

VeciApp

Plataforma mòbil de gestió integral de comunitats de veïns

Memòria Tècnica del Projecte

Projecte de Desenvolupament

CFGS Desenvolupament d'Aplicacions Multiplataforma

Autors: Edgar Garcia i Santiago Cardosa

Centre: Institut Puig Castellar, Santa Coloma de Gramenet

Curs acadèmic: 2025-2026

Llicència: [CC BY-NC-ND 3.0 ES](#)

Resum del projecte

VeciApp és una aplicació mòbil nativa per a iOS i Android dissenyada per digitalitzar i centralitzar la gestió integral de comunitats de veïns. El projecte aborda una problemàtica real i estesa: la majoria de comunitats de propietaris a Espanya segueixen gestionant els seus assumptes interns mitjançant canals informals i dispersos com grups de WhatsApp, correus electrònics i trucades telefòniques, cosa que provoca pèrdua d'informació, manca de traçabilitat, conflictes de comunicació i una sobrecàrrega administrativa per al president de la comunitat.

L'aplicació ofereix vuit mòduls completament funcionals i interconnectats: (1) xat en temps real amb grups tipificats i missatges privats, (2) gestió econòmica amb quotes automàtiques, repartiment proporcional de gastos i pressupostos per projecte, (3) sistema de votacions digitals amb creació, participació i tancament, (4) registre d'incidències amb prioritats i seguiment d'estats, (5) repositori de documents categoritzats, (6) directori de contactes amb xat directe, (7) contactes d'emergència amb trucada directa, i (8) calendari d'esdeveniments comunitaris.

Des del punt de vista tècnic, el projecte suma més de 6.900 línies de codi TypeScript repartides en 17 fitxers principals. La capa de presentació utilitza React Native 0.81.5 amb Expo 54 i Expo Router per a la navegació basada en fitxers. El backend es recolza íntegrament en Supabase, que proporciona una base de dades PostgreSQL 15 amb 18 taules, autenticació per email/contrasenya, subscripcions Realtime mitjançant WebSockets, i emmagatzematge de fitxers. TypeScript garanteix la correcció de tipus en tot el projecte.

El sistema implementa un model de permisos basat en tres rols: President (gestió completa), Propietari (accés total excepte administració) i Treballador (accés limitat a comunicació i incidències). Les dades estan completament aïllades per comunitat mitjançant el camp `comunidad_id` present a totes les taules, garantint que cada comunitat és un entorn completament independent.

Com a conclusions, tots els objectius específics plantejats inicialment han estat assolits satisfactòriament. L'aplicació està en funcionament, ha estat provada en dispositius físics iOS i Android, i dona resposta a una necessitat real de milers de comunitats. Les principals línies de futur inclouen l'activació de Row Level Security per millorar la seguretat, la implementació de notificacions push, i el desenvolupament d'una versió web.

Paraules clau: comunitat de veïns, React Native, Expo, TypeScript, Supabase, PostgreSQL, Realtime, WebSockets, gestió econòmica, votacions digitals, sistema de rols, aplicació mòbil multiplataforma

Abstract

VeciApp is a native mobile application for iOS and Android designed to digitize and centralize the comprehensive management of residential communities. The project addresses a real and widespread problem: most homeowners' communities in Spain still manage their internal affairs through informal and dispersed channels such as WhatsApp groups, emails, and phone calls, which causes information loss, lack of traceability, communication conflicts, and administrative overload for the community president.

The application provides eight fully functional and interconnected modules: real-time chat with typed groups and private messages, financial management with automatic dues, proportional expense splitting and project budgets, digital voting system with creation, participation and closure, incident registry with priorities and status tracking, categorized document repository, contact directory with direct chat, emergency contacts with direct calling, and a shared community events calendar.

From a technical perspective, the project spans over 6,900 lines of TypeScript code across 17 main files. The presentation layer uses React Native 0.81.5 with Expo 54 and Expo Router for file-based navigation. The backend relies entirely on Supabase, which provides a PostgreSQL 15 database with 18 tables, email/password authentication, Realtime subscriptions via WebSockets, and file storage. TypeScript ensures type correctness throughout the project.

Keywords: residential community, React Native, Expo, TypeScript, Supabase, PostgreSQL, Realtime, WebSockets, financial management, digital voting, role-based access, cross-platform mobile application

Resum del projecte	2
Abstract.....	3
1. Introducció	7
1.1 Context.....	7
1.2 Justificació.....	7
1.3 Objectius del projecte.....	8
1.3.1 Objectiu general.....	8
1.3.2 Objectius específics.....	8
1.4 Metodologia de treball.....	9
Fases de desenvolupament.....	9
Fase 1 — Planificació i disseny.....	9
Fase 2 — Configuració de l'entorn.....	9
Fase 3 — Autenticació i onboarding.....	9
Fase 4 — Mòduls principals.....	9
Fase 5 — Política de rols i refinament.....	9
1.5 Estudi econòmic i pressupostari.....	10
Costos de desenvolupament.....	10
Escalabilitat de costos.....	10
2. Anàlisi de requisits	12
2.1 Requisits funcionals.....	12
Autenticació i gestió d'usuaris.....	12
Comunicació.....	12
Gestió econòmica.....	13
Votacions, incidències, documents i més.....	13
2.2 Requisits no funcionals.....	14
3. Tecnologies i arquitectura	15
3.1 Comparativa de tecnologies valorades.....	15
Framework mòbil.....	15
Backend / BaaS.....	15
3.2 Stack tecnològic escollit.....	15
3.3 Arquitectura del sistema.....	16
Estructura de directoris del projecte.....	16
4. Base de dades	18
4.1 Taules del sistema.....	18
comunidad.....	18
usuario.....	18
chat i chat_miembro.....	19
pago, movimiento i gasto_reparto.....	20
encuesta, encuesta_opcion i encuesta_respuesta.....	21
incidencia.....	21

evento.....	21
5. Funcionalitats implementades.....	22
5.1 Autenticació i onboarding.....	22
Flux de registre.....	22
Flux de login.....	22
Flux d'onboarding.....	22
5.2 Xat en temps real.....	22
Arquitectura tècnica.....	22
Llista de xats.....	23
Crear grup (president).....	23
Conversa individual.....	23
Gestionar membres (president).....	23
Silenciar.....	23
5.3 Gestió econòmica.....	23
Pestanya Resum.....	23
Pestanya Cuotes.....	24
Pestanya Pressupostos.....	24
Pestanya Moviments.....	24
Modal crear gasto.....	24
5.4 Votacions.....	24
Crear enquesta (president).....	24
Votar.....	24
Tancar votació.....	25
5.5 Incidències.....	25
5.6 Documents.....	25
5.7 Contactes i emergències.....	25
5.8 Calendari d'esdeveniments.....	26
6. Sistema de permisos per rols.....	27
6.1 Definició de rols.....	27
6.2 Taula de permisos detallada.....	27
6.3 Implementació tècnica dels rols.....	28
BottomTabBar dinàmica.....	28
Guards de pantalla.....	28
Exclusió en repartiment.....	28
Renderitzat condicional.....	28
7. Disseny visual i experiència d'usuari.....	29
7.1 Principis de disseny.....	29
7.2 Paleta de colors.....	29
7.3 Tipografia i espaiat.....	30
7.4 Gestió del teclat.....	30
8. Conclusions.....	31

8.1	Conclusions generals del projecte.....	31
8.2	Consecució dels objectius.....	31
8.3	Valoració de la metodologia.....	32
9.	Visió de futur.....	33
9.1	Notificacions push.....	33
9.2	Publicació a les botigues.....	33
9.3	Funcionalitats addicionals.....	33
10.	Glossari.....	34
	Tecnologies i conceptes tècnics.....	34
12.	Bibliografia.....	36

1. Introducció

Les comunitats de propietaris són una forma d'organització social que afecta directament la qualitat de vida de milions de persones. A Espanya, més de 10 milions de persones viuen en edificis de propietaris gestionats per un president i, en molts casos, per un administrador de finques. La gestió d'aquestes comunitats implica coordinar des de reparacions urgents fins a votacions sobre obres, passant per la recaptació de quotes mensuals, la comunicació d'incidències o l'arxiu de documents legals.

