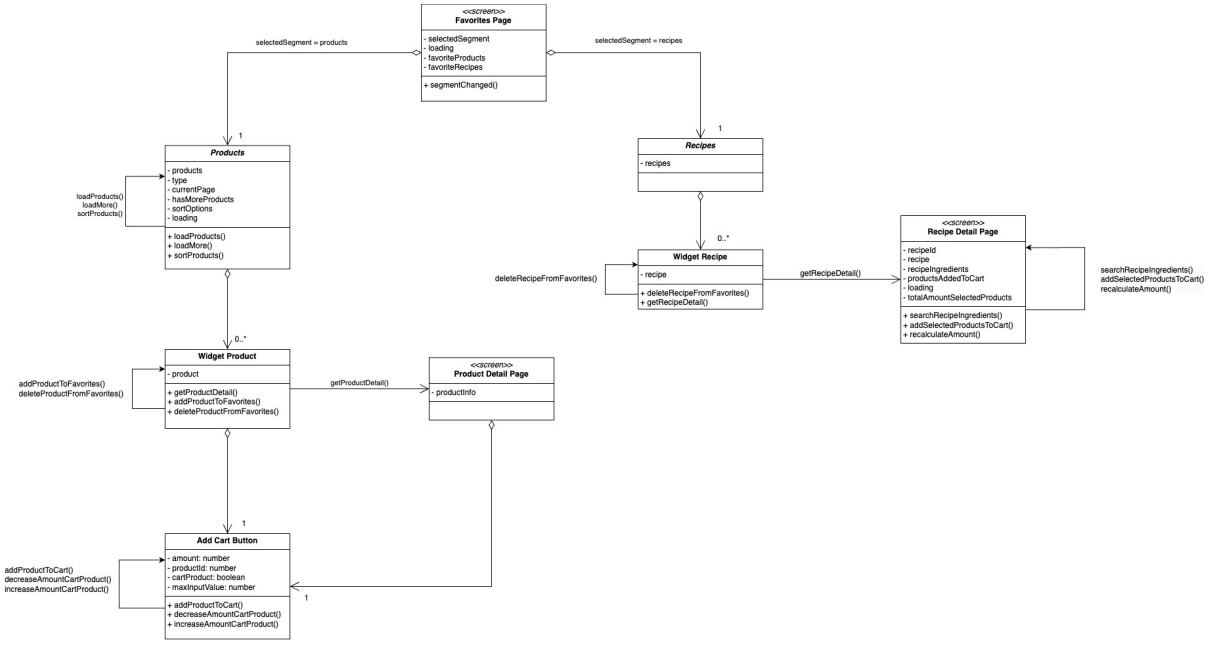

Figura 5.15: External Design de la gestió de favorits. (Font pròpia)

Pantalla de favorits

La *Favorites Page* permet als usuaris veure els productes i les receptes que tenen destacades a favorits. La pantalla permet navegar entre productes i receptes a través de dos segments. Si l'atribut selectedSegment és "products", llavors es mostra el llistat de favoriteProducts mentre que amb l'atribut selectedSegment canvia a "recipes" mitjançant el mètode segmentChanged(), llavors es mostra el llistat de favoriteRecipes. El llistat de productes es gestiona mitjançant el component *Products*, el mateix que s'utilitza per mostrar la grid de productes. Com ja s'ha explicat anteriorment, aquest component gestiona el llistat de productes, en aquest cas, el llistat de productes existents als favorits de l'usuari que s'envia des de la *FavoritesPage*. Cada producte es representat pel component *Widget Product* que es comporta igual que s'ha descrit a la gestió de productes. Per altra banda, el llistat de receptes es gestiona mitjançant el component *Recipes* que té un comportament similar al que s'utilitza per al llistat de productes. Aquest component té l'atribut recipes que és el llistat de receptes guardades per l'usuari i s'envia des de la pàgina de favorits. Cada recepta se representa amb el component *Widget Recipe* que té l'objecte recipe amb tota la informació necessaria i dos mètodes: el deleteRecipeFromFavorites() que permet esborrar la recepta del llistat de favorits i getRecipeDetail() que redirigeix a l'usuari a la pantalla del detall de la recepta guardada. La *Recipe Detail Page* conté l'atribut recipeId que permetrà buscar la informació adicional de la recepta mitjançant el mètode searchRecipeIngredients() i l'atribut recipe que és el nom de la recepta a la que estem accedint. També té dos mètodes ja explicats anteriorment a la gestió de receptes: addSelectedProductsToCart() i recalculateAmount().

Pantalla de favorits

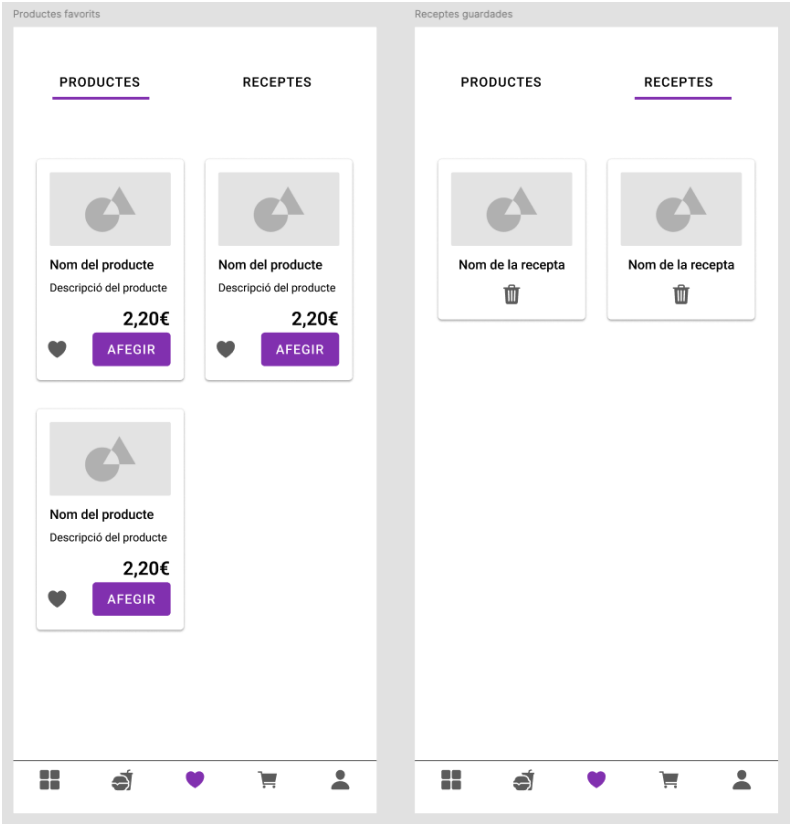

Figura 5.16: Disseny de la pantalla de favorits. (Font pròpia)

Detall d'una recpeta

Figura 5.17: Disseny de la pantalla del detall d'una recepta. (Font pròpia)

AUTENTIFICACIÓ

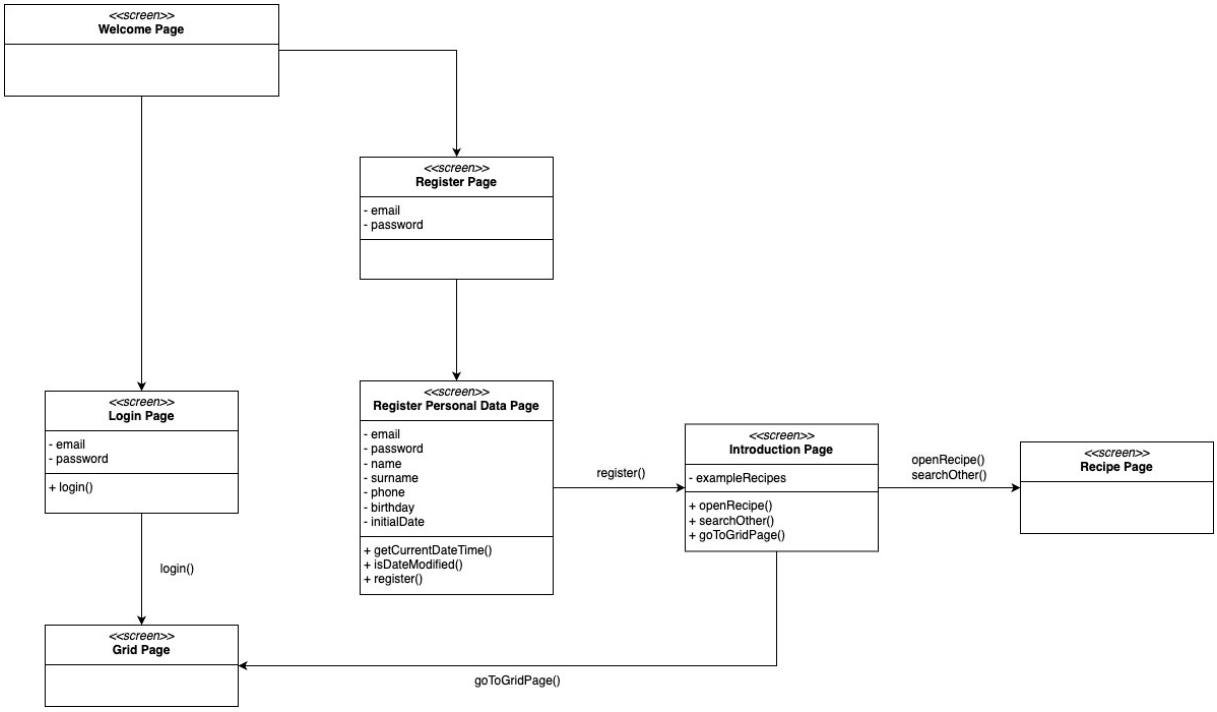

Figura 5.18: External Design de l'autentificació. (Font pròpia)

La *WelcomePage* és una pantalla de benvinguda/inici on l'usuari pot escollir si iniciar sessió, o en cas que no tingui cap compte creat, registrar-se a l'aplicació. Aquesta pantalla no té ni atributs ni mètodes específics, al clicar als botons es crea un *routerLink* per dirigir a l'usuari a les pàgines d'inici de sessió o de registre.
En cas d'escollir l'opció d'iniciar sessió, es redirigeix a l'usuari a la pantalla *Login Page*. Aquesta pantalla conté dos atributs obligatoris que son el correu electrònic i la contrasenya i també el mètode login() que autentica l'usuari i el redirigeix a la *Grid Page* descrita anteriorment.
Per altra banda, si l'usuari escolleix l'opció de registrar-se, se'l redirigeix a la pàgina *Register Page*. Aquesta pàgina, igual que la pàgina d'inici de sessió, conté dos atributs: email i password que són necessaris per poder continuar amb el procés de registre. Al clicar al botó de continuar, es crea un routerLink per redirigir a l'usuari a la pàgina *Register Personal Data Page* per tal de permet a l'usuari completar la informació adicional per al seu registre. Aquesta pàgina conté els atributs email i password, que provenen de la info completada anteriorment i també els atributs name, surname, phone, birthday i initialDate (data actual obtinguda a partir del mètode getCurrentDateTime()). Al prémer el botó "Registra't", s'executa la funció register(), aquesta necessita tenir els camps name i surname complets mentre que els valors per als atributs phone i birthday poden ser nuls. Amb la funció isDateModified() és verifica si l'usuari ha modificat la data de naixement o si realment l'usuari no a introduit cap data i el valor de birthday és el mateix que el de initialDate. Si el registre es completa correctament sense errors, es redirigeix a l'usuari a

la pàgina *Introduction Page*.
La *Introduction Page* es mostra al pas final del flow de registre per endinsar a l'usuari a l'aplicació i introduir-lo a l'ús de la funció més destacable de l'aplicació: la cerca d'ingredients i productes a partir del nom d'una recepta. Aquesta pantalla té un atribut exampleRecipes que conté una sèrie d'exemples de receptes per mostrar a l'usuari un exemple del funcionament de l'apartat de receptes de forma ràpida i senzilla. Mitjançant els mètodes searchRecipe() i searchOther() es redirigirà a l'usuari a la pàgina de receptes i així s'endinsarà a aquesta destacable funcionalitat. Per altra banda, si l'usuari decideix ometre aquesta introducció, se'l redireccionarà a la grid de productes.

Pantalla inicial

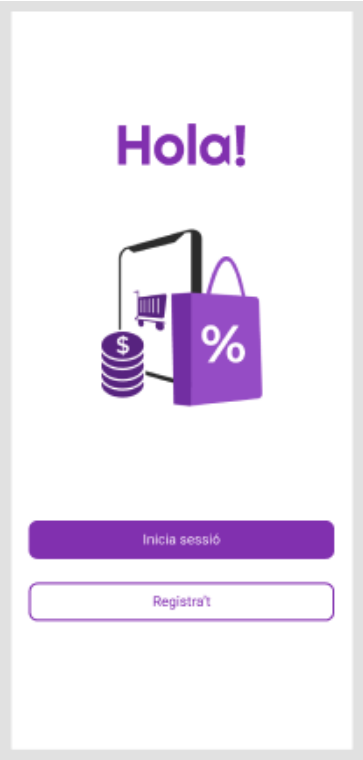

Figura 5.19: Disseny de la pantalla inicial de l'aplicació. (Font pròpia)

Inici de sessió

Figura 5.20: Disseny de la pantalla d'inici de sessió. (Font pròpia)

Registre

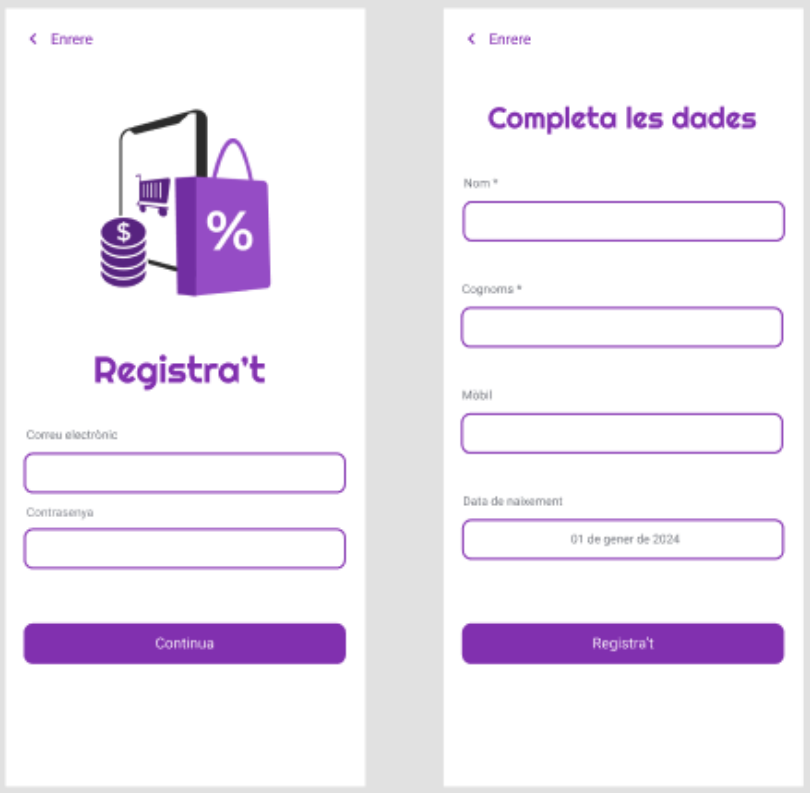

Figura 5.21: Disseny de les pantalles de registre. (Font pròpia)

