



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERIA

DESARROLLO DE UNA PLATAFORMA MÓVIL ACCESIBLE DE COHOUSING PARA MITIGAR EL AISLAMIENTO SOCIAL EN ADULTOS MAYORES AUTÓNOMOS

Por: Lizandro Enrique Ruiz Mansilla

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Informático

Profesor Patrocinante: Geoffrey Jean-Pierre Christophe Hecht

Profesor Copatrocinante: Gonzalo Eduardo Pérez Correa

Supervisora Externa: Marcela Alejandra Ruiz Aguilar

Marzo 2025

Concepción, Chile

© 2025, Lizandro Enrique Ruiz Mansilla

Ninguna parte de esta tesis puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

Dedico esta memoria de titulo a mi padre, madre y hermana.

AGRADECIMIENTOS

Quiero agradecer a todas las personas que me acompañaron en mi proceso universitario.

Comenzando por mi familia, quienes me brindaron su apoyo incondicional ante cada tropiezo durante esta etapa. A mi padre y a mi madre, por ponerme siempre como prioridad; y a mi hermana, por ese apoyo emocional que solo una hermana sabe dar. También, a mi pareja, Claudia, por el abrigo emocional y físico en todo este camino.

Asimismo, quiero agradecer a Marcela Ruiz por confiarme la oportunidad de desarrollar este proyecto junto a ella; y a Jorge Ruiz, que en paz descanse, por haber dado el primer voto de confianza a mi familia para que yo pudiera estudiar lejos de mi ciudad natal.

Por último, pero no menos importantes, a mis amigos, que siempre estuvieron conmigo y me regalaron perspectivas nuevas cuando más las necesité.

A cada uno y una: gracias por sostenerme cuando flaqueé y por celebrar conmigo cada paso. Esta memoria también es suya.

Resumen

El aislamiento social en personas mayores representa un problema creciente en Chile, agravado por el envejecimiento poblacional y la insuficiencia de pensiones que limita las opciones habitacionales. Este proyecto desarrolló una plataforma móvil de cohousing orientada a facilitar la formación de comunidades habitacionales compartidas entre adultos mayores autónomos.

La metodología empleada incluyó un proceso de empatización con usuarios mayores mediante entrevistas y observación directa, identificando barreras específicas en el uso de tecnología móvil. Se establecieron principios de diseño centrados en simplicidad y accesibilidad, aplicando las dos reglas de oro: simplificación máxima con máximo dos reglas principales por pantalla, y elementos de gran tamaño con separación generosa entre controles interactivos.

La implementación utilizó React Native con Expo SDK para el desarrollo móvil multiplataforma, junto con Supabase como backend relacional que proporciona autenticación, base de datos PostgreSQL y funcionalidades en tiempo real. La arquitectura de estado separó datos remotos mediante TanStack Query y estado local a través de Zustand, optimizando el rendimiento y la gestión de caché.

La plataforma implementa tres roles de usuario: administradores que gestionan aprobación de propiedades, inquilinos que buscan espacios de convivencia, y anfitriones que publican sus propiedades. Los flujos principales incluyen registro y autenticación con integración Google, publicación de propiedades con galería fotográfica, sistema de mensajería bidireccional, creación y gestión de contratos de convivencia, y módulo de declaración de pagos compartidos.

La validación se realizó mediante evaluación heurística basada en los principios de Nielsen, aplicada por cinco evaluadores con experiencia en desarrollo de software. Los resultados mostraron un promedio general de 4.12/5.0, destacando fortalezas en consistencia de diseño (4.80/5.0), prevención de errores (4.73/5.0) y diseño minimalista (4.73/5.0). Las áreas de mejora identificadas incluyen limitaciones en flexibilidad de uso (2.33/5.0) y ausencia de funciones para deshacer acciones (2.60/5.0).

El proyecto establece una base funcional para facilitar el cohousing entre personas mayores, con potencial impacto social al reducir el aislamiento, económico al optimizar costos habitacionales, y tecnológico al demostrar el desarrollo de software inclusivo. Las

mejoras futuras propuestas incluyen implementación de función deshacer, mejora de mensajes de error, sistema de búsqueda en ayuda, y expansión hacia funcionalidades como gestión financiera automatizada y contratación de servicios comunitarios.

Summary

Social isolation among older adults represents a growing problem in Chile, aggravated by population aging and pension inadequacy that limits housing options. This project developed a mobile cohousing platform aimed at facilitating the formation of shared housing communities among autonomous older adults.

The methodology employed included an empathy process with elderly users through interviews and direct observation, identifying specific barriers in mobile technology usage. Design principles centered on simplicity and accessibility were established, applying two golden rules: maximum simplification with no more than two main principles per screen, and large-sized elements with generous separation between interactive controls.

The implementation used React Native with Expo SDK for cross-platform mobile development, along with Supabase as a relational backend providing authentication, PostgreSQL database, and real-time functionality. The state architecture separated remote data through TanStack Query and local state via Zustand, optimizing performance and cache management.

The platform implements three user roles: administrators who manage property approval, tenants who search for living spaces, and hosts who publish their properties. Main flows include registration and authentication with Google integration, property publishing with photo gallery, bidirectional messaging system, creation and management of cohabitation contracts, and shared payment declaration module.

Validation was conducted through heuristic evaluation based on Nielsen's principles, applied by five evaluators with software development experience. Results showed an overall average of 4.12/5.0, highlighting strengths in design consistency (4.80/5.0), error prevention (4.73/5.0), and minimalist design (4.73/5.0). Identified improvement areas include limitations in usage flexibility (2.33/5.0) and absence of undo functions (2.60/5.0).

The project establishes a functional foundation for facilitating cohousing among older adults, with potential social impact by reducing isolation, economic by optimizing housing costs, and technological by demonstrating inclusive software development. Proposed future improvements include undo function implementation, error message enhancement, help search system, and expansion toward functionalities such as automated financial management and community service contracting.

Índice general

AGRADECIMIENTOS	I
Resumen	II
Summary	IV
1. Introducción	1
1.1. Objetivos Asumidos	2
1.1.1. Objetivo General	2
1.1.2. Objetivos Específicos	2
2. Empatización Y Estado del Arte	4
2.1. Metodología de Investigación Cualitativa	4
2.2. Perfiles de Usuarios Entrevistados	5
2.2.1. Mapa de Empatía: Síntesis de Testimonios	6
2.3. Análisis de Hallazgos Principales	8
2.3.1. Validación de la Problemática	8
2.3.2. Caracterización de Perfiles	9
2.3.3. Necesidades Identificadas en el Proceso de Empatización	10
2.4. Hallazgos para el desarrollo de la plataforma	11
2.4.1. Principios de Diseño Identificados	11
2.4.2. Oportunidades de Desarrollo Identificadas	12
2.5. Estado del arte: Soluciones similares existentes	12
2.5.1. Soluciones Existentes	12
2.5.2. Limitaciones de las Soluciones Existentes	13

2.6.	Marco Teórico de Usabilidad para Personas Mayores	13
2.6.1.	Fundamentos Científicos del Diseño	14
2.6.2.	Las Dos Reglas de Oro del Diseño para Personas Mayores	14
2.6.3.	Directrices Específicas de Diseño	15
3.	Diseño	18
3.1.	Metodología de trabajo	18
3.1.1.	Desarrollo por sprints	18
3.2.	Requerimientos	20
3.2.1.	Requerimientos funcionales	20
3.2.2.	Requerimientos no funcionales	23
3.3.	Casos de uso	24
3.3.1.	Inquilino	24
3.3.2.	Anfitrión	25
3.3.3.	Administrador	26
3.4.	Diseño de la base de datos	26
3.4.1.	Modelo Entidad-Relación	26
3.4.2.	Políticas de seguridad (RLS)	29
3.5.	Arquitectura del Sistema	32
3.5.1.	Contenedores del sistema	32
3.5.2.	Componentes de la aplicación móvil	33
4.	Implementación	35
4.1.	Implementación	35
4.1.1.	Selección de Tecnologías	35
4.1.2.	Consideraciones Técnicas Transversales	36
4.1.3.	Flujo de Autenticación	37
4.1.4.	Flujo de Registro de Propiedad	38
4.1.5.	Flujo de Formación de Comunidad	40
4.1.6.	Flujo de Vida en Comunidad	45
4.1.7.	Flujos Secundarios	47

5. Verificación y validación	51
5.1. Verificación	51
5.1.1. Metodología de Verificación	51
5.1.2. Proceso de Verificación y Resultados	52
5.2. Validación	53
5.2.1. Metodología	53
5.2.2. Resultados de la Evaluación	56
5.2.3. Análisis de Resultados	58
5.2.3.1. Fortalezas Identificadas	58
5.2.3.2. Áreas de Mejora Identificadas	58
6. Conclusión y mejoras futuras	60
6.1. Conclusiones	60
6.1.1. Logro de Objetivos	60
6.1.2. Valor Generado e Impacto Potencial	61
6.2. Mejoras Futuras	61
6.3. Reflexiones Finales	63
Referencias	64
Anexo	66
A. Anexos	66
A1. Alertas de Manejo de Excepciones	66
A2. Menú Central de Anfitrión Sin Inquilinos	67
A3. Vista de Pagos de Anfitrión Sin Datos	68
A4. Plantilla de Evaluación Heurística	69
A5. Repositorio de Imágenes en Alta Resolución	69

Índice de cuadros

1.	Mapa de Empatía: Testimonios organizados por categorías de observación	7
2.	Directrices de diseño móvil para personas mayores - Parte 1	16
3.	Directrices de diseño móvil para personas mayores - Parte 2	17
4.	Descripción de cada sprint realizado en el desarrollo de la plataforma .	19
5.	Requerimientos funcionales de autenticación y gestión de usuarios . . .	20
6.	Requerimientos funcionales del sistema de propiedades	21
7.	Requerimientos funcionales del sistema de mensajería	21
8.	Requerimientos funcionales del sistema de comunidad y contratos . . .	22
9.	Requerimientos funcionales del sistema de gestión financiera	22
10.	Requerimientos no funcionales de accesibilidad	23
11.	Requerimientos no funcionales de seguridad y rendimiento	23
12.	Políticas de seguridad RLS para la tabla Usuario	29
13.	Políticas de seguridad RLS para la tabla Propiedad	29
14.	Políticas de seguridad RLS para las tablas Chat y Mensaje	30
15.	Políticas de seguridad RLS para la tabla Acuerdo	30
16.	Políticas de seguridad RLS para la tabla Pago	31
17.	Políticas de seguridad RLS para la tabla Gasto Mensual	31
18.	Criterios de evaluación para visibilidad del estado del sistema	53
19.	Criterios de evaluación para relación con el mundo real	53
20.	Criterios de evaluación para control y libertad del usuario	54
21.	Criterios de evaluación para consistencia y estándares	54
22.	Criterios de evaluación para prevención de errores	54

23.	Criterios de evaluación para reconocimiento sobre recuerdo	54
24.	Criterios de evaluación para flexibilidad y eficiencia	54
25.	Criterios de evaluación para diseño estético y minimalista	55
26.	Criterios de evaluación para recuperación de errores	55
27.	Criterios de evaluación para ayuda y documentación	55
28.	Resultados de evaluación heurística - Parte 1	56
29.	Resultados de evaluación heurística - Parte 2	57
30.	Resumen de resultados por heurística	57
31.	Categorización de mejoras futuras por tipo y prioridad	62

Índice de figuras

1.	Diagrama de casos de uso de la plataforma	24
2.	Modelo Entidad-Relación del sistema	27
3.	Diagrama de contenedores del sistema	32
4.	Diagrama de componentes de la aplicación móvil (C3)	33
5.	Mapa de navegación del flujo de autenticación	37
6.	Mapa de navegación del flujo de registro de propiedad	38
7.	Mapa de navegación: Subida de fotografías adaptado para personas mayores	39
8.	Mapa de navegación: Notificación de propiedad aprobada	40
9.	Mapa de navegación: Búsqueda y primer contacto	41
10.	Mapa de navegación: Notificación y respuesta del anfitrión	42
11.	Mapa de navegación: Creación de invitación por el anfitrión	43
12.	Mapa de navegación: Aceptación de invitación y formación de comunidad	44
13.	Mapa de navegación: Declaración de pagos por inquilinos	46
14.	Mapa de navegación: Confirmación de pagos por anfitriones	47
15.	Mapa de navegación: Gestión de foto de perfil	48
16.	Mapa de navegación: Panel de administración para gestión de propiedades	49
17.	Mapa de navegación: Flujo de rechazo de propiedad	49
18.	Tours virtuales contextuales por tipo de usuario	50
A1.1.	Ejemplos de alertas implementadas en la plataforma	66
A2.1.	Vista del menú central para anfitrión sin inquilinos	67
A3.1.	Vista de pagos de anfitrión sin declaraciones registradas	68

A4.1. Plantilla de evaluación heurística completada por los evaluadores durante el proceso de validación	69
--	----

Capítulo 1

Introducción

El envejecimiento de la población es una realidad que afecta a diversas sociedades en todo el mundo. En Chile, la Séptima Encuesta Nacional sobre la Inclusión y la Exclusión Social de las Personas Mayores reveló que un 81,5% de los encuestados considera que el país está mal preparado para atender las necesidades de las personas mayores (Cooperativa Ciencia, 2024). Este dato pone en evidencia una creciente preocupación por la falta de preparación institucional para enfrentar el envejecimiento, especialmente en áreas críticas como la salud y el cuidado personal. La sensación de desprotección y la falta de confianza en las instituciones políticas refuerzan la idea de que el apoyo actual no es suficiente.

A nivel global, una de las principales problemáticas que enfrentan las personas mayores es la soledad. El cohousing, un modelo de vivienda comunitaria, ha surgido como una alternativa para combatir esta problemática al promover la convivencia y el apoyo mutuo entre sus residentes. En países como Brasil, proyectos de cohousing están ayudando a las personas mayores a evitar el aislamiento y a vivir en comunidades que fomentan la interacción social (Lemos, 2024). Esta tendencia también ha ganado terreno en Europa, donde el concepto de *senior living* ofrece soluciones de vivienda adaptadas a las necesidades de las personas mayores, promoviendo un envejecimiento activo y social (Culmia, 2024).

Ante el reto de abordar no solo las necesidades básicas sino también el bienestar emocional y social de las personas mayores, surge la propuesta de desarrollar una plataforma móvil que conecte a personas mayores autónomas para facilitar el acceso a alternativas de cohousing, buscando mejorar la calidad de vida de esta población.

mediante un enfoque centrado en la usabilidad y accesibilidad. Este sistema pretende facilitar la convivencia en un entorno compartido, considerando las limitaciones cognitivas, físicas y tecnológicas que pueden enfrentar las personas mayores. Al diseñar interfaces simples e intuitivas, el proyecto busca reducir barreras tecnológicas que tradicionalmente han excluido a este grupo demográfico de las soluciones digitales modernas.

El diseño prioriza la experiencia del usuario mayor, implementando principios de diseño universal que faciliten la navegación, con elementos con textos grandes, contraste alto y flujos de trabajo simplificados. Esta herramienta busca contribuir a que las personas mayores puedan disfrutar de una vida tranquila y socialmente activa dentro de una comunidad que procura fomentar la interacción entre residentes con intereses similares y adaptarse a sus necesidades específicas.

1.1. Objetivos Asumidos

1.1.1. Objetivo General

Desarrollar una plataforma móvil de cohousing para personas mayores autónomos que facilite la búsqueda y gestión de viviendas compartidas, combatiendo el aislamiento social mediante una interfaz diseñada específicamente para las necesidades y limitaciones de este grupo demográfico.

1.1.2. Objetivos Específicos

1. Estudiar el estado del arte de soluciones similares existentes y tecnologías aplicables al problema, para así, definir la arquitectura del software, asegurando que sea escalable, segura y flexible para integrar las funcionalidades clave del proyecto, con capacidad para futuras expansiones y mejoras.
2. Diseñar y desarrollar una interfaz de usuario (UI) intuitiva y una experiencia de usuario (UX) optimizada, específicamente adaptadas para personas mayores, facilitando la navegación, selección y gestión de viviendas compartidas de manera eficaz y sencilla.
3. Realizar pruebas iterativas de las funcionalidades durante el desarrollo de la aplicación, asegurando la calidad técnica en cada fase, y validar la experiencia

de usuario mediante evaluación heurística basada en principios de usabilidad reconocidos, identificando problemas de diseño que puedan afectar la experiencia de las personas mayores al usar la plataforma.

Capítulo 2

Empatización Y Estado del Arte

Reconociendo la complejidad del aislamiento social en la tercera edad y la necesidad de soluciones tecnológicas verdaderamente adaptadas a las realidades de las personas mayores chilenas, se realizó un proceso de empatización mediante entrevistas en profundidad. Este proceso permite comprender no solo las manifestaciones externas de la problemática, sino también las emociones, pensamientos y comportamientos que caracterizan la experiencia vivida de los potenciales usuarios.

2.1. Metodología de Investigación Cualitativa

Se adoptó un enfoque de investigación cualitativa basado en entrevistas semi-estructuradas en profundidad, complementado con la técnica de mapas de empatía para organizar y analizar los hallazgos. El mapa de empatía es una herramienta de síntesis que estructura las observaciones en cuatro cuadrantes —lo que los usuarios hacen, dicen, piensan y sienten— permitiendo capturar la riqueza de las experiencias subjetivas e identificar patrones de comportamiento que no serían evidentes mediante métodos cuantitativos (Kelley, 2018). Las entrevistas se diseñaron para explorar estas cuatro dimensiones fundamentales. El reclutamiento se realizó mediante muestreo por referencia, iniciando con una participante conocida que facilitó el contacto con otras personas mayores de su red social, diversificando así los perfiles de convivencia. Las entrevistas se realizaron en los domicilios de los participantes previa coordinación telefónica.

Los participantes fueron seleccionados aplicando los siguientes criterios:

- Personas mayores entre 60 y 80 años
- Condición de autovalencia física y cognitiva
- Residencia independiente (propia o familiar)
- Diversidad en situaciones de convivencia (solos, en pareja, con familia)
- Disponibilidad para participar voluntariamente en el estudio

2.2. Perfiles de Usuarios Entrevistados

Se realizaron cinco entrevistas en profundidad con personas mayores entre 67 y 76 años, lo que permitió identificar arquetipos representativos del segmento objetivo. La caracterización minuciosa de estos perfiles resulta esencial para comprender las diversas perspectivas y requerimientos dentro del grupo demográfico.

■ **Matrimonio Autovalente: María y Walter**

María (67 años) y Walter (70 años) representan el perfil de personas mayores que mantienen una vida independiente en pareja. Ambos conservan rutinas estructuradas y participan activamente en actividades sociales y familiares. Su situación actual es estable, con apoyo familiar cercano, pero reconocen las limitaciones futuras de su situación habitacional actual.

Walter muestra una personalidad más cautelosa respecto a cambios en su estilo de vida, expresando que “sería muy difícil” adaptarse a nuevas modalidades de convivencia. María, por su parte, valora significativamente su privacidad personal, manifestando que “no le gusta que invadan su privacidad”. Este perfil ilustra un tipo de persona mayor que reconoce la validez de alternativas habitacionales futuras sin experimentar urgencia inmediata por cambiar su situación actual.

■ **Persona Mayor Proactiva: Teresa**

Teresa (71 años) representa un perfil de persona mayor altamente proactiva y pensante en la búsqueda de alternativas habitacionales. A pesar de su edad, mantiene actividad laboral por necesidad económica y demuestra una actitud pragmática hacia el futuro. Su declaración “el día que no sea autovalente me voy a un hogar” revela una planificación consciente de su envejecimiento.

Teresa demostró ser la más receptiva al concepto de cohousing, expresando

directamente “quiero irme a un cohousing”. Su disposición al cambio y búsqueda activa de soluciones ilustran un perfil de persona mayor que podría beneficiarse significativamente de la plataforma. Este tipo de usuarios resulta fundamental para comprender las motivaciones y expectativas de quienes buscan activamente alternativas al envejecimiento tradicional.

