



Aplicación Móvil de Gestión y Control para el Conjunto Residencial Hacienda San Rafael en
Zipaquirá

Luisa F. Romero Suarez

Omar S. Rodríguez Puerto

Corporación Universitaria Minuto de Dios

Rectoría Bogotá Región (Cundinamarca)

Centro Universitario Zipaquirá (Cundinamarca)

Programa Ingeniería de Sistemas

octubre de 2025

Aplicación Móvil de Gestión y Control para el Conjunto Residencial Hacienda San Rafael en
Zipaquirá

Luisa Fernanda Romero Suarez

Omar Santiago Rodríguez Puerto

Asesor

Juan Carlos Herrera Estrada

Corporación Universitaria Minuto de Dios

Rectoría Bogotá Región (Cundinamarca)

Centro Universitario Zipaquirá (Cundinamarca)

Programa Ingeniería de Sistemas

octubre de 2025

Contenido

Resumen	5
Abstract	6
Introducción	7
CAPÍTULO I	8
1 Planteamiento Problema	8
2 Formulación del Problema	9
3 Objetivos	9
3.1 Objetivo General	9
3.2 Objetivos Específicos	9
3.3 Justificación	10
CAPÍTULO II	11
4 Diseño Metodológico	11
5 Tipo Investigación	13
6 Marcos de Referencia	14
6.1 Marco Antecedentes	14
6.2 Marco Teórico	15
6.3 Marco Conceptual	15
6.4 Marco Institucional	16
6.5 Marco Organizacional	16
CAPÍTULO III	17
7 Estudio Técnico	17
7.1 Recursos Tecnológicos	18
7.2 Recursos Humanos	19
8 Cronograma	20
9 Requerimientos	25
9.1 Requerimientos del Sistema	25
9.2 Requerimientos de Seguridad	35
9.3 Requerimientos de Fiabilidad	37
10 Diseño Arquitectónico	38
10.1 Arquitectura del sistema	38
10.1.1 Patrones de Diseño	38
10.1.2 Diseño Orientado a objetos	39
CAPÍTULO IV	40
11 Seguridad	40
11.1 Matriz de Riesgos	40
11.2 Análisis de Riesgos	41
11.3 Planeación de Riesgos	42
11.4 Marcos de Referencia	43
11.4.1 Marco Legal y Normativo	43

11.4.2 Marco de Calidad y Pruebas	44
12 Presupuesto	45
12.1 Costos Directos	45
12.2 Costos Indirectos	46
CAPÍTULO V	47
13 Exposición de Resultados	47
13.1 Prototipo Interfaz	47
13.2 Casos de Uso	55
13.3 Diagramas de Secuencia	57
13.4 Diagramas de Estado	87
14 Conclusiones	99
14.1 Recomendaciones	99
Referencias	101

Resumen

La gestión de conjuntos residenciales se caracteriza por una notoria lentitud en la ejecución de tareas y resolución de problemas, situación que afecta la calidad de vida de los residentes y la eficiencia operativa de estas comunidades. Esta problemática tiene su origen en una estructura de toma de decisiones compleja, donde cualquier acción relevante debe pasar por demasiados filtros perdiendo así la importancia según el asunto. La comunicación ineficiente entre residentes, administración y celadores constituye otro cuello de botella importante, ya que los canales tradicionales como carteleras y circulares resultan obsoletos y generan desinformación constante a los residentes. Finalmente, la resistencia al cambio y el apego a sistemas manuales anticuados, motivados por el temor a invertir en tecnología o por simple desconocimiento de alternativas más eficientes, perpetúan este círculo vicioso.

El resultado es generar una **aplicación móvil Android de Gestión y Control para el conjunto residencial Hacienda San Rafael**, con el objetivo de centralizar y automatizar procesos administrativos y de seguridad que actualmente se manejan de manera manual o con herramientas dispersas. Entre las funcionalidades que se busca integrar están: un inicio de novedades, sección de notificaciones, manejo de paquetería, pagos, un menú principal donde se gestionan los accesos, reservas, quejas, mascotas y el perfil.

El proyecto responde a la necesidad de mejorar la eficiencia operativa, la trazabilidad de la información y la calidad de vida de los residentes mediante una plataforma digital intuitiva y adaptable. Este sistema contará con roles definidos para administrador, residentes y celadores, lo que permitirá asignar responsabilidades, gestionar procesos y reducir la carga laboral del personal para tener un sistema de administración organizado.

Palabras clave: eficiencia, seguridad, control, roles, información, automatización, prototipado y procesos.

Abstract

The management of residential complexes is characterized by a noticeable slowness in the execution of tasks and problem solving, a situation that affects the quality of life of residents and the operational efficiency of these communities. This problem stems from a complex decision-making structure, where any relevant action must pass through too many filters, thus losing importance depending on the issue. Inefficient communication between residents, management, and security guards is another major bottleneck, as traditional channels such as bulletin boards and circulars are obsolete and generate constant misinformation for residents. Finally, resistance to change and attachment to outdated manual systems, motivated by fear of investing in technology or simple ignorance of more efficient alternatives, perpetuate this vicious cycle.

The result is the creation of an Android mobile management and control application for the Hacienda San Rafael residential complex, with the aim of centralizing and automating administrative and security processes that are currently handled manually or with scattered tools. Among the features to be integrated are: a news feed, notifications section, parcel management, payments, and a main menu where access, reservations, complaints, pets, and profiles are managed.

The project responds to the need to improve operational efficiency, information traceability, and residents' quality of life through an intuitive and adaptable digital platform. This system will have defined roles for administrators, residents, and security guards, which will allow for the assignment of responsibilities, management of processes, and reduction of staff workload to achieve an organized administration system.

Keywords: efficiency, security, control, roles, information, automation, prototyping, and processes.

Introducción

Actualmente, en los conjuntos residenciales se presentan deficiencias en la gestión administrativa.

Un ejemplo de ello es el conjunto residencial Hacienda San Rafael en Zipaquirá, donde la organización para la comunicación de novedades, la sección de notificaciones, el manejo de paquetería, el pago de administración, la gestión de accesos, las reservas de zonas comunes, el seguimiento de quejas, la información sobre mascotas y la administración de los perfiles de los residentes resultan ineficientes para el adecuado funcionamiento, ya que se realizan de forma manual —en papel, archivos escritos a mano o mediante canales desorganizados—, lo cual genera retrasos, errores, inconformidad y descontento entre los residentes, afectando la comunicación efectiva con los celadores y la administración.

Frente a esta problemática, surge la necesidad de desarrollar un producto mínimo viable para la Gestión y Control que integre todos estos procesos en una sola plataforma, accesible para todos los usuarios adaptada según los roles de administrador, residente y celador.

Este proyecto busca transformar la gestión interna del conjunto residencial mediante la digitalización, lo que permitirá automatizar tareas rutinarias, facilitar la comunicación interna, mejorar la seguridad y optimizar los tiempos de respuesta en los procesos administrativos. La aplicación se diseñará con un enfoque en usabilidad, escalabilidad y eficiencia, con el propósito de ofrecer una herramienta moderna que fortalezca la convivencia y organización dentro de estas comunidades. La propuesta se articula en torno a objetivos claros y a una metodología de desarrollo Scrum adaptada. El plan de implementación está orientado a brindar una solución efectiva, funcional y sostenible en el tiempo.

CAPÍTULO I

1 Planteamiento Problema

El desarrollo de una aplicación móvil de Gestión y Control para el conjunto residencial Hacienda San Rafael se justifica por la necesidad urgente de modernizar y sistematizar los procesos administrativos y de seguridad dentro de la comunidad residencial. La vida en conjunto exige una coordinación efectiva entre múltiples actores, y para ello se requiere una herramienta tecnológica capaz de responder a los retos actuales y en tiempo real que enfrentan estos espacios. Uno de los principales motivos para implementar esta solución es la lentitud y desorganización en los procesos actuales. Las acciones más básicas como dar avisos importantes, llegada de recibos o paquetería, entrega de recibo administración y demás actividades dentro del conjunto se convierten en tareas complejas por la falta de un sistema que centralice la información y facilite la interacción.

Esta aplicación permitirá digitalizar todos estos procedimientos, mejorando la eficiencia y reduciendo significativamente el margen de error. Además, la aplicación permitirá una mejor distribución de funciones gracias a la implementación de tres roles claramente definidos: administrador, residente y celador. Cada uno tendrá permisos específicos dentro de la aplicación, lo que agiliza la toma de decisiones, evita duplicidad de funciones y mejora la comunicación interna del conjunto. El administrador podrá controlar los accesos, gestionar y administrar reservas, editar o eliminar información. El residente podrá generar accesos, publicar información de sus mascotas o novedades, y gestionar sus notificaciones. Por su parte, el celador podrá controlar accesos de forma rápida, verificando la información en tiempo real y visualizando los diferentes módulos de la aplicación.

Finalmente, la implementación de esta aplicación responde a la transformación digital que atraviesan las sociedades actuales. A través de una plataforma amigable, escalable y adaptable, se podrá ofrecer un servicio más moderno y eficiente que incremente la calidad de vida de los residentes, fortalezca la administración del conjunto y fomente una convivencia armónica y ordenada.

2 Formulación del Problema

¿Cómo puede una aplicación móvil centralizar y automatizar los procesos de gestión administrativa y operativa en un conjunto residencial de Zipaquirá, mejorando la eficiencia, trazabilidad y comunicación entre residentes, celadores y administración?

3 Objetivos

3.1 Objetivo General

Desarrollar una aplicación móvil para Android que permita la gestión y el control integral del conjunto residencial Hacienda San Rafael, centralizando y automatizando los procesos a partir de los diferentes roles, incluyendo la administración de novedades, notificaciones y el acceso a un menú principal dentro de un único aplicativo.

3.2 Objetivos Específicos

- Analizar los requerimientos del sistema mediante el levantamiento de información con un residente del conjunto residencial Hacienda San Rafael en Zipaquirá, identificando los procesos críticos que requieren digitalización y definiendo las funcionalidades para cada rol del aplicativo (administrador, residente y celador), estableciendo la base para el desarrollo del proyecto.
- Diseñar las interfaces de usuario según los distintos roles de la aplicación móvil, definiendo la estructura de la base de datos, los flujos de navegación entre módulos y los prototipos de alta fidelidad, con el fin de garantizar una experiencia intuitiva y accesible para todos los perfiles.
- Implementar los módulos funcionales a través de sprints de 1-2 semanas, codificando en Android Studio las funcionalidades de autenticación, inicio de novedades, notificaciones y menú principal agregando nuevas funciones en cada iteración con sus respectivas validaciones.

3.3 Justificación

El desarrollo de una aplicación móvil para la gestión y control del conjunto residencial se justifica por la necesidad urgente de modernizar y sistematizar los procesos administrativos y de seguridad dentro de la comunidad. La convivencia en un espacio compartido exige una coordinación efectiva entre los diferentes actores —como la administración, los celadores y los residentes—, lo que demanda una herramienta tecnológica capaz de responder a los retos actuales de organización, comunicación y control que enfrentan este tipo de comunidades.

Uno de los principales motivos para implementar esta solución es la lentitud y desorganización en los procesos actuales. Acciones tan básicas como autorizar una visita, dar notificaciones, presentar una queja o reservar una zona común se convierten en tareas complejas y demoradas por la falta de un aplicativo que centralice la información y facilite la interacción. Con la implementación de la aplicación móvil, todos estos procedimientos podrán digitalizarse, mejorando la eficiencia y reduciendo significativamente el margen de error.

Asimismo, la aplicación móvil contribuirá a una distribución más eficiente de las funciones mediante la implementación de los 3 roles definidos. Esta diferenciación de permisos permitirá agilizar la toma de decisiones, evitar la duplicidad de tareas y fortalecer la comunicación interna del conjunto, garantizando una gestión más organizada y efectiva.