Pantalla d'introducció

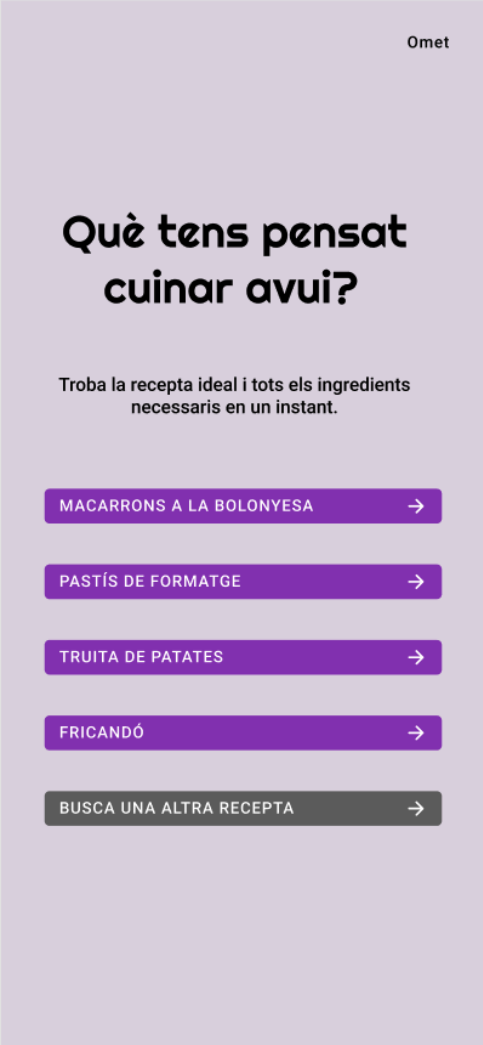

Figura 5.22: Disseny de la pantalla d'introducció. (Font pròpia)

PERFIL DE L'USUARI

La pantalla de perfil de l'usuari és una única pàgina que inclou tant la informació personal de l'usuari com el detall de les comandes realizades. Actualment, no s'ha creat un component específic per a la informació d'aquestes comandes ja que no es reutiliza per a cap altra pantalla, però és veritat que de cara un treball futur hagués sigut més correcte crear un component per a aquesta informació que sigui reutilitzable per a possibles amplicacions del treball i facilitar l'escalabilitat del projecte. A més a més, això faria que el codi sigui més fàcil de mantenir i de testejar i es reduiria la complexitat de la pantalla del perfil en general, permetent que cada part de l'aplicació es centri en una única funcionalitat.

Pantalla de perfil

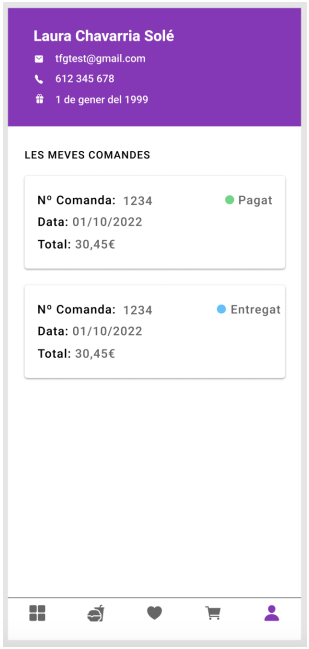

Figura 5.23: Disseny de la pantalla del perfil de l'usuari. (Font pròpia)

Implementació

6.1 Tecnologies, llenguatges utilitzats i eines de desenvolupament

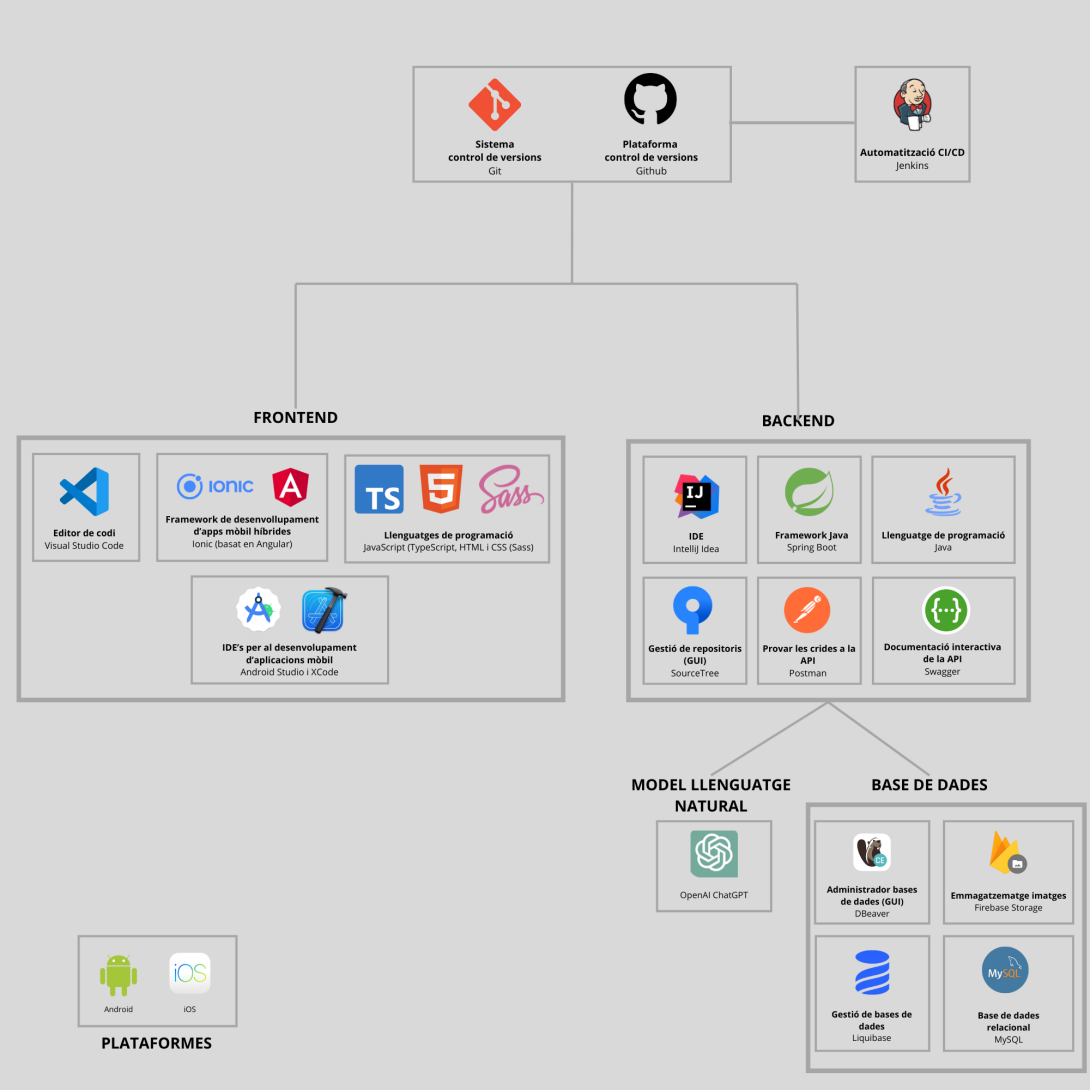

Figura 6.1: Tecnologies, llenguatges utilitzats i eines de desenvolupament. (Font pròpia)

La figura anterior 6.1, que també es pot veure més ampliada fent clic aquí, mostra les tecnologies, llenguatges de programació i eines de desenvolupament que s'han utilitzat per a dur a terme una aplicació mòbil e-commerce per a aquest projecte de final de grau. Aquesta imatge està organitzada en diverses seccions clau: control de versions i automatització, frontend, backend, base de dades, model de llenguatge natural i plataformes.

6.1.1 Sistema de control de versions i automatització

En la part superior del diagrama, es detallen les eines de control de versions i d'automatització utilitzades per al projecte. S'utilitza Git com a sistema de control de versions, una eina essencial per gestionar el codi font i les seves diferents versions. Per altra banda, GitHub és la plataforma triada per allotjar els repositoris Git, la qual permet una col·laboració eficient entre els membres de l'equip. En aquest cas, tot i que només hi ha un únic desenvolupador (l'autor del TFG), s'han utilitzat les funcionalitats de GitHub per a guardar el codi i mantenir un historial detallat dels canvis realitzats. Finalment, per a l'automatització del procés d'integració i desplegament continu (CI/CD), s'ha utilitzat Jenkins, una eina que facilita l'automatització de les tasques de construcció, proves i desplegament del programari, millorant així l'eficiència del desenvolupament.

6.1.2 Frontend

Per al desenvolupament de la part frontal de l'aplicació, s'han seleccionat diverses tecnologies i eines. Visual Studio Code és l'editor de codi principal utilitzat per escriure i editar el codi del frontend. Pel que fa al framework de desenvolupament, s'ha triat Ionic [8] (basat en Angular [9]) que permet crear aplicacions híbrides utilitzant tecnologies com HTML, CSS i JavaScript. Aquest framework és especialment valuós perquè permet desenvolupar aplicacions per a dispositius Android i iOS a partir d'un sol codi base.

Els llenguatges de programació utilitzats per al frontend són JavaScript (en la seva variant TypeScript), HTML i CSS (amb el preprocessador Sass). Aquests llenguatges són la base per construir una interfície d'usuari dinàmica i estilitzada. A més, per al desenvolupament d'aplicacions mòbils específiques per Android i iOS, es fan servir Android Studio i Xcode respectivament, que són les eines oficials per desenvolupar aplicacions en aquestes plataformes.

6.1.3 Backend

La part backend de l'aplicació es desenvolupa utilitzant Spring Boot, una extensió del framework Spring que facilita la configuració i el desplegament d'aplicacions Java. Per altra banda, l'entorn de desenvolupament integrat (IDE) triat és IntelliJ IDEA, conegut per la seva eficàcia en la programació en Java.

Per gestionar els repositoris de codi amb una interfície gràfica, s'utilitza SourceTree, que facilita el treball amb Git i permet visualitzar els canvis i les branques del codi de manera clara i intuïtiva, millorant així la gestió del codi font. Postman és l'eina seleccionada per provar les crides a l'API, assegurant que totes les funcionalitats del backend es comporten com s'espera. I, finalment, per la documentació interactiva de l'API s'utilitza Swagger, que permet generar documentació fàcilment comprensible i utilitzable per a desenvolupadors.

6.1.4 Base de dades

La gestió de la base de dades es fa amb Liquibase, una eina que permet controlar i aplicar canvis a la base de dades de manera estructurada. La base de dades utilitzada és MySQL, un sistema de gestió de bases de dades relacional conegut per la seva fiabilitat i eficiència. Per administrar la base de dades amb una interfície gràfica, es fa servir DBeaver (o també es podria utilitzar HeidiSQL per a sistemes operatius Windows), una eina que simplifica les tasques de gestió i consulta de la base de dades. Per l'emmagatzematge d'imatges, s'utilitza Firebase Storage, que ofereix una solució escalable i fàcil d'usar per gestionar els arxius multimèdia de l'aplicació (en aquest cas imatges únicament).

6.1.5 Model de llenguatge natural

El model de llenguatge natural utilitzat en aquest projecte es basa en l'OpenAI ChatGPT. ChatGPT permet implementar la funcionalitat que facilita als usuaris l'accés ràpid i eficient als ingredients necessaris per a qualsevol recepta que triïn. El model està entrenat per comprendre les consultes en llenguatge natural i retornar una llista precisa d'ingredients, optimitzant l'experiència de compra dins de l'aplicació mòbil d'e-commerce.

6.1.6 Plataformes

Finalment, les plataformes objectiu per a l'aplicació són Android i iOS. Aquestes plataformes són els principals sistemes operatius per a dispositius mòbils i la compatibilitat amb totes dues és crucial per arribar a la majoria dels usuaris potencials.

Aquest conjunt de tecnologies, llenguatges i eines ha estat triat estratègicament per assegurar un desenvolupament eficient, escalable i mantenible de l'aplicació mòbil d'e-commerce, tenint en compte tant l'experiència prèvia de l'equip com les necessitats específiques del projecte.

6.2 Anàlisi d'alternatives

En aquest apartat es presenten les alternatives considerades per al desenvolupament de la nostra aplicació mòbil d'e-commerce i es justifica el perquè de l'elecció, tant per al frontend com per al backend.

6.2.1 Frontend

Ionic és el framework escollit per al desenvolupament de la nostra aplicació mòbil. Aquest framework permet el desenvolupament híbrid basat en tecnologies com HTML, CSS i JavaScript i utilitzant Angular com a framework subjacent, permetent crear aplicacions per a dispositius Android i iOS en un únic codi base. Es va escollir aquesta tecnologia perquè al haver treballat anteriorment en Angular, amb aquest framework es facilitava la corba d'aprenentatge. Tot i així, anteriorment a l'inici del projecte es va realitzar un curs d'Ionic per endinsar-se al desenvolupament d'aplicacions mòbils amb aquesta tecnologia.

D'altra banda, es van estudiar altres alternatives com React Native o Flutter. React Native era l'altra possible elecció ja que s'havia treballat anteriorment amb aquesta tecnologia però com l'amplitud de coneixements d'Angular era molt més ampli, es va acabar descartant. Per altra part, Flutter era un altra alternativa de desenvolupament híbrid però degut que no es tenien coneixements previs sobre Dart es va excloure de les opcions ja que augmentava en excés la corba d'aprenentatge.

Finalment, cal mencionar altres tecnologies com Swift o Kotlin que ja no es van tenir en compte ja que només permitien el desenvolupament exclusiu per a iOS i Android respectivament.

6.2.2 Backend

Spring Boot és el framework escollit per al desenvolupament del backend de la nostra aplicació mòbil. Sping Boot és una extensió del framework Spring que simplifica la configuració i el desplegament d'aplicacions basades en Java. Es va escollir aquesta tecnologia perquè proporciona una solució robusta i escalable per al desenvolupament d'APIs RESTful, i l'equip de desenvolupament té experiència prèvia treballant amb Java, la qual cosa facilita la corba d'aprenentatge i la integració de la solució. A més, Spring Boot té una gran comunitat de suport i una àmplia gamma de biblioteques i eines disponibles.