■ **Coordinadora Comunitaria: Irma**

Irma se caracteriza por su rol de liderazgo en actividades grupales de personas mayores en su comunidad religiosa. Su perfil combina la comprensión empática de las necesidades de otras personas mayores con capacidad de influencia social. Además, su entusiasmo por el proyecto (“sería súper bueno para nosotros”) y su trabajo como coordinadora de grupos definen un perfil de persona mayor que no solo conoce sus propias necesidades, sino que entiende las necesidades colectivas de su grupo demográfico. Estos perfiles son especialmente valiosos en términos de ejemplificar cómo las soluciones nuevas y habilitantes pueden impactar positivamente a toda la comunidad, más allá de aquellos directamente involucrados.

■ **Personas Mayores con Experiencias de Soledad**

Carmen (67 años) y Emilia (76 años) representan perfiles complementarios de personas mayores que han experimentado diferentes formas de dependencia. Carmen vivió sola durante tres años, describiendo vívidamente que “días enteros sola se me hacía largo”, mientras que Emilia ha transitado hacia una dependencia parcial de sus familiares para la gestión de aspectos económicos y tecnológicos.

Desde una perspectiva sobre el cuidado, Emilia (“prefiero que me cuiden otras personas y no mis familiares”) reveló su temor a que su necesidad de cuidado fuera una carga para su familia. Los dos perfiles muestran cuán paradójico puede ser el tema de la necesidad de vivienda en la vejez; una persona mayor desea independencia, pero al mismo tiempo necesita apoyo cada vez mayor.

2.2.1. Mapa de Empatía: Síntesis de Testimonios

Para organizar y visualizar de manera estructurada los hallazgos del proceso de empatización, se elaboró un mapa de empatía que consolida los testimonios de los cinco entrevistados según cuatro dimensiones de observación: lo que Hacen, Dicen,

Piensen y Sienten, permitiendo identificar patrones de comportamiento y necesidades comunes que fundamentan el diseño de la plataforma.

HACER	DECIR	PENSAR	SENTIR
Mantienen rutinas estructuradas (levantarse temprano, tareas del hogar)	“Días enteros sola se me hacía largo” (Carmen)	La confianza es fundamental para compartir vivienda	Terror/Miedo: A la soledad, dependencia, caídas, ser una carga familiar
Realizan actividades físicas (gimnasia por YouTube, kinesiólogo, jardinear)	“Le tengo pánico a la soledad” (Irma)	Los chilenos son culturalmente desconfiados con extraños	Gratitud: Por tener familia cercana, salud, trabajo
Participan en actividades sociales (talleres, bochas, grupos de iglesia)	“Es súper importante sociabilizar” (Teresa)	Las pensiones son insuficientes para vivir dignamente	Entusiasmo: Por alternativas innovadoras como el cohousing
Gestionan tareas domésticas (cocinar, limpiar, lavar)	“El día que no sea autovalente me voy a un hogar” (Teresa)	Los asilos tradicionales son negligentes y caros	Ansiedad: Por cambios en rutinas, pérdida de privacidad
Usan tecnología básica (WhatsApp, juegos en celular)	“Prefiero que me cuiden otras personas y no mis familiares” (Emilia)	La privacidad personal es muy valorada	Esperanza: En soluciones dignas para personas mayores
Algunos manejan sus propios gastos, otros delegan en familiares	“Quiero irme a un cohousing” (Teresa)	Es importante mantener la autovalencia el mayor tiempo posible	Empatía: Por otras personas mayores en situación de abandono
Realizan compras organizadas (mensuales y semanales)	“Sería súper bueno para nosotros” (Irma)	El cohousing es mejor alternativa que asilos tradicionales	Frustración: Con la tecnología pero con determinación de aprender
Mantienen visitas familiares regulares	“Es buena la idea” (Emilia)	Es una solución económica viable para bajas pensiones	Inquietud: Por el futuro económico y pérdida de autonomía
Coordinan grupos de personas mayores	“Yo creo que sería muy difícil acá” (Walter)	Es necesario para combatir la soledad	
	“A mí no me gusta que me invada mi privacidad” (María)	Requiere compatibilidad y confianza entre residentes	
	“Me gusta ver lo que me gusta tener dentro del refrigerador” (Carmen)	Debe preservar espacios privados	
		Funcionaría mejor con personas solas	

Cuadro 1: Mapa de Empatía: Testimonios organizados por categorías de observación

2.3. Análisis de Hallazgos Principales

El análisis de las entrevistas reveló patrones consistentes que validan la problemática del aislamiento social en personas mayores y confirman la relevancia del cohousing como alternativa habitacional.

2.3.1. Validación de la Problemática

Soledad como Factor Crítico

El análisis mostró que el 60 % de los entrevistados estaban preocupados por la soledad. Es importante comprender que no solo es la presencia física de otras personas, sino también la oportunidad de interacción humana significativa y el propósito dentro de la comunidad lo que importa.

Irma expresa la dimensión emocional más intensa: “le tengo pánico a la soledad”, revelando que para algunas personas mayores, la soledad no es simplemente una incomodidad, sino una fuente de ansiedad existencial. Teresa complementa esta perspectiva enfatizando que “es súper importante sociabilizar”, posicionando la interacción social como una necesidad básica para el ser humano, más no un lujo accesorio.

Insuficiencia Económica Estructural

La dimensión económica surgió como una preocupación transversal que justifica la relevancia de alternativas económicamente viables como el cohousing. Teresa, a sus 71 años, debe mantener actividad laboral debido a la insuficiencia de su pensión para cubrir gastos básicos. Esta dura realidad refleja un problema sistémico que afecta a una porción considerable de personas mayores en Chile.

La creencia generalizada de que “las pensiones no alcanzan” evidencia una necesidad real de alternativas habitacionales que optimicen costos sin comprometer la calidad de vida. En este contexto, el cohousing se presenta como una alternativa concreta, al permitir compartir gastos fijos entre pares, sin sacrificar la autonomía personal.

Rechazo a Alternativas Tradicionales

Todos los entrevistados manifestaron una visión negativa hacia los asilos tradicionales, caracterizándolos como espacios de “negligencia y maltrato” con “costos prohibitivamente elevados”. Esta percepción generalizada evidencia una carencia en las opciones

habitacionales disponibles para las personas mayores, justificando la relevancia de desarrollar alternativas como la plataforma desarrollada.

La resistencia no solo se basa únicamente en aspectos económicos, sino también en la sensación de pérdida de dignidad y autonomía. Emilia prefiere “que la cuiden otras personas y no su familia”, pero dentro de un marco que preserve su identidad e independencia.

2.3.2. Caracterización de Perfiles

Analizando los patrones de respuesta, se logró identificar tres tipos de perspectivas hacia las alternativas habitacionales innovadoras:

Usuarios Proactivos

Representados por Teresa e Irma, este grupo se caracteriza por:

- Búsqueda activa de alternativas habitacionales
- Alta disposición al cambio y adaptación
- Comprensión empática de problemáticas comunitarias
- Capacidad de articular claramente los beneficios del cohousing

Usuarios Reflexivos con Reservas

Carmen y Emilia representan este grupo, que se caracteriza por:

- Reconocer la validez de la necesidad pero mantener cautela
- Requerir información detallada y garantías de funcionamiento
- Valorar altamente la preservación de rutinas establecidas
- Considerar alternativas bajo condiciones específicas de confianza

Usuarios Estables con Perspectiva Futura

María y Walter ejemplifican este grupo que:

- Mantiene estabilidad satisfactoria en su situación actual
- Reconoce la necesidad en otras personas mayores sin sentirla personalmente
- Podría considerar alternativas ante cambios en sus circunstancias futuras

- Proporciona perspectivas valiosas sobre resistencias y facilitadores

2.3.3. Necesidades Identificadas en el Proceso de Empatización

Durante las entrevistas, los participantes expresaron necesidades específicas que consideran fundamentales para adoptar una solución de cohousing. Más que requisitos técnicos, son criterios que guían la evolución de la plataforma y respaldan la toma de decisiones a lo largo del desarrollo:

1. Transparencia en la Gestión de Gastos: La complejidad percibida en la división de gastos comunes emerge como una barrera significativa. Los usuarios expresaron la necesidad de un sistema transparente que facilite el registro y seguimiento de gastos compartidos, minimizando potenciales conflictos.

2. Mecanismos de Construcción de Confianza: La preocupación por la convivencia con “extraños” revela la necesidad de establecer procesos que permitan conocer a potenciales compañeros de vivienda antes de comprometerse, facilitando la construcción gradual de confianza.

3. Preservación de Espacios Privados: Todos los entrevistados enfatizaron la importancia de mantener espacios personales inviolables, señalando que cualquier solución debe equilibrar la vida comunitaria con el respeto absoluto a la privacidad individual.

4. Apoyo Personalizado: Los participantes reconocieron que mudarse a una vivienda compartida representaría un cambio significativo en sus vidas, por lo que manifestaron la necesidad de contar con diferentes niveles de apoyo y servicios que se adapten a sus circunstancias cambiantes, incluyendo tanto asistencia técnica para el uso de la plataforma como acompañamiento emocional durante el proceso de transición.

5. Opciones de Salida Claras: La capacidad de modificar o terminar arreglos que no funcionen emergió como un factor crítico para reducir el riesgo percibido al tomar la decisión inicial de participar en cohousing.

Barreras Identificadas

Desconfianza Cultural: La caracterización de los chilenos como “culturalmente desconfiados con extraños” representa un desafío que debe abordarse mediante mecanismos de verificación y construcción gradual de confianza.

Resistencia al Cambio: La valoración de rutinas establecidas requiere un proceso de transición gradual y apoyo durante la adaptación.

Preocupaciones de Higiene: Las diferencias en estándares de limpieza e higiene personal emergen como fuente potencial de conflictos.

2.4. Hallazgos para el desarrollo de la plataforma

El análisis realizado permitió identificar directrices concretas para el desarrollo de la plataforma, organizadas en principios de diseño y oportunidades de implementación técnica.

2.4.1. Principios de Diseño Identificados

El proceso de empatización reveló cuatro principios fundamentales para el diseño de la plataforma, basados en las necesidades expresadas por los usuarios entrevistados:

Transparencia como Requerimiento Fundamental

Los usuarios expresaron la necesidad de visibilidad completa sobre costos, reglas de convivencia y características de potenciales compañeros de vivienda. Los hallazgos indican que la información incompleta o ambigua genera desconfianza y abandono.

Control del Usuario como Principio Central

Los entrevistados manifestaron la importancia de mantener control sobre aspectos específicos de su experiencia, desde la selección de compañeros hasta la configuración de espacios privados. Esto sugiere que el diseño futuro debe priorizar la autonomía del usuario.

Adopción Gradual como Estrategia

Los hallazgos indican que la adopción debería facilitarse mediante experiencias graduales que permitan familiarización antes del compromiso total. Los usuarios sugirieron la importancia de procesos como visitas o períodos de conocimiento previo.

Flexibilidad como Factor de Confianza

Los usuarios manifestaron que la posibilidad de realizar modificaciones posteriores reduce significativamente el riesgo percibido al tomar decisiones iniciales. Este hallazgo indica la importancia de diseñar sistemas adaptables.

2.4.2. Oportunidades de Desarrollo Identificadas

Basándose en las necesidades expresadas por los usuarios, se identificaron las siguientes oportunidades para el desarrollo de la plataforma:

Facilitación de la Construcción de Confianza

Los hallazgos sugieren la importancia de desarrollar mecanismos que permitan a los usuarios conocerse gradualmente antes de comprometerse a vivir juntos, abordando la barrera cultural de desconfianza identificada.

Sistema de Transparencia Financiera

La necesidad expresada de claridad en los gastos compartidos indica una oportunidad para desarrollar herramientas que faciliten el registro y seguimiento transparente de los gastos comunitarios.

Flexibilidad y Adaptabilidad

Las preocupaciones sobre comprometerse a arreglos rígidos sugieren la importancia de diseñar sistemas que permitan ajustes y modificaciones según evolucionen las necesidades de los usuarios.

2.5. Estado del arte: Soluciones similares existentes

Para contextualizar el desarrollo de la plataforma y validar las oportunidades identificadas durante el proceso de empatización, se realizó un análisis de soluciones similares existentes que buscan enfrentar las necesidades identificadas anteriormente. La búsqueda de estas iniciativas se realizó mediante investigación web para identificar alternativas de cohousing orientadas a adultos mayores, buscando soluciones que abordaran problemáticas similares a las expresadas por los entrevistados.

2.5.1. Soluciones Existentes

El análisis reveló que las soluciones especializadas en cohousing para personas mayores son considerablemente limitadas:

Silvernest/HomeShare Online (EE.UU.): Plataforma especializada en personas mayores que ofrece servicios legales y verificación de antecedentes por \$24.95 mensuales. Disponible solo en 32 estados de EE.UU. y únicamente en inglés (Silvernest, 2024).

Nesterly (EE.UU.): Modelo intergeneracional basado en web que conecta personas mayores con jóvenes, incluyendo intercambio de servicios por descuentos en renta. Aunque innovador, se enfoca principalmente en usuarios jóvenes y carece de aplicación móvil nativa (Nesterly, 2024).

Fundación Cohousing (Chile): Organización sin fines de lucro especializada en "Senior Cohousing" para adultos mayores activos, enfocada en la innovación y diseño de comunidades autogestionadas. Sin embargo, no cuenta con plataforma tecnológica digital para conectar personas interesadas en cohousing, limitándose a la promoción conceptual del modelo (Fundación Cohousing, 2024).

2.5.2. Limitaciones de las Soluciones Existentes

El análisis comparativo revela limitaciones críticas en las soluciones actuales que impiden su adopción efectiva por personas mayores:

- 1. Accesibilidad Digital Ausente:** Ninguna plataforma digital implementa comprehensivamente los estándares de accesibilidad para personas mayores identificados en la investigación académica.
- 2. Funcionalidades Específicas Inexistentes:** Las aplicaciones internacionales carecen de herramientas para gestión transparente de gastos compartidos, coordinación de cuidados, o supervisión familiar opcional.
- 3. Falta de Procesos Graduales:** Las plataformas existentes no ofrecen mecanismos para construcción progresiva de confianza ni períodos de conocimiento mutuo antes del compromiso habitacional, aspectos críticos identificados en el proceso de empatización.
- 4. Falta de Herramientas de Conexión:** Aunque organizaciones como Fundación Cohousing promueven activamente el modelo, la ausencia de herramientas digitales para conectar personas interesadas dificulta la implementación práctica de comunidades de cohousing.

2.6. Marco Teórico de Usabilidad para Personas Mayores

Las limitaciones de accesibilidad digital identificadas en las soluciones existentes evidencian la necesidad de fundamentar científicamente el diseño de interfaces para

personas mayores. Para abordar esta carencia, se realizó una investigación de la literatura académica especializada en usabilidad móvil para este grupo demográfico.

2.6.1. Fundamentos Científicos del Diseño

La investigación identificó dos trabajos fundamentales que proporcionan el marco teórico para el diseño de la plataforma:

Revisión Sistemática de Directrices de Diseño (Gómez-Hernández et al., 2023)

Realizaron una revisión sistemática y análisis temático de 40 artículos académicos (2010-2021) que incluyeron pruebas de usabilidad con usuarios reales de 60 años o más. Este estudio resultó en la identificación de 27 directrices de diseño específicas y la definición de 2 reglas de oro fundamentales que deben seguirse en el diseño de cualquier aplicación móvil destinada a usuarios mayores.

Priorización de Barreras de Usabilidad (Awan et al., 2021)

Realizaron una revisión sistemática utilizando el Proceso de Jerarquía Analítica (AHP) para identificar y priorizar 15 barreras críticas de usabilidad en personas mayores. Entre las barreras más significativas destacan las dificultades de lectura y selección por pequeño tamaño de fuente y pantalla, además de la tendencia de los adultos mayores a perderse navegando menús complejos.

2.6.2. Las Dos Reglas de Oro del Diseño para Personas Mayores

En el primer estudio mencionado se establecen dos principios fundamentales que sintetizan los requerimientos esenciales para interfaces móviles accesibles a personas mayores:

Regla de Oro 1: Simplificar al Máximo Reducir conceptos, pasos y opciones visibles en cada interfaz. Limitar a máximo 2 ideas principales por pantalla. Esta regla aborda directamente las limitaciones cognitivas identificadas, minimizando la carga mental requerida para completar tareas.

Regla de Oro 2: Aumentar Tamaño y Separación Implementar botones grandes y holgados con área táctil mayor que el elemento visual, considerando las limitaciones motoras y visuales características del envejecimiento.

2.6.3. Directrices Específicas de Diseño

Las 27 directrices científicamente validadas que fundamentan las reglas de oro, se organizan en seis categorías: Ayuda y formación (1-3) abarca formación inicial, guías audiovisuales y ayuda contextual para reducir barreras tecnológicas; navegación (4-6) establece rutas poco profundas entre ideas principales de vistas, salidas seguras visibles y foco en la acción principal; diseño visual (7-15) incluye ubicación de controles, separación generosa, iconografía familiar, tipografía grande y alto contraste; carga cognitiva (16-20) propone una acción por pantalla, lenguaje simple, confirmaciones explícitas y eliminación de límites temporales; finalmente, interacción (21-27) define gestos simples, áreas táctiles amplias, feedback multimodal y consistencia en patrones de uso.

Las siguientes tablas presentan cada directriz junto con su aplicabilidad y consideración en el diseño de la plataforma:

N°	Categoría	Directriz	Consideración en el Diseño
1	Ayuda y formación	Formación inicial presencial	Demo guiada durante registro inicial
2	Ayuda y formación	Tutoriales en vídeo	Tips contextuales con animaciones cortas
3	Ayuda y formación	Ayuda contextual paso a paso	Tooltips persistentes en funciones críticas
4	Navegación	Rutas poco profundas	Máximo 3 niveles de navegación por idea principal de pantalla
5	Navegación	Salida segura	Botón de retroceso visible en todas las pantallas
6	Navegación	Foco en la acción actual	Ocultación de funciones secundarias
7	Diseño visual	Controles lejos de bordes	Botones centrados con margen seguro
8	Diseño visual	Pocos elementos por pantalla	Máximo 5-7 elementos principales por vista
9	Diseño visual	Iconos concretos y familiares	Iconos universales (casa, mensaje, perfil)
10	Diseño visual	Iconos siempre con etiqueta	Texto descriptivo bajo cada icono
11	Diseño visual	Controles claramente tocables	Botones con relieve y sombras
12	Diseño visual	Separación generosa	Mínimo 2mm entre elementos interactivos
13	Diseño visual	Tamaño de fuente grande	20pt para texto general, 30pt para crítico
14	Diseño visual	Alto contraste	Mínimo 4.5:1 en todos los textos

Cuadro 2: Directrices de diseño móvil para personas mayores - Parte 1

N°	Categoría	Directriz	Consideración en el Diseño
15	Diseño visual	Colores diferenciadores	Paleta de 4 colores principales
16	Carga cognitiva	Una acción principal por pantalla	Botón de acción principal destacado
17	Carga cognitiva	Lenguaje simple y concreto	Evitar jerga técnica, usar palabras familiares
18	Carga cognitiva	Confirmaciones explícitas	Diálogos de confirmación para acciones importantes
19	Carga cognitiva	Mensajes de estado claros	Indicadores de progreso y mensajes informativos
20	Carga cognitiva	Evitar timeouts cortos	Sesiones extendidas, sin límites de tiempo
21	Interacción	Preferir tap sobre gestos	Evitar swipe, pinch y gestos complejos
22	Interacción	Área táctil grande	Mínimo 48dp para todos los controles
23	Interacción	Feedback multimodal	Visual + vibración en interacciones
24	Interacción	Evitar doble tap	Todas las acciones con tap simple
25	Interacción	Tiempo de respuesta rápido	Máximo 1 segundo para respuestas del sistema
26	Interacción	Opción deshacer	Posibilidad de revertir acciones críticas
27	Interacción	Consistencia en patrones	Mismos gestos y ubicaciones en toda la app

Cuadro 3: Directrices de diseño móvil para personas mayores - Parte 2

Capítulo 3

Diseño

3.1. Metodología de trabajo

El desarrollo de la plataforma siguió una metodología ágil teniendo reuniones periódicas con el profesor patrocinante y con Marcela Ruiz, creadora e impulsora inicial del proyecto. Marcela, además de proporcionar retroalimentación sobre el enfoque del desarrollo, lidera la visión estratégica y gestiona activamente la búsqueda de financiamiento mediante postulaciones a programas de inversión, siendo su rol fundamental para guiar tanto las decisiones actuales de desarrollo como la proyección futura hacia la eventual comercialización del proyecto.

