
This is the **published version** of the bachelor thesis:

Vacas Martínez, Albert. *CampusFix : Plataforma de Gestió Col·laborativa d'Incidències al Campus UAB*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334390>

under the terms of the  license.

CampusFix: Plataforma de Gestió Col·laborativa d'Incidències al Campus UAB

Albert Vacas Martínez

30 de juny de 2026

Resum Aquest projecte presenta un software multiplataforma per a la gestió d'incidències al campus universitari, connectant els alumnes, els tècnics i els administradors. Mitjançant una màquina d'estats finits (XState), es garanteix la traçabilitat del manteniment en cinc estats: 1) Oberta, 2) Assignada, 3) En procés, 4) Validada i 5) Tancada. El sistema integra geolocalització avançada (PostGIS i Leaflet) per visualitzar els mapes de calor i els clústers de desperfectes. A més, s'utilitza la intel·ligència artificial (Google Gemini i LangGraph) per a la classificació i revisió automàtica en la creació d'incidències i assignació de tasques als tècnics. La solució es completa amb notificacions en temps real i un sistema de gamificació per incentivar la participació de l'estudiant.

Paraules clau Gestió d'incidències, Màquina d'estats finits, Sistema d'Informació Geogràfica (SIG), Intel·ligència Artificial, Notificacions en temps real, Gamificació, Aplicació multiplataforma.

Abstract This project presents a multiplatform software for incident management on the university campus, connecting students, technicians and administrators. Using a finite state machine (XState), the traceability of maintenance is guaranteed in five states: 1) Open, 2) Assigned, 3) In process, 4) Validated and 5) Closed. The system integrates advanced geolocation (PostGIS and Leaflet) to visualize heat maps and clusters of defects. In addition, artificial intelligence (Google Gemini and LangGraph) is used for automatic classification and review in the creation of incidents and assignment of tasks to technicians. The solution is completed with real-time notifications and a gamification system to encourage student participation.

Keywords Incident management, Finite state machine, Geographic Information System (GIS), Artificial Intelligence, Real-time notifications, Gamification, Cross-platform application.

1 INTRODUCCIÓ

EN els darrers anys, l'avenç de la Intel·ligència Artificial (IA) ha estat notori, transformant radicalment la nostra capacitat per processar la informació i automatitzar les tasques complexes. Aquesta eclosió tecnològica ofereix una oportunitat sense precedents per millorar diversos aspectes de la societat, aportant eficiència i precisió en àmbits que fins ara depenien de processos manuals i lents.

Dins d'aquest paradigma, es proposa el present projecte com una solució innovadora per al manteniment del Campus. L'objectiu és garantir que les instal·lacions es mantinguin en un bon estat mitjançant una plataforma que apro-

fiti la potència de la IA per a la classificació d'incidències i l'anàlisi de prioritats basant-se en les imatges. Aquesta aplicació no només busca l'eficiència operativa, sinó que vol millorar substancialment l'experiència dels alumnes, convertint-los en agents actius del seu entorn.

L'adopció d'una solució àgil i flexible com la que es presenta, comporta avantatges significatius: la capacitat de resposta en temps real, la transparència en el seguiment de les incidències mitjançant una màquina d'estats definida i la facilitat de report des de dispositius mòbils amb geolocalització. La implementació d'aquest sistema suposarà un gran canvi en la gestió del Campus, modernitzant la comunicació entre la comunitat universitària i els equips de manteniment.

2 ESTAT DE L'ART

La gestió d'infraestructures en grans complexos com un campus universitari ha estat tradicionalment un procés reactiu i unidireccional. Històricament, la detecció i resolució de desperfectes s'ha basat en formularis web estàtics, correus electrònics a centres de suport o trucades telefòniques.

• E-mail de contacte: 1665473@uab.cat
 • Menció: Enginyeria del Software
 • Treball tutoritzat per: Sergio Bachiller Rubia (Àrea de Ciències de la Computació i Intel·ligència Artificial)
 • Curs 2025/26

Aquest model genera colls d'ampolla en l'administració a causa del triatge manual, i un efecte de "caixa negra" per a l'usuari, el qual rarament rep retroalimentació sobre l'estat de la seva petició. El present projecte s'emmarca en la necessitat de modernitzar aquest flux, analitzant les solucions actuals del mercat i determinant quins aspectes diferencials s'aporten per transformar la UAB en un entorn col·laboratiu i intel·ligent.

2.1 L'Evolució dels sistemes tradicionals

Fins fa poc, el programari de manteniment es limitava a grans sistemes de gestió de manteniment limitats, dissenyats exclusivament per a tècnics i administradors. Aquests sistemes han evolucionat a sistemes més flexibles i útils per als usuaris aprofitant l'ús dels telèfons mòbils i sistemes de comunicacions moderns.

Avui dia, els dispositius mòbils no només permeten la comunicació, sinó que actuen com a sensors distribuïts gràcies a l'equip integrat (càmeres d'alta resolució, mòduls GPS de gran precisió). Les empreses han aprofitat aquesta onada tecnològica per poder desenvolupar les aplicacions que permeten gestionar i processar les incidències de les infraestructures de manera remota. Aquests sistemes permeten als usuaris crear eventuais modificacions amb comoditat i veure el seguiment d'aquestes. Aquest projecte presenta una arquitectura híbrida (Web per a administració, App per a usuaris/tècnics) que descentralitza la detecció d'incidències i n'automatitza el processament, reduint dràsticament els temps de resposta.

2.2 L'Anàlisi de Solucions Existents

Per situar el projecte en el panorama tecnològic actual, s'ha analitzat dos models de referència: l'aplicació "Find It, Fix It" de la ciutat de Seattle i el model de mobilitat compartida de Bolt CityFix. Aquest estudi comparatiu permet identificar els buits operatius que la present plataforma pretén resoldre.

- **Find It, Fix It (Ciutat de Seattle)**[10]: Aquest sistema actua sota una arquitectura d'integració en sistemes empresarials complexos, permet gestionar les incidències de la ciutat de Seattle. S'usa per adjuntar evidències visuals a les incidències i utilitza metadades d'ubicació per centralitzar els reports. Permet visualitzar gràficament en un mapa on s'ubiquen les incidències.

Tot i això, no hi ha cap plataforma externa web d'administració i els usuaris sovint perceben una "manca d'acció" quan els seus casos es tanquen prematurament a causa de redireccionaments interns entre departaments.

- **Bolt CityFix**[11]: Bolt és una enorme aplicació europea de mobilitat que ofereix serveis de transport sota demanda (VTC i taxi), lloguer de patinets/bicicletes elèctriques i repartiment de menjar. Bolt aprofita la seva xarxa de micromobilitat preexistent per convertir els conductors i usuaris en sensors actius de la ciutat mitjançant el report d'incidències i millores.

Destaca per la immediatesa i el volum de reports (forats, senyals, semàfors) inabastable per a un ciutadà

mitjà, integrant l'eina dins d'una "super-app" de mobilitat. El seu enfocament és eminentment reactiu i centrat en la infraestructura de transport, per això manca d'una interfície d'administració profunda per a tècnics.

2.3 Proposta de valor

A diferència de les solucions estàndard de mercat, aquest projecte es posiciona com una plataforma activa i intel·ligent, reforçant en tres aspectes que resolen les mancances de les aplicacions de gestió actuals:

- **Gestió d'incidències amb agents i IA:** Mitjançant la integració de models de llenguatge (LLMs) via LangChain/LangGraph i Google Gemini, el sistema no és un simple gestor de base de dades. Els agents d'IA analitzen el llenguatge natural i les imatges aportades per l'usuari, classifiquen automàticament la categoria de la incidència, en determinen la prioritat segons criteris de seguretat i generen un resum sintètic per facilitar el triatge administratiu. Això descarrega el personal administratiu de tasques repetitives. Aquest aspecte té molt de potencial, ja que amb la implementació d'agents es pot escalar l'aplicació automatitzant la majoria de les funcionalitats. Fins a arribar al punt de crear una plataforma d'incidències completament automàtica que actuï sota la revisió i control humà.
- **Interfície web d'administració:** Aquest projecte implementa una plataforma tant mòbil com d'escriptori, que permet als usuaris administradors la visualització de dades i gràfics en una interfície còmoda d'escriptori. Gràcies a l'ús de Server-Sent Events (SSE), les mètriques dels gràfics i mapes s'actualitzen instantàniament, cosa que permet una coordinació d'emergències altament eficient.
- **Sistema de recompenses (Gamificació):** Per combatre el report deficient o malintencionat, s'implementa un mecanisme de puntuació on les bones pràctiques es premien (fomentant l'esperit de cuidar el campus), mentre que el mal ús deliberat està vinculat a sancions acadèmiques i bloqueig del compte, garantint així l'alta fiabilitat de les dades d'entrada.