Tot i així, abans de prendre la decisió final, es van estudiar altres alternatives per al desenvolupament del backend, com ara Node.js amb Express però al haver treballat anteriorment en Java per al desenvolupament de backend, ja es va descartar.

6.2.3 Model de llenguatge natural

Per enendre aquest apartat, cal diferenciar entre dos tipus de models de llenguatge natural.

- **Models accedits via API:** Aquests models, com el OpenAI ChatGPT, són accessibles mitjançant una interfície de programació d'aplicacions (API). L'usuari envia dades al servidor del proveïdor de serveis, on el model processa les dades i retorna els resultats. Aquests models són gestionats, actualitzats i mantinguts pel proveïdor de serveis, reduint la càrrega de gestió de recursos i infraestructura per a l'usuari final.
- **Models desplegats localment:** Models com Ollama Llama3 o Hugging Face es poden instal·lar i executar en el hardware local de l'usuari. Aquests models ofereixen un major control sobre les dades i el processament, ja que tot es manté dins de l'infraestructura de l'usuari, augmentant així la privacitat i la seguretat.

A continuació, es troba una taula 6.1 que resumeix les principals diferències entre aquests dos tipus:

Característica	**Models via API**	**Models desplegats localment**
Gestió	Servidor pel proveïdor	Gestió local per l'usuari
Privacitat de les dades	Dades enviades a servidors externs	Dades processades localment
Costos	Costos basats en l'ús	Costos més baixos de gestió a llarg termini
Dependència	Alta (depèn del proveïdor)	Baixa (independència del proveïdor)
Facilitat d'ús	Alta (poca configuració)	Mitjana (requereix configuració)
Actualitzacions	Automàtiques	Manual

Taula 6.1: Comparació de models de llenguatge natural: API vs Local. (Font pròpia)

OpenAI ChatGPT és el model de llenguatge natural escollit per proporcionar la funcionalitat d'IA a la nostra aplicació. Es va escollir aquest model perquè proporciona una solució robusta i escalable per al processament del llenguatge natural (NLP). A més, ChatGPT té una gran comunitat de suport i una àmplia gamma de biblioteques i eines disponibles. Per altra banda, es va estudiar també la possibilitat d'utilitzar Ollama Model llama3. Aquest model permet l'execució en local, augmentant així la privacitat i la seguretat de les dades. Tot i això, aquest model no té el mateix nivell d'entrenament que el model d'OpenAI i té una comunitat menys desenvolupada. AA diferència del model d'OpenAI, aquest no requereix una subscripció de pagament, cosa que pot ser beneficiosa per reduir costos a llarg termini. Per aquestes raons, tot i que OpenAI ChatGPT és l'elecció actual per les seves capacitats avançades i el suport ampli, en un futur es podria considerar la migració a un model com Ollama Llama3 per millorar la privacitat i reduir els costos operatius.

A la taula següent 6.2, es poden veure les caracterísitques de cada model descrit a les alternatives anteriors.

Característica	**OpenAI ChatGPT**	**Ollama Llama3**
Accessibilitat	Alta (accés via API)	Mitjana (requerix instal·lació local)
Privacitat	Les dades es poden enviar a servidors externs	Les dades es processen localment, augmentant la privacitat
Costos	Depèn de la tarifa d'ús de l'API, pot ser alt per a un ús intensiu	No requereix costos de subscripció, pot ser més econòmic a llarg termini
Manteniment	Gestionat pel proveïdor	Requereix manteniment local
Actualitzacions	Automàtiques	Manuals
Comunitat i Suport	Gran comunitat i ampli suport	Comunitat més petita i suport limitat
Capacitat d'entrenament	Models altament entrenats i optimitzats	Pot requerir més personalització i entrenament local

Taula 6.2: Comparació d'Avantatges i Desavantatges entre OpenAI ChatGPT i Ollama Llama3. (Font pròpia)

6.3 Aspectes rellevants del codi d'implementació

6.3.1 Procés implementat per a la cerca d'ingredients i productes a partir del nom d'una recepta

D'aquesta aplicació, es vol destacar especialment la funcionalitat de les receptes. Des del front existeix una pantalla que a través d'un input es permet buscar els ingredients necessaris per poder cuinar la recepta introdu¨da. Això és gràcies al model de llenguatge natural OpenAI. L'usuari introdueix el nom d'una recepta a l'input, per exemple, "Espaguetis a la carbonara" i envia la sol·licitud a l'API mitjançant una petició GET. El controlador rep la sol·licitud i invoca el servei "ChatGPTService" per processar la recep-

ta introduïda mitjançant el mètode sendMesssage(recipe) on recipe és la query "Enumera únicament els ingredients bàsics per cuinar valor introduït al input, sense incloure quantitats ni explicacions addicionals.". D'aquesta manera, el servei processa la sol·licitud i genera una resposta que conté únicament els ingredients necessaris per a la recepta, obtenint així una estructura de resposta més estructurada i limitada, cosa que facilita el processament posterior dels valors d'aquesta resposta a la nostra API. Aquesta resposta s'envia al controlador i aquest invoca el servei ProductService per buscar els productes relacionats amb els ingredients. Aquest servei busca tots els identificadors dels productes de la base de dades que coincideixen i els ordena segons el nombre de paraules claus de l'ingredient que coincideixen amb el nom del producte. Per exemple, si tenim l'ingredient "Formatge parmesà ratllat" i tenim 2 productes diferents: formatge parmesà ratllat i formatge fos ratllat, s'afegiran tots dos productes amb aquest mateix ordre a la llista de productes coincidents per a l'ingredient corresponent i es guardarà també les paraules claus de l'ingredient que no apareixen al nom del producte. Un cop obtinguts aquests identificadors, el servei s'encarregarà de buscar i mapejar la informació de cada producte i retornarà la resposta al controlador. Finalment, és aquest el que s'encarregarà d'enviar la resposta de la petició al front i allí es processarà la resposta per mostrar-la tal com s'ha dissenyat mostrant com a seleccionats el producte amb més percentatge de coincidències per cada ingredient. Des del llistat de productes per ingredient, es pot seleccionar o desseleccionar qualsevol producte, però no es pot seleccionar més d'un producte per ingredient. Quan se selecciona un producte es desselecciona automàticament el producte anteriorment seleccionat i passa a mostrar-se a la primera posició de la llista, recalculant el pressupost total dels productes seleccionats.

Tot aquest procés es pot veure esquematitzat a la següent imatge o fent clic aquí.

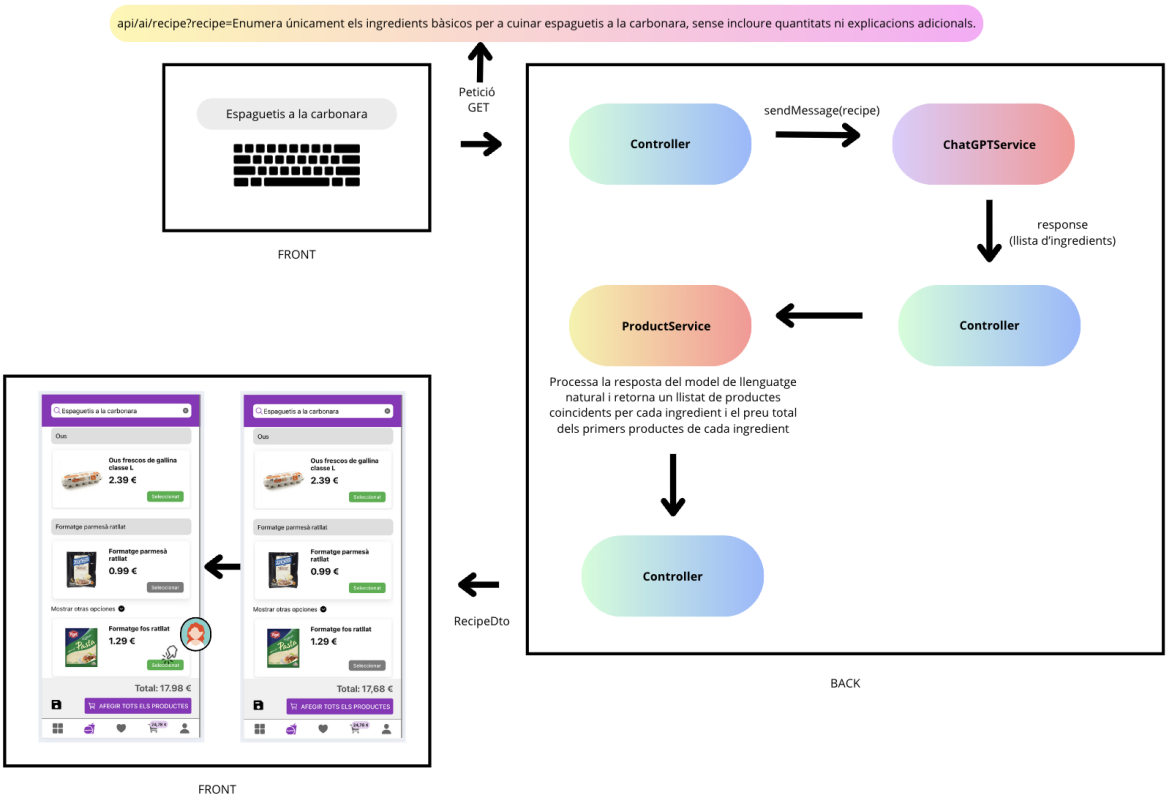

Figura 6.2: Procés implementat per a la cerca d'ingredients i productes a partir del nom d'una recepta. (Font pròpia)

6.3.2 Alternativa utilitzada per al procés de pagament

Per al desenvolupament del sistema de pagament estava pensat implementar-ho amb Redsys, però finalment es va decidir utilitzar Stripe com a alternativa, ja que vam tenir alguns problemes en la integració de Redsys al nostre projecte Spring Boot i a causa de les limitacions de temps establertes per la planificaió inicial del projecte. Vam creure que com aquesta funcionalitat era simplement per simular un pagament abans de completar la comanda i no era un dels requisits més importants del projecte, llavors era més convenient tirar endavant amb un altre sistema de pagament alternatiu per completar la funcionalitat i no endarrerir-nos amb el desenvolupament de la resta d'èpiques. Stripe és una plataforma bastant coneguda mundialment que permet acceptar pagaments i que destaca per la seva facilitat d'integració, seguretat i extensibilitat. A l'inici del projecte es va plantejar desenvolupar aquesta funcionalitat mitjançant Redsys, ja que era el sistema de pagament conegut per l'autor, però després d'un anàlisi exhaustiu d'alternatives, vam trobar aquesta opció que ens ha permès finalitzar l'èpica amb èxit.

Anàlisi d'avantatges i desavantatges d'ambdós sistemes de pagament

A continuació, es mostra una taula 6.3 amb els principals avantatges i desavantatges del sistemes de pagament Redsys i Stripe.

Sistemes de pagament	**Redsys**	**Stripe**
Avantatges	Integració amb bancs espanyols Preus, comissions i tarifes negociables Seguretat Acceptació de targetes locals	Facilitat d'integració Ampla varietat de mètodes de pagament Seguretat Suport tècnic Escalabilitat
Desavantatges	Complexitat d'integració Flexibilitat limitada Suport tècnic	Comissions Mètodes de pagament locals

Taula 6.3: Taula comparativa entre els sistemes de pagament Redsys i Stripe. (Font pròpia)

Proves

En aquest apartat s'expliquen tots els tests que s'han realitzat durant el desenvolupament del projecte per comprovar que el sistema funciona correctament.

Aquests inclouen tests unitaris de la part del backend, tests de disseny i usabilitat, així com la integració contínua mitjançant Jenkins per automatitzar i optimitzar el procés de desenvolupament. Els tests unitaris s'han realitzat per comprovar que les respostes de l'API retornen les dades correctes i els tests de disseny s'han realitzat per comprovar que el disseny del sistema és correcte i que els usuaris són capaços d'entendre i d'utilitzar les interfícies.

7.1 Tests unitaris *Backend*

S'han implementat tests unitaris per diverses funcionalitats del backend, amb l'objectiu principal de comprovar que la informació que retorna l'API és precisa i correcta tant en format com en contingut. Aquestes proves són essencials per garantir la fiabilitat i l'estabilitat de l'API, la qual serveix com a pont de comunicació entre la nostra base de dades i el frontend de la nostra aplicació.

Per a la implementació dels tests unitaris hem utilitzat JUnit, un framework de testing per a Java que és àmpliament reconegut i utilitzat. JUnit s'integra perfectament amb entorns de desenvolupament integrats (IDEs) com IntelliJ IDEA, així com amb sistemes d'integració contínua/desplegament continu (CI/CD) com Jenkins. Aquesta integració ens ha facilitat l'execució i la revisió dels resultats dels tests d'una manera automàtica i contínua.

7.2 Tests de disseny i usabilitat

S'han realitzat tests de disseny després de la realització dels prototips abans de fer la seva implementació i tests d'usabilitat després de la implementació de cada èpica per

comprovar que els usuaris entenen les funcionalitats de cada pantalla sense necessitat de cap explicació o aclariment.

Per als tests de disseny, s'han presentat els dissenys a 3 persones diferents i se'ls ha demanat la seva clara opinió per saber si el prototip de la pantalla els semblava atractiu i si entenien clarament les funcionalitats de cada pantalla. Al ser pantalles de compra dissenyades molt semblants a altres aplicacions per evitar confusions i facilitar l'ús a l'usuari, els tests van ser completats amb èxit.

Per altra banda, per als tests d'usabilitat, es va fer un llistat de les tasques que havien de completar els testers (sense donar cap tipus d'explicació). Aquesta llista es va presentar a 3 persones de diferents edats des de 18 anys fins als 70 anys) per comprovar que sigui usable per a tots els usuaris possibles de l'aplicació. L'objectiu d'aquests tests era que els usuaris completessin les tasques en un temps limitat i també es va aconseguir amb èxit amb tots els casos, ja que com hem explicat anteriorment, el funcionament d'aquesta aplicació és semblant a la majoria d'aplicacions e-commerce existents al mercat i això facilita el seu ús.

7.3 Integració contínua amb Jenkins

S'ha implementat un sistema d'integració contínua (CI) utilitzant Jenkins, essencial per millorar l'eficiència i la qualitat del nostre desenvolupament tant de frontend com de backend. Cada vegada que es fa un *commit* al repositori, Jenkins automàticament inicia un procés de CI que inclou la compilació del codi i l'execució de tests automatitzats en el cas del repositori backend. Aquesta automatització assegura que tots els canvis són validats en termes de funcionalitat i estabilitat abans de procedir a la seva integració final.

Per garantir la integritat del codi principal, és obligatori que el procés de CI completat per Jenkins finalitzi amb èxit abans de fer un *merge* a la branca principal.