3.1.1. Desarrollo por sprints

El proyecto se estructuró en 8 sprints según el itinerario del **Cuadro 4**. El primer sprint se dedicó a la empatización con usuarios y toma de requerimientos, los siguientes 5 sprints se enfocaron en el desarrollo de funcionalidades principales, mientras que los sprints 7 y 8 se dedicaron a la investigación de usabilidad y posterior adaptación de interfaces para personas mayores.

Es importante destacar que el **Cuadro 4** presenta la duración real que tomó cada sprint durante el desarrollo del proyecto. Si bien la planificación inicial contemplaba completar el trabajo en un semestre académico, el desarrollo se extendió debido a que durante los primeros dos meses y medio del proyecto fue necesario atender exámenes y tratamientos médicos que limitaron temporalmente la dedicación al proyecto. Una vez superada esta situación, se retomó el desarrollo con normalidad hasta su finalización.

Al finalizar cada sprint, el profesor patrocinante proporcionaba retroalimentación valiosa sobre el trabajo realizado, mientras que se solicitó apoyo puntual al profesor copatrocinante cuando surgieron dudas específicas relacionadas con tecnologías e implementación técnica.

Sprint	Duración	Objetivo
1	3 semanas	Empatización y toma de requerimientos
2	4 semanas	Configuración de arquitectura base e implementación de sistema de autenticación
3	3 semanas	Desarrollo del sistema de gestión de usuarios y perfiles
4	4 semanas	Implementación del sistema de gestión de propiedades
5	4 semanas	Desarrollo del sistema de mensajería en tiempo real entre usuarios
6	5 semanas	Implementación del sistema de gestión comunitaria y pagos
7	4 semanas	Investigación de usabilidad y principios de diseño para personas mayores
8	8 semanas	Adaptación de interfaz de usuario según principios identificados

Cuadro 4: Descripción de cada sprint realizado en el desarrollo de la plataforma

3.2. Requerimientos

Los requerimientos de la plataforma surgieron del proceso de empatización complementado con el marco teórico de usabilidad para personas mayores establecido anteriormente, concretando en requerimientos funcionales del sistema y requerimientos no funcionales de accesibilidad.

3.2.1. Requerimientos funcionales

Los requerimientos funcionales se organizan según tres roles de usuario: **Inquilinos**, **Anfitriones** y **Administradores**, los cuales se detallan mas adelante.

Requerimiento	Descripción
RF1	El sistema debe permitir el registro de usuarios con datos personales validos
RF2	El sistema debe permitir autenticación segura mediante credenciales (email/contraseña) o a través de Google Auth
RF3	El sistema debe verificar la identidad de usuarios nuevos y mantener sesiones activas
RF4	El sistema debe gestionar diferentes tipos de usuarios con roles y permisos específicos
RF5	Los usuarios deben poder visualizar su perfil completo con foto y datos personales

Cuadro 5: Requerimientos funcionales de autenticación y gestión de usuarios

El sistema de autenticación del **Cuadro 5** busca facilitar el proceso de registro a los usuarios mayores. Tras autenticarse, el sistema mantiene la sesión activa, asigna los permisos correspondientes según el tipo de usuario y le permite revisar su perfil para verificar la información ingresada al registrarse.

Requerimiento	Descripción
RF6	Los anfitriones deben poder registrar propiedades y ser guiados en el proceso
RF7	Las propiedades deben pasar por un proceso de aprobación antes de ser públicamente visibles
RF8	El sistema debe permitir búsqueda contextual que filtre propiedades disponibles
RF9	Los usuarios deben poder visualizar información completa y detallada de cada propiedad
RF10	El sistema debe permitir a los usuarios contactar directamente al anfitrión de una propiedad

Cuadro 6: Requerimientos funcionales del sistema de propiedades

Los requerimientos del *Cuadro 6* implementan un flujo donde los anfitriones registran propiedades siguiendo un proceso guiado, luego deben esperar la aprobación de un administrador para ser publicadas. Los inquilinos pueden buscar y filtrar propiedades/comunidades disponibles, y visualizar información detallada de las mismas, mientras que el contacto directo con el anfitrión simplifica el inicio de la comunicación.

Requerimiento	Descripción
RF11	El sistema debe facilitar la comunicación directa entre inquilinos y anfitriones
RF12	Los mensajes deben intercambiarse en tiempo real con diferenciación visual clara
RF13	El sistema debe mostrar una lista organizada de conversaciones activas
RF14	El sistema debe proporcionar ayuda contextual para facilitar la comunicación inicial
RF15	Los usuarios deben recibir notificaciones de nuevos mensajes entrantes

Cuadro 7: Requerimientos funcionales del sistema de mensajería

El sistema de mensajería del *Cuadro 7* permite a inquilinos y anfitriones comunicarse directamente con diferenciación visual clara entre mensajes y ayuda contextual para facilitar la comunicación inicial. Los usuarios pueden acceder a una lista organizada de conversaciones activas, mientras las notificaciones en tiempo real los mantienen informados sin requerir monitoreo manual constante.

Requerimiento	Descripción
RF16	Los anfitriones deben poder crear contratos de convivencia
RF17	Los inquilinos deben poder responder a contratos de convivencia
RF18	El sistema debe gestionar automáticamente los estados de los contratos
RF19	Los miembros de una comunidad deben poder ver información básica de contacto de otros miembros
RF20	El sistema debe proporcionar información acerca de la comunidad

Cuadro 8: Requerimientos funcionales del sistema de comunidad y contratos

El **Cuadro 8** muestra los requerimientos que permiten a los anfitriones crear invitaciones de convivencia, mientras los inquilinos pueden aceptar o rechazar estas propuestas. Una vez formada la comunidad, los miembros acceden a información de contacto y datos generales de su grupo, facilitando la interacción entre compañeros de vivienda.

Requerimiento	Descripción
RF21	Los inquilinos deben poder declarar sus pagos mensuales
RF22	El sistema debe permitir a los anfitriones visualizar y confirmar las declaraciones de pago realizadas
RF23	El sistema debe permitir consulta histórica de pagos declarados
RF24	El sistema debe calcular automáticamente el monto mensual a pagar por cada inquilino
RF25	El sistema debe notificar a los usuarios sobre fechas y actividades de pago

Cuadro 9: Requerimientos funcionales del sistema de gestión financiera

El sistema financiero del **Cuadro 9** permite a los inquilinos declarar sus pagos de forma sencilla, mientras los anfitriones pueden visualizar y confirmar estas declaraciones. El sistema calcula automáticamente los montos mensuales a declarar y envía recordatorios, manteniendo a todos los miembros informados sobre el estado financiero de la comunidad.

3.2.2. Requerimientos no funcionales

Requerimiento	Descripción
RNF1	La interfaz de usuario debe respetar las 2 reglas de oro de diseño UI para personas mayores
RNF2	El sistema debe usar tipografía grande y alto contraste para facilitar la lectura
RNF3	Los elementos interactivos deben ser suficientemente grandes y estar bien separados
RNF4	La navegación debe limitarse a máximo 3 niveles de profundidad por idea principal de vista, con botón atrás siempre visible
RNF5	El sistema debe permitir personalización básica de la interfaz según las necesidades del usuario

Cuadro 10: Requerimientos no funcionales de accesibilidad

Los requerimientos de accesibilidad del *Cuadro 10* se basan directamente en las reglas de oro y directrices establecidas en el marco teórico (sección 2.6).

Requerimiento	Descripción
RNF6	El sistema debe implementar medidas de seguridad robustas para proteger datos de usuarios
RNF7	El sistema debe usar autenticación segura y protección adecuada para archivos sensibles
RNF8	El sistema debe gestionar eficientemente los datos remotos para optimizar el rendimiento
RNF9	El sistema debe garantizar comunicación en tiempo real estable
RNF10	El sistema debe garantizar disponibilidad continua y persistencia de sesiones

Cuadro 11: Requerimientos no funcionales de seguridad y rendimiento

Los requerimientos de seguridad y rendimiento del *Cuadro 11* garantizan protección robusta de datos y comunicación estable sin comprometer la simplicidad de uso.

3.3. Casos de uso

Una vez establecidos los requerimientos del sistema, se crearon los siguientes Casos de Uso representados en el diagrama de la *Figura 1*.

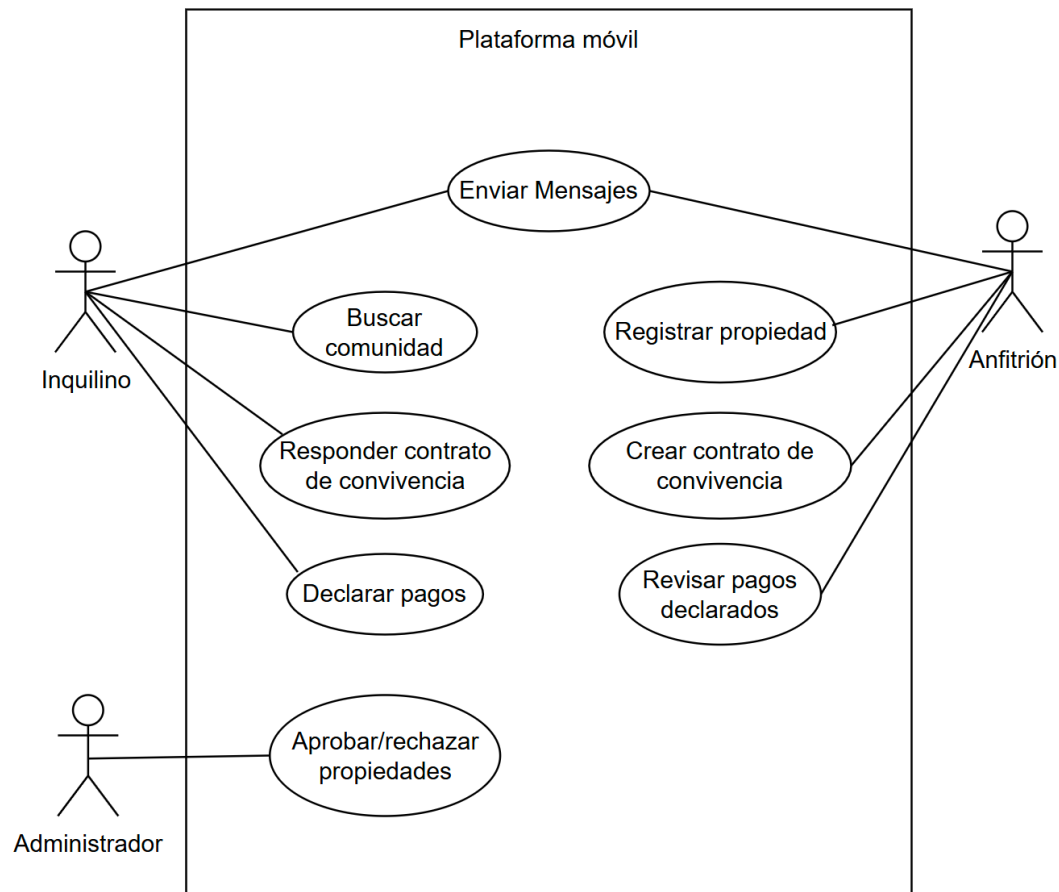


Figura 1: Diagrama de casos de uso de la plataforma

Como se puede observar, el sistema se compone de tres actores principales que reflejan los diferentes roles identificados durante el proceso de empatización:

3.3.1. Inquilino

Este actor representa a las personas mayores que buscan vivir en una comunidad de cohousing. El inquilino puede realizar las siguientes acciones:

- **Buscar comunidad**

Permite explorar y filtrar las propiedades/comunidades de cohousing disponibles según sus preferencias de ubicación, precio y tipo de propiedad.

- **Enviar mensajes**

Comunicarse directamente con los anfitriones mediante el sistema de mensajería en tiempo real para intercambiar información sobre la convivencia y las características específicas de la comunidad. El primer contacto se realiza desde la vista de detalle de una propiedad.

- **Responder contrato de convivencia**

Aceptar o rechazar los contratos de convivencia creados por los anfitriones, formalizando así el acuerdo habitacional.

- **Declarar pagos**

Registrar mensualmente los pagos realizados correspondientes a su participación en la comunidad.

3.3.2. Anfitrión

Este actor representa a las personas mayores que ofrecen espacios en sus propiedades para formar una comunidad de cohousing. El anfitrión interactúa con el sistema de la siguiente manera:

- **Registrar propiedad**

Registrar su propiedad en la plataforma siguiendo un proceso guiado, proporcionando información detallada y fotografías del espacio. La propiedad debe ser aprobada por un administrador antes de ser visible públicamente.

- **Enviar mensajes**

Comunicarse con inquilinos interesados o actuales mediante el sistema de mensajería para coordinar aspectos de la convivencia y responder consultas sobre la comunidad.

- **Crear contrato de convivencia**

Formalizar acuerdos habitacionales con inquilinos específicos, estableciendo fechas de inicio y término del período de convivencia.

- **Revisar pagos declarados**

Visualizar y confirmar las declaraciones de pago mensuales realizadas por los inquilinos de su propiedad.

3.3.3. Administrador

Este actor representa al personal encargado de supervisar y moderar la plataforma. Actualmente debido al estado temprano del proyecto, el administrador tiene una sola responsabilidad directa:

- **Aprobar/rechazar propiedades**

Revisar las solicitudes de registro de propiedades enviadas por los anfitriones, aprobándolas para su publicación en la plataforma o rechazándolas proporcionando retroalimentación específica sobre los motivos del rechazo.

3.4. Diseño de la base de datos

El sistema utiliza PostgreSQL con Supabase como Backend as a Service (BaaS), aprovechando sus capacidades de Row Level Security (RLS), autenticación integrada y funcionalidades en tiempo real. La elección de esta arquitectura responde a la necesidad de escalabilidad identificada en los objetivos, permitiendo que la plataforma crezca desde unos pocos usuarios iniciales hasta potencialmente miles de comunidades activas sin requerir cambios estructurales significativos. PostgreSQL proporciona robustez y características avanzadas como índices parciales y particionamiento, mientras que Supabase ofrece escalabilidad automática de infraestructura y distribución geográfica cuando sea necesario.

3.4.1. Modelo Entidad-Relación

Tomando en cuenta las necesidades del sistema y los requerimientos funcionales identificados, se decidió utilizar una base de datos relacional que implementa un esquema de 11 tablas principales.

Se adoptaron las siguientes decisiones de diseño para optimizar tanto la experiencia de los usuarios mayores como la escalabilidad del sistema:

Separación Usuario-Rol: Se optó por una tabla independiente de roles para facilitar la asignación de permisos específicos según la función de cada usuario en el sistema. Esta normalización permite agregar nuevos roles sin modificar la estructura de usuarios, facilitando la escalabilidad horizontal cuando se requieran nuevos tipos de actores sin impactar el esquema existente.

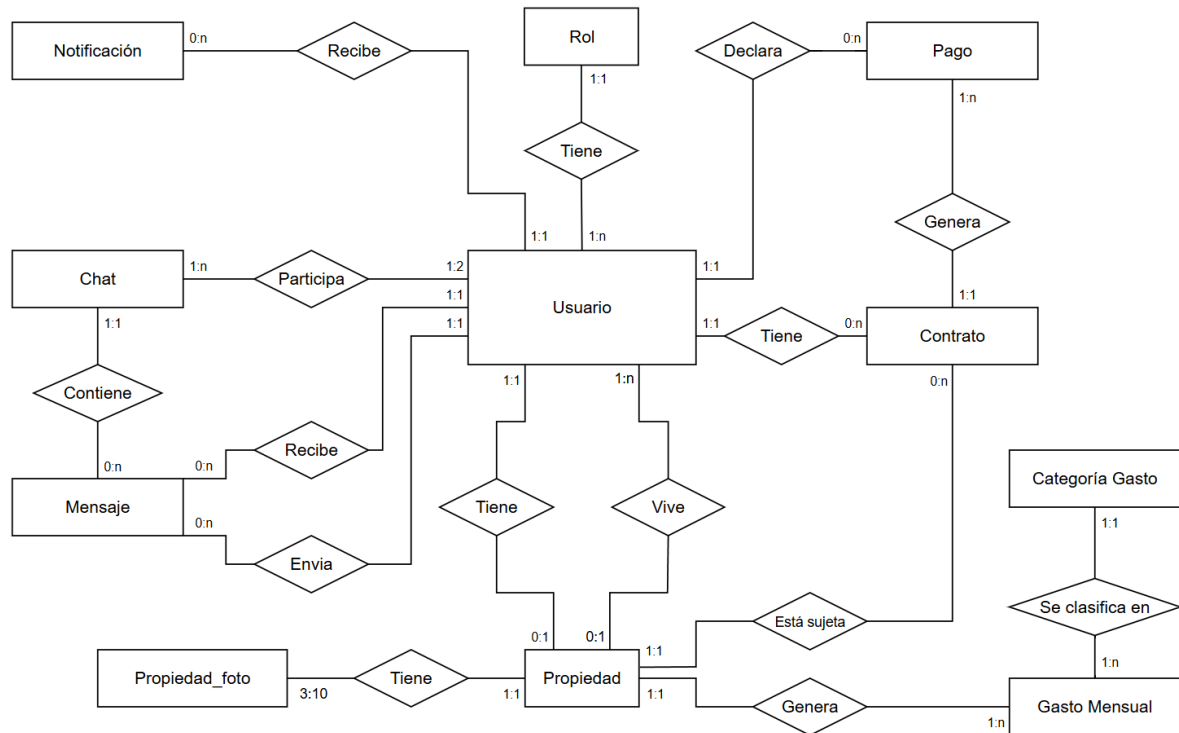


Figura 2: Modelo Entidad-Relación del sistema

Sistema de Chat Bidireccional: El chat se diseñó con una cardinalidad específica de dos usuarios (1:2) para garantizar conversaciones privadas únicamente entre inquilino y anfitrión, evitando chats grupales que podrían generar confusión en personas mayores. Esta decisión también optimiza el rendimiento al simplificar las consultas de mensajes y reducir la complejidad de indexación, manteniendo el sistema eficiente incluso con el crecimiento de usuarios activos.

Gestión de Propiedades con Límites: Se estableció un límite de 3 a 10 fotos por propiedad para equilibrar la necesidad de mostrar información visual suficiente sin sobrecargar cognitivamente a los usuarios, siguiendo los principios de diseño identificados. Este límite también previene el crecimiento descontrolado del almacenamiento y mantiene tiempos de carga predecibles, contribuyendo a la eficiencia del sistema a medida que crece la cantidad de propiedades registradas.

Contratos Individualizados: Cada contrato se asocia a un solo inquilino para mantener claridad en los acuerdos y simplificar la gestión de pagos, evitando la complejidad de contratos grupales. La simplicidad de esta relación uno-a-uno también beneficia el rendimiento del sistema al reducir la complejidad de las consultas y facilitar futuras optimizaciones de base de datos.

Sistema de Notificaciones Centralizado: Se optó por una tabla dedicada de notificaciones para mantener un registro persistente y permitir diferentes tipos de alertas según los eventos del sistema, reduciendo la necesidad de navegación compleja para usuarios con menor experiencia. La estructura permite implementar políticas de retención y optimizaciones futuras para mantener el rendimiento del sistema conforme aumente el volumen de notificaciones.