Finalmente, esta propuesta responde a la necesidad de iniciar un proceso de transformación digital dentro del conjunto residencial. A través de una herramienta funcional, escalable y adaptable, se busca ofrecer una primera versión que digitalice los procesos más críticos, con la proyección de evolucionar hacia un aplicativo más robusto que incremente la calidad de vida de los residentes, fortalezca la administración del conjunto y fomente una convivencia armónica y ordenada.

CAPÍTULO II

4 Diseño Metodológico

OBJETIVO ESPECÍFICO	ACTIVIDAD	TÉCNICA	INSTRUMENTO	POBLACIÓN	RESULTADO ESPERADO
Analizar los requerimientos del sistema mediante el levantamiento de información con un residente del conjunto residencial Hacienda San Rafael en Zipaquirá, identificando los procesos críticos que requieren digitalización y definiendo las funcionalidades para cada rol del aplicativo (administrador, residente y celador), estableciendo la base para el desarrollo mediante ciclo de vida incremental .	Priorizar requerimientos y funcionalidades según criticidad para planificar incrementos del desarrollo. Identificar procesos manuales actuales y puntos críticos de gestión. Definir requerimientos para cada rol (administrador, residente, celador).	Observación directa de procesos operativos en portería y administración. Análisis de documentos y formatos físicos utilizados actualmente. Metodología Scrum	Formato requerimientos	Administrador del conjunto Desarrollador Backend Desarrollador Fronend Residente	Documento de requerimientos funcionales y no funcionales del sistema. Plan de incrementos definido con módulos por sprint.
Diseñar la arquitectura del software y las interfaces	Diseñar la arquitectura Modelar la base de datos	Modelado UML: diagramas de	Android Studio	Administrador del conjunto	Arquitectura del sistema

<p>de usuario de la aplicación móvil siguiendo un enfoque incremental, estableciendo la estructura de la base de datos, los flujos de navegación entre módulos y los prototipos de alta fidelidad que garanticen una experiencia intuitiva y accesible para todos los perfiles de usuario, priorizando los módulos según su criticidad para entregas progresivas</p>	<p>relacional Supabase.</p> <p>Definir flujos de navegación entre módulos por rol.</p> <p>Diseñar interfaces orientadas a usabilidad y accesibilidad.</p>	<p>casos de uso, clases y secuencia.</p> <p>Herramientas de prototipado: Figma.</p> <p>Diseño de base de datos relacional</p> <p>Principios de diseño UI/UX centrado en el usuario.</p> <p>Metodología Scrum adaptable</p>	<p>Herramientas UML</p> <p>Base de Datos Supabase</p> <p>IntelliJ IDEA</p>	<p>Desarrollador Backend</p> <p>Desarrollador Fronend</p>	<p>definida (diagramas UML, modelo de base de datos normalizado, prototipo de interfaces). Mockups de las pantallas principales.</p>
<p>Implementar los módulos funcionales mediante desarrollo incremental por sprints de 1-2 semanas, codificando en Android Studio las funcionalidades de autenticación,</p>	<p>Desarrollo e integración de módulos en Android Studio con pruebas progresivas.</p> <p>Configurar el entorno de IntelliJ IDEA y JPA necesarios.</p> <p>Implementar módulos de forma incremental:</p>	<p>Programación en Java/Kotlin para IntelliJ/Android.</p> <p>Supabase para autenticación, base de datos en tiempo real y notificaciones.</p>	<p>Android Studio</p> <p>Herramientas UML</p> <p>Base de Datos Supabase</p> <p>IntelliJ IDEA</p>	<p>Administrador del conjunto</p> <p>Desarrollador Backend</p> <p>Desarrollador Fronend</p> <p>Residente</p>	<p>Aplicación móvil con módulos básicos implementados y funcionales en versión inicial.</p>

control de accesos, registro de mascotas, recepción de paquetes, reservaciones, pagos y notificaciones de forma progresiva, agregando incrementos funcionales en cada iteración con sus respectivas validaciones y notificaciones.	Sprint 1 - Sprint 6 Realizar Daily y Sprint Review al finalizar cada sprint.	Control de versiones con Git/GitHub. Metodología Scrum adaptable Programación Orientada a Objetos			
--	---	---	--	--	--

Tabla 1, Diseño Metodológico

5 Tipo Investigación

El presente proyecto corresponde a una investigación aplicada de tipo descriptiva y tecnológica, orientada al desarrollo de una aplicación móvil que responda a necesidades reales identificadas en el conjunto residencial Hacienda San Rafael en Zipaquirá.

La investigación aplicada se centra en la utilización de conocimientos teóricos y técnicos de la ingeniería de sistemas para resolver un problema concreto: la falta de digitalización en los procesos de comunicación, registro y gestión administrativa del conjunto. A diferencia de la investigación pura, no busca generar teoría, sino aplicar la existente para diseñar una herramienta funcional que ofrezca soluciones prácticas.

Desde el punto de vista metodológico, el enfoque es mixto, ya que integra componentes cualitativos y cuantitativos. En el componente cualitativo se emplean técnicas como la observación directa y análisis documental de los procesos actuales en portería, administración y comunicación

interna. Estas herramientas permiten comprender el contexto, identificar necesidades de los usuarios (administrador, residentes y celadores) y reconocer los problemas asociados a la gestión manual de la información.

Por su parte, el enfoque cuantitativo se refleja en la medición, priorización y validación de las funcionalidades, así como en el uso de métricas relacionadas con tiempos de respuesta, eficiencia de los procesos y niveles de satisfacción de los usuarios durante las pruebas del producto mínimo viable. De este modo, la combinación de ambos enfoques garantiza una visión integral del problema y una base sólida para la propuesta tecnológica.

6 Marcos de Referencia

6.1 Marco Antecedentes

Neivor — Software para gestión de conjuntos residenciales en Colombia

Este desarrollo colombiano, mencionado por la Universidad de los Andes, es una plataforma que “mejora la gestión y la administración de conjuntos residenciales, digitalizando los pagos y estableciendo nuevas vías de comunicación entre residentes y administradores”. (UniAndes,2022)

Kovive360 — Cohabit

Es una plataforma de software diseñada para simplificar y digitalizar la administración de conjuntos residenciales y propiedades horizontales. Siendo una herramienta para mejorar la gestión y la convivencia dentro de las comunidades de propietarios y residentes. (2021)

Webdificio — Plataforma digital para gestión de edificios y conjuntos residenciales

Este artículo habla de una iniciativa colombiana para digitalizar la gestión de edificios, ahorrar tiempo y facilitar trámites que normalmente se realizan en la propiedad horizontal. (Mejia,2017)

6.2 Marco Teórico

El desarrollo de este proyecto se fundamenta en la identificación de cuellos de botella, demoras y errores asociados a la gestión manual en los conjuntos residenciales. En este contexto, se resalta la importancia de la digitalización, la trazabilidad y la automatización de procesos a través de la implementación de una aplicación móvil centralizada, que permita mejorar la eficiencia operativa, la seguridad y la calidad del servicio.

Las bases técnicas del proyecto se sustentan en la arquitectura modular, el uso de patrones de diseño, la gestión de roles y los protocolos de comunicación digital, los cuales garantizan escalabilidad, flexibilidad y sostenibilidad en el tiempo. Estos elementos, aplicados bajo los principios de la ingeniería de procesos, buscan asegurar la mejora continua en la administración de los conjuntos residenciales y optimizar la interacción entre los diferentes actores de la comunidad.

6.3 Marco Conceptual

La gestión de un conjunto residencial implica la coordinación efectiva de múltiples actores (administrador, residentes y celadores) y la ejecución de diversos procesos operativos. En este contexto, la digitalización mediante una aplicación móvil se plantea como un mecanismo para mejorar la eficiencia, fortalecer la comunicación interna, reducir errores y garantizar la trazabilidad de las actividades.

Entre los conceptos fundamentales que guían este proyecto se encuentra el ciclo de vida incremental, que divide el desarrollo en múltiples entregas parciales, permitiendo priorizar funcionalidades críticas, obtener retroalimentación temprana y reducir riesgos. De igual manera, se

retoman los principios de la metodología ágil Scrum, cuyos elementos —como el Product Backlog, Sprint Planning, Daily Scrum y Sprint Review— facilitan la organización iterativa y flexible del trabajo.

En cuanto al diseño técnico, se adopta una **arquitectura modular**, que separa la lógica de negocio, la interfaz de usuario y la gestión de datos en componentes independientes. Esta modularidad resulta esencial para el desarrollo incremental, al permitir que las funcionalidades se construyan, prueben y desplieguen de manera independiente sin afectar la estabilidad del resto del aplicativo.

6.4 Marco Institucional

La Corporación Universitaria UNIMINUTO es una reconocida institución colombiana con presencia en varias regiones del país, que se caracteriza por su compromiso en la formación integral de profesionales y contribuir al desarrollo de un país más justo y equitativo.

6.5 Marco Organizacional

El desarrollo de la aplicación móvil se llevó a cabo bajo la **metodología ágil Scrum**, seleccionada por su adaptabilidad, flexibilidad y enfoque iterativo. Dentro de este marco, se definieron cinco roles fundamentales: **Scrum Master**, responsable de garantizar la correcta aplicación de la metodología; **Desarrollador Backend** y **Desarrollador Frontend**, encargados de la implementación técnica de las funcionalidades; **Product Owner**, orientado a priorizar los requerimientos y necesidades del cliente; y los **Stakeholders**, como actores clave en la validación y retroalimentación del producto. Esta organización permitió mantener una comunicación fluida, entregas continuas y una gestión estructurada del proyecto.

La estructura organizativa del equipo se complementa con la adopción de un **ciclo de vida incremental**, que permitió construir el proyecto en versiones sucesivas y agregar nuevas funcionalidades en cada iteración, manteniendo siempre el control de la calidad y la estabilidad del aplicativo. A nivel

técnico, se implementó una **arquitectura modular**, lo que facilitó el mantenimiento, la escalabilidad y las pruebas por componentes, asegurando que los entregables respondieran a los objetivos definidos. De esta forma, la organización del equipo de trabajo y la aplicación de prácticas de ingeniería de software garantizaron una base sólida para la evolución y mejora continua del producto.

CAPÍTULO III

7 Estudio Técnico

El proyecto se centra en el desarrollo de una aplicación móvil integral de gestión y control para el conjunto residencial Hacienda San Rafael en Zipaquirá, implementada mediante tecnologías nativas de Android utilizando Kotlin como lenguaje principal de programación en Android Studio, complementada con un backend robusto desarrollado en Java con el framework Spring Boot y JPA (Java Persistence API) en IntelliJ IDEA, que incorpora 48 dependencias especializadas para garantizar seguridad, escalabilidad y mantenibilidad del sistema. La arquitectura del sistema se organiza en módulos funcionales diferenciados según los roles de usuario —administrador, residente y celador— con el propósito de optimizar la gestión de procesos administrativos y operativos en cada nivel de interacción.

El módulo de celadores permite verificar visitas autorizadas previamente por residentes, consultar el registro de mascotas del conjunto, enviar notificaciones de paquetería a usuarios y visualizar las reservas activas para control de acceso a las instalaciones; el módulo de residentes facilita la autorización anticipada de visitantes, la recepción de notificaciones automáticas sobre paquetes disponibles recogidos, la publicación de novedades, actualización de información de sus mascotas, y la reserva de espacios comunes consultando disponibilidad en tiempo real; mientras que el módulo de administrador supervisa de manera centralizada todos los procesos, gestiona usuarios del sistema, edita o revoca las

autorizaciones de visitas cuando sea necesario, maneja el catálogo completo de mascotas registradas, y modifica o visualiza las reservas de los residentes.

