

**Yani Aici Lounis**

**APLICACIÓ DE GAMIFICACIÓ I ANÀLISI PER MERCATS**

**TREBALL DE FI DE GRAU**

**dirigit per Sr. Roy González Salgado**

**Grau d'Enginyeria Informàtica**



**UNIVERSITAT ROVIRA I VIRGILI**

**Tarragona**

**2025**

**Resum.**

Mercat Mediterrani és un TFG que desenvolupa un sistema de gestió de tiquets digitals per a mercats municipals, amb interfície en català i panells per a usuari, venedor i administrador. La plataforma combina gamificació (punts, recompenses, insígnies) i IA (Google Gemini i opcionalment Ollama) per escanejar i processar tiquets, detectar duplicats i analitzar compres per oferir recomanacions. Inclou un CRM per segmentar clients, gestionar campanyes i notificacions. L'arquitectura es basa en microserveis amb FastAPI i un frontend en React, base de dades PostgreSQL i desplegament amb Docker, pensada per a escalabilitat, seguretat i modernització de la fidelització als mercats locals.

**Resumen.**

Mercat Mediterrani es un TFG que desarrolla un sistema de gestión de tickets digitales para mercados municipales, con interfaz en catalán y paneles para usuario, vendedor y administrador. La plataforma combina gamificación (puntos, recompensas, insignias) e IA (Google Gemini y opcionalmente Ollama) para escanear y procesar tickets, detectar duplicados y analizar compras para recomendaciones. Incluye un CRM para segmentar clientes, gestionar campañas y notificaciones. La arquitectura se basa en microservicios con FastAPI y un frontend en React, base de datos PostgreSQL y despliegue con Docker, enfocada en escalabilidad, seguridad y modernización de la fidelización en mercados locales.

**Abstract.**

Mercat Mediterrani is a Final Degree Project delivering a digital ticket management system for municipal markets, with a Catalan UI and role-based dashboards for users, vendors, and admins. It blends gamification (points, rewards, badges) and AI (Google Gemini with optional local Ollama) to scan and process tickets, detect duplicates, and analyze purchases for recommendations. An independent CRM handles customer segmentation, campaigns, and notifications. Built with a microservices backend (FastAPI), a React frontend, PostgreSQL, and Docker-based deployment, the system targets scalability, security, and modern customer loyalty in local markets.

## **Agraïments**

Vull agrair al meu tutor Roy González per donar l'idea principal d'aquest TFG i per tota la seva ajuda durant la realització d'aquest, el seu feedback ha sigut fonamental pel desenvolupament tècnic d'aquest.

També vull donar les gràcies a la meva parella, Nero Soler, pel seu constant suport i també pel feedback no tècnic que m'ha donat durant la realització d'aquest TFG.

# Índex

<b>1</b>	<b>INTRODUCCIÓ</b> .....	<b>5</b>
1.1	PROPOSTA DE SOLUCIÓ .....	5
1.2	LA GAMIFICACIÓ COM A PALANCA D'ATRACCIÓ I FIDELITAT JUVENIL .....	7
1.3	ANALÍTICA DE PRODUCTE I SEGMENTACIÓ .....	7
<b>2</b>	<b>DESCRIPCIÓ GENERAL DEL PROJECTE</b> .....	<b>8</b>
2.1	ARQUITECTURA I COMPONENTS .....	8
2.2	FLUX FUNCIONAL I ROLS .....	8
2.3	GAMIFICACIÓ I CRM .....	8
2.4	PROCESSAMENT AMB IA I QUALITAT DE LA DADA .....	9
2.5	SEGURETAT, PRIVADESA I OPERACIÓ .....	9
2.6	PREVISIONS D'ÚS, ESCALABILITAT I DECISIONS D'IA .....	9
<b>3</b>	<b>PLANIFICACIÓ</b> .....	<b>11</b>
3.1	CRONOGRAMA RESUM (JUNY-AGOST 2025) .....	12
3.2	GESTIÓ DE DEPENDÈNCIES I PARAL·LELITZACIÓ .....	12
3.3	TANCAMENT I LLIURAMENT FINAL .....	12
<b>4</b>	<b>METODOLOGIA DE TREBALL</b> .....	<b>13</b>
4.1	FLUX DE TREBALL AMB TRELLO I ESTADIS DE LES TASQUES .....	13
4.2	INTEGRACIÓ CONTÍNUA AMB DOCKER .....	14
<b>5</b>	<b>REQUISITS DEL PROJECTE</b> .....	<b>15</b>
5.1	REQUISITS FUNCIONALS .....	15
5.1.1	<i>MoSCoW</i> .....	15
5.2	REQUISITS DE GESTIÓ D'USUARIS I AUTENTICACIÓ .....	16
5.3	REQUISITS DEL SISTEMA DE CAPTURA, PROCESSAMENT I CONSULTA DE TIQUETS .....	16
5.4	REQUISITS DEL HISTORIAL DE COMPRES I ANALÍTICA D'USUARI .....	18
5.5	REQUISITS DEL SISTEMA DE GAMIFICACIÓ .....	19
5.6	REQUISITS DEL SISTEMA DE NOTIFICACIONS I CRM .....	19
5.7	REQUISITS DEL PERFILATGE I SEGMENTACIÓ AMB IA .....	20
5.8	REQUISITS DE LA APLICACIÓ WEB I LA APLICACIÓ MÒBIL .....	20
5.9	REQUISITS NO FUNCIONALS .....	20
5.10	RESTRICCIONS I LIMITACIONS DEL PROJECTE .....	21
5.10.1	<i>Abast i dependències</i> .....	21
5.10.2	<i>Processament amb IA i qualitat de dades</i> .....	21
5.10.3	<i>Duplicats i coherència</i> .....	22
5.10.4	<i>Multimercat i aïllament</i> .....	22
5.10.5	<i>Notificacions i CRM</i> .....	22
5.10.6	<i>Experiència al punt de venda i operativa quotidiana</i> .....	22
5.10.7	<i>Compliment i govern de dades</i> .....	22
5.11	CASOS D'ÚS .....	22
5.11.1	<i>CU-1. Registre i inici de sessió de l'usuari</i> .....	23
5.11.2	<i>CU-2. Pujada d'un tiquet escanejat</i> .....	24
5.11.3	<i>CU-3. Processament amb IA d'un tiquet</i> .....	25
5.11.4	<i>CU-4. Detecció i marcat de duplicats</i> .....	27
5.11.5	<i>CU-5. Emissió d'un tiquet digital pel venedor</i> .....	28
5.11.6	<i>CU-6. Gestió de botigues del mercat</i> .....	29
5.11.7	<i>CU-7. Consulta de tiquets i historial</i> .....	31
5.11.8	<i>CU-8. Assignació de punts i gamificació</i> .....	32
5.11.9	<i>CU-9. Redempció de premis al punt de venda</i> .....	33
5.11.10	<i>CU-10. Enviament de notificacions i campanye</i> .....	34
5.11.11	<i>CU-11. Administració d'usuaris i rols</i> .....	36
5.12	DIAGRAMA DE CLASSES .....	38

<b>6</b>	<b>DISSENY</b> .....	<b>39</b>
6.1	ARQUITECTURA GENERAL DE L' APLICATIU .....	39
6.2	MICROSERVEIS DE L' APLICATIU.....	40
6.3	INTERFÍCIE GRÀFICA .....	42
6.4	PANTALLES D'USUARI.....	43
6.4.1	<i>Pantalla de login</i> .....	43
6.4.2	<i>Dashboard d'usuari</i> .....	43
6.4.3	<i>Historial de tiquets</i> .....	45
6.4.4	<i>QR del perfil</i> .....	45
6.4.5	<i>Pantalla d'escaneig de tiquet</i> .....	46
6.4.6	<i>Pantalla de recompenses</i> .....	47
6.4.7	<i>Detalls del tiquet digital</i> .....	48
6.5	PANTALLES DEL VENEDOR.....	48
6.5.1	<i>Dashboard del venedor</i> .....	48
6.5.2	<i>Tiquet digital</i> .....	49
6.5.3	<i>Validació de recompenses</i> .....	49
6.6	PANTALLA D'ADMINISTRADOR .....	50
6.7	PERSISTÈNCIA A LA BD.....	52
<b>7</b>	<b>IMPLEMENTACIÓ</b> .....	<b>53</b>
7.1	TECNOLOGIES I CONTROL DE VERSIONS.....	53
7.2	ORGANITZACIÓ DEL CODI.....	53
7.3	IMPLEMENTACIÓ DEL BACKEND PER SERVEI.....	55
7.3.1	<i>Auth-service</i> .....	55
7.3.2	<i>Main Backend (facade)</i> .....	56
7.3.3	<i>Ticket-service</i> .....	57
7.3.4	<i>AI ticket-processor</i> .....	58
7.3.5	<i>Gamification-service</i> .....	59
7.3.6	<i>CRM-service</i> .....	60
7.3.7	<i>Notification-Sender</i> .....	62
7.4	IMPLEMENTACIÓ DE L' APLICACIÓ MÒBIL .....	63
7.5	DOMINI I HOSTING .....	64
7.5.1	<i>Allotjament</i> .....	64
7.5.2	<i>Domini I DNS</i> .....	64
7.5.3	<i>HTTPS</i> .....	65
<b>8</b>	<b>AVALUACIÓ</b> .....	<b>66</b>
8.1	PROVES D' AUTENTICACIÓ I USUARIS .....	66
8.2	PROVES DE GESTIÓ DE TICKETS.....	67
8.3	PROVES DE IA PER PROCESSAR TICKETS .....	67
8.4	PROVES DE CRM I SEGMENTACIÓ .....	68
8.5	PROVES DE NOTIFICACIONS.....	69
8.6	PROVES AL FRONTEND .....	69
8.7	PROVES DE FLUXOS PRINCIPALS.....	70
<b>9</b>	<b>LEGISLACIÓ I PROTECCIÓ DE DADES</b> .....	<b>72</b>
<b>10</b>	<b>ÈTICA I RESPONSABILITAT SOCIAL</b> .....	<b>73</b>
<b>11</b>	<b>CONCLUSIONS</b> .....	<b>75</b>
<b>12</b>	<b>RECURSOS UTILITZATS</b> .....	<b>76</b>
	REFERÈNCIES .....	77
	ANNEX .....	78

## Índex de taules

TAULA 1. REQUISITS FUNCIONALS DE GESTIÓ D'USUARIS I AUTENTICACIÓ.....	16
TAULA 2. REQUISITS FUNCIONALS DE CAPTURA PROCESSAMENT I CONSULTA DE TIQUETS .....	18
TAULA 3. REQUISITS DEL HISTORIAL DE COMPRES I ANALÍTICA DEL USUARI.....	18
TAULA 4. REQUISITS DEL SISTEMA DE GAMIFICACIÓ.....	19
TAULA 5. REQUISITS DEL SISTEMA DE NOTIFICACIONS I CRM.....	19
TAULA 6. REQUISITS DEL PERFILATGE I SEGMENTACIÓ AMB IA .....	20
TAULA 7. REQUISITS DE LA APLICACIÓ WEB I APLICACIÓ MÒBIL.....	20
TAULA 8. REQUISITS NO FUNCIONALS .....	21
TAULA 9. REGISTRE I INICI DE SESSIÓ D'USUARI .....	24
TAULA 10. PUJADA D'UN TIQUET ESCANEJAT.....	25
TAULA 11. PROCESSAMENT AMB IA D'UN TIQUET.....	26
TAULA 12. DETECCIÓ I MARCAT DE DUPLICATS.....	27
TAULA 13. EMISSIÓ D'UN TIQUET DIGITAL PEL VENEDOR.....	29
TAULA 14. GESTIÓ DE BOTIGUES DEL MERCAT .....	30
TAULA 15. CONSULTA DE TIQUETS I HISTORIAL .....	31
TAULA 16. ASSIGNACIÓ DE PUNTS I GAMIFICACIÓ.....	32
TAULA 17. REDEMPCIÓ DE PREMIS AL PUNT DE VENDA.....	33
TAULA 18. ENVIAMENT DE NOTIFICACIONS I CAMPANYES.....	34
TAULA 19. ADMINISTRACIÓ D'USUARIS I ROLS .....	36
TAULA 20. PROVES D'AUTENTICACIÓ I USUARIS.....	66
TAULA 21. PROVES DE GESTIÓ DE TICKETS .....	67
TAULA 22. PROVES DE IA PER PROCESSAR TIQUETS .....	68
TAULA 23. PROVES DE CRM I SEGMENTACIÓ.....	69
TAULA 24. PROVES DE NOTIFICACIONS.....	69
TAULA 25. PROVES AL FRONTEND .....	70
TAULA 26. PROVES DE FLUXOS PRINCIPALS .....	71

## Índex de figures

FIGURA 1. DIAGRAMA DE GANTT .....	11
FIGURA 2. ORGANITZACIÓ EN TRELLO DEL TFG .....	13
FIGURA 3. DIAGRAMA GENERAL DE CASOS D'ÚS.....	23
FIGURA 4. DIAGRAMA DEL CU-1.....	24
FIGURA 5. DIAGRAMA DEL CU-2.....	25
FIGURA 6. DIAGRAMA DEL CU-3.....	26
FIGURA 7. DIAGRAMA DEL CU-4.....	28
FIGURA 8. DIAGRAMA DEL CU-5.....	29
FIGURA 9. DIAGRAMA DEL CU-6.....	30
FIGURA 10. DIAGRAMA DEL CU-7 .....	31
FIGURA 11. DIAGRAMA DEL CU-8.....	32
FIGURA 12. DIAGRAMA DEL CU-9.....	33
FIGURA 13. DIAGRAMA DEL CU-10 .....	35
FIGURA 14. DIAGRAMA DEL CU-11 .....	37
FIGURA 15. DIAGRAMA DE CLASSES.....	38
FIGURA 16. DIAGRAMA DE L'ARQUITECTURA DE L'APLICACIÓ .....	39
FIGURA 17. PARÀMETRE D'ENTORN A LES VARIABLES D'ENTORN .....	40
FIGURA 18. MAPEIG DE LA VARIABLE D'ENTORN AL CONTENIDOR DOCKER .....	40
FIGURA 19. MICROSERVEIS I DEPENDÈNCIES DEL PROJECTE DESPLEGATS COM A CONTENIDORS DOCKER.....	42
FIGURA 20. PANTALLA DE LOGIN.....	43
FIGURA 21. PANTALLA DE DASHBOARD (PART SUPERIOR) .....	43
FIGURA 22. MENÚ DESPLEGABLE DE NOTIFICACIONS.....	44
FIGURA 23. PANTALLA DEL HISTORIAL DE TIQUETS .....	45
FIGURA 24. MENÚ "QR DEL TEU PERFIL" .....	45
FIGURA 25. PANTALLA D'ESCANEIG DE TIQUET .....	46
FIGURA 26. PANTALLA DE RECOMPENSES (PART SUPERIOR).....	47
FIGURA 27. PANTALLA DE RECOMPENSES (PART INFERIOR) .....	47
FIGURA 28. DETALLS DEL TIQUET DIGITAL .....	48
FIGURA 29. PANELL DEL VENEDOR.....	48
FIGURA 30. TIQUET DIGITAL.....	49
FIGURA 31. PANTALLA DE VALIDACIÓ DE RECOMPENSA .....	49
FIGURA 32. PANTALLA DE RECOMPENSA JA VALIDADA .....	50
FIGURA 33. PANTALLA DE ADMINISTRACIÓ (USUARIS).....	50
FIGURA 34. PANTALLA DE ADMINISTRACIÓ (CRM) .....	51
FIGURA 35. PANTALLA DE ADMINISTRACIÓ (RECOMPENSES ESPECIALS).....	51
FIGURA 36. EXEMPLE D'UN FITXER MODELS.PY DE CADA MICROSERVEI .....	54
FIGURA 37. ORGANITZACIÓ DELS FITXERS DEL FRONTEND .....	54
FIGURA 38. CONTROLLER DEL AUTH-SERVICE.....	56
FIGURA 39. LÒGICA DE COMPARACIÓ DE HASHES.....	58
FIGURA 40. CODI AMB EL PROMPT A GEMINI.....	59
FIGURA 41. CODI AMB LA COMPROVACIÓ SI EL USUARI POT O NO ACCEDIR A UNA RECOMPENSA ESPECIAL .....	60
FIGURA 42. CODI ON RECULL I NORMALITZA COM A SEGMENT EL PROMPT DEL USUARI.....	62
FIGURA 43. CODI DE LA APP MÒBIL .....	63
FIGURA 44. CONFIGURACIÓ DE REGISTRES DNS.....	65
FIGURA 45. RAILS EXPLÍCITS A LA GENERACIÓ PER IA DE LES NOTIFICACIONS.....	73

## 1 Introducció

Els mercats municipals són un pilar del comerç de proximitat i de l'alimentació saludable, però la digitalització accelerada ha redefinit les expectatives dels consumidors. Plataformes i supermercats han consolidat experiències d'usuari molt fluides (pagaments sense fricció, ofertes personalitzades, historial accessible, notificacions oportunes) que eleven el llindar de conveniència. En aquest marc, una part significativa del públic jove percep el mercat com un entorn complex: no sap què comprar, li manca orientació sobre la temporada o les combinacions de producte, i sovint desconeix avantatges rellevants com les ofertes d'aparcament o el servei de cuina al lloc. Aquest dèficit d'informació en el moment i el canal adequats erosiona la sensació de control i utilitat, i fa menys probable l'elecció del mercat quan es comparen alternatives més previsibles.

A aquesta bretxa d'experiència s'hi afegeix una bretxa de dades. La majoria de paradistes operen amb sistemes de cobrament i TPV heterogenis, sovint sense capacitat nativa per integrar dades o segmentar la clientela. Són petites empreses, amb recursos i temps limitats per explorar eines d'analítica o campanyes dirigides. El resultat és un mosaic tecnològic on la informació queda fragmentada per parada i no es transforma en coneixement accionable. Des de la gestió del mercat, això es tradueix en manca de visibilitat granular: costa identificar perfils de client, entendre la composició de les cistelles, detectar patrons temporals d'afluència i mesurar l'efecte de promocions o activitats. Sense aquesta base, la presa de decisions tendeix a ser reactiva i poc mesurable, i és difícil articular iniciatives específiques per captar i fidelitzar públic jove o per reduir el malbaratament amb estratègies de rotació i temporada.

Aquestes limitacions no només afecten el creixement, sinó que comprometen la proposta de valor del mercat. La qualitat del producte, la traçabilitat, la sostenibilitat i la relació humana amb el venedor —actius diferencials— queden desdibuixats si no s'aconsegueix fer-los visibles i tangibles en el recorregut digital del client. En síntesi, el repte no és únicament competir amb el canal en línia, sinó traduir els valors del mercat a una experiència contemporània que redueixi la fricció, aportï orientació i generi confiança basada en dades.

### 1.1 Proposta de solució

Aquest Treball de Fi de Grau planteja una aplicació que modernitza l'experiència sense renunciar a l'essència del mercat. La peça central és el tiquet digital: cada compra es converteix en una font de dades estructurada que permet entendre què es compra, amb quina freqüència i en quins moments. A partir d'aquí, l'aplicació ofereix una experiència mobile-first que acompanya l'usuari abans, durant i després de la visita: recomanacions de producte de temporada i combinacions per a receptes senzilles, recordatoris contextuais d'avantatges com l'aparcament o la cuina al lloc, i un espai on el client pot consultar els seus tiquets, gestionar recompenses i rebre ofertes ajustades als seus interessos reals.

Per als paradistes, la solució es dissenya perquè l'adopció sigui viable en entorns heterogenis. S'habiliten connectors lleugers i importacions periòdiques de fitxers senzills que permeten unificar les dades de venda sense exigir canvis de TPV. Amb aquesta base, cada parada obté una lectura clara dels patrons de compra —cistell mitjà, recurrències, franges d'afluència— i pot planificar promocions amb criteri, incloses iniciatives per donar sortida a producte de rotació lenta o excés de temporada. Des de la direcció del mercat, els quadres de comandament agreguen la informació necessària per programar accions de dinamització, coordinar campanyes, mesurar-ne l'impacte i justificar decisions amb indicadors sòlids.

La gamificació actua com a motor d'activació i fidelització, especialment en públic jove. El sistema de punts i recompenses introdueix una dinàmica d'objectius clars i retorns tangibles que fan més atractiva la recurrència. A més de les recompenses habituals, s'incorporen recompenses especials amb condicions específiques (per exemple, incentius per combinar compres entre parades, per triar producte de temporada o per participar en accions de sostenibilitat) que alineen el benefici del client amb els objectius del mercat i dels venedors. Aquestes dinàmiques, informades per les dades dels tiquets, permeten personalitzar l'experiència sense caure en la sobreexposició de missatges.

Tot el disseny s'articula sota criteris de seguretat, privacitat i ús responsable de les dades, amb una arquitectura modular que facilita l'evolució del producte i la incorporació de noves funcionalitats. L'objectiu no és substituir la relació humana característica del mercat, sinó reforçar-la: dotar venedors i clients d'eines senzilles que facin més fàcil trobar-se, entendre's i construir confiança sobre evidències. Així, la transformació digital es converteix en una oportunitat per posar en valor el que el mercat ja fa millor —proximitat, qualitat i sostenibilitat— i fer-ho visible en un llenguatge i uns canals que la nova generació reconeix com a propis.

## 1.2 La gamificació com a palanca d'atracció i fidelitat juvenil

La **literatura científica** [1] [2] apunta efectes positius de la gamificació sobre la motivació, el compromís i la retenció, especialment en mostres juvenils i universitàries. Una meta-anàlisi en context educatiu identifica efectes petits a moderats en motivació i resultats d'aprenentatge quan s'empren punts, insígnies i reptes. També es conclou que la gamificació tendeix a millorar la participació i a facilitar canvis de conducta, tot i que l'eficàcia depèn del disseny i del context. En l'àmbit digital, una revisió sistemàtica mostra increments d'ús i d'adhesió a programes en línia gràcies a elements com classificacions, recompenses i feedback immediat, que són especialment valorats per usuaris joves habituats a experiències mòbils. De manera transversal, els estudis de síntesi apunten que els bucles de progrés, els objectius clars i el reconeixement social són mecanismes que connecten bé amb preferències de la Generació Z: immediatesa, personalització i sensació de progrés.

Aquesta evidència dona suport a l'ús de mecàniques de joc en programes de fidelització i apps de consum: la combinació de punts, insígnies, reptes i recompenses activables en el punt de venda facilita la creació d'hàbits i la recurrència de visita. En el cas dels mercats de fresc, aquests mecanismes permeten transformar la compra en una experiència digital amb valor afegit (descomptes contextuals, recomanacions saludables, reptes de temporada), fent-la més atractiva per al públic jove sense desvirtuar l'essència del comerç de proximitat. [12]

## 1.3 Analítica de producte i segmentació

Més enllà del format del tiquet, el que resulta clau és que, quan l'usuari escaneja els seus tiquets de compra, el sistema recull de manera estructurada els productes adquirits (línies d'article, quantitats, imports, data/hora i parada). Aquesta informació, processada i normalitzada amb suport d'IA, permet entendre què es compra realment, quins són els supervendes per categoria i temporada, com es compon el cistell mitjà, i quines són les franges d'afluència i la recurrència de visita. A partir d'aquí, es poden construir segments útils (p. ex. RFM, preferències per categoria, sensibilitat al preu, cohorts per antiguitat) que ajuden a personalitzar l'experiència i a dissenyar millor l'oferta.

Aquest coneixement s'opera en forma de campanyes i comunicacions personalitzades que incentiven la visita al mercat: cupons i recompenses vinculats a hàbits reals, reptes de temporada (fruita local, producte ecològic), recomanacions afins al perfil i notificacions rellevants en el moment oportú. El cicle es tanca mesurant l'impacte (taxa d'obertura, conversió, canje, increment de cistell) per aprendre i optimitzar. Tot plegat es desenvolupa amb criteris de privacitat i compliment normatiu (RGPD [3][4]): consentiment informat, minimització de dades, pseudonimització i polítiques de retenció clares. En síntesi, l'escaneig de tiquets converteix cada compra en un actiu d'informació que impulsa l'analítica, la segmentació i les campanyes personalitzades, aconseguint que l'usuari trobi més motius per tornar al mercat.

## 2 Descripció general del projecte

Aquest TFG desenvolupa una plataforma digital modular i multi-tenant orientada a la digitalització de mercats de proximitat. L'objectiu és disposar d'un únic sistema white-label que es pugui desplegar i configurar ràpidament per a diferents mercats (tenants) preservant l'aïllament de dades, permetent personalitzar imatge i regles, i mantenint un nucli tècnic comú fàcil de mantenir i evolucionar. Com a demostrador funcional, s'ha instanciat un primer tenant fictici —el “Mercat Mediterrani”— que il·lustra l'experiència d'usuari, la integració de mòduls i els fluxos end-to-end, però el disseny està pensat perquè altres mercats es puguin incorporar amb mínims canvis operatius i sense canviar el codi base.