Categorización de Gastos: Se optó por una tabla independiente para categorías de gasto que actualmente maneja gastos comunes generales, pero se implementó con el objetivo de preparar el modelo para futuras categorizaciones más específicas sin requerir cambios estructurales en el esquema de base de datos.

3.4.2. Políticas de seguridad (RLS)

El sistema implementa Row Level Security (RLS), un mecanismo de control de acceso que permite definir políticas que determinan qué filas pueden ser devueltas por consultas o modificadas por comandos de manipulación de datos en función del usuario que ejecuta la consulta (PostgreSQL, 2024). RLS proporciona control granular sobre los datos, permitiendo que múltiples usuarios accedan a la misma tabla mientras ven únicamente las filas para las cuales tienen autorización, lo que resulta especialmente útil en aplicaciones multi-usuario como la plataforma desarrollada.

Las políticas específicas implementadas por tabla son las siguientes:

Operación	Política de acceso
SELECT	Usuarios autenticados pueden ver todos los perfiles
INSERT	Registro público para nuevos usuarios
UPDATE	Usuarios pueden modificar solo sus propios datos sin cambiar su rol
DELETE	Operación restringida para preservar integridad del sistema

Cuadro 12: Políticas de seguridad RLS para la tabla Usuario

Las políticas de la tabla Usuario permiten registro para nuevos usuarios y visibilidad completa de perfiles para usuarios autenticados, mientras que las actualizaciones están restringidas a datos propios del usuario.

Operación	Política de acceso
SELECT	Usuarios autenticados pueden ver todas las propiedades
INSERT	Solo inquilinos pueden registrar como maximo 1 propiedad
UPDATE	Administradores pueden actualizar estado de aprobación
DELETE	Administradores pueden eliminar cualquier propiedad; anfitriones pueden eliminar solo su propiedad rechazadas

Cuadro 13: Políticas de seguridad RLS para la tabla Propiedad

Las políticas de la tabla Propiedad implementan un flujo de aprobación donde inquilinos pueden registrar propiedades para convertirse en anfitriones, los administradores controlan el proceso de aprobación, y los anfitriones tienen permitido eliminar propiedad rechazada para corregir errores de registro.

Operación	Política de acceso
SELECT	Acceso limitado a participantes del chat específico
INSERT	Acceso limitado a participantes del chat específico
UPDATE	Acceso limitado a participantes del chat específico
DELETE	Operación restringida para preservar historial de mensajes

Cuadro 14: Políticas de seguridad RLS para las tablas Chat y Mensaje

Las políticas de Chat y Mensaje priorizan la privacidad de las conversaciones limitando el acceso únicamente a los participantes involucrados, con prohibición de eliminación que preserve el historial completo de comunicaciones para referencia futura en caso de necesitarla.

Operación	Política de acceso
SELECT	Solo anfitriones, inquilinos involucrados y administradores pueden visualizar contratos
INSERT	Exclusivamente anfitriones pueden crear nuevos contratos de convivencia
UPDATE	Anfitriones pueden modificar contratos pendientes; inquilinos cambian el estado del acuerdo al responder al mismo
DELETE	Operación no permitida para preservar historial de contratos

Cuadro 15: Políticas de seguridad RLS para la tabla Acuerdo

Las políticas de la tabla Acuerdo implementan un flujo de estado donde únicamente los anfitriones pueden iniciar contratos de convivencia, los inquilinos pueden responder exclusivamente a invitaciones dirigidas a ellos, y ambas partes mantienen acceso de consulta junto con los administradores.

Operación	Política de acceso
SELECT	Inquilinos ven sus propias declaraciones; anfitriones ven y confirman pagos de sus propiedades; administradores acceso completo
INSERT	Solo inquilinos pueden declarar pagos para contratos donde participan
UPDATE	Anfitriones pueden confirmar pagos declarados; operación restringida para otros casos
DELETE	Operación restringida para preservar registro de pagos

Cuadro 16: Políticas de seguridad RLS para la tabla Pago

Las políticas de la tabla Pago garantizan que los inquilinos solo puedan declarar pagos relacionados con sus contratos vigentes, mientras que los anfitriones pueden consultar y confirmar todas las declaraciones asociadas a sus propiedades, manteniendo la transparencia necesaria sin comprometer la privacidad de datos entre diferentes comunidades.

Operación	Política de acceso
SELECT	Anfitriones pueden ver gastos de sus propiedades; inquilinos con contratos vigentes pueden consultar gastos de su propiedad
INSERT	Tanto anfitriones como inquilinos con contratos vigentes pueden registrar gastos en sus respectivas propiedades
UPDATE	Operación restringida para mantener integridad del historial de gastos
DELETE	Operación restringida para preservar registro completo de gastos

Cuadro 17: Políticas de seguridad RLS para la tabla Gasto Mensual

Las políticas de Gasto Mensual permiten que tanto anfitriones como inquilinos registren gastos específicos individuales en sus propiedades. En la versión actual del sistema esta funcionalidad no está implementada en la interfaz, pero las políticas están preparadas para cuando se requiera un sistema más detallado donde usuarios puedan ingresar gastos específicos individuales en futuras versiones de la plataforma.

3.5. Arquitectura del Sistema

Con los requerimientos y el diseño de la base de datos establecidos, se presenta la arquitectura del sistema utilizando el modelo C4. Este modelo permite visualizar la arquitectura de software en diferentes niveles de abstracción.

3.5.1. Contenedores del sistema

El diagrama de la *Figura 3* muestra la arquitectura de contenedores del sistema.

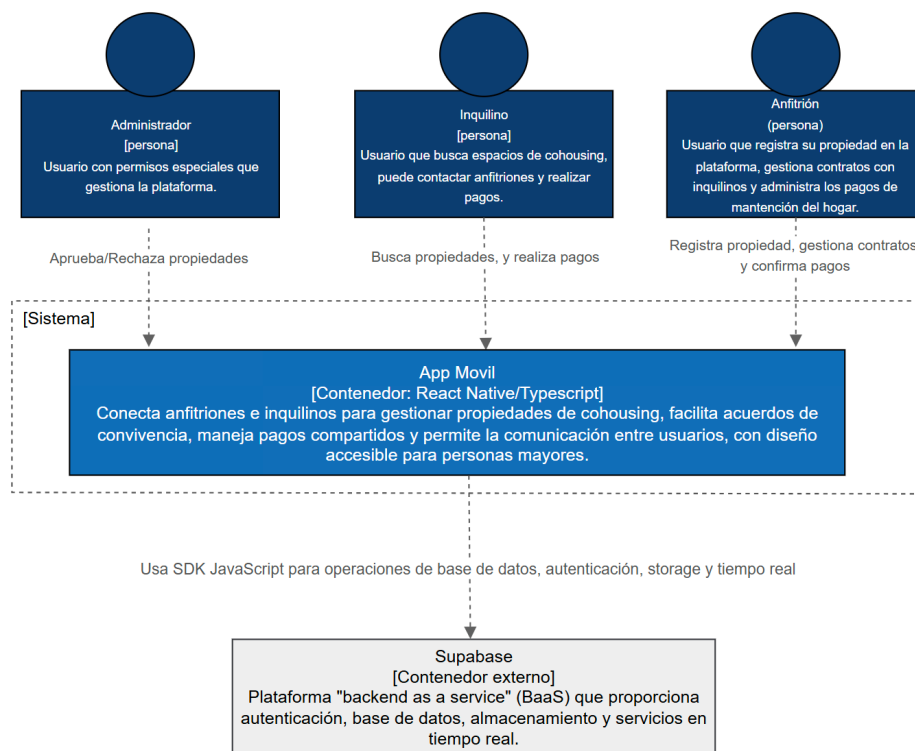


Figura 3: Diagrama de contenedores del sistema

El diagrama presenta tanto el contexto como los contenedores del sistema, los cuales resultan similares dada la simplicidad de la arquitectura. Los actores incluyen: Administrador con permisos de gestión de propiedades, Inquilinos que buscan espacios de vivienda compartida, y Anfitriones que ofrecen su propiedad para formar una comunidad. El sistema se compone de dos contenedores principales: la App Móvil, desarrollada en React Native con TypeScript, que implementa toda la lógica de presentación y adapta la interfaz según el rol del usuario; y Supabase como Backend as a Service (BaaS), que proporciona base de datos PostgreSQL, autenticación integrada, almacenamiento

de archivos y servicios en tiempo real. La comunicación entre contenedores se realiza mediante el SDK JavaScript de Supabase sobre HTTPS.

3.5.2. Componentes de la aplicación móvil

La **Figura 4** detalla los componentes internos de la aplicación móvil y sus interacciones.

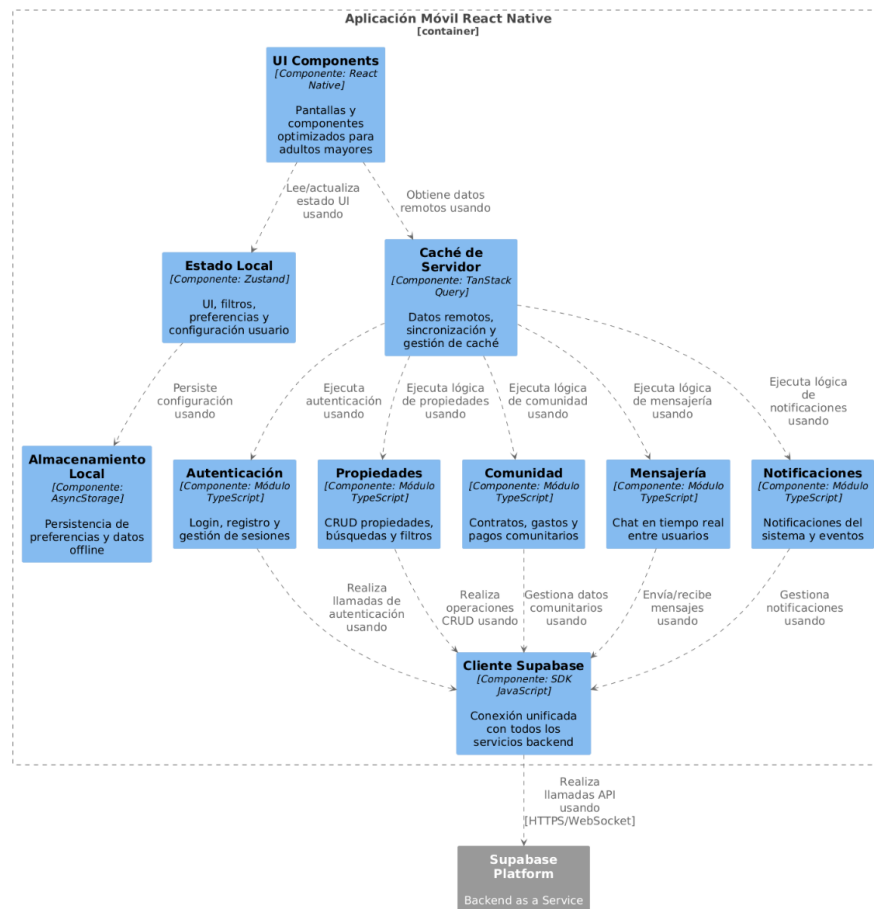


Figura 4: Diagrama de componentes de la aplicación móvil (C3)

La aplicación móvil se estructura internamente con componentes UI diseñados para personas mayores según las directrices mencionadas anteriormente. El estado se gestiona mediante Zustand para estados locales (preferencias, filtros, borradores) y TanStack Query para datos del servidor con caché inteligente. Los componentes de lógica de negocio incluyen: Autenticación para gestión de usuarios y roles, Propiedades para búsqueda y registro con flujo de aprobación, Comunidad para acuerdos y pagos, Mensajería con chat en tiempo real, y Notificaciones para alertas del sistema. El Cliente Supabase centraliza la comunicación con el backend mientras AsyncStorage persiste las preferencias del usuario. Los componentes UI solicitan datos mediante hooks que ejecutan la lógica de negocio y

se comunican con Supabase, recibiendo respuestas que actualizan automáticamente la interfaz.

Capítulo 4

Implementación

4.1. Implementación

El desarrollo de la plataforma se realizó utilizando un enfoque centrado en el usuario mayor, aplicando las directrices de usabilidad identificadas en el marco teórico. Esta sección presenta los flujos de interacción desde la perspectiva de experiencia de usuario, complementándolos con los aspectos técnicos relevantes de la implementación.

4.1.1. Selección de Tecnologías

Para la implementación de la plataforma se evaluaron diferentes alternativas tecnológicas, seleccionando aquellas que mejor se alineaban con los requerimientos del proyecto y las capacidades del equipo de desarrollo.

Framework de Desarrollo Móvil

Se analizaron dos opciones principales para el desarrollo móvil multiplataforma:

React Native (seleccionado): Se optó por React Native debido a su sintaxis basada en JavaScript, lenguaje previamente conocido, y su integración con Expo SDK que simplifica el acceso a funcionalidades nativas del dispositivo.

Flutter (descartado): Aunque Flutter ofrece excelente rendimiento, se descartó debido a la necesidad de aprender Dart como lenguaje adicional y la menor familiaridad con su ecosistema de herramientas.

Backend as a Service (BaaS)

Para la infraestructura del backend se compararon dos servicios principales:

Supabase (seleccionado): La elección de Supabase se fundamentó en su base de datos PostgreSQL relacional, que facilita el modelado de relaciones entre entidades del sistema. Las políticas RLS (Row Level Security) nativas proporcionaron seguridad a nivel de fila sin código adicional, mientras que las funciones en tiempo real facilitaron la implementación del sistema de mensajería.

Firebase (descartado): Aunque Firebase ofrece excelente integración con Google y escalabilidad automática, se optó por una solución de base de datos relacional en lugar de su enfoque NoSQL para mejor manejo de las relaciones entre entidades del sistema.

Gestión de Estado

La arquitectura de estado implementó una separación clara entre datos remotos y estado local:

TanStack Query: Seleccionado para gestión de datos del servidor por su sistema de caché inteligente que reduce consultas redundantes al backend de Supabase, reintentos automáticos ante fallos de conexión y sincronización optimista que mejora la experiencia de usuario.

Zustand: Elegido para estado local de UI por su arquitectura simple que evita re-renders innecesarios y facilita la separación entre estado persistente y temporal de la aplicación.

4.1.2. Consideraciones Técnicas Transversales

Los flujos implementados comparten elementos técnicos comunes que fundamentan su funcionamiento:

Sistema de Notificaciones Dual: La plataforma utiliza alertas inmediatas (toast) para usuarios activos y notificaciones persistentes para revisión posterior, garantizando que ningún evento importante pase desapercibido.

Persistencia Automática: Formularios y estados se preservan automáticamente mediante Zustand, evitando pérdida de información ante navegación accidental o cierre de la aplicación.

Gestión de Datos: TanStack Query maneja el caché de datos del servidor con actualización automática cuando los datos cambian, mientras WebSockets de Supabase

proporcionan actualizaciones en tiempo real.

Validaciones Multicapa: Cada flujo implementa verificación de formato en el cliente, validación de completitud antes del envío, y verificación final en el servidor contra las políticas RLS de la base de datos.

4.1.3. Flujo de Autenticación

La *Figura 5* presenta el mapa de navegación del flujo de autenticación implementado en la plataforma.

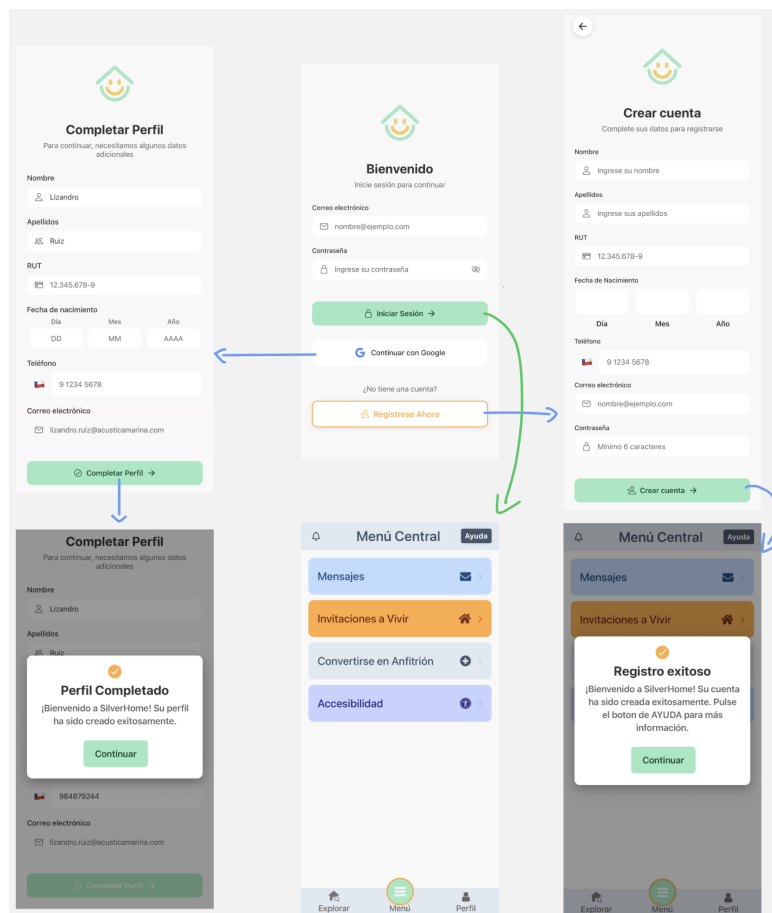


Figura 5: Mapa de navegación del flujo de autenticación

El flujo inicia con una pantalla de bienvenida que ofrece dos caminos: iniciar sesión o registrarse. Para iniciar sesión, el usuario ingresa correo y contraseña. Para registrarse, el usuario completa sus datos personales básicos en un formulario que incluye nombre, apellidos, RUT, fecha de nacimiento, teléfono y credenciales de acceso. Como alternativa a ambos procesos, existe la opción “Continuar con Google” que funciona tanto para

usuarios nuevos como existentes: si el usuario ya tiene cuenta, ingresa directamente al menú principal; si es nuevo, el sistema lo detecta y lo dirige a completar sus datos personales faltantes. Una característica particular del formulario es el auto-avance entre campos que detecta cuando un input contiene información válida y automáticamente enfoca el siguiente campo. Los datos del formulario persisten temporalmente para evitar pérdida de información si el usuario navega hacia atrás o cierra la plataforma sin querer. Tras completar el registro, aparece un modal de confirmación y el sistema dirige al usuario al menú principal adaptado a su rol específico.

La arquitectura de autenticación utiliza Supabase Auth como servicio base, separando el estado de sesión del perfil de usuario mediante stores independientes. Los tokens de acceso se renuevan automáticamente y mantienen la sesión activa entre usos de la plataforma.

4.1.4. Flujo de Registro de Propiedad

La **Figura 6** muestra el proceso mediante el cual los usuarios inquilinos pueden registrar su propiedad para convertirse en anfitriones.