Las interfaces de usuario se diseñan siguiendo los principios de Material Design 3, utilizando Jetpack Compose para construcción declarativa de UI cuando sea apropiado, y componentes XML tradicionales para vistas más complejas, garantizando consistencia visual, accesibilidad para usuarios con diferentes capacidades y una experiencia intuitiva que minimiza la curva de aprendizaje. La navegación entre módulos se implementa mediante Navigation Component de Android Jetpack.

Las funcionalidades del sistema se orientan a incrementar significativamente la eficiencia operativa del conjunto residencial, minimizar errores humanos en registros manuales, mejorar la trazabilidad y auditoría de eventos, fortalecer la seguridad mediante control digital de accesos, optimizar la comunicación entre todos los actores y elevar sustancialmente la experiencia y satisfacción de los residentes con los servicios administrativos ofrecidos.

7.1 Recursos Tecnológicos

El aplicativo móvil del conjunto residencial Hacienda San Rafael se basa en los siguientes recursos tecnológicos:

- Software Desarrollo: Java para la lógica del backend y Kotlin para la lógica del frontend.
- Base de datos: Supabase.
- Herramientas de trabajo: Android Studio, IntelliJ IDEA
- Marcos: Spring Boot, Jetpack Compose, Material Design 3
- Colaboración: GitHub, Jira/Trello
- Arquitectura: Modular, navegación por componentes, pruebas unitarias y de integración.

7.2 Recursos Humanos

El desarrollo del aplicativo móvil será ejecutado en la metodología Scrum adaptable, asegurando un proceso colaborativo y eficiente. En esta metodología se divide con Scrum Master, Desarrollador Frontend, Desarrollador Backend, Product Owner y Stakeholders.

- Luisa Fernanda Romero Suarez tiene el rol de desarrollador frontend, donde se usa el lenguaje de programación Kotlin. El objetivo se basa en diseñar una interfaz intuitiva y de fácil uso para los roles en la navegación de los módulos.
- Omar Santiago Rodríguez Puerto tiene el rol de Desarrollador Backend, donde usa el lenguaje de programación Java, así como la configuración de la base de datos Supabase asegurando una conexión exitosa.

8 Cronograma

El cronograma de este proyecto con metodología Scrum adaptable tiene un periodo desde el 01 de Agosto hasta el 25 de Noviembre del presente año con una duración de 16 semanas para poder realizar la entrega de un producto mínimo viable.



Ilustración 1, Cronograma de Gantt

Fase	Actividad	Fecha Inicio	Fecha Fin	Duración	Responsable
Análisis de Requerimientos	Observación de procesos operativos	04/08/2025	05/08/2025	2 días	Product Owner
Análisis de Requerimientos	Definición de Product Backlog	06/08/2025	07/08/2025	2 días	Product Owner
Diseño de Arquitectura	Modelado UML (casos de uso, clases, secuencia)	06/08/2025	08/08/2025	3 días	Desarrollador Backend
Diseño de Arquitectura	Diseño de base de datos en Supabase	08/08/2025	09/08/2025	2 días	Desarrollador Backend
Diseño de Interfaces	Creación de wireframes y prototipos en Figma	06/08/2025	10/08/2025	5 días	Desarrollador Frontend
Diseño de Interfaces	Validación de prototipos con usuarios	10/08/2025	10/08/2025	1 día	Stakeholders

Tabla 2, Actividades Sprint 0

Sprint 1	Configuración entorno Android Studio e IntelliJ	12/08/2025	13/08/2025	2 días	Desarrollador Backend
Sprint 1	Implementación autenticación con Supabase	14/08/2025	17/08/2025	4 días	Desarrollador Backend
Sprint 1	Desarrollo interfaces de login y roles	14/08/2025	17/08/2025	4 días	Desarrollador Frontend
Sprint 1	Implementación módulo inicio de novedades	18/08/2025	21/08/2025	4 días	Desarrollador Backend

Tabla 3, Actividades Sprint 1

Sprint 2	Integración sistema notificaciones push	26/08/2025	30/08/2025	5 días	Desarrollador Backend
Sprint 2	Desarrollo interfaz panel notificaciones	26/08/2025	30/08/2025	5 días	Desarrollador Frontend
Sprint 2	Configuración notificaciones en tiempo real	31/08/2025	04/09/2025	5 días	Desarrollador Backend
Sprint 2	Pruebas funcionales y de integración	05/09/2025	06/09/2025	2 días	Scrum Master

Tabla 4, Actividades Sprint 2

Sprint 6 - Despliegue	Monitoreo post-despliegue y ajustes	09/11/2025	15/11/2025	1 semana	Scrum Master
Sprint 6 - Despliegue	Elaboración documentación final	16/11/2025	22/11/2025	1 semana	Scrum Master
Sprint 6 - Despliegue	Entrega final del MVP	23/11/2025	25/11/2025	3 días	Product Owner

Tabla 8, Actividades Sprint 6

Sprint 0 - Preparación

⌚ 01/08/2025 - 10/08/2025 1.5 semanas

Objetivo: Análisis de requerimientos y diseño de arquitectura

Entregables:

- ✓ Documento de requerimientos funcionales y no funcionales
- ✓ Product Backlog priorizado
- ✓ Diagramas UML (casos de uso, clases, secuencia)
- ✓ Modelo de base de datos en Supabase
- ✓ Prototipos de alta fidelidad en Figma
- ✓ Plan de incrementos definido

Ilustración 2, Sprint 0

Sprint 1 - Autenticación e Inicio de Novedades

⌚ 11/08/2025 - 24/08/2025 2 semanas

Objetivo: Implementar sistema de autenticación y módulo de inicio de novedades

Entregables:

- ✓ Sistema de login con roles (Administrador, Residente, Celador)
- ✓ Módulo de inicio de novedades funcional
- ✓ Integración con Supabase para autenticación
- ✓ Pruebas unitarias y de integración
- ✓ Incremento 1 validado

Ilustración 3, Sprint 1

Sprint 2 - Notificaciones

⌚ 25/08/2025 - 07/09/2025 2 semanas

Objetivo: Implementar sistema de notificaciones push y en tiempo real

Entregables:

- ✓ Sistema de notificaciones push integrado
- ✓ Notificaciones en tiempo real con Supabase
- ✓ Panel de notificaciones por rol
- ✓ Pruebas de integración con Sprint 1
- ✓ Incremento 2 validado

Ilustración 4, Sprint 2

Sprint 3 - Menú Principal (Accesos, Reservas, Quejas, Mascotas)

⌚ 08/09/2025 - 21/09/2025 2 semanas

Objetivo: Implementar módulos principales de gestión

Entregables:

- ✓ Módulo de control de accesos y visitas
- ✓ Sistema de reservas de zonas comunes
- ✓ Módulo de quejas y sugerencias
- ✓ Registro de mascotas
- ✓ Pruebas funcionales de todos los módulos
- ✓ Incremento 3 validado

Ilustración 5, Sprint 3

Sprint 4 - Menú Principal (Perfil y Pagos)

⌚ 22/09/2025 - 05/10/2025 2 semanas

Objetivo: Implementar gestión de perfil y pagos de administración

Entregables:

- ✓ Módulo de perfil de usuario
- ✓ Sistema de pagos de administración
- ✓ Módulo de manejo de paquetería
- ✓ Integración completa de todos los módulos
- ✓ Pruebas de integración del sistema completo
- ✓ Incremento 4 validado

Ilustración 6, Sprint 4

Sprint 5 - Pruebas Finales y Ajustes

⌚ 06/10/2025 - 19/10/2025 2 semanas

Objetivo: Validación completa del MVP con usuarios reales

Entregables:

- ✓ Pruebas de aceptación con usuarios piloto
- ✓ Corrección de errores identificados
- ✓ Optimización de rendimiento

Ilustración 7, Sprint 5

Sprint 6 - Despliegue y Entrega

⌚ 20/10/2025 - 25/11/2025 5 semanas

Objetivo: Documentación final y Producto Mínimo Viable de la aplicación

Entregables:

- ✓ Documentación técnica completa
- ✓ Manual de Usuario por rol
- ✓ Aplicación funcional terminada

Ilustración 8, Sprint 6

9 Requerimientos

9.1 Requerimientos del Sistema

Funcional

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	 EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Diseño y creación de la arquitectura del sistema		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 1	FECHA: 06/08/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 3 días
PRE-REQUISITO DEL REQUERIMIENTO: Ninguno		
DESCRIPCION: Definir estructura general del aplicativo móvil, componentes principales y flujo de datos con <u>Supabase</u> .		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Debe garantizar escalabilidad y compatibilidad con <u>Kotlin</u> y Java.		
Reviso: _____	Aprobó: _____	

Requerimiento 1

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Modelado de base de datos en Supabase		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 2	FECHA: 08/08/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 2 días
PRE-REQUISITO DEL REQUERIMIENTO: Arquitectura definida		
DESCRIPCION: Crear las tablas necesarias para la gestión de usuarios, roles y datos operativos del conjunto.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Debe accederse mediante autenticación.		
Reviso: _____	Aprobó: _____	

Requerimiento 2

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Diseño de interfaz en Figma		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 3	FECHA: 06/08/2025
RESPONSABLE DEL REQUERIMIENTO: Luisa Fernanda Romero Suarez		TIEMPO REQUERIDO: 5 días
PRE-REQUISITO DEL REQUERIMIENTO: Ninguno		
DESCRIPCION: Diseñar las pantallas principales del aplicativo móvil según las necesidades del conjunto residencial.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguno		
Reviso: _____	Aprobó: _____	

Requerimiento 3

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO: 	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Implementación de autenticación con Supabase		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 4	FECHA: 14/08/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 4 días
PRE-REQUISITO DEL REQUERIMIENTO: Base de datos configurada		
DESCRIPCION: Configurar autenticación de usuarios con roles diferenciados (Administrador, Residente, Celador)		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Usuarios registrados podrán acceder.		
Reviso: _____	Aprobó: _____	

Requerimiento 4

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO: 	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Desarrollo de interfaces de login y roles		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 5	FECHA: 14/08/2025
RESPONSABLE DEL REQUERIMIENTO: Luisa Fernanda Romero Suarez		TIEMPO REQUERIDO: 4 días
PRE-REQUISITO DEL REQUERIMIENTO: Diseño aprobado en Figma		
DESCRIPCION: Implementar las pantallas de inicio de sesión y acceso según el rol asignado.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguna		
Reviso: _____	Aprobó: _____	

Requerimiento 5

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Implementación módulo de inicio de novedades		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 6	FECHA: 18/08/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 4 días
PRE-REQUISITO DEL REQUERIMIENTO: Autenticación funcional		
DESCRIPCION: Crear módulo que permita registrar y visualizar novedades del conjunto (por rol).		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguno		
Reviso: _____	Aprobó: _____	

Requerimiento 6

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Integración de sistema de notificaciones push		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 7	FECHA: 26/08/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 5 días
PRE-REQUISITO DEL REQUERIMIENTO: Autenticación funcional		
DESCRIPCION: Implementar el envío de notificaciones automáticas a los usuarios según su rol y evento generado.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Requiere conexión a Internet estable		
Reviso: _____	Aprobó: _____	

Requerimiento 7

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	 EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Desarrollo interfaz del panel de notificaciones		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 8	FECHA: 26/08/2025
RESPONSABLE DEL REQUERIMIENTO: Luisa Fernanda Romero Suarez		TIEMPO REQUERIDO: 5 días
PRE-REQUISITO DEL REQUERIMIENTO: Diseño de interfaz aprobado		
DESCRIPCION: Crear panel de visualización de notificaciones con conexión en tiempo real.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguna		
Reviso: _____	Aprobó: _____	

Requerimiento 8

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	 EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Desarrollo de módulo de control de accesos y visitas		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 9	FECHA: 09/09/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 4 días
PRE-REQUISITO DEL REQUERIMIENTO: Autenticación y roles implementados		
DESCRIPCION: Permitir registro de ingresos y salidas de visitantes, asociados al residente.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguna		
Reviso: _____	Aprobó: _____	