### 2.1 Arquitectura i components

L'arquitectura es basa en microserveis amb FastAPI (Python 3.11 [11]) i un frontend en React [9] [10] + TypeScript. Els serveis principals són: autenticació/autorització (gestió d'usuaris i rols user|vendor|admin amb JWT, preferències en JSONB, rutes d'administració i overview); servei de tiquets (recepció d'escanejors, emissió, validació, detecció de duplicats i integració amb gamificació i historial); gamificació (puntuació, recompenses, insígnies, validacions via QR); CRM (segmentació, campanyes i notificacions); i processament d'IA per a extracció/normalització de dades de tiquets. La persistència es fa amb PostgreSQL (administrable via PgAdmin) i l'orquestració amb Docker/Docker Compose [6], amb Nginx com a proxy i terminació TLS en producció.

### 2.2 Flux funcional i rols

El flux central comença quan el client escaneja un tiquet amb la càmera del mòbil o quan el venedor li emet un tiquet digital escanejant el QR personal de l'usuari. El sistema crea el tiquet, executa validacions (inclosa la detecció configurable de duplicats), persisteix les línies de producte i actualitza l'historial de compres (data, botiga, import, productes). En paral·lel, s'activa el càlcul de punts i la disponibilitat de recompenses. El client disposa d'un panell per registrar-se, iniciar sessió, escanejar/rebre tiquets, veure punts, recompenses i l'historial de tiquets. El venedor, des d'un panell autenticat, identifica l'usuari pel seu QR, emet tiquets digitals i valida redempcions amb control d'idempotència i auditories. L'administrador del mercat opera un panell de gestió on configura catàleg i recompenses, defineix segments i campanyes, i consulta mètriques d'impacte; en un entorn multi-tenant, només pot veure i operar el seu mercat.

### 2.3 Gamificació i CRM

La gamificació és una capa configurable per mercat: taules de punts, llistats i caducitats, reptes de temporada (producte local, antimalbarament) i recompenses activables per l'administrador. El feedback immediat tanca un bucle de progrés que incentiva hàbits i recurrència. El CRM construeix segments a partir de l'historial estructurat (RFM, preferències per categoria, sensibilitat al preu, cohorts, resposta a campanyes) i orquestra comunicacions contextuais (cupons, punts extra, reptes). Quan hi ha VAPID, les notificacions s'envien com a push; si no, es persisteixen i es mostren in-app.

## 2.4 Processament amb IA i qualitat de la dada

El servei d'IA integra Google Gemini i, opcionalment, Ollama per inferència local, combinat amb OpenCV i Pillow per tractament d'imatge. Automatitza l'extracció de línies d'article, imports, impostos i metadades, i aplica normalització de descripcions/format per assegurar consistència analítica. Hi ha mecanismes de fallback quan la inferència no és disponible o fiable, i validadors de domini per assegurar integritat.

## 2.5 Seguretat, privadesa i operació

El sistema aplica autenticació per JWT amb expiració, validació estricta d'esquemes amb Pydantic, CORS per domini/subdomini i gestió de secrets via variables d'entorn. El disseny de privadesa segueix RGPD: consentiment informat, minimització, pseudonimització quan escau i polítiques de retenció clares. L'aïllament per tenant evita qualsevol filtració entre mercats i simplifica el compliment. L'operació quotidiana es facilita amb scripts de desplegament, logs per servei, reinicis selectius i reconstrucció d'imatges; la modularitat permet escalar components de forma independent (p. ex., el processador d'IA en pics d'escaneig).

## 2.6 Previsions d'ús, escalabilitat i decisions d'IA

El sistema està pensat per aguantar bé els patrons reals d'ús dels mercats: l'activitat es concentra especialment en caps de setmana i franges de matí, on l'escaneig o l'emissió de tiquets s'intensifica i la latència percebuda ha de ser baixa perquè el procés al punt de venda sigui fluid. En canvi, les campanyes del CRM i part de l'anàlisi es poden desplaçar a hores valls, i les consultes d'historial i panells mostren un patró més irregular, amb repunts immediatament després de compres o notificacions. Sobre aquesta realitat, la plataforma s'ha construït amb contenidors Docker i serveis lleugers perquè escalar no impliqui canvis de codi ni interrupcions llargues: es poden alçar més rèpliques dels serveis que ho requereixin i mantenir un punt d'entrada únic amb Nginx. La separació de responsabilitats és clau; el processador d'IA, que concentra la feina pesada, es pot dimensionar de manera independent del servei de tiquets o de l'autenticació. A nivell de robustesa, s'han introduït controls com la idempotència per `ticket_id` per evitar duplicats, timeouts i reintents acotats, i una validació estricta d'esquemes perquè la informació que arriba a la base de dades (PostgreSQL) sigui consistent. En l'etapa actual, una instància de PostgreSQL ben indexada és suficient; a mesura que creixi el volum, és raonable verticalitzar primer i, si cal, incorporar rèpliques de lectura. Val la pena remarcar que el desplegament és actualment single-tenant per instància (no hi ha cap `tenant_id` a les taules), i que l'estratègia per servir diversos mercats passa per instàncies separades i configuració per entorn; la multi-tenència plena es contempla com a evolució.

La decisió sobre la IA parteix d'un criteri de modularitat i eficiència, separant clarament la capa de visió (lectura del tiquet) de la textualització (normalització i enriquiment semàntic). En la capa de visió s'utilitza Google Gemini per fer OCR amb consciència de maquetació i extreure estructura (línies d'article, imports, impostos, dates).

El flux s'inicia amb el preprocessament d'imatge local (OpenCV/Pillow); tot seguit s'envia la imatge a Gemini per a resolució/OCR i la generació d'un JSON estructurat, i s'apliquen validacions de domini abans de persistir la informació. La textualització i l'enriquiment es resolen en local amb Ollama: s'homogeneïtzen descripcions, s'assignen categories del

catàleg intern i es redacten missatges breus orientats a l'usuari. Aquesta separació permet utilitzar el millor motor en cada tasca, reduir la latència i limitar l'enviament d'informació a tercers estrictament a la imatge necessària per a l'OCR (transmissió xifrada i polítiques de retenció controlades). Un cop consolidada la compra, les dades alimenten el CRM, que construeix perfils i segments (RFM, preferències per categoria, sensibilitat al preu i cohorts de comportament) per personalitzar campanyes i activar la gamificació —incloses les recompenses especials—.

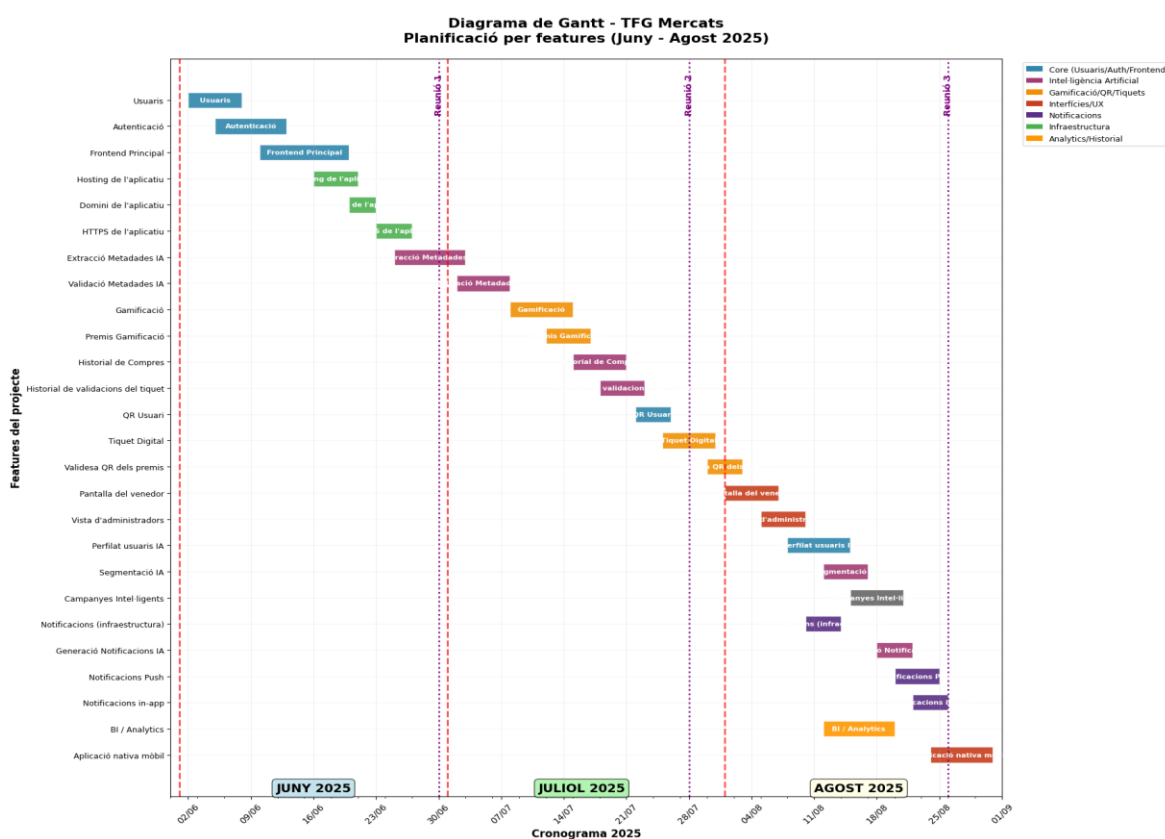
En cas d'incidències (per exemple, indisponibilitat temporal de Gemini), el sistema degrada de manera controlada: posa en cua el processament, permet captura manual guiada de línies crítiques i difereix la textualització fins disposar de l'extracció definitiva, amb reintents i traça d'auditoria.

### 3 Planificació

La planificació s'ha dut a terme amb una metodologia en cascada (waterfall), perquè des de l'inici hi havia requeriments clars, abast tancat i dates de lliurament fixades; per això no s'han utilitzat enfocaments àgils ni sprints, sinó fases seqüencials amb punts d'acceptació formals.

Cada fase s'ha tancat amb una revisió amb el tutor (stage-gate) i només s'ha avançat a la següent un cop obtinguda l'aprovació; el rework derivat d'aquestes revisions s'ha absorbit dins de matalassos temporals planificats segons el risc, de manera que no ha impactat el camí crític ni el calendari.

La traçabilitat s'ha gestionat amb Trello (backlog → in progress → review → done) i un control de canvis conservador amb finestra de code-freeze abans de cada revisió. Per assegurar coherència entre entorns i facilitar les demos, s'han utilitzat Docker Compose i la parametrització via arxius .env, reproduint localment les condicions de producció. Aquesta disciplina de planificació i govern del canvi ha garantit l'alineament amb els requisits i el compliment de les fites compromeses.



### 3.1 Cronograma resum (juny-agost 2025)

La planificació s'estructura en tres blocs principals:

- Juny 2025: Fonaments del sistema: definició d'usuaris i autenticació, desenvolupament del frontend base i muntatge d'infraestructura (hosting, domini, HTTPS). Inici del pipeline d'IA amb l'extracció de metadades (Google Gemini) i preprocés d'imatges.
- Juliol 2025: Cor funcional: validació de metadades, implementació de la gamificació, gestió de premis, historial de compres i validacions, QR d'usuari i tiquet digital. Integració i proves end-to-end dels fluxos de compra i redempció.
- Agost 2025: Finalització i IA avançada: pantalles del venedor i admin, perfilat i segmentació amb IA, campanyes intel·ligents, sistema de notificacions (framework, generació amb IA, push i in-app), BI/Analytics i construcció de l'aplicació nativa mòbil. També s'ha dedicat temps intensiu a polir, corregir bugs i incorporar feedback del professor.

S'han realitzat reunions planificades de revisió amb el professor al tancament de les fites principals del projecte, complementades amb trobades ad hoc per aclarir dubtes i resoldre incidències.

### 3.2 Gestió de dependències i paral·lelització

Les tasques s'han ordenat respectant dependències tècniques (p. ex. autenticació abans d'implementar historial i redempcions) però afavorint el paral·lelisme quan ha estat possible:

- Infraestructura (hosting, domini, HTTPS) paral·lel al frontend per permetre demos estables.
- Pipeline d'IA desenvolupat com a servei independent (ai-ticket-processor) per poder ser dimensionat i provat de manera separada.
- Notificacions, BI i mobile treballats en paral·lel amb les funcionalitats de CRM i segmentació gràcies a l'arquitectura modular de microserveis.

### 3.3 Tancament i lliurament final

Durant les darreres setmanes es va prioritzar polir bugs, fer proves d'acceptació amb escenaris reals (venedor, client, admin) i incorporar el feedback del professor. El lliurament final inclou el codi etiquetat, fitxers Docker Compose per al desplegament i documentació tècnica (arquitectura, API, operacions).

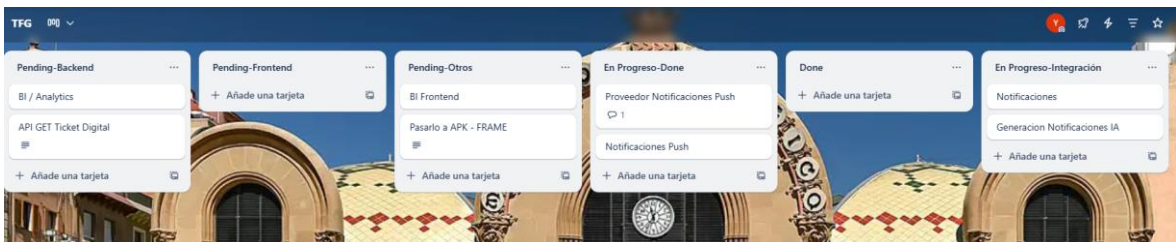
## 4 Metodologia de Treball

S’ha seguit un enfocament iteratiu centrat en features, gestionat amb un tauler de Trello que diferencia clarament: pendents (frontend/backend), en progrés, integració, integrat a dev i integrat a prod, a més de dos fluxos per a bugs (a arreglar/arreglat). Aquest circuit assegura que cada funcionalitat no només s’implementa, sinó que es valida en conjunt i es desplega amb estabilitat, incorporant feedback del professor en punts de control i mesurant qualitat amb proves, contractes d’API i mètriques bàsiques.

### 4.1 Flux de treball amb Trello i estadis de les tasques

He organitzat tot el desenvolupament amb un tauler de Trello que reflecteix l’estat real de cada feature des del moment que es defineix fins al seu desplegament en producció. L’objectiu ha estat doble: d’una banda, donar visibilitat al progrés i a les dependències; de l’altra, assegurar que cada funcionalitat no només queda implementada, sinó també verificada en conjunt amb la resta de l’ecosistema abans d’integrar-se definitivament.

Per començar, mantenia dues columnes de pendents separades per àmbit tècnic: Pending – Frontend i Pending – Backend. Hi entraven totes les funcionalitats amb una definició mínima clara (descripció, API o mòduls afectats i criteris d’acceptació). Quan una feature es prioritzava, passava a En progrés – Done, que és on feia la codificació activa: creació de branca, assignació de tasques i primers commits fins a tenir la funcionalitat implementada localment amb proves unitàries i d’integració bàsiques superades i el canvi llest per integrar.



**Figura 2.** Organització en Trello del TFG

Des d’aquí, la següent etapa era En progrés – Integració. En aquest punt ja compila i ha passat els tests locals, però cal validar-la amb l’stack complet (Docker Compose) i veure com conviu amb la resta de serveis. És on faig proves end-to-end, reviso compatibilitats entre mòduls, ajusto contractes (schemas i endpoints), resolc regressions i estabilitzo el comportament global. La sortida d’aquesta fase és que la feature funciona en conjunt sense trencar fluxos existents i està llesta per integrar-se a l’entorn de desenvolupament.

Quan tot això és estable, passa a Integrat – Dev. Aquí ja queda desplegada en “development” via .env, i faig les validacions funcionals i de rendiment mínimes. Només quan la validació amb dades de prova és completa, he incorporat el feedback i no hi ha errors crítics pendents, avanço a Integrat – Prod. L’entrada a producció implica haver fet una demo o revisió amb el professor, haver passat un check de seguretat (variables d’entorn, CORS, scopes/tokens) i comptar amb observabilitat bàsica. La sortida és un monitoratge sense incidències rellevants i la documentació d’API i d’operació actualitzada.

En paral·lel a aquest flux principal, he mantingut dues columnes específiques per a la qualitat i el manteniment: Bugs – A arreglar i Bugs – Arreglat. A la primera, registro

qualsevol incidència detectada en proves o en ús real, amb descripció clara, passos de reproducció, severitat i component afectat. A la segona, deixo traça de la correcció aplicada i verificada, enllaçant el commit o PR i indicant la versió on queda resolta.

Aquest disseny de columnes m'ha permès separar de manera explícita el moment en què una feature “ja funciona” del moment en què “està verificada en conjunt”. És en aquesta segona fita on realment s'assoleix l'estabilitat del sistema i es minimitzen regressions en integrar noves capacitats.

## 4.2 Integració contínua amb Docker

L'objectiu d'aquesta pràctica ha estat tenir un cicle curt de validació, amb el màxim de realisme possible, però sense dependència d'una infraestructura de CI/CD completa. Per això he utilitzat Docker Compose com a base de l'entorn d'integració local i una disciplina de passos repetibles per a cada feature.

En primer lloc, cada feature es desenvolupa i es prova en local reproduint l'stack complet amb Docker Compose. Això inclou els serveis backend (auth-service, ticket-service, gamification-service, ai-ticket-processor), la base de dades PostgreSQL i, quan cal, el proxy (Nginx) i utilitats de suport. La configuració es governa mitjançant arxius .env específics per entorn, de manera que variables com credencials de BD, claus de serveis externs (p. ex. Gemini), ports i flags d'entorn queden desacoblades del codi. Aquesta reproducció fidel evita “works on my machine” i permet detectar incompatibilitats de dependències, discrepàncies de schemas o problemes de network entre contenidors abans d'integrar.

Quan una feature entra a fase d'integració, executo una bateria de smoke tests orientada a fluxos crítics. No busquen exhaustivitat, sinó verificar ràpid que el sistema no queda trencat: recorregut d'escaneig o emissió de tiquet (des del client/venedor fins a la persistència), assignació de punts i validació de premis en el punt de venda, i consulta d'historials.

La parametrització d'entorns es resol amb un únic stack i dos perfils governats per .env: development i producció. En el perfil de development, l'efecte principal és que els enllaços i endpoints apunten a la versió de desenvolupament (localhost) en lloc de la de producció. A més, existeixen paràmetres per ajustar aspectes com el nivell de log i altres opcions diagnòstiques, però no s'activen automàticament comportaments addicionals (p. ex., seeding o hot-reload). En producció es mantenen optimitzacions i restriccions: logs més discrets i estructurats, timeouts i límits de concurrència ajustats, claus i secrets reals, CORS restringit i configuració de TLS al proxy. El pas d'un entorn a l'altre no requereix canvis de codi: n'hi ha prou amb seleccionar el fitxer .env corresponent i, si cal, executar el docker compose up amb el perfil adequat.

## 5 Requisits del projecte

### 5.1 Requisits funcionals

Els requisits funcionals descriuen, amb claredat què ha de fer la plataforma per donar resposta als objectius del projecte: digitalitzar la gestió de tiquets als mercats municipals, facilitar una experiència còmoda i atractiva per al públic jove, i dotar els venedors i l'administració d'eines que millorin la relació amb la clientela i la dinamització comercial del mercat.

Aquesta especificació s'articula al voltant dels principals actors i dels seus fluxos d'ús. Per a les persones usuàries, el sistema ha de permetre registrar-se, iniciar sessió i gestionar el seu perfil; guardar i consultar els seus tiquets (tant els escanejats com els digitals emesos pels venedors), i visualitzar l'historial de compres amb indicadors útils sobre despesa i hàbits. La pujada de tiquets ha de ser senzilla i fiable, i el seu processament automatitzat ha d'extreure'n la informació rellevant amb prou precisió com per alimentar l'historial i la gamificació sense friccions. Quan hi hagi errors o incerteses, el sistema ha de ser robust i informar-ne de manera entenedora, sense comprometre la integritat de les dades.

Des del punt de vista dels venedors, la plataforma ha de possibilitar l'emissió de tiquets digitals associats a la persona compradora, la validació de recompenses i la consulta d'informació mínima necessària per operar amb agilitat (p. ex., mitjançant QR). Tot això s'ha d'integrar en una experiència àgil en el punt de venda, amb la mínima corba d'aprenentatge i sense interrompre el ritme de l'activitat diària.

Pel que fa als administradors del mercat, cal que puguin supervisar l'evolució de la plataforma: accedir a vistes agregades d'activitat, gestionar rols i venedors, i mantenir el catàleg de botigues del mercat. La traçabilitat —que vincula tiquets, compres i punts de gamificació— és un element central per garantir transparència operativa i possibilitar una auditoria bàsica quan calgui.

Un altre eix fonamental és la gamificació: el sistema ha d'assignar punts i activar dinàmiques de recompensa de manera coherent, incentivant compres a les botigues del mercat i afavorint la recurrència. Aquestes dinàmiques s'han de reflectir de forma clara a la interfície, perquè la persona usuària entengui què ha guanyat, com i per què. Complementàriament, el sistema de notifikacions i el CRM han de permetre una comunicació rellevant i respectuosa amb les preferències de cada usuari, amb segments i campanyes capaces de dinamitzar el mercat sense convertir-se en una font d'intrusivitat.

#### 5.1.1 MoSCoW

La tècnica MoSCoW [7][8] és un mètode de priorització de requisits que classifica cada ítem en quatre categories: Must Have (imprescindible), Should Have (important però posposable), Could Have (desitjable, de baix impacte) i Won't Have this time (fora d'aquesta iteració).

L'emprarem per ordenar i limitar l'abast del projecte, alineant expectatives i recursos amb el calendari: cada requisit de les taules queda anotat amb la seva prioritat MoSCoW, cosa que ens permet decidir què entra segur en la versió actual, què pot esperar sense comprometre l'MVP, quines millores es consideren oportunes si hi ha temps i quins elements queden explícitament exclosos per evitar desviacions de temps.