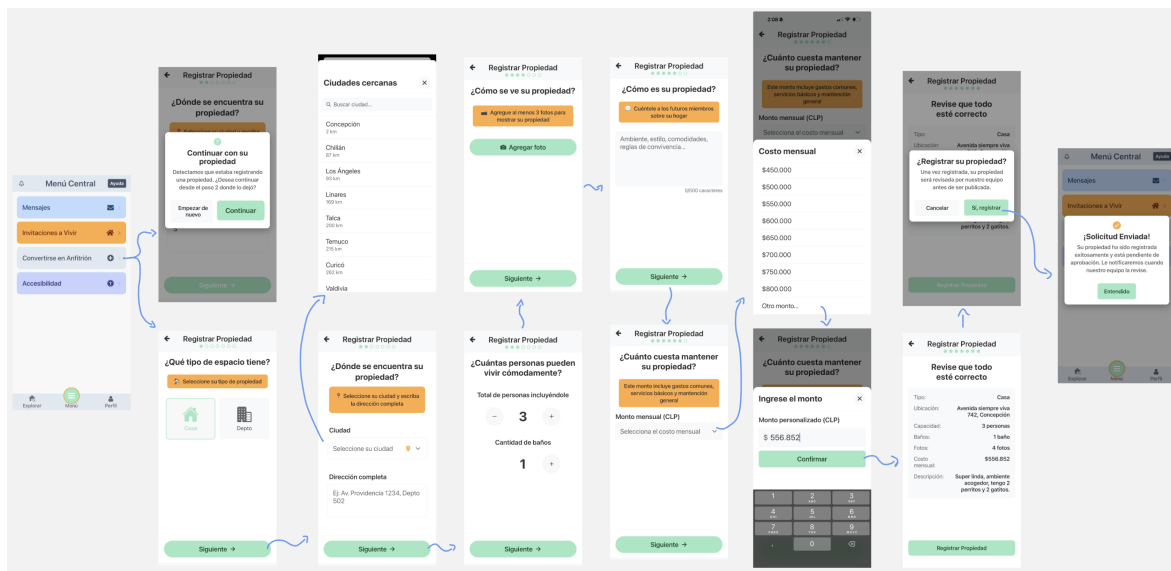


Figura 6: Mapa de navegación del flujo de registro de propiedad

El flujo se inicia desde el menú principal con la opción “Convertirse en Anfitrión” y se estructura en siete pasos secuenciales. El primer paso solicita seleccionar el tipo de propiedad entre casa o apartamento. El segundo paso requiere completar la ubicación, incluyendo una lista de ciudades predefinidas y un campo de dirección completa. El

tercer paso define la capacidad habitacional, especificando el número total de personas que pueden vivir en la propiedad y la cantidad de baños disponibles. El cuarto paso gestiona la subida de fotografías, requiriendo un mínimo de tres imágenes para mostrar la propiedad. El quinto paso solicita una descripción textual de al menos 50 caracteres. El sexto paso establece el costo mensual de mantención, ofreciendo opciones predefinidas o ingreso personalizado. Finalmente, el séptimo paso presenta un resumen completo para revisión antes del envío.

La subida de fotografías, ilustrada en la **Figura 7**, prioriza botones de gran tamaño y opciones claras entre galería y cámara. Las miniaturas se presentan con dimensiones generosas para facilitar la identificación, mientras que la organización utiliza botones con nombres intuitivos adaptados al usuario mayor.

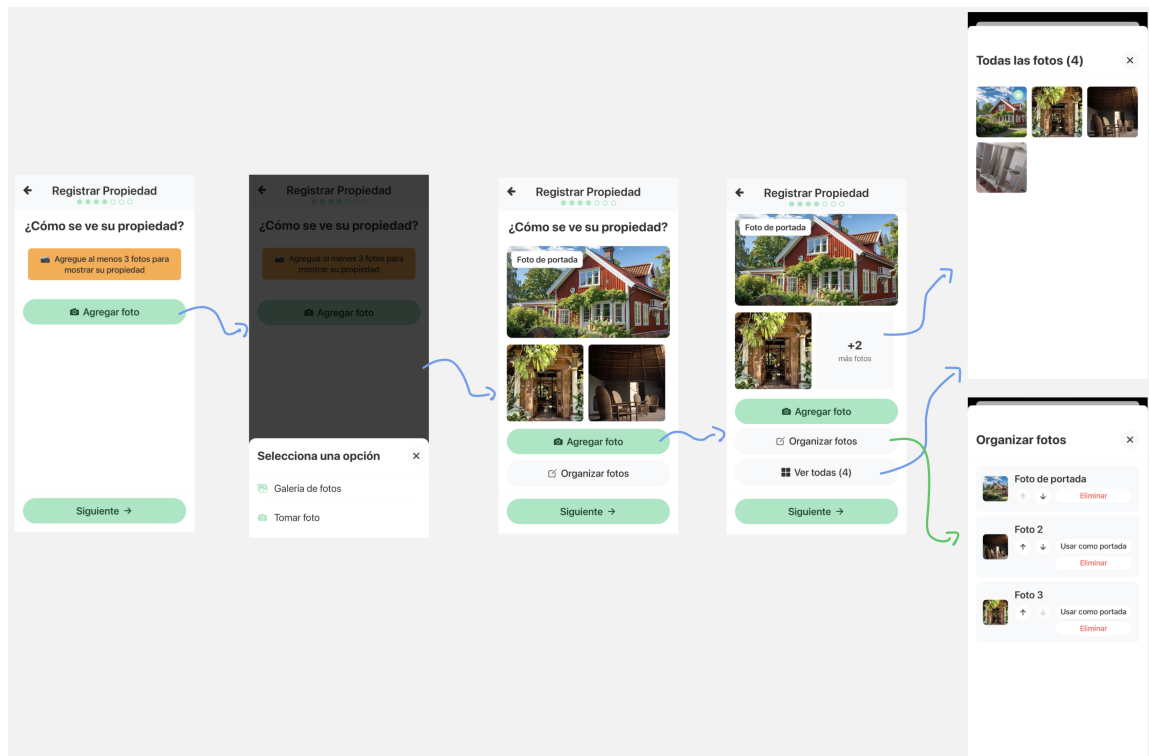


Figura 7: Mapa de navegación: Subida de fotografías adaptado para personas mayores

Cada paso incluye validaciones que impiden avanzar sin completar la información requerida. La subida de fotografías maneja compresión automática y almacenamiento en Supabase Storage, con actualización inmediata de la interfaz. Una vez enviado, la propiedad se registra con estado “pendiente”, aparece un modal de confirmación, y se notifica al administrador para análisis y decisión.

Cuando un administrador aprueba la propiedad, se activa el flujo de notificación ilustrado

en la **Figura 8**. Las notificaciones resultantes muestran un mensaje de felicitaciones y actualizan simultáneamente el estado del menú central.

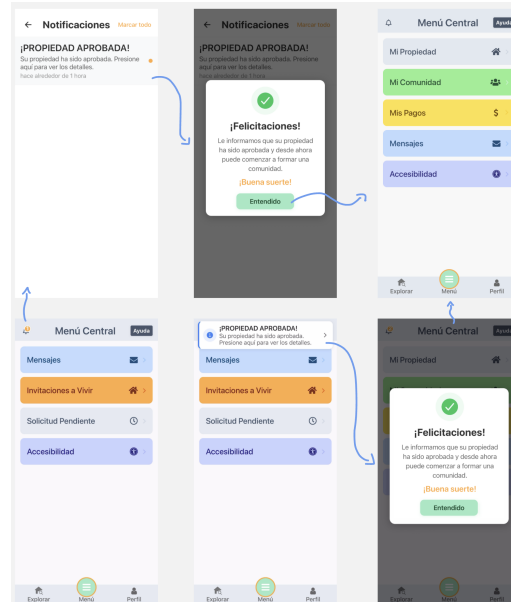


Figura 8: Mapa de navegación: Notificación de propiedad aprobada

Al recibir la notificación de aprobación, la plataforma actualiza automáticamente el caché global, provocando el re-renderizado inmediato del menú central. Este proceso refleja el nuevo rol del usuario como anfitrión, habilitando todas las funcionalidades correspondientes a este perfil.

4.1.5. Flujo de Formación de Comunidad

El proceso de formación de una comunidad representa el núcleo funcional de la plataforma, permitiendo que inquilinos y anfitriones establezcan contratos de convivencia. Este flujo completo se divide en cuatro mapas de navegación que documentan desde la búsqueda inicial hasta la aceptación formal del contrato.

Búsqueda y Primer Contacto

La **Figura 9** presenta el proceso inicial donde un inquilino busca una comunidad y establece el primer contacto con un anfitrión.

El flujo inicia con el usuario inquilino explorando propiedades desde la pestaña “Explorar” del menú principal, donde visualiza únicamente las propiedades aprobadas por los administradores. Al seleccionar una propiedad específica, el sistema navega a la vista de detalles completos, presentando información detallada incluyendo fotografías,

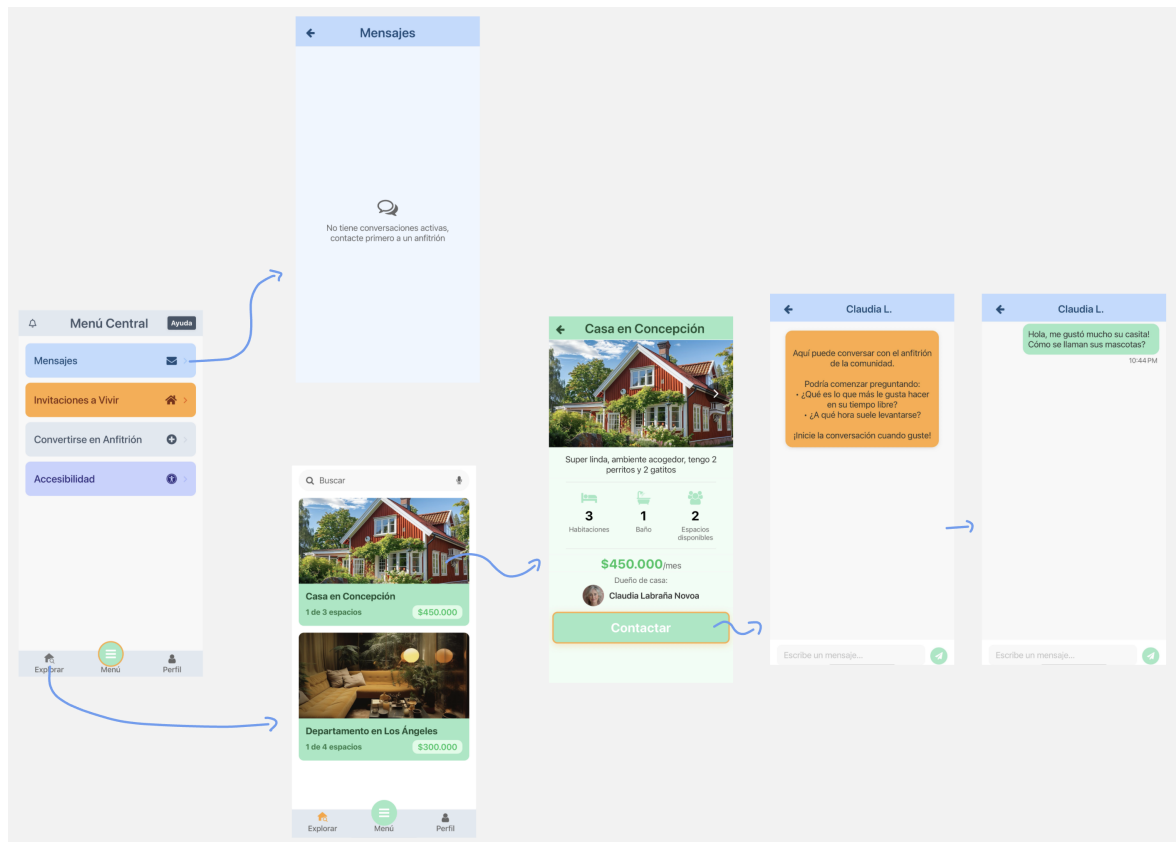


Figura 9: Mapa de navegación: Búsqueda y primer contacto

descripción, ubicación y características del espacio. En esta vista, el usuario puede contactar directamente al anfitrión mediante el botón “Contactar”, acción que genera automáticamente un chat entre ambos usuarios. El sistema contextualiza inmediatamente al inquilino con un mensaje de apoyo que facilita el inicio de la conversación, permitiendo que comience a intercambiar información con el anfitrión sobre dudas específicas de la convivencia.

La creación del chat verifica automáticamente si existe una conversación previa entre los usuarios y, en caso contrario, establece una nueva vinculada a la propiedad específica. El acceso se controla validando permisos de inquilino antes de permitir el contacto directo.

Notificación y Respuesta del Anfitrión

La **Figura 10** ilustra el proceso de notificación al anfitrión y el desarrollo de la conversación bidireccional.

El mapa muestra el ejemplo de un anfitrión que abre la plataforma 2 horas después de haber recibido un mensaje. El proceso continúa cuando el anfitrión observa que

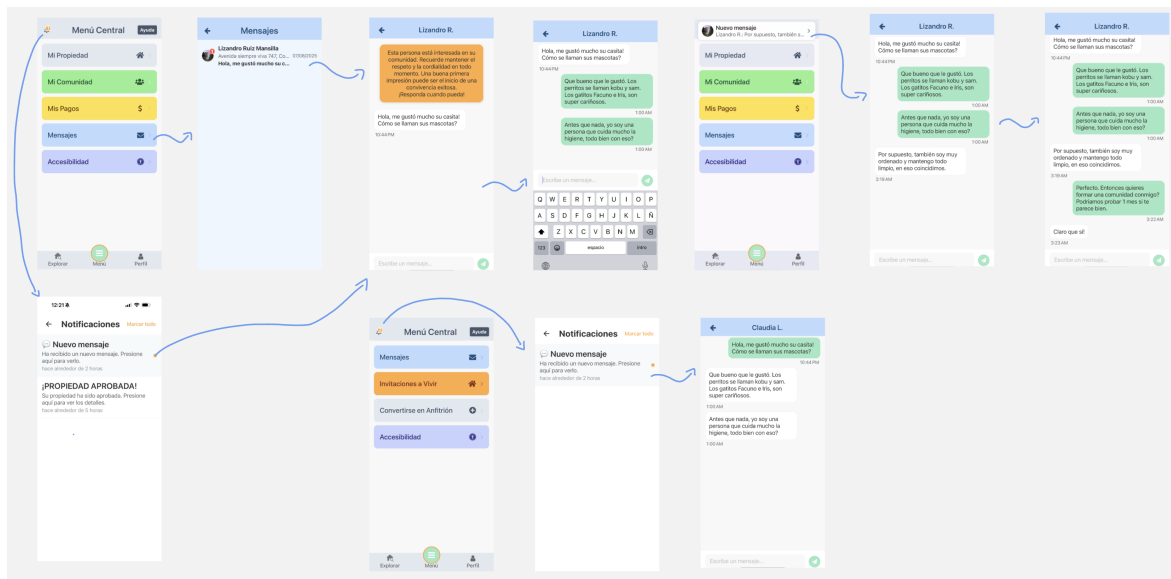


Figura 10: Mapa de navegación: Notificación y respuesta del anfitrión

tiene un nuevo mensaje y se dispone a responderlo. Al acceder a la notificación desde el menú de campana, el sistema redirige automáticamente al chat donde fue contactado por el inquilino. Similar al flujo anterior, el sistema contextualiza al anfitrión con un mensaje informativo, proporcionando orientación sobre cómo responder al nuevo contacto. Paralelamente, cuando el inquilino también ingresa a la aplicación después de un tiempo prolongado, recibe igualmente notificaciones persistentes de mensajes no leídos. Cuando cualquier usuario está activo en la aplicación y recibe un nuevo mensaje, se le muestra inmediatamente una notificación toast temporal, mientras que ambos tipos de usuario tienen acceso tanto a notificaciones persistentes como temporales según el contexto de uso.

Ambos tipos de usuarios pueden acceder posteriormente a sus conversaciones desde la pestaña “Mensajes” del menú central, donde visualizan una lista organizada de todos sus chats activos. El flujo culmina cuando ambos usuarios, tras intercambiar información y resolver sus dudas, deciden proceder con el proceso de convivencia.

Los listeners globales gestionan las notificaciones entrantes, diferenciando entre alertas temporales y persistentes según el contexto. Las alertas toast se configuraron para mostrarse durante 7 segundos.

Creación de Invitación por el Anfitrión

La **Figura 11** documenta el proceso mediante el cual el anfitrión formaliza la invitación de convivencia.

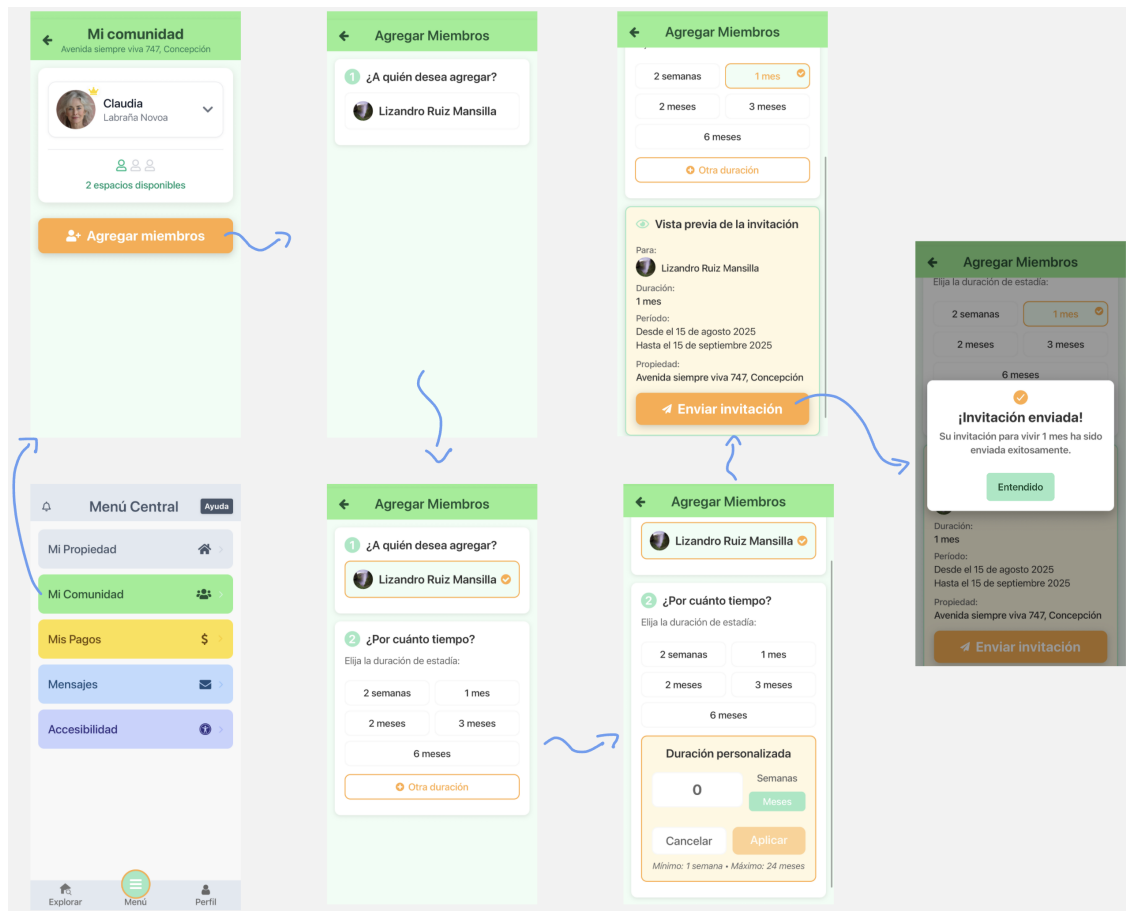


Figura 11: Mapa de navegación: Creación de invitación por el anfitrión

Una vez establecido el acuerdo verbal, el anfitrión procede a formalizar la invitación accediendo al menú “Mi Comunidad” y seleccionando la opción “Agregar Miembros”. El sistema presenta una vista donde puede seleccionar específicamente a las personas con quienes mantiene conversaciones activas, mostrando únicamente inquilinos con los cuales no existe un contrato vigente para evitar duplicaciones. Tras seleccionar al inquilino deseado, el anfitrión debe establecer la duración de la estadía utilizando una interfaz optimizada para personas mayores que ofrece botones preestablecidos con duraciones comunes o la opción de establecer una duración personalizada mediante un selector simple que acepta valores en semanas o meses.