Malgrat la importància d'aquesta gestió, la immensa majoria de comunitats operen amb eines no dissenyades per a aquest propòsit. WhatsApp s'utilitza per a comunicació urgent, el correu electrònic per a actes i documents, i les reunions presencials per a decisions col·lectives. Aquesta dispersió genera problemàtiques concretes: missatges importants es perden entre converses, els veïns no saben què s'ha decidit, la informació econòmica no és accessible per a tots, i el president dedica hores excessives a tasques administratives.

VeciApp neix com a resposta directa a aquesta necessitat. És una aplicació mòbil nativa, gratuïta, i dissenyada específicament per a comunitats de veïns, que centralitza en una sola plataforma tots els fluxos d'informació i gestió que una comunitat necessita.

1.1 Context

El mercat actual d'aplicacions per a comunitats de veïns està dominat per plataformes web amb models de subscripció mensual (Fincatech, Comunidadesplus, iProperty). Aquestes solucions presenten barreres d'entrada significatives: costos de 20-50€/mes, requeriment de mediació d'un administrador de finques, interfície complexa poc adequada per a usuaris no tècnics, i manca d'aplicació mòbil nativa de qualitat.

En paral·lel, l'ecosistema tecnològic actual permet construir aplicacions mòbils completes amb costos operatius mínims. React Native permet un única base de codi per iOS i Android. Supabase ofereix un backend complet gratuït fins a 500MB de base de dades. Expo simplifica el cicle de desenvolupament i proves. Aquest context tecnològic fa viable la creació d'una alternativa lliure i mòbil a les plataformes existents.

1.2 Justificació

La justificació del projecte s'articula en quatre dimensions:

- **Impacte social:** millorar la convivència en comunitats té un efecte directe en el benestar dels residents. La comunicació transparent i la gestió àgil de conflictes redueix friccions i promou la cohesió.
- **Oportunitat de mercat:** no existeix cap solució mòbil nativa gratuïta i completa per a comunitats de veïns en el mercat espanyol.
- **Aprenentatge tècnic:** el projecte permet aplicar i aprofundir en tecnologies de primera línia: React Native, TypeScript, PostgreSQL, WebSockets i arquitectures de SaaS.
- **Viabilitat econòmica:** amb menys de 134€ d'inversió anual, s'obté una solució completament operativa que podria escalar-se a milers de comunitats.

1.3 Objectius del projecte

1.3.1 Objectiu general

Desenvolupar una aplicació mòbil multiplataforma (iOS i Android), completament funcional i connectada a una base de dades real, que permeti a qualsevol comunitat de veïns gestionar de manera digital, transparent i eficient tots els aspectes de la seva organització interna.

1.3.2 Objectius específics

OE01: Implementar un sistema d'autenticació segur basat en Supabase Auth que permeti registre i inici de sessió per email i contrasenya.

OE02: Dissenyar i implementar un sistema d'invitació per codi (6 caràcters alfanumèrics) que permeti crear noves comunitats o unir-se a existents sense intervenció tècnica.

OE03: Construir un mòdul de xat en temps real amb subscripcions WebSocket (Supabase Realtime), suport per grups tipificats (edifici, serveis, urgent) i missatges privats entre veïns.

OE04: Implementar un mòdul econòmic amb quotes mensuals per veï, gastos de comunitat, gastos repartits proporcionals excloent treballadors, pressupostos per projecte, i historial de moviments.

OE05: Desenvolupar un sistema de votacions digital amb creació d'enquestes, votació individual garantida (constraint UNIQUE), visualització de resultats amb percentatges i tancament controlat.

OE06: Crear un mòdul de gestió d'incidències amb tres nivells de prioritat, quatre estats possibles i seguiment cronològic.

OE07: Integrar un repositori de documents amb pujada a Supabase Storage, categorització predefinida i accés per descarrega.

OE08: Implementar un directori de contactes amb accés directe a xat privat i un sistema de contactes d'emergència amb trucada directa.

OE09: Construir un calendari d'esdeveniments comunitaris amb creació, visualització mensual i tipificació per color.

OE10: Dissenyar i implementar un sistema de permisos basat en tres rols (President, Propietari, Treballador) que controli l'accés a funcionalitats a nivell de UI i de query.

OE11: Garantir l'aïllament de dades entre comunitats mitjançant filtratge per `comunidad_id` en totes les consultes.

OE12: Assegurar la compatibilitat amb iOS 13+ i Android 8+ amb una interfície visual unificada i responsiva.

1.4 Metodologia de treball

El projecte ha seguit una metodologia de desenvolupament àgil adaptat, amb iteracions curtes i validació constant en dispositiu físic. No s'ha fet servir cap eina de gestió de projectes formal (Jira, Trello) sinó una planificació per tasques setmanes gestionada directament entre els dos membres de l'equip.

Fases de desenvolupament

Fase 1 — Planificació i disseny

Definició dels requisits funcionals i no funcionals mitjançant sessions de brainstorming.

Investigació i avaluació de tecnologies: React Native vs Flutter, Supabase vs Firebase vs PocketBase.

Disseny de l'esquema de base de dades amb totes les taules i relacions.

Creació de wireframes de baix nivell per a les pantalles principals.

Fase 2 — Configuració de l'entorn

Creació del projecte Expo amb TypeScript i configuració d'Expo Router.

Connexió amb el projecte Supabase i creació de les primeres taules.

Configuració de les migracions de base de dades.

Implementació del sistema de navegació i estructura de fitxers.

Fase 3 — Autenticació i onboarding

Pantalles de login i registre amb validació de formularis.

Integració de Supabase Auth i gestió de sessions persistents.

Flux de creació i unió a comunitats amb codis d'invitació.

Fase 4 — Mòduls principals

- Xat en temps real amb Supabase Realtime.
- Mòdul econòmic complet.
- Votacions i incidències.
- Documents, contactes i calendari.

Fase 5 — Política de rols i refinament

- Implementació del sistema de rols i permisos.
- Unificació de la paleta de colors i estils visuals.
- Correcció de bugs: `KeyboardAvoidingView`, filtratge per comunitat, gestió d'errors.

- Proves intensives en dispositius físics iOS i Android.

1.5 Estudi econòmic i pressupostari

Un dels objectius implícits del projecte és demostrar que és possible crear una aplicació mòbil professional amb costos operatius mínims gràcies a l'ecosistema open-source i als plans gratuïts dels serveis cloud moderns.

Costos de desenvolupament

Concepte	Cost mensual	Cost anual	Observacions
Supabase Free Tier	0 €	0 €	500MB BD, 1GB Storage, 50.000 usuaris Auth
Expo (React Native)	0 €	0 €	Open source, MIT License
TypeScript	0 €	0 €	Open source, Apache 2.0
VS Code + extensions	0 €	0 €	Gratuït per a ús personal
GitHub (repositori)	0 €	0 €	Pla gratuït il·limitat
Compte Apple Developer	8,25 €	99 €	Requerit per publicar a App Store
Compte Google Play	—	25 €	Pagament única vegada
Domini web (opcional)	0,83 €	10 €	Per a landing page del producte
Vercel (hosting)	0 €	0 €	Pla gratuït per a landing page
TOTAL (primer any)		134 €	
TOTAL (anys posteriors)		109 €/any	Sense el pagament Google Play

Comparativament, les alternatives comercials existents com Fincatech o Comunidadesplus cobren entre 240€ i 600€ anuals, i requereixen a més les despeses de l'administrador de finques. VeciApp ofereix funcionalitats equivalents o superiors a un cost 3-5 vegades inferior.