Aquesta pràctica no només ajuda a mantenir el codi lliure de bugs, sinó que també garanteix que el nou codi s'integra sense complicacions amb les funcionalitats existents, proporcionant una base sòlida per a un desenvolupament continu i eficaç.

Execució del projecte

El core desenvolupat és el ***Chef Mate***, una aplicació mòbil e-commerce destinada als supermercats. S'ha escollit aquest nom per destacar la funcionalitat més important de l'aplicació, la cerca de receptes. Totes les aplicacions ja existents tenen gairebé sempre les mateixes funcionalitats bàsiques, però per innovar i endinsar-se al món de la intel·ligència artificial es va decidir implementar aquesta nova funció que permet als usuaris buscar productes per cuinar les seves receptes d'una forma molt ràpida i senzilla. D'aquí l'eslògan del producte: "La teva recepta, els nostres ingredients. Cuina amb un clic!".

Figura 8.1: Logotip creat per a l'aplicació del nostre projecte. (Font pròpia)

8.1 Canvis en la planificació inicial

En aquest apartat es descriuen els canvis que s'han hagut d'aplicar a la planificació inicial fins al tercer sprint. Cal anar comparant aquests canvis amb la planificació inicial, ja que ens permetrà analitzar els errors de planificació i aprendre per poder crear una millor planificació en projectes futurs.

8.1.1 Canvis de la planificació de les tasques respecte a la planificació inicial

Les hores de dedicació de cada sprint s'han mantingut, però s'ha hagut d'ajustar les hores dedicades a cada tasca, posant més èmfasi a les funcionalitats de l'èpica i menys importància als bugs, ja que la quantitat de bugs a arreglar no són tants com els que ens esperàvem en un inici.

Aquests canvis estan dins del que s'esperava en la metodologia SCRUM, per tant, malgrat que hi ha hagut variacions, no es considera que hi hagi hagut un problema greu que afecti el termini de lliurament del projecte.

A l'*inception* i el primer *sprint* no s'ha hagut d'aplicar cap canvi en la dedicació d'hores de cada tasca, en canvi, a partir de l'sprint 2, sí. A les imatges següents, es poden veure els petits canvis aplicats.

Canvis aplicats a l'sprint 2

	Sprint 2	70		
C	Gestió de la cistella de la compra	62	P	-
GB	Gestió de bugs	2	P, M	Jira + altres
REV	Sprint Review	2	C, GB	Overleaf, Confluence i Jira
RET	Sprint Retrospective	2	REV	Overleaf, Confluence, EasyRetro i Jira
DOC	Documentació	2	-	Overleaf

Figura 8.2: Canvis aplicats a l'spint 2. (Font pròpia)

Codi	Tasca	Descripció	Duració	Dependències	Recursos
C	Gestió de la cistella de la compra		62		
C1	[LIQUIBASE] - Creació de taules	Creació de les taules i atributs necessaris.	2	-	IntelliJ, Heidi SQL
C2	[BACK] - Cart controller	Desenvolupament del controlador de la cistella de la compra, responsable de gestionar les operacions relacionades.	2	C1	IntelliJ
C3	[INFO] - Sistemes de pagament i gestió de la cistella	Cerca d'informació per poder realitzar el pagament de la compra.	8	-	-
C4	[BACK] - Sistema de pagament	Desenvolupament de funcionalitats relacionades amb el pagament de la compra.	24	C2, C3	IntelliJ, Postman
C5	[BACK] - Endpoints	Creació de endpoints realacionats amb la cistella de la compra per a la interacció amb l'aplicació.	12	C4	IntelliJ, Postman
C6	[DISSENY UI/UX] - Disseny cistella + pantalla pagament	Disseny i desenvolupament de l'estructura que ha de tenir la pantalla de la cistella de la compra i el pagament.	2	-	Figma
C7	[FRONT] - Cistella de la compra + pantalla pagament	Implementació de la cistella de la compra i la pantalla de pagament.	12	C6	VSC

Figura 8.3: Canvis aplicats a la distribució d'hores de les tasques de l'sprint 2. (Font pròpia)

Com es pot veure a les figures 8.2 8.3, a l'*sprint 2* s'ha hagut d'augmentar el nombre d'hores destinades a la implementació de l'èpica de la cistella de la compra, reduint així les hores dedicades a la gestió de bugs i a la documentació. Aquest augment d'hores va ser provocat pel fet que la implementació del sistema de pagament va comportar més hores

de recerca perquè el primer intent d'implementació amb Redsys ens va comportar molts problemes i es va haver de buscar una solució per poder implementar aquesta funcionalitat.

Canvis aplicats a l'sprint 3

	Sprint 3	70		
R	Gestió de receptes	48	P	-
GB	Gestió de bugs	8	C	Jira + altres
REV	Sprint Review	2	R, GB	Overleaf, Confluence i Jira
RET	Sprint Retrospective	2	REV	Overleaf, Confluence, EasyRetro i Jira
DOC	Documentació	10	-	Overleaf

Figura 8.4: Canvis aplicats a l'spint 3. (Font pròpia)

A l'*sprint 3* només s'ha reduït el nombre d'hores destinades a la gestió de bugs, ja que l'abast no era tan gran i s'ha augmentat la dedicació a la redacció de la part escrita perquè en l'sprint anterior no es va poder redactar tota la informació corresponent a l'*sprint*.

Canvis aplicats a l'sprint 4 i 5

	Sprint 4	62		
A	Autentificació	40	-	-
~~F~~	~~Gestió de favorits~~	~~46~~	~~P~~	-
GB	Gestió de bugs	10	R	Jira + altres
REV	Sprint Review	2	A, F, GB	Overleaf, Confluence i Jira
RET	Sprint Retrospective	2	REV	Overleaf, Confluence, EasyRetro i Jira
DOC	Documentació	8	-	Overleaf

Figura 8.5: Canvis aplicats a l'spint 4. (Font pròpia)

	Sprint 5	68		
F	Gestió de favorits	46	P	-
~~B~~	~~Búsqueda de productes~~	~~0~~	~~P~~	-
GB	Gestió de bugs	10	A, F	Jira + altres
REV	Sprint Review	2	P, GB	Overleaf, Confluence i Jira
RET	Sprint Retrospective	2	REV	Overleaf, Confluence, EasyRetro i Jira
DOC	Documentació	8	-	Overleaf

Figura 8.6: Canvis aplicats a l'spint 5. (Font pròpia)

Codi	Tasca	Descripció	Duració	Dependències	Recursos
A	Autentificació		40		
A1	[LIQUIBASE] - Creació de taules	Creació de les taules i atributs necessaris.	2	-	IntelliJ, Heidi SQL
A2	[INFO] - Registre + Inici de sessió	Cerca d'informació per poder fer el registre i l'inici de sessió.	2	-	-
A3	[BACK] - User controller	Desenvolupament del controlador dels usuaris, responsable de gestionar les operacions relacionades.	2	A1, A2	IntelliJ
A4	[BACK] - Endpoints	Creació de endpoints realacionats amb la gestió d'usuaris per a la interacció amb l'aplicació.	16	A3	IntelliJ, Postman
A5	[DISSENY UI/UX] - Disseny perfil usuari	Disseny i desenvolupament de l'estructura que ha de tenir la pantalla de perfil.	2	-	Figma
A6	[FRONT] - Perfil	Implementació de la pantalla de perfil de l'usuari.	16	A5	VSC
F	Gestió de favorits		46		
F1	[LIQUIBASE] - Creació de taules	Creació de les taules i atributs necessaris.	2	-	IntelliJ, Heidi SQL
F2	[BACK] - Endpoints	Creació de endpoints realacionats amb la gestió de favorits per a la interacció amb l'aplicació.	18	F1	IntelliJ, Postman
F3	[DISSENY UI/UX] - Disseny favorits	Disseny i desenvolupament de l'estructura que ha de tenir la pantalla de favorits.	2	-	Figma
F4	[FRONT] - Favorits	Implementació de la pantalla de favorits.	24	F3	VSC

Figura 8.7: Canvis aplicats a la distribució d'hores de les tasques de l'sprint 4 i 5. (Font pròpia)

Com es pot veure a les imatges anteriors 8.5, 8.6 i 8.7, les èpiques d'autentificació i gestió de favorits han requerit una destinació d'hores superior a la planificada inicialment. És per això, que s'ha decidit repartir-les en dos sprints diferents i eliminar l'èpica de cerca de productes. S'ha pres aquesta decisió ja que s'ha preferit aprofundir més amb l'èpica de receptes i la seva gestió perquè és una funcionalitat relativament nova que la majoria d'aplicacions e-commerce no tenen i pot ser interessant fer un desenvolupament més extens en aquesta èpica en lloc de focalitzar-nos amb la cerca de productes, que és una funcionalitat que ja tenen totes les aplicacions i que pot ser desenvolupada en un futur.

El nombre total d'hores dedicades a la realització del projecte no s'ha modificat, ja que s'ha adaptat als requisits d'aquest projecte de final de grau.

TASQUES BUGS I MILLORES ABORDADES EN CADA SPRINT

En aquest subapartat es descriuen tots els bugs i les millores que s'han abordat en cada sprint tenint en compte les hores destinades a la gestió de bugs, que també inclou algunes millores.

Sprint 2

- **TFG24-59:** [FRONT] - L'scroll del detall del producte en mòbils petits, no permet veure tota la informació
  - **Duració:** 2h

Sprint 3

- **TFG24-60:** [FRONT] - Actualització de dades al canviar de pantalles (Grid-Cart)
  - **Duració:** 4h
- **TFG24-62:** [FRONT] - Redirect al finalitzar el pagament d'una compra
  - **Duració:** 2h
- **TFG24-66:** [FRONT] - Millora: Revisar labels
  - **Duració:** 2h

Sprint 4

- **TFG24-69:** [DISSENY + FRONT] - Afegir un contingut estàtic a la pantalla de receptes quan no s'ha realitzat cap cerca d'una recepta.
  - **Duració:** 10h

Sprint 5

- **TFG24-70:** [BACK] - Arreglar error a l'endpoint de registre.
  - **Duració:** 8h
- **TFG24-71:** [FRONT] - Configurar colors de l'estat de la comanda de la pantalla del perfil
  - **Duració:** 2h

8.1.2 Canvis dels recursos de software respecte a la planificació inicial

Cal destacar també que, a part dels canvis relacionals en la planificació de les tasques, també s'han hagut de modificar i afegir alguns recursos de software.

- Com a eina que ens permet administrar bases de dades amb una interfície gràfica d'usuari, s'ha hagut d'utilitzar també *DBeaver*, ja que *HeidiSQL* no està disponible per a dispositius amb sistema operatiu iOS.
- Per poder portar a terme l'èpica de la gestió de receptes, hem utilitzat *OpenAI Platform*, una API d'OpenAI que permet als desenvolupadors integrar fàcilment les capacitats d'intel·ligència artificial. També cal destacar que s'ha buscat una alternativa local gratuïta (Ollama) que es comentarà amb posterioritat.

8.2 Afectacions dels canvis de la planificació inicial al pressupost

Respecte als costos humans 8.8, cal destacar que, el total d'hores treballades per rol ha canviat augmentant el nombre d'hores del programador júnior i reduint el nombre d'hores del cap de projecte i del dissenyador. D'aquesta manera, com el preu per hora del cap de projecte i del dissenyador és més elevada que el del programador, el total dels costos humans s'ha reduït 27,54€.

- **Tasca C3:** *(2 hores estimades - 8 hores reals) x 15,39 €/hora = -92,34 €.*

  La tasca C3 ha costat 92,34€ més del previst degut a que va requerir 6 hores més de les estimades inicialment.

- **Tasca C4:** *(16 hores estimades - 24 hores reals) x 15,39 €/hora = -123,12 €.*

  La tasca C4 ha costat 123,12€ més del previst degut a que va requerir 8 hores més de les estimades inicialment.

- **Gestó de bugs (Sprint 2):** *(10 hores estimades - 2 hores reals) x 15,39 €/hora = 123,12 €.*

  Les tasques relacionades amb la gestió de bugs han suposat un estalvi de 123,12€ degut a que va requerir 8 hores menys de les estimades inicialment.

- **Documentació (Sprint 2):** *(8 hores estimades - 2 hores reals) x 21,96 €/hora = 131,76 €.*

  La tasca relacionada amb la redacció de la part escrita del treball ha suposat un estalvi de 131,76€ degut a que es van dedicar 6 hores menys de les estimades inicialment.

  Finalment, el cost de les tasques de l'sprint 2 han suposat un estalvi de 39,42€.

- **Gestó de bugs (Sprint 3):** *(10 hores estimades - 8 hores reals) x 15,39 €/hora = 30,78 €.*

  Les tasques relacionades amb la gestió de bugs han suposat un estalvi de 30,78€ degut a que va requerir 2 hores menys de les estimades inicialment.

- **Documentació (Sprint 3):** *(8 hores estimades - 10 hores reals) x 21,96 €/hora = -43,92 €.*

  La tasca relacionada amb la redacció de la part escrita del treball ha costat 43,92€ més del previst ja que es van dedicar 2 hores més de les estimades inicialment.

  Finalment, el cost de les tasques de l'sprint 3 ha suposat un cost de 13,14€ més que el previst inicialment ja que les dues hores no destinades a la gestió de bugs es

van destinar a la documentació i com per cost per hora del cap de projecte és més elevat, llavors s'ha desviat el cost.

- **Tasca A6:** *(4 hores estimades - 16 hores reals) x 15,39 €/hora = -184,68 €*

  La tasca A6 ha costat 184,68€ més del previst degut a que va requerir 12 hores més de les estimades inicialment.

  Finalment, el cost de les tasques de l'sprint 4 ha suposat un estalvi de 307,80€ ja que al eliminar l'èpica de gestió de favorits i passar-la a l'sprint 5, la quantitat d'hores destinades a l'implementació de codi s'ha reduït un total de 20 hores.

- **Tasca F2:** *(24 hores estimades - 18 hores reals) x 15,39 €/hora = 92,34 €*

  La tasca F2 ha suposat un estalvi de 92,34€ degut a que va requerir 6 hores menys de les estimades inicialment.

- **Tasca F4:** *(4 hores estimades - 24 hores reals) x 15,39 €/hora = -307,80 €*

  La tasca F4 ha costat 307,80€ més del previst degut a que va requerir 20 hores més de les estimades inicialment.

  Finalment, el cost de les tasques de l'sprint 5 ha suposat un cost de 61,56€ ja que al afegir tota l'èpica de gestió de favorits i eliminar l'èpica de búsqueda de productes, la quantitat d'hores destinades a l'implementació de codi s'ha augmentat un total de 4 hores.