Requerimiento 9

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	 San Rafael 24H	
NOMBRE DEL PROYECTO: Desarrollo del sistema de reservas		
ANALISTA RESPONSABLE: Scrum Adaptable	REQUERIMIENTO No.: 10	FECHA: 13/09/2025
RESPONSABLE DEL REQUERIMIENTO: Luisa Fernanda Romero Suarez		TIEMPO REQUERIDO: 4 días
PRE-REQUISITO DEL REQUERIMIENTO: Control de accesos operativo		
DESCRIPCION: Permitir a los residentes reservar espacios como la piscina, salón comunal, gimnasio y zona bbq.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: No permitir reservas simultáneas del mismo espacio.		
Reviso: _____	Aprobó: _____	

Requerimiento 10

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	 San Rafael 24H	
NOMBRE DEL PROYECTO: Desarrollo de módulo de quejas y registro de mascotas		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 11	FECHA: 17/09/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 3 días
PRE-REQUISITO DEL REQUERIMIENTO: Autenticación y residentes registrados		
DESCRIPCION: Implementar formulario de quejas y registro de mascotas con almacenamiento en Supabase.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguno		
Reviso: _____	Aprobó: _____	

Requerimiento 11

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Desarrollo del módulo de perfil de usuario		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 12	FECHA: 23/09/2025
RESPONSABLE DEL REQUERIMIENTO: Luisa Fernanda Romero Suarez		TIEMPO REQUERIDO: 4 días
PRE-REQUISITO DEL REQUERIMIENTO: Autenticación establecida		
DESCRIPCION: Permitir la edición de datos personales y ver información del usuario.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguno		
Reviso: _____	Aprobó: _____	

Requerimiento 12

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Implementación del sistema de pagos PSE		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 13	FECHA: 27/09/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 5 días
PRE-REQUISITO DEL REQUERIMIENTO: Perfil de usuario activo		
DESCRIPCION: Integrar opción de pago en línea para cuotas de administración del conjunto.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Requiere pasarela PSE validada y conexión HTTPS.		
Reviso: _____	Aprobó: _____	

Requerimiento 13

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Implementación del módulo de manejo de paquetería		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 14	FECHA: 27/09/2025
RESPONSABLE DEL REQUERIMIENTO: Luisa Fernanda Romero Suarez		TIEMPO REQUERIDO: 5 días
PRE-REQUISITO DEL REQUERIMIENTO: Autenticación y roles configurados		
DESCRIPCION: Permitir a celadores registrar recepción y entrega de paquetes a residentes.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Solo el rol celador puede generar entregas		
Reviso: _____ _____	Aprobó: _____ _____	

Requerimiento 14

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Integración completa del sistema		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 15	FECHA: 02/10/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 2 días
PRE-REQUISITO DEL REQUERIMIENTO: Todos los módulos previos desarrollados		
DESCRIPCION: Unificar módulos en una única aplicación móvil conectada a Supabase.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Validar comunicación entre módulos antes del despliegue.		
Reviso: _____ _____	Aprobó: _____ _____	

Requerimiento 15

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	 EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Gestión de visitantes con QR		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 25	FECHA: 06/10/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 4 días
PRE-REQUISITO DEL REQUERIMIENTO:		
DESCRIPCION: Generar códigos QR de un solo uso para visitantes, con vigencia limitada y asociada al residente.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: X NO FUNCIONAL: ____		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguna		
Reviso: ____	Aprobó: ____	

Requerimiento 25

No funcional

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	 EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Integridad de datos		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 16	FECHA: 02/10/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 1 día
PRE-REQUISITO DEL REQUERIMIENTO: Base de datos sincronizada		
DESCRIPCION: Asegurar que la información registrada no se pierda ni duplique durante transacciones.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: ____ NO FUNCIONAL: X		
RESTRICCIONES DEL FUNCIONAMIENTO: Las operaciones deben ser atómicas (transaccionales).		
Reviso: ____	Aprobó: ____	

Requerimiento 16

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
		
NOMBRE DEL PROYECTO: Usabilidad en dispositivos Android		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 17	FECHA: 13/09/2025
RESPONSABLE DEL REQUERIMIENTO: Luisa Fernanda Romero Suarez		TIEMPO REQUERIDO: 4 días
PRE-REQUISITO DEL REQUERIMIENTO:		
DESCRIPCION: Garantizar que la aplicación sea intuitiva, con diseño accesible y navegación fluida.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input type="checkbox"/> NO FUNCIONAL: <input checked="" type="checkbox"/> X		
RESTRICCIONES DEL FUNCIONAMIENTO: Debe soportar Android 9 o superior.		
Reviso: <hr/>	Aprobó: <hr/>	

Requerimiento 17

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
		
NOMBRE DEL PROYECTO: Rendimiento del sistema (tiempo de respuesta)		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 18	FECHA: 17/10/2025
RESPONSABLE DEL REQUERIMIENTO: Luisa Fernanda Romero Suarez		TIEMPO REQUERIDO: 2 días
PRE-REQUISITO DEL REQUERIMIENTO: Sistema completo integrado		
DESCRIPCION: Asegurar que las pantallas carguen en menos de 3 segundos con conexión estable.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input type="checkbox"/> NO FUNCIONAL: <input checked="" type="checkbox"/> X		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguno		
Reviso: <hr/>	Aprobó: <hr/>	

Requerimiento 18

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Documentación técnica y control de calidad		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 19	FECHA: 16/11/2025
RESPONSABLE DEL REQUERIMIENTO: Luisa Fernanda Romero Suarez		TIEMPO REQUERIDO: 1 semana
PRE-REQUISITO DEL REQUERIMIENTO:		
DESCRIPCION: Elaborar documentación técnica que asegure trazabilidad y reproducibilidad del sistema.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input type="checkbox"/> NO FUNCIONAL: <input checked="" type="checkbox"/>		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguno		
Reviso: <hr/>	Aprobó: <hr/>	

Requerimiento 19

9.2 Requerimientos de Seguridad

Funcional

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Validar entrada de datos en formularios		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 21	FECHA: 22/11/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 4 días
PRE-REQUISITO DEL REQUERIMIENTO: Módulos formularios listos		
DESCRIPCION: Se restringirá la inyección de código y datos inválidos.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: <input type="checkbox"/>		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguno		
Reviso: <hr/>	Aprobó: <hr/>	

Requerimiento 21

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
		
NOMBRE DEL PROYECTO: Cumplimiento de la Ley 1581 de 2012 (protección de datos personales)		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 22	FECHA: 10/11/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 2 días
PRE-REQUISITO DEL REQUERIMIENTO: Políticas de privacidad definidas		
DESCRIPCION: Garantizar el tratamiento y almacenamiento seguro de datos personales.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguno		
Reviso: _____	Aprobó: _____	

Requerimiento 22

No funcional

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
		
NOMBRE DEL PROYECTO: Gestión de contraseñas seguras		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 26	FECHA: 02/11/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 3 días
PRE-REQUISITO DEL REQUERIMIENTO:		
DESCRIPCION: Contraseñas deben tener longitud mínima 8 caracteres		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: _____ NO FUNCIONAL: <input checked="" type="checkbox"/>		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguna		
Reviso: _____	Aprobó: _____	

Requerimiento 26

9.3 Requerimientos de Fiabilidad

Funcional

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Control de transacciones seguras en el módulo de pagos		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 23	FECHA: 26/12/2025
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 1 semana
PRE-REQUISITO DEL REQUERIMIENTO: Integración PSE funcional		
DESCRIPCION: Garantizar que cada transacción se complete sin duplicaciones o pérdidas de información.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: <input checked="" type="checkbox"/> NO FUNCIONAL: _____		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguno		
Reviso: _____	Aprobó: _____	

Requerimiento 23

No funcional

CAPTURA Y DESCRIPCION DE REQUERIMIENTOS		
LOGO:	EMPRESA: San Rafael 24H	
NOMBRE DEL PROYECTO: Escalabilidad horizontal del sistema		
ANALISTA RESPONSABLE: Scrum Master	REQUERIMIENTO No.: 24	FECHA: 29/01/2026
RESPONSABLE DEL REQUERIMIENTO: Omar Santiago Rodríguez Puerto		TIEMPO REQUERIDO: 5 días
PRE-REQUISITO DEL REQUERIMIENTO: Infraestructura modular lista		
DESCRIPCION: Asegurar que se puedan agregar nuevos módulos o usuarios sin afectar el rendimiento.		
TIPO DE REQUERIMIENTO:		
FUNCIONAL: _____ NO FUNCIONAL: <input checked="" type="checkbox"/>		
RESTRICCIONES DEL FUNCIONAMIENTO: Ninguna		
Reviso: _____	Aprobó: _____	

Requerimiento 24

10 Diseño Arquitectónico

En este capítulo se describe la arquitectura general del sistema propuesto, así como el diseño orientado a objetos que respalda su funcionamiento. Se presentan las decisiones técnicas adoptadas, el enfoque arquitectónico utilizado y los elementos estructurales que permiten garantizar modularidad, escalabilidad y claridad en el desarrollo del software.

10.1 Arquitectura del sistema

Este proyecto se estructuró bajo una arquitectura modular, donde se separan las responsabilidades en:

- **Desarrollo incremental por módulos:** Sprint 0 - Sprint 6
- **Integración progresiva:** Cada sprint entrega incrementos modulares diferentes
- **Facilidad de Mantenimiento y Escalabilidad:** Se puede actualizar o corregir errores en un módulo específico sin afectar todo el sistema

10.1.1 Patrones de Diseño

La arquitectura del aplicativo se basa en un **patrón modular**, lo que significa que la aplicación se organiza en componentes independientes, cada uno responsable de una funcionalidad específica. Esta separación facilita el mantenimiento, la escalabilidad y las pruebas, ya que los módulos pueden desarrollarse, validarse e incluso reemplazarse sin afectar al resto del sistema.

En este esquema, cada módulo cumple un rol bien definido: gestión de novedades, notificaciones, paquetería, pagos y menú principal (accesos, reservas, quejas, mascotas y perfil de usuario). Todos estos módulos interactúan de forma coordinada a través de interfaces claras, permitiendo que la aplicación evolucione de manera progresiva mediante la incorporación de nuevas funcionalidades o mejoras, sin comprometer la estabilidad general.

10.1.2 Diseño Orientado a objetos

El desarrollo de la aplicación móvil para el Conjunto Residencial Hacienda San Rafael utiliza un modelo de datos relacional representado mediante diagramas UML que consta de 11 entidades principales que mapean la estructura de la base de datos en Supabase. Este modelo define las entidades del dominio del negocio (Usuario, Visitante, Mascota, Paquetería, Queja, Reserva Zona Común, Notificación, Pago Administración, Acceso Peatonal, Acceso Vehicular y Vehículo Residente) y sus relaciones mediante claves primarias y foráneas.

El diagrama de clases UML proporciona una vista estructural del sistema, especificando los atributos de cada entidad, sus tipos de datos, restricciones de integridad y las cardinalidades de las relaciones (uno a uno, uno a muchos, muchos a muchos). Por ejemplo, un Usuario puede tener múltiples accesos, un Residente puede generar múltiples reservas, y cada Reserva pertenece a un Usuario específico.