3 OBJECTIUS

El projecte present es planteja amb l'objectiu principal de resoldre la necessitat de manteniment i millora de les infraestructures del campus universitari. Mitjançant tecnologies de la informació serem capaços de notificar als estudiants, mantenir i gestionar el Campus en bones condicions per garantir una bona experiència als alumnes i professors. Els objectius se centren a maximitzar l'eficiència del manteniment i potenciar el sentiment de comunitat.

- **Optimització de la disponibilitat dels recursos del campus:** Reduir dràsticament el temps de permanència de les infraestructures en mal estat (com ara ordinadors de sales d'estudi, mobiliari d'aules o il·luminació). L'objectiu és que qualsevol element defectuós sigui detectat i reportat en qüestió de minuts, minimitzant l'impacte en l'activitat acadèmica.

- **Foment de la corresponsabilitat i la participació ciutadana:** Transformar l'alumne de "subjecte passiu" a "agent actiu" del manteniment del campus. Mitjançant el sistema de recompenses i gamificació, s'estimula un comportament cívic i proactiu, creant una comunitat que vetlla pel seu entorn.
- **Millora de l'eficiència operativa dels equips tècnics:** Proporcionar als operaris de manteniment una eina que prioritzí les tasques segons la gravetat i la ubicació. Això permet assignar recursos d'una forma més intel·ligent, passant d'un model de manteniment reactiu a un model més organitzat i basat en dades reals.
- **Anàlisi de dades per a la gestió estratègica del Campus:** Utilitzar la informació geolocalitzada recollida per identificar problemes estructurals o zones amb una degradació superior a la mitjana. Això servirà de suport a la presa de decisions de la universitat i al control de l'estat actual del Campus.
- **Reducció de costos administratius mitjançant l'automatització:** Delegar en la Intel·ligència Artificial el triatge i la classificació d'incidències per reduir la càrrega de treball manual dels gestors. L'objectiu és permetre que els recursos humans es concentrin en les tasques essencials.

4 METODOLOGIA

Per a la realització d'aquest projecte s'ha optat per una metodologia de desenvolupament àgil basada en el marc de treball Scrum[4], adaptada a la naturalesa individual d'un Treball de Final de Grau. Aquest enfocament permet una gestió iterativa i incremental, assegurant que les funcionalitats més crítiques es trobin en una base sòlida abans d'afegir capes de complexitat.

Per dur a terme aquesta metodologia, en primer lloc, es defineix el *Product Backlog* a partir dels requisits generats i funcionalitats definides. A partir d'aquí, s'extreu el *Sprint Backlog*, que representa el subconjunt de tasques seleccionades per a ser executades en cada *Sprint*. Finalment, cada cicle culmina amb un Increment, que és el resultat tangible i funcional del programari que s'afegeix de forma acumulativa a les versions anteriors, garantint així una evolució del projecte constant, robusta i verificable en cada lliurament.

En aquest cas, en lloc de fer reunions diàries es farà un seguiment mensual amb el tutor del TFG (*Sprint Review i Retrospective*) per avaluar l'increment generat.

Aquesta metodologia és ideal per a aquest projecte perquè permet integrar components complexos (com la IA i els SIG) de manera progressiva, garantint que la base del sistema sigui sòlida abans d'afegir-hi capes addicionals.

5 PLANIFICACIÓ

Per establir una referència temporal clara i assolible, la planificació es divideix en blocs de treball de dues setmanes de durada, anomenats Sprints. Aquest enfocament permet fragmentar la complexitat del sistema en etapes consecutives, on cada cicle té com a objectiu el lliurament d'un increment funcional del programari. En aquest projecte hem definit Sprints de dues setmanes per tenir més marge de desen-

volupament. A continuació es mostra el diagrama de Gantt complet del projecte:

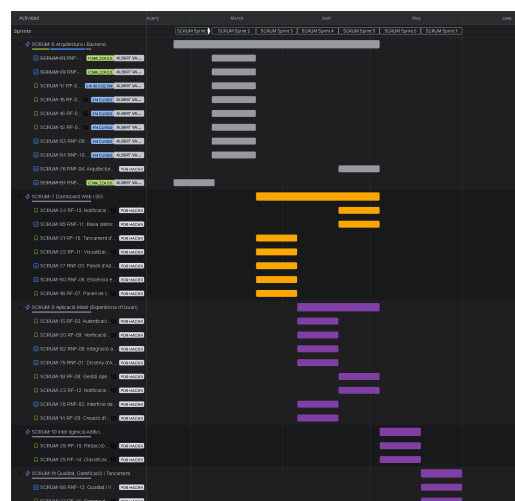


Fig. 1: Diagrama de Gantt

Els sprints que es duran a terme en aquest projecte són els següents:

Sprint 1: Definició i Disseny d'Arquitectura.

Aquest primer cicle es focalitza en la concreció dels objectius, l'anàlisi de l'estat de l'art, desenvolupament dels requisits i el disseny de l'esquema de dades relacional. Es defineix l'arquitectura del sistema i el model conceptual de la màquina d'estats que governarà les incidències.

Sprint 2: Infraestructura Backend i Lògica de Control.

Durant aquesta fase es realitza la configuració del servidor amb Node.js i l'ORM Prisma. S'implementa la base de dades PostgreSQL amb l'extensió PostGIS per a la gestió geogràfica i es programa la lògica de transicions d'estats amb XState.

Sprint 3: Desenvolupament de la Interfície Web i SIG.

Es construeix el dashboard d'administració amb React, integrant les peticions necessàries al server API de backend per a la visualització de clústers i mapes de calor. S'estableixen les eines de filtratge i assignació manual de tasques.

Sprint 4: Mòdul de l'Alumne i Aplicació Mòbil.

Inici del desenvolupament amb React Native i Expo[3]. Es crea el formulari de report d'incidències, integrant l'accés natiu a la càmera i la captura de coordenades GPS.

Sprint 5: Sistema de Notificacions i Comunicació en Temps Real.

Es desenvolupa el sistema de notificacions del producte: push notifications al mòbil mitjançant Expo Push Service per a estudiants i tècnics, i actualització en temps real del panell admin via Server-Sent Events (SSE). Tot orquestrat des d'un *NotificationService* centralitzat al backend que decideix qui ha de rebre cada esdeveniment de domini.

Sprint 6: Integració d'Intel·ligència Artificial i Auto-assignació.

En aquesta etapa s'integren Google Gemini i LangGraph al backend per permetre la classificació automàtica d'incidències a partir del text i la imatge. S'implementa també l'auto-assignació en lot al panell admin, basada en la categoria d'expertesa del tècnic, la càrrega de tasques i ubicació.

Sprint 7: Gamificació i Control de Qualitat (QA).

Es programa la lògica del sistema de recompenses i rànquings. Es realitza una fase intensiva de proves de sistema i correcció d'errors per garantir la robustesa de la plataforma i el compliment del RGD.

Sprint 8: Preparació de la defensa

El darrer cicle es dedica al tancament de la memòria escrita, el refinament final de l'experiència d'usuari (UI/UX) i la preparació dels materials per a la defensa pública del projecte.

6 REQUISITS

Per garantir l'èxit en el disseny i desenvolupament de la plataforma integral de gestió de desperfectes del campus universitari, s'han identificat i classificat els següents requisits funcionals i no funcionals. Aquests requisits delimiten l'abast tècnic del projecte i satisfan les necessitats dels diferents actors involucrats.