COSTOS HUMANS	Sou brut / any	Sou brut / hora	Costos SS / any [18]	Costos SS / hora	Cost total / hora	Hores treballades	Cost total
Cap de projecte [14]	34.236,00 €	16,46 €	11.434,82 €	5,50 €	21,96 €	193	4.237,73 €
Junior Developer [15]	24.000,00 €	11,54 €	8.016,00 €	3,85 €	15,39 €	308	4.740,83 €
Junior QA [16]	22.768,00 €	10,95 €	7.604,51 €	3,66 €	14,60 €	25	365,05 €
Dissenyador UI/UX [17]	25.000,00 €	12,02 €	8.350,00 €	4,01 €	16,03 €	14	224,47 €
TOTAL							9.568,08 €

Figura 8.8: Canvis dels costos dels recursos humans. (Font pròpia)

Per altra banda, s'ha hagut d'afegir el cost de l'ús de l'API del OpenAI Platform als costos de software 8.9. És un cost suficientment baix que no afecta gaire al pressupost final però en un futur, si el nombre de crides a l'API augmenta, aquest cost augmenataria. És per això que s'estudia l'alternativa d'un model OpenSource per evitar aquest cost.

COSTOS SOFTWARE	Preu
Compte desenvolupador Google Play [19]	25,00 €
Compte desenvolupador App Store [19]	100,00 €
OpenAI Platform (usage credit)	5,89 €
TOTAL	130,89 €

Figura 8.9: Canvis dels costos de software. (Font pròpia)

Aspectes legals

Aquest apartat descriu els aspectes legals rellevants per al desenvolupament i distribució de la nostra aplicació. Inclou les lleis aplicables al projecte i les llicències dels components utilitzats.

9.1 Lleis aplicables al projecte

- **Llei Orgànica de Protecció de Dades i Garantia dels Drets Digitals [40]:** És una normativa espanyola que adapta el Reglament General de Protecció de Ddes (GDPR) de la Unió Europea al marc jurídic espanyol. L'objectiu principal és garantir la protecció de dades personals dels ciutadans i regular els drets digitals. Aquesta llei estableix les bases per a la recollida, emmagatzematge i tractament de dades personals. Aquesta llei és aplicable al nostre projecte ja que s'han de tractar dades personals dels usuaris com ara informació de contacte o adreces d'enviament.

- **Llei de Serveis de la Societat de la Informació i del Comerç Electrònic [41]:** És una normativa espanyola que regula els serveis prestats per via electrònica, incloent-hi el comerç electrònic. L'objectiu principal és establir un marc legal clar per a les activitats que es desenvolupen a través d'Internet. Aquesta llei exigeix que les empreses proporcionin informació clara i accessible sobre la seva identitat i les seves dades de contacte.

- **Llei General per a la Defensa dels Consumidors i Usuaris [42]:** És una normativa espanyola que té com a objectiu principal protegir els drets dels consumidors i usuaris en les seves relacions amb les empreses i els proveïdors de serveis. Aquesta llei recull i adapta la normativa europea sobre protecció dels consumidors, incloent-hi directives específiques sobre la contractació a distància i les vendes fora d'establiment comercial.

- **Llei de Condicions Generals de Contractació [43]:** És una normativa espanyola que regula les clàusules i condicions establertes de manera unilateral per una de les parts en un contracte, habitualment per l'empresari. L'altra part (el consumidor o usuari) ha d'acceptar íntegrament si vol utilitzar el servei o adquirir el producte.

Aquesta llei té com a objectiu garantir la transparència i la protecció dels drets dels consumidors en els contractes que continguin condicions generals de contractació.

- **Text Refós de la Llei de Propietat Intel·lectual [44]:** És una normativa espanyola que recull i actualitza les disposicions legals relatives a la propietat intel·lectual. Aquesta llei regula els drets d'autor i altres drets connexos que protegeixen les creacions intel·lectuals, com ara programari. Aquesta llei estableix els drets morals i patrimonials dels creadors sobre les seves obres i defineix les condicions d'ús, explotació i protecció d'aquestes obres. Això significa que el desenvolupador té drets exclusius sobre l'ús, la distribució i la modificació de l'aplicació, impedint que terceres parts utilitzin o copiïn el codi sense permís.

9.2 Llicències

- **MIT [45]:** És una llicència de programari lliure permissiva creada per l'Institut Tecnològic de Massachusetts (MIT). La llicència permet reutilitzar el programari en programari propietari, sota altres llicències de programari lliure o sota la mateixa llicència MIT. (Programari llicenciat sota la llicència MIT: *Ionic Framework, Angular, Jenkins i VSCode*)

- **Apache 2.0 [46]:** És una llicència de programari lliure permissiva creada per la Apache Software Foundation (ASF). La llicència permet reutilitzar el programari en programari propietari, sota altres llicències de programari lliure o sota la mateixa llicència Apache. (Programari llicenciat sota la llicència Apache 2.0: *Sping Boot, IntelliJ IDEA i DBeaver*)

- **GPL [47]:** És una llicència de programari lliure creada per la Free Software Foundation. Permet copiar, distribuir i modificar el codi, sempre que les modificacions es distribueixin amb la mateixa llicència GPL. Aquesta llicència va ser dissenyada per Richard Stallman i el grup GNU. (Programari llicenciat sota la llicència GPL: *Git i MySQL*)

Informe de sostenibilitat

Després de fer l'enquesta del Projecte de investigació EDINSOST2-ODS, he tingut l'oportunitat de reflexionar sobre el meu coneixement en l'àmbit de la sostenibilitat enfocat a la creació de projectes.

Durant aquest anys del grau, una de les dimensions més treballades sobre la sostenibilitat ha sigut la econòmica. Això em permet tenir una consciència molt clara de la importància de l'estudi i la planificació de tot allò relacionat amb la gestió financera i la viabilitat d'un projecte.

Respecte a la dimensió ambiental, reconec la preocupació pels possibles efectes negatius dels components hardware. Tot i això, manco del coneixement necessari per a avaluarne l'impacte de manera precisa. No obstant, estic familiaritzada amb les pràctiques de reciclatge que poden ajudar a minimitzar aquest impacte negatiu.

Per acabar, pel que fa a la dimensió social, he començat a explorar com millorar l'accessibilitat en el software. Tot i que encara tinc un llarg camí per recórrer en aquest àmbit, progressivament estic adquirint una comprensió més profunda de l'ètica i l'accessibilitat en el desenvolupament.

10.1 Dimensió econòmica

PREGUNTES INICIALS

Reflexió sobre el cost que has estimat per la realització del projecte

En la nostra estimació de costos per a la realització del projecte, hem realitzat un pressupost exhaustiu tenint en compte tots els recursos necessaris per desenvolupar un core altament especialitzat i personalitzable. Hem dedicat temps a considerar els costos associats amb la creació i la implementació d'aquesta solució, assegurant-nos que estiguin alineats amb els objectius del projecte i les expectatives dels stakeholders.

Com es resolen actualment els aspectes de costos del problema que vols abordar (estat de l'art)?

Actualment, els aspectes econòmics del problema que volem abordar són generalment resolts amb solucions poc eficients. Les empreses o consultores es veuen forçades a destinar temps i recursos rellevants en el desenvolupament i gestió d'aplicacions de comerç electrònic personalitzades per a cada client. Mentre que, la nostra solució ofereix una nova alternativa coherent i eficient a aquest model, ja que permet una gestió centralitzada i configuració individualitzada, a la vegada que redueix el temps de desenvolupament i esforços duplicats.

En què millorarà econòmicament (costos...) la teva solució respecte a les existents?

La nostra solució proporciona una millora en termes de costos i eficiència. La gestió centralitzada i les eines de configuració permetran una major eficiència en la gestió dels projectes d'aplicacions de comerç electrònic, alhora que mantenen la flexibilitat necessària per satisfer les necessitats particulars de cada client. Això es traduirà en una millora econòmica per a les empreses i consultores que adoptin la nostra solució.

PREGUNTES FINALS

Has quantificat el cost (recursos humans i materials) de la realització del projecte? Quines desicions has pres per reduir-ne el cost? Has quantificat aquest estalvi?

Sí, s'ha quantificat el cost de la realització del projecte considerant els recursos humans i materials. A més, també s'inclou un estudi dels imprevistos i se'n reserva una part a contingències.
Per tal de reduir els costos, s'han fet servir en la mesura del possible versions gratuïtes de les eines programari utilitzades. Tot i així, s'ha utilitzat el model OpenAI que suposa un cost molt reduït actualment ja que la quantitat de crides és molt limitada però en un futur, aquest cost està previst eliminar-se entrenant un model propi com Ollama per evitar un augment molt gran del cost. Aquest entrenament suposaria uns costos que a llarg termini es veurien clarament compensats.

S'ha ajustat el cost previst al cost final? Has justificat les diferències?

El cost final del projecte s'ha ajustat bastant bé al cost inicial, de fet, el pressupost s'ha reduit una quantitat molt poc significativa ja que algunes hores de documentació s'han dedicat a implementació de codi on el cost per hora del programador júnior és inferior al cost del cap de projecte que redacta la documentació del projecte. Les diferències que s'han creat es deuen principalment a imprevistos en la integració de certes tecnologies i la

necessitat d'hores extres per implementar certes funcionalitats. Hem justificat aquestes diferències detallant cada imprevist i les solucions aplicades per minimitzar l'impacte econòmic. A més a més, com el cost destinat a les contingències no s'ha hagut de gastar, suposaria que es reduis el cost significativament.

Quin cost estimes que tindrà el projecte durant la seva vida útil? Es podria reduir aquest cost per fer-lo més viable?

Els únics costos del projecte durant la vida útil serà el cost de manteniment i el cost d'infraestructura.

S'ha tingut en compte el cost dels adjustaments/actualitzacions/reparacions durant la vida útil del projecte?

El cost de manteniment del projecte s'ha identificat com el principal cost durant la vida útil del projecte. Estimar aquest cost és complicat perquè depèn de la freqüència i la complexitat de les incidències i les peticions de millora. Una manera de reduir aquest cost és minimitzant les incidències, assegurant que la solució sigui molt estable i realitzant proves exhaustives abans de cada desplegament.

Podrien produir-se escenaris que perjudiquessin la viabilitat del projecte?

L'únic escenari que podria perjudicar la viabilitat del projecte és que l'eina no aporti el valor esperat i, per tant, no es pugui amortitzar el cost de desenvolupament. A més, altres factors com canvis significatius en la tecnologia o el mercat, increments inesperats en els costos de manteniment, o problemes de seguretat greus que requereixin solucions costoses també podrien afectar negativament la viabilitat del projecte.

10.2 Dimensió ambiental

PREGUNTES INICIALS

Has estimat l'impacte ambiental que tindrà la realització del projecte?

L'impacte ambiental que pot tenir aquest projecte es basa principalment en el consum d'electricitat a l'hora de desenvolupar i utilitzar l'aplicació mòbil. A part d'aquest impacte, no hi ha cap altre recurs que influeixi negativament en l'aspecte ambiental.

T'has plantejat minimitzar-ne l'impacte, per exemple, reutilitzant recursos?

Tots els aparells que es faran servir per al seu desenvolupament, són dispositius ja amortitzats evitant així haver de comprar-ne de nous. L'únic punt on es podria millorar és en

l'ús d'una energia renovable i l'aprofitament màxim de la llum solar per evitar encendre la llum artificial.

Com es resol actualment el problema que vols abordar (estat de l'art)? En què millorarà ambientalment la teva solució respecte a les existents?

En l'estat de l'art actual, el desenvolupament d'aplicacions mòbils per al comerç electrònic sovint es caracteritza per una manca d'optimització en l'ús de recursos i temps. Les pràctiques existents sovint inclouen el desenvolupament d'aplicacions des de zero per a cada client, una tasca que pot ser costosa i, per tant, consumir molt temps i energia.

En l'àmbit de l'aspecte ambiental, la nostra solució destaca per la seva contribució a la reducció de l'impacte ecològic. Mitjançant l'ús d'un core altament especialitzat i personalitzable, eliminem la necessitat de crear cada aplicació des de zero. Aquest enfocament permet reduir significativament el consum de recursos materials i energètics associats al desenvolupament d'aplicacions.

PREGUNTES FINALS

Has quantificat l'impacte ambiental de la realització del projecte?Quines mesures has pres per reduir l'impacte? Has quantificat aquesta reducció?

Com ja s'ha explicat a les preguntes anteriors, l'únic impacte ambiental considerable és el consum elèctric. No s'ha realitzat cap estudi extra en particular per reduir-lo ja que es considera que la forma de treballar de ja és respectuosa amb el medi ambient.

Si fessis el projecte de nou, podries realitzar-lo amb menys recursos?

Es considera que no es poden reduir més els recursos ambientals ja que no s'han fet despesses innecessàries en aquest àmbit.

Quins recursos estimes que s'utilitzaran durant la vida útil del projecte? Quin serà l'impacte ambiental d'aquests recursos?

Els recursos que s'utilitzaran durant la vida útil del projecte són els mateixos que s'han utilitzat durant el desenvolupament, ja que no es preveu l'adquisició de cap recurs nou. Per tant, l'impacte ambiental d'aquests recursos és el mateix que l'impacte ambiental dels recursos utilitzats durant el desenvolupament del projecte.

El projecte permitirà reduir l'ús d'altres recursos? Globalment, l'ús del projecte millorarà o empijorarà la petjada ecològica?

Tal com s'ha explicat a les preguntes inicials, el projecte desenvolupat reduirà considerablement el consum de recursos materials i energètics associats al desenvolupament d'aplicacions ja que serà molt més ràpid i senzill crear una aplicació i no serà necessari

destinar una quantitat tant gran d'hores i materials.
Com que durant la vida útil del projecte no es preveu cap impacte ambiental negatiu, es considera que l'ús de l'eina creada per a aquest TFG millorarà l'empremta ecològica.

Podrien produir-se escenaris que fessin augmentar la petjada ecològica del projecte?

Si el projecte s'usa àmpliament, això comportarà un increment en el consum de recursos dels servidors. Tot i així, aquest escenari no es veu com un aspecte negatiu, ja que l'ús extensiu de l'eina indica que està aportant un gran valor, i per tant, es considera que l'impacte ambiental resultant és positiu.

10.3 Dimensió social

PREGUNTES INICIALS

Què creus que t'aportarà a nivell personal la realització d'aquest projecte?

Realitzar aquest projecte m'aportarà un gran creixement professional i personal. Em permetrà desenvolupar habilitats tècniques en el desenvolupament d'aplicacions mòbils. També em donarà la satisfacció de contribuir a la creació d'una solució innovadora que pot millorar l'eficiència de la creació d'aplicacions mòbils e-commerce.

Com es resol actualment el problema que vols abordar (estat de l'art)? En què millorarà socialment (qualitat de vida) la teva solució respecte les existents?