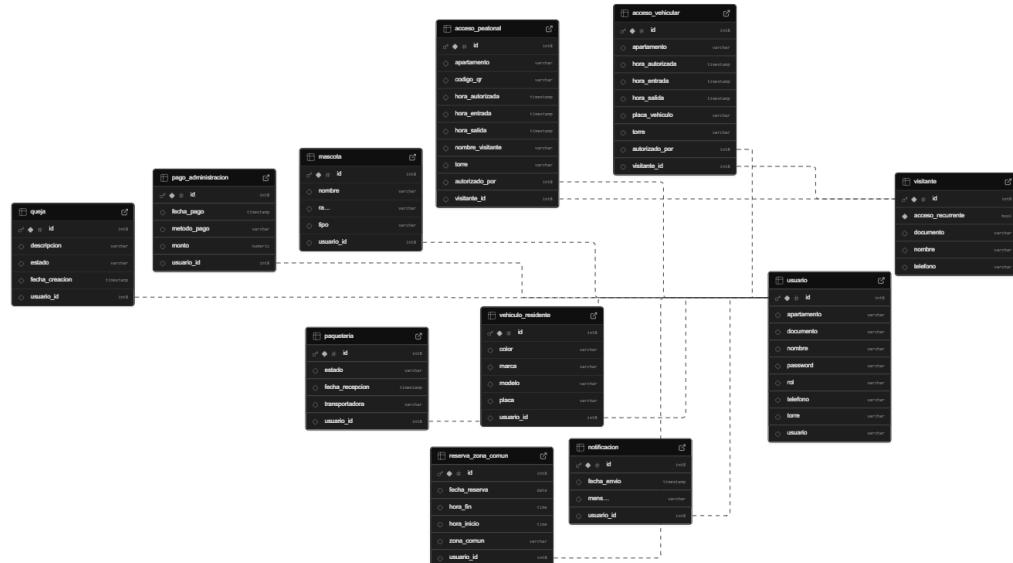


Diagrama 1, UML

CAPÍTULO IV

11 Seguridad

11.1 Matriz de Riesgos

⚠ Identificación de Riesgos del Proyecto

Aplicación Móvil para Gestión del Conjunto Residencial Hacienda San Rafael



Ilustración 9, Identificación de Riesgos

TIPO DE RIESGO	IMPACTO EN	RIESGO POSIBLE
Técnico	Proyecto	Incompatibilidad entre versiones de Android Studio, IntelliJ IDEA y las dependencias de Supabase, generando errores de compilación y retrasos en el desarrollo.
Técnico	Creación de aplicación	Problemas de integración entre el frontend Android y el backend Supabase, causando fallas en la autenticación, consultas de datos o notificaciones push.
Técnico	Aplicación en funcionamiento	Errores de sincronización en tiempo real de la base de datos cuando múltiples usuarios (administrador, residentes, celadores) accedan simultáneamente, provocando inconsistencias en los datos.
De Gestión	Proyecto	Retrasos en la entrega de incrementos por mala estimación de tiempos en los sprints, comprometiendo la fecha final de entrega del MVP (25 de noviembre de 2025).
De Gestión	Proyecto	Cambios continuos en los requerimientos por parte del Product Owner después de iniciados los sprints, generando retrabajos y desviaciones del plan incremental.
Económico	Proyecto	Costos no previstos en servicios de Supabase si se exceden los límites del plan gratuito debido al volumen de datos o consultas en tiempo real durante las pruebas.
Económico	Proyecto	Necesidad de adquirir dispositivos Android adicionales para pruebas en diferentes versiones del sistema operativo, incrementando costos no contemplados inicialmente.
De Seguridad	Creación de aplicación	Vulnerabilidades en el sistema de autenticación que permitan accesos no autorizados a datos sensibles de residentes, comprometiendo la privacidad y confidencialidad.
Externo	Aplicación en funcionamiento	Caidas del servicio de Supabase por mantenimientos no programados o problemas en su infraestructura, dejando la aplicación sin acceso a datos en momentos críticos de operación del conjunto.
Externo	Aplicación en funcionamiento	Cambios en la normativa de protección de datos personales (Ley 1581 de 2012 en Colombia) que exijan modificaciones en el manejo y almacenamiento de información de residentes.

Tabla 9, Matriz de Riesgo

8	[De Seguridad] Vulnerabilidades en el sistema de autenticación que permitan accesos no autorizados a datos sensibles de residentes, comprometiendo la privacidad y confidencialidad.	Prevenir	Implementar autenticación segura con Supabase Auth (JWT y refresh tokens). Aplicar políticas de Row Level Security (RLS) en todas las tablas. Realizar auditorías de seguridad en code reviews. Implementar MFA opcional para administrador.
9	[Externo] Caídas del servicio de Supabase por mantenimientos no programados o problemas en su infraestructura, dejando la aplicación sin acceso a datos en momentos críticos.	Aceptar	Implementar manejo de errores robusto con mensajes amigables al usuario. Configurar reintentos automáticos con backoff exponencial. Suscribirse al status page de Supabase. Documentar requerimiento de conexión a internet en manual de usuario.
10	[Externo] Cambios en la normativa de protección de datos personales (Ley 1581 de 2012 en Colombia) que exijan modificaciones en el manejo y almacenamiento de información.	Monitorear	Implementar mejoras prácticas de privacidad desde el inicio: recolección mínima de datos, cifrado, almacenamiento seguro en Supabase. Incluir políticas de privacidad y funcionalidades para derechos de usuarios. Scrum Master revisa actualizaciones normativas trimestralmente.

Tabla 11, Matriz de Riesgo

11.4 Marcos de Referencia

11.4.1 Marco Legal y Normativo

Ley 1581 de 2012: “Protección de Datos Personales” La presente ley tiene por objeto desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos y se aplica junto con el Decreto 1377 de 2013, que reglamenta su aplicación práctica.

Ley 1266 de 2008: “Habeas Data Financiero” Esta ley regula el manejo de la información financiera, crediticia, comercial y de servicios de las personas, estableciendo el derecho al habeas data aplicado al módulo de Pago Administración.

Ley 527 de 1999: “Comercio Electrónico” Esta ley regula el comercio electrónico en Colombia y establece disposiciones sobre la validez legal de los contratos electrónicos, la protección al consumidor en transacciones en línea y la responsabilidad de los proveedores de servicios en línea.

Ley 1341 de 2009: “TIC” Promueve el desarrollo y uso responsable de las Tecnologías de la Información y las Comunicaciones en Colombia.

Ley 1263 de 2013: “Delitos Informáticos” Tipifica los delitos relacionados con la violación de datos personales, acceso no autorizado a sistemas informáticos y manipulación de información.

ISO/IEC 27001: Gestión de la seguridad para proteger la confidencialidad, integridad y disponibilidad de la información mediante la implementación de controles de seguridad técnicos, físicos y legales.

ISO/IEC 25010: Calidad del software (funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad).

ISO 31000: Gestión del riesgo que proporciona un marco para identificar, analizar, evaluar, tratar, supervisar y comunicar los riesgos de manera sistemática.

11.4.2 Marco de Calidad y Pruebas

Para este proyecto se establecen los lineamientos necesarios para garantizar que el aplicativo desarrollado para el conjunto residencial cumpla con altos estándares de desempeño, seguridad, usabilidad y fiabilidad. Se basa principalmente en las normas ISO/IEC 25010, que define los criterios de calidad del software, e ISO/IEC 29119, que orienta el proceso de pruebas. A lo largo del desarrollo, se aplican pruebas unitarias, de integración, funcionales, de rendimiento y de seguridad, con el fin de validar que cada módulo (como pagos, notificaciones, reservas y perfil) funcione correctamente y cumpla con los requerimientos del usuario.