6.1 Requisits Funcionals

Aquests requisits defineixen les capacitats d'interacció i el comportament del sistema davant les accions dels usuaris:

- **RF-01: Gestió de rols i privilegis.** El sistema ha de permetre l'accés a tres perfils diferenciats: Administrador (visió global i control), Tècnic (resolució operativa) i Alumne (report d'incidències).
- **RF-02: Autenticació d'usuaris.** El sistema ha de permetre el registre i inici de sessió amb comptes de Google i Outlook per a aquells que disposin de compte @autonoma.cat.
- **RF-03: Creació d'incidències mòbils.** Els alumnes han de poder registrar incidències des de l'app amb estat "Oberta", adjuntant descripció, coordenades GPS i proves fotogràfiques.
- **RF-04: Control del flux d'estats (XState).** El sistema ha de governar el cicle de vida de la incidència mitjançant 5 estats finits: Oberta, Assignada, En procés, Validada i Tancada.
- **RF-05: Restricció de transicions.** Només es pot assignar un usuari amb rol "Tècnic" per moure una incidència que està "Oberta" a l'estat "Assignada".
- **RF-06: Traçabilitat d'actors.** Cada incidència ha d'estar vinculada obligatòriament a un ID d'usuari (creador) i a un ID de tècnic (assignat).
- **RF-07: Panell de control d'administració.** Els administradors han de poder visualitzar llistats, filtrar incidències i assignar-les a tècnics recomanats.
- **RF-08: Gestió operativa per a tècnics i auto-assignació.** Els tècnics han de poder consultar i gestionar les tasques que tenen assignades. Els administradors han de disposar d'una eina d'auto-assignació per distribuir les incidències entre els tècnics.
- **RF-09: Verificació fotogràfica de resolució.** Els tècnics han de reportar la solució adjuntant fotos de la tasca solucionada per canviar l'estat a "Validada".
- **RF-10: Tancament d'incidències.** L'administrador ha de rebre les peticions "Validades" i determinar si es tanquen definitivament o es reassignen.
- **RF-11: Visualització cartogràfica (SIG).** El sistema ha de representar les incidències en un mapa interactiu amb clustering i mapes de calor per zones.
- **RF-12: Notificacions Push.** Enviament d'alertes als dispositius mòbils d'usuaris i tècnics davant canvis d'estat o avisos crítics.
- **RF-13: Actualització en temps real.** El tauler d'administració s'ha d'actualitzar instantàniament en rebre nous reports o canvis de disponibilitat de tècnics.
- **RF-14: Classificació automàtica per IA.** Ús d'agents d'IA per analitzar imatges, classificar el tipus d'incidència i determinar la prioritat.
- **RF-15: Generació de resums automàtics per IA.** L'agent d'IA genera un resum d'una línia (*aiSummary*) per a cada incidència classificada, facilitant la lectura ràpida del panell d'administració.
- **RF-16: Sistema de Recompenses i moderació.** Implementació d'un rànquing de punts per a usuaris col·laboradors. El mal ús no es penalitza automàticament amb una resta de punts, sinó que es gestiona a nivell de política institucional (sancions acadèmiques) i, operativament, mitjançant les eines de moderació de l'administrador —bloqueig o eliminació de comptes (RF-17).
- **RF-17: Gestió d'usuaris i perfils.** El sistema ha de permetre la cerca d'usuaris, la pujada de fotos de perfil i que l'administrador pugui bloquejar, reactivar o eliminar comptes amb tancament de sessió immediat dels afectats, preservant l'històric d'incidències.

6.2 Requisits No Funcionals

Requisit	Descripció
RNF-01 Usabilitat (UX/UI)	Interfície intuïtiva i adaptada a cada plataforma (web i app mòbil); reportar una incidència en menys de 30 s.
RNF-02 Multiplataforma	Lògica centralitzada al backend, accessible des del web i d'iOS/Android (Expo).
RNF-03 Rendiment geoespacial	Consultes de proximitat amb PostGIS (ST_Distance) en < 500 ms.
RNF-04 Recursos natius	Captura de GPS i càmera amb precisió mínima de 10-20 m.
RNF-05 Seguretat (RBAC)	Control d'accés per rols que restringeix les rutes i operacions de l'API.
RNF-06 Privacitat (RGPD)[5]	Xifratge de dades confidencials i trànsit segur via TLS (HTTPS) a producció; anonimització en l'esborrament de comptes (dret a l'oblit).
RNF-07 Temps real	Actualitzacions del panell via SSE amb latència gairebé instantània (< 200 ms).
RNF-08 Mantenibilitat	Disseny modular i ORM (Prisma) amb tipatge estàtic (TypeScript) per facilitar l'ampliació.
RNF-09 Internacionalització (i18n)	Interfície disponible en català, castellà i anglès, amb canvi d'idioma en calent i persistència de la preferència (localStorage al web, AsyncStorage al mòbil).

7 EINES UTILITZADES

Tot el projecte s'ha desenvolupat en **TypeScript**. El **backend** corre sobre **Node.js** amb **Express**, accedeix a les dades amb l'ORM **Prisma**[7, 8], governa el cicle de vida amb la màquina d'estats **XState**, i resol la seguretat amb **JWT** i **bcrypt**. La capa d'IA s'orquestra amb **LangGraph**[15] i el model multimodal **Google Gemini 2.5 Flash**[16]. El **frontend web** és una SPA amb **React**[12] i **Vite**, estilitzada amb **Tailwind CSS**, amb **Recharts** per als gràfics i **Leaflet** per als mapes. L'**app mòbil** usa **React Native + Expo**[13], reaprofitant l'estil del web via **NativeWind** i els SDK natius d'Expo per al GPS, la càmera, l'emmagatzematge segur del JWT i les notificacions. Finalment, la **base de dades** PostgreSQL amb l'extensió geoespacial **PostGIS** i l'emmagatzematge d'imatges s'allotgen a **Supabase**[9], i el codi es versiona amb **GitHub**.

8 DESENVOLUPAMENT

8.1 Estructura backend

L'arquitectura del backend segueix el patró Routes - Middleware - Controller - Service - Database, on cada capa té una responsabilitat clarament definida. Aquesta separació de responsabilitats facilita la mantenibilitat i l'escalabilitat del sistema, ja que permet modificar la lògica de negoci sense afectar a les capes de presentació o d'accés a dades.

8.1.1 Arquitectura de la Base de Dades

La base de dades és PostgreSQL allotjada a Supabase, amb l'extensió PostGIS per a les capacitats geoespacionals. L'esquema es defineix amb Prisma ORM, que proporciona tipatge automàtic, migracions versionades i un client generat.

S'ha definit un esquema robust amb les següents taules principals:

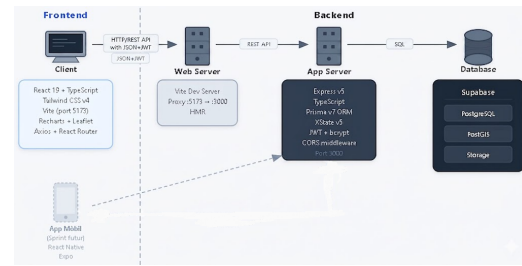


Fig. 2: Arquitectura Frontend web/Backend

Taula	Descripció
users	Usuaris amb rol (STUDENT, ADMIN, TECHNICAL), credencials, estat actiu i punts de gamificació. Els tècnics afegeixen workCategory (àmbit d'expertesa usat per l'auto-assignació), càrrec i empresa.
reports	Incidències, amb estat (XState), prioritat, categoria, coordenades i columna geogràfica PostGIS (geography(Point, 4326)), creador i tècnic assignat, resolvedAt i auditoria de la IA (aiSummary, aiClassifiedAt).
comments	Fil de discussió per incidència (autor i report), amb un camp opcional transitionEvent que lliga el comentari a una transició d'estat (p. ex. el motiu d'un rebuig).
images	Imatges d'un report, tipificades pel cicle de vida (INITIAL, PROGRESS, RESOLUTION); a la BD només es desa l'URL (el fitxer viu a Supabase Storage).
invites	Invitacions per a registre privilegiat, amb email, rol, token únic, estat (PENDING / USED / REVOKED) i caducitat (7 dies per defecte).
push.tokens	Tokens de push per dispositiu, lligats a l'usuari. El camp active permet el soft-delete quan Expo informa que el dispositiu s'ha desregistrat.
notifications	Historial persistent de notificacions (tipus, títol, cos, flag de llegida); font de veritat del badge de la campanya, independent de si la push s'ha entregat.
points-transactions	Auditoria de la gamificació: lliga cada report tancat amb els punts atorgats i la prioritat (snapshot). La restricció UNIQUE sobre reportId en garanteix la idempotència.

8.1.2 Màquina d'Estats

Per governar el cicle de vida de les incidències (RF-04), s'ha implementat una màquina d'estats finits utilitzant la llibreria **XState** v5 amb l'API setup(). Aquesta decisió respon a la necessitat de centralitzar i formalitzar les regles de transició, evitant condicions disperses pel codi que serien difícils de mantenir i verificar.

La màquina defineix cinc estats (Oberta, Assignada, En procés, Validada i Tancada) i sis transicions, cadascuna protegida per guards que verifiquen el rol de l'usuari que sol·licita el canvi. Per exemple, només un administrador pot tancar definitivament una incidència.

La integració amb la base de dades és especialment rellevant: quan un servei necessita executar una transició, crea un actor **XState**, restaurant l'estat actual des de la base de dades amb resolveState(). Abans d'executar la transició, valida amb snapshot.can() si és permesa, permetent retornar missatges d'error descriptius sense haver de modificar les dades.