Escalabilitat de costos

El pla gratuït de Supabase suporta fins a 500MB de base de dades i 1GB d'Storage. Per una comunitat típica de 30 veïns amb ús actiu durant 2 anys, s'estima un consum aproximat de 50-80MB de base de dades i 200-400MB d'Storage (documents PDF). Per tant, el pla gratuït és suficient per a la majoria de comunitats durant anys. En cas de creixement, el pla Pro de Supabase costa 25€/mes i suporta 8GB de base de dades i 100GB d'Storage.

2. Anàlisi de requisits

L'anàlisi de requisits es va realitzar mitjançant sessions de definició entre els dos membres de l'equip, complementades per un estudi d'aplicacions similars i entrevistes informals amb presidents de comunitats. Es van identificar 30 requisits funcionals i 8 de no funcionals.

2.1 Requisits funcionals

Autenticació i gestió d'usuaris

ID	Nom	Descripció	Estat
RF0 1	Registre d'usuari	L'usuari introdueix nom, email i contrasenya. El sistema valida i crea compte via Supabase Auth	✓ Impl.
RF0 2	Inici de sessió	Login amb email i contrasenya. Sessió persistent amb AsyncStorage	✓ Impl.
RF0 3	Crear comunitat	Usuari crea comunitat amb nom i adreça. Sistema genera codi de 6 caràcters única	✓ Impl.
RF0 4	Unir-se a comunitat	L'usuari introdueix codi de 6 caràcters. Sistema valida i assigna rol PROPIETARIO	✓ Impl.
RF0 5	Editar perfil	L'usuari pot modificar nom, cognoms i telèfon des d'Ajustos	✓ Impl.
RF0 6	Canviar contrasenya	Canvi de contrasenya autenticat des d'Ajustos	✓ Impl.
RF0 7	Eliminar compte	Eliminació de compte amb confirmació per text	✓ Impl.

Comunicació

ID	Nom	Descripció	Estat
RF0 8	Enviar missatge	Enviar text en grups i xats privats. INSERT a taula mensaje, trigger actualitza ultimo_mensaje_at	✓ Impl.
RF0 9	Rebre en temps real	Subscripció Supabase Realtime a INSERT en taula mensaje filtrant per chat_id	✓ Impl.
RF1 0	Crear grup (president)	Modal amb nom, tipus (edifici/servei/urgent), selecció múltiple de membres	✓ Impl.
RF1 1	Silenciar grup	Opcions: per sempre, 1 dia, 1 setmana. S'actualitza chat_miembro.silenciado_hasta	✓ Impl.
RF1 2	Veure participants	Modal amb llista completa de membres del grup amb els seus rols	✓ Impl.

RF1 3	Sortir del grup	Elimina la fila corresponent de chat_miembro. Confirmació prèvia	✓ Impl.
RF1 4	Gestionar membres (pres.)	Afegir veïns no membres o eliminar membres existents. Confirmació en eliminació	✓ Impl.

Gestió econòmica

ID	Nom	Descripció	Estat
RF1 5	Quotes mensuals	Gestió de quotes per veï i mes. Camps mes (1-12), anio, importe, estado	✓ Impl.
RF1 6	Marcar quota pagada	President marca quota com pagada: UPDATE pago + INSERT movimiento (ingres)	✓ Impl.
RF1 7	Crear gasto comunitat	President crea gasto global: INSERT movimiento tipo=gasto, tipo_gasto=comunidad	✓ Impl.
RF1 8	Crear gasto repartit	President selecciona veïns (excloent TRABAJADOR), calcula import/persona, INSERT gasto_reparto per cada un	✓ Impl.
RF1 9	Marcar reparto pagat	Usuari marca el seu reparto com pagat: UPDATE gasto_reparto.estado = PAGADO	✓ Impl.
RF2 0	Pressupostos	Crear i visualitzar pressupostos per projecte amb barra de progrés gastado/total	✓ Impl.
RF2 1	Historial moviments	Llista cronològica inversa de tots els ingressos i gastos de la comunitat	✓ Impl.

Votacions, incidències, documents i més

ID	Nom	Descripció	Estat
RF2 2	Crear votació (pres.)	Modal amb pregunta i 2-6 opcions dinàmiques. INSERT encuesta + encuesta_opcion	✓ Impl.
RF2 3	Votar	Usuari vota una opció. Constraint UNIQUE(usuario_id, encuesta_id) prevé vots duplicats	✓ Impl.
RF2 4	Veure resultats	Barres de progrés amb percentatges calculats per COUNT d'encuesta_respuesta agrupat per opció	✓ Impl.
RF2 5	Tancar votació (pres.)	UPDATE encuesta SET activa=false, fecha_cierre=now(). Opció guanyadora destacada en verd	✓ Impl.
RF2 6	Reportar incidència	Formulari amb títol, descripció, prioritat (1-3). INSERT incidencia. Estado ABIERTA per defecte	✓ Impl.

RF2 7	Pujar document	Selector de PDF + camp títol + selector categoria (7 opcions). Upload Supabase Storage + INSERT documento	✓ Impl.
RF2 8	Xat directe veï	Al clicar veï en directori: busca o crea chat personal, navega a conversa	✓ Impl.
RF2 9	Contactes emergència	Llista amb botons de trucada. Linking.openURL('tel:+34112'). President pot afegir-ne	✓ Impl.
RF3 0	Calendari esdeveniments	Graella mensual. Modal per crear esdeveniment amb títol, descripció, hora, tipus	✓ Impl.

2.2 Requisits no funcionals

ID	Categoria	Descripció	Verificació
RNF 01	Rendiment	Temps de resposta de qualsevol pantalla inferior a 2 segons en xarxa WiFi normal	Proves manuals
RNF 02	Temps real	Missatges de xat lliurats en menys de 500ms en condicions normals de xarxa	WebSocket
RNF 03	Compatibilitat	Suport per iOS 13+ i Android 8+. Responsive entre 4.7" i 6.7" de pantalla	Proves disp.
RNF 04	Usabilitat	Interfície intuïtiva comprensible sense manual. Paleta de colors i tipografia consistents	Tests usuari
RNF 05	Aïllament dades	Les dades de cada comunitat són completament invisibles per a usuaris d'altres comunitats	Tests query
RNF 06	Seguretat auth	Contrasenyes mai emmagatzemades en plain text. Sessió invàlida detectada i redirigida a login	Supabase Auth
RNF 07	Escalabilitat	Arquitectura suporta com a mínim 100 veïns per comunitat i 50 missatges/dia sense degradació	Càlcul BD
RNF 08	Mantenibilitat	Codi TypeScript tipat, components reutilitzables, estils centralitzats per facilitar manteniment	Revisió codi

3. Tecnologies i arquitectura

3.1 Comparativa de tecnologies valorades

Framework mòbil

Criteri	React Native + Expo	Flutter (Dart)	Ionic (web)
Llenguatge	TypeScript / JavaScript	Dart (propietari)	HTML/CSS/JS
Rendiment	Natiu (JSI bridge)	Nadiu (compila)	WebView (lent)
Ecosistema	npm enorme	pub.dev creixin	npm
Corba aprenentatge	Baixa (JS familiar)	Alta (Dart nou)	Molt baixa
Hot reload	Sí (Expo Go)	Sí	Sí
Decisió	✓ ESCOLLIT	Descartat (Dart)	Descartat (WebView)

React Native amb Expo es va escollir per la familiaritat de l'equip amb JavaScript/TypeScript, el gran ecosistema de llibreries npm, i la capacitat de Expo Go per a provar canvis en temps real en dispositiu físic sense necessitat de compilar.