CAPÍTULO V

13 Exposición de Resultados

13.1 Prototipo Interfaz

Administrador

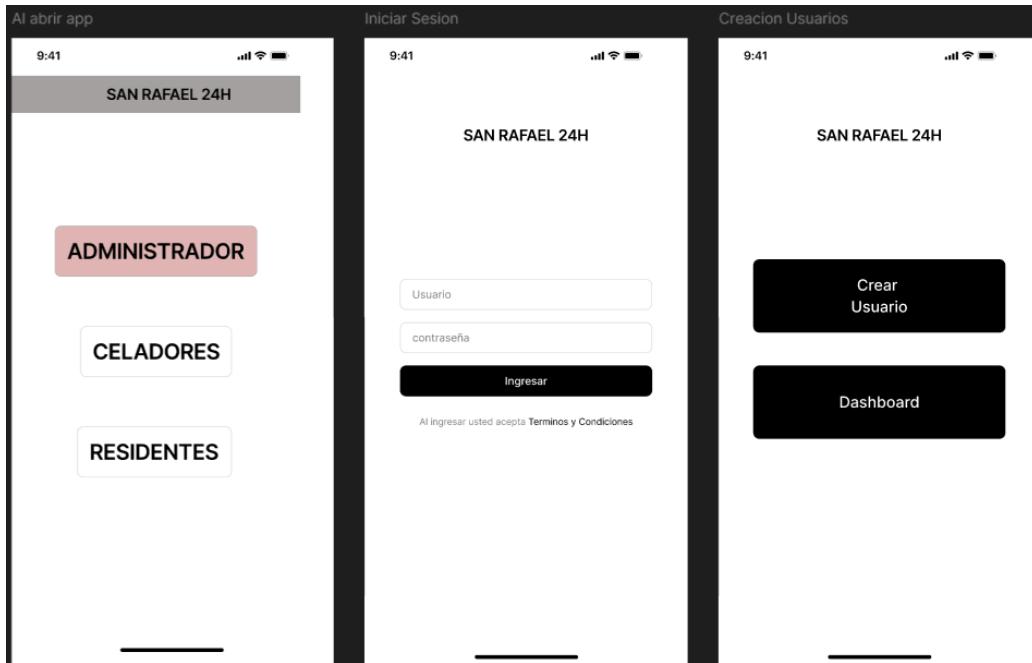


Ilustración 14, Inicio Sesión Administrador

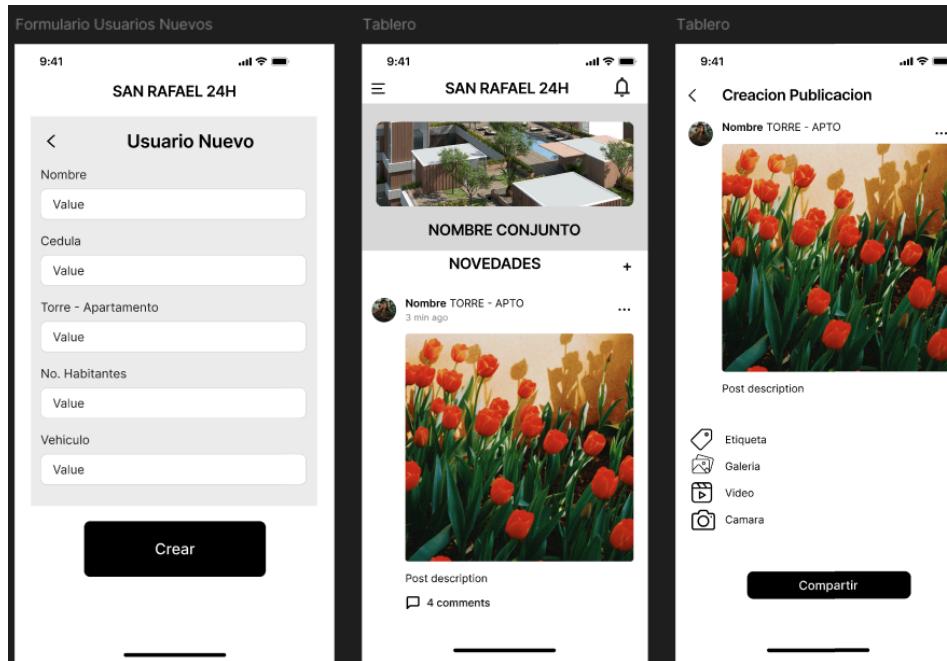


Ilustración 15, Creación Usuario e Inicio de Novedades

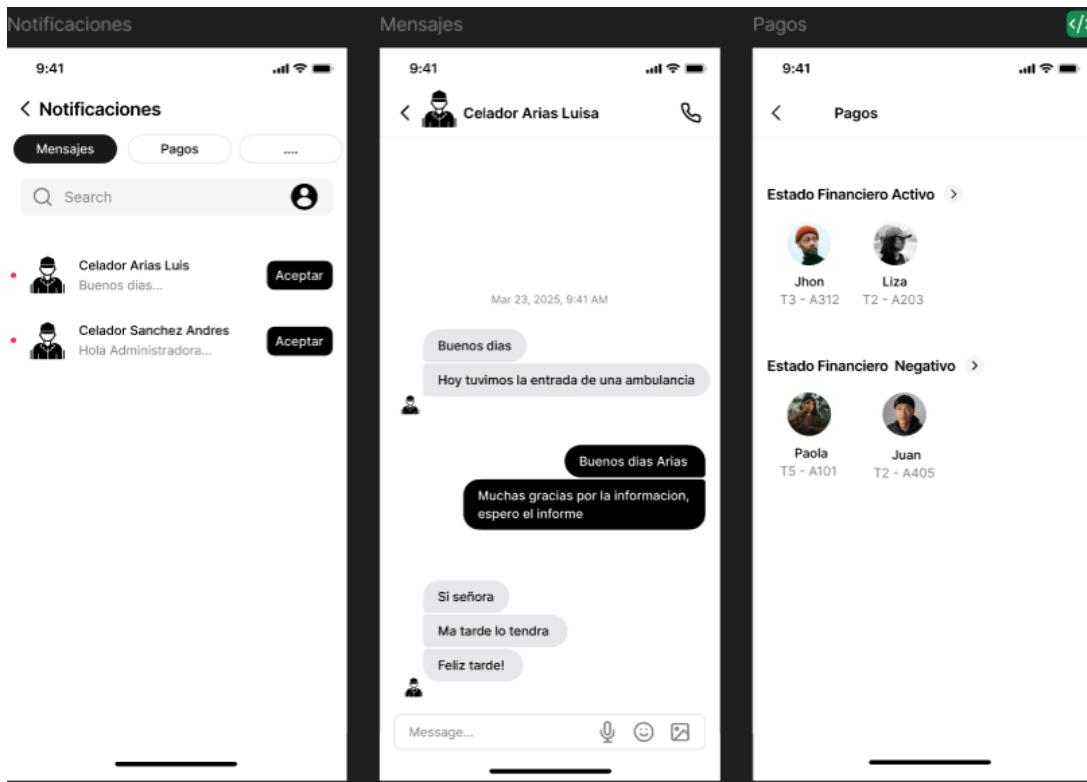


Ilustración 16, Notificaciones Mensajes y Pagos

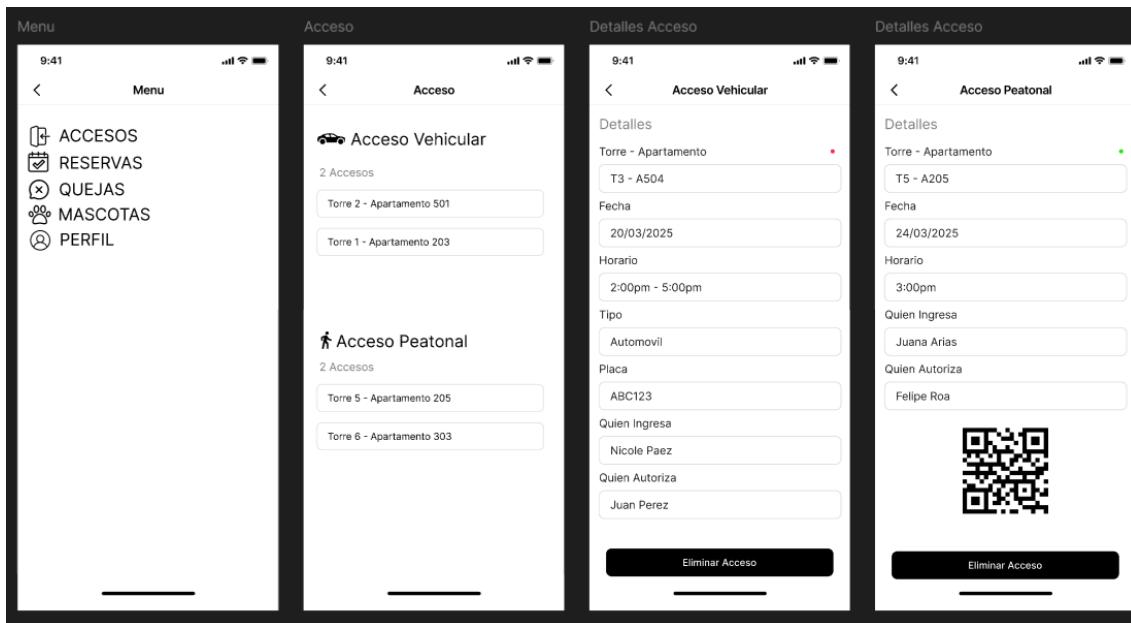


Ilustración 17, Menú Principal (Accesos)

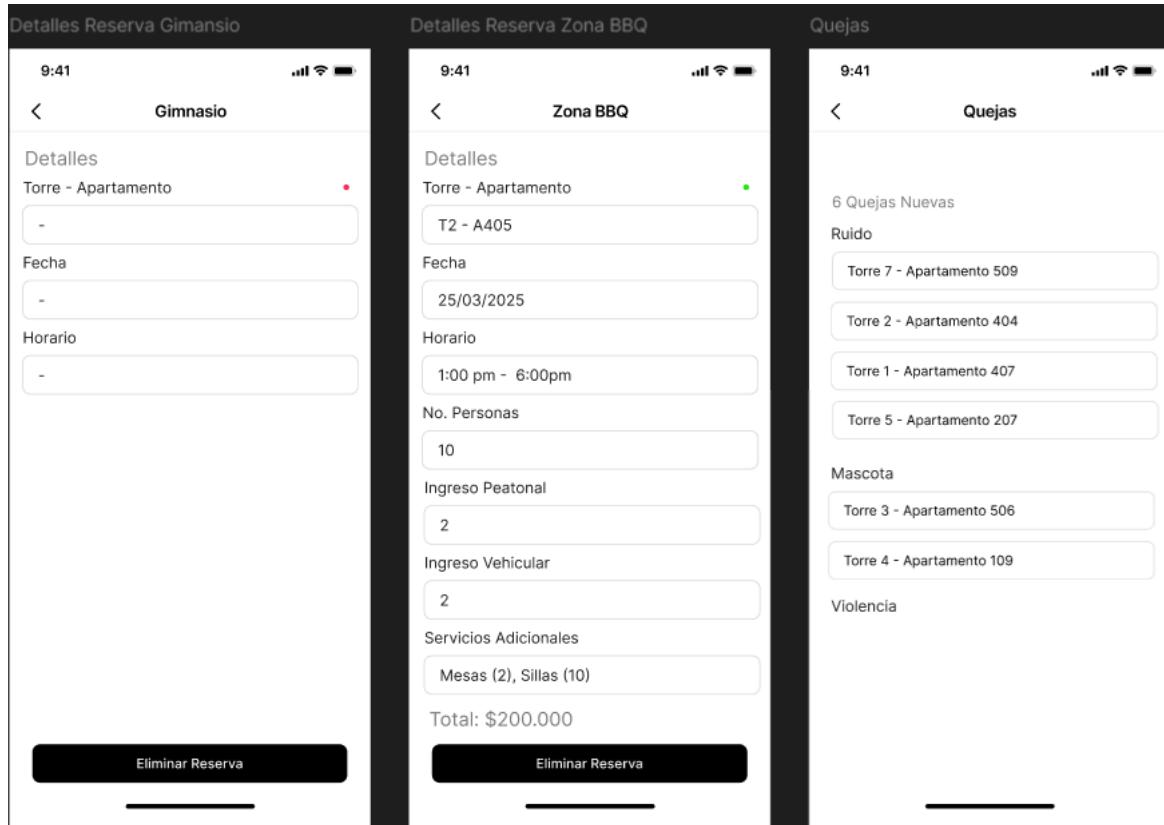


Ilustración 18, Menú Principal (Reservas)

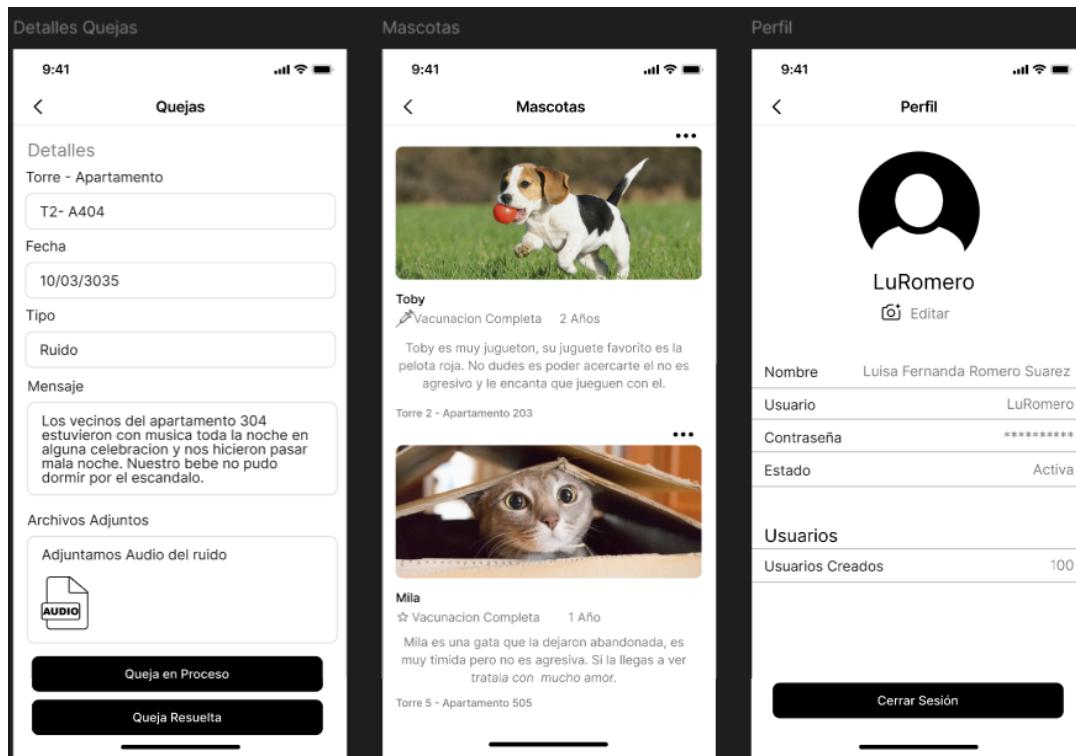


Ilustración 19, Menú Principal (Quejas, Mascotas y Perfil)