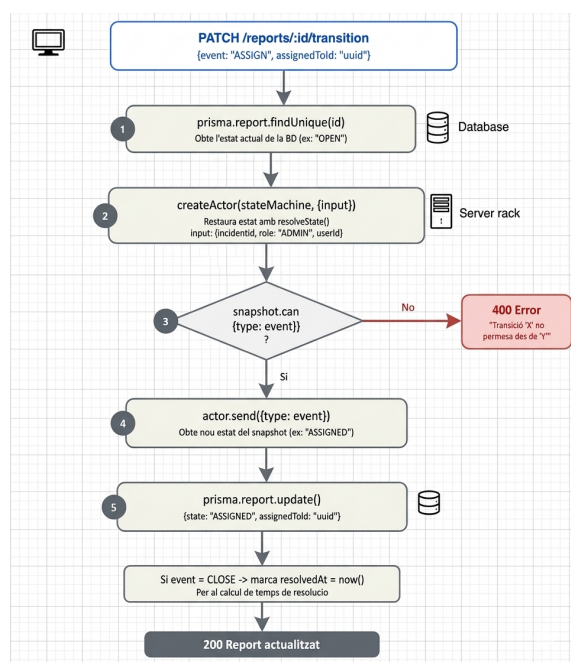


Fig. 3: Executar transició

8.1.3 Middleware: Xifratge i RBAC

Per controlar la seguretat del backend i restringir l'accés a determinats usuaris s'ha afegit un middleware que actua com a capa intermèdia, filtrant totes aquelles peticions malicioses o invàlides i controlant l'accés dels usuaris.

La primera part del middleware (authenticate) s'encarrega de verificar la sessió d'un usuari lícit. En rebre una petició, extreu el token del header Authorization, verifica la seva signatura criptogràfica mitjançant `jwt.verify()`. Si és vàlid vol dir que l'usuari és correcte i es retorna el payload descodificat (identificador d'usuari i rol). En cas contrari es rebutja amb un codi 401.

La segona part del middleware (authorize) implementa el control d'accés basat en rols (RBAC). Verifica si l'usuari autenticat posseeix un dels rols requerits. En cas contrari, retorna un codi 403.

Per al registre, la contrasenya es protegeix amb `bcrypt`. Per a l'inici de sessió, es compara la contrasenya proporcionada amb el hash emmagatzemat i, si coincideix, es genera un token JWT amb una expiració de 24 hores. En ambdós casos, la contrasenya mai es retorna al client, complint amb els principis de protecció de dades del RGPD.

8.1.4 Seguretat

S'ha implementat el patró de registre per invitació, inspirat en el model Zero Trust: el registre d'estudiants és públic, però el backend ignora el camp `role` i força `STUDENT` programàticament. Per als rols privilegiats (`ADMIN` i `TECHNICAL`), es requereix un token d'invitació generat prèviament per un administrador a l'hora de registrar-se.

Un cop un usuari privilegiat s'ha registrat, la creació de l'usuari i el marcatge de la invitació com a utilitzada s'executen dins d'una transacció atòmica de Prisma. Si una de les dues operacions fallen, cap de les dues s'aplica, prevenint inconsistències com un token reutilitzable o un token perdut.

Per prevenir situacions irreversibles, s'han establert dues regles de protecció:

- La primera protegeix un administrador root (configurat via variable d'entorn) que no pot ser revocat sota cap circumstància, garantint que sempre existeixi almenys un punt d'accés al sistema.
- La segona impedeix revocar l'últim administrador actiu mitjançant un comptatge previ a la base de dades. Si només resta un administrador actiu, l'operació es cancel·la, protegint el sistema de possibles deadlocks.

8.2 Panell de control

L'aplicació consta de set pàgines principals, organitzades dins d'un layout comú amb una barra lateral de navegació fixa i una zona de contingut principal.

8.2.1 Dashboard

S'ha escollit `Recharts` com a llibreria de visualització per la seva integració nativa amb `React`. El tauler disposa de dos filtres globals a la capçalera: granularitat (diària, setmanal o mensual) i període temporal (30, 90, 180 dies o un any). S'organitzen les gràfiques en blocs:

- El Bloc Operatiu respon a la pregunta «Quin és l'estat actual del sistema?» mitjançant targetes KPI clicables, una targeta destacada amb el percentatge d'incidències crítiques i altes, i un gràfic de rosca que mostra la distribució proporcional dels cinc estats.
- El Bloc Temporal analitza les tendències amb un gràfic d'incidències per categoria que incorpora un **filtre de rang de dates** propi (es pot seleccionar un interval concret o acotar-lo fins a un sol dia), i un gràfic de barres agrupades que compara les incidències creades amb les tancades al llarg del temps.
- El Bloc de Rendiment avalua l'eficiència dels recursos amb un gràfic de barres horitzontals apilades per tècnic i un gràfic de dispersió que relaciona el temps de resolució amb la prioritat.
- El Bloc Social mostra la distribució global per categoria i un rànquing dels usuaris que més incidències han reportat.

8.2.2 Administració d'usuaris

S'ha implementat una pàgina inicial d'inici de sessió que permet només als usuaris administradors accedir al panell de control. L'accés es verifica tant al frontend (comprovació del rol `ADMIN`) com al backend mitjançant el middleware (mòduls `authenticate` i `authorize`).

La cerca d'usuari (RF-17) està disponible tant a l'app com al panell web. La cerca es resol al servidor amb paginació, de manera que escala a milers d'usuaris.

8.2.3 Visualització geogràfica i SIG

S'ha habilitat l'extensió `PostGIS` a `Supabase` per afegir capacitats geoespacials a la base de dades `PostgreSQL`.

S'ha escollit Leaflet com a motor de mapes i s'ha creat un trigger PL/pgSQL que transforma automàticament els camps decimals latitud i longitud en un punt `geography(Point, 4326)` de PostGIS.

Cal distingir dues responsabilitats. Al **servidor**, la columna geomètrica s'utilitza per a les consultes espacials de *proximitat*: el desempat de l'auto-assignació (secció 8.6.2) calcula amb `ST_Distance` quin tècnic té la incidència activa més propera a la nova, aprofitant un índex GiST sobre la columna. Al **client**, en canvi, l'agrupació visual de markers i el mapa de calor es resolen amb Leaflet a partir dels punts (latitud/longitud) servits pel backend.

S'han integrat dos plugins específics: `leaflet.markercluster` per a l'agrupació visual de markers per proximitat amb animacions de dispersió, i `leaflet.heat` per a la generació de mapes de calor amb gradient configurable.

Aquests plugins permeten visualitzar en el mapa de manera ordenada els punts on s'ubiquen les incidències i s'agrupen per proximitat (clustering).

D'altra banda, també s'ha implementat un mapa de calor. Es pot seleccionar el criteri de pes entre tres opcions (prioritat, densitat i antiguitat) per obtenir perspectives analítiques diferents.

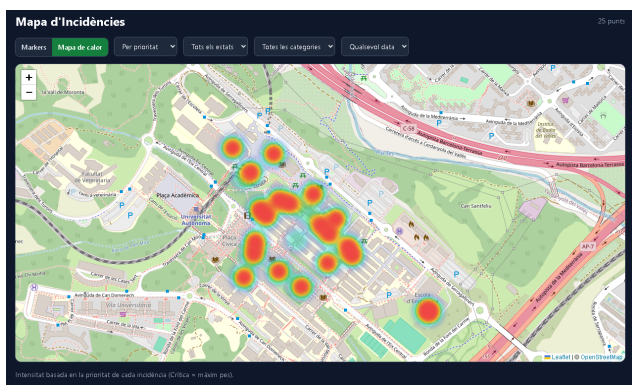


Fig. 4: Heatmap

8.3 Frontend app

L'aplicació mòbil està desenvolupada amb React Native i Expo, amb navegació basada en l'estructura de carpetes (Expo Router) que distingeix les pantalles públiques (inici de sessió i registre) de les protegides per autenticació. L'estil reutilitza Tailwind via NativeWind, mantenint coherència amb el panell web.

Tant el panell web com l'app mòbil suporten diversos idiomes (català, castellà i anglès) mitjançant `i18next` (RNF-09). Els textos d'interfície i les etiquetes de domini (categories, estats i prioritats) es resolen per claus de traducció.

8.3.1 Inici de sessió i registre

La sessió se centralitza en un context d'autenticació que restaura l'usuari en arrencar l'app si troba un JWT vàlid. A diferència del navegador (on el token és vulnerable a XSS), al mòbil s'utilitza l'emmagatzematge segur del sistema operatiu (Keychain a iOS, Keystore a Android), de manera que només la mateixa aplicació hi pot accedir (RNF-06). El registre suporta dos fluxos: obert per a estudiants —on el

backend força el rol, ignorant qualsevol manipulació del client— i amb token d'invitació per als rols privilegiats, evitant escalades de privilegis.