Posteriormente se solicita revisar el tiempo de estadía, estableciendo automáticamente la fecha de inicio una semana después para permitir preparativos. Una vez confirmados todos los detalles, el anfitrión puede enviar la invitación, recibiendo retroalimentación inmediata mediante un modal de confirmación.

La implementación filtra automáticamente los chats para mostrar únicamente inquilinos

disponibles para nuevos contratos. El selector de duración utiliza botones de gran tamaño con separación generosa, mientras que la creación del contrato valida fechas, calcula períodos y verifica permisos antes de la confirmación.

Aceptación de Invitación y Formación de Comunidad

La **Figura 12** presenta la culminación del proceso con la respuesta del inquilino y la formalización de la comunidad.

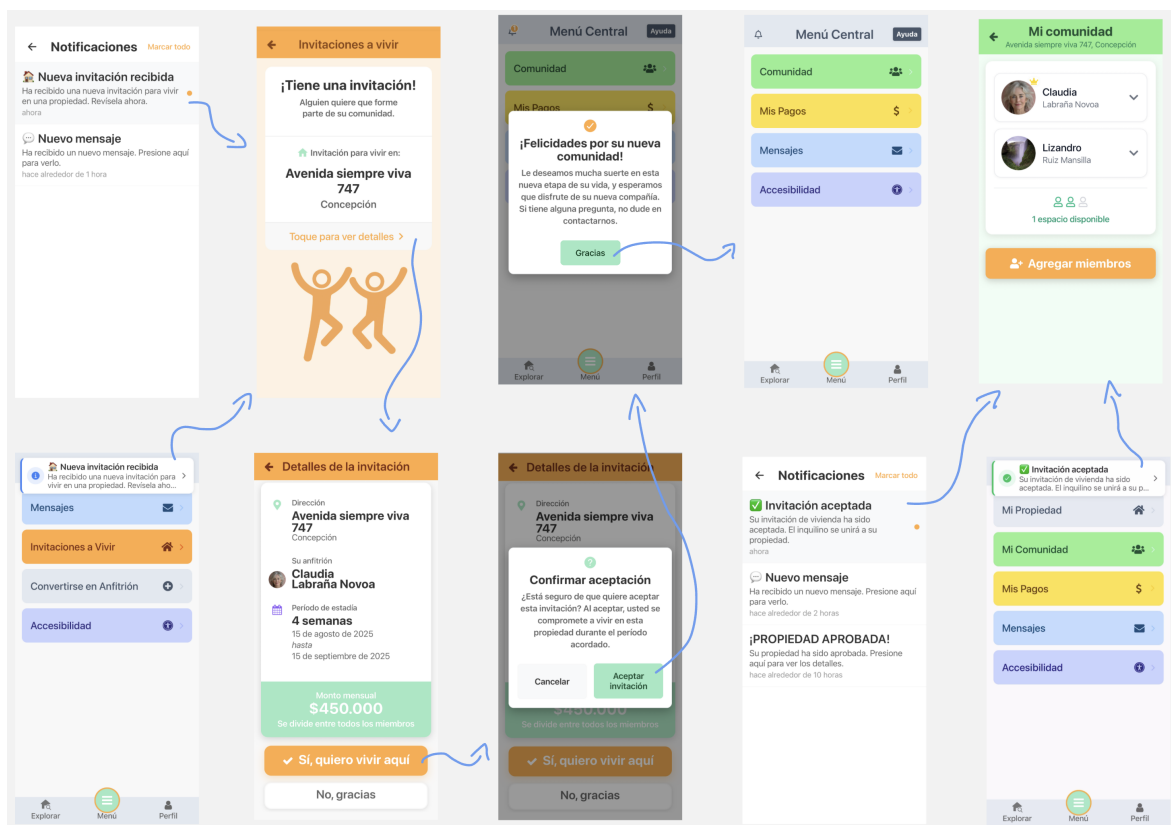


Figura 12: Mapa de navegación: Aceptación de invitación y formación de comunidad

El proceso final inicia cuando el inquilino recibe una notificación sobre la invitación recibida, ya sea mediante notificación push o persistente. Al acceder a cualquiera de estas notificaciones, el sistema redirige a la vista “Invitaciones a Vivir” donde puede revisar todos los detalles de la propuesta, incluyendo duración, fechas y condiciones específicas. Tras analizar la información, el inquilino puede responder presionando el botón “Sí, quiero vivir aquí”, momento en que el sistema solicita revalidar la decisión mediante un modal de confirmación adicional para evitar aceptaciones accidentales.

Una vez confirmada la aceptación, el sistema presenta un mensaje de bienvenida a la comunidad y actualiza instantáneamente el estado del menú central, habilitando todas

las funcionalidades correspondientes a un inquilino perteneciente a una comunidad activa. Simultáneamente, el anfitrión recibe una notificación sobre la aceptación de su invitación, y al presionar dicha notificación es redirigido a la vista “Mi Comunidad” donde puede ver a su nuevo compañero en la comunidad y consultar su número de teléfono al presionar sobre su tarjeta.

La plataforma controla que los inquilinos solo puedan tener una invitación pendiente a la vez, impidiendo recibir nuevas hasta responder la actual. Al aceptar una invitación, se actualiza el estado del contrato e incrementa el contador de ocupantes, bloqueando automáticamente futuras invitaciones.

4.1.6. Flujo de Vida en Comunidad

Una vez formada la comunidad, los usuarios acceden a funcionalidades básicas para coordinar su convivencia. La vista “Mi Comunidad” muestra contactos de compañeros y fotografías de la propiedad compartida. La gestión de pagos funciona como un registro simplificado donde los inquilinos declaran sus aportes mensuales y los anfitriones confirman la recepción mediante simples toques en pantalla.

Declaración de Pagos por Inquilinos

La **Figura 13** presenta el proceso mediante el cual los inquilinos gestionan sus obligaciones financieras dentro de la comunidad.

El mapa muestra inicialmente la vista “Mi Comunidad” donde los usuarios pueden consultar información de contacto de sus compañeros, incluyendo números de teléfono y fotografías de perfil, además de visualizar imágenes de la propiedad compartida. Esta vista proporciona el contexto de la comunidad activa donde se realizarán las declaraciones de pago.

El flujo de declaración inicia cuando el inquilino decide registrar el pago de su parte de los gastos comunes de la comunidad. Durante los primeros 3 a 4 días de cada mes, el sistema envía notificaciones toast automáticas a aquellos inquilinos que aún no han declarado su pago, recordándoles que deben realizar su aporte el día 5 del mes. Al acceder a “Mis Pagos” desde el menú central, ya sea por iniciativa propia o mediante la notificación recordatoria, el usuario visualiza el monto que le corresponde aportar, calculado automáticamente dividiendo el costo de mantención mensual de la propiedad entre todos los miembros activos de la comunidad, y puede proceder con la declaración

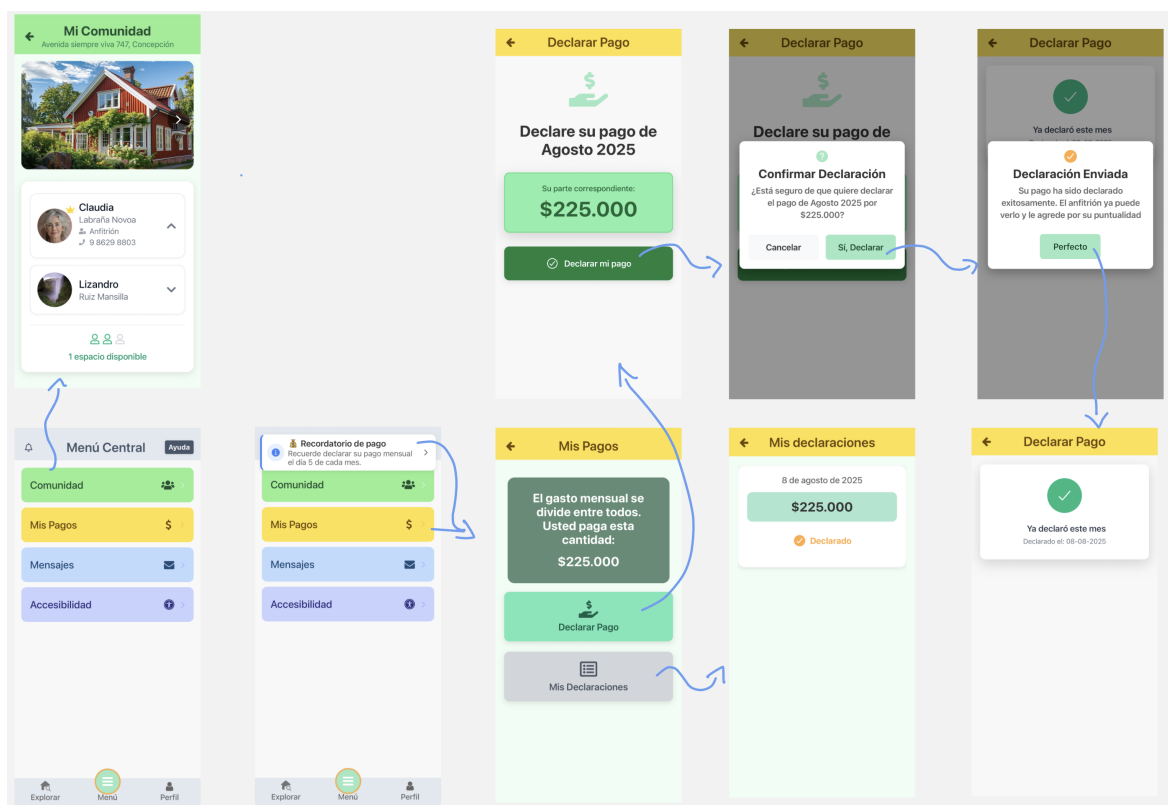


Figura 13: Mapa de navegación: Declaración de pagos por inquilinos

presionando el botón “Declarar Pago”. El proceso continúa con la declaración de haber realizado el pago, seguida de una confirmación de la declaración y finalmente un mensaje de feedback de éxito. La declaración queda registrada con estado “declarado”, indicando que el inquilino reporta haber efectuado el pago y está pendiente de que el anfitrión confirme haberlo recibido.

Adicionalmente, la vista “Mis Declaraciones” permite al inquilino consultar de forma simple todo el historial de pagos que ha declarado para su comunidad, mostrando fechas y montos en un formato claro y accesible para el usuario mayor.

Confirmación de Pagos por Anfitriones

La **Figura 14** ilustra el proceso de confirmación de pagos desde la perspectiva del anfitrión.

El flujo inicia cuando el anfitrión recibe una notificación sobre un nuevo pago declarado por algún miembro de su comunidad. Al presionar la notificación, la plataforma redirige directamente a la vista de confirmación donde puede revisar los detalles de la declaración y proceder a confirmarla, verificando que efectivamente recibió el pago correspondiente.

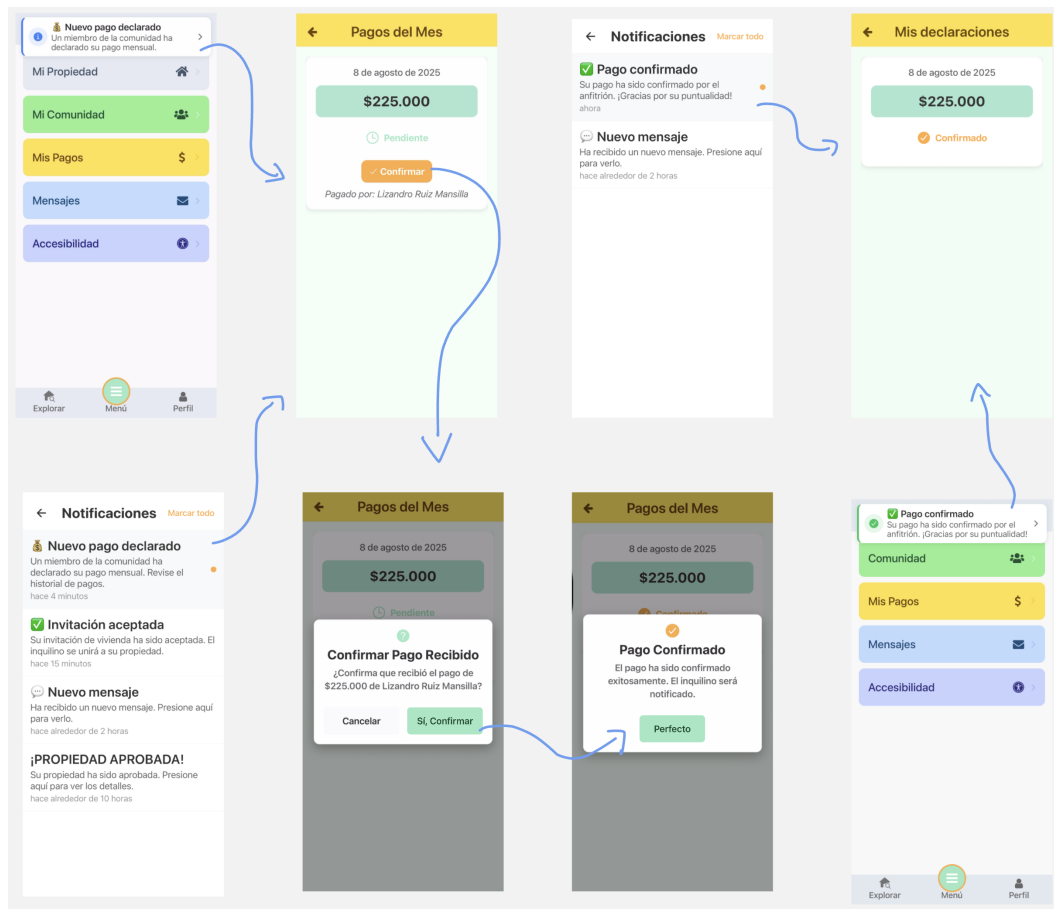


Figura 14: Mapa de navegación: Confirmación de pagos por anfitriones

Una vez que el anfitrión confirma haber recibido el pago, la plataforma actualiza automáticamente el estado de la declaración y notifica inmediatamente al inquilino correspondiente sobre la confirmación de recepción.

Este mecanismo funciona como un registro donde cada inquilino declara haber realizado su aporte mensual (todos pagan el mismo monto) y el anfitrión confirma haberlo recibido. Solo el anfitrión tiene visibilidad completa del estado de pagos de la comunidad, mientras que cada inquilino visualiza únicamente su propio historial de declaraciones.

4.1.7. Flujos Secundarios

Gestión de Foto de Perfil

La **Figura 15** ilustra el proceso de actualización de foto de perfil implementado en la plataforma.

Por defecto, todos los usuarios inician sin foto de perfil, mostrando sus iniciales en

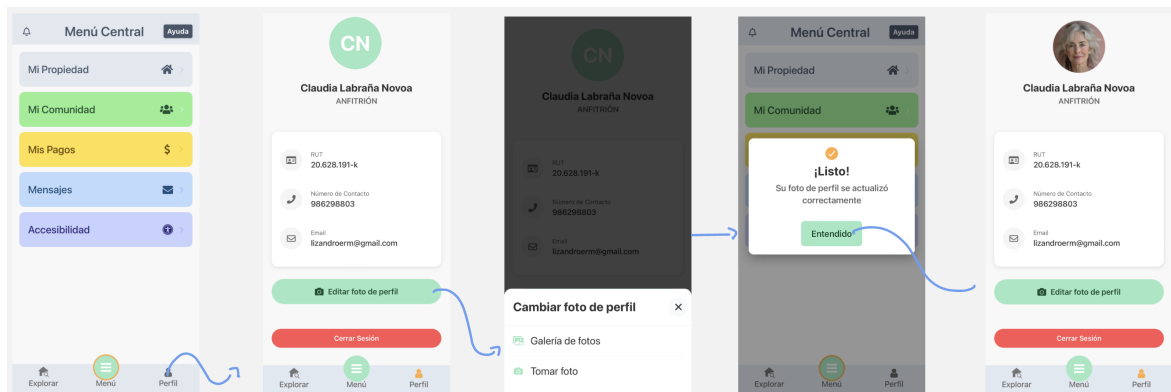


Figura 15: Mapa de navegación: Gestión de foto de perfil

un círculo de color como identificador visual en lugar de una imagen. El flujo inicia cuando el usuario accede a la pestaña “Perfil” desde el menú central, donde visualiza su información personal. En esta vista, el botón “Editar foto de perfil” permite iniciar el proceso de actualización de imagen.

Al presionar el botón de edición, aparece un modal con dos opciones claramente diferenciadas: “Galería de fotos” para seleccionar una imagen existente del dispositivo, y “Tomar foto” para capturar una nueva imagen con la cámara. Una vez que el usuario selecciona su foto mediante cualquiera de las dos opciones disponibles, la plataforma procesa la imagen y presenta un mensaje de éxito confirmando que “Su foto de perfil se actualizó correctamente”, proporcionando retroalimentación inmediata sobre la acción realizada.

El modal de selección utiliza el mismo patrón de interfaz implementado en la subida de fotografías de propiedad, manteniendo consistencia en la experiencia del usuario. El proceso completo se simplifica en 3 pasos: acceder al perfil, seleccionar método de imagen, y confirmar actualización.

Panel de Administración

La **Figura 16** muestra la interfaz del panel de administración implementado en la plataforma.

El panel de administración permite gestionar las solicitudes de propiedades pendientes. La interfaz muestra el total de propiedades pendientes y aprobadas, seguido de una lista con las propiedades por revisar, incluyendo datos del anfitrión, características del espacio y fotografías.

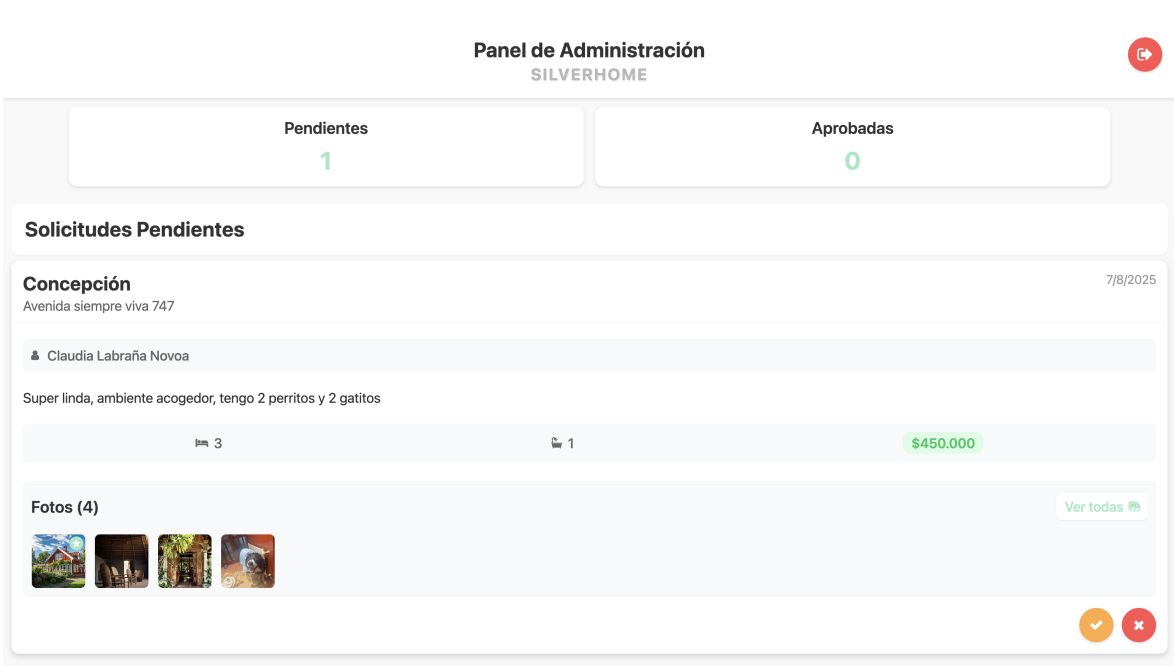


Figura 16: Mapa de navegación: Panel de administración para gestión de propiedades

Cada propiedad puede aprobarse directamente con un botón verde o rechazarse con un botón rojo. El proceso de rechazo requiere ingresar una razón obligatoria que se enviará al anfitrión como retroalimentación. Ambas acciones notifican automáticamente al anfitrión y actualizan el estado de la propiedad en el sistema.