## 5.2 Requisits de gestió d'usuaris i autenticació

ID i Requisit	Descripció	Prioritat (MoSCoW)
RF-1.1 Registre d'usuaris	El sistema ha de permetre registrar un usuari amb correu i contrasenya; assignarà per defecte el rol "user", i només "vendor" o "admin" quan correspongui i estigui autoritzat.	Must Have
RF-1.2 Inici de sessió	El sistema ha de permetre iniciar sessió amb credencials vàlides i emetre un token JWT amb caducitat configurable.	Must Have
RF-1.3 Verificació i renovació	El sistema ha de validar tokens vigents i permetre'n la renovació per part d'usuaris autenticats.	Must Have
RF-1.4 Informació pública per a QR	El sistema ha d'exposar informació pública mínima d'un usuari (id, email, nom estimat) sense autenticació per a usos de QR al punt de venda.	Must Have
RF-1.5 Llistat d'usuaris	Un administrador ha de poder llistar usuaris amb paginació.	Should Have
RF-1.6 Gestió de rols	Un administrador ha de poder promocionar usuaris a "vendor" o "admin".	Should Have
RF-1.7 Llistat de venedors	Un administrador ha de poder llistar usuaris amb rol "vendor".	Should Have
RF-1.8 Visió general	Un administrador ha de poder consultar mètriques agregades (usuaris totals, venedors, administradors, compres, despesa total).	Must Have

**Taula 1.** Requisits funcionals de gestió d'usuaris i autenticació

## 5.3 Requisits del sistema de captura, processament i consulta de tiquets

ID i Requisit	Descripció	Prioritat (MoSCoW)
---------------	------------	--------------------

RF-2.1 Pujada de tiquets (imatge)	Pujada de tiquets (imatge): el sistema ha de permetre pujar imatges de tiquets; ha de desar el fitxer, crear un registre en estat pendent i associar-hi metadades (mida, tipus MIME, marca temporal).	Must Have
RF-2.2 Tiquets digitals (emissió pel venedor)	Tiquets digitals (emissió pel venedor): un venedor ha de poder emetre tiquets digitals per a un usuari identificat (p. ex., via QR), amb botiga, data, productes i total; aquests tiquets s'aproven automàticament.	Must Have
RF-2.3 Processament amb IA	El sistema ha d'enviar la imatge i el conjunt de noms de botigues del mercat al servei d'IA i rebre'n un resultat estructurat (botiga, productes, totals, data, si és botiga del mercat, etc.).	Must Have
RF-2.4 Detecció de duplicats	El sistema ha de detectar possibles tiquets duplicats (mateixa finestra temporal $\pm 5$ min i coincidència de productes normalitzats) i marcar-los com a "duplicate" per evitar doble còmput.	Must Have
RF-2.5 Estats del tiquet	El sistema ha de gestionar els estats, com a mínim, "pending", "done_approved", "done_rejected", "duplicate", "failed", i actualitzar el resultat de processament associat.	Must Have
RF-2.6 Processament massiu	El sistema ha d'oferir el processament en bloc de tots els tiquets pendents.	Should Have
RF-2.7 Consulta de tiquets	L'usuari ha de poder consultar els seus tiquets amb filtres per estat; el quadre de comandament ha de mostrar preferentment aprovats i digitals.	Could Have
RF-2.8 Historial de tiquets	El sistema ha de permetre llistar l'historial complet de tiquets de l'usuari, diferenciant clarament tiquets digitals i escanejats.	Should Have

RF-2.9 Botigues del mercat	Un operador autoritzat ha de poder crear, actualitzar, eliminar i consultar botigues del mercat; el sistema ha de poder verificar si un nom de botiga pertany al mercat.	Should Have
RF-2.10 Configuració de duplicats (només desenvolupament)	El sistema ha de permetre habilitar/deshabilitar la detecció de duplicats en mode debug.	Could Have

**Taula 2.** Requisits funcionals de captura processament i consulta de tiquets

#### 5.4 Requisits del historial de compres i analítica d'usuari

ID i Requisit	Descripció	Prioritat (MoSCoW)
RF-3.1 Creació de registre de compra	Quan un tiquet es processa correctament (i no és duplicat), el sistema ha de crear un registre de compra per a aquell usuari i tiquet, amb data, botiga, total i línies de producte.	Must Have
RF-3.2 Idempotència per tiquet	El sistema no ha de permetre més d'un registre de compra per ticket_id.	Must Have
RF-3.3 Consulta d'historial	L'usuari autenticat ha de poder consultar el seu historial amb paginació; no pot accedir al d'altres usuaris.	Should Have
RF-3.4 Resum de compres	L'usuari ha de poder obtenir un resum agregat del seu historial de compra.	Should Have
RF-3.5 Despesa per període	L'usuari ha de poder consultar la seva despesa en una finestra temporal (dies) configurable.	Could Have
RF-3.6 Accés administratiu	Un administrador ha de poder consultar l'historial i el resum de compres d'un usuari (suport/operacions).	Could Have

**Taula 3.** Requisits del historial de compres i analítica del usuari

### 5.5 Requisits del sistema de gamificació

ID i Requisit	Descripció	Prioritat (MoSCoW)
RF-4.1 Esdeveniment per tiquet	En processar un tiquet (escanejat o digital), el sistema ha d'enviar un esdeveniment al motor de gamificació amb el resultat (vàlid o no, total, botiga).	Must Have
RF-4.2 Regles de punts	El motor de gamificació ha de calcular punts segons regles (p. ex., si és botiga del mercat i l'import) i mantenir el saldo de l'usuari.	Must Have
RF-4.3 Redempció/validació de premis	El venedor ha de poder validar redempcions mitjançant QR o flux equivalent, comprovant la validesa del premi.	Should Have
RF-4.4 Traçabilitat	S'ha de registrar la relació entre tiquets, punts atorgats i redempcions per a una auditoria bàsica.	Should Have

**Taula 4.** Requisits del sistema de gamificació

### 5.6 Requisits del sistema de notificacions i CRM

ID i Requisit	Descripció	Prioritat (MoSCoW)
RF-5.1 Infraestructura de notificacions	El sistema ha de disposar d'un servei d'enviament amb adaptadors (p. ex., web push, iOS, Android) i gestionar dispositius registrats per usuari.	Should Have
RF-5.2 Campanyes intel·ligents	El sistema ha de permetre definir campanyes adreçades a segments d'usuaris basats en perfilatge/segmentació (IA) i llançar notificacions.	Must Have
RF-5.3 Notificacions push i in-app	El sistema ha de poder enviar notificacions push i mostrar-les in-app; la generació de contingut pot recolzar-se en IA per a personalització.	Could Have

**Taula 5.** Requisits del sistema de notificacions i CRM

### 5.7 Requisits del perfilatge i segmentació amb IA

ID i Requisit	Descripció	Prioritat (MoSCoW)
RF-6.1 Perfilatge d'usuaris	El sistema ha de generar un perfil bàsic (hàbits de compra, botigues freqüents, tiquet mitjà) a partir de l'historial.	Must Have
RF-6.2 Segmentació	El sistema ha d'agrupar usuaris en segments (freqüència, tiquet mitjà, afinitat a botigues o categories) per a campanyes.	Must Have
RF-6.3 Campanyes recomanades	El sistema ha de proposar campanyes o missatges per segment i permetre'n el llançament, amb mesura de taxes d'obertura/conversió.	Could Have

**Taula 6.** Requisits del perfilatge i segmentació amb IA

### 5.8 Requisits de la aplicació web i la aplicació mòbil

ID i Requisit	Descripció	Prioritat (MoSCoW)
RF-7.1 Frontend web	El sistema ha d'oferir una interfície per a usuaris (dashboard, tiquets, punts, premis), venedors (validació/emissió) i administradors (overview, gestió bàsica).	Must Have
RF-7.2 App mòbil nativa	S'ha de disposar d'una app mòbil nativa que permeti, com a mínim, gestionar el QR d'usuari, visualitzar tiquets i recompenses, i rebre notificacions push.	Should Have

**Taula 7.** Requisits de la aplicació web i aplicació mòbil

### 5.9 Requisits no funcionals

ID i Requisit	Descripció	Prioritat (MoSCoW)
RNF-1.1	Les operacions interactives (consulta de tiquets, historial, punts) han de respondre	Could Have

	habitualment en menys de 500 ms sota càrrega normal.	
RNF-1.2	Els serveis crítics s'han de poder escalar horitzontalment per contenidors, sense canvis de codi.	Should Have
RNF-1.3	Disponibilitat objectiu del servei en producció $\geq 99,5\%$ .	Could Have
RNF-1.4	Operacions críiques idempotents per clau natural (p. ex., un sol registre de compra per ticket_id).	Must Have
RNF-1.5	Transicions d'estat dels tiquets consistents i traçables; cap error ha de deixar registres en estat incoherent.	Could Have
RNF-1.6	Autenticació per token amb expiració i control d'accessos per rol (usuari, venedor, admin).	Should Have
RNF-1.7	Comunicacions xifrades (HTTPS) i gestió de secrets via variables d'entorn; CORS limitat a orígens de confiança.	Could Have
RNF-1.8	Minimització de dades personals i respecte a preferències de comunicació; registre bàsic d'accions administratives per auditoria.	Must Have

**Taula 8.** Requisits no funcionals

## 5.10 Restriccions i limitacions del projecte

### 5.10.1 Abast i dependències

L'abast inicial se centra en digitalització de tiquets, historial de compres, gamificació bàsica i notificacions/CRM essencials. Funcions avançades (recomanadors sofisticats, analítica semàntica) queden fora. Es pressuposa un entorn amb Docker, Nginx i Postgres; altres plataformes poden requerir ajustos.

### 5.10.2 Processament amb IA i qualitat de dades

L'extracció depèn de la qualitat de la imatge i de la variabilitat dels tiquets, amb possibles errors o incerteses. La normalització de productes és limitada i pot afectar l'analítica. El temps de processament pot allargar-se en pics i requerir cues i escalat específic.

### **5.10.3 Duplicats i coherència**

La detecció de duplicats és heurística (finestra temporal i coincidència de productes) i pot produir falsos positius/negatius. S'aplica idempotència en operacions clau (p. ex., un sol registre de compra per tiquet) per preservar la coherència.

### **5.10.4 Multimercat i aïllament**

Tot i que la solució està pensada per ser adaptable a diversos mercats, l'edició actual no incorpora un identificador de llogater (`tenant_id`) transversal a dades i serveis. Això limita l'aïllament complet entre mercats quan conviuen en una mateixa instància. La consolidació d'un multilloguer robust exigiria una evolució específica de l'esquema de dades i de les interfícies internes.

### **5.10.5 Notificacions i CRM**

L'eficàcia de les notificacions depèn d'integracions amb plataformes de tercers i de permisos al dispositiu de l'usuari. En entorns com iOS hi ha restriccions que poden reduir la cobertura real. Igualment, la segmentació i el perfilatge proporcionen un valor inicial, però el seu abast és deliberadament acotat per equilibrar simplicitat i utilitat; per assolir estratègies més fines caldria enriquir dades i models.

### **5.10.6 Experiència al punt de venda i operativa quotidiana**

L'emissió de tiquets digitals, la validació de recompenses i l'ús de codis QR demanen una participació consistent per part de venedors i compradors. En contextos amb connectivitat irregular o amb dispositius limitats, l'experiència pot veure's afectada i caldrà contemplar pràctiques operatives de suport (reintents, processos alternatius temporals). Així mateix, els tiquets o compres anteriors a la posada en marxa no es poden reconstruir automàticament: la traçabilitat es construeix a partir del moment en què el sistema entra en funcionament.

### **5.10.7 Compliment i govern de dades**

El compliment normatiu en matèria de privadesa no depèn només del programari. Calen processos organitzatius per a la gestió de consentiments, polítiques de retenció i mecanismes d'exercici de drets (com el dret d'oblit). Sense aquests processos, el grau de compliment es manté parcial. A nivell tècnic, s'aplica minimització de dades i traçabilitat bàsica d'accions administratives, però la plena adequació requereix també mesures de govern i procediments operatius associats.

## **5.11 Casos d'ús**

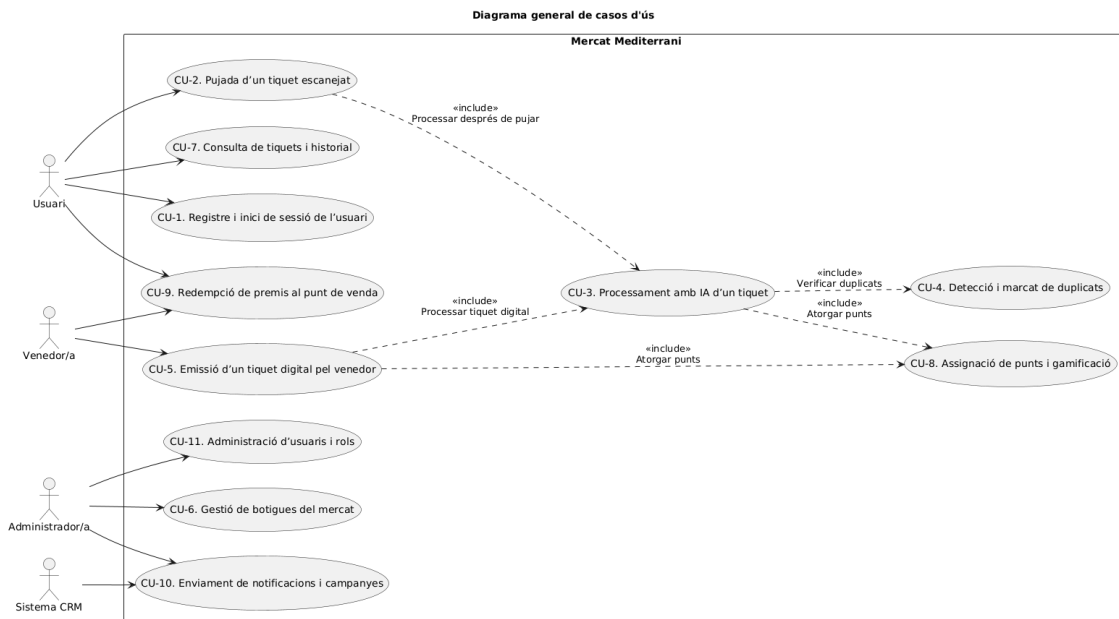
Els casos d'ús descriuen les principals interaccions entre els diferents actors i el sistema, especificant les funcionalitats que aquest ha de proporcionar per satisfer les seves necessitats. Aquest enfocament permet entendre de manera estructurada el comportament esperat del sistema des del punt de vista dels usuaris i ajuda a concretar els requisits i les proves d'acceptació.

En el cas del projecte, s'identifiquen tres actors principals:

Persona usuària (client): Comprador/a que utilitza l'aplicació per gestionar el seu compte, pujar tiquets, consultar l'historial i participar en la gamificació.

Venedor/a (punt de venda): Professional del mercat que emet tiquets digitals, valida redempcions i consulta informació operativa mínima (p. ex., via QR).

Administrador/a del mercat: Usuari amb accés ampli per supervisar l'activitat, gestionar rols i mantenir el catàleg de botigues.



**Figura 3.** Diagrama general de casos d'ús

**5.11.1 CU-1. Registre i inici de sessió de l'usuari**

Element	Descripció
Resum	Permet a la persona usuària crear un compte amb correu i contrasenya i iniciar sessió per obtenir accés a l'aplicació.
Paràmetres d'entrada	Email, contrasenya; confirmació de contrasenya (registre).
Paràmetres de sortida	Confirmació de compte creada; token d'accés vàlid; dades bàsiques de perfil inicial.
Actors	Persona usuària.
Precondició	El correu no ha d'estar registrat prèviament; servei d'autenticació operatiu.
Postcondició	Sessió activa amb token vàlid; perfil inicialitzat.
Procés normal	1) Obrir "Registrar". 2) Introduir email i contrasenyes. 3) Crear compte. 4) El sistema valida i crea. 5) Redirigeix a iniciar sessió (si escau). 6) Iniciar sessió i

	rebre token. 7) Mostrar tauler inicial.
Alternatives i excepcions	Email existent → error i suggeriment de login; validacions fallides → missatge; credencials incorrectes → reintent.

Taula 9. Registre i inici de sessió d'usuari

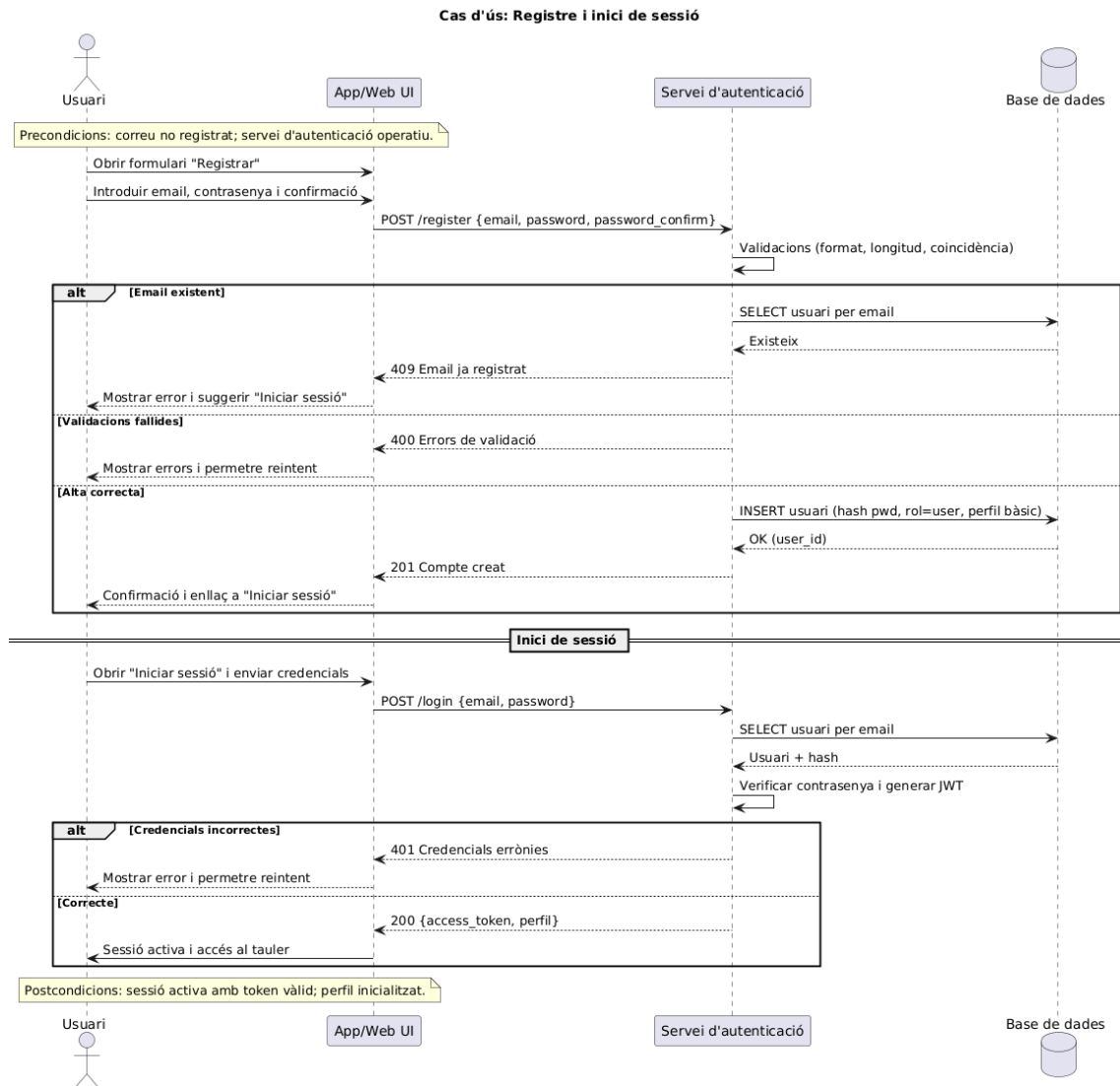


Figura 4. Diagrama del CU-1

5.11.2 CU-2. Pujada d'un tiquet escanejat

Element	Descripció
Resum	La persona usuària puja la imatge d'un tiquet, que queda en estat pending i inicia el processament amb IA.
Paràmetres d'entrada	Fitxer d'imatge; user_id; metadades (mida, mime).

Paràmetres de sortida	Confirmació de pujada; identificador del tiquet i estat pending.
Actors	Persona usuària.
Precondició	Sessió iniciada; fitxer vàlid disponible.
Postcondició	Tiquet creat a la base de dades; imatge emmagatzemada; estat pending.
Procés normal	1) Obrir "Pujar tiquet". 2) Seleccionar imatge. 3) Pujar. 4) El sistema desa i crea el registre. 5) Retorna confirmació i estat pending.
Alternatives i excepcions	Fitxer invàlid/corrupt → error; fallada d'emmagatzematge → error i no es crea el tiquet.

Taula 10. Pujada d'un tiquet escanejat

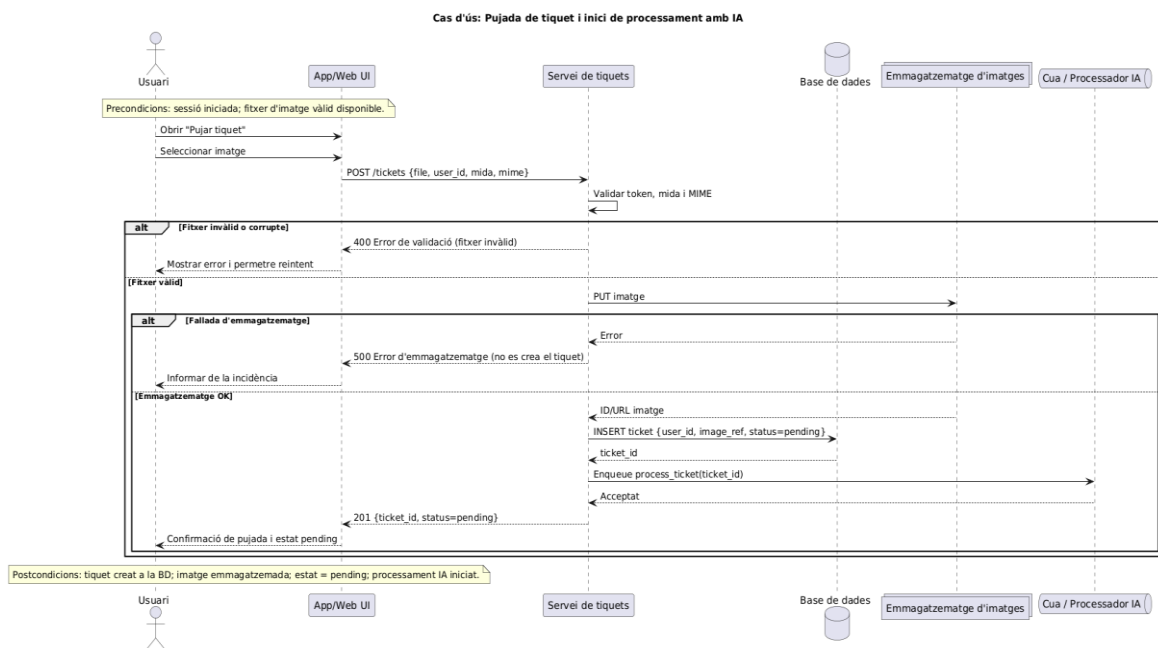


Figura 5. Diagrama del CU-2

### 5.11.3 CU-3. Processament amb IA d'un tiquet

Element	Descripció
Resum	El sistema envia la imatge i el llistat de botigues del mercat al servei d'IA i actualitza l'estat segons el resultat.
Paràmetres d'entrada	ticket_id; imatge; llista de botigues del mercat.

Paràmetres de sortida	Resultat estructurat (botiga, productes, total, data, validació) i nou estat.
Actors	Sistema.
Precondició	Tiquet en estat pending; servei d'IA disponible.
Postcondició	Tiquet en done_approved, done_rejected, duplicate o failed amb resultat associat.
Procés normal	1) Llançar processament. 2) Enviar a IA. 3) Rebre resultat. 4) Validar esquema. 5) Actualitzar estat i desar.
Alternatives i excepcions	Timeout/indisponibilitat d'IA → failed; resultat inconsistent → failed i incidència registrada.

Taula 11. Processament amb IA d'un tiquet

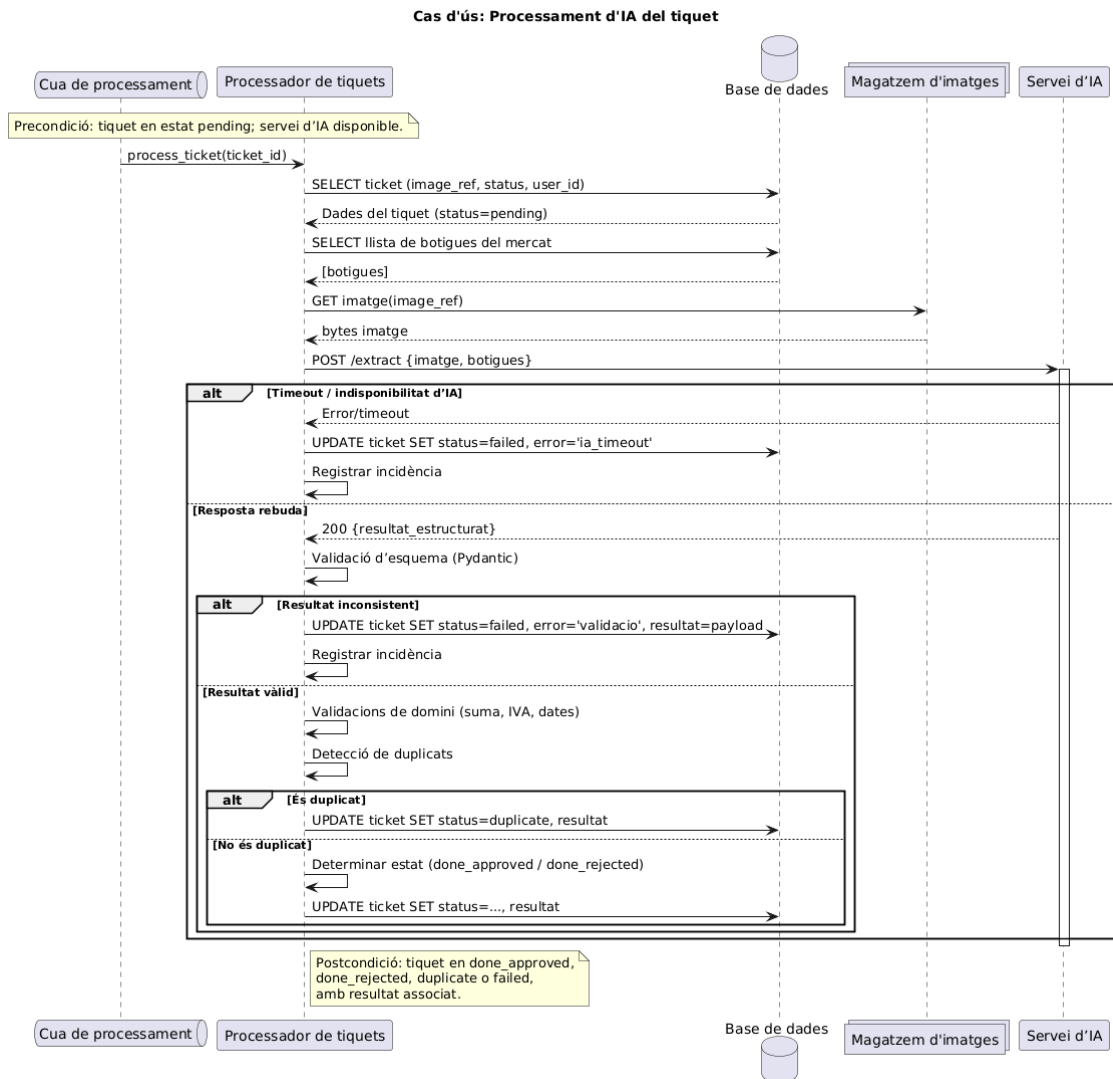


Figura 6. Diagrama del CU-3

**5.11.4 CU-4. Detecció i marcat de duplicats**

<b>Element</b>	<b>Descripció</b>
Resum	Identifica si un tiquet processat és duplicat comparant finestra temporal i coincidència de productes.
Paràmetres d'entrada	Dades del tiquet (data, productes); user_id; historial recent.
Paràmetres de sortida	Senyalització de duplicat; estat duplicate.
Actors	Sistema.
Precondició	Resultat de processament disponible; verificació de duplicats habilitada.
Postcondició	Tiquet marcat duplicate si escau; s'evita doble còmput.
Procés normal	1) Obtenir data i productes. 2) Cercar tiquets del mateix usuari en finestra $\pm 5$ minuts. 3) Comparar productes normalitzats. 4) Si coincideix, marcar duplicate.
Alternatives i excepcions	Sense data/productes $\rightarrow$ ometre comprovació; error de parseig de data $\rightarrow$ ometre i registrar avís.