Backend / BaaS

Criteri	Supabase	Firestore (Google)	PocketBase
Base de dades	PostgreSQL (SQL)	Firestore (NoSQL)	SQLite
Queries complexes	Sí (SQL complet)	Limitades	Sí
Temps real	WebSockets nadius	Sí	SSE
Autenticació	Completa	Completa	Bàsica
Open Source	Sí (Apache 2.0)	No	Sí (MIT)
Pla gratuït	500MB BD, 1GB Store	Spark Plan limitat	Self-hosted
Decisió	✓ ESCOLLIT	Descartat (vendor lock)	Descartat (self-host)

Supabase es va escollir per la potencia de PostgreSQL que permet relacions complexes, queries SQL avançades i constraints d'integritat. La capacitat de Realtime per subscriure's a canvis en taules específiques va ser clau per al mòdul de xat.

3.2 Stack tecnològic escollit

Tecnologia	Versió	Rol en el projecte
React Native	0.81.5	Framework principal per construir la UI nativa. Renderitza components nadius iOS/Android
Expo	~54.0	Capa d'abstracció sobre React Native. Gestiona build, updates OTA, i accés a APIs natives
Expo Router	~4.0	Sistema de navegació basat en fitxers. Cada fitxer a /app/ correspon a una ruta
TypeScript	~5.3	Superset de JS amb tipatge estàtic. Preveu errors en compilació, millora autocomplete i refactoring
Supabase Client	^2.0	SDK oficial per interactuar amb Supabase: auth, queries, Realtime i Storage
PostgreSQL	15	SGBD relacional al backend Supabase. Suporta UUID, JSONB, triggers, constraints, RLS
React Native SVA	~4.3	SafeAreaView per gestionar les zones segures de notch i barra inferior en iOS
Expo Vector Icons	~14.0	Set d'icones Ionicons utilitzades a tota la interfície per consistència visual
Expo Document Picker	~12.0	Accés al sistema de fitxers del dispositiu per seleccionar PDFs
Expo Linking	~7.0	Obertura d'URLs externes: trucades telefòniques i navegadors

3.3 Arquitectura del sistema

L'aplicació segueix una arquitectura client-servidor clàssica on el client React Native es comunica directament amb el backend Supabase a través del SDK oficial. No hi ha cap servidor intermediari propi, cosa que simplifica radicalment el deployment i el manteniment.

Estructura de directoris del projecte

app/ — Pantalles i navegació (Expo Router file-based routing)

- |— _layout.tsx — Layout raíz amb AuthProvider (32 línies)
- |— index.tsx — Redireccionament inicial segun sessió (56 línies)
- |— login.tsx — Pantalla d'autenticació (130 línies)
- |— registro.tsx — Registre d'usuaris (144 línies)
- |— comunidad.tsx — Creació/unió a comunitat (588 línies)
- |— ajustes.tsx — Configuració i perfil (467 línies)
- |— nueva-incidencia.tsx — Formulari incidència (158 línies)
- |— nuevo-anuncio.tsx — Formulari anunci (115 línies)
- |— (tabs)/ — Pantalles amb barra de navegació inferior

- | |— index.tsx — Pantalla principal (Home) (214 línies)
- | |— chat.tsx — Llista de xats (698 línies)
- | |— economia.tsx — Gestió econòmica (928 línies)
- | |— contactos.tsx — Directori veïns (697 línies)
- | |— documentos.tsx — Repositori documents (511 línies)
- | |— incidencias.tsx — Llista incidències (176 línies)
- | |— votaciones.tsx — Enquestes i votacions (502 línies)
- | |— chat/[id].tsx — Conversa de xat individual (693 línies)

components/ — Components reutilitzables

- |— BottomTabBar.tsx — Barra de navegació (110 línies)
- |— CalendarioComunidad.tsx — Calendari d'esdeveniments (337 línies)

src/lib/supabase.ts — Configuració i instància Supabase

4. Base de dades

La base de dades és PostgreSQL 15 allotjada a Supabase Cloud. L'esquema conté 18 taules, 7 migracions aplicades, i més de 40 constraints d'integritat (CHECK, UNIQUE, FOREIGN KEY). A continuació es detalla cada taula amb les seves columnes, tipus, valors per defecte, constraints i relacions.

4.1 Taules del sistema

comunidad

Taula principal que representa cada comunitat de veïns. Cada registre és un entorn independent.

Columna	Tipus	Defecte	Nullable	Descripció
id	uuid	gen_random_uuid()	NO	Clau primària
nombre	text		NO	Nom de la comunitat
direccion	text		NO	Adreça completa
codigo	text		YES (UNIQUE)	Codi d'invitació de 6 caràcters única
max_vecinos	integer	100	NO	Límit de veïns (expansible)
created_at	timestampz	now()	NO	Data de creació

usuario

Usuaris de l'aplicació, vinculats a auth.users de Supabase. Camp rol amb CHECK constraint.

Columna	Tipus	Defecte	Nullable	Descripció
id	uuid	gen_random_uuid()	NO	PK. Coincideix amb auth.users.id
comunidad_id	uuid		YES	FK → comunidad.id. NULL si no té comunitat
rol	text		NO (CHECK)	CHECK: PRESIDENTE PROPIETARIO TRABAJADOR INQUILINO
nombre	text		NO	Nom de pila
apellidos	text		YES	Cognoms (opcionals)
email	text		NO (UNIQUE)	Email d'autenticació (coincideix auth)
telefono	text		YES	Telèfon opcional

created_at	timestampz	now()	NO	Data de registre
------------	------------	-------	----	------------------

chat i chat_miembro

Taules que implementen el sistema de xat. chat conté les converses; chat_miembro la relació N:M usuaris-converses.

chat: columna	Tipus	Defecte	Nullable	Descripció
id	uuid	gen_random_uuid()	NO	Clau primària
comunidad_id	uuid		NO	FK → comunidad.id
nombre	text		NO	Nom del grup o conversa
tipo	text	'grupo'	NO (CHECK)	personal grupo edificio servicio urgente
creado_por	uuid		YES	FK → usuario.id
ultimo_mensaje_at	timestampz	now()	YES	Per ordenar xats per activitat. Actualitzat per trigger

chat_miembro: columna	Tipus	Defecte	Nullable	Descripció
chat_id	uuid		NO	FK → chat.id
usuario_id	uuid		NO	FK → usuario.id
silenciado_hasta	timestampz	NULL	YES	NULL = no silenciado. Data futura = silenciado fins aquella data
activo	boolean	true	NO	false quan l'usuari surt del grup

pago, movimiento i gasto_reparto

Les tres taules que implementen el mòdul econòmic. pago registra quotes individuals, movimiento el registre comptable general, i gasto_reparto el desglossament dels gastos repartits.

pago: columna	Tipus	Defecte	Nullable	Descripció
mes	integer		NO (CHECK)	CHECK: mes >= 1 AND mes <= 12
anio	integer		NO (CHECK)	CHECK: anio >= 2000 AND anio <= 2100
importe	numeric		NO	Import de la quota en euros
estado	text	'PAGADO'	NO (CHECK)	PAGADO PENDIENTE FALLIDO
fecha_pago	timestampz	now()	YES	NULL si pendent. Actualitzat en marcar pagat

movimiento: columna	Tipus	Defecte	Nullable	Descripció
concepto	text		NO	Descripció del moviment
tipo	text		NO (CHECK)	ingreso gasto
importe	numeric		NO (CHECK)	CHECK: importe > 0
tipo_gasto	text	'comunidad'	NO (CHECK)	comunidad repartido. Permet filtrar el tipus de gasto

gasto_reparto: columna	Tipus	Defecte	Nullable	Descripció
movimiento_id	uuid		NO	FK → movimiento.id. El gasto origen
usuario_id	uuid		NO	FK → usuario.id. El veï que ha de pagar
importe	numeric		NO	Import individual (total / número veïns seleccionats)
estado	text	'PENDIENTE'	NO (CHECK)	PENDIENTE PAGADO
fecha_pago	timestampz	NULL	YES	NULL si pendent

encuesta, encuesta_opcion i encuesta_respuesta

Les taules de votacions implementen un model complet amb garantia d'un vot per persona mitjançant un UNIQUE constraint.