Actualment, les empreses i consultores que gestionen projectes d'aplicacions mòbils s'enfronten a reptes relacionats amb la complexitat i la diversitat de les necessitats dels clients en l'àmbit del comerç electrònic. Per abordar aquests reptes, les empreses recorren a diverses solucions com ara desenvolupar aplicacions des de zero per a cada client o a partir d'un repositori de codi amb funcionalitats bàsiques, escollir aquelles que són necessàries per a l'aplicació i reutilitzar-les, fet que comporta moltes duplicacions d'esforços i costos.

La nostra solució millorarà la qualitat de vida de consultores i empreses al proporcionar-los una eina eficient i personalitzable per al desenvolupament d'aplicacions mòbils. Això permetrà un desenvolupament més ràpid i coordinat que permetrà gestionar d'una forma més eficient les necessitats del client.

Existeix una necessitat real del projecte?

Sí. El desenvolupament d'una aplicació mòbil que ofereixi un core altament especialitzat

i personalitzable per al comerç electrònic respon a una demanda creixent en el mercat. Les empreses i consultores que gestionen múltiples clients amb necessitats similars en l'àmbit del comerç electrònic necessiten eines eficaces per optimitzar el desenvolupament i la gestió dels seus projectes.

PREGUNTES FINALS

La realització d'aquest projecte ha implicat reflexions significatives a nivell personal, professional o ètic de les persones que han intervingut?

Sí, la realització d'aquest projecte ha implicat reflexions significatives a nivell personal i professional. L'autor ha desenvolupat habilitats tècniques i ha après a gestionar un projecte des de zero. A nivell ètic, s'ha considerat la importància de crear una solució que no només sigui eficient, sinó també respectuosa amb l'ambient i beneficiosa per a la societat.

Qui es beneficiarà de l'ús del projecte? Hi ha algun colectiu que pugui veure's perjudicat pel projecte? Amb quina mesura?

Els principals beneficiaris del projecte seran les empreses i consultores que gestionen projectes d'aplicacions mòbils, ja que la nostra solució els permetrà optimitzar el desenvolupament i la gestió dels seus projectes. No es preveu que cap col·lectiu es vegi perjudicat directament pel projecte. No obstant això, si no es gestionen adequadament les dades dels usuaris, podria haver-hi preocupacions sobre la privacitat i la seguretat.

Amb quina mesura soluciona el projecte el problema plantejat inicialment?

El projecte soluciona el problema plantejat inicialment proporcionant una plataforma centralitzada i personalitzable per al desenvolupament d'aplicacions de comerç electrònic. Això redueix significativament el temps i els costos associats al desenvolupament de noves aplicacions i permet una gestió més eficient de les necessitats dels clients.

Podrien produir-se escenaris que fessin que el projecte fossi perjudicial per algun segment particular de la població?

No es preveu que el projecte sigui perjudicial per a cap segment particular de la població. No obstant això, cal assegurar que la plataforma compleixi amb totes les normatives de privacitat i seguretat de dades per evitar possibles perjudicis als usuaris finals.

Podria crear el projecte algun tipus de dependència que deixi als usuaris en posició de debilitat?

Treball futur

11.1 Implementació

11.1.1 Ampliació de l'èpica de receptes

Actualment, la funcionalitat implementada per buscar els ingredients i els seus productes relacionats a partir del nom d'una recepta es limita bàsicament a obtenir un llistat d'ingredients sense calcular les quantitats necessàries i mostrar els productes existents de la base de dades relacionats amb aquests ingredients. En un futur, es podria ampliar aquesta funcionalitat utilitzant una query que permeti no només identificar els ingredients, sinó també les seves quantitats específiques.

Per exemple, suposem que tenim una recepta per fer un pastís de xocolata que requereix 1kg de farina, 900g de sucre i 500g de xocolata. Es podria consultar la base de dades per obtenir informació sobre els productes disponibles i les seves mides d'empaquetament. En aquest cas, si la base de dades conté farina en paquets de 500g, sucre en paquets de 1kg i xocolata en barres de 200g, l'algorisme podria calcular que necessites:

- **Farina: 1kg** - 2 paquets de 500g.
- **Sucre: 900g** - 1 paquet de 1kg (sobrarien 100 grams però no es pot comprar menys d'un paquet).
- **Xocolata: 500g** - 3 barres de 200g (cada barra cobreix 200 grams, per tant, es necessitarien 2.5 barres, però s'han de comprar 3).

En resum, l'ampliació de la funcionalitat permetria una recerca més detallada i precisa dels ingredients i les seves quantitats necessàries, així com una millor adaptació als productes disponibles a la base de dades. Això milloraria l'experiència de l'usuari en planificar i preparar receptes, assegurant que tinguin els ingredients adequats en les quantitats correctes.

11.1.2 Ampliació de funcionalitats bàsiques

Una aplicació mòbil e-commerce és un projecte molt ampli que permet desenvolupar moltes funcionalitats addicionals a les que ja s'han implementat. A continuació, es llisten algunes de les principals funcionalitats que es podrien treballar en un futur, a més a més de la comentada en l'apartat anterior que per limitacions de temps d'aquest projecte no s'han pogut abastar però serien convenients si en un futur es vol continuar desenvolupant aquest projecte.

- Poder buscar productes i filtrar-los per marques o altres característiques.
- Poder afegir una direcció d'entrega en fer una comanda.
- Poder modificar les dades addicionals de l'usuari com el nom i cognoms, número de telèfon, data de naixement i idioma.
- Rebre un correu electrònic amb la confirmació de la comanda.
- Poder tancar la sessió, mantenir la sessió iniciada i augmentar la seguretat de la sessió.

11.1.3 Ampliació de la personalització de l'aplicació per al client

Actualment, s'ha creat un projecte client que utilitza el core desenvolupat en aquest treball i s'ha personalitzat els colors primaris de l'aplicació i les imatges com el logotip. A més, tal com s'ha plantejat el nou projecte, seria també possible personalitzar fàcilment els labels. Una de les futures millores seria estendre les possibilitats de configuració i personalització de les funcionalitats mostrades l'usuari o inclús l'aspecte de la interfície. Per les limitacions de temps del projecte no s'ha pogut fer, però caldria crear uns paràmetres a la configuració com podria ser "showBithdayDateOnProfile=false" i fer ús d'aquest paràmetre a l'hora de dissenyar la pantalla del perfil.

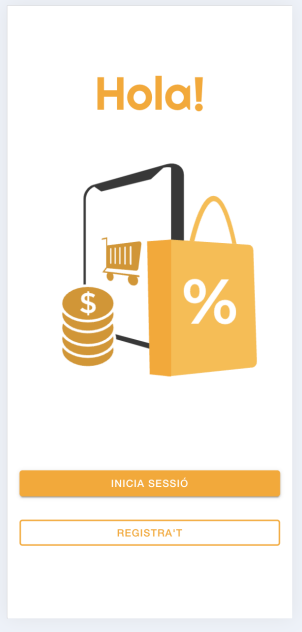

Figura 11.1: Pantalla de benvinguda de l'aplicació creada a través del core desenvolupat. (Font pròpia)

11.2 Canvi a un model de llenguatge natural local

Com s'ha explicat a l'anàlisi d'alternatives de tecnologies, podria ser recomanable utilitzar un model de llenguatge natural local com Ollama, en lloc de l'OpenAI ChatGPT, per millorar la seguretat i privadesa de l'aplicació i també reduir els costos. Per fer aquesta millora, només caldria modificar la lògica de la crida que accedeix a l'API externa de ChatGPT cridant al seu lloc al servei local que hàgim preparat.

11.2.1 Entrenament d'un model de llenguatge natural especialitzat en receptes culinàries

Per millorar la funcionalitat de receptes de la nostra aplicació seria recomanable entrenar un model de llenguatge natural especialitzat en receptes culinàries. Aquest procés inclouria la recopilació de dades mitjançant web scraping, així com el fine-tuning del model per assegurar que comprengui i generi receptes de manera efectiva.

El primer pas per entrenar el model és obtenir una gran quantitat de dades de receptes. Això es pot aconseguir mitjançant tècniques de *web scraping*, que consisteixen a extreure informació de llocs web de receptes populars. Aquestes dades es guardaran per ser utilitzades en la fase d'entrenament del model.

Una vegada recopilades les dades, es procedirà a l'entrenament del model de llenguatge natural. En lloc de crear un model des de zero, es pot utilitzar un model preentrenat, com ara Ollama Llama3, que ja ha estat entrenat amb una gran quantitat de text general. Aquest model preentrenat ja entén com generar text coherent i gramaticalment correcte en diversos contextos. Per especialitzar-lo en receptes, es farà un ajustament addicional (fine-tuning) amb les dades específiques de receptes. Aquest procés permet al model aprendre el vocabulari i els patrons específics del domini culinari.

Després de l'entrenament, es validaran els resultats per assegurar que el model genera receptes de qualitat i útils. Es faran ajustaments segons sigui necessari per corregir errors i millorar la precisió.

Aquesta estratègia permetrà desenvolupar un model de llenguatge natural robust i especialitzat en receptes culinàries.

Conclusions

12.1 Competències tècniques treballades

- **CES1.1: Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics. [Bastant]:** S'ha desenvolupat amb èxit una aplicació mòbil amb Ionic que es comunica amb una API Rest i aquesta amb una base de dades, oferint un core personalitzable i escalable.
- **CES1.2: Donar solució a problemes d'integració en funció de les estratègies, dels estàndards i de les tecnologies disponibles. [En profunditat]:** S'han integrat tecnologies com Ionic i Spring Boot, utilitzant Jenkins per a la integració continua facilitant la gestió eficient de les versions del codi i automatització dels tests.
- **CES1.3: Identificar, avaluar i gestionar els riscos potencials associats a la construcció de software que es poguessin presentar. [Bastant]:** S'han identificat i avaluat els principals possibles riscos associats al desenvolupament d'aquest projecte. Aquest procés ha inclòs un anàlisi detallat de les possibles amenaces i la seva probabilitat d'ocurrència, implementant estratègies de minimització adequades. Si han sorgit riscos durant el desenvolupament, s'han gestionat eficaçment segons el pla establert, minimitzant així la pèrdua de temps i recursos.
- **CES1.4: Desenvolupar, mantenir i avaluar serveis i aplicacions distribuïdes amb suport de xarxa. [Una mica]:** S'ha iniciat la implementació de funcions de xarxa bàsiques, utilitzant comunicacions API Rest per connectar l'aplicació mòbil amb el servidor, però encara falta profunditzar en aspectes més avançats de sistemes distribuïts i suport de xarxa.
- **CES1.5: Especificar, dissenyar, implementar i avaluar bases de dades. [Bastant]:** S'han especificat i implementat esquemes de bases de dades eficaços per gestionar les dades de l'aplicació mòbil, utilitzant tecnologies com MySQL per assegurar la integritat i l'optimització dels accessos a les dades.
- **CES1.6: Administrar bases de dades (CIS4.3) [Bastant]:** L'administració de les bases de dades s'ha gestionat de manera efectiva, aplicant Liquibase per a la

gestió de versions, facilitant l'actualització i manteniment dels esquemes de manera controlada.

- **CES1.7: Controlar la qualitat i dissenyar proves en la producció de software.** [**Bastant**]**:** S'han fet tests unitaris per a les funcionalitats del backend i proves manuals al final del desenvolupament de cada èpica per comprovar el correcte funcionament de la part frontal. A més a més, s'ha integrat l'ús de Jenkins per a la integració contínua, permetent la realització automàtica de proves i assegurant que qualsevol canvi en el codi mantingui la qualitat i funcionalitat del sistema.
- **CES1.8: Desenvolupar, mantenir i avaluar sistemes de control i de temps real.** [**Una mica**]**:** En aquest projecte d'aplicació mòbil d'e-commerce, l'ús de sistemes de control i de temps real es limita principalment a la gestió d'inventari en temps real. Aquesta funcionalitat és essencial per actualitzar l'estat de l'estoc i evitar incidències com les sobre vendes.
- **CES1.9: Demostrar comprensió en la gestió i govern dels sistemes software.** [**En profunditat**]**:** Durant el desenvolupament de l'aplicació mòbil d'e-commerce, s'ha demostrat una profunda comprensió en la gestió i govern dels sistemes software. Això inclou la implementació de metodologies àgils per a la gestió del projecte i assegurar el compliment dels requisits funcionals i no funcionals.
- **CES2.1: Definir i gestionar els requisits d'un sistema software.** [**Bastant**]**:** A l'inici del projecte, s'han especificat uns requisits funcionals i no funcionals que s'han anat aconseguint al llarg del desenvolupament.
- **CES2.2: Dissenyar solucions apropiades en un o més dominis d'aplicació, utilitzant mètodes d'enginyeria del software que integrin aspectes ètics, socials, legals i econòmics.** [**Bastant**]**:** S'han dissenyat solucions software que no només satisfan les necessitats funcionals sinó que també consideren els impactes ètics, socials i legals, incorporant normatives de privacitat de dades i accessibilitat.
- **CES3.1: Desenvolupar serveis i aplicacions multimèdia.** [**Una mica**]**:** S'han introduït elements multimèdia bàsics a l'aplicació, com la gestió i visualització d'imatges, però aquest aspecte encara necessita un desenvolupament més profund i especialitzat.
- **CES3.2: Dissenyar i gestionar un magatzem de dades (data warehouse).** [~~**Bastant**~~ **Res**]

12.2 Relació del TFG amb el grau i l'especialitat

Durant el desenvolupament d'aquest projecte, he tingut l'oportunitat d'aplicar una àmplia gamma de coneixements adquirits al llarg dels últims quatre anys en les assignatures del

Grau en Enginyeria Informàtica. Aquest apartat detalla els coneixements tècnics integrats que han estat aplicats en el desenvolupament de la nostra aplicació mòbil d'e-commerce.

L'assignatura Gestió de Projectes de Software (GPS) ha estat fonamental per a la realització de la primera fase del projecte, on es va abordar la planificació del projecte i es va realitzar el diagrama de Gantt. Aquesta assignatura també ha sigut essencial per a l'elecció de la metodologia que es volia seguir per completar el projecte.

Per seguir amb l'especificació dels requisits, cal destacar la importància de l'assignatura Enginyeria de Requisits (ER) que va proporcionar les bases per identificar i descriure els requisits i històries d'usuari de manera efectiva. Així mateix, la Introducció a l'Enginyeria del Software (IES) i l'Arquitectura del Software (AS) van enriquir els meus coneixements sobre els models conceptuals i diagrames de seqüència, essencials per al disseny del projecte.