**Taula 12.** Detecció i marcat de duplicats

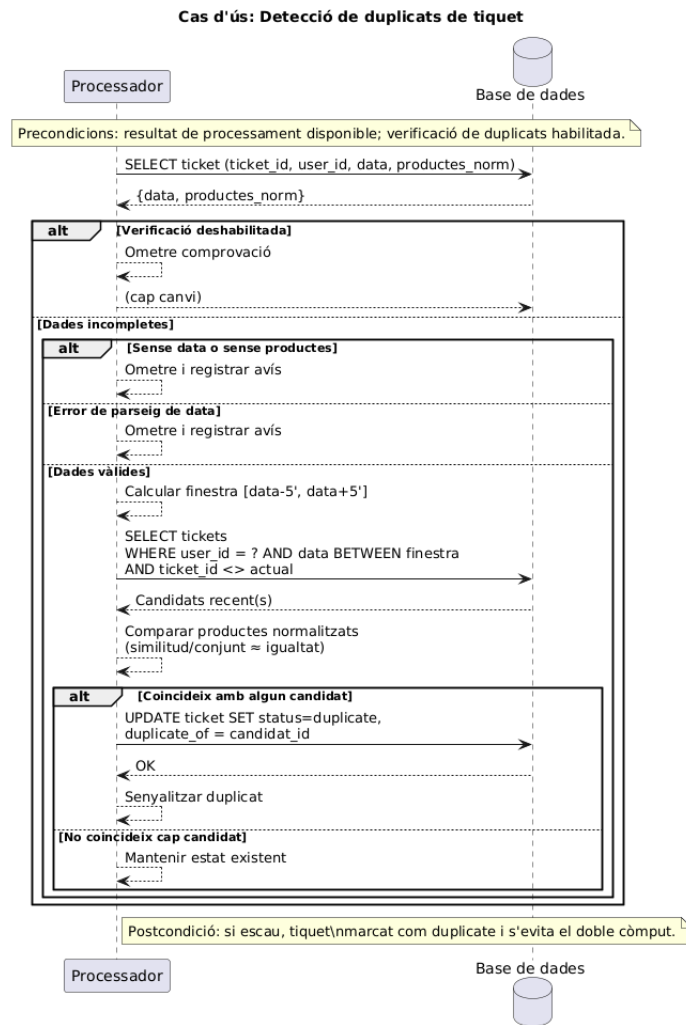


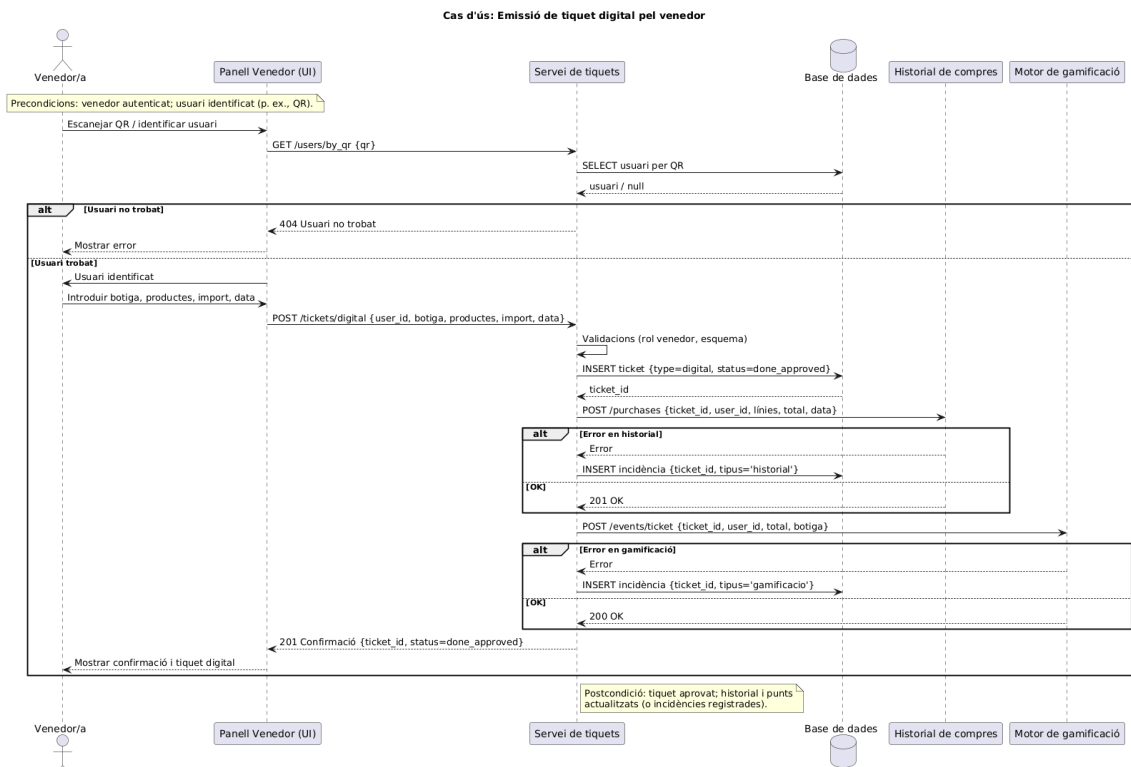
Figura 7. Diagrama del CU-4

5.11.5 CU-5. Emissió d'un tiquet digital pel venedor

Element	Descripció
Resum	El venedor emet un tiquet digital associat a una persona usuària identificada; s'aprova automàticament.
Paràmetres d'entrada	Identificador d'usuari (p. ex., QR), botiga, productes, import, data.
Paràmetres de sortida	Tiquet digital creat; confirmació; actualització d'historial i punts.
Actors	Venedor/a.
Precondició	Venedor autènticat; usuari identificat.
Postcondició	Tiquet en estat aprovat; historial i gamificació actualitzats.

Procés normal	1) Identificar usuari. 2) Introduir dades. 3) Emetre. 4) Crear tiquet i actualitzar serveis.
Alternatives i excepcions	Usuari no trobat → error; fallada en actualitzar historial/punts → es registra incidència.

**Taula 13.** Emissió d'un tiquet digital pel venedor



**Figura 8.** Diagrama del CU-5

**5.11.6 CU-6. Gestió de botigues del mercat**

Element	Descripció
Resum	L'administrador manté el catàleg de botigues (alta, edició, baixa) i verifica pertinença al mercat.
Paràmetres d'entrada	Dades de botiga (nom, identificadors); operació.
Paràmetres de sortida	Confirmació d'alta/edició/baixa; resultat de verificació.
Actors	Administrador/a.
Precondició	Administrador autènticat.
Postcondició	Catàleg actualitzat; verificació disponible.

Procés normal	1) Llistar botigues. 2) Afegir/editar/eliminar. 3) Desar i verificar.
Alternatives i excepcions	Botiga duplicada → error; fallada de persistència → missatge i reintent.

Taula 14. Gestió de botigues del mercat

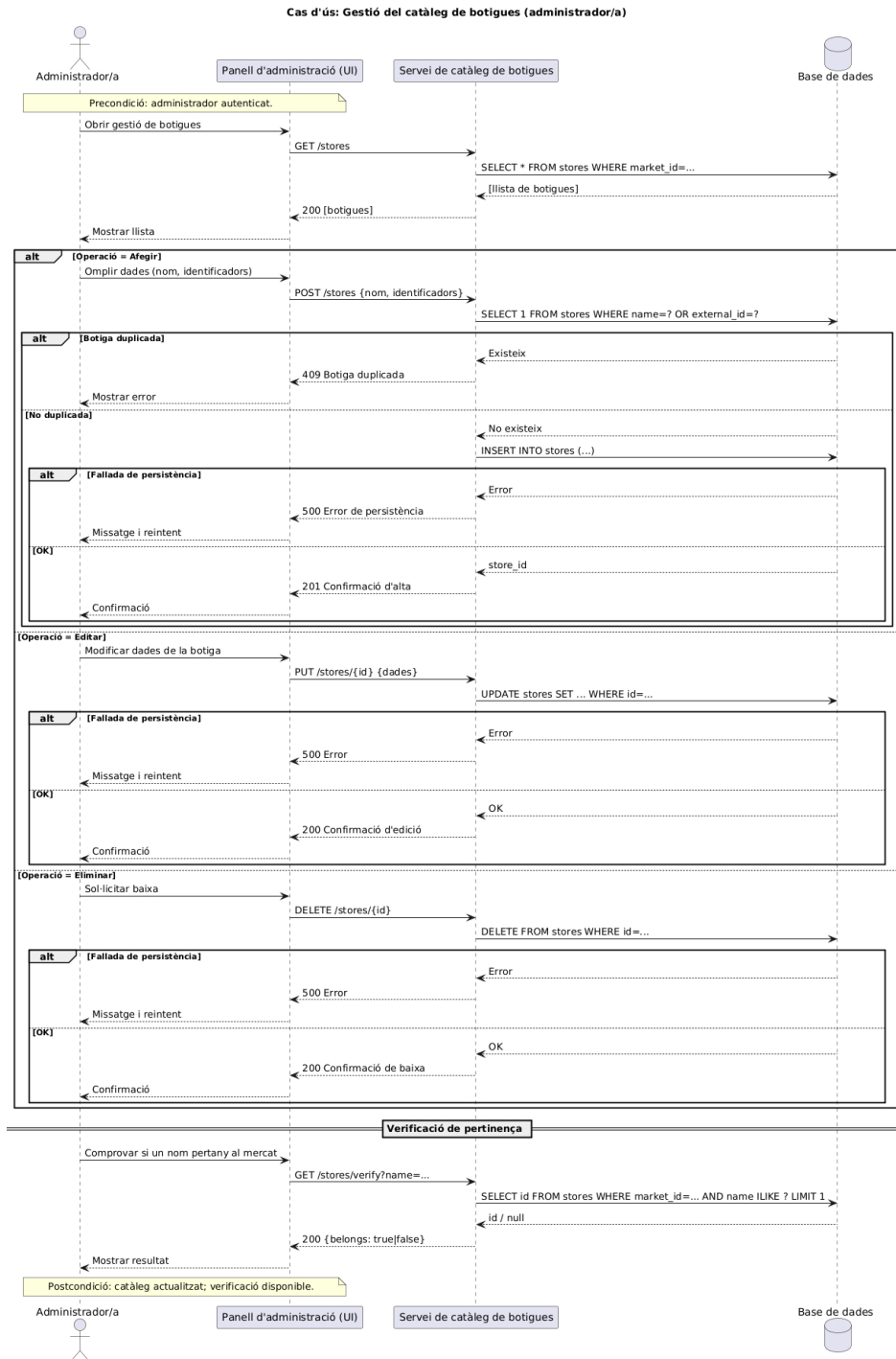


Figura 9. Diagrama del CU-6

5.11.7 CU-7. Consulta de tiquets i historial

Element	Descripció
Resum	La persona usuària consulta el llistat de tiquets i l'historial amb filtres i detalls.
Paràmetres d'entrada	Token d'usuari; filtres (estat, període).
Paràmetres de sortida	Llistat de tiquets; detall; resums d'historial.
Actors	Persona usuària.
Precondició	Sessió iniciada.
Postcondició	Informació mostrada; sense canvis d'estat.
Procés normal	1) Obrir "Tiquets/Historial". 2) Aplicar filtres. 3) Veure detalls.
Alternatives i excepcions	Sessió caducada → demanar login; sense resultats → missatge informatiu.

Taula 15. Consulta de tiquets i historial

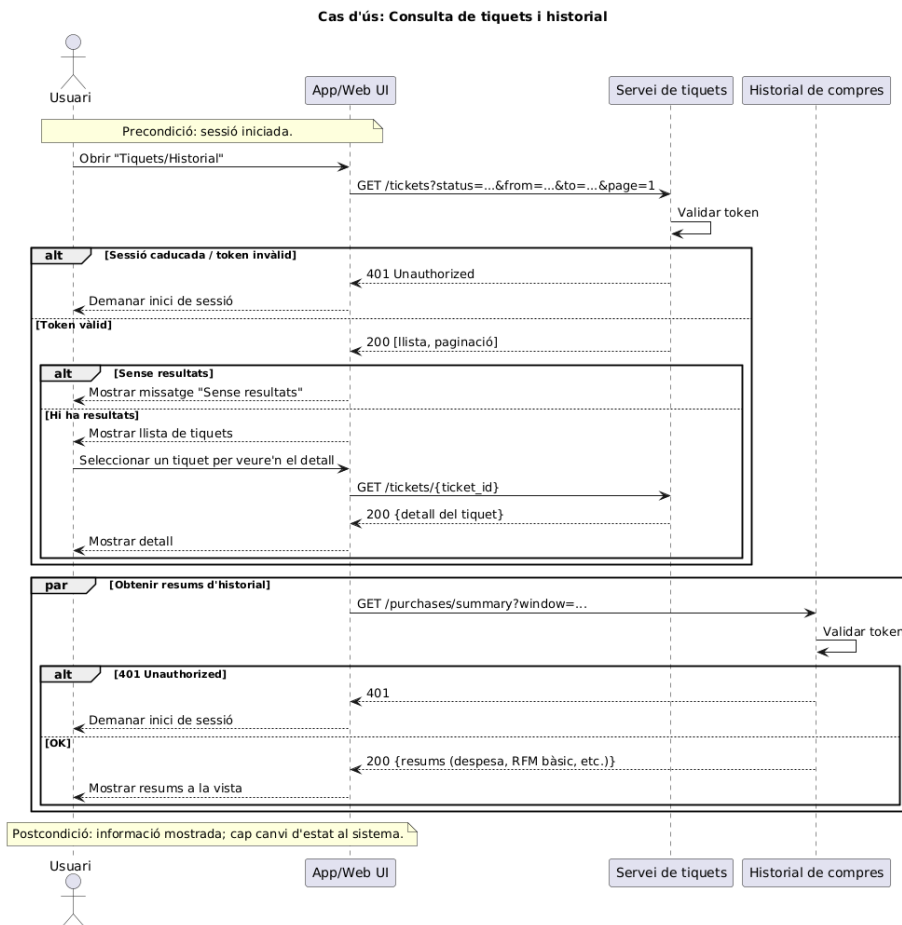


Figura 10. Diagrama del CU-7

5.11.8 CU-8. Assignació de punts i gamificació

Element	Descripció
Resum	Després de processar un tiquet vàlid, s'actualitza el saldo de punts i, si escau, premis.
Paràmetres d'entrada	user_id, ticket_id, import, botiga, validesa.
Paràmetres de sortida	Nou saldo de punts; confirmació de l'esdeveniment.
Actors	Sistema.
Precondició	Tiquet processat i no duplicat.
Postcondició	Saldo actualitzat i traça vinculada al tiquet.
Procés normal	1) Generar esdeveniment. 2) Calcular punts. 3) Actualitzar saldo.
Alternatives i excepcions	Error de comunicació amb el motor → reintents i registre d'error.

Taula 16. Assignació de punts i gamificació

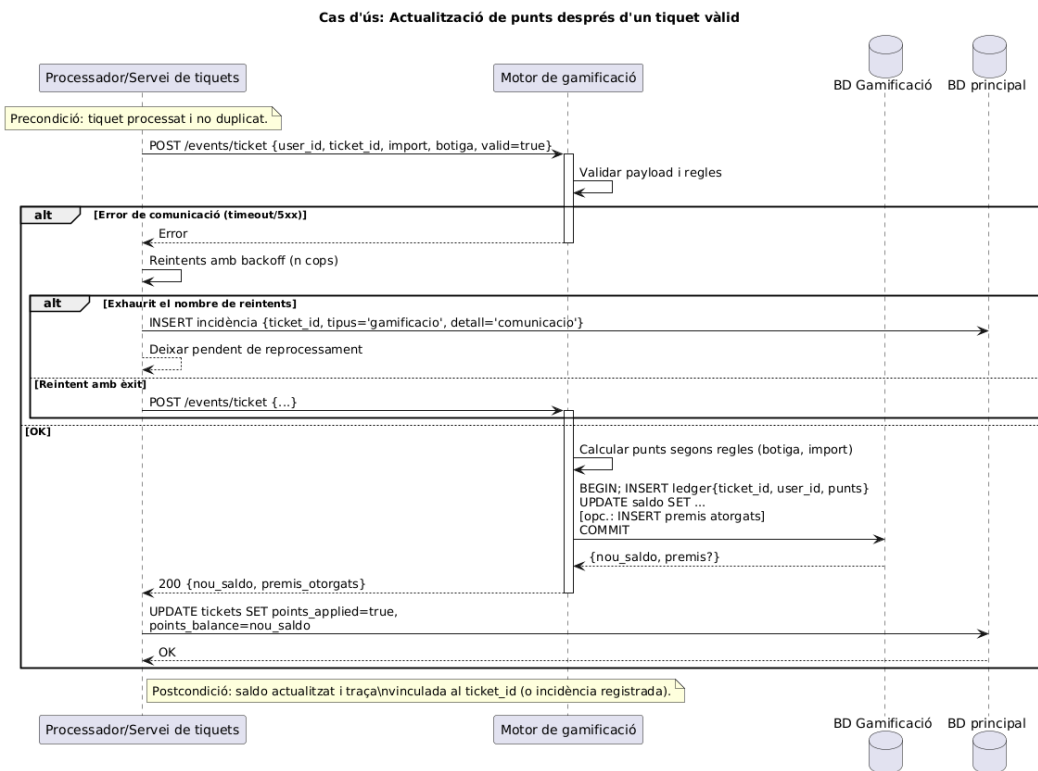


Figura 11. Diagrama del CU-8

5.11.9 CU-9. Redempció de premis al punt de venda

Element	Descripció
Resum	La persona usuària redimeix un premi i el venedor n'en valida l'elegibilitat.
Paràmetres d'entrada	QR/codi de l'usuari; premi.
Paràmetres de sortida	Confirmació de redempció o motiu de denegació.
Actors	Usuari i Venedor/a.
Precondició	Usuari amb punts i premi disponible; venedor autènticat.
Postcondició	Premi marcat com redimit; traça registrada.
Procés normal	1) Mostrar QR/codi. 2) Validar elegibilitat. 3) Confirmar redempció.
Alternatives i excepcions	Punts insuficients o premi caducat → denegar; error de xarxa → reintent.

Taula 17. Redempció de premis al punt de venda

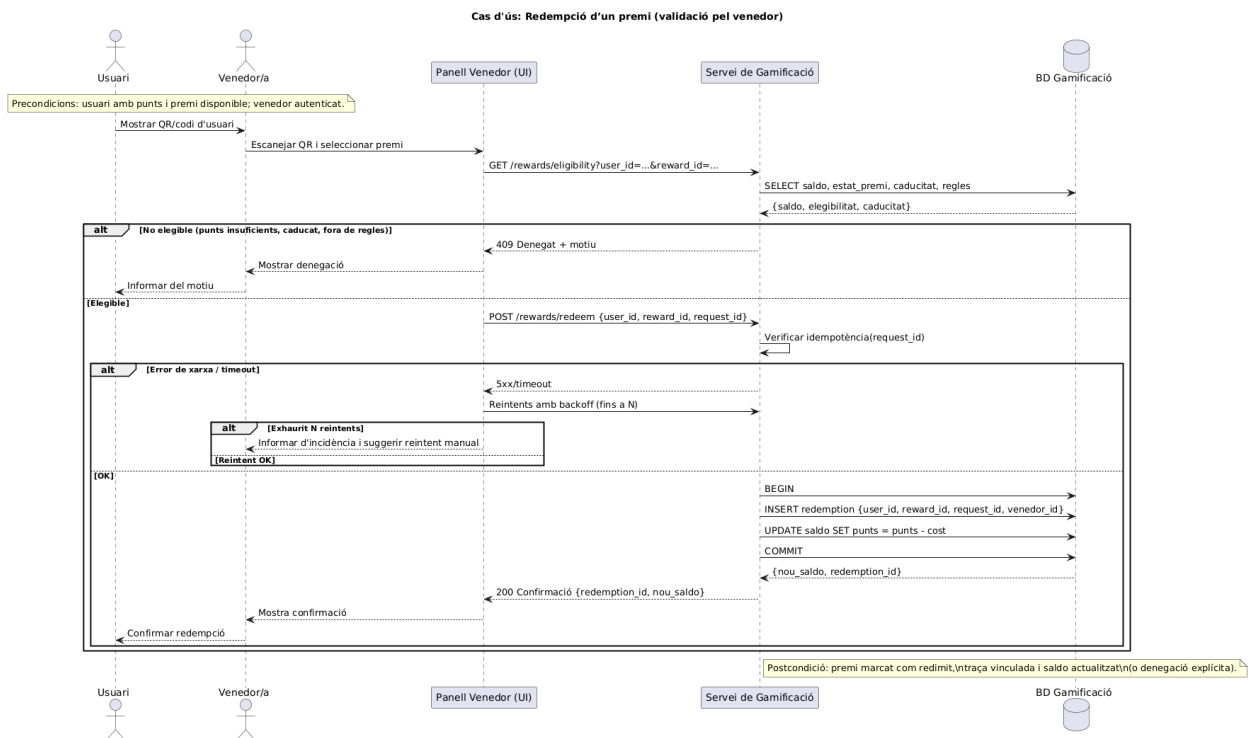
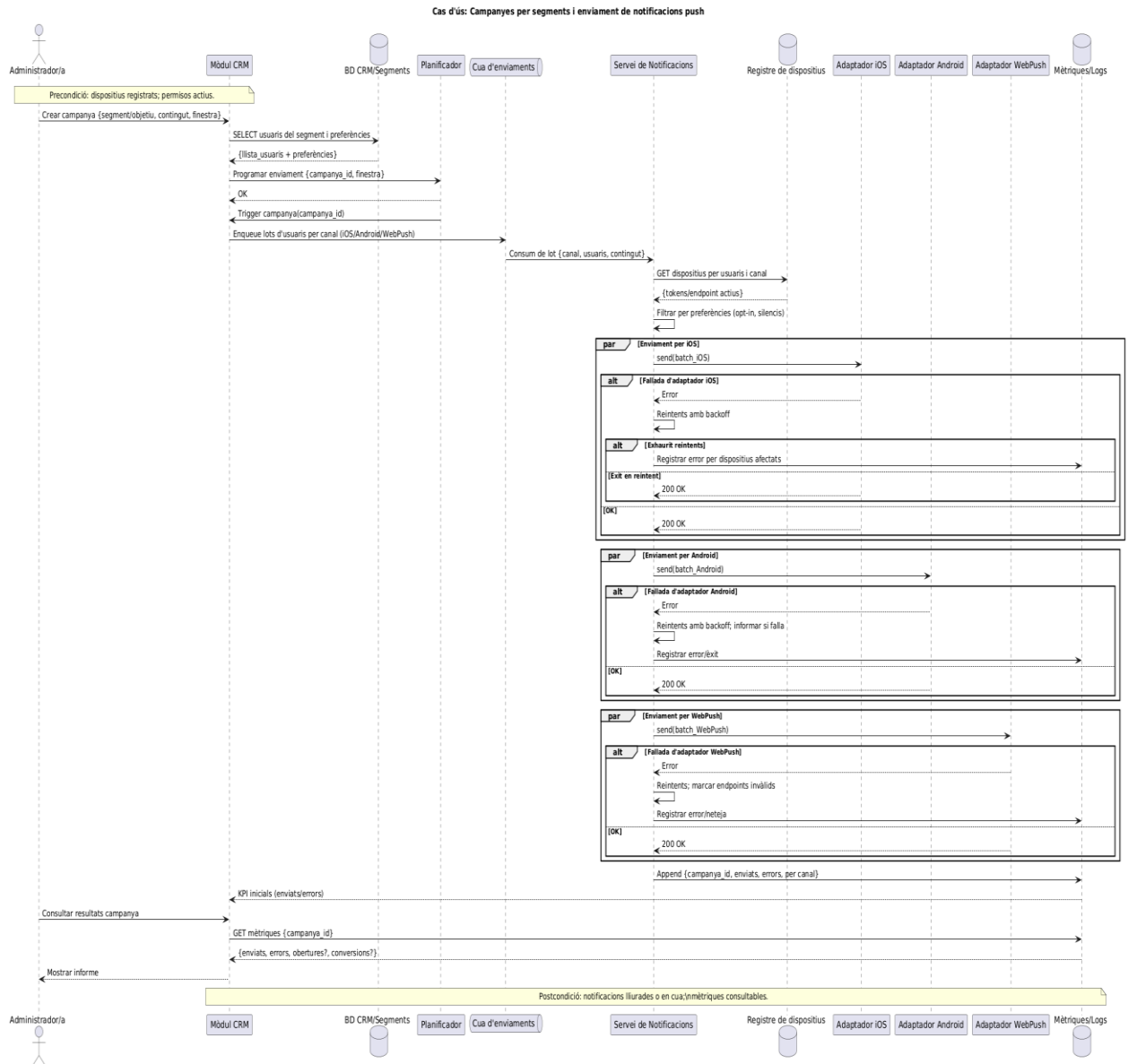


Figura 12. Diagrama del CU-9

**5.11.10 CU-10. Enviament de notificacions i campanye**

<b>Element</b>	<b>Descripció</b>
Resum	Es defineixen campanyes per segments i s'envien notificacions push segons preferències.
Paràmetres d'entrada	Segment/objectiu, contingut, finestra d'enviament.
Paràmetres de sortida	Resultat d'enviament; mètriques bàsiques.
Actors	Administrador/a o sistema CRM.
Precondició	Dispositius registrats; permisos actius.
Postcondició	Notificacions lliurades o en cua; mètriques consultables.
Procés normal	1) Seleccionar segment. 2) Redactar i programar. 3) Enviar i monitoritzar.
Alternatives i excepcions	Fallada d'adaptador (iOS/Android/WebPush) → reintents i informe d'errors.