Taula	Camps clau	Descripció
encuesta	pregunta, activa, fecha_cierre	Enquesta principal. activa=true mentre està oberta. fecha_cierre s'omple en tancar
encuesta_opcion	encuesta_id, texto	Cada opció de resposta. Entre 2 i 6 opcions per enquesta
encuesta_respuesta	UNIQUE(usuario_id, encuesta_id)	Un vot per usuari per enquesta. El constraint UNIQUE al nivell de BD impedeix doble votació

incidencia

Registre de problemes reportats amb sistema de prioritats numèriques i estats amb CHECK constraints.

Columna	Tipus	Defecte	Nullable	Descripció
titulo	text		NO	Títol breu de la incidència
descripcion	text		YES	Descripció detallada opcional
estado	text	'ABIERTA'	NO (CHECK)	ABIERTA EN_PROCESO RESUELTA CERRADA
prioridad	integer	2	NO (CHECK)	CHECK: 1 (baixa) 2 (mitja) 3 (alta)
creada_por	uuid		YES	FK → usuario.id. L'autor del reporte

evento

Taula del calendari comunitari. El camp hora és text (no time) per simplicitat; l'aplicació auto-completa ':00' si l'usuari escriu només l'hora.

Columna	Tipus	Defecte	Nullable	Descripció
titulo	text		NO	Títol de l'esdeveniment
hora	text	'00:00'	NO	Hora en format HH:MM. Auto-completat des del client
tipo	text	'otro'	NO (CHECK)	reunion mantenimiento otro. Determina el color al calendari
fecha	date		NO	Data de l'esdeveniment en format YYYY-MM-DD

5. Funcionalitats implementades

A continuació es descriu detalladament el funcionament de cada mòdul: el flux d'usuari, la implementació tècnica, les queries involucrades i les decisions de disseny.

5.1 Autenticació i onboarding

Flux de registre

La pantalla de registre (`app/registro.tsx`, 144 línies) mostra un formulari amb nom, email, contrasenya i confirmació de contrasenya. La validació en client comprova que els camps no estiguin buits, que l'email tingui format vàlid, i que les contrasenyes coincideixin. Un vegada validat, es crida `supabase.auth.signUp({email, password})` que crea el compte a `auth.users`. En paral·lel es fa un INSERT a la taula `usuario` amb el nom, email, i sense comunitat assignada.

Flux de login

La pantalla de login (`app/login.tsx`, 130 línies) crida `supabase.auth.signInWithPassword()`. Si l'usuari ja té comunitat assignada (consulta a taula `usuario`), navega directament a la pantalla principal. Si no té comunitat, navega a la pantalla d'onboarding (`comunidad.tsx`).

Flux d'onboarding

La pantalla de comunitat (`app/comunidad.tsx`, 588 línies) ofereix dues opcions: crear comunitat o unir-se. En crear: s'introdueix nom i adreça, es genera un codi de 6 caràcters alfanumèrics amb `Math.random()`, i s'insereix a taula `comunidad`. Simultaniament es fa UPDATE a `usuario SET comunidad_id, rol='PRESIDENTE'`. El codi s'ensenya en un modal amb botó de copiar. En unir-se: l'usuari introdueix el codi, es fa `SELECT a comunidad on codigo=input`, i si existeix, UPDATE a `usuario SET comunidad_id, rol='PROPIETARIO'`.

5.2 Xat en temps real

Arquitectura tècnica

El mòdul de xat consta de dos fitxers principals: `app/(tabs)/chat.tsx` (698 línies) per a la llista de converses, i `app/chat/[id].tsx` (693 línies) per a la conversa individual. Total: 1.391 línies de codi.

La comunicació en temps real s'implementa mitjançant Supabase Realtime. A l'obrir una conversa, es crea una subscripció amb `supabase.channel()` que escolta events INSERT a la taula `mensaje` filtrats per `chat_id` igual al de la conversa actual. Quan arriba un missatge nou, s'afegeix directament a l'estat local de React (`setMensajes(prev => [...prev, nou])`) sense necessitat de refrescar tota la llista.

Llista de xats

La pantalla principal del xat carrega tots els chats on l'usuari és membre actiu (`chat_miembro.activo=true`), ordenats per `ultimo_mensaje_at DESC`. Cada element de la llista mostra el nom del grup, l'hora de l'últim missatge, i si el xat està silenciats. El president veu un botó '+' per crear nous grups.

Crear grup (president)

Modal amb: camp de text per al nom del grup, selector de tipus (edifici, serveis, urgent), i llista de tots els veïns de la comunitat amb checkboxes per seleccionar membres. En confirmar: INSERT a chat (nombre, tipo, comunidad_id, creado_por) i INSERT múltiples a chat_miembro, incloent-hi sempre l'usuari creador.

Conversa individual

Les bombolles de missatge segueixen l'estètica de WhatsApp: missatges propis a la dreta amb fons blau (#2F67E8) i text blanc, missatges aliens a l'esquerra amb fons blanc i vora subtil. Cada bombolla mostra l'avatar circular amb inicials del nom, el text, i el timestamp en format HH:MM. L'ScrollView automàticament fa scroll al final en arribar missatge nou.

Gestionar membres (president)

Des del menú '...' del header, el president accedeix a una pantalla de gestió amb dues opcions: afegir membres (llista de veïns de la comunitat que no són al grup) i eliminar membres (llista actual amb confirmació). En afegir: INSERT a chat_miembro. En eliminar: UPDATE chat_miembro SET activo=false.

Silenciar

L'usuari pot silenciar qualsevol grup via 3 opcions: per sempre (NULL, mai més notificació), 1 dia (`now() + 1 day`), o 1 setmana (`now() + 7 days`). S'actualitza `chat_miembro.silenciado_hasta`. A la llista de xats, els grups silenciats mostren una icona de campana ratllada.

5.3 Gestió econòmica

El mòdul econòmic és el més extens del projecte amb 928 línies de codi en `app/(tabs)/economia.tsx`. Està reservat per a President i Propietari; els Treballadors veuen una pantalla bloquejada.

Pestanya Resum

La pestanya principal mostra: saldo acumulat de la comunitat (calculat sumant tots els moviments: ingressos positius, gastos negatius), dos indicadors del mes actual (ingressos i gastos), i un selector de mes/any per navegar l'històric. A més, a la part inferior apareix la secció 'Gastos que em corresponen' amb tots els `gasto_reparto` assignats a l'usuari actual, mostrant els pagats (badge verd) i pendants (botó marcar pagat).

Pestanya Cuotes

Taula amb totes les quotes del mes seleccionat. Columnes: nom del veï, import, estat (badge verd PAGADO / vermell PENDIENTE), i botó d'acció per al president. Filtres sobre la taula: Tots / Pagats / Pendants. A l'extrem inferior: total cobrat. En marcar una quota: UPDATE pago SET estado='PAGADO', fecha_pago=now() + INSERT movimiento(tipo='ingreso', importe=quota.importe, concepto='Quota mes-any - nom') + reload de dades.