8.3.2 Mòdul d'alumne

L'alumne disposa d'una navegació per pestanyes (inici, mapa, reportar, llista d'incidències, punts i configuració). Es disposa d'un formulari de creació d'incidències (RF-03) que integra tres recursos nadius del dispositiu: la càmera i galeria per adjuntar fotografies, el GPS per autocompletar la ubicació amb precisió mínima de 10-20 metres (RNF-04), i la validació de camps obligatoris abans d'enviar. Un cop creada, a la pantalla d'incidències o d'inici s'ha de mostrar el report creat i quan l'estat canviï es rebran avisos gràcies a les notificacions push (vegeu 8.5.1).

8.4 Mòdul tècnic

El tècnic comparteix la mateixa aplicació que l'alumne, però amb permisos i vistes diferents: el backend filtra les dades automàticament, de manera que un tècnic només veu les incidències que té assignades.

8.4.1 Comentaris

Cada incidència disposa d'un fil de comentaris restringit a tres rols (l'autor del report, el tècnic assignat i els administradors), protegint la privadesa del flux. El sistema distingeix els comentaris lliures dels lligats a una transició d'estat (per exemple, el motiu obligatori d'un rebuig), cosa que permet renderitzar un *timeline* on canvis d'estat i missatges apareixen intercalats en ordre cronològic. Tot nou comentari dispara una notificació push als implicats, fent que el fil es comporti com una conversa en temps real.

8.4.2 Report d'incidències

L'eina principal del tècnic és la gestió de les incidències assignades. Des de la pantalla de detall pot executar les transicions que la màquina XState li autoritza: iniciar la feina i resoldre-la, deixant-la pendent de validació per l'administrador. Per garantir la traçabilitat de la resolució (RF-09), cada report distingeix tres tipus d'imatges —la inicial (de l'alumne), les de progrés i la de resolució (del tècnic)—, i cada càrrega vàlida que el rol que la puja hi estigui autoritzat.

8.5 Notificacions en temps real

El sistema cobreix dues necessitats complementàries: notificar usuaris mòbils encara que no tinguin l'app oberta, i mantenir el panell web actualitzat sense recàrregues manuals. Tots dos canals comparteixen un únic punt d'orquestració al backend, el `NotificationService`, que decideix qui ha de rebre cada esdeveniment de domini (transicions, comentaris, classificacions) amb la regla universal que **l'actor que el provoca mai s'auto-notifica**. Aquesta centralització evita duplicar lògica i facilita afegir nous canals en el futur.

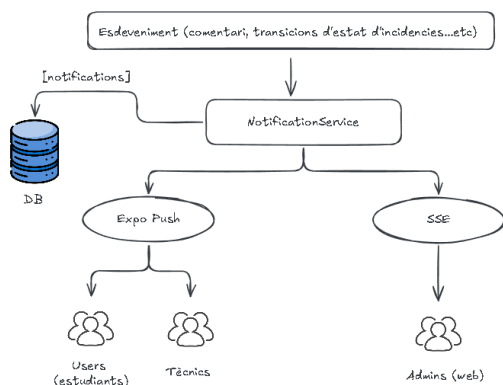


Fig. 5: Flux del sistema de notificacions.

8.5.1 Push notifications (Expo)

Per al mòbil s'ha optat per **Expo Push Service**[13], un proxy gratuït que unifica els serveis propietaris d'iOS (APNs) i Android (FCM) darrere d'una sola crida HTTP i un únic identificador. L'app demana permís en el primer inici de sessió i envia el token al backend, que l'associa a l'usuari; quan Expo informa que un dispositiu s'ha desregistrat, el token es desactiva en lloc d'esborrar-se, conservant la traça històrica.

El `NotificationService` aplica regles de *routing* per cada tipus d'esdeveniment (l'autor del report rep les transicions, el tècnic rep les assignacions, etc.). Cada notificació es persisteix també com a historial in-app, independentment que la push efímera s'hagi pogut entregar o no.

8.5.2 SSE Web notifications

Per a l'actualització en temps real del panell d'administració (RF-13) s'ha optat per **Server-Sent Events (SSE)**[14] en lloc de WebSockets, ja que el dashboard només rep esdeveniments (mai n'envia) i SSE cobreix aquest cas com a estàndard HTTP amb reconexió automàtica i una API client (`EventSource`) nativa al navegador. Com que aquesta API no admet capçaleres personalitzades per passar el JWT, s'ha implementat un patró de **tickets efímers d'un sol ús** (vàlids 60 segons i invalidats en obrir la connexió) que evita exposar el token a l'URL.

Al frontend, una sola connexió SSE oberta des del layout principal redistribueix els esdeveniments entre les pàgines (dashboard, reports, mapa, validacions i gestió d'accesos) mitjançant un bus intern, minimitzant connexions i re-renderitzats.

8.6 Classificació per IA i auto-assignació

Aquesta secció cobreix dos mecanismes d'automatització del triatge que, tot i complementar-se, són de naturalesa ben diferent. La **classificació automàtica d'incidències** (RF-14) sí que és un component d'IA: s'orquestra amb **LangGraph**[15] que és una llibreria de l'ecosistema LangChain que modela els fluxos com a grafs d'estats amb nodes condicionals. S'utilitza el model de **Google Gemini 2.5 Flash**[16], que processa text i imatge simultàniament.

En canvi, l'**auto-assignació de tasques** (RF-08) **no és un component d'IA**: és un algoritme determinista de classificació per *workCategory*, càrrega de treball i proximitat geogràfica (PostGIS), sense intervenció de cap model de llenguatge. Es presenta en aquesta secció per la seva relació amb el flux de triatge automàtic.

8.6.1 Classificació automàtica d'incidències

Quan un alumne crea una incidència, el backend dispara un procés asíncron que analitza el text i la imatge i decideix tres camps: la categoria correcta (revisant la triada per l'usuari), la prioritat real i un resum sintètic per al panell d'administració.

El flux és un graf de dos nodes: el primer crida Gemini Flash amb un prompt específic que li permet corregir la categoria de l'usuari si la imatge o el raonament la contradueix; el segon és una capa determinista que valida la sortida, garantint que cap al·lucinació del LLM corrompi la base de dades. S'ha optat per una sola crida multimodal —en lloc d'agents separats— perquè aporta més precisió que la fragmentació i alhora redueix la latència i el cost.

Per no degradar la latència en crear un report, la classificació es dispara amb un petit retard després del *commit* (perquè el mòbil acabi de pujar la imatge inicial). Un cop completada, el resultat es persisteix i s'emet un esdeveniment SSE que actualitza el panell administrador en temps real amb la informació resultant de la crida al model.

8.6.2 Autoassignació intel·ligent de tasques

L'auto-assignació viu dins la pàgina d'**Assignacions** del panell web, que ja agrupava les incidències pendents d'assignar. L'administrador pot seleccionar múltiples incidències amb checkboxes i prémer un sol botó perquè el sistema les reparteixi entre els tècnics combinant tres criteris jeràrquics: primer la **coincidència de categoria** entre la incidència i la categoria d'expertesa del tècnic; després, com a desempat, la **càrrega de feina actual** (nombre de reports actius assignats); i finalment la **proximitat geogràfica**, calculada amb la consulta espacial PostGIS `ST.Distance` sobre la columna `geography`, de manera que entre tècnics igualment carregats es prioritza el que ja té una incidència activa més a prop (agrupant els desplaçaments).

Els tècnics sense categoria queden exclosos del *pool*; si cap té la categoria del report, es marca com a no assignable. El resultat es presenta en un *"notification box"* amb dues seccions: els reports assignats amb el tècnic triat i els no assignats amb la raó.

8.7 Gamificació i sistema de recompenses

Per fomentar la co-responsabilitat de l'estudiantat (RF-16), s'ha implementat un sistema de gamificació que premia l'alumne amb punts quan una incidència que ha reportat es tanca definitivament. La quantitat de punts se sumen segons la criticitat (5, 10, 20 i 40 punts) i només la reben els estudiants. Es premia en el tancament i cada premi es registra en una taula d'auditoria amb una restricció d'unicitat per incidència, que en garanteix la idempotència dins d'una transacció atòmica. El sistema reaprofitja la infraestructura existent: el premi es dispara des de la mateixa transició `XState` i s'orquestra des del `NotificationService`, que

envia un push a l'estudiant i un esdeveniment SSE que refresca el rànquing en temps real. El rànquing s'exposa tant al panell web com a l'app mòbil.

9 RESULTATS

El resultat és **CampusFix**, una plataforma multiplataforma funcional articulada en dues interfícies (app mòbil per a estudiants i tècnics, panell web per a administradors) sobre un únic backend que centralitza la lògica de negoci. Cobreix el cicle complet d'una incidència: report mòbil amb geolocalització i foto, classificació automàtica amb IA, assignació al tècnic, validació fotogràfica i tancament amb premi de gamificació.