**Taula 18.** Enviament de notificacions i campanyes



**Figura 13.** Diagrama del CU-10

**5.11.11 CU-11. Administració d'usuaris i rols**

<b>Element</b>	<b>Descripció</b>
Resum	L'administrador llista usuaris, canvia rols i consulta una visió agregada d'activitat i compres.
Paràmetres d'entrada	Operació (llistar/promoure), identificador d'usuari.
Paràmetres de sortida	Llista paginada; confirmació de canvi de rol; mètriques d'overview.
Actors	Administrador/a.
Precondició	Administrador autènticat amb permisos.
Postcondició	Rols actualitzats; mètriques accessibles.
Procés normal	1) Obrir "Usuaris". 2) Llistar i cercar. 3) Seleccionar usuari i promoure
Alternatives i excepcions	Usuari inexistent → error; falta de permisos → denegar i registrar.

**Taula 19.** Administració d'usuaris i rols

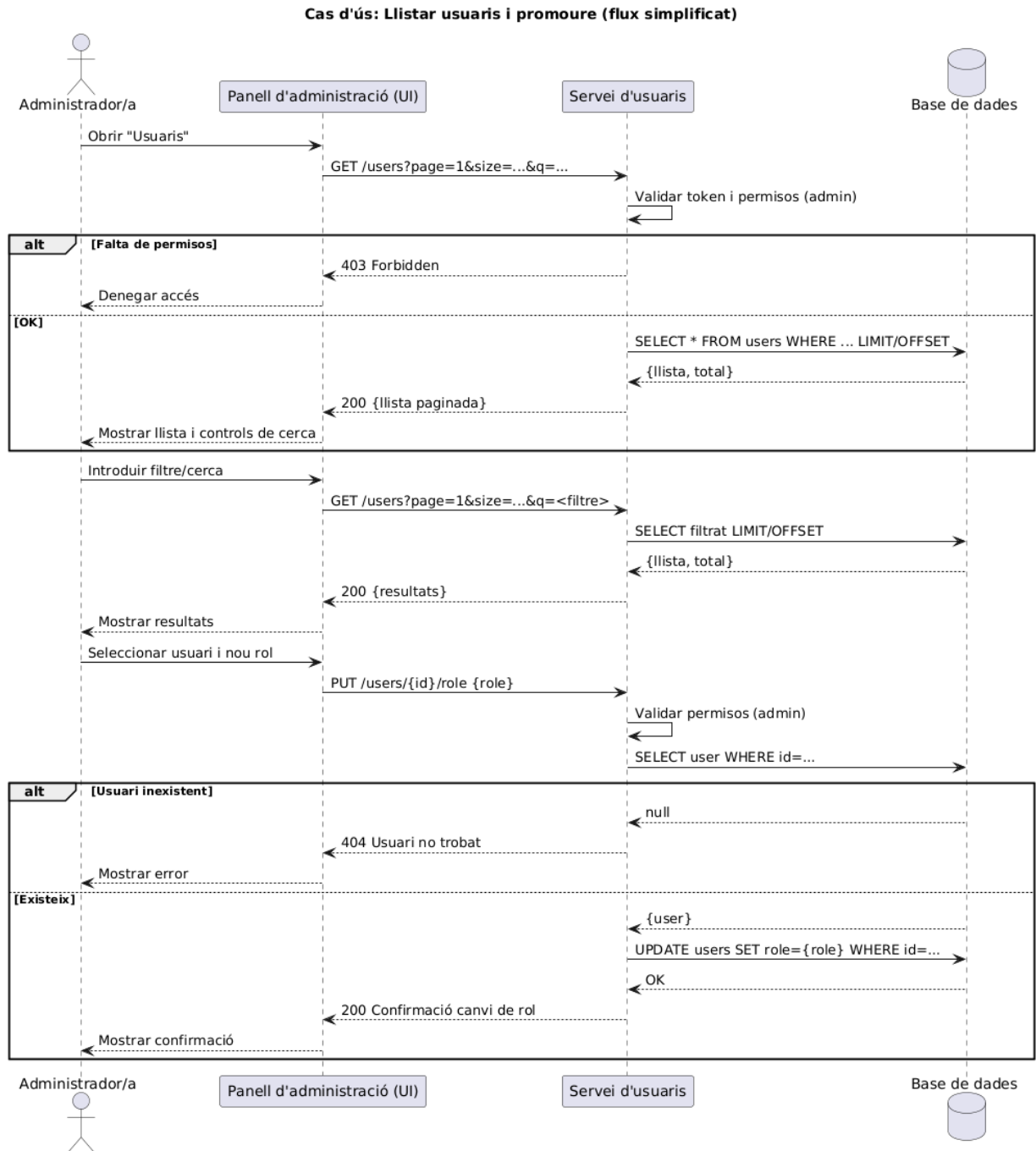


Figura 14. Diagrama del CU-11



## 6 Disseny

Aquest capítol defineix el disseny tècnic del sistema: les peces que el componen, els contractes entre serveis i els invariants d'integritat que en garanteixen el funcionament. El backend s'exposa mitjançant APIs REST amb autenticació per token i rols; els tiquets segueixen un cicle amb estats i idempotència; i els processos de suport (p. ex., IA i notifikacions) s'executen de manera asíncrona. En persistència es fixa un model de domini amb claus UUID i marques temporals, ús acotat de camps JSON i separació entre la "font" (Ticket) i la projecció normalitzada (PurchaseHistory) per assegurar coherència temporal i reprocessament controlat. El client consumeix aquestes APIs i mostra feedback d'estat del processament.

La configuració operativa es parametritza amb fitxers .env. El perfil de desenvolupament redirigeix enllaços i endpoints a entorns locals (localhost); existeixen paràmetres addicionals (com el nivell de log), però el canvi d'entorn no requereix canvis de codi. A la següent secció es presenta la visió d'arquitectura d'alt nivell, centrada en l'ús de Nginx com a proxy invers i en l'organització en microserveis.

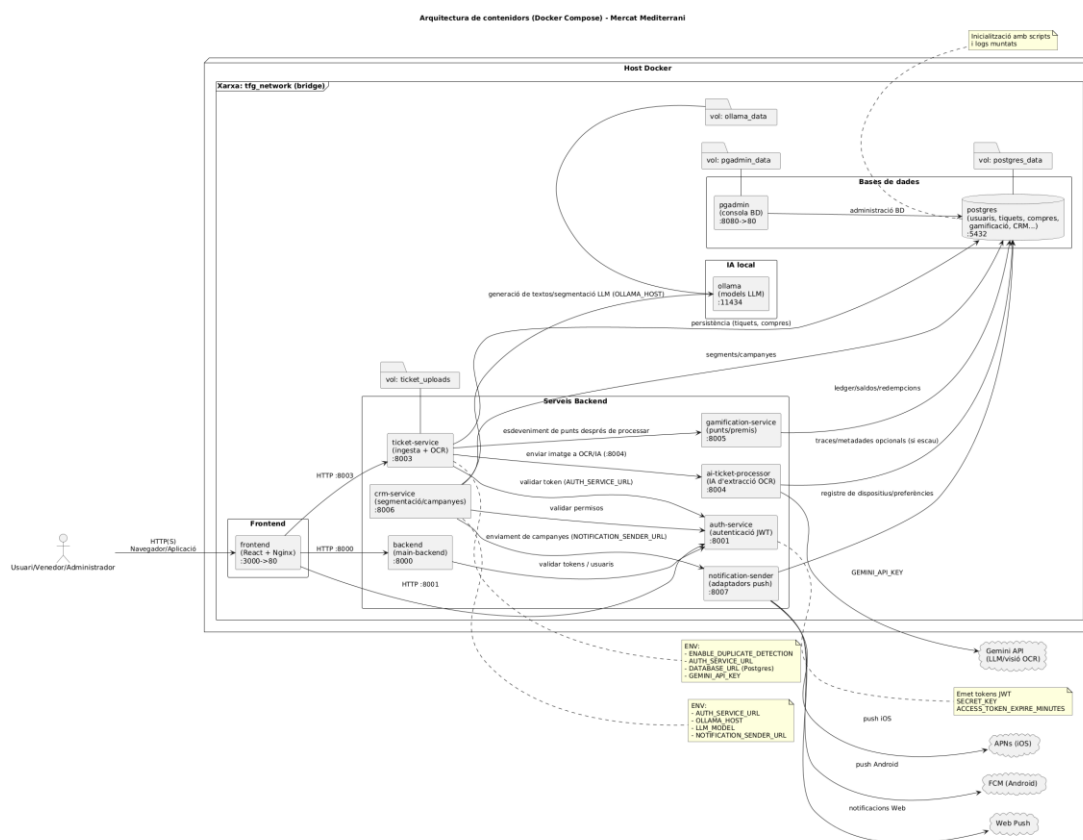


Figura 16. Diagrama de l'arquitectura de l'aplicació

### 6.1 Arquitectura general de l'aplicatiu

L'aplicació s'organitza com un conjunt de microserveis desplegats amb Docker Compose i comunicats per HTTP dins la xarxa tfg\_network. L'exposició cap a l'exterior es fa mitjançant Nginx com a proxy invers: en desenvolupament, el contenidor del frontend ja serveix els estàtics amb Nginx als ports 80 i 443 i el client apunta a endpoints locals (localhost); en producció es tracta d'un Nginx extern amb terminació TLS que entrega el

frontend i dirigeix les rutes d'API cap als serveis interns. El canvi d'entorn es governa amb `.env` (sense tocar codi): el client canvia d'enllaços `dev`→`prod` i la resta de serveis conserven els seus contractes HTTP.

```
# Configuración del entorno de producción
REACT_APP_ENVIRONMENT=development
```

**Figura 17.** Paràmetre d'entorn a les variables d'entorn

```
# =====
# FRONTEND
# =====

frontend:
  build:
    context: ./modules/frontend
    dockerfile: Dockerfile
    args:
      - REACT_APP_ENVIRONMENT=${REACT_APP_ENVIRONMENT}
```

**Figura 18.** Mapeig de la variable d'entorn al contenidor Docker

## 6.2 Microserveis de l'aplicatiu

**frontend (3000→80):** SPA React compilada i servida per Nginx dins del contenidor. El build rep `REACT_APP_ENVIRONMENT` per apuntar a endpoints de `dev` o `prod`, i confia que el proxy extern ruti les rutes d'API (`/api/*`) cap als microserveis. Inclou healthcheck a `/` per validar disponibilitat.

**backend (8000):** API agregada que ofereix endpoints transversals i coordinació bàsica entre dominis. Llegeix `DATABASE_URL` i `SECRET_KEY` de `.env` i exposa `/health` per monitoratge. Serveix com a façana simple per a funcionalitats que combinen dades de diversos serveis.

**auth-service (8001):** Servei d'autenticació/autorització amb JWT i gestió de rols. Dona d'alta usuaris/vendedors, valida tokens i actua com a autoritat d'identitat per a la resta de serveis. Persistència en PostgreSQL i healthcheck propi per arrencada ordenada.

**ticket-service (8003):** Ingesta i gestió de tiquets (pujada d'imatges a volum `ticket_uploads`), control d'estat i idempotència (p. ex., per hash). Integra amb `auth-service` per validar usuaris i amb `ai-ticket-processor` per a l'extracció OCR/IA; pot parametritzar la detecció de duplicats via `ENABLE_DUPLICATE_DETECTION`. Limita mida d'arxiu (`MAX_FILE_SIZE`) i exposa `/health`.

**ai-ticket-processor (8004):** Microservei d'IA per extreure metadades del tiquet (OCR + Gemini). Opera via HTTP intern, rep GEMINI\_API\_KEY, i suporta LOG\_LEVEL/DEBUG/ENVIRONMENT per diagnosi. Llegeix imatges del directori muntat en read-only i retorna estructures utilitzables pel ticket-service.

**gamification-service (8005):** Calcula i persisteix punts, nivells, insígnies i redempcions. Proveeix endpoints per consultar i actualitzar l'estat de gamificació derivat d'esdeveniments de compra. Inclou script d'inicialització de recompenses (init\_rewards.py) i healthcheck.

**crm-service (8006):** Segmentació de clients, campanyes i inferència de preferències; integra amb Ollama (11434) per LLM local i amb notification-sender per a l'enviament. Manté pròpia API per crear segments/campanyes i consultar resultats; disposa de fallbacks si l'IA o les notificacions no són disponibles.

**notification-sender (8007):** Enviament de notificacions (WebPush; adapters també per iOS/Android). Requereix claus VAPID i guarda traça d'enviaments a BD; rep sol·licituds del CRM i altres serveis. Exposa healthcheck per a supervisió.

**postgres (5432):** PostgreSQL amb volums per a dades persistents, logs i scripts d'inicialització. Serveix de backend de persistència compartit pels microserveis, amb sanejament d'arrencada via healthcheck (pg\_isready).

**pgadmin (8080):** Consola d'administració per PostgreSQL (credencials definides al compose). Útil per inspecció d'esquemes, execució de consultes i diagnosi en entorns de desenvolupament.

**ollama (11434):** Servidor LLM local utilitzat pel CRM per a inferències i assistents internes. Emmagatzema models i cache en volum dedicat; funciona de manera opcional si no hi ha accés a serveis externs d'IA.

Tots els contenidors comparteixen la xarxa tfg\_network i es resolen pel seu nom de servei (DNS intern). Utilitzem una xarxa pròpia perquè aporta aïllament lògic respecte de l'host i d'altres stacks, resolució de noms estable sense haver de fixar IPs, evita conflictes de ports (només exposem públicament els necessaris) i permet comunicació privada entre serveis interns sense exposar-los a l'exterior. A més, facilita el desplegament multi-entorn mantenint els mateixos al·liances i configuració, i, si es repliquen serveis, el DNS intern pot fer round-robin entre instàncies. Les probes /health, els paràmetres LOG\_LEVEL/DEBUG i la configuració via .env faciliten l'observabilitat i el canvi d'entorn sense modificar codi; CORS/TLS i el ruteig públic es gestionen al Nginx extern en producció.

<input type="checkbox"/>	▼	<input type="radio"/>	tfg	-	-
<input type="checkbox"/>		<input type="radio"/>	tfg_ollama	4386f8ed0059	<a href="#">ollama/ollama:latest</a>
<input type="checkbox"/>		<input type="radio"/>	tfg_ai_ticket_processor	1a8a4f0241a3	<a href="#">tfg-ai-ticket-processor</a>
<input type="checkbox"/>		<input type="radio"/>	tfg_crm_service	f1de6f60c4f8	<a href="#">tfg-crm-service</a>
<input type="checkbox"/>		<input type="radio"/>	tfg_notification_sender	f62f701ce9a4	<a href="#">tfg-notification-sender</a>
<input type="checkbox"/>		<input type="radio"/>	tfg_pgadmin	bc449368a9eb	<a href="#">dpage/pgadmin4:latest</a>
<input type="checkbox"/>		<input type="radio"/>	tfg_postgres	d3c6e30bbf9b	<a href="#">tfg-postgres</a>
<input type="checkbox"/>		<input type="radio"/>	tfg_auth_service	7e1d25ef6e15	<a href="#">tfg-auth-service</a>
<input type="checkbox"/>		<input type="radio"/>	tfg_backend	f4dd4b6b2baa	<a href="#">tfg-backend</a>
<input type="checkbox"/>		<input type="radio"/>	tfg_ticket_service	bc7fbafcc661	<a href="#">tfg-ticket-service</a>
<input type="checkbox"/>		<input type="radio"/>	tfg_frontend	60c18ec7e349	<a href="#">tfg-frontend</a>
<input type="checkbox"/>		<input type="radio"/>	tfg_gamification_service	d27e37c16197	<a href="#">tfg-gamification-service</a>

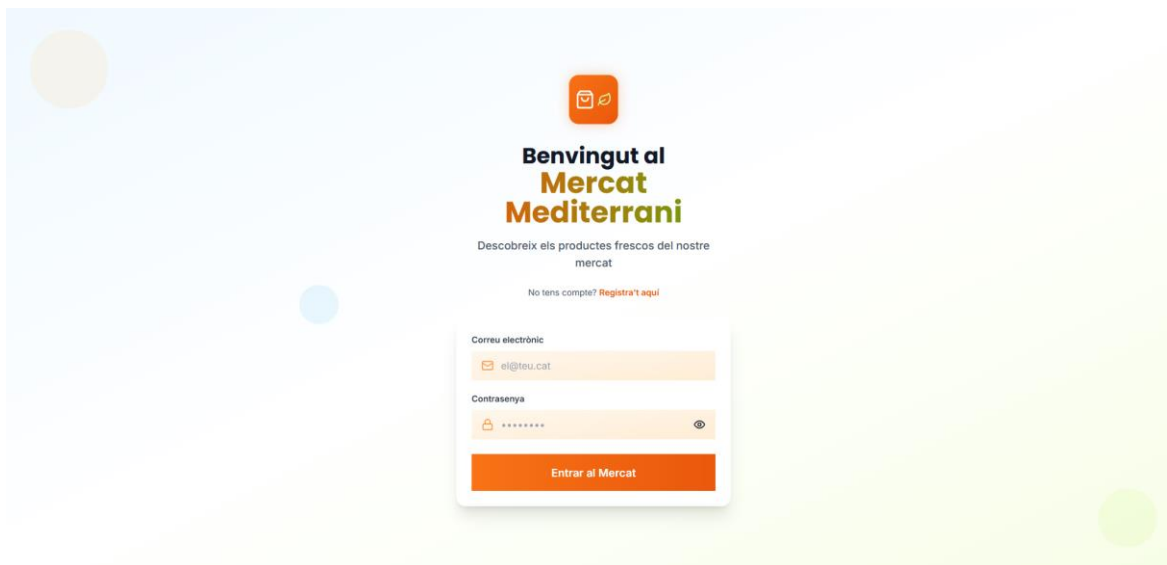
**Figura 19.** Microserveis i dependències del projecte despleats com a contenidors Docker

### 6.3 Interfície gràfica

L'interfície d'usuari es construeix com una SPA en React + TypeScript, estilada amb Tailwind CSS i servida per Nginx en el contenidor de frontend. El router client (React Router) organitza la navegació per rols i àrees (usuari final, venedor i administrador), amb rutes privades que exigeixen token JWT i validació de permisos. La capa d'accés a dades es fa amb Axios cap a les APIs dels microserveis; l'entorn `REACT_APP_ENVIRONMENT` selecciona els endpoints (localhost en desenvolupament, domini en producció). El disseny prioritza: coherència visual, feedback immediat (loading/empty/error), accessibilitat bàsica (semàntica i focus), i adaptació responsive per a mòbil (escaneig de tiquets, QR).

## 6.4 Pantalles d'usuari

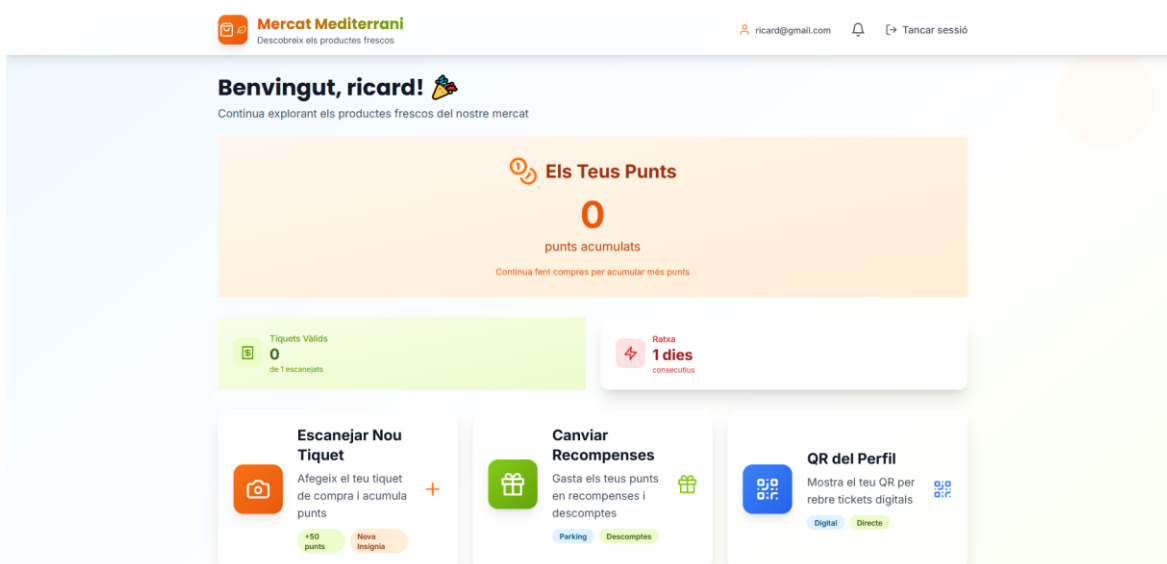
### 6.4.1 Pantalla de login



**Figura 20.** Pantalla de login

La pantalla de login usa un layout centrat amb jerarquia tipogràfica clara: pictograma de marca en pastilla amb gradient taronja, titular “Benvingut al Mercat Mediterrani” (modificable segons el tenant) amb gradient taronja→verd i fons suau en degradat amb cercles difuminats.

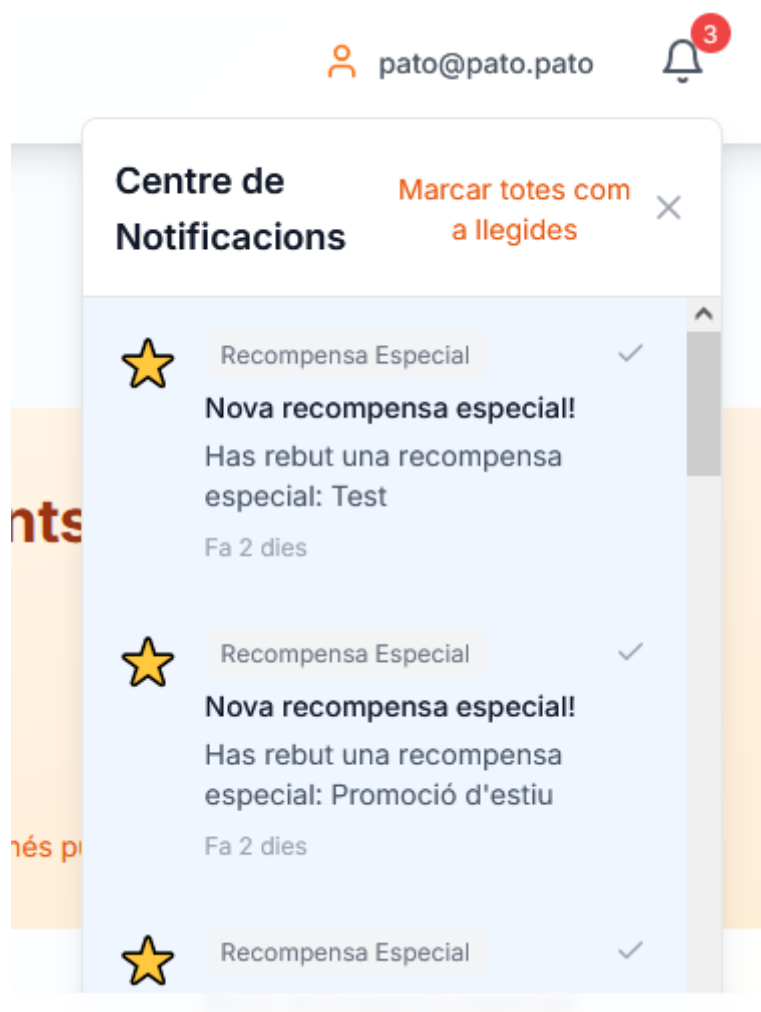
### 6.4.2 Dashboard d'usuari



**Figura 21.** Pantalla de dashboard (part superior)

El dashboard s'ha dissenyat com a “companion” per a la visita física al mercat, no com una experiència 100% virtual. Per això es prioritza una sola línia d'accions principals i el

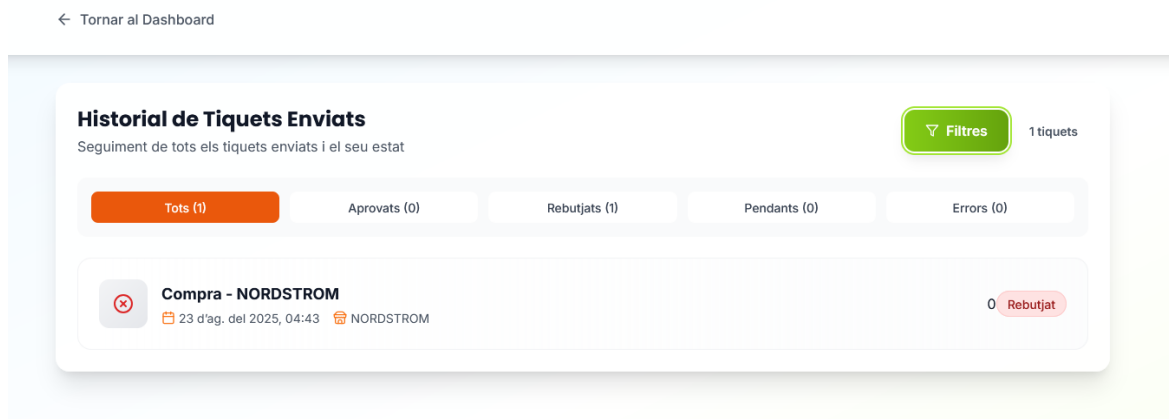
mínim de botons en horitzontal: escanejar tiquet, canviar recompenses i QR del perfil. Aquesta contenció redueix càrrega cognitiva i facilita l'ús tàtil (CTAs amples, espaiat generós), pensant en tots els públics. La informació clau (punts, tiquets vàlids, ratxa) es presenta com a KPIs de lectura immediata.