Pestanya Pressupostos

Cards verticals amb barra de progrés horitzontal per a cada projecte. La barra mostra el percentatge gastado/total amb el color personalitzat del pressupost. Exemple real: 'Ascensor nuevo' 12.000€ total, 4.800€ gastats = 40% en blau.

Pestanya Moviments

Historial cronològic invers de tots els moviments. Cada fila té : icona (fletxa avall verda=ingrés, fletxa amunt vermella=gasto comunitat, icona people morada=gasto repartit), concepte en negreta, data formatada (3 abr 2026), import amb color. A dalt del tot: dues cards amb el total d'ingressos i gastos acumulats.

Modal crear gasto

Accessible només per al president via botó '+' al header. El modal té dues pestanyes: 'Comunitat' i 'Repartit'. En mode Comunitat: concepte, import, data, INSERT a movimiento(tipo='gasto', tipo_gasto='comunidad'). En mode Repartit: concepte, import total, llista de veïns de la comunitat exclouent TRABAJADOR (query .neq('rol','TRABAJADOR')), checkboxes per seleccionar qui paga, i un indicador en temps real que mostra el cost per persona calculat com import/numSeleccionats. En confirmar: INSERT movimiento(tipo_gasto='repartido') + INSERT múltiples gasto_reparto amb importe_individual = importTotal / numSeleccionats.

5.4 Votacions

Crear encuesta (president)

Modal amb un camp de text per a la pregunta i una llista dinàmica d'opcions de resposta. L'usuari pot afegir opcions pulsant '+' (fins a 6) i eliminar-ne pulsant '-' (mínim 2). En confirmar: INSERT a encuesta(pregunta, activa=true, comunidad_id, creado_por) + INSERT múltiples a encuesta_opcion(encuesta_id, texto).

Votar

Cada encuesta activa es mostra com una card amb la pregunta i els botons d'opció. Quan l'usuari selecciona una opció: INSERT a encuesta_respuesta(encuesta_id, opcion_id, usuario_id). Si la INSERT falla per constraint UNIQUE (ja havia votat), es mostra error elegant. Després de votar, es recarreguen les opcions amb els totals actualitzats: SELECT COUNT(*) FROM

encuesta_respuesta WHERE encuesta_id=X GROUP BY opcion_id. Cada opció mostra barra de progrés proporcional i percentatge.

Tancar votació

Botó 'Tancar votació' visible només per al president quan activa=true. En confirmar: UPDATE encuesta SET activa=false, fecha_cierre=now(). Les enquestes tancades mostren l'opció amb més vots destacada amb fons verd i una icona de trofeu.

5.5 Incidències

Pantalla principal (app/(tabs)/incidencias.tsx, 176 línies) que mostra la llista d'incidències de la comunitat. Formulari separat (app/nueva-incidencia.tsx, 158 línies) per crear-ne de noves. Les incidències es filtren per comunidad_id i s'ordenen per prioridad DESC, created_at DESC.

Cada incidència es mostra com una card amb: títol en negreta, descripció (truncada a 2 línies), badge de prioritat (verd=baixa, groc=mitja, vermell=alta), badge d'estat (gris=oberta, blau=en procés, verd=resolta, negre=tancada), i la data de creació formatada. El formulari inclou: títol (requerit), descripció (multiline opcional), i selector de prioritat amb 3 opcions visuals.

5.6 Documents

Mòdul de repositori (app/(tabs)/documentos.tsx, 511 línies) amb llistat de documents i modal per pujar-ne. La pujada utilitza expo-document-picker per accedir al sistema de fitxers, que retorna un URI i les metadades del fitxer. L'arxiu es puja a Supabase Storage al bucket 'documentos' amb ruta: {comunidad_id}/{timestamp}_{nom_fitxer}. Paral·lelament s'insereix a la taula documento: titulo, categoria, tipo, archivo_path, fecha, comunidad_id.

Les categories disponibles (selector desplegable, no text lliure) són: Actes, Contractes, Assegurances, Normatives, Pressupostos, Factures, Altres. Cada document de la llista mostra: icona PDF, títol, badge de categoria, data de pujada, i botó de descarrega que obre la URL signada de Supabase Storage al navegador del sistema.

5.7 Contactes i emergències

Pantalla de contactes (app/(tabs)/contactos.tsx, 697 línies) amb dues pestanyes: Veïns i Emergències. La pestanya de Veïns carrega tots els usuaris de la comunitat, mostrant avatar amb inicials, nom complet, badge de rol (President blau, Propietari verd, Treballador groc) i indicador 'Tu' per al propi usuari. Cercador en temps real filtra per nom sense queries addicionals.

En clicar un veï: l'aplicació cerca si existeix un chat de tipus 'personal' entre els dos usuaris (SELECT chat_miembro cm1 JOIN chat_miembro cm2 ON cm1.chat_id=cm2.chat_id JOIN chat c ON c.id=cm1.chat_id WHERE cm1.usuario_id=A AND cm2.usuario_id=B AND c.tipo='personal'). Si existeix, navega directament. Si no, crea el chat i navega.

El president pot canviar el rol de qualsevol veï fent long-press sobre la seva fila. Apareix un Alert amb les opcions de rol disponibles (Propietari, Treballador, Inquilí, excloent el rol actual). En seleccionar: UPDATE usuario SET rol=nouRol WHERE id=veild.

La pestanya d'emergències mostra contactes crítics: 112 (Emergències), 092 (Policia Local), 080 (Bombers), 061 (Sanitat). El botó de trucada utilitza Linking.openURL('tel:+34112'). El president pot afegir contactes personalitzats (empresa d'ascensor, lampista habitual, etc.) via modal amb nom, telèfon, descripció i icona.

5.8 Calendari d'esdeveniments

El component CalendarioComunidad.tsx (337 línies) renderitza una graella mensual completa. Cada cel·la de la graella representa un dia i mostra punts de color si hi ha esdeveniments (verd=reunió, blau=manteniment, gris=altre). El selector de mes/any permet navegar entre mesos. En seleccionar un dia: es mostra la llista d'esdeveniments i un botó per afegir-ne un de nou.

El modal de crear esdeveniment té els camps: títol (requerit), descripció (opcional), hora (text, auto-complèta ':00' si l'usuari escriu només '19' → '19:00'), i tipus (reunió/manteniment/altre com a botons de selecció). El modal té KeyboardAvoidingView per evitar que el teclat tapi els camps. En guardar: INSERT a taula evento(comunidad_id, creado_por, titulo, descripcion, hora, tipo, fecha).

6. Sistema de permisos per rols

VeciApp implementa un model de control d'accés basat en rols (RBAC) amb tres nivells principals i un quart en reserves. Els rols es defineixen a nivell de base de dades amb un CHECK constraint, i s'implementen a nivell d'aplicació mitjançant guards condicionals, queries filtrades, i components adaptats.

6.1 Definició de rols

Rol	Descripció i permisos generals
PRESIDENTE	Administrador total de la comunitat. Pot realitzar qualsevol acció: crear grups, gestionar membres, crear gastos, marcar quotes, crear i tancar votacions, afegir contactes d'emergència, canviar rols als veïns. És l'única persona que té visió global de totes les dades.
PROPIETARIO	Propietari d'un pis a la comunitat. Té accés complet al xat, pot reportar incidències, consultar documents, votar en enquestes, i veure la seva situació econòmica. No pot crear gastos, marcar quotes d'altri, ni administrar grups.
TRABAJADOR	Personal de serveis (conserge, neteja, manteniment). Accés mínim: xat, incidències, documents i contactes. Sense accés a economia ni votacions. No comptabilitzat en repartiment de gastos.
INQUILINO	Rol en reserves per a inquilins. Definit al constraint de BD però no completament implementat a la UI. Permisos similars al Treballador previstos.