S'han assolit tots els requisits funcionals (RF-01 a RF-17) excepte l'autenticació real amb proveïdors externs (RF-02), reservada com a treball futur (sincronització de comptes reals d'Outlook dels estudiants). La lògica de negoci crítica (transicions d'estat RF-04/RF-05 i control d'accés RBAC RNF-05) es concentra íntegrament al backend, que n'és l'única font de veritat. Els requisits de rendiment (RNF-03, RNF-07) i la precisió de la IA (RF-14/RF-15) s'han mesurat empíricament amb bancs de proves reproduïbles. Les xifres són **indicatives, no constants exactes**, ja que depenen de factors externs com el RTT cap a Supabase, la càrrega del núvol i, en la IA, del caràcter no determinista del model.

9.1 Validacions

El rendiment de les consultes geoespacial i del dashboard s'ha mesurat a nivell HTTP extrem-a-extrem: el cicle complet de petició (Express, autenticació, BD i serialització), que és el més proper a la latència de l'usuari final (Taula 1). El pitjor cas és de 129 ms (p95), molt per sota dels 500 ms de RNF-03. Tanmateix, els temps en producció podrien variar lleugerament degut a factors externs com la càrrega de la xarxa.

TAULA 1: LATÈNCIA HTTP - 100 ITERACIONS.

Consulta	p50	p95	p99
Carregar tots els punts del mapa	93,1	96,8	102,7
Punts del mapa (filtre últims 30 dies)	85,2	87,5	89,3
Dades del mapa de calor	87,8	90,7	98,9
Mètriques del dashboard	127,2	129,1	130,5

La latència del mecanisme SSE (RNF-07) és d'1 ms (p95), molt per sota dels 200 ms; cobreix el cost intern de la propagació, no el RTT de xarxa real ni el render del navegador, que depenen de factors externs no controlables.

Sobre 10 casos etiquetats manualment, el model va encertar la categoria en tots (100 %) i la prioritat en el 60 % de forma exacta i en el 100 % amb un marge de ± 1 nivell (Taula 2); a més, va corregir les 2 categories que hi havia errònies en els exemples de prova.

TAULA 2: PRECISIÓ CLASSIFICACIÓ - 10 CASOS

Mètrica	Resultat
Exactitud de categoria	10/10 (100 %)
Correcció de categoria errònia	2/2 (100 %)
Prioritat (encert exacte)	6/10 (60 %)
Prioritat (encert ± 1 nivell)	10/10 (100 %)

Les discrepàncies de prioritat són sempre d'un sol nivell i sobre fronteres subjectives (on la frontera HIGH/CRITICAL és la més difusa). Els resums automàtics (RF-15) van ser sintètics i informatius. L'avaluació s'ha realitzat amb una mostra reduïda i un model no determinista, de manera que els valors són **indicatius, no concloents**, tot i això aquestes proves ens indiquen que el model treballa bé amb la configuració i entorn actual.

A tall d'exemple, un dels casos del conjunt d'avaluació (Fig. 6). Per a aquesta entrada el model va respondre prioritat HIGH en lloc de la CRITICAL esperada: l'única mena de discrepància observada, sempre d'un sol nivell i sobre la frontera subjectiva HIGH/CRITICAL.

```
{
  "id": "C08",
  "title": "Branca a punt de caure",
  "description": "Una branca grossa d'un pi del passeig està mig trencada i penja sobre la zona de pas dels vianants.",
  "userCategory": "GREEN_AREAS",
  "expectedCategory": "GREEN_AREAS",
  "expectedPriority": "CRITICAL"
}
```

Fig. 6: Exemple de cas del conjunt d'avaluació de la IA

9.2 Tests

El projecte disposa d'una bateria de **proves automàtiques**, executables abans de qualsevol desplegament com a porta de qualitat: si fallen, el canvi no es publica.

Al **backend**, 54 **proves unitàries** (*Vitest*) validen la lògica de negoci crítica amb el client de dades *mockejat*: la màquina d'estats (tota la matriu estat-esdeveniment-rol de les transicions, RF-04/RF-05), el control d'accés per rols (RBAC, RNF-05), la concessió de punts de gamificació i l'auto-assignació de tasques. A més, un *smoke test* de 18 comprovacions exercita l'API real d'extrem a extrem (autenticació, permisos 401/403, visibilitat forçada al servidor i transicions il·legals); aquesta prova va permetre detectar i corregir un codi d'estat HTTP erroni en el registre.

Al **frontend**, 15 **proves de renderitzat** comproven que cada interfície carrega el contingut adequat segons la petició: al **panell web** (*Vitest* i *Testing Library*), l'inici de sessió i la protecció de rutes d'administrador; a l'**app mòbil** (*jest-expo* i *Testing Library*), els fluxos principals de cada rol: report d'incidència junt amb consulta de punts per part de l'alumne i iniciar una tasca junt amb la resolució d'una incidència per part del tècnic.

10 CONCLUSIONS

L'objectiu d'aquest TFG era modernitzar la gestió del manteniment del campus, substituint un model reactiu i opac (formularis, correus, trucades) per una plataforma àgil i transparent que connectés estudiants, tècnics i administradors. Després dels vuit sprints de desenvolupament, es conclou que aquest objectiu s'ha assolit amb un producte funcional i cohesionat.

El balanç respecte als objectius inicials és positiu: el report mòbil geolocalitzat, la gamificació, l'assignació basada en expertesa i proximitat, el dashboard SIG i la classificació

amb IA cobreixen els objectius plantejats. El desenvolupament ha validat les decisions de fons —centralitzar la lògica al backend, formalitzar el cicle de vida amb una màquina d'estats i reaprofitar els canals d'infraestructura—, que han permès afegir noves funcionalitats amb un cost marginal mínim.

El projecte deixa diverses línies d'extensió. A curt termini, integrar l'**autenticació amb el sistema institucional de la UAB** (RF-02), de manera que estudiants i personal puguin accedir directament amb el seu compte d'Outlook de l'autònoma, eliminant el registre manual. A llarg termini, la via amb més recorregut és una **implementació més extensa de la IA per automatitzar i simplificar el funcionament de la plataforma**: l'arquitectura amb agents està pensada per escalar gradualment, acostant la plataforma —sempre sota control humà— a una gestió gairebé autònoma. Algunes línies d'evolució són: report automàtic des d'una sola foto (l'agent s'encarrega d'omplir els camps i de preguntar a l'usuari en cas de dubte), agents guia per a l'administrador o fins i tot agents d'observabilitat per analitzar patrons i anticipar esdeveniments.

Una tercera línia, tècnica, és l'**escalabilitat de la visualització geoespacial**: l'agrupació de markers, ara resolta al client amb Leaflet, obliga a transferir tots els punts i es degrada amb volums grans. La solució és traslladar l'agregació al servidor amb PostGIS —un filtre *bounding-box* sobre l'índex GiST combinat amb *clustering* per nivell de detall segons el zoom (ST_SnapToGrid o ST_ClusterDBSCAN)—, de manera que el volum transferit depengui del que es veu i no del total.

ÚS DE LA IA GENERATIVA

Durant el desenvolupament s'han emprat eines d'intel·ligència artificial generativa (assistents de codi) com a suport, sempre sota la supervisió i validació de l'autor. La seva aportació s'ha concentrat en tasques de **frontend** i de naturalesa repetitiva com ara components d'interfície o classes d'estil. S'ha destinat l'ús més gros a feina d'esforç manual elevat però amb poc valor intel·lectual, mecànica i **secundària** respecte al nucli del projecte.

El veritable valor del TFG, on l'autoria és íntegrament pròpia, resideix en les **decisions de disseny, la lògica de negoci i l'arquitectura**: la màquina d'estats, el modelatge de dades, l'estratègia de seguretat, el flux de notifikacions i la integració dels agents d'IA. La IA hi ha actuat com a eina d'implementació de tasques accessòries, mai com a font de criteri, sense substituir la comprensió ni la responsabilitat sobre el funcionament global de la plataforma.