**Figura 22.** Menú desplegable de notificacions

El menú desplegable de notificacions (campana a la topbar) és un panell ancorat i desplaçable on arriben totes les notificacions de l'usuari, incloses les “Recompenses Especials”. Mostra targetes cronològiques amb títol, descripció breu, xip de categoria i marca temporal, amb comptador a la campana i acció “Marcar totes com a llegides”. Permet marcar lectures individuals, tancar el panell ràpidament i manté l'històric sense abandonar el dashboard.

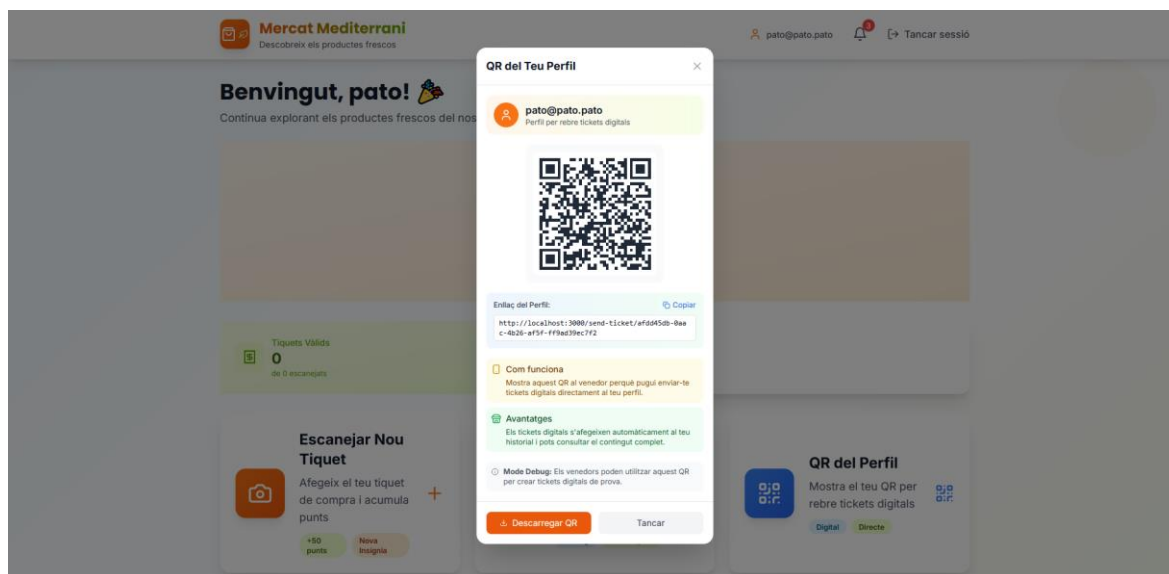
### 6.4.3 Historial de tiquets



**Figura 23.** Pantalla del historial de tiquets

L’“Historial de tiquets” és una vista de seguiment pensada per diagnosticar ràpidament tiquets rebutjats o aprovats. Es basa en un selector de pestanyes (Tots/Aprovats/Rebutjats/Pendents/Errors) amb comptadors i un botó de filtres destacat, mentre la llista mostra cada tiquet en card condensada amb comerç, data i xip d’estat (p. ex., Rebutjat). El disseny prioritza lectura a cop d’ull, canvi de filtre sense fricció i zero-states clars per facilitar la revisió i la resolució d’incidències.

### 6.4.4 QR del perfil



**Figura 24.** Menú “QR del teu perfil”

El diàleg “QR del Teu Perfil” és un modal centrat que mostra el codi QR i l’enllaç únic del perfil per rebre tiquets digitals: el venedor l’escaneja i el tiquet s’assigna automàticament a l’usuari. Inclou accions ràpides de copiar enllaç i descarregar el QR (per imprimir o portar-lo al mòbil), i seccions d’ajuda breus (“Com funciona”, “Avantatges”) amb un mode “Debug” per proves. El disseny manté el context del dashboard.

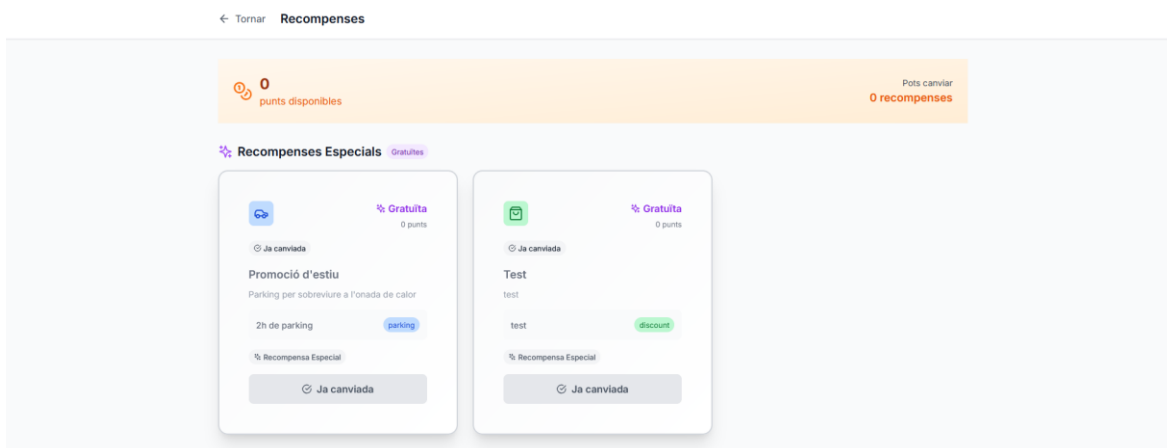
### 6.4.5 Pantalla d'escaneig de tiquet



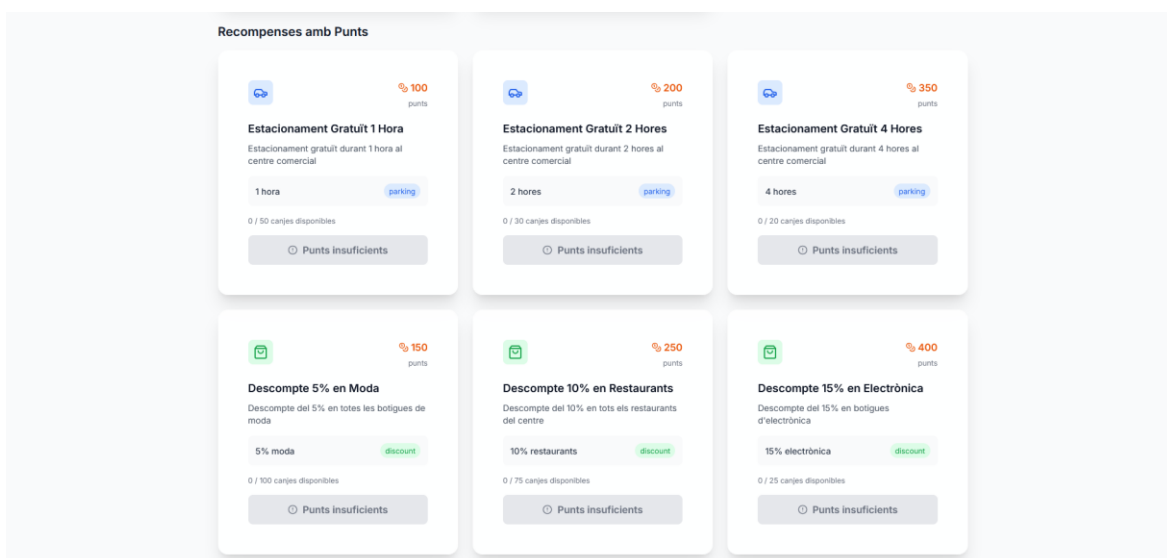
**Figura 25.** Pantalla d'escaneig de tiquet

Pantalla mòbil “Escanejar Tiquet de Compra” dissenyada mobile-first com a companion: app bar amb “Tornar” i títol, targeta d'instruccions i previsualització de càmera amb marc discontinu blau que delimita la zona d'escaneig. A la zona segura inferior hi ha tres accions grans i accessibles: Cancel·lar (secundari), Pujar Foto (Debug) per pujar una imatge des d'arxiu (opció en estudi i marcada com a debug per no desincentivar l'escaneig en el moment de la compra) i el CTA principal “Capturar Tiquet” en gradient taronja.

### 6.4.6 Pantalla de recompenses



**Figura 26.** Pantalla de recompenses (part superior)

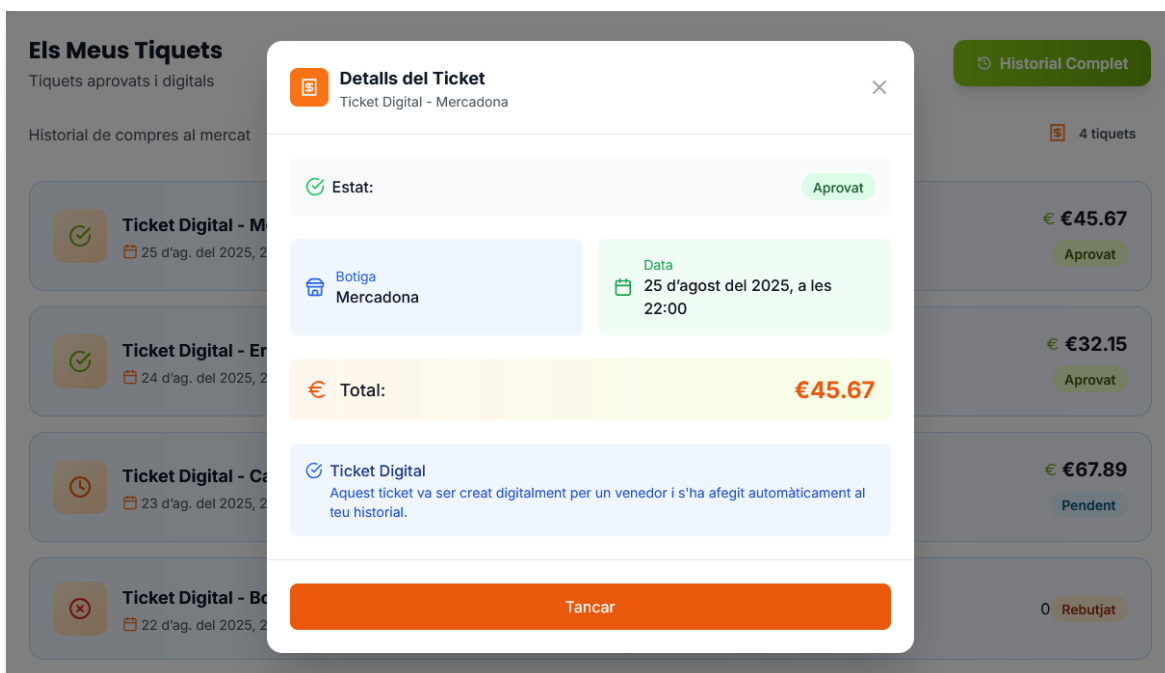


**Figura 27.** Pantalla de recompenses (part inferior)

**Recompenses amb Punts:** catàleg en grid responsive de targetes (3 columnes) pensat per comparar d'un cop d'ull.

**Recompenses Especiales:** vista amb banner de saldo i recompenses gratuïtes diferenciades. Les targetes indiquen l'estat ("Ja canviada"). Conserva el mateix disseny.

### 6.4.7 Detalls del tiquet digital



**Figura 28.** Detalls del tiquet digital

Pantalla de “Detalls del Tiquet” (modal). Mostra, per als tiquets digitals, tota la informació: estat, botiga, data/hora i total, amb una nota que indica que el tiquet ha estat creat digitalment per un venedor. A l’Historial, només els tiquets digitals permeten obrir aquest detall complet; els tiquets escanejats no mostren el detall de línies, perquè els productes detectats s'utilitzen per alimentar el CRM i no per a la consulta de l’usuari.

## 6.5 Pantalles del venedor

### 6.5.1 Dashboard del venedor



**Figura 29.** Panell del venedor

Panell de Venedor amb escàner QR a pantalla ampla pensat per a dos casos: 1) llegir el QR de perfil del client per enviar-li un tiquet digital i 2) validar/canviar QRs de recompenses. Dues accions visibles: Aturar càmera i “Pujar Imatge (Debug)” per proves.

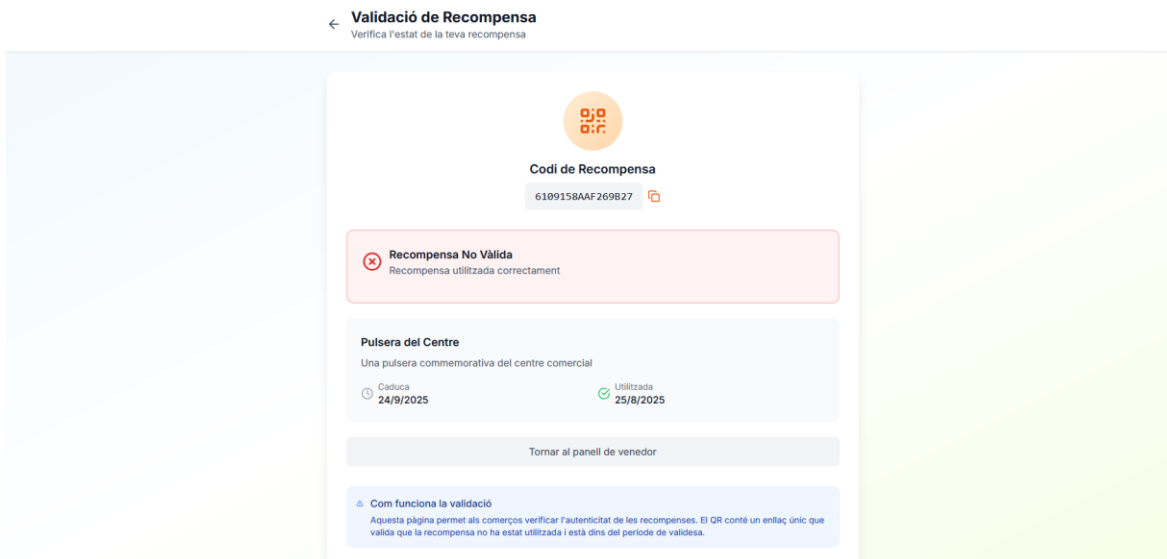
### 6.5.2 Tiquet digital

Figura 30. Tiquet digital

Pantalla “Enviar Tiquet Digital” per a venedors: a l’esquerra es mostra el client destinatari (derivat del QR) i l’avís de Mode Debug; a la dreta, el formulari per omplir botiga, data i línies de producte (nom, quantitat, preu), amb càlcul automàtic del total. CTA final “Enviar Tiquet Digital” per registrar-lo a l’historial del client.

### 6.5.3 Validació de recompenses

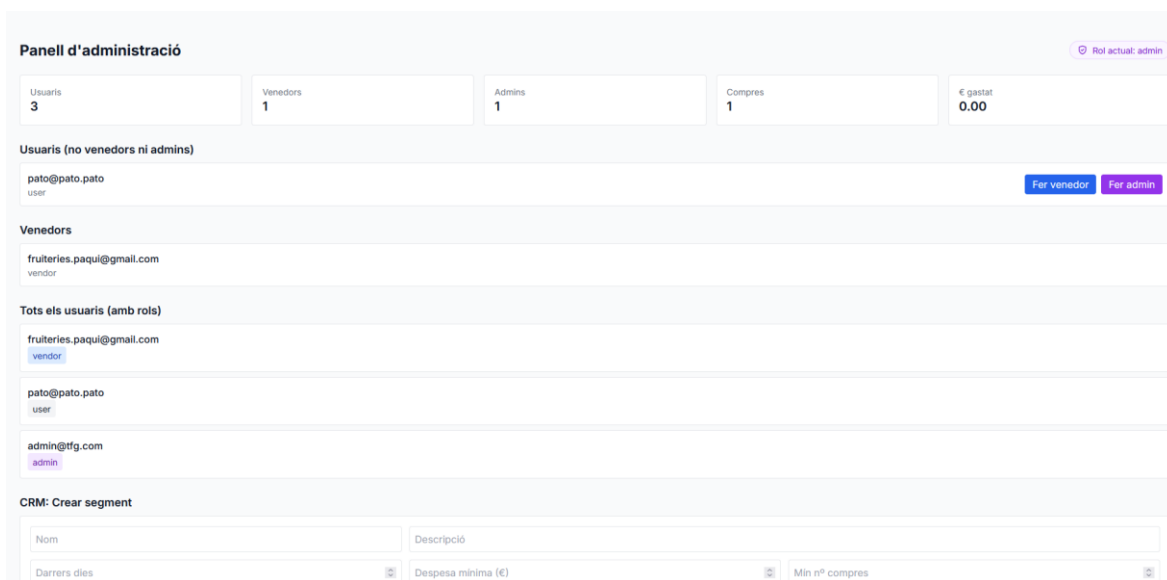
Figura 31. Pantalla de validació de recompensa



**Figura 32.** Pantalla de recompensa ja validada

Pantalla de Validació de Recompensa. Cada QR disposa d'un identificador únic (codi sota el QR). Quan el venedor l'escaneja des del panell i prem "Utilitzar Recompensa", aquesta queda registrada com a canviada i es torna inutilitzable. La vista indica l'estat en verd (vàlida i disponible) o en vermell (no vàlida/ja utilitzada), i mostra descripció, data de caducitat i, si escau, la data d'ús, amb accés per tornar al panell del venedor.

### 6.6 Pantalla d'administrador



**Figura 33.** Pantalla de administració (usuaris)

The screenshot displays the CRM administration interface with the following sections:

- CRM: Crear segment**: Includes input fields for 'Nom', 'Descripció', 'Darrers dies', 'Despesa mínima (€)', and 'Min nº compres'. A text area for 'Prompt de preferències (ex: clients que prefereixen productes ecològics i locals)' and a 'Crear segment' button are also present.
- CRM: Segments**: A header section.
- CRM: Crear campanya**: Includes input fields for 'Nom' and 'Descripció', a text area for 'Missatge (opcional; si es deixa buit es generarà amb IA)', a 'Selecciona segments' dropdown, and a 'Crear campanya' button.
- CRM: Campanyes**: A header section.
- CRM: Notificacions (en cua)**: A header section.
- Recompenses Especials: Crear recompensa**: Includes input fields for 'Nom', 'Descripció', a 'Descompte' dropdown, 'Valor (ex: 10% descompte)', and 'Màxim canjes per usuari'. A date picker and a 'Crear recompensa especial' button are also included.

**Figura 34.** Pantalla de administració (CRM)

The screenshot displays the CRM administration interface for special rewards with the following sections:

- CRM: Campanyes**: A header section.
- CRM: Notificacions (en cua)**: A header section.
- Recompenses Especials: Crear recompensa**: Includes input fields for 'Nom', 'Descripció', a 'Descompte' dropdown, 'Valor (ex: 10% descompte)', and 'Màxim canjes per usuari'. A checkbox for 'Per a tots els usuaris' and a date picker are also present, along with a 'Crear recompensa especial' button.
- Recompenses Especials: Llista**: A list of special rewards with details and actions:
  - Promoció d'estiu**: Parking per sobreviure a l'onada de calor. Especifica • parking • 2h de parking. [Eliminar]
  - Test**: test. Global • discount • test. [Eliminar]
- Recompenses Especials: Distribuir**: Includes a 'Selecciona una recompensa' dropdown, a 'Per a tots els usuaris' dropdown, a checked checkbox for 'Enviar notificacions', and a 'Distribuir recompensa' button.

**Figura 35.** Pantalla de administració (Recompenses especials)

El panell d'administració concentra la gestió operativa del sistema en un entorn de taula clar i dens. A la capçalera, unes targetes KPI mostren usuaris, venedors, admins, compres i € gastat, i des de la llista d'usuaris es poden ajustar rols amb accions ràpides (“Fer venedor / Fer admin”).

L'àrea de CRM permet crear segments combinant filtres (dies recents, despesa mínima, nombre de compres) i un prompt d'IA per definir preferències; aquests segments s'utilitzen immediatament per orquestrar campanyes i generar notificacions en cua.

El mòdul de Recompenses Especials facilita donar d'alta beneficis (descompte, parking o gratuïta), establir-ne el valor, la caducitat i el màxim de canvis per usuari, i posteriorment distribuir-los a un usuari concret, a un segment o a tota la base, amb enviament de notificacions integrat.

El disseny és deliberadament més simple i amb major densitat d'informació: es pressuposa l'ús en ordinador per part d'un perfil administrador, de manera que es prioritza la lectura ràpida, els formularis alineats i les taules sense distraccions.

## 6.7 Persistència a la BD

La plataforma persisteix la informació en PostgreSQL amb una organització per dominis clars: identitat, tiquets i compres, gamificació, recompenses, CRM i notificacions. El disseny manté un model auditable, amb dades estructurades per a l'operativa i camps flexibles en JSONB per guardar resultats d'extracció i metadades canviants. Es treballa sempre amb identificadors únics i segells temporals per poder reconstruir fàcilment l'activitat.

El model d'usuari és únic i senzill: un registre per persona amb el seu rol (usuari, venedor o administrador) i un petit perfil.

El tiquet és l'entitat central: desa qui el rep, qui el genera (venedor), el tipus (digital o escanejat), l'estat (aprovat, pendent, rebutjat), el comerç i els totals. Les línies (producte, quantitat, preu) viuen a part per alimentar el CRM i l'anàlisi, i hi ha un històric materialitzat per consultes eficients. En els tiquets escanejats es desa la ruta de la imatge i les metadades d'OCR en JSONB; per disseny, l'usuari veu el total però no les línies, que s'utilitzen pel CRM comercial. Els tiquets digitals, creats des del panell del venedor, són consultables amb tot detall. L'usuari disposa d'un QR de perfil per rebre tiquets digitals directament.

La gamificació es basa en un "llibre major" de punts i un resum de saldo per a lectura ràpida, que es manté de forma consistent després de cada moviment. El catàleg de recompenses i les redempcions treballen amb codis QR únics; quan el venedor valida una recompensa, el codi queda inutilitzat de manera idempotent i, dins la mateixa transacció, es descompten els punts i s'actualitzen els estats. També es contemplen recompenses especials i les seves verificacions.

La consistència derivada i el rendiment de lectura es resolen amb vistes i lògica de manteniment automàtic; el subsistema de notificacions utilitza una cua d'enviament perquè el lliurament sigui fiable sense bloquejar la interacció. En conjunt, el model a PostgreSQL s'alinea amb el disseny funcional: identitat unificada, tiquet com a nucli, gamificació auditada, recompenses amb QR únic i segments de CRM materialitzats per actuar amb latència baixa.

## 7 Implementació

Aquest capítol descriu com s'ha materialitzat el disseny en un sistema executable, segur i mantenible. Ens centrem en detalls tècnics: tecnologia emprada, contractes i fluxos, models de dades, algorismes específics (OCR, gamificació, validació de recompenses), seguretat, observabilitat i configuració. L'objectiu és deixar clar el “com” de la implementació: què fa cada servei, amb quines dependències i sota quines garanties d'integritat i rendiment. Partim d'una arquitectura per serveis i d'un desplegament contenitzat.

### 7.1 Tecnologies i control de versions

**Python + FastAPI:** Serveis backend lleugers amb APIs REST, validació d'esquemes tipada i documentació OpenAPI automàtica. Permet integrar fàcilment lògica de negoci, autenticació i accés a dades.

**React + Nginx:** Frontend SPA modular i responsiu servit de forma eficient amb Nginx (caché, compressió i fallback per rutes). Comunicació directa amb les APIs internes.