6.2 Taula de permisos detallada

Funcionalitat	President	Propietari	Treballador
Tab 'Economia' visible	✓	✓	✗ Ocult
Veure economia	✓ Completa	✓ Completa	✗ Bloq.
Crear gasto comunitat	✓	✗	✗
Crear gasto repartit	✓	✗	✗
Inclouút en reparto	✓	✓	✗ Exclouút
Marcar quota pagada	✓	✗	✗
Marcar reparto pagat	✓ (propi)	✓ (propi)	✗
Veure/votar votacions	✓	✓	✗ Bloq.
Crear votació	✓	✗	✗
Tancar votació	✓	✗	✗
Home: tauler anuncis	✓ + crear	✓ veure	✗ Ocult
Home: accés economia	✓	✓	✗ Ocult

Home: accés votacions	✓	✓	✗ Ocult
Crear grup de xat	✓	✗	✗
Gestionar membres grup	✓	✗	✗
Xat i missatgeria	✓	✓	✓
Reportar incidència	✓	✓	✓
Documents (veure/pujar)	✓	✓	✓
Directorí veïns + emerg.	✓	✓	✓
Afegir contactes emerg.	✓	✗	✗
Canviar rol veïns	✓ long-press	✗	✗
Calendari (veure/crear)	✓	✓	✓

6.3 Implementació tècnica dels rols

BottomAppBar dinàmica

El component components/BottomAppBar.tsx (110 línies) carrega el rol de l'usuari actual en muntar-se via useEffect. La definició de tabs inclou un camp opcional 'roles' que, si està present, filtra la tab per mostrar-la només als rols especificats. La tab 'Economia' té rols: ['PRESIDENTE', 'PROPIETARIO'], per tant és invisible per als Treballadors.

Guards de pantalla

Les pantalles Economia i Votacions implementen un guard al retorn del component. Després de carregar l'usuari, si rol === 'TRABAJADOR', es retorna una pantalla alternativa en comptes del contingut normal. Aquesta pantalla mostra una icona de cademat, el text 'Accés restringit' i 'Només propietaris i president'.

Exclusió en repartiment

Al modal de crear gasto repartit, la query que carrega la llista de veïns inclou .neq('rol', 'TRABAJADOR'). Això garanteix que els treballadors no apareguin com a opcions de selecció i, per tant, mai rebin ni paguin gastos repartits.

Renderitzat condicional

Botons i seccions que requereixen rol President s'engloben en condicionals del tipus {esPresidente && <Boto />}. La variable esPresidente es calcula com usuario?.rol === 'PRESIDENTE' després de carregar les dades. En casos on el rol tarda a carregar-se, s'utilitza l'estat cargandoRol per evitar parpadeos i mostrar una pantalla en blanc fins que el rol està disponible.

7. Disseny visual i experiència d'usuari

7.1 Principis de disseny

El disseny visual de VeciApp segueix quatre principis fonamentals: consistència (mateixa paleta, tipografia i espaiat a totes les pantalles), jerarquia visual clara (diferenciació entre títols, cos de text i informació secundària), retroalimentació immediata (estats de càrrega, missatges d'error clars, confirmacions), i accessibilitat (contrast suficient, 'rees de toc adequades, texts llegibles).

7.2 Paleta de colors

Element	Color HEX	Justificació i ús específic
Fons principal	#F0F4FB	Blau molt clar que dona sensació de netedat. Aplicat a totes les SafeAreaView
Color d'accent principal	#2F67E8	Blau corporatiu. Header, botons principals, elements actius, bombolles de xat pròpies
Color fosc (header)	#1E3A8A	Blau molt fosc per a backgrounds d'header en varies pantalles
Cards i contingut	#FFFFFF	Fons blanc per a targetes sobre el fons blau clar. Crea jerarquia visual
Text principal	#111827	Gairebé negre per a títols i text important. Contrast màxim
Text secundari	#6B7280	Gris mig per a descripcions, metadades, noms de seccions
Text terciari	#9CA3AF	Gris clar per a placeholders, timestamps, informació poc rellevant
Èxit / Pagat	#16A34A	Verd per a pagat, resultat, actiu. També #DCFCE7 com a fons del badge
Error / Pendent	#DC2626	Vermell per a pendent, error, alta prioritat. #FEE2E2 com a fons
Repartiment gastos	#7C3AED	Morat per a gastos repartits. Diferencia visualment d'ingressos i gastos normals
Advertència	#D97706	Taronja per a prioritat mitja en incidències
Fons subtil	#EEF2FF	Blau molt pàl·lid per a badges de president, fons de chips actius
Vores	#E5E7EB	Gris molt clar per a vores de cards, separadors, inputs

7.3 Tipografia i espaiat

La tipografia utilitza la font del sistema en cada plataforma (San Francisco a iOS, Roboto a Android) sense importar fonts externes. Això garanteix la màxima llegibilitat i un ús mínim de l'espai de l'aplicació. Les mides de text segueixen una escala consistent: 11px (etiquetes molt petites, timestamps), 13px (text secundari, descripcions), 15px (text de cos normal), 17px (títols de seccions), 20-24px (títols de pantalles).

L'espaiat segueix una graella de 4px: padding intern de cards: 16px, sep, margin entre elements llista: 12px, margin entre seccions: 16-24px, padding del contingut respecte als marges de pantalla: 16px. Les cards utilitzen border-radius de 12px, els botons 8px, i les bombolles de xat 20px per donar un aspecte suau i modern.

7.4 Gestió del teclat

Un dels problemes més comuns en aplicacions mòbils és que el teclat virtual tapi els inputs. VeciApp implementa `KeyboardAvoidingView` en totes les pantalles que contenen formularis: `login.tsx`, `registro.tsx`, `nou-anunci.tsx`, `nova-incidència.tsx`, `ajustes.tsx` (2 modals), `chat.tsx`, `contactos.tsx`, `documentos.tsx`, `votaciones.tsx`, i `CalendarioComunidad.tsx`. La propietat `behavior` es configura dinàmicament: `Platform.OS === 'ios' ? 'padding' : undefined`, ja que en Android el sistema ja gestiona el desplaçament automàticament.

8. Conclusions

8.1 Conclusions generals del projecte

VeciApp ha estat un projecte de gran envergadura que ha requerit integrar múltiples tecnologies, dissenyar una arquitectura escalable, i prendre decisions tècniques fonamentades. El resultat és una aplicació mòbil completament funcional amb més de 6.900 línies de codi TypeScript que dona resposta a una necessitat real de milers de comunitats de veïns.

Des del punt de vista acadèmic, el projecte ha permès aplicar i consolidar coneixements d'àrees molt diverses: disseny d'interfícies mòbils, bases de dades relacionals, autenticació i autorització, comunicació en temps real per WebSockets, gestió de fitxers al cloud, i control de versions amb Git.