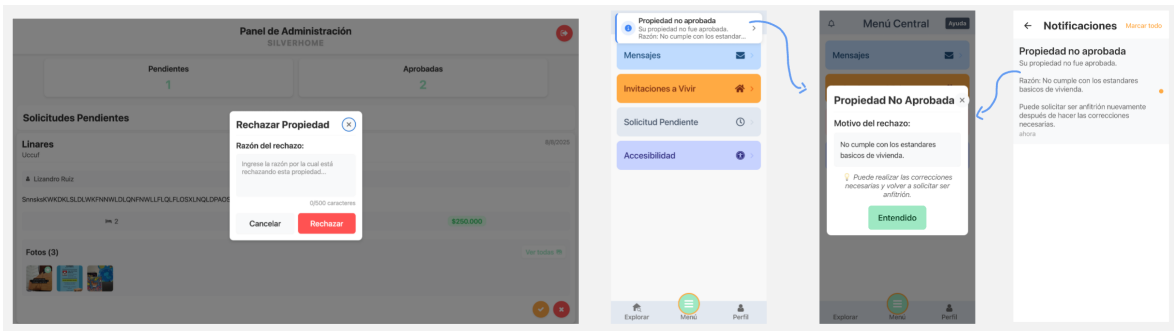


Figura 17: Mapa de navegación: Flujo de rechazo de propiedad

Sistema de Ayuda Contextual

La *Figura 18* presenta los diferentes tours virtuales implementados según el contexto del usuario.

La plataforma incluye un tour virtual accesible desde el botón “Ayuda” del menú central que se adapta al rol y estado del usuario, explicando qué puede hacer en cada sección disponible. La imagen muestra los tres tipos de tour: inquilinos sin comunidad,

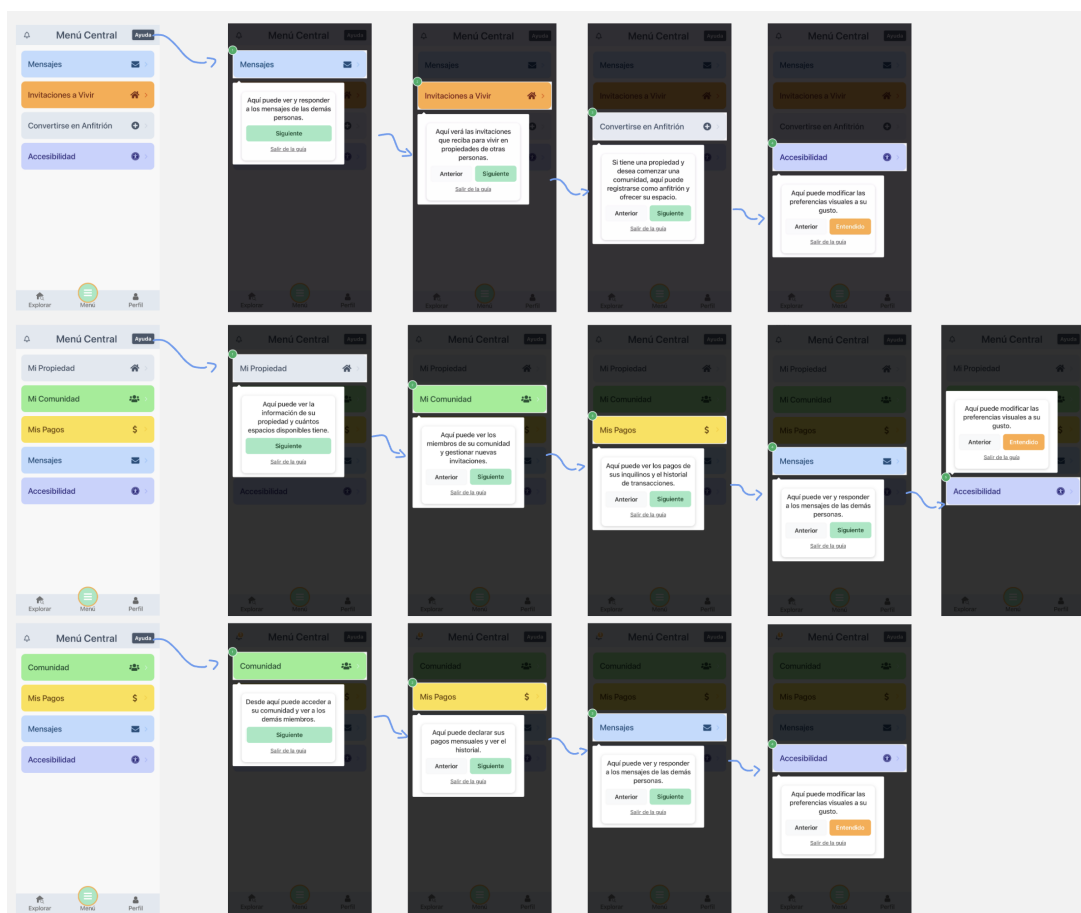


Figura 18: Tours virtuales contextuales por tipo de usuario

anfitriones, e inquilinos con comunidad activa, cada uno contextualizado para sus funciones específicas. La navegación utiliza botones simples de “Siguiente” y “Anterior” con texto claro y directo.

Todos los flujos documentados se implementaron aplicando consistentemente las directrices de usabilidad para personas mayores identificadas en el marco teórico. La interfaz completa respeta las dos reglas de oro fundamentales: simplificación máxima con máximo dos conceptos por pantalla, y elementos de gran tamaño con separación generosa entre controles interactivos.

La arquitectura de navegación mantiene máximo tres niveles de profundidad entre ideas principales, evitando que los usuarios se pierdan en estructuras complejas. Adicionalmente, todos los botones implementan seguridad anti-doble presión, bloqueándose temporalmente tras el primer toque para prevenir inconsistencias causadas por temblores involuntarios, característica especialmente relevante para usuarios mayores con menor precisión motriz.

Capítulo 5

Verificación y validación

Para evaluar la plataforma desarrollada, se implementó un proceso de verificación y validación que comprobó el funcionamiento del sistema y su usabilidad para el público objetivo. Esta fase del proyecto integró pruebas manuales con evaluación heurística, proporcionando información sobre el estado de la plataforma.

5.1. Verificación

Se realizaron pruebas manuales sistemáticas para verificar que cada funcionalidad implementada cumpliera con los requerimientos establecidos. El proceso consistió en tomar cada caso de uso, dividirlo en funcionalidades específicas, y probar cada una con diferentes valores de entrada interactuando directamente con la aplicación.

5.1.1. Metodología de Verificación

Para cada caso de uso documentado en el diseño, se aplicó el siguiente proceso:

1. **Implementación iterativa:** Se desarrolló cada funcionalidad específica del caso de uso.
2. **Diseño de pruebas:** Para cada funcionalidad se definieron valores de entrada normales, límite y erróneos.
3. **Ejecución manual:** Se probó cada funcionalidad interactuando con la aplicación en dispositivos móviles.

4. **Verificación de estabilidad:** Se validó que la plataforma respondiera como se esperaba sin caídas ni problemas de memoria.

Este enfoque permitió verificar tanto el comportamiento esperado como el manejo de casos excepcionales, asegurando que cada componente funcionara correctamente antes de integrarlo en flujos más complejos.

5.1.2. Proceso de Verificación y Resultados

El proceso de verificación evaluó sistemáticamente los casos de uso implementados para cada actor del sistema: *Buscar comunidad* y *Declarar pagos* para inquilinos, *Registrar propiedad* y *Crear contrato de convivencia* para anfitriones, y *Aprobar/rechazar propiedades* para administradores, junto con las funcionalidades transversales de *Enviar mensajes* compartida entre inquilinos y anfitriones. Para cada funcionalidad se establecieron criterios específicos: completitud funcional sin errores críticos, validación correcta de formularios con retroalimentación clara, persistencia adecuada de datos entre sesiones, entrega oportuna de notificaciones y tiempos de respuesta que garantizaran una fluida experiencia de usuario.

La estabilidad del sistema constituyó un aspecto fundamental de la verificación. Inicialmente, la gestión de caché y estado global se implementaba mediante código explícito que requería control directo de sincronización y persistencia de datos. Esta aproximación generaba código repetitivo con uso extensivo de `useEffect` y manejo manual de condiciones de carrera, resultando en implementaciones complejas donde las llamadas concurrentes a la API de Supabase provocaban memory leaks que comprometían la estabilidad de la aplicación.

La integración de TanStack Query para datos remotos y Zustand para estado local proporcionó mecanismos probados de manejo de caché y sincronización, estableciendo como criterio principal que la aplicación funcionara sin interrupciones, reinicios inesperados o colapsos durante su operación.

Los resultados confirmaron el correcto funcionamiento del sistema en todas las funcionalidades evaluadas. La gestión automática de caché resolvió los problemas de concurrencia identificados inicialmente, mientras que el manejo simplificado del estado local redujo considerablemente la complejidad del código. Durante todas las pruebas realizadas, el sistema mantuvo su estabilidad operacional cumpliendo con los criterios establecidos para cada funcionalidad verificada.

5.2. Validación

Para evaluar la usabilidad de la plataforma desarrollada, se realizó una evaluación heurística siguiendo la metodología descrita por Zaytseva (Zaytseva, 2024). Esta metodología permite identificar problemas de usabilidad mediante la inspección sistemática de la interfaz basándose en los 10 principios heurísticos de Jakob Nielsen.

5.2.1. Metodología

Se evaluaron las 10 heurísticas de Nielsen mediante 30 criterios específicos (3 por heurística), utilizando una escala de 1 a 5 donde 1 indica no cumplimiento y 5 excelente cumplimiento. Los criterios de evaluación fueron tomados directamente de la plantilla de evaluación heurística creada por Ana Rangel (Rangel, 2024), la cual proporciona preguntas específicas para cada principio heurístico. La evaluación fue realizada por 5 compañeros de carrera de Ingeniería Civil Informática con experiencia en desarrollo de software y familiaridad con principios de usabilidad.

Los criterios evaluados para cada heurística fueron los siguientes:

H1	Visibilidad del estado del sistema
P1.1	¿La interfaz muestra un indicador de carga o progreso cuando se está procesando algo?
P1.2	¿Recibe el usuario una confirmación o mensaje cuando una acción se completa exitosamente?
P1.3	¿Se destacan claramente los errores o problemas que ocurren en el sistema?

Cuadro 18: Criterios de evaluación para visibilidad del estado del sistema

H2	Relación entre el sistema y el mundo real
P2.1	¿Los términos y etiquetas utilizados en la app son comprensibles para el usuario promedio?
P2.2	¿Los iconos y elementos visuales representan objetos o acciones que los usuarios ya conocen?
P2.3	¿El flujo de navegación sigue una lógica natural, como en experiencias del mundo real?

Cuadro 19: Criterios de evaluación para relación con el mundo real

H3	Control y libertad del usuario
P3.1	¿Se pueden deshacer acciones importantes (como eliminar un archivo)?
P3.2	¿Los usuarios pueden cancelar un proceso en cualquier momento si cambiaron de opinión?
P3.3	¿Existe una opción de volver atrás o regresar a un estado anterior?

Cuadro 20: Criterios de evaluación para control y libertad del usuario

H4	Consistencia y estándares
P4.1	¿Los botones, menús y acciones funcionan y se ven de la misma manera en todas las secciones?
P4.2	¿Los colores y símbolos tienen el mismo significado en toda la app?
P4.3	¿Los elementos interactivos tienen un comportamiento predecible?

Cuadro 21: Criterios de evaluación para consistencia y estándares

H5	Prevención de errores
P5.1	¿El sistema evita que el usuario envíe formularios incompletos o con datos incorrectos?
P5.2	¿Se pide confirmación antes de acciones irreversibles?
P5.3	¿El sistema evita acciones destructivas sin confirmación?

Cuadro 22: Criterios de evaluación para prevención de errores

H6	Reconocimiento en lugar de recuerdo
P6.1	¿Los nombres de botones y menús son claros sin adivinar su función?
P6.2	¿El usuario puede ver opciones disponibles sin recordar dónde están?
P6.3	¿Existen hints o ejemplos visuales que ayuden a completar tareas?

Cuadro 23: Criterios de evaluación para reconocimiento sobre recuerdo

H7	Flexibilidad y eficiencia de uso
P7.1	¿Existen atajos o accesos rápidos para usuarios avanzados?
P7.2	¿El usuario puede personalizar opciones según sus necesidades?
P7.3	¿El sistema ofrece sugerencias para agilizar tareas repetitivas?

Cuadro 24: Criterios de evaluación para flexibilidad y eficiencia

H8	Diseño estético y minimalista
P8.1	¿La interfaz evita el exceso de texto o elementos innecesarios?
P8.2	¿Los elementos gráficos siguen una jerarquía clara?
P8.3	¿Los colores e imágenes ayudan a comprender la información?

Cuadro 25: Criterios de evaluación para diseño estético y minimalista

H9	Ayuda a reconocer y recuperarse de errores
P9.1	¿Los mensajes de error explican claramente qué pasó y cómo solucionarlo?
P9.2	¿Los errores permiten ser corregidos sin perder datos?
P9.3	¿El sistema sugiere soluciones cuando ocurre un problema?

Cuadro 26: Criterios de evaluación para recuperación de errores

H10	Ayuda y documentación
P10.1	¿Existe una sección de ayuda accesible en la interfaz?
P10.2	¿Las instrucciones están bien organizadas y son fáciles de entender?
P10.3	¿El usuario puede buscar soluciones dentro de la app?

Cuadro 27: Criterios de evaluación para ayuda y documentación

5.2.2. Resultados de la Evaluación

La siguiente tabla presenta los resultados obtenidos de los 5 evaluadores para cada criterio:

Criterio	E1	E2	E3	E4	E5	Promedio
H1: Visibilidad del estado del sistema						
Indicador de carga/progreso	4	4	5	4	4	4.2
Confirmación de acciones exitosas	5	4	5	5	4	4.6
Visibilidad de errores	4	4	4	5	4	4.2
Promedio H1:						4.33
H2: Relación entre sistema y mundo real						
Términos comprensibles	5	5	4	5	5	4.8
Iconos familiares	5	4	5	4	5	4.6
Flujo de navegación lógico	4	5	4	5	4	4.4
Promedio H2:						4.60
H3: Control y libertad del usuario						
Deshacer acciones importantes	3	2	3	3	2	2.6
Cancelar procesos	4	4	5	4	4	4.2
Opción de volver atrás	5	5	5	5	5	5.0
Promedio H3:						3.93
H4: Consistencia y estándares						
Botones y menús consistentes	5	5	4	5	5	4.8
Colores y símbolos consistentes	5	5	5	5	5	5.0
Comportamiento predecible	4	5	5	4	5	4.6
Promedio H4:						4.80
H5: Prevención de errores						
Validación de formularios	5	4	5	5	5	4.8
Confirmación antes de acciones irreversibles	5	5	5	4	5	4.8
Evita acciones destructivas sin confirmar	4	5	5	5	4	4.6
Promedio H5:						4.73

Cuadro 28: Resultados de evaluación heurística - Parte 1

Criterio	E1	E2	E3	E4	E5	Promedio
H6: Reconocimiento en lugar de recuerdo						
Nombres de botones claros	5	5	5	5	5	5.0
Opciones visibles	4	5	4	5	4	4.4
Hints y ejemplos visuales	4	3	4	4	3	3.6
Promedio H6:						4.33
H7: Flexibilidad y eficiencia de uso						
Atajos para usuarios avanzados	2	2	1	2	2	1.8
Personalización de opciones	3	3	3	2	3	2.8
Sugerencias para tareas repetitivas	2	3	2	2	3	2.4
Promedio H7:						2.33
H8: Diseño estético y minimalista						
Evita exceso de elementos	5	5	5	5	5	5.0
Jerarquía visual clara	5	4	5	5	4	4.6
Colores e imágenes informativos	4	5	5	4	5	4.6
Promedio H8:						4.73
H9: Recuperación de errores						
Mensajes de error claros	4	4	4	5	4	4.2
Corrección sin perder datos	4	3	4	4	4	3.8
Sugerencias de solución	3	3	4	3	3	3.2
Promedio H9:						3.73
H10: Ayuda y documentación						
Sección de ayuda accesible	5	5	5	5	5	5.0
Instrucciones bien organizadas	4	5	4	5	4	4.4
Búsqueda de soluciones en la app	2	2	2	1	2	1.8
Promedio H10:						3.73

Cuadro 29: Resultados de evaluación heurística - Parte 2

Heurística	Promedio	Evaluación
H4: Consistencia y estándares	4.80	Excelente
H5: Prevención de errores	4.73	Excelente
H8: Diseño estético y minimalista	4.73	Excelente
H2: Relación con el mundo real	4.60	Muy bueno
H1: Visibilidad del estado	4.33	Muy bueno
H6: Reconocimiento sobre recuerdo	4.33	Muy bueno
H3: Control y libertad	3.93	Bueno
H9: Recuperación de errores	3.73	Bueno
H10: Ayuda y documentación	3.73	Bueno
H7: Flexibilidad y eficiencia	2.33	Deficiente
PROMEDIO GENERAL	4.12	Muy bueno

Cuadro 30: Resumen de resultados por heurística

5.2.3. Análisis de Resultados

La evaluación heurística realizada proporcionó información valiosa sobre el estado actual de la plataforma y las áreas que requieren atención futura. Los resultados obtenidos demuestran que la plataforma cumple satisfactoriamente con los principios de usabilidad establecidos, alcanzando un promedio general de 4.12/5.0 puntos.

5.2.3.1. Fortalezas Identificadas

Los resultados de la validación destacan tres áreas donde la plataforma demuestra excelencia:

Consistencia y estándares (4.80/5.0): La plataforma mantiene un diseño uniforme con colores, iconos y comportamientos predecibles en todas las secciones. Esta consistencia es fundamental para usuarios mayores, ya que reduce la carga cognitiva y facilita el aprendizaje progresivo de la interfaz.

Prevención de errores (4.73/5.0): Los mecanismos implementados para validación de formularios y confirmación de acciones importantes han demostrado ser efectivos. Esta característica es especialmente relevante considerando las limitaciones motoras y visuales que pueden incrementar la probabilidad de errores en usuarios mayores.

Diseño minimalista (4.73/5.0): La adherencia a las dos reglas de oro establecidas en el marco teórico se refleja en la alta puntuación obtenida. La interfaz presenta máximo dos ideas principales por pantalla con elementos grandes y bien espaciados, cumpliendo exitosamente con los principios de diseño para personas mayores.

5.2.3.2. Áreas de Mejora Identificadas

La evaluación también reveló tres áreas principales que requieren atención:

Flexibilidad y eficiencia (2.33/5.0): La plataforma carece de funcionalidades avanzadas como atajos o personalización extensa. Sin embargo, esta limitación es parcialmente intencional, dado que se priorizó la simplicidad sobre la eficiencia para el público objetivo. No obstante, existe espacio para incorporar opciones básicas que no comprometan la usabilidad.

Función deshacer (2.60/5.0): La ausencia de mecanismos para revertir acciones representa una oportunidad de mejora significativa. Implementar esta funcionalidad

aumentaría la confianza del usuario al explorar la plataforma sin temor a cometer errores irreversibles.

Búsqueda en ayuda (1.80/5.0): Aunque la plataforma incluye un tour virtual, la falta de una función de búsqueda dentro de la documentación limita la capacidad de los usuarios para encontrar respuestas específicas cuando las necesitan.