**PostgreSQL:** Motor relacional ACID amb suport per UUID i JSONB per a metadades flexibles (OCR). Índexs i constraints per garantir integritat i bon rendiment.

**Docker i Docker Compose:** Contenització i orquestració del stack complet (serveis, BD, IA, web) amb xarxa interna, volums i healthchecks per a una operació estable.

**IA (Gemini + Ollama):** Gemini per OCR/LLM d'extracció de tiquets escanejats; Ollama (LLM local) per assistència en CRM (segments/missatges) sense dependència externa constant.

**Web Push (VAPID):** Enviament de notificacions al navegador amb subscripcions per usuari i traçabilitat d'estats (queued/sent/failed).

**Autenticació i seguretat (JWT + hashing):** Sessions amb JWT signats i expiració; contrasenyes amb hashing robust (p. ex., bcrypt/argon2) i política de mínim privilegi per rutes.

**Control de versions (Git/GitHub):** Flux simple amb branques main/dev i feature.

### 7.2 Organització del codi

El codi s'organitza per domini i servei, aplicant una separació clara per capes. A cada microservei del backend es distingeixen: API (rutes i dependències, p. ex. main.py), esquemes/DTO (schemas.py amb Pydantic per a validació i contractes), lògica d'aplicació (serveis/ús-casos encapsulats en mòduls dedicats) i accés a dades (models.py per ORM/taules i database.py per connexió/queries). La configuració i secrets es centralitzen (config.py, env.example) i els clients externs es mantenen com a mòduls independents (p. ex., crm-service/ai\_client.py, auth\_client.py) per evitar acoblaments. Aquest patró es repeteix a auth-service (auth.py, models.py, schemas.py), ticket-service (gestió d'uploads i hash), ai-ticket-processor (ai\_system.py, auto\_processor.py per pipeline d'OCR) i resta de serveis, garantint que l'API només orquestra i la lògica de domini viu a la capa de servei.

```

from sqlalchemy import Column, String, DateTime, Boolean, Integer, ForeignKey, Text, Numeric, Index
from sqlalchemy.dialects.postgresql import UUID, JSONB
from sqlalchemy.orm import relationship
from sqlalchemy.sql import func
import uuid

from database import Base

class User(Base):
    __tablename__ = "users"

    id = Column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    email = Column(String(255), nullable=True)
    preferences = Column(JSONB, default={})
    registration_date = Column(DateTime(timezone=True), server_default=func.now())

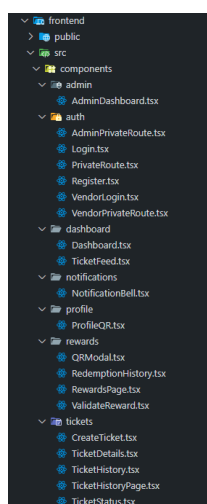
class PurchaseHistory(Base):
    __tablename__ = "purchase_history"

    id = Column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    user_id = Column(UUID(as_uuid=True), nullable=False)
    ticket_id = Column(UUID(as_uuid=True), nullable=False)
    purchase_date = Column(DateTime(timezone=True), nullable=False)
    store_name = Column(String(255), nullable=False)
    total_amount = Column(Numeric(10, 2), nullable=False)
    products = Column(JSONB, default=[])
    num_products = Column(Integer, default=0)
    ticket_type = Column(String(100), nullable=True)
    is_market_store = Column(Boolean, default=False)
    created_at = Column(DateTime(timezone=True), server_default=func.now())
    updated_at = Column(DateTime(timezone=True), server_default=func.now(), onupdate=func.now())

```

**Figura 36.** Exemple d'un fitxer models.py de cada microservei

Al frontend s'adopta una organització per funcionalitats, amb components i pantalles en src/components/..., contextos d'autenticació i rol en src/contexts/ (AuthContext, VendorAuthContext) i un client HTTP únic en src/config/api.ts. Les rutes privades (PrivateRoute, AdminPrivateRoute, VendorPrivateRoute) encapsulen l'autorització i els serveis de domini (p. ex., crmService.ts, gamificationService.ts) declaren endpoints i tipatges, mantenint els components UI lliures de lògica d'API.



**Figura 37.** Organització dels fitxers del frontend

### 7.3 Implementació del backend per servei

En aquest apartat descrivim com s'han materialitzat funcionalment els serveis backend, posant el focus en els casos d'ús que realment mouen el producte (autenticació i rols, creació i processament de tiquets, assignació de punts i redempció, segmentació i campanyes, i enviament de notificacions). Per a cada servei, destaquem les rutes clau, els límits transaccionals que garanteixen integritat (què passa “tot o res”) i els punts d'idempotència que eviten duplicats o estats inconsistents sota reintents.

Es descriu com es valida i es firma l'accés, com es normalitza un tiquet escanejat i es reconcilia el total, com es registra l'experiència de forma idempotent vinculada a un `ticket_id`, com es tanca una redempció amb codi únic en una sola transacció, o com es materialitzen destinataris d'una campanya i es lliuren notificacions multi-canal.

#### 7.3.1 *Auth-service*

Gestiona registre i inici de sessió, emissió/validació de JWT i control de rols. El flux de login/registre valida credencials (hash robust), emet access token amb expiració i registra/renova sessions. Les rutes protegides usen middleware de verificació del token i del rol. Inclou endpoints de manteniment de compte (verificació de correu/restabliment de contrasenya) i un servei per recuperar l'historial de compres de l'usuari autenticat (agregat per mostrar al perfil i alimentar gamificació). Idempotència en verificacions i revocació de sessions per evitar estats inconsistents.

Endpoints principals:

- POST `/auth/register` → crea usuari amb hash de contrasenya (bcrypt/argon2); valida unicitat d'email.
- POST `/auth/login` → valida credencials, emet JWT d'accés (exp), crea/renova sessió a BD.
- POST `/auth/logout` → revoca sessió/refresh token.
- GET `/auth/me` → retorna perfil des del JWT.
- GET `/auth/me/purchases` → historial de compres (agregat) per a l'usuari autenticat.
- POST `/auth/password/reset/request` i `/confirm` → tokens únics amb expiració.

Models i esquemes:

- `models.py`: User, UserSession, PasswordResetToken, EmailVerification.
- `schemas.py`: RegisterIn, LoginIn, TokenOut, UserOut, PurchaseItemOut.

Patrons tècnics:

- Validació i serialització amb Pydantic; signatura JWT amb `SECRET_KEY` i scope/rol.
- Idempotència: revocacions i verificacions amb `UNIQUE` a tokens; `on_conflict DO NOTHING` per evitar duplicats.

```

@app.post("/auth/login")
async def login_user(user_credentials: UserLogin, db: Session = Depends(get_db)):
    """Iniciar sesión de usuario"""
    user = db.query(User).filter(User.email == user_credentials.email).first()

    if not user or not pwd_context.verify(user_credentials.password, user.password_hash):
        raise HTTPException(
            status_code=status.HTTP_401_UNAUTHORIZED,
            detail="Credenciales incorrectas"
        )

    if not user.is_active:
        raise HTTPException(
            status_code=status.HTTP_401_UNAUTHORIZED,
            detail="Usuario inactivo"
        )

    # Crear token de acceso
    access_token = create_access_token(data={"sub": str(user.id)})

    expires_in = settings.ACCESS_TOKEN_EXPIRE_MINUTES * 60
    print(f"🔑 Token configurado para expirar en {settings.ACCESS_TOKEN_EXPIRE_MINUTES} minutos ({expires_in} segundos)")

    return {
        "access_token": access_token,
        "token_type": "bearer",
        "user": {
            "id": user.id,
            "email": user.email,
            "role": user.role,
            "preferences": user.preferences,
            "is_active": user.is_active,
            "registration_date": user.registration_date,
        }
    }

```

**Figura 38.** Controller del auth-service

### 7.3.2 Main Backend (facade)

Actua com a passarel·la “orientada a casos d’ús”: agrega dades de diversos serveis i aplica filtres/paginació comuns. Exemples: dashboard d’usuari (resum de tiquets + punts + notifikacions recents) o endpoints que combinen estat de recompenses i perfil. Centralitza CORS, codis d’error coherents i tradueix respostes heterogènies a un model únic de sortida. Manté la lògica de cola “glue” fora del frontend, simplificant el client.

Endpoints:

- GET /api/dashboard → { tickets\_summary, gamification\_summary, notifications }
- GET /api/rewards/catalog → catàleg agregat amb estat de l’usuari.
- GET /api/tickets/search → passarel·la amb filtres/paginació unificats.

Models i esquemes:

- Schemas d’“output compostat”: DashboardOut, RewardsCatalogOut, PagedListOut.
- Adaptadors de resposta per servei: mappers de Ticket/Gamification/Notifikacions a un model comú.

Patrons tècnics:

- Clients HTTP interns amb forward del header Authorization.
- Normalització d’errors a { code, detail } i llistats a { items, total, page, page\_size }.
- Timeouts i reintents acotats; si un servei falla, “graceful degradation” amb seccions buides.

### 7.3.3 Ticket-service

Dona d'alta tiquets digitals (emessos pel venedor a partir del QR de l'usuari) i processa pujades escanejades. En la pujada valida mida/MIME, guarda només el path de la imatge, calcula hash (detecció de duplicats) i crea un registre en estat "pendent". L'aprovació (manual o via AI) és transaccional: es fixa l'estat definitiu, es persisteix el resultat estructurat i es notifica gamificació. Les consultes llisten amb filtres/paginació; als escanejats, l'usuari veu totals/resum (privadesa).

Endpoints principals:

- POST /tickets/digital → crea tiquet digital a partir del QR d'usuari i línies declarades.
- POST /tickets/scanned (multipart) → valida imatge, desa, calcula image\_hash i registra "pendent".
- GET /tickets, GET /tickets/{id} → llistats i detall; en escanejats, el client veu totals/resum (no línies).
- POST /tickets/{id}/approve | /reject → fixa estat, escriu metadades d'OCR i notifica.

Models i esquemes:

- models.py: Ticket, TicketImage (image\_path, image\_hash, expires\_at), TicketLine (per digitals), Audit.
- schemas.py: TicketCreateDigitalIn, TicketOut, TicketApproveIn (json OCR), PageQuery.

Patrons tècnics:

- Duplicats: UNIQUE(image\_hash) i resposta idempotent (retorna el ticket existent).
- Aprovació transaccional: estat + JSONB OCR + encolat/avís a gamificació en un sol commit.
- Privadesa: per escanejats, el GET del client final limita camps (total, comerç, data).

Calculem un hash (SHA-256) de cada imatge de tiquet per garantir idempotència i evitar duplicats: si un mateix rebut es puja diverses vegades (accidentalment o amb intenció de frau), el sistema el reconeix i no crea nous registres ni torna a atorgar punts. Això ens dona coherència transaccional i una defensa simple però efectiva contra la doble comptabilització. A més, el hash permet consultes i comprovacions molt ràpides (índex únic a BD) sense haver de reprocessar el fitxer, i millora la privadesa perquè només guardem el camí i metadades, no el binari dins la base de dades. També facilita la traçabilitat i la neteja programada d'imatges, mantenint el sistema eficient i ordenat.

```

if time_window_start <= existing_datetime <= time_window_end:
    print(f" 🕒 Coincidencia de fecha encontrada: {existing_fecha} está en ventana de ±5 minutos")
    # Verificar si los productos coinciden
    existing_productos_normalized = [normalize_product(p) for p in existing_productos]
    existing_productos_sorted = sorted([str(p) for p in existing_productos_normalized])
    existing_productos_hash = hash(tuple(existing_productos_sorted))

    print(f" 📋 Productos del ticket existente (normalizados): {existing_productos_sorted[:3]}...") # Mostrar solo 3
    print(f" 📋 Comparando hashes: actual={productos_hash}, existente={existing_productos_hash}")

    # Comparación directa de productos (más robusta)
    productos_actual_set = set(str(p) for p in productos_normalized)
    productos_existente_set = set(str(p) for p in existing_productos_normalized)

    print(f" 📋 Comparación directa: {len(productos_actual_set)} vs {len(productos_existente_set)} productos únicos")

    if existing_productos_hash == productos_hash:
        print(f" 🕒 Duplicado encontrado (hash): fecha={existing_fecha}, productos={len(existing_productos)}")
        return True
    elif productos_actual_set == productos_existente_set:
        print(f" 🕒 Duplicado encontrado (comparación directa): fecha={existing_fecha}, productos={len(existing_productos)}")
        return True
    else:
        print(f" ❌ Productos no coinciden (hash ni comparación directa)")

```

Figura 39. Lògica de comparació de hashes

### 7.3.4 AI ticket-processor

Pipeline dedicat a OCR/LLM: preprocés d'imatge, invocació a Gemini, normalització (línies, total, comerç, data) i reconciliació de totals amb tolerància. Gestiona errors (imatge deficient, total inconsistent, format no reconegut) i aplica reintents amb backoff. La correlació per ticket\_id i image\_hash garanteix idempotència: el mateix tiquet no es procesa ni s'escriu dues vegades. El resultat actualitza l'estat del tiquet a "aprovat" o "rebutjat" i afegeix metadades en JSONB.

Endpoints/processos:

- POST /process/{ticket\_id} → dispara processament d'un pendent (o worker intern que escaneja pendents).
- Callback PUT /tickets/{id}/approve|reject al Ticket-service amb resultat estructurat.

Models i esquemes:

- Estructures d'AI: ExtractedTicket { merchant, date, lines[], total }, LineItem { name, qty, unit\_price, subtotal }.
- Config: ai\_system.py (prompt, parsing), auto\_processor.py (pipeline), main.py (servei/router).

Patrons tècnics:

- Reintents amb backoff per errors transitòris; classificació d'errors (qualitat d'imatge, inconsistència de totals).
- Idempotència: si el ticket ja té estat final o un hash processat, s'omet el reprocessat.
- Persistència de metadades OCR en JSONB i traça de temps/processos per auditoria.

```

def call_gemini_api(self, image_base64: str) -> str:
    """
    Llamar a la API de Gemini con la imagen
    """
    headers = {
        'Content-Type': 'application/json',
        'X-goog-api-key': self.api_key
    }

    # Prompt específico para procesar tickets
    prompt = """
    Analiza esta imagen de un ticket de compra y extrae la siguiente información en formato JSON:

    {
      "fecha": "fecha del ticket (formato DD/MM/YYYY o similar)",
      "hora": "hora del ticket (formato HH:MM)",
      "tienda": "nombre de la tienda o establecimiento",
      "total": "importe total del ticket (solo el número)",
      "tipo_ticket": "tipo de ticket (supermercado, restaurante, gasolinera, farmacia, otros)",
      "productos": [
        {
          "cantidad": "cantidad del producto",
          "nombre": "nombre del producto",
          "precio": "precio del producto (solo el número)"
        }
      ]
    }

    Reglas importantes:
    - Si no encuentras algún campo, ponlo como null
    - Para productos, extrae solo los que sean claramente productos (no totales, impuestos, etc.)
    - Los precios deben ser solo números (sin símbolos de moneda)
  """

```

**Figura 40.** Codi amb el prompt a Gemini

### 7.3.5 Gamification-service

Aquest servei manté el perfil de gamificació de cada usuari (punts, ratxa, compres), el registre d'experiència i les insignies, i implementa el cicle complet de recompenses: les "normals" (amb cost en punts) i les "ESPECIALS" (0 punts, dirigides per objectiu i amb distribució).

En aprovar-se un tiquet, registra un assentament idempotent per `ticket_id`; gestiona catàleg de recompenses, redempcions i validació amb codi únic.

Endpoints principals:

- POST /events/ticket-approved → registra experiència/punts de forma idempotent (key=ticket\_id).
- GET /me/summary → nivell, xp, streak, totals.
- GET /rewards → catàleg; POST /rewards/redeem → crea redempció; POST /rewards/validate → el venedor valida el codi.
- POST /rewards/redeem → crea redempció (reserva punts i, si és ESPECIAL, també reserva stock/condicions).
- POST /rewards/validate → validació al punt de venda amb QR/codi únic; per ESPECIALS, comprova condicions extra (temps, venedor, stock, límits).
- GET /rewards/redemptions → historial i estat de canvis (pending/used/expired).

Models i esquemes:

- models.py: ExperienceLog, UserGamificationSummary, Reward, RewardRedemption.

- schemas.py: ExperienceEventIn, GamificationSummaryOut, RewardOut, RedeemIn, ValidateIn.

Patrons tècnics:

- Idempotència de punts: ExperienceLog amb UNIQUE(ticket\_id). Si l'event ja existeix, no duplica i el sumari no es desquadra.

```
def _can_user_access_special_reward(self, user_id: uuid.UUID, reward: SpecialReward, user_segments: List[str] = None) -> bool:
    """Verifica si un usuario puede acceder a una recompensa especial"""
    user_id_str = str(user_id)

    # Si es global, todos pueden acceder
    if reward.is_global:
        return True

    # Verificar usuarios especificos
    if user_id_str in reward.target_users:
        return True

    # Verificar segmentos
    if user_segments and any(segment in reward.target_segments for segment in user_segments):
        return True

    # Verificar si el usuario ya tiene un registro de distribución para esta recompensa
    existing_redemption = self.db.query(SpecialRewardRedemption).filter(
        SpecialRewardRedemption.user_id == user_id,
        SpecialRewardRedemption.special_reward_id == reward.id
    ).first()

    if existing_redemption:
        return True

    return False
```

**Figura 41.** Codi amb la comprovació si el usuari pot o no accedir a una recompensa especial

### 7.3.6 CRM-service

Defineix segments com a filtres (JSONB) i permet crear campanyes que, en enviar-se, materialitzen destinataris i escriuen notificacions d'inbox. Inclou un client d'IA (Ollama) per suggerir segments o refinar missatges a partir d'un prompt, sense fer-lo obligatori al flux. Els endpoints exposen llistats amb filtres i auditen qui/quan ha creat o llançat una campanya. La selecció de destinataris és determinista i traçable (consulta generada a partir del filtre guardat).

Endpoints principals:

- POST /segments, GET /segments → alta i llistat de segments amb filtres JSONB.
- POST /campaigns → crea campanya; POST /campaigns/{id}/send → genera notificacions.
- POST /segments/suggest → AI helper (Ollama) per construir filtres des d'un prompt.
- GET /health → salut del servei

Models i esquemes:

- Segment { id, name, filter\_json(JSONB), created\_at, updated\_at }, Campaign { id, name, message, segments[], status, created\_by, created\_at }, CampaignLog { id, campaign\_id, sent\_count, requested\_by, created\_at }
- SegmentIn/SegmentOut, CampaignIn/CampaignOut, CampaignSendIn (opcions d'enviament, p. ex. dry\_run, batch\_size), NotificationIn (payload per a la inbox/Notification-Sender)

## Dependències:

- ai\_client.py → connexió a Ollama (<http://ollama:11434>) amb LLM\_MODEL configurable.
- auth\_client.py → validació d'identitat i rols contra Auth-Service.
- Notification-Sender → canalitza la difusió escrita prèviament a la inbox.

## Patrons tècnics:

- Traducció de filtres JSONB a consulta SQL determinista en el moment d'enviament (evita “drift” entre definició i execució).
- Audit trail (CampaignLog): qui/quan envia una campanya, resultat i volum de destinataris generats.
- Escalabilitat: processament per lots (batching) i buffer a la taula de notificacions; reintents sobre fallades transitòries del canal.
- Control d'accessos: només rols “gestor” o “admin” poden crear/enviar campanyes.
- Privadesa: segments basats en atributs agregats/derivats; no es manipulen dades sensibles al prompt.

## Interacció amb la IA (Ollama):

- Suggeriment de segments: a partir d'un prompt (p. ex. “joves que han comprat peix l'últim mes”), l'assistent proposa un esborrany de filter\_json (operadors, intervals, categories). El resultat es valida i “lint-eja” abans de persistir.
- Refinament de missatges: genera variacions de copy segons segment i to (“informal”, “proximity”), amb límits de longitud i placeholders segurs (mai dades personals).
- Configuració: OLLAMA\_HOST i LLM\_MODEL (per defecte qwen2.5:0.5b-instruct); timeouts i nombre de tokens acotats.
- Robustesa: si Ollama no respon, es retorna una resposta degradada (sense suggeriment) i es registra l'error; els endpoints de campanyes/segments continuen operatius.

```

@app.post("/segments", response_model=SegmentResponse)
async def create_segment(payload: SegmentCreate, db: Session = Depends(get_db), authorization: str | None = Header(default=None),
    await require_admin(authorization.replace("Bearer ", "") if authorization else ""))

# Mantener solo last_days, min_total_spent, min_num_purchases; ignorar tiendas
filters = payload.filters or {}
allowed_keys = {"last_days", "min_total_spent", "min_num_purchases"}
base_filters = {k: v for k, v in filters.items() if k in allowed_keys}

# Si llega prompt, invocar IA para derivar preferences_contains a partir del prompt
if payload.prompt:
    try:
        system = (
            "Eres un generador de preferencias estandarizadas. Convierte descripciones "
            "a preferencias usando SIEMPRE estas claves exactas: diet, store_preference, "
            "language, organic, budget_level, product_category. "
            "Valores posibles: diet (vegetariano, vegano, omnivoro), "
            "store_preference (mercadona, carrefour, lidl, dia), "
            "language (catalan, spanish, english), "
            "organic (true, false), "
            "budget_level (low, medium, high), "
            "product_category (fruits, vegetables, dairy, meat, bread, beverages, snacks, organic, gourmet, baby). "
            "Responde solo JSON."
        )
        gen = await generate_text(
            prompt=(
                f"Convierte a JSON estandarizado usando las claves exactas: {payload.prompt}. "
                "Usa solo las claves y valores especificados arriba."
            ),
            system=system,
            temperature=0.2,
            max_tokens=120,
        )
    )

```

**Figura 42.** Codi on recull i normalitza com a segment el prompt del usuari

### 7.3.7 Notification-Sender

Administra subscripcions per usuari i canal (webpush/android/ios) i en fa el lliurament. Web Push utilitza VAPID per signar i enviar; els adapters d'Android/iOS unifiquen payloads i estats. Cada enviament transita d'queued a sent/failed amb timestamps, i es depuren subscripcions caducades o invàlides. Els reintents són acotats i les respostes d'error es normalitzen per no bloquejar el flux de campanyes.

Endpoints principals:

- POST /subscriptions → registra subscripció per { canal, endpoint/pushData }.
- DELETE /subscriptions/{id} → baixa.
- POST /deliver → procés intern que consumeix notificaciones queued i envia (WebPush VAPID; adapters Android/iOS).

Lliurament i estats:

- queued → sent/failed amb timestamps; reintents acotats; baixa automàtica de subscripcions invàlides.

Capa d'adapters:

- iOS (ios\_adapter): mapeig a notificaciones APNs (prioritat, badge, sound, category), validació de token, normalització d'errors i codis de reintents.
- Android (android\_adapter): mapeig a FCM (priority, ttl, data vs notification), gestió de collapse\_key i canal, validació de registration tokens i tractament d'errors.
- WebPush (webpush\_adapter): signatura VAPID, TTL i topic opcional, gestió d'errors HTTP.

## 7.4 Implementació de l'aplicació mòbil

Per validar ràpidament el concepte i reduir la complexitat de desenvolupament, he construït una app Android en format APK que encapsula el nostre frontend dins d'un WebView apuntant al domini del projecte. Aquesta aposta respon a un objectiu explícit de simplificació: una sola base de codi per a la interfície, desplegaments continus i actualitzacions instantànies sense haver de publicar versions natives cada cop. Som conscients, però, que per a molta gent és més senzill instal·lar des d'una store que no pas descarregar i instal·lar un APK via navegador.

L'objectiu funcional principal de l'app és habilitar notificacions push natives de manera fiable. El contenidor nadiu registra un token FCM i el sincronitza amb el backend; a partir d'aquí, el nostre servei Notification-Sender utilitza l'adaptador d'Android per lliurar els missatges amb els paràmetres adequats (prioritat, TTL, canals, collapse keys) i, quan l'usuari interactua amb la notificació, el WebView s'obre directament a la secció rellevant mitjançant deep links (històric de tiquets, recompenses, o escaneig). D'aquesta manera esquivem les limitacions dels Web Push dins de WebView i garantim una cadena d'entrega i traçabilitat coherents amb la resta de canals del sistema.

```
public class MainActivity extends Activity {
    private WebView webView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        webView = findViewById(R.id.webview);

        // Configurar WebView
        WebSettings webSettings = webView.getSettings();
        webSettings.setJavaScriptEnabled(true);
        webSettings.setDomStorageEnabled(true);
        webSettings.setLoadWithOverviewMode(true);
        webSettings.setUseWideViewPort(true);
        webSettings.setBuiltInZoomControls(true);
        webSettings.setDisplayZoomControls(false);

        // Evitar que se abra en navegador externo
        webView.setWebViewClient(new WebViewClient());

        // Cargar la URL
        webView.loadUrl("https://mercatmediterrani.com");
    }
}
```

Figura 43. Codi de la app mòbil

## 7.5 Domini i hosting

### 7.5.1 Allotjament


























El sistema s’ha dissenyat en clau on-premises i cloud-agnòstica: tots els components s’executen en contenidors Docker orquestrats amb Docker Compose, cosa que permet portar la mateixa topologia a qualsevol entorn (CPD propi, màquines virtuals o qualsevol cloud IaaS/PaaS) sense canvis de codi. Aquest enfocament dona control sobre dades i xarxa (compliment i sobirania), aïllament lògic, i deixa oberta l’escalabilitat vertical i horitzontal (balancejadors, rèpliques, serveis gestionats de base de dades, etc.).