REFERÈNCIES

- [1] ¿Qué es el Quality Assurance (QA)? [Online]. Disponible en: <https://www.mtp.es/blog/quality-assurance/que-es-el-quality-assurance-y-por-que-es-esencial-en-el-desarrollo-de-software/> [27 de febrer de 2026]
- [2] Cómo funciona Socket IO. [Online]. Disponible en: <https://keepcoding.io/blog/que-es-y-como-funciona-socket-io/> [28 de febrer de 2026]
- [3] Documentació Expo. [Online]. Disponible en: <https://docs.expo.dev/> [02 de març de 2026]
- [4] Metodología Scrum. [Online]. Disponible en: <https://www.atlassian.com/es/agile/scrum> [03 de març de 2026]
- [5] Normativa protecció de dades. [Online]. Disponible en: <https://www.boe.es/buscar/doc.php?id=DOUE-L-2016-80807> [03 de març de 2026]
- [6] Langchain-OpenAI Docs and Models. [Online]. Disponible en: <https://reference.langchain.com/python/langchain-openai> [04 de març de 2026]
- [7] Prisma Schema [Online]. Disponible en: <https://www.prisma.io/docs/orm/prisma-schema/overview> [20 de març de 2026]
- [8] Prisma Postgres [Online]. Disponible en: <https://www.prisma.io/docs/postgres> [21 de març de 2026]
- [9] Supabase Prisma [Online]. Disponible en: <https://supabase.com/docs/guides/database/prisma> [21 de març de 2026]
- [10] Find It, Fix It [Online]. Disponible en: <https://www.seattle.gov/customer-service-bureau/find-it-fix-it-mobile-app> [22 de març de 2026]
- [11] Bolt CityFix [Online]. Disponible en: <https://bolt.eu/en/blog/bolt-cityfix-reporting-tool/> [22 de març de 2026]
- [12] Documentació React [Online]. Disponible en: <https://react.dev/learn> [28 de març de 2026]
- [13] Expo Push Notifications Overview. [Online]. Disponible en: <https://docs.expo.dev/push-notifications/overview/> [4 de abril de 2026]
- [14] Mozilla Developer Network — Using server-sent events. [Online]. Disponible en: https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events [4 de abril de 2026]
- [15] LangGraph.js — Build resilient language agents as graphs. [Online]. Disponible en: <https://langchain-ai.github.io/langgraphjs/> [2 de maig de 2026]
- [16] Google Gemini API Documentation. [Online]. Disponible en: <https://ai.google.dev/gemini-api/docs> [15 de maig de 2026]
- [17] Supabase Storage Documentation. [Online]. Disponible en: <https://supabase.com/docs/guides/storage> [15 de maig de 2026]