Pel que fa al desenvolupament de l'aplicació mòbil, l'assignatura Projecte d'Enginyeria de Software (PES) va ser una de les assignatures que ens va introduir al desenvolupament d'aplicacions mòbils i una de les principals motivacions per realitzar una aplicació mòbil com a projecte de final de grau. A més a més, les pràctiques d'empresa es van realitzar a una consultora tecnològica que gestionava diferents aplicacions web e-commerce, fet que va permetre introduir-se en aquesta indústria i aprendre nous coneixements com ara Angular. Amb aquests coneixemenets previs d'Angular i una petita introducció al framework Ionic mitjançant un curs d'OpenWebinars, es va decidir tirar endavant la part frontal del nostre projecte. Tot i això, iniciar un projecte de manera individual permet expandir els coneixements i l'experiència necessària per oferir una bona solució. A més a més, el disseny de prototips i la integració de noves biblioteques són tasques completament lligades al desenvolupament de frontend que aquest projecte ha permès millorar.

En l'àmbit del backend, els coneixements de Java obtinguts en l'assignatura de Programació Orientada a Objectes (PROP) i els conceptes d'API REST explorats en Arquitectura de Software (ASW) van ser fonamentals. El desenvolupament d'un projecte Spring Boot des de zero per implementar l'API REST ha permès aprofundir en aquestes tecnologies. A més, l'assignatura de Bases de Dades (BD) va facilitar la gestió eficient de les dades mitjançant Spring Data JPA i SQL.

Finalment, per tal de poder utilitzar eines de CI/CD com Jenkins, s'ha hagut d'aprendre sobre automatització de la integració i desplegament continu, una habilitat que, encara que no ha estat aprofundida en cap assignatura específica, durant les pràctiques d'empresa, es va tenir l'oportunitat d'aprendre la importància d'aquesta integració en els projectes de software i és per això que s'ha decidit implementar-ho també en aquest projecte de final de grau.

12.3 Conclusions personals

La realització d'aquest projecte ha estat una experiència única que m'ha permès profunditzar en molts dels coneixements adquirits al llarg del grau. Ha sigut un repte personal, ja que, al llarg de la carrera, estem acostumats a seguir unes directrius ben marcades per desenvolupar els projectes i durant la meva experiència professional, he treballat en projectes que ja tenen un funcionament establert per al bon rendiment de l'empresa. En canvi, el desenvolupament d'aquest projecte des de zero ha sigut més difícil perquè no hi havia un camí predeterminat ni un model a seguir, això ha suposat haver de prendre moltes decisions de planificació i implementació de manera gairebé independent. D'aquesta manera he après a aplicar tots els meus coneixements de forma pràctica i a buscar solucions i alternatives a totes aquelles solucions presentades al llarg del desenvolupament. Això no ha sigut fàcil, però el resultat final ha estat enormement gratificant i és un projecte amb el qual m'agradaria continuar treballant en un futur.

Per altra banda, a més a més de millorar les meves capacitats decisives i d'organització, aquest projecte també m'ha permès adquirir molts més coneixements tècnics sobre diferents tecnologies i llenguatges de programació. Això ha enriquit la meva comprensió global del desenvolupament de software i ha ampliat el meu repertori d'habilitats tècniques, preparant-me per a futurs projectes més ambiciosos.

En conclusió, aquest projecte no només ha estat un test de les meves habilitats tècniques sinó que també ha estat una prova del meu creixement personal. M'ha preparat millor per als desafiaments del món real i m'ha motivat a continuar aprenent i millorant com a professional en el camp de l'enginyeria del software.

Referències

[1] *Evolución trimestral del volumen de comercio electrónico*. Comisión Nacional de los Mercados y la Competencia
URL: `https://shorturl.at/lpqCW`
(17 de gener de 2024)

[2] *eCommerce - Worldwide*. Statista
URL: `https://shorturl.at/cgntX`
(16 de març de 2024)

[3] *The Impact of COVID-19 on mobile app usage and Development*. Medium
URL: `https://shorturl.at/arBM0`
(16 de març de 2024)

[4] *¿Cuánto tiempo se tarda en crear una aplicación?*. Redwerk
URL: `https://shorturl.at/lK012`
(20 de febrer de 2024)

[5] *Cuánto tiempo se tarda el desarrollo una aplicación móvil*. ZOOM. Marketing digital
URL: `https://shorturl.at/gklxM`
(20 de febrer de 2024)

[6] *La tecnología es nuestro "core"*. Aktios
URL: `https://shorturl.at/mnqCJ`
(25 de gener de 2024)

[7] *Soluciones m-commerce*. 480-Cuatroochenta
URL: `https://shorturl.at/cdiky`
(7 de febrer de 2024)

[8] *Ionic Framework*. Ionic
URL: `https://shorturl.at/bFK48`
(20 de febrer de 2024)

[9] *Angular Framework*. Angular
URL: `https://shorturl.at/4HBwx`
(20 de febrer de 2024)

[10] *Spring Boot Framework*. Spring Boot
URL: https://shorturl.at/qsLO1
(20 de febrer de 2024)

[11] *Plantilla d'especificació de requisits de Volere*. Volere Requirements Resources
URL: https://shorturl.at/NUV24
(20 de febrer de 2024)

[12] *Las metodologías ágiles*. IEBS
URL: https://shorturl.at/juCE9
(23 de febrer de 2024)

[13] *What is Scrum?*. Scrum.org
URL: https://shorturl.at/osvzM
(23 de febrer de 2024)

[14] *What is Kanban?*. Asana
URL: https://shorturl.at/sCDKP
(23 de febrer de 2024)

[15] *Jira*
URL: https://shorturl.at/tSwJ9
(13 de juny de 2024)

[16] *GitHub*
URL: https://shorturl.at/t997B
(13 de juny de 2024)

[17] *Jenkins*
URL: https://shorturl.at/YBn4n
(13 de juny de 2024)

[18] *GitFlow*
URL: https://shorturl.at/qrRhd
(13 de juny de 2024)

[19] *Git*
URL: hhttps://shorturl.at/XzmSw
(13 de juny de 2024)

[20] *Confluence*
URL: https://shorturl.at/9t4xE
(13 de juny de 2024)

[21] *Overleaf*
URL: https://shorturl.at/cB8ER
(13 de juny de 2024)

[22] *Cap de projectes informàtics*. Barcelona Activa
URL: https://shorturl.at/cexR1
(6 de març de 2024)

[23] *Cuánto gana un programador en España en 2024*. Hack a boss
URL: https://shorturl.at/ktGO0
(6 de març de 2024)

[24] *¿Cuánto se gana como uno Tester/a en España?*. Indeed
URL: https://shorturl.at/jwGH6
(6 de març de 2024)

[25] *Diseñador UX UI: ¿Cuál es su función y cuánto gana?*. CEI
URL: https://shorturl.at/fov08
(6 de març de 2024)

[26] *Coste de un trabajador para la empresa*. Factorial
URL: https://shorturl.at/LPV12
(6 de març de 2024)

[27] *¿Cuánto cuesta subir una app a Appstore y Play Store?*. ZOOM. Marketing digital
URL: https://shorturl.at/rtCGX
(6 de març de 2024)

[28] *MSI Modern 14*. PC Components
URL: https://shorturl.at/jwNR2
(6 de març de 2024)

[29] *MacBook Pro*. Apple
URL: https://shorturl.at/cKWZ9
(6 de març de 2024)

[30] *Xiaomi Redmi 10A*. Amazon
URL: https://shorturl.at/acmrJ
(6 de març de 2024)

[31] *iPhone 13*. Apple
URL: https://shorturl.at/hnOR3
(6 de març de 2024)

[32] *KLIM Chroma Teclado*. Amazon
URL: https://shorturl.at/hLSY7
(6 de març de 2024)

[33] *VIVIJO Ratón Gaming*. Amazon
URL: https://shorturl.at/etDG8
(6 de març de 2024)

[34] *Monitor ThinkVision*. Lenovo
URL: https://shorturl.at/awET4
(6 de març de 2024)

[35] *Coworking – Plans Mensuals*. Aurea coworking
URL: https://shorturl.at/lqwZ3
(6 de març de 2024)

[36] *Aplicando el patrón de diseño MVVM*. Medium
URL: https://shorturl.at/ssTxl
(13 de juny de 2024)

[37] *Model–view–controller*. Wikipèdia URL: https://shorturl.at/KJYH1
(13 de juny de 2024)

[38] *MVC Architecture and MVC in Odoo ERP*. LinkedIn URL: https://shorturl.at/4ua2y
(13 de juny de 2024)

[39] *Guía legal para lanzar una aplicación móvil*. MCT Abogados
URL: https://shorturl.at/EbPU6
(16 de maig de 2024)

[40] *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.* BOE
URL: https://shorturl.at/qQypH
(16 de maig de 2024)

[41] *Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico.* BOE
URL: https://shorturl.at/POFC7
(16 de maig de 2024)

[42] *Real Decreto Legislativo 1/2007, de 16 de noviembre, por el que se aprueba el texto refundido de la Ley General para la Defensa de los Consumidores y Usuarios y otras leyes complementarias.* BOE
URL: https://shorturl.at/NWb54
(16 de maig de 2024)

[43] *Ley 7/1998, de 13 de abril, sobre condiciones generales de la contratación.* BOE
URL: https://shorturl.at/MH2j5
(16 de maig de 2024)

[44] *Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia.* BOE

URL: https://shorturl.at/51Dhg
(16 de maig de 2024)

[45] *Licencia MIT*. Wikipedia
URL: https://shorturl.at/NZ42p
(15 de maig de 2024)

[46] *Apache License*. Wikipedia
URL: https://shorturl.at/RMhwH
(15 de maig de 2024)

[47] *GNU General Public License*. Wikipedia
URL: https://ca.wikipedia.org/wiki/GNU_General_Public_License
(15 de maig de 2024)

Glossari

API (Interfície de Programació d'Aplicacions) Mecanisme que permet a dos components de *software* comunicar-se entre si mitjançant un conjunt de definicions i protocols. 42

Bugs Error o defecte que produeix un comporament no desitjat o incorrecte. 17

Core En el context de *Software*, és la part central i fonamental d'un sitema o aplicació. 13

E-commerce És la pràctica de comprar i vendre productes a través d'Internet. 13

Framework Conjunt d'eines, biblioteques i components predefinits per facilitar el desenvolupament d'aplicacions. 42

Product Backlog És una llista ordenada de totes les tasques que es faran per desenvolupar un producte. 18

Sprint És un període breu de temps fix en què un equip treballa per completar una quantitat de treball establerta. 18

Sprint planning És una reunió que es fa al començament de cada Sprint on participa l'equip Scrum al complet. Serveix per inspeccionar el *Product Backlog* i que l'equip de desenvolupament seleccioni les tasques que treballarà durant el següent *sprint*. 18

Sprint retrospective És un tipus de reunió on l'equip reflexiona sobre què ha sortit bé i què es podria millorar en el proper *sprint*.. 18

Sprint review És una reunió on es demostra i s'avalua la feina de l'equip durant l'*sprint*.. 18

Anexos

A.1 Hores treballades per rols (Gestió econòmica)

Tasca	Duració	Cap del projecte	Programador Junior	Tester	Dissenyador UI/UX
Contextualització i abast	24	24			
Planificació temporal	16	16			
Gestió econòmica i sostenibilitat	16	16			
Document final	16	16			
[FRONT] - Creació projecte Ionic	2		2		
[FRONT] - Implementació "multilanguage" (i18n)	2		2		
[BACK] - Creació projecte Spring Boot	2		2		
[BBDD] - Creació base de dades	2		2		
[BACK] - Configuració CORS + Base de dades	2		2		
[LIQUIBASE] - Integració liquibase	4		4		
[CI] - Integració Jenkins	4		4		
Model conceptual de dades	8	8			
Definició de les tasques i èpiques detallades	8	8			
[LIQUIBASE] - Creació de taules	2		2		
[BACK] - Product controller	2		2		
[INFO] - Gestió d'imatges dels productes	4		4		
[BACK] - Gestió d'imatges	12		12		
[BACK] - Endpoints	20		20		
[DISSENY UI/UX] - Disseny grid + detall	4				4
[FRONT] - Grid de productes i detall del producte	8		8		
[DISSENY UI/UX] - Disseny barra menú	2				2
[FRONT] - Barra de menú + navegació	4		4		
Sprint Review	2	1		1	
Sprint Retrospective	2	2			
Documentació	8	8			
[LIQUIBASE] - Creació de taules	2		2		
[BACK] - Cart controller	2		2		
[INFO] - Sistemes de pagament i gestió de la cistella	2		2		
[BACK] - Sistema de pagament Redsys	16		16		
[BACK] - Endpoints	12		12		
[DISSENY UI/UX] - Disseny cistella + pantalla pagament	2				2
[FRONT] - Cistella de la compra + pantalla pagament	12		12		
Gestió de bugs	10		10		
Sprint Review	2	1		1	
Sprint Retrospective	2	2			
Documentació	8	8			

Tasca	Duració	Cap del projecte	Programador Junior	Tester	Dissenyador UI/UX
[INFO] - Integració IA	4		4		
[BACK] - Controller	2		2		
[BACK] - Endpoints	24		24		
[DISSENY UI/UX] - Disseny pantalla receptes	2				2
[FRONT] - Receptes	16		16		
Gestió de bugs	10		10		
Sprint Review	2	1		1	
Sprint Retrospective	2	2			
Documentació	8	8			
[LIQUIBASE] - Creació de taules	2		2		
[INFO] - Registre + Inici de sessió	2		2		
[BACK] - User controller	2		2		
[BACK] - Endpoints	16		16		
[DISSENY UI/UX] - Disseny perfil usuari	2				2
[FRONT] - Perfil	4		4		
[LIQUIBASE] - Creació de taules	2		2		
[BACK] - Endpoints	8		8		
[DISSENY UI/UX] - Disseny favorits	2				2
[FRONT] - Favorits	4		4		
Gestió de bugs	10		10		
Sprint Review	2	1		1	
Sprint Retrospective	2	2			
Documentació	8	8			
[BACK] - Filtratge de productes	8		8		
[BACK] - Recomanacions personalitzades	16		16		
[DISSENY UI/UX] - Disseny pantalla búsqueda	2				2
[FRONT] - Pantalla búsqueda	16		16		
Gestió de bugs	10		10		
Sprint Review	2	1		1	
Sprint Retrospective	2	2			
Documentació	8	8			
Testing i refactoring	24		12	12	
Desplegar aplicació	16		8	8	
Memòria final	30	30			
Presentació final	24	24			
TOTAL	**540**	**197**	**302**	**25**	**16**

Figura A.1: Hores treballades per rols. (Font pròpia)