Celadores

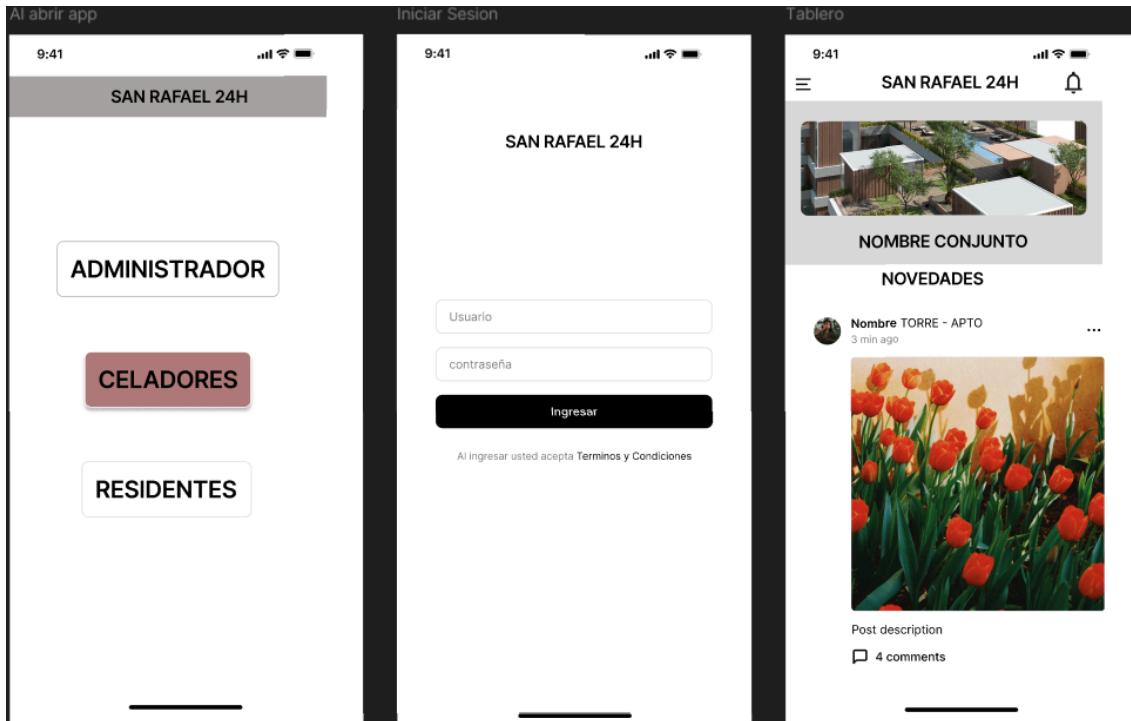


Ilustración 20, Inicio Sesión Celador e Inicio Novedades

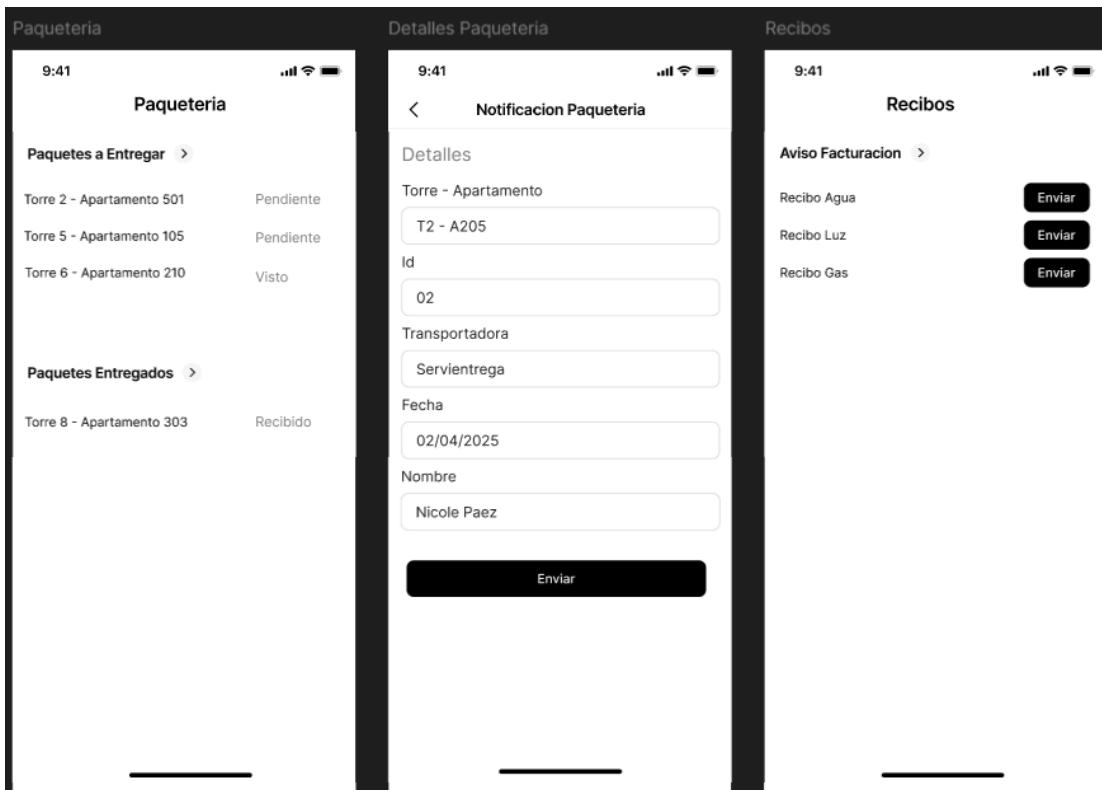


Ilustración 21, Inicio Sesión Celador e Inicio Novedades

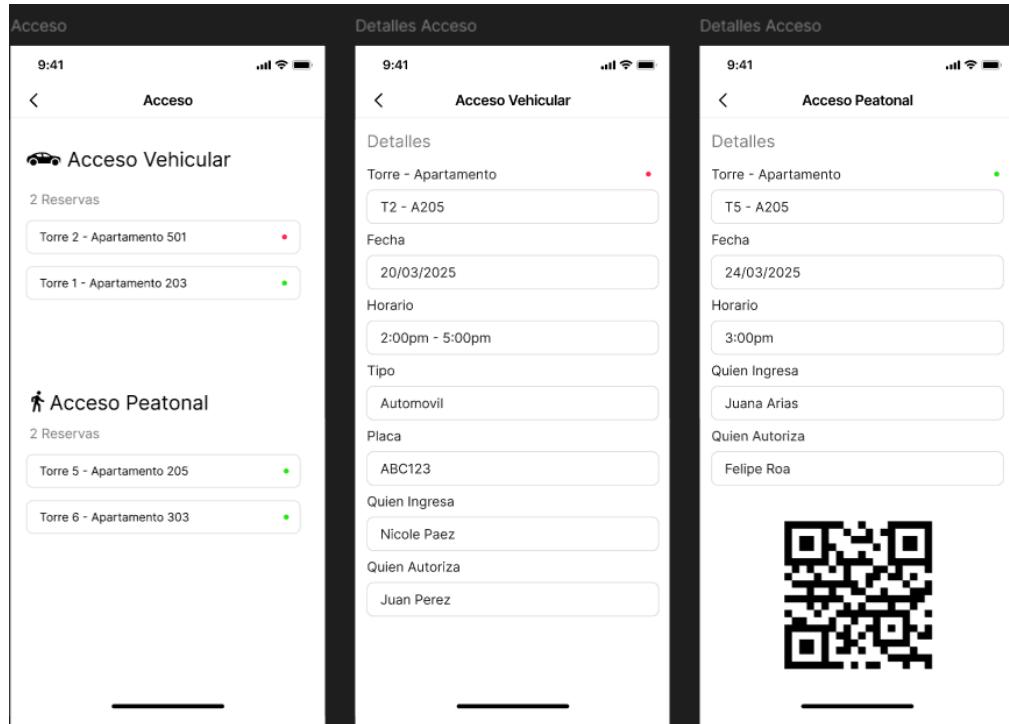


Ilustración 22, Menú Principal (Accesos)

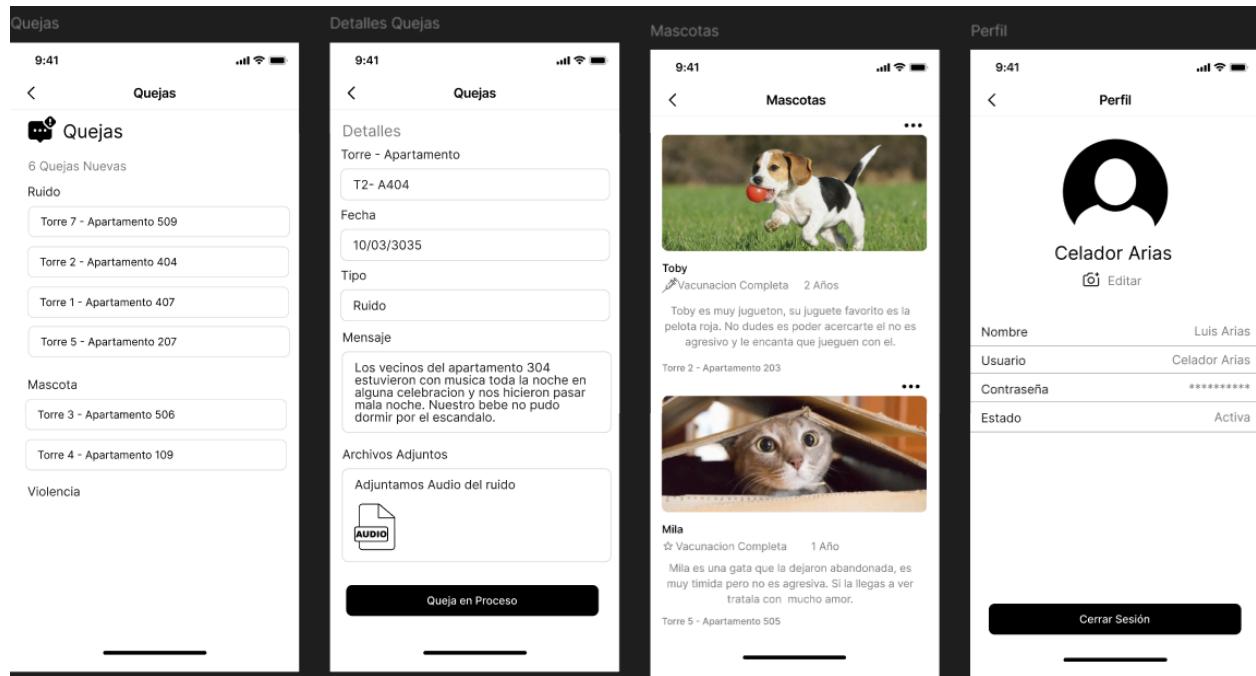


Ilustración 23, Menú Principal (Quejas, Mascotas)