APÈNDIX

A.1 Arquitectura de la Base de Dades

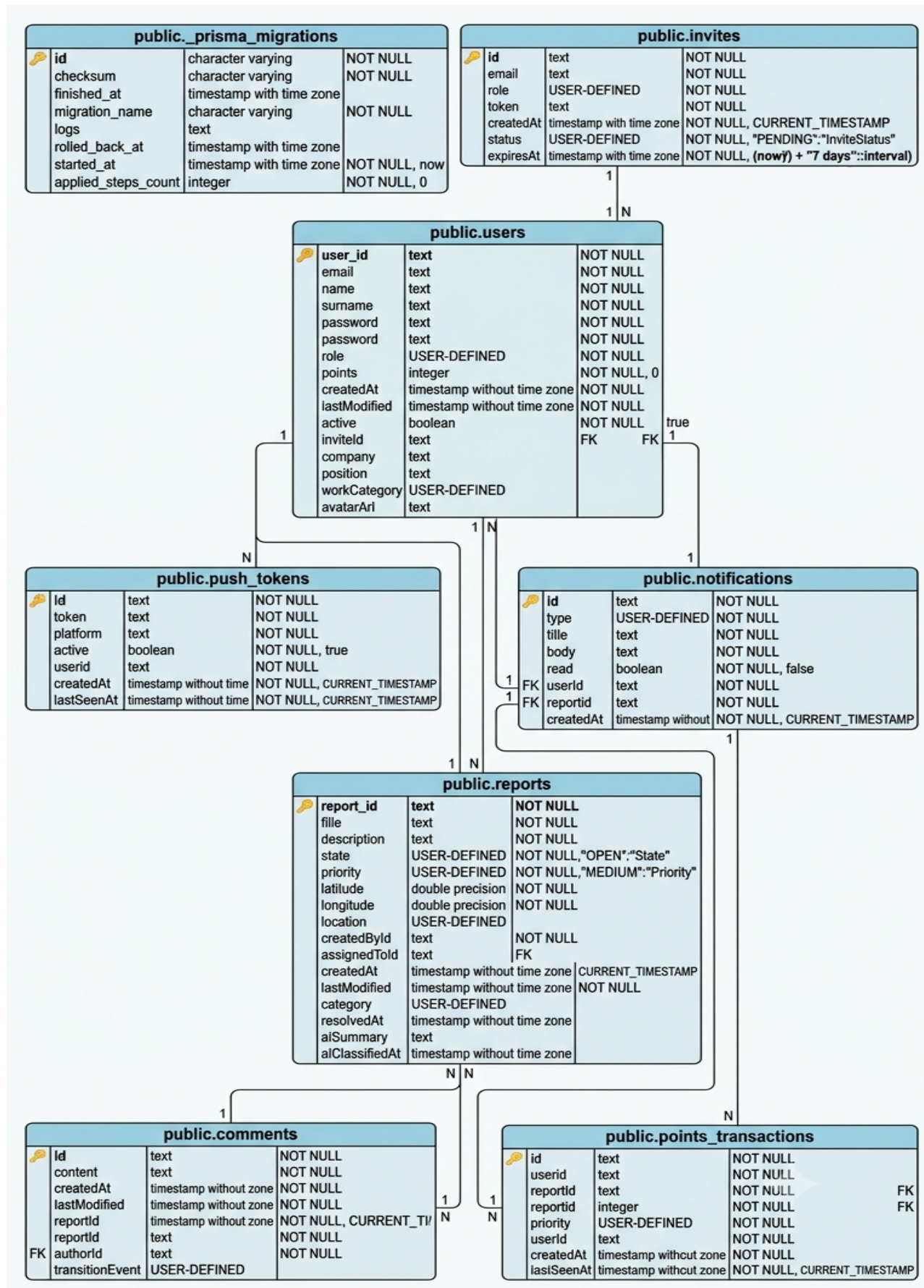


Fig. 7: Arquitectura de la base de datos

A.2 Flux General de Peticions

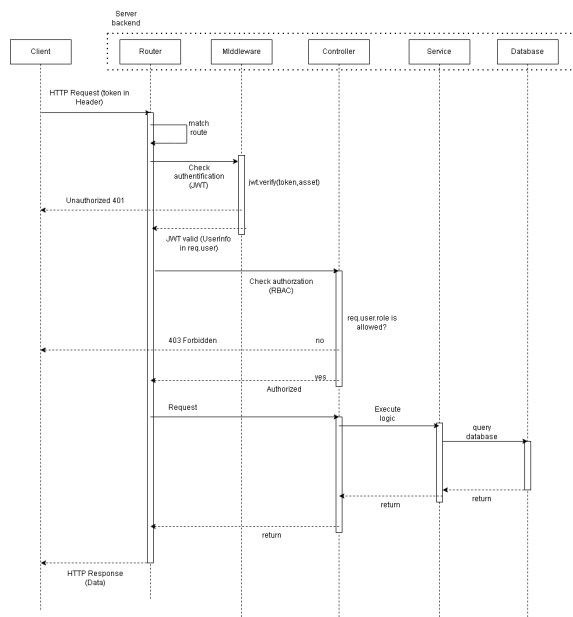


Fig. 8: Flux de peticions HTTP

A.3 Flux màquina XState

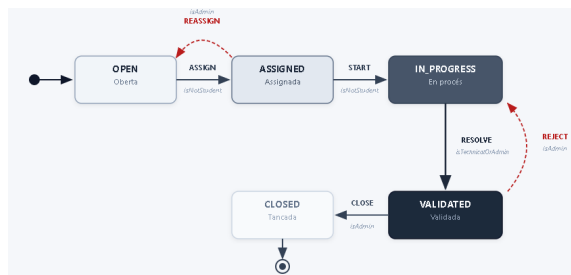


Fig. 9: Transició d'estat validation

A.4 Inici de sessió i registre d'usuaris admin

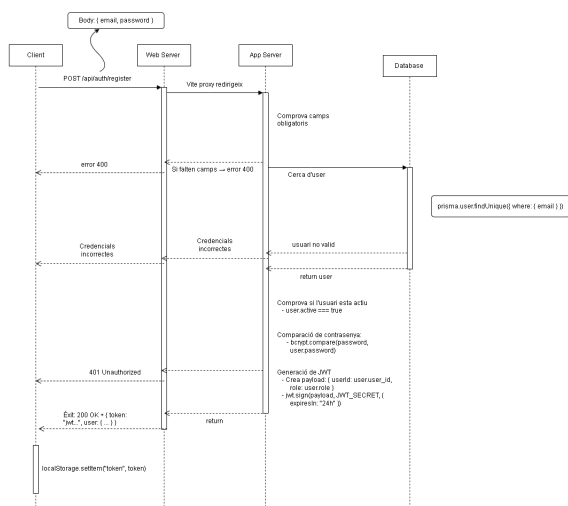


Fig. 10: Diagrama de seqüència inici de sessió

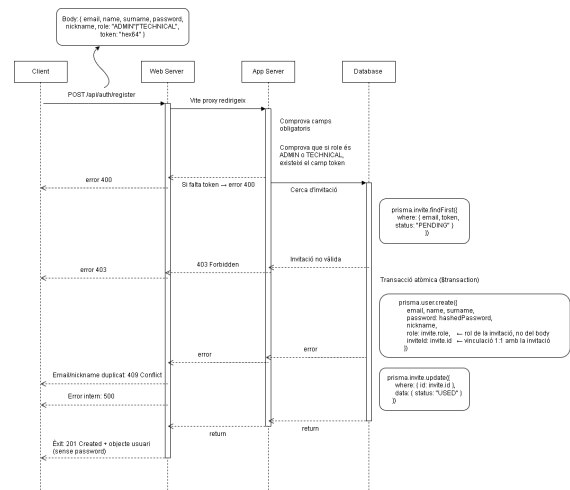


Fig. 11: Diagrama de seqüència registre usuaris admin

A.5 Dashboard

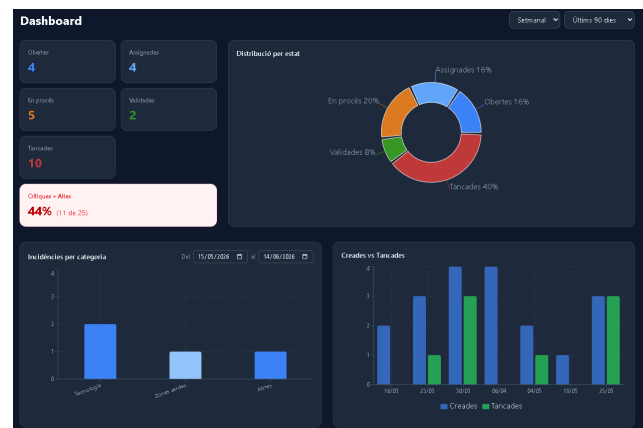


Fig. 12: Dashboard

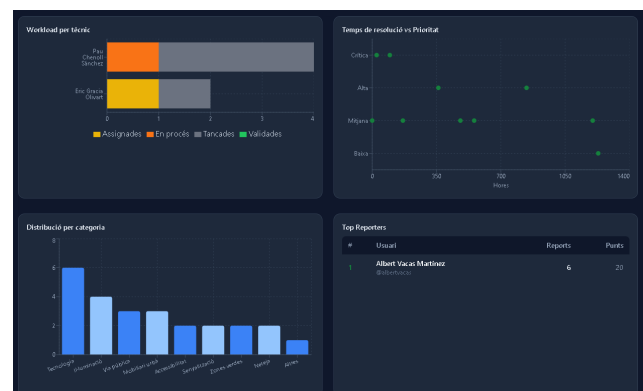


Fig. 13: Dashboard

A.6 Visualització mapa d'incidències

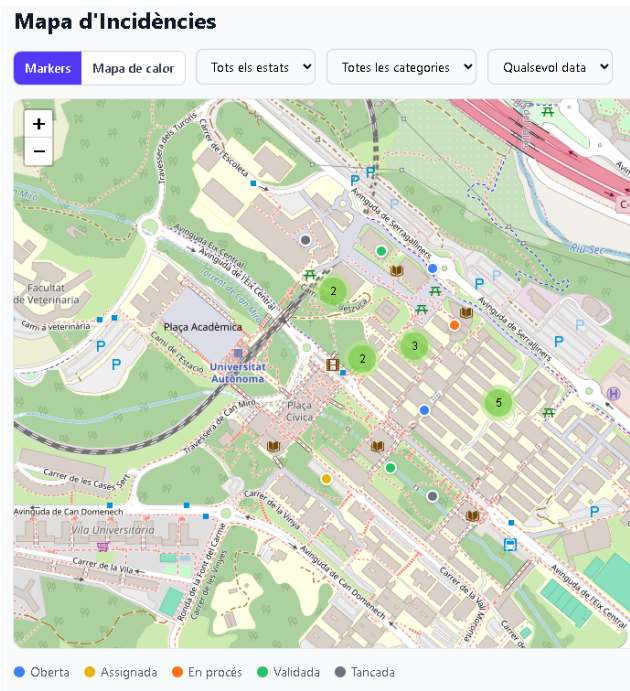


Fig. 14: Mapa d'incidències per punts

A.8 Visualització app



Fig. 16: Pantalla Home

A.7 Gamificació

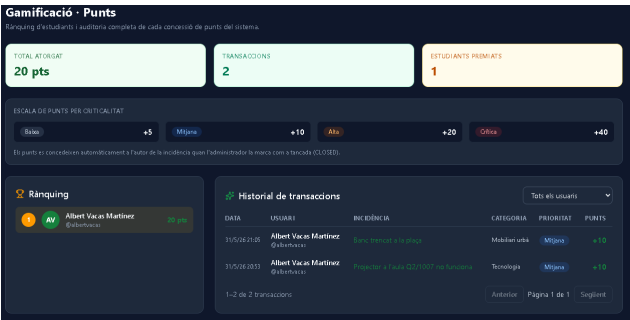


Fig. 15: Pantalla ranking punts

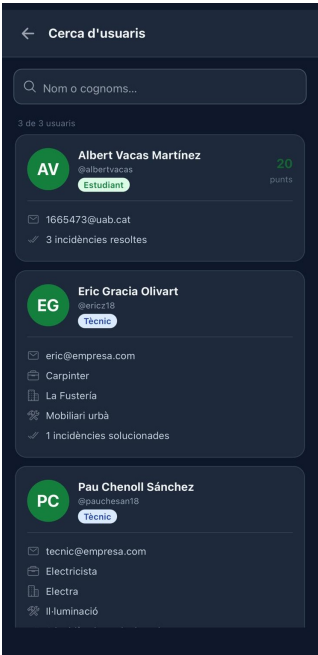


Fig. 17: Cerca d'usuaris

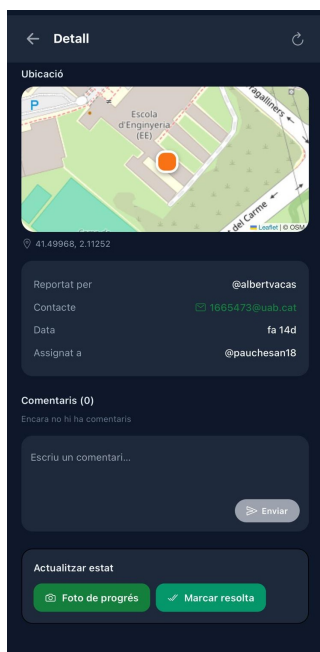


Fig. 18: Vista de detall d'un report des del rol tècnic, mostrant els botons d'acció disponibles

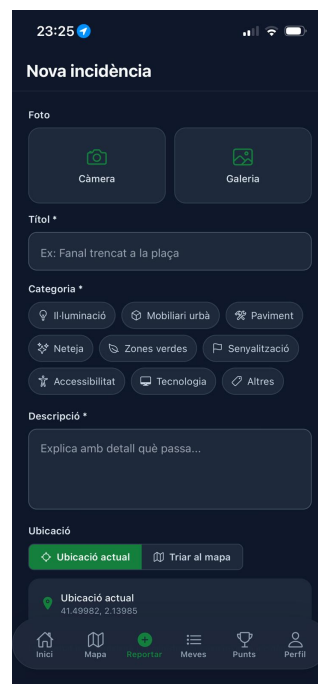


Fig. 20: Formulari de creació d'incidència i pantalla de detall des del rol d'estudiant

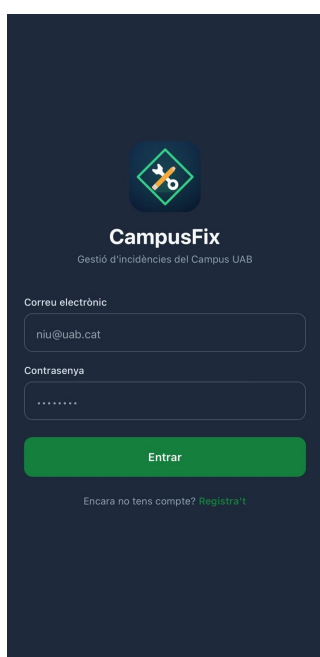


Fig. 19: Pantalles de login i registre de l'aplicació mòbil