Com a implementació de referència, he desplegat en un droplet de DigitalOcean amb Ubuntu: he clonat el repositori i he aixecat tots els serveis amb Docker Compose, publicant els ports només a localhost i exposant l’accés extern a través d’un Nginx reverse proxy. En aquesta fase, l’entorn s’ha fixat a `environment=development` per facilitar proves i diagnòstic; per a un “production-like” només canvien paràmetres sensibles (secrets, TTLs, nivells de log, CORS/TLS i límits de recursos), mantenint la mateixa arquitectura. La xarxa Docker `tfg_network` proporciona DNS intern per nom de servei i comunicació privada; les probes `/health` i la configuració via `.env` faciliten l’observabilitat i el canvi d’entorn sense modificar codi. Migrar a un cloud gestionat implicaria, bàsicament, mapejar Postgres a un servei gestionat, posar Nginx darrere d’un load balancer i, si cal, incrementar rèpliques dels serveis.

El servidor Nginx resident al droplet és la peça d’entrada: rep totes les peticions al port 443 i les distribueix internament als serveis corresponents (p. ex. `/api` → 8000, `/auth` → 8001, `/tickets` → 8003, `/ai` → 8004, `/gamification` → 8005, `/crm` → 8006, `/notifications` → 8007), mantenint els contenidors protegits darrere de localhost. La configuració inclou compressió gzip, cache d’estàtics amb policies immutables, capçaleres de seguretat (X-Frame-Options, X-Content-Type-Options, HSTS) i un rate limiting específic per als endpoints d’API per esmorteir pics o abusos.

### 7.5.2 Domini I DNS

El domini `mercatmediterrani.com` s’ha adquirit a IONOS. Un cop el droplet estava operatiu i servint l’aplicació, vam actualitzar la zona DNS al panell d’IONOS per apuntar l’entrada A (i, si escau, AAAA) a la IP pública del droplet. Amb la propagació completada, Nginx ja podia atendre peticions per `mercatmediterrani.com` i `www.mercatmediterrani.com` gràcies a la directiva `server_name` de la seva configuració. Aquest pas tanca el cercle: el domini resol cap al nostre servidor, i el servidor sap com despatxar cada ruta cap al microservei intern pertinent.

<input type="checkbox"/>	TIPO	NOMBRE DE HOST	VALOR	SERVICIO ▲	ACCIONES
Filtrado por "Solo registros del dominio principal" <a href="#">x Restablecer filtro</a>					
<input type="checkbox"/>	CNAME	_domainconnect	_domainconnect.ionos.com	Domain Connect	 
<input type="checkbox"/>	MX	@	mx00.ionos.de	Mail	 
<input type="checkbox"/>	MX	@	mx01.ionos.de	Mail	 
<input type="checkbox"/>	TXT	@	"v=spf1 include:_spf-eu.ionos.com ~all"	Mail	  
<input type="checkbox"/>	CNAME	s1-ionos_domainkey	s1.dkim.ionos.com	Mail	  
<input type="checkbox"/>	CNAME	s2-ionos_domainkey	s2.dkim.ionos.com	Mail	  
<input type="checkbox"/>	CNAME	s42582890_domainkey	s42582890.dkim.ionos.com	Mail	  
<input type="checkbox"/>	CNAME	autodiscover	adsredir.ionos.info	Mail	  
<input type="checkbox"/>	A	@	159.65.125.178	-	 
<input type="checkbox"/>	A	www	159.65.125.178	-	 

**Figura 44.** Configuració de registres DNS

### 7.5.3 HTTPS

El trànsit xifrat és obligatori en aquest context, tant per seguretat bàsica com per compatibilitat amb funcionalitats del navegador i integracions. S'ha configurat Nginx amb redirecció automàtica d'HTTP a HTTPS, suportant TLS 1.2 i 1.3 i habilitant Strict-Transport-Security per reforçar l'ús exclusiu d'HTTPS en futures connexions. Els certificats s'han instal·lat sota /etc/ssl (la clau privada i el certificat complet de cadena), i el bloc del servidor HTTPS de Nginx en fa ús directament.

Per simplificar la posada en marxa i comprovar la configuració, s'ha utilitzat certbot: a més de validar la correcta exposició d'HTTPS i les redireccions, deixa el terreny preparat per a l'automatització de renovacions si en el futur es decideix migrar a certificats gestionats per Let's Encrypt. En l'estat actual, Nginx serveix el domini amb:

- redirecció 301 d'HTTP a HTTPS
- certificats instal·lats al sistema
- capçaleres de seguretat i gzip
- proxy\_pass per a cada microservei
- i limit\_req per a rutes d'API sensibles

## 8 Avaluació

### 8.1 Proves d'autenticació i usuaris

Cas	Descripció	Esperat	OK
Registre d'usuari vàlid	L'usuari introdueix email, contrasenya i nom vàlids al formulari de registre.	L'usuari es registra correctament i rep un token JWT.	OK
Registre amb email duplicat	L'usuari intenta registrar-se amb un email que ja existeix.	Missatge d'error "Email ja registrat".	OK
Registre amb dades invàlides	L'usuari introdueix dades incompletes o invàlides.	Missatge d'error de validació.	OK
Login vàlid	L'usuari introdueix credencials correctes a la pàgina d'inici de sessió.	Redirigit a la pantalla principal del dashboard.	OK
Login invàlid	L'usuari introdueix contrasenya o email incorrectes.	Missatge "Credencials invàlides".	OK
Accés sense autenticar	Intentar accedir a una ruta protegida sense una sessió activa.	Redirigit a vista indicant "No tens accés a aquest recurs".	OK
Verificació de token	Verificar la validesa d'un token JWT.	Informació de l'usuari retornada si el token és vàlid.	OK
Refresh de token	Renovar un token expirat.	Nou token vàlid generat.	OK
Logout	L'usuari tanca la sessió.	Token invalidat i redirigit a la pàgina de login.	OK

**Taula 20.** Proves d'autenticació i usuaris

## 8.2 Proves de gestió de tickets

Cas	Descripció	Esperat	OK
Crear ticket digital	El venedor crea un ticket digital per a un usuari.	Ticket creat amb ID únic i dades estructurades.	OK
Crear ticket amb dades incompletes	Intentar crear ticket sense dades obligatòries.	Missatge d'error de validació.	OK
Pujar imatge de ticket	L'usuari puja una foto de ticket per processar amb IA.	Imatge processada i dades extretes automàticament.	OK
Pujar imatge invàlida	Intentar pujar un fitxer que no és una imatge.	Missatge d'error "Format d'imatge no suportat".	OK
Pujar imatge massa gran	Intentar pujar una imatge que excedeix el límit de mida.	Missatge d'error "Imatge massa gran".	OK
Obtenir historial de tickets	L'usuari consulta el seu historial de tickets.	Llista de tots els tickets de l'usuari amb detalls.	OK
Obtenir historial buit	Usuari sense tickets consulta el seu historial.	Llista buida amb missatge informatiu.	OK
Obtenir detalls de ticket	Consultar informació detallada d'un ticket específic.	Detalls complets del ticket incloent productes.	OK
Obtenir ticket inexistent	Intentar accedir a un ticket que no existeix.	Missatge d'error "Ticket no trobat".	OK

**Taula 21.** Proves de gestió de tickets

## 8.3 Proves de IA per processar tickets

Cas	Descripció	Esperat	OK
Processar imatge de ticket vàlid	Enviar imatge de ticket vàlid al processador d'IA.	Dades estructurades extretes correctament.	OK
Processar imatge de	Enviar imatge borrosa o mal	Dades extretes amb possibles errors o	OK

baixa qualitat	il·luminada.	missatges d'advertència.	
Processar imatge sense text	Enviar imatge que no conté text de ticket.	Missatge d'error "No s'ha pogut detectar text de ticket".	OK
Processar imatge de ticket en diferent idioma	Enviar imatge de ticket en anglès o francès.	Dades extretes correctament independentment de l'idioma.	OK
Verificar disponibilitat de l'IA	Comprovar si el servei d'IA està funcionant.	Status "healthy" retornat.	OK
Processar múltiples imatges simultàniament	Enviar diverses imatges al mateix temps.	Totes les imatges processades correctament.	OK
Timeout en processament	Imatge que triga massa temps a processar.	Missatge d'error "Timeout en el processament".	OK

Taula 22. Proves de IA per processar tiquets

#### 8.4 Proves de CRM i Segmentació

Cas	Descripció	Esperat	OK
Crear segment amb prompt d'usuari	Administrador crea segment usant llenguatge natural.	Segment creat amb criteris tècnics inferits per IA.	OK
Crear segment amb criteris manuals	Administrador crea segment amb filtres específics.	Segment creat amb els filtres especificats.	OK
Crear segment amb prompt invàlid	Intentar crear segment amb prompt que no es pot interpretar.	Missatge d'error i suggeriments d'ajuda.	OK
Llistar segments	Consultar tots els segments disponibles.	Llista de segments amb informació detallada.	OK
Obtenir detalls de segment	Consultar informació específica d'un segment.	Detalls complets del segment incloent filtres.	OK
Previsualitzar	Veure quins usuaris pertanyen a un	Llista d'usuaris que compleixen els	OK

usuaris de segment	segment.	critèris.	
Crear campanya	Administrador crea campanya per a un o més segments.	Campanya creada amb missatge i segments associats.	OK
Crear campanya sense missatge	Crear campanya sense especificar missatge.	Missatge generat automàticament per IA.	OK
Enviar campanya	Enviar notificacions de campanya als usuaris.	Notificacions enviades i estadístiques generades.	OK
Inferir preferències d'usuari	Analitzar historial de compres per inferir preferències.	Preferències inferides i actualitzades automàticament.	OK
Inferir preferències per múltiples usuaris	Inferir preferències per tots els usuaris amb compres recents.	Preferències inferides per tots els usuaris elegibles.	OK

**Taula 23.** Proves de CRM i segmentació

### 8.5 Proves de notificacions

Cas	Descripció	Esperat	OK
Enviar notificació individual	Enviar notificació a un usuari específic.	Notificació enviada i registrada.	OK
Enviar notificació a múltiples usuaris	Enviar notificació a un grup d'usuaris.	Notificacions enviades a tots els usuaris.	OK
Enviar notificació push	Enviar notificació push a dispositius mòbils.	Notificació rebuda al dispositiu de l'usuari.	OK

**Taula 24.** Proves de notificacions

### 8.6 Proves al frontend

Cas	Descripció	Esperat	OK
Accedir a la pàgina principal	Obrir l'aplicació web al navegador.	Pàgina principal carregada correctament.	OK
Navegació entre	Navegar entre les	Navegació fluida	OK

pàgines	diferents seccions de l'aplicació.	sense errors.	
Responsive design	Accedir des de dispositius mòbils o tablets.	Interfície adaptada al tamany de pantalla.	OK
Càrrega de components	Carregar components dinàmics de l'aplicació.	Components carregats correctament.	OK
Gestió d'estats	Canviar entre diferents estats de l'aplicació.	Estats gestionats correctament.	OK
Gestió d'errors	Mostrar errors de manera amigable.	Errors mostrats amb missatges clars.	OK
Càrrega de dades	Carregar dades des dels serveis backend.	Dades carregades i mostrades correctament.	OK
Formularis de validació	Validar entrades d'usuari en formularis.	Validació en temps real amb missatges d'error.	OK
Pujada de fitxers	Interfície per pujar imatges de tickets.	Interfície de pujada funcional i intuïtiva.	OK
QR codes	Generar i mostrar QR codes per identificació.	QR codes generats i mostrats correctament.	OK

Taula 25. Proves al frontend

### 8.7 Proves de fluxos principals

Caso	Descripció	Esperat	OK
Flux complet d'usuari regular	Usuari es registra, puja ticket, rep punts i consulta historial.	Flux complet funcionant sense errors.	OK
Flux complet de venedor	Venedor envia ticket digital, usuari rep notificació i es guanya punts.	Flux complet funcionant correctament.	OK
Flux complet d'administrador	Admin crea segment amb IA, campanya i envia	Flux complet d'administració	OK

	notificacions.	funcionant.	
Integració entre serveis	Comprovació que tots els serveis es comuniquen correctament.	Integració completa funcionant.	OK
Gestió d'estats de l'aplicació	Comprovació que els estats es gestionen correctament.	Gestió d'estats funcionant.	OK

**Taula 26.** Proves de fluxos principals

## 9 Legislació i protecció de dades

En aquesta versió TFG he prioritzat la seguretat tècnica i la minimització de dades: identitats pseudonimitzades amb UUID, no exposo dades personals a les APIs públiques i tot el trànsit va xifrat (HTTPS amb Nginx), amb els serveis aïllats per microserveis. Això redueix la superfície d'atac i el risc de reidentificació, però per si sol no garanteix el compliment integral del RGPD/LOPDGDD.

Deliberadament, no he implementat la documentació legal (política de privacitat, avís legal, condicions d'ús), l'exercici de drets de les persones usuàries (accés, supressió, etc.), un sistema de consentiment granular i revocable ni una política de retenció i depuració de dades. Per tant, el projecte compleix en allò tècnic descrit, però resta pendent aquesta capa legal-organitzativa per a un llançament en producció.

## 10 Ètica i responsabilitat social

Pel que fa a la igualtat de gènere, l'aplicació acostava el mercat a la generació Z i ajuda a trencar l'estereotip que "és cosa d'àvies o de senyores de la casa". Presenta la compra de proximitat com una activitat compartida i actual, visible per a nois i noies per igual. El llenguatge i el disseny són inclusius, i la gamificació és neutra i flexible perquè cadascú participi al seu ritme, sense penalitzar qui disposa de menys temps.

Des de les fases inicials, el model de dades, les variables i els controladors no incorporen camps ni filtres discriminatoris (per raça, edat, gènere, orientació o qualsevol altra condició), evitant l'ús de variables proxy i la inferència de gènere. Això contribueix a normalitzar que la gestió de la compra i la cura sigui més equilibrada entre gèneres, alhora que minimitza la introducció i propagació de biaixos a nivell de producte.

Tal com recull l'article d'ICEX [5] sobre com la IA tendeix a estereotipar professions per sexes, els models generatius poden reproduir i amplificar rols de gènere si les dades i els controls no són adequats. Per això, en el nostre sistema sí que hem incorporat controls anti-biaix: rails explícits al "system prompt" (neutralitat, llenguatge inclusiu i prohibició d'inferir gènere), exclusió d'atributs demogràfics i variables proxy tant al model de dades com a les crides a la IA, verificacions post-sortida que detecten assignacions de gènere no sol·licitades i ofereixen alternatives neutres, i proves de regressió orientades a paritat. Amb això, reduïm la probabilitat de reproduir estereotips en CRM, notificacions i fluxos d'IA.

```

OLLAMA_HOST = os.getenv("OLLAMA_HOST", "http://ollama:11434")
LLM_MODEL = os.getenv("LLM_MODEL", "qwen2.5:0.5b-instruct")
SYSTEM_RAILS = os.getenv(
    "SYSTEM_RAILS",
    """Ets un assistent imparcial i no discriminator. Rails obligatoris:
- No infereixis gènere, raça, edat, religió o classe social si l'usuari no els proporciona explícitament.
- Usa llenguatge inclusiu i neutre; evita estereotips i rols de gènere (p. ex. associar professions a un gènere).
- No utilitzis cap atribut demogràfic per a decisions de segmentació, recomanació o prioritització excepte si és estrictament necessari i pro
- Si detectes que el prompt de l'usuari força estereotips o discriminació, ofereix una alternativa neutral i explica la limitació.
- Quan calgui referir-se a persones desconegudes, utilitza fórmules neutres i exemples equilibrats (paritat en noms i perfils).
- No utilitzis variables proxy (adreça, codi postal, nom propi, etc.) per inferir gènere o raça.
- Per a sortides creatives (còpies, notificacions), proporciona variants que mantinguin paritat i neutralitat.
- Per a extracció de dades (tiquets), limita't a camps objectius (import, comerç, data) i mai inferis gènere/raça.
- Si el context aporta pronoms preferits, respecta'ls; si no, mantén neutralitat."""
)

```

**Figura 45.** Rails explícits a la generació per IA de les notificacions

En medi ambient, s'incentiva l'ús del tiquet digital per defecte, reduint paper i residus. Les notificacions i els continguts poden promoure producte local i de temporada i l'aprofitament (descomptes de darrera hora, propostes anti-malbaratament), alhora que ajuden a planificar millor la visita al mercat i evitar desplaçaments innecessaris. L'historial de compres dona visibilitat sobre hàbits de consum i facilita decisions més conscients.

Des de la responsabilitat social, l'app reforça el mercat com a eix de barri: no el substitueix, sinó que l'enriqueix amb informació útil, comunicació directa entre paradistes i veïnat, i activitats que cohesionen la comunitat. L'experiència es manté senzilla i accessible (interfície clara, missatges entenedors), pensada perquè qualsevol persona la pugui utilitzar sense barreres. Som conscients que existeix bretxa d'edat i situacions d'exclusió digital; per això el sistema permet generar un codi QR personalitzat i delegable perquè un familiar o els serveis socials el puguin custodiar i presentar en nom de la persona

sense smartphone, de manera que també pugui gaudir dels descomptes i promocions. Aquesta delegació es fa amb consentiment explícit i es pot revocar en qualsevol moment.

En l'àmbit ètic, es prioritza el respecte a la privadesa i l'ús responsable de la tecnologia: es recullen les dades mínimes per oferir el servei, les opcions de notificacions i comunicacions són voluntàries i reversibles, i s'eviten pràctiques intrusives o confuses. La intel·ligència artificial s'utilitza per estalviar feina a l'usuari (extreure dades del tiquet) i no per prendre decisions que el perjudiquin. L'objectiu és aportar utilitat real al mercat i al barri, amb criteris de respecte, transparència i sentit comú.

## 11 Conclusions

Aquest projecte ha estat molt més que completar un treball acadèmic: he viscut una experiència realment enriquidora tant en l'àmbit formatiu com en l'organitzatiu. Tot i haver materialitzat idees en altres aplicacions durant la carrera, no havia afrontat mai un desenvolupament d'aquesta escala. Programar serveis web en Python ha estat una novetat per a mi i m'ha permès adquirir un bagatge tècnic rellevant. També ha estat la primera vegada que he integrat IA en un producte propi, fet que m'ha obligat a aprendre i a prendre decisions tecnològiques amb criteri. Paral·lelament, he consolidat una metodologia de treball per organitzar-me millor i avançar de manera sostinguda.

La meua experiència prèvia com a DevOps ha estat clau per sostenir gran part del pes del projecte: l'ús de Docker, la posada en marxa del servidor d'allotjament i la configuració de l'entorn han simplificat molt el desplegament. Habilitar l'SSL i l'HTTPS m'ha suposat un repte nou que he resolt amb èxit, sumant competències pràctiques que van més enllà del codi i entren de ple en l'operativa d'un servei real.

També reconec que queden línies de treball per desenvolupar amb temps: integrar un mòdul de BI per analítica avançada, explorar a fons la publicació i proves a iOS, i afegir funcionalitats que han quedat pendents per calendari. Aquestes millores apunten a una evolució natural del projecte cap a més maduresa i impacte.

En conclusió, he après molt i em sento satisfet amb el resultat i amb el camí recorregut.

## 12 Recursos Utilitzats

Docker, Inc. (s. f.). Docker 101 Tutorial. Docker. Recuperat el 2 de setembre de 2025, de <https://www.docker.com/101-tutorial/>

Agile Business Consortium. (s. f.). Chapter 10: MoSCoW Prioritisation. DSDM Agile Project Framework Handbook. Recuperat el 2 de setembre de 2025, de <https://www.agilebusiness.org/dsdm-project-framework/moscow-prioritisation.html>

ProductPlan. (2025). What is MoSCoW Prioritization? ProductPlan Glossary. Recuperat el 2 de setembre de 2025, de <https://www.productplan.com/glossary/moscow-prioritization/>

React. (s. f.). Quick Start – React. React Documentation. Recuperat el 2 de setembre de 2025, de <https://react.dev/learn>

MDN Web Docs. (s. f.). Getting started with React. Mozilla Developer Network. Recuperat el 2 de setembre de 2025, de [https://developer.mozilla.org/en-US/docs/Learn\\_web\\_development/Core/Frameworks\\_libraries/React\\_getting\\_started](https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Frameworks_libraries/React_getting_started)

Python Software Foundation. (2025). Python 3.13.7 documentation. Recuperat el 2 de setembre de 2025, de <https://docs.python.org/3/>

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining “gamification.” In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (pp. 9–15). ACM.

### **Programari utilitzat:**

Bayer, M. (2006). SQLAlchemy [Llibreria de programari].

Colvin, S., i l'equip de Pydantic. (2018). Pydantic [Llibreria de programari].

Docker, Inc. (2013). Docker [Programari].

Docker, Inc. (2014). Docker Compose [Programari].

Encode OSS Ltd. (2018). Uvicorn [Servidor ASGI].

Internet Engineering Task Force. (2016). The Web Push Protocol (RFC 8030) [Estàndard].

Internet Engineering Task Force. (2018). Voluntary Application Server Identification (VAPID) for Web Push (RFC 8292) [Estàndard].

PostgreSQL Global Development Group. (1996). PostgreSQL [Sistema de gestió de bases de dades].

Ramírez, S. (2019). FastAPI [Marc de treball].

Schlawack, H. (2013). structlog [Llibreria de programari].

Python Software Foundation. (1991). Python (Versió 3.x) [Llenguatge de programació i entorn].

Ollama, Inc. (2023). Ollama (Local LLM runtime) [Plataforma].

Google. (2024). Gemini (multimodal AI system) [Plataforma].

## Referències

- [1] Looyestyn, J., Kernot, J., Boshoff, K., Ryan, J., Edney, S., & Maher, C. (2017). Does gamification increase engagement with online programs? A systematic review. PLOS ONE, 12(3), e0173403. <https://doi.org/10.1371/journal.pone.0173403> Text complet a PLOS ONE | PMC | PubMed
- [2] Sailer, M., & Homner, L. (2020). The gamification of learning: A meta-analysis. Educational Psychology Review, 32, 77–112. <https://doi.org/10.1007/s10648-019-09498-w>
- [3] Parlamento Europeo y del Consejo. (2016, 27 de abril). Reglamento (UE) 2016/679 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos, y por el que se deroga la Directiva 95/46/CE (Reglamento general de protección de datos). Diario Oficial de la Unión Europea, L 119, 1–88 (4 de mayo de 2016). <https://www.boe.es/doue/2016/119/L00001-00088.pdf>
- [4] España. Jefatura del Estado. (2018, 5 de diciembre). Ley Orgánica 3/2018, de Protección de Datos Personales y garantía de los derechos digitales. Boletín Oficial del Estado (BOE), núm. 294, 6 de diciembre de 2018. <https://www.boe.es/buscar/pdf/2018/BOE-A-2018-16673-consolidado.pdf>
- [5] ICEX España Exportación e Inversiones. (s. f.). La inteligencia artificial estereotipa profesiones por sexos. Radar ICEX – Mujer e internacionalización. Recuperat el 2 de setembre de 2025, de <https://www.icex.es/es/radar-icex/mujer-internacionalizacion/al-dia/inteligencia-artificial-estereotipa-profesiones-por-sexos>
- [6] Docker, Inc. (s. f.). Docker 101 Tutorial. Docker. Recuperat el 2 de setembre de 2025, de <https://www.docker.com/101-tutorial/>
- [7] Agile Business Consortium. (s. f.). Chapter 10: MoSCoW Prioritisation. DSDM Agile Project Framework Handbook. Recuperat el 2 de setembre de 2025, de <https://www.agilebusiness.org/dsdm-project-framework/moscow-prioritisation.html>
- [8] ProductPlan. (2025). What is MoSCoW Prioritization? ProductPlan Glossary. Recuperat el 2 de setembre de 2025, de <https://www.productplan.com/glossary/moscow-prioritization/>
- [9] React. (s. f.). Quick Start – React. React Documentation. Recuperat el 2 de setembre de 2025, de <https://react.dev/learn>
- [10] MDN Web Docs. (s. f.). Getting started with React. Mozilla Developer Network. Recuperat el 2 de setembre de 2025, de [https://developer.mozilla.org/en-US/docs/Learn\\_web\\_development/Core/Frameworks\\_libraries/React\\_getting\\_started](https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Frameworks_libraries/React_getting_started)
- [11] Python Software Foundation. (2025). Python 3.13.7 documentation. Recuperat el 2 de setembre de 2025, de <https://docs.python.org/3/>
- [12] Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining “gamification.” In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (pp. 9–15). ACM.

## **Annex**

El codi font es troba en aquest repositori de GitHub:

[https://github.com/yaniaici/TFG\\_Mercats](https://github.com/yaniaici/TFG_Mercats)

El projecte es troba publicat al domini:

<https://mercatmediterrani.com>