Tasca	Duració	Cap del projecte	Programador Junior	Tester	Dissenyador UI/UX	Tasca	Duració	Cap del projecte	Programador Junior	Tester	Dissenyador UI/UX
Contextualització i abast	24	24				[INFO] - Integració IA	4		4		
Planificació temporal	16	16				[BACK] - Controller	2		2		
Gestió econòmica i sostenibilitat	16	16				[BACK] - Endpoints	24		24		
Document final	16	16				[DISSENY UI/UX] - Disseny pantalla receptes	2				2
[FRONT] - Creació projecte Ionic	2		2			[FRONT] - Receptes	16		16		
[FRONT] - Implementació "multilanguage" (i18n)	2		2			Gestió de bugs	8		8		
[BACK] - Creació projecte Spring Boot	2		2			Sprint Review	2	1		1	
[BBDD] - Creació base de dades	2		2			Sprint Retrospective	2	2			
[BACK] - Configuració CORS + Base de dades	2		2			Documentació	10	10			
[LIQUIBASE] - Integració liquibase	4		4			[LIQUIBASE] - Creació de taules	2		2		
[CI] - Integració Jenkins	4		4			[INFO] - Registre + Inici de sessió	2		2		
Model conceptual de dades	8	8				[BACK] - User controller	2		2		
Definició de les tasques i èpiques detallades	8	8				[BACK] - Endpoints	16		16		
[LIQUIBASE] - Creació de taules	2		2			[DISSENY UI/UX] - Disseny perfil usuari	2				2
[BACK] - Product controller	2		2			[FRONT] - Perfil	16		16		
[INFO] - Gestió d'imatges dels productes	4		4			[LIQUIBASE] - Creació de taules	2		2		
[BACK] - Gestió d'imatges	12		12			[BACK] - Endpoints	18		18		
[BACK] - Endpoints	20		20			[DISSENY UI/UX] - Disseny favorits	2				2
[DISSENY UI/UX] - Disseny grid + detall	4				4	[FRONT] - Favorits	24		24		
[FRONT] - Grid de productes i detall del producte	8		8			Gestió de bugs	10		10		
[DISSENY UI/UX] - Disseny barra menú	2				2	Sprint Review	2	1		1	
[FRONT] - Barra de menú + navegació	4		4			Sprint Retrospective	2	2			
Sprint Review	2	1		1		Documentació	8	8			
Sprint Retrospective	2	2				~~[BACK] - Filtratge de productes~~	~~8~~		~~8~~		
Documentació	8	8				~~[BACK] - Recomanacions personalitzades~~	~~16~~		~~16~~		
[LIQUIBASE] - Creació de taules	2		2			~~[DISSENY UI/UX] - Disseny pantalla búsqueda~~	~~2~~				~~2~~
[BACK] - Cart controller	2		2			~~[FRONT] - Pantalla búsqueda~~	~~16~~		~~16~~		
[INFO] - Sistemes de pagament i gestió de la cistella	8		8			Gestió de bugs	10		10		
[BACK] - Sistema de pagament	24		24			Sprint Review	2	1		1	
[BACK] - Endpoints	12		12			Sprint Retrospective	2	2			
[DISSENY UI/UX] - Disseny cistella + pantalla pagament	2				2	Documentació	8	8			
[FRONT] - Cistella de la compra + pantalla pagament	12		12			Testing i refactoring	24		12	12	
Gestió de bugs	2		2			Desplegar aplicació	16		8	8	
Sprint Review	2	1		1		Memòria final	30	30			
Sprint Retrospective	2	2				Presentació final	24	24			
Documentació	2	2				**TOTAL**	**540**	**193**	**308**	**25**	**14**

Figura A.2: Modificacions d'hores treballades per rols. (Font pròpia)

A.2 Endpoints

A.2.1 Gestió de productes

Description	**PRODUCT.001** - Llistar tots els productes existents
Method	GET
URL	`/api/products`
Headers	**languageId:** string
Query Params	**pageSize:** number **pageNumber:** number **sortBy (opcional):** string **direction (opcional):** string
Responses	**200: OK** - Retorna un llistat de productes paginat i ordenat (PageProductDto) **500: Internal Server Error**

Description	**PRODUCT.002** - Crear un producte
Method	POST
URL	`/api/products/add`
Body	ProductRequestDto
Responses	**201: CREATED** - Producte creat **400: BAD REQUEST** **404: NOT FOUND** **500: Internal Server Error**

Description	**PRODUCT.003** - Eliminar un producte
Method	DELETE
URL	`/api/products/:productId`
Responses	**200: OK** - Producte eliminat **404: NOT FOUND** **500: Internal Server Error**

Description	**PRODUCT.004** - Obtenir el detall del producte
Method	GET
URL	`/api/products/:productId`
Headers	**languageId:** string
Responses	**200: OK** - Retorna el detall del producte (ProductDetailDto) **404: NOT FOUND** **500: Internal Server Error**

A.2.2 Gestió de la cistella de la compra

Description	**CART.001** - Llistar tots els productes existents a la cistella de la compra d'un usuari
Method	GET
URL	`/api/cart`
Headers	**languageId:** string **userId:** number
Responses	**200: OK** - Retorna el detall de la cistella de la compra actual de l'usuari (UserCurrentCartDto) **500: Internal Server Error**

Description	**CART.002** - Afegir un producte a la cistella de la compra de l'usuari
Method	POST
URL	`/api/cart/products`
Headers	**languageId:** string **userId:** number
Body	AddCartProductRequestDto
Responses	**200: OK** - Retorna el detall de la cistella de la compra actual de l'usuari (UserCurrentCartDto) **500: Internal Server Error**

Description	**CART.002** - Modificar la quantitat d'un producte existent a la cistella de la compra de l'usuari
Method	PUT
URL	`/api/cart/products/:productId`
Headers	**languageId:** string **userId:** number
Body	number (amount)
Responses	**200: OK** - Retorna el detall de la cistella de la compra actual de l'usuari (UserCurrentCartDto) **404: NOT FOUND** - El producte amb l'identificador especificat no existeix a la cistella actual de l'usuari **500: Internal Server Error**

Description	**CART.004** - Eliminar un producte a la cistella de la compra de l'usuari
Method	DELETE
URL	`/api/cart/products`
Headers	**languageId:** string **userId:** number
Body	number (productId)
Responses	**200: OK** - Retorna el detall de la cistella de la compra actual de l'usuari (UserCurrentCartDto) **500: Internal Server Error**

Description	**CART.004** - Eliminar un producte a la cistella de la compra de l'usuari
Method	DELETE
URL	`/api/cart/products`
Headers	**languageId:** string **userId:** number
Body	number (productId)
Responses	**200: OK** - Retorna el detall de la cistella de la compra actual de l'usuari (UserCurrentCartDto) **500: Internal Server Error**

Description	**CART.005** - Eliminar la cistella de la compra de l'usuari
Method	DELETE
URL	`/api/cart`
Headers	**userId:** number
Responses	**200: OK** **500: Internal Server Error**

Description	**CART.006** - Afegir els productes seleccionats d'una recepta a la cistella de l'usuari
Method	POST
URL	`/api/cart/recipe/products`
Headers	**languageId:** string **userId:** number
Body	[number] (productIds)
Responses	**200: OK** - Retorna el detall de la cistella de la compra actual de l'usuari (UserCurrentCartDto) **500: Internal Server Error**

A.2.3 Pagament

Description	**PAYMENT.001** - Crear un "PaymentIntent"
Method	POST
URL	`/api/payment/intent`
Body	PaymentIntentDto
Responses	**200: OK** - Retorna les dades de l'intent de pagament creat **500: Internal Server Error**

Description	**PAYMENT.002** - Confirmar un "PaymentIntent"
Method	POST
URL	`/api/payment/confirm/:id`
Responses	**200: OK** - Retorna les dades de l'intent de pagament confirmat **500: Internal Server Error**

Description	**PAYMENT.003** - Cancelar un "PaymentIntent"
Method	POST
URL	`/api/payment/cancel/:id`
Responses	**200: OK** - Retorna les dades de l'intent de pagament cancelat **500: Internal Server Error**

A.2.4 Gestió de les comandes

Description	**ORDER.001** - Registrar una comanda
Method	POST
URL	`/api/order`
Headers	**languageId:** string **userId:** number
Responses	**200: OK** - Retorna la informació de la comanda que s'ha realitzat (OrderDto) **500: Internal Server Error**

Description	**ORDER.002** - Llistar totes les comandes registrades d'un usuari
Method	GET
URL	`/api/user/orders`
Headers	**userId:** number
Responses	**200: OK** - Retorna un llistat amb la informació de les comandes que ha realitzat l'usuari corresponent (List<OrderDto>) **500: Internal Server Error**

A.2.5 Gestió de receptes

Description	**RECIPES.001** - Obtenir el detall d'una recepta a partir del seu nom
Method	GET
URL	`/api/ai/recipe`
Headers	**languageId:** string
Params	**recipe: string**
Responses	**200: OK** - Retorna un llistat de cada producte guardat per al cada ingredient de la recepta i el preu total (RecipeDetailDto) **500: Internal Server Error**

A.2.6 Gestió de favorits

Description	**FAVORITES.001** - Afegir un producte a favorits
Method	POST
URL	`/api/favorites/product/:productId`
Headers	**languageId:** string **userId:** number
Responses	**200: OK** - Retorna un llistat de tots els productes favorits de l'usuari (List<ProductDto>) **500: Internal Server Error**

Description	**FAVORITES.002** - Eliminar un producte de favorits
Method	DELETE
URL	`/api/favorites/product/:productId`
Headers	**languageId:** string **userId:** number
Responses	**200: OK** - Retorna un llistat de tots els productes favorits de l'usuari (List<ProductDto>) **500: Internal Server Error**

Description	**FAVORITES.003** - Llistar tots els producte favorits d'un usuari
Method	GET
URL	`/api/favorites/products`
Headers	**languageId:** string **userId:** number
Responses	**200: OK** - Retorna un llistat de tots els productes favorits de l'usuari (List<ProductDto>) **500: Internal Server Error**

Description	**FAVORITES.004** - Afegir una recepta a favorits
Method	POST
URL	`/api/favorites/recipe`
Headers	**userId:** number
Body	UserFavoriteRecipesDto
Responses	**200: OK** - Retorna un llistat de totes les receptes favorites de l'usuari (List<FavoriteRecipeDto>) **500: Internal Server Error**

Description	**FAVORITES.005** - Llistar totes les receptes favorites d'un usuari
Method	GET
URL	`/api/favorites/recipes`
Headers	**userId:** number
Responses	**200: OK** - Retorna un llistat de totes les receptes favorites de l'usuari (List<FavoriteRecipeDto>) **500: Internal Server Error**

Description	**FAVORITES.006** - Eliminar una recepta de favorits
Method	DELETE
URL	`/api/favorites/recipe/:recipeId`
Headers	**userId:** number
Responses	**200: OK** - Retorna un llistat de totes les receptes favorites de l'usuari (List<FavoriteRecipeDto>) **500: Internal Server Error**

Description	**FAVORITES.007** - Obtenir el detall d'una recepta de favorits
Method	GET
URL	`/api/favorites/recipe/:recipeId`
Headers	**userId:** number **languageId:** string
Responses	**200: OK** - Retorna un llistat de cada producte guardat per al cada ingredient de la recepta i el preu total (RecipeDetailDto) **500: Internal Server Error**

A.2.7 Gestió dels usuaris

Description	**USER.001** - Registrar un usuari
Method	POST
URL	`/api/user/register`
Body	UserRegisterDto
Responses	**200: OK** - Retorna l'identificador de l'usuari registrat **500: Internal Server Error**

Description	**USER.002** - Iniciar la sessió d'un usuari anteriorment registrat
Method	POST
URL	`/api/user/login`
Body	LoginDto
Responses	**200: OK** - Retorna l'identificador de l'usuari registrat **500: Internal Server Error**

Description	**USER.003** - Obtenir l'informació de l'usuari
Method	GET
URL	`/api/user/:userId`
Responses	**200: OK** - Retorna un objecte amb la informació personal de l'usuari (UserInfoDto) **500: Internal Server Error**

A.3 Schemas

PageProductDto

```
{
  "totalPages": number,
  "totalElements": number,
  "first": boolean,
  "last": boolean,
  "size": number,
  "content": [ProductDto],
  "number": number,
  "sort": {
    "empty": boolean,
    "sorted": boolean,
    "unsorted": boolean
  },
  "pageable": {
    "offset": number,
    "sort": {
      "empty": true,
      "sorted": true,
      "unsorted": true
    },
    "pageNumber": number,
    "pageSize": number,
    "paged": boolean,
    "unpaged": boolean
  },
  "numberOfElements": number,
  "empty": boolean
}
```

Listing A.1: PageProductDto

ProductDto

```
{
      "productId": number,
      "price": string,
      "stock": number,
      "brand": string,
      "name": string,
      "image": string
}
```

Listing A.2: ProductDto

ProductRequestDto

```
{
  "price": number,
  "stock": number,
  "categoryId": number,
  "typeId": number,
  "brand": string,
  "image": string,
  "primary": boolean,
  "productLanguages": [
    {
      "languageId": string,
      "name": string,
      "description": string,
      "ingredients": string
    }
  ],
  "nutritionalAttributes": [
    { "key": string }
  ]
}
```

Listing A.3: ProductRequestDto

ProductDetailDto

```
{
  "productId": number,
  "price": string,
  "stock": number,
  "categoryId": number,
  "typeId": number,
  "brand": string,
  "name": string,
  "description": string,
  "ingredients": string,
  "image": string,
  "nutritionalAttributes": [NutritionalAttributeDto]
}
```

Listing A.4: ProductDetailDto

NutritionalAttributeDto

```
{
  "attributeId": number,
  "attributeName": string,
  "attributeValue": string,
  "includeProgress": boolean,
  "children": [NutritionalAttributeDto]
}
```

Listing A.5: NutritionalAttributeDto

AddCartProductRequestDto

```
{
  "productId": number,
  "amount": number
}
```

Listing A.6: AddCartProductRequestDto

PaymentIntentDto

```
{
  "token": string,
  "description": string,
  "amount": number
}
```

Listing A.7: PaymentIntentDto

OrderDto

```
{
  "orderId": number,
  "userId": number,
  "totalAmount": string,
  "state": string,
  "date": string,
}
```

Listing A.8: OrderDto

FavoriteRecipeDto

```
{
  "recipeId": number,
  "name": string,
  "image": string,
}
```

Listing A.9: FavoriteRecipeDto

RecipeDetailDto

```
{
  "productsByIngredient": [Map<String, Object>],
  "amountSelectedProducts": string
}
```

Listing A.10: RecipeDetailDto

UserFavoriteRecipesDto

```
{
  "name": string,
  "ingredients": {
    "key": number
  }
}
```

Listing A.11: UserFavoriteRecipesDto

UserRegisterDto

```
{
  "email": string,
  "password": string,
  "name": string,
  "surname": string,
  "phone": string,
  "birthday": string
}
```

Listing A.12: UserRegisterDto

LoginDto

```
{
  "email": string,
  "password": string
}
```

Listing A.13: LoginDto

UserInfoDto

```
{
  "name": string,
  "email": string,
  "birthday": string,
  "phone": string
}
```

Listing A.14: UserInfoDto