Residente

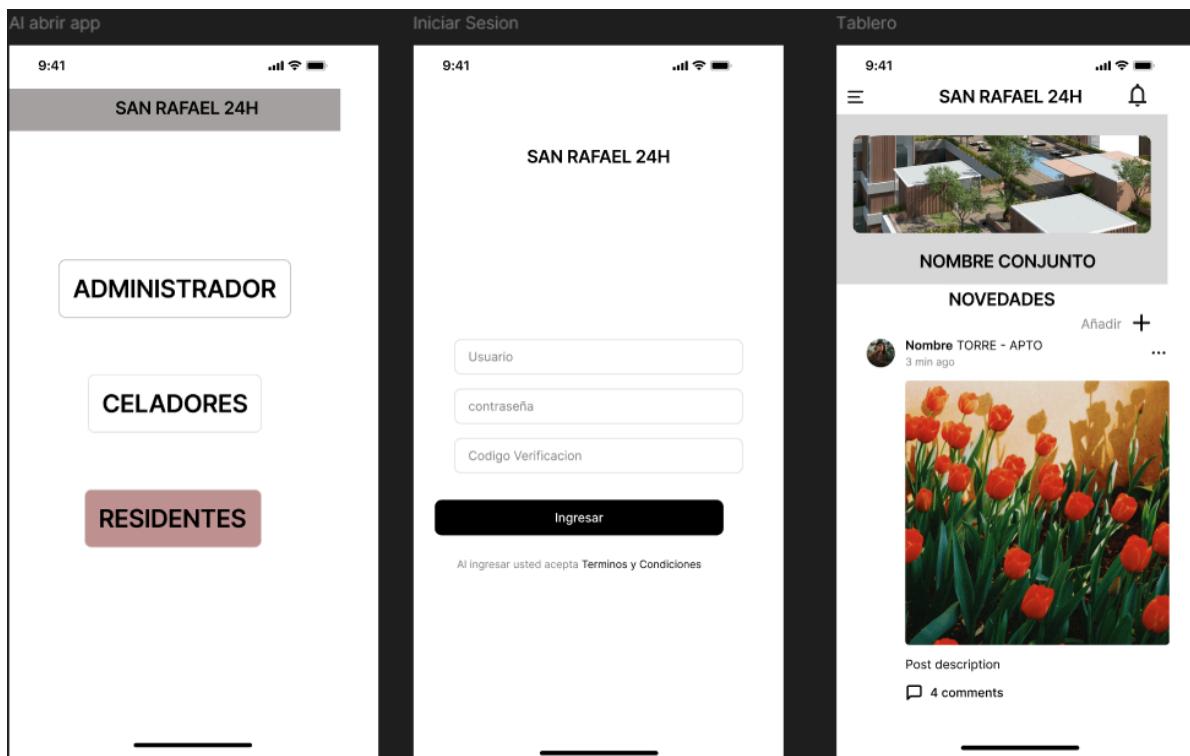


Ilustración 24, Inicio Sesión Residentes

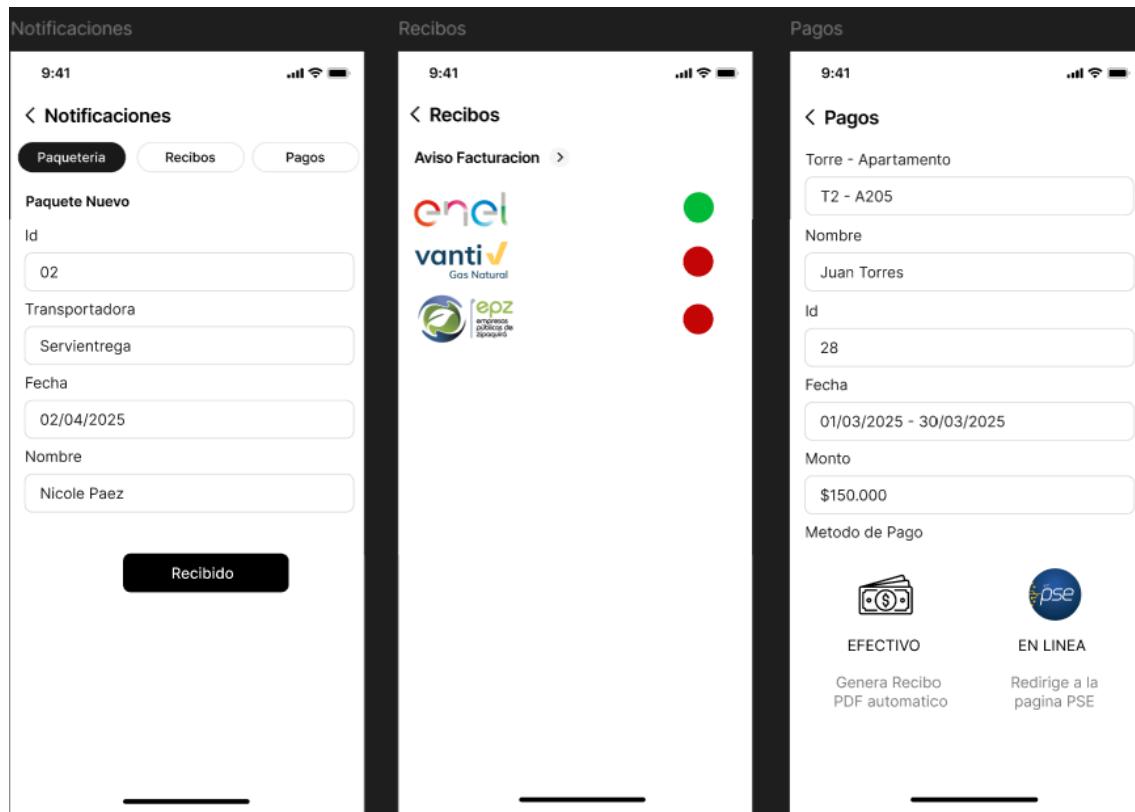


Ilustración 25, Notificaciones (Paquetería, Recibos y Pagos)

Ilustración 26, Accesos(Vehicular y Peatonal)
Ilustración 27, Reservas (Piscina, Salón Comunal, Gimnasio y Zona BBQ)

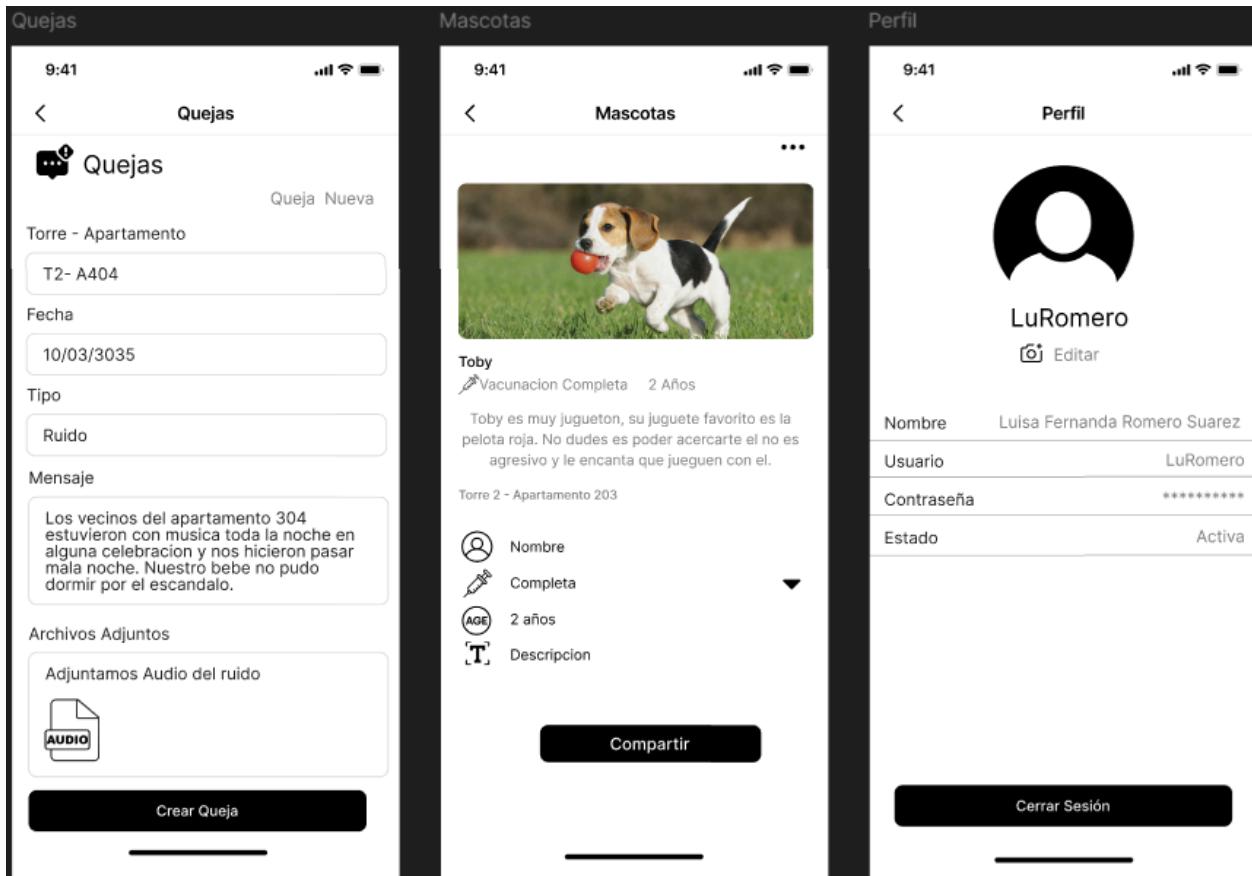


Ilustración 28, Reservas (Quejas, Mascotas y Perfil)

13.2 Casos de Uso

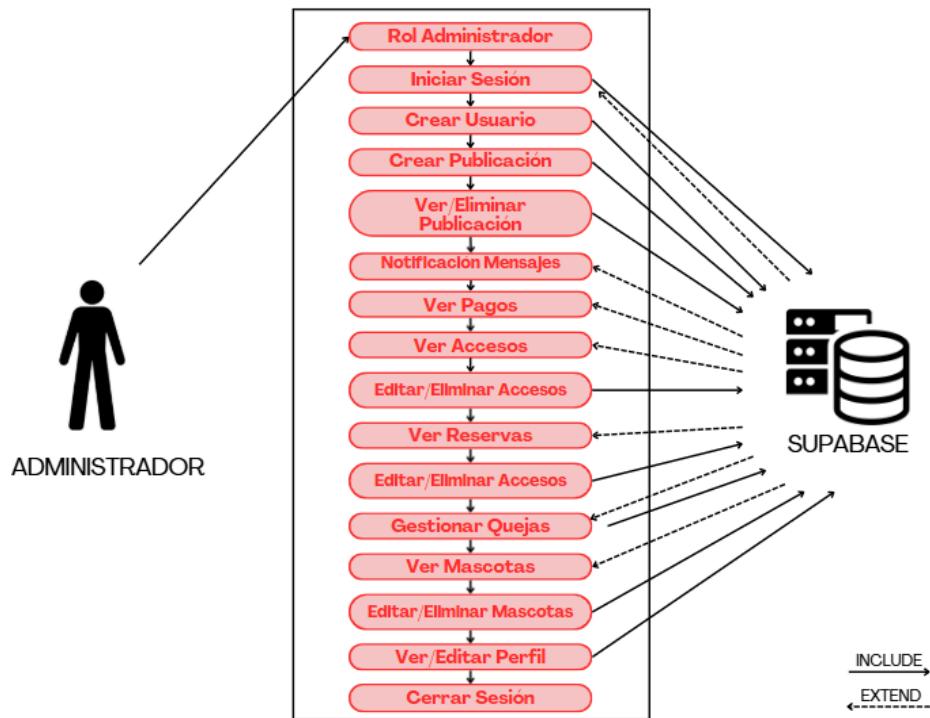


Ilustración 29, Caso de Uso Administrador

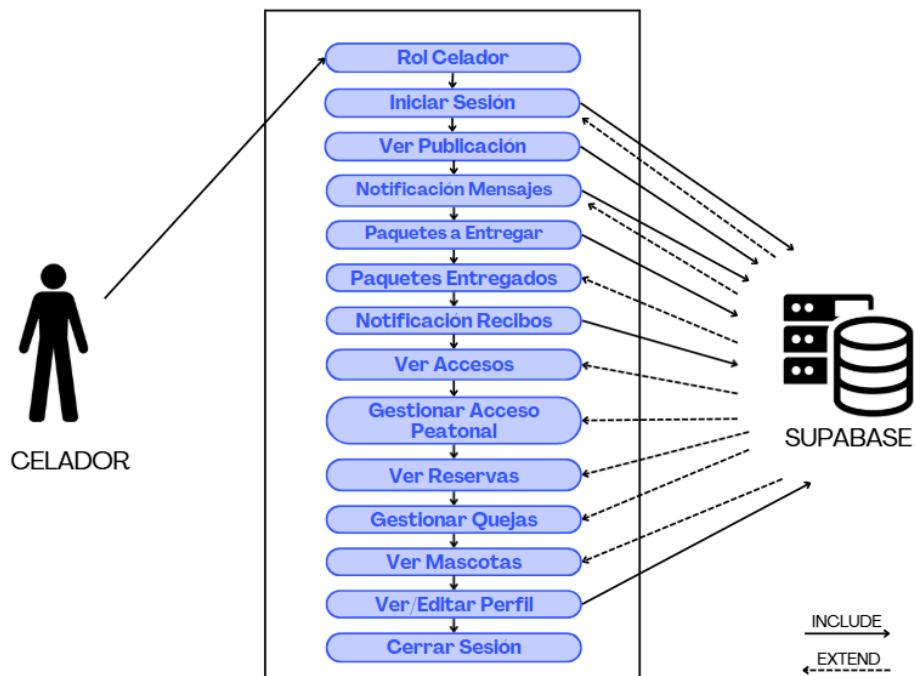


Ilustración 30, Caso de Uso Celador