8.2 Consecució dels objectius

ID	Objectiu	Resultat	Estat
OE0 1	Sistema autenticació	Login, registre i sessió persistent funcionant via Supabase Auth	✓ ASSOLIT
OE0 2	Invitació per codi	Codi de 6 caràcters generat, compartit i validat correctament	✓ ASSOLIT
OE0 3	Xat temps real	Realtime WebSockets funcionant. 1.391 línies de codi de xat	✓ ASSOLIT
OE0 4	Mòdul econòmic	928 línies. Quotes, gastos, repartiments, pressupostos i historial	✓ ASSOLIT
OE0 5	Votacions digitals	Creació, participació única i tancament amb resultats visuals	✓ ASSOLIT
OE0 6	Incidències	3 prioritats, 4 estats, filtres per comunitat	✓ ASSOLIT
OE0 7	Repositori documents	Pujada PDF a Supabase Storage, categorització, descarrega	✓ ASSOLIT
OE0 8	Contactes i emerg.	Directori veïns, xat directe, contactes d'emergència amb trucada	✓ ASSOLIT
OE0 9	Calendari	Graella mensual, creació d'esdeveniments amb types i colors	✓ ASSOLIT
OE1 0	Sistema de rols	3 rols amb permisos diferenciats a UI i queries	✓ ASSOLIT
OE1 1	Aïllament dades	comunidad_id a totes les taules, filtres a totes les queries	✓ ASSOLIT
OE1 2	Compatibilitat mòbil	Provat en iOS 16/17 i Android 11/13. Paleta visual unificada	✓ ASSOLIT

8.3 Valoració de la metodologia

La metodologia de desenvolupament incremental ha demostrat ser adequada per a un projecte d'aquest tipus. El fet de poder provar cada mòdul en dispositiu físic immediatament després de la implementació ha permès detectar problemes ràpidament (com la gestió del teclat, les rac, o l'ordre de càrrega de dades) i corregir-los sense que s'acumulés deute tècnic.

La principal desviació respecte a la planificació inicial va ser la complexitat del mòdul de xat, que va requerir gairebé el doble del temps estimat a causa de la gestió del cicle de vida de les subscripcions Realtime i els problemes de memòria amb l'scroll automàtic. El mòdul econòmic també va créixer en complexitat en afegir el repartiment proporcional i l'exclusió de treballadors.

9. Visió de futur

9.1 Notificacions push

La funcionalitat més demanada pels usuaris de proves és rebre notificacions fora de l'aplicació. Expo Notifications proporciona una API completa per a notificacions push en iOS i Android. La implementació requereix: obtenir un token de notificació per cada dispositiu i guardar-lo a la BD, crear una Edge Function de Supabase que s'activi mitjançant un trigger quan s'insereixi un missatge nou, i enviar la notificació via Expo Push API als dispositius dels membres del xat que no estiguin actius i no tinguin el xat silenciado.

9.2 Publicació a les botigues

L'aplicació està pràcticament preparada per ser publicada. Els passos pendents són: crear l'icona i la splash screen definitives, generar el build de producció amb eas build de Expo Application Services, signar l'app per iOS amb el certificat de distribució d'Apple, i submetre als processos de revisió d'App Store i Google Play.

9.3 Funcionalitats addicionals

Curt termini: mode fosc (dark mode) amb un switch a Ajustos, suport per imatges i fitxers als missatges de xat, gravació de veu per missatges, i sistema de confirmació de lectura (ticks de Llegit) a les converses.

Mittjà termini: versió web responsive amb Next.js reutilitzant el backend Supabase, sistema de reserves d'espais comuns (piscina, sala de reunions) amb calendari de disponibilitat, i integració amb pasarel·les de pagament com Stripe per a quotes automàtiques.

Llarg termini: API pública per a integracions amb tercers (portals automàtics intel·ligents, comptadors elèctrics), sistema de reportes i analytics per al president, i assistent de IA per respondre preguntes sobre normativa de comunitats.

10. Glossari

Tecnologies i conceptes tècnics

React Native: Framework de JavaScript de Meta per construir aplicacions mòbils natives per iOS i Android usant una única base de codi. Renderitza components nadius reals, no una WebView.

Expo: Plataforma de codi obert que s'executa sobre React Native i simplifica el desenvolupament, build i deployment. Expo Go permet provar l'app en dispositius físics sense compilar.

Expo Router: Sistema de navegació per Expo basat en fitxers. Cada fitxer dins el directori /app/ es converteix automàticament en una ruta de l'aplicació.

TypeScript: Superset de JavaScript que afegeix tipatge estàtic opcional. Permet detectar errors en temps de compilació en lloc d'execució, millora l'autocompletat i facilita el refactoring.

Supabase: Plataforma Backend as a Service (BaaS) de codi obert basada en PostgreSQL. Proporciona base de dades, autenticació, Realtime (WebSockets), Storage (fitxers) i Edge Functions en un sol servei.

PostgreSQL: Sistema de gestió de bases de dades relacionals (SGBDR) de codi obert. Suporta tipus de dades avançats (UUID, JSONB, arrays), funcions, triggers, i Row Level Security.

Realtime (Supabase): Funcionalitat que permet subscriure's a canvis en taules de la BD via WebSockets. Quan s'insereix, actualitza o elimina una fila, els clients subscrits reben l'event immediatament.

WebSocket: Protocol de comunicació bidireccional sobre una connexió TCP permanent. A diferència de HTTP (request-response), un WebSocket permet enviar dades en qualsevol moment des del servidor al client.

UUID: Universally Unique Identifier. Identificador de 128 bits generat aleatòriament que té una probabilitat pràcticament nul·la de col·lisió. S'utilitza com a clau primària a totes les taules.

RBAC: Role-Based Access Control. Model de control d'accés on els permisos s'assignen a rols en lloc d'usuaris individuals, i els usuaris s'assignen a rols.

KeyboardAvoidingView: Component de React Native que ajusta automàticament la posició del contingut quan el teclat virtual s'obra, evitant que tapi els camps d'entrada.

SafeAreaView: Component que garanteix que el contingut no queda tapat pel notch, la barra d'estat o la barra d'inici en dispositius moderns iOS i Android.

Supabase Storage: Servei d'emmagatzematge de fitxers de Supabase, similar a Amazon S3. Organitza fitxers en buckets i permet accés via URLs signades temporals.

Expo Document Picker: Mòdul d'Expo que permet a l'usuari seleccionar fitxers del sistema d'arxius del dispositiu (Documents, iCloud Drive, Google Drive, etc.).

Trigger (PostgreSQL): Funció que s'executa automàticament davant un event (INSERT, UPDATE, DELETE) en una taula. S'utilitza per actualitzar `ultimo_mensaje_at` en insertar missatges nous.

UNIQUE constraint: Restricció de base de dades que garanteix que un valor (o combinació de valors) no es pot repetir en cap altra fila. Clau per prevenir dobles vots a la taula `encuesta_respuesta`.

12. Bibliografía

- [1] Meta Open Source. React Native Documentation. Disponible a: <https://reactnative.dev/docs>.
- [2] Expo Team. Expo Documentation. Disponible a: <https://docs.expo.dev>.
- [3] Expo Team. Expo Router Documentation. Disponible a: <https://docs.expo.dev/router>.
- [4] Supabase Inc. Supabase Documentation. Disponible a: <https://supabase.com/docs>.
- [5] Supabase Inc. Supabase Realtime. Disponible a: <https://supabase.com/docs/guides/realtime>.
- [6] Microsoft. TypeScript Handbook. Disponible a: <https://www.typescriptlang.org/docs/handbook>.
- [7] The PostgreSQL Global Development Group. PostgreSQL 15 Documentation. Disponible a: <https://www.postgresql.org/docs/15>.
- [8] FETT MDN Web Docs. WebSockets API. Disponible a: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API.
- [9] Expo Team. expo-document-picker. Disponible a: <https://docs.expo.dev/versions/latest/sdk/document-picker>.
- [10] Expo Team. expo-linking. Disponible a: <https://docs.expo.dev/versions/latest/sdk/linking>.
- [11] Evan Bacon. Expo Router: Write Once, Route Everywhere. Blog Expo. Disponible a: <https://expo.dev/blog/expo-router>.
- [12] Supabase Inc. The Architecture of Supabase. Disponible a: <https://supabase.com/blog/supabase-architecture>.