Capítulo 6

Conclusión y mejoras futuras

El desarrollo de la plataforma representa una respuesta concreta a la problemática del aislamiento social en la tercera edad mediante la implementación de una solución móvil de cohousing diseñada específicamente para las necesidades y limitaciones de personas mayores. A través de un proceso metodológico que integró empatización con usuarios reales, investigación de principios de usabilidad y validación sistemática, se logró crear una herramienta tecnológica que busca facilitar la formación de comunidades habitacionales compartidas.

6.1. Conclusiones

6.1.1. Logro de Objetivos

El proyecto cumplió satisfactoriamente con los tres objetivos específicos planteados inicialmente. El estudio del estado del arte de soluciones similares y tecnologías aplicables permitió definir una arquitectura escalable y flexible basada en React Native con Supabase que permite escalabilidad horizontal y vertical, con capacidad demostrada para soportar múltiples comunidades simultáneas y expansión futura de funcionalidades sin comprometer la seguridad del sistema. Paralelamente, se logró diseñar y desarrollar una interfaz de usuario intuitiva y una experiencia de usuario optimizada, específicamente adaptadas para personas mayores y facilitando la navegación, selección y gestión de viviendas compartidas.

La validación de estos logros se evidencia en los resultados obtenidos. La evaluación heurística con un promedio general de 4.12/5.0 valida que la interfaz desarrollada

cumple con los principios de usabilidad para el público objetivo, con 9 de 10 heurísticas obteniendo puntuaciones superiores a 3.5 puntos. Además, el proceso de desarrollo por sprints permitió realizar pruebas iterativas durante cada fase, asegurando la calidad técnica y permitiendo ajustes continuos basados en retroalimentación, culminando en una evaluación formal que identificó tanto fortalezas como oportunidades de mejora para el perfeccionamiento futuro de la plataforma.

6.1.2. Valor Generado e Impacto Potencial

El proyecto generó valor en múltiples aspectos académicos y prácticos. La plataforma demuestra cómo los principios de accesibilidad para personas mayores pueden integrarse efectivamente en una aplicación móvil moderna sin comprometer la funcionalidad, mientras que la adaptación de la evaluación heurística considerando específicamente las necesidades de usuarios mayores proporciona un marco replicable para proyectos similares. Además, la plataforma representa una herramienta concreta para abordar el aislamiento social en la tercera edad, problema identificado como crítico en el contexto chileno.

Estos aportes tienen el potencial de generar impacto en múltiples dimensiones prácticas. Desde la perspectiva social, facilitar la formación de comunidades de cohousing puede reducir el aislamiento social en personas mayores, mejorando su calidad de vida y bienestar emocional. En términos económicos, el modelo de vivienda compartida permite optimizar costos habitacionales, aspecto importante considerando la insuficiencia de pensiones y los altos costos de vida identificados durante el proceso de empatización. Desde el punto de vista tecnológico, esto demuestra que es posible desarrollar tecnología inclusiva que no excluya a usuarios con menor alfabetización digital.

6.2. Mejoras Futuras

Las mejoras propuestas se priorizan según su impacto en la experiencia del usuario mayor y la complejidad de implementación, equilibrando la corrección de deficiencias actuales con la expansión hacia funcionalidades más avanzadas.

Prioridad	Mejora	Descripción	Impacto
Mejoras de Usabilidad			
Alta	Función deshacer	Desarrollar mecanismos para revertir acciones críticas en gestión de contratos y declaraciones de pago	Alto
Alta	Mensajes de error mejorados	Enriquecer mensajes con sugerencias específicas y pasos concretos para resolver problemas	Alto
Alta	Pruebas con usuarios reales	Realizar sesiones de prueba con personas mayores para validar decisiones de diseño	Alto
Media	Búsqueda en ayuda	Implementar función de búsqueda dentro de la documentación de ayuda	Medio
Media	Sugerencias contextuales	Agregar más ayudas visuales en formularios complejos, especialmente registro de propiedades	Medio
Baja	Personalización básica	Completar implementación de ajuste de tamaño de fuente y modo oscuro	Bajo
Baja	Atajos contextuales	Evaluar inclusión de atajos simples para acciones frecuentes	Bajo
Mejoras Técnicas			
Media	Notificaciones push nativas	Desarrollar notificaciones para mantener usuarios informados fuera de la aplicación	Medio
Baja	Modo offline parcial	Desarrollar capacidades offline básicas para visualización de información comunitaria	Bajo
Funcionalidades Futuras			
Futura	Eventos comunitarios	Permitir organización y coordinación de actividades sociales dentro de la plataforma	Medio
Futura	Integración servicios salud	Conectar con servicios médicos y de emergencia relevantes para personas mayores	Alto
Futura	Gestión financiera avanzada	Automatización de pagos, división detallada de gastos y reportes mensuales	Alto
Futura	Contratación servicios	Facilitar búsqueda y contratación conjunta de servicios del hogar	Medio
Futura	Evaluación compatibilidad	Implementar algoritmos para sugerir compañeros de vivienda compatibles	Medio

Cuadro 31: Categorización de mejoras futuras por tipo y prioridad

La organización por categorías responde a diferentes momentos del ciclo de vida del producto. Las mejoras de usabilidad emergen directamente de las deficiencias detectadas en la evaluación heurística y representan el refinamiento necesario para optimizar la

experiencia actual sin alterar la arquitectura fundamental. Las mejoras técnicas buscan fortalecer la infraestructura existente, preparando el sistema para mayor adopción y uso intensivo. Las funcionalidades futuras responden a las necesidades expresadas durante el proceso de empatización pero que exceden el alcance inicial del proyecto, requiriendo investigación adicional sobre su impacto en la simplicidad que caracteriza el diseño actual.

6.3. Reflexiones Finales

El desarrollo de la plataforma permitió aplicar principios de diseño centrado en el usuario para crear una herramienta tecnológica adaptada a personas mayores. La combinación de investigación con usuarios reales, aplicación de principios de usabilidad y validación sistemática resultó en una plataforma funcional que considera las capacidades y limitaciones de sus usuarios objetivo, mientras que la evaluación heurística con un promedio de 4.12/5.0 indica un cumplimiento adecuado de los principios de usabilidad. Las áreas de mejora identificadas establecen puntos específicos para trabajo futuro y el proyecto documenta un proceso metodológico para abordar problemas sociales mediante desarrollo de software accesible.

Es importante señalar que, si bien la plataforma fue validada mediante evaluación heurística, aún está pendiente realizar una marcha blanca con usuarios finales. Esta fase, contemplada como siguiente paso antes del lanzamiento, incluirá la aplicación de la escala SUS (System Usability Scale) para medir la usabilidad de forma cuantitativa. La marcha blanca junto con la evaluación SUS permitirán validar las decisiones de diseño en condiciones reales de uso y realizar ajustes basados en la retroalimentación directa de los usuarios.

El impacto real de la plataforma dependerá de su adopción y uso efectivo por parte de personas mayores interesadas en formar comunidades de cohousing. Los aspectos técnicos y de diseño implementados constituyen una primera versión funcional que establece las bases para un desarrollo iterativo, mientras que las mejoras propuestas permitirán adaptar la plataforma según la retroalimentación de usuarios reales. La experiencia confirma que es posible desarrollar tecnología inclusiva cuando se priorizan las necesidades reales de los usuarios sobre las barreras tecnológicas tradicionales.

Bibliografía

- Awan, M., Ali, S., Ali, M., Abrar, M. F., Ullah, H., & Khan, D. (2021). Usability barriers for elderly users in smartphone-app usage: an analytical hierarchical process-based prioritization. *Scientific Programming*, 2021, 2780257. <https://doi.org/10.1155/2021/2780257>
- Cooperativa Ciencia. (2024, agosto). *Personas mayores: Un 81 % cree que el país está mal preparado para atender sus necesidades*. Consultado el 22 de agosto de 2024, desde https://www.cooperativaciencia.cl/radiociencia/2024/08/07/personas-mayores-un-81-cree-que-el-pais-esta-mal-preparado-para-atender-sus-necesidades/?utm_campaign=feed&utm_medium=referral&utm_source=later-linkinbio
- Culmia. (2024). *Senior Living: Todo lo que debes saber sobre este modelo de viviendas para la tercera edad*. Consultado el 22 de agosto de 2024, desde <https://www.culmia.com/blog/senior-living-modelo-de-viviendas-tercera-edad>
- Dar, C., Hershcovitch, M., & Morrison, A. (2023). RLS Side Channels: Investigating Leakage of Row-Level Security Protected Data Through Query Execution Time. *Proceedings of the ACM on Management of Data*, 1(1), 89:1-89:25. <https://doi.org/10.1145/3588943>
- Fundación Cohousing. (2024). *Senior Cohousing - Innovación y Diseño en Hábitat Comunitario*. Consultado el 5 de septiembre de 2024, desde <https://www.fundacioncohousing.cl/>
- Gómez-Hernández, M., Ferre, X., Moral, C., & Villalba-Mora, E. (2023). Design guidelines of mobile apps for older adults: systematic review and thematic analysis. *JMIR mHealth and uHealth*, 11, e43186. <https://doi.org/10.2196/43186>
- Kelley, T. (2018, noviembre). *Build Your Creative Confidence: Empathy Maps*. IDEO. Consultado el 9 de diciembre de 2024, desde <https://www.ideo.com/journal/build-your-creative-confidence-empathy-maps>
- Lemos, V. (2024). *Cohousing: la creciente tendencia de amigos que se juntan para compartir la vejez en una vivienda comunitaria*. BBC Mundo. Consultado el 22 de agosto de 2024, desde <https://www.bbc.com/mundo/articles/cd1qxx7pdvo>
- Nesterly. (2024). *Nesterly - Intergenerational Home Sharing*. Consultado el 5 de septiembre de 2024, desde <https://www.nesterly.com>
- Poimandres. (2024). *Zustand - Bear necessities for state management in React*. Consultado el 10 de octubre de 2024, desde <https://zustand.docs.pmnd.rs/getting-started/introduction>

- PostgreSQL. (2024). *PostgreSQL Documentation: Row Security Policies*. Consultado el 10 de septiembre de 2024, desde <https://www.postgresql.org/docs/current/ddl-rowsecurity.html>
- Rangel, A. (2024). *Evaluaciones Heurísticas - Plantilla de Evaluación*. Consultado el 25 de julio de 2024, desde <https://www.figma.com/community/file/1471302204031138595/evaluaciones-heuristicas>
- Silvernest. (2024). *Silvernest - Safe, Simple Home Sharing for 50+*. Consultado el 5 de septiembre de 2024, desde <https://www.silvernest.com>
- Supabase. (2024). *Supabase Documentation*. Consultado el 5 de septiembre de 2024, desde <https://supabase.com/docs>
- TanStack. (2024). *TanStack Query - Powerful asynchronous state management for TS/JS, React, Solid, Vue, Svelte and Angular*. Consultado el 10 de octubre de 2024, desde <https://tanstack.com/query/latest>
- Zaytseva, D. (2024). *How to Conduct a Heuristic Analysis: A Step-by-Step Guide for Improving UX*. Consultado el 10 de febrero de 2025, desde <https://medium.com/@dariazaytseva07/how-to-conduct-a-heuristic-analysis-a-step-by-step-guide-for-improving-ux-302c422865c4>

Apéndice A

Anexos

A1. Alertas de Manejo de Excepciones

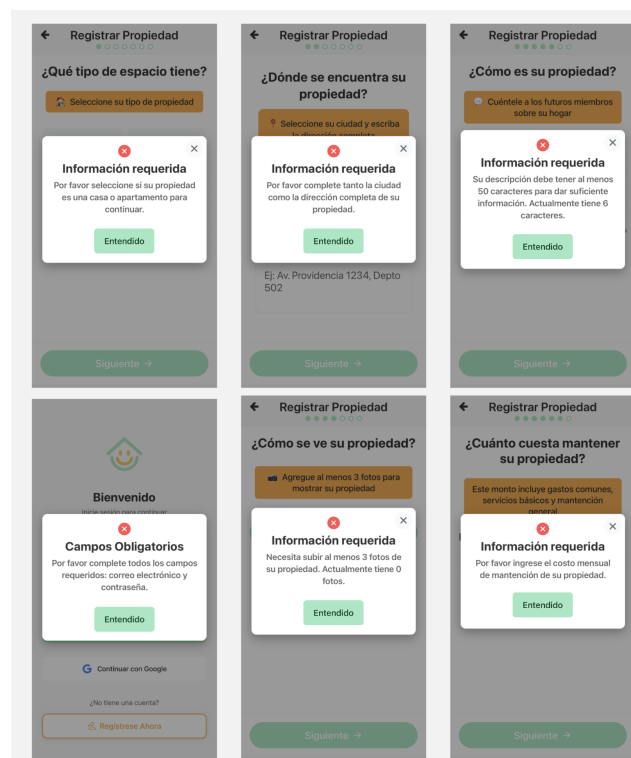


Figura A1.1: Ejemplos de alertas implementadas en la plataforma

A2. Menú Central de Anfitrión Sin Inquilinos



Figura A2.1: Vista del menú central para anfitrión sin inquilinos

A3. Vista de Pagos de Anfitrión Sin Datos

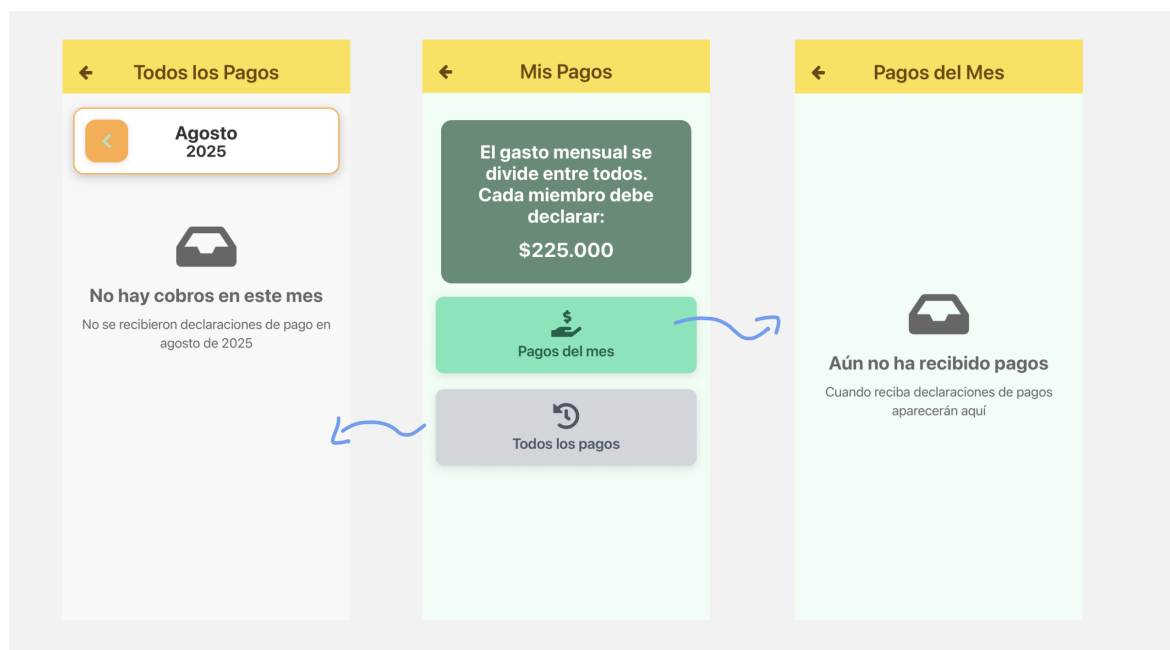


Figura A3.1: Vista de pagos de anfitrión sin declaraciones registradas

A4. Plantilla de Evaluación Heurística

Heurística	Descripción	Pregunta 1	Puntaje 1	Pregunta 2	Puntaje 2	Pregunta 3	Puntaje 3	Promedio Heurística
Visibilidad del estado del sistema	Los usuarios deben saber en todo momento qué está pasando en la interfaz.	¿La interfaz muestra un indicador de carga o progreso cuando se está procesando algo?	0	¿Recibe el usuario una confirmación visual o mensaje cuando una acción se completa exitosamente?	0	¿Se destacan claramente los errores o problemas que ocurren en el sistema?	0	0
Relación entre el sistema y el mundo real	La interfaz debe hablar el mismo idioma que los usuarios, usando términos familiares.	¿Los términos y etiquetas utilizados en la app son comprensibles para el usuario promedio?	0	¿Los iconos y elementos visuales representan objetos o acciones que los usuarios ya conocen?	0	¿El flujo de navegación sigue una lógica natural, como en experiencias del mundo real?	0	0
Control y libertad del usuario	Los usuarios deben poder corregir errores y navegar libremente.	¿Se pueden deshacer acciones importantes (como eliminar un archivo)?	0	¿Los usuarios pueden cancelar un proceso en cualquier momento si cambiaron de opinión?	0	¿Existe una opción de volver atrás o regresar a un estado anterior?	0	0
Consistencia y estándares	La interfaz debe ser predecible y seguir estándares reconocidos.	¿Los botones, menús y acciones funcionan y se ven de la misma manera en todas las secciones de la app?	0	¿Los colores y símbolos tienen el mismo significado en toda la app?	0	¿Los elementos interactivos (botones, enlaces, iconos) tienen un comportamiento predecible?	0	0
Prevención de errores	Es mejor prevenir errores que corregirlos después.	¿El sistema evita que el usuario envíe formularios incompletos o con datos incorrectos?	0	¿Se pide una confirmación antes de acciones irreversibles, como eliminar una cuenta o un archivo?	0	¿El sistema evita acciones destructivas sin confirmación, como cerrar una pestaña sin guardar cambios?	0	0
Reconocimiento en lugar de recuerdo	Es más fácil reconocer algo que recordarlo.	¿Los nombres de los botones y menús son claros y fáciles de entender sin adivinar su función?	0	¿El usuario puede ver opciones y acciones disponibles sin tener que recordar dónde están?	0	¿Existen hints, tooltips o ejemplos visuales que ayuden al usuario a completar tareas sin esfuerzo?	0	0
Flexibilidad y eficiencia de uso	Una buena interfaz se adapta tanto a principiantes como a expertos.	¿Existen atajos de teclado o accesos rápidos para usuarios frecuentes o avanzados?	0	¿El usuario puede personalizar opciones para adaptar la experiencia a sus necesidades?	0	¿El sistema ofrece sugerencias inteligentes o automatización para agilizar tareas repetitivas?	0	0
Diseño estético y minimalista	Menos es más: evitar elementos innecesarios en la interfaz.	¿La interfaz evita el exceso de texto o elementos visuales innecesarios?	0	¿Los colores, tipografías y elementos gráficos siguen una jerarquía clara para destacar lo importante?	0	¿Los colores e imágenes ayudan a comprender la info?	0	0
Ayuda a reconocer, diagnosticar y recuperarse de errores	Cuando los usuarios cometen errores, la interfaz debe ayudar a corregirlos.	¿Los mensajes de error explican claramente qué pasó y cómo solucionarlo?	0	¿Los errores permiten ser corregidos sin perder datos o reiniciar el proceso desde cero?	0	¿El sistema sugiere soluciones o enlaces a ayuda cuando ocurre un problema?	0	0
Ayuda y documentación	Debe haber ayuda y documentación accesible cuando el usuario la necesite.	¿Existe una sección de ayuda o preguntas frecuentes accesible en la interfaz?	0	¿Las instrucciones y guías están bien organizadas y son fáciles de entender?	0	¿El usuario puede buscar soluciones dentro de la app sin salir a otra web?	0	0

Figura A4.1: Plantilla de evaluación heurística completada por los evaluadores durante el proceso de validación

A5. Repositorio de Imágenes en Alta Resolución

Todas las imágenes de mapas de navegación y capturas de pantalla presentadas en este documento están disponibles en mayor resolución en el repositorio público de Google Drive del proyecto: https://drive.google.com/drive/folders/1I_AOkz4DXo8mhBFcVjBL23WPpZ1bg8ap?usp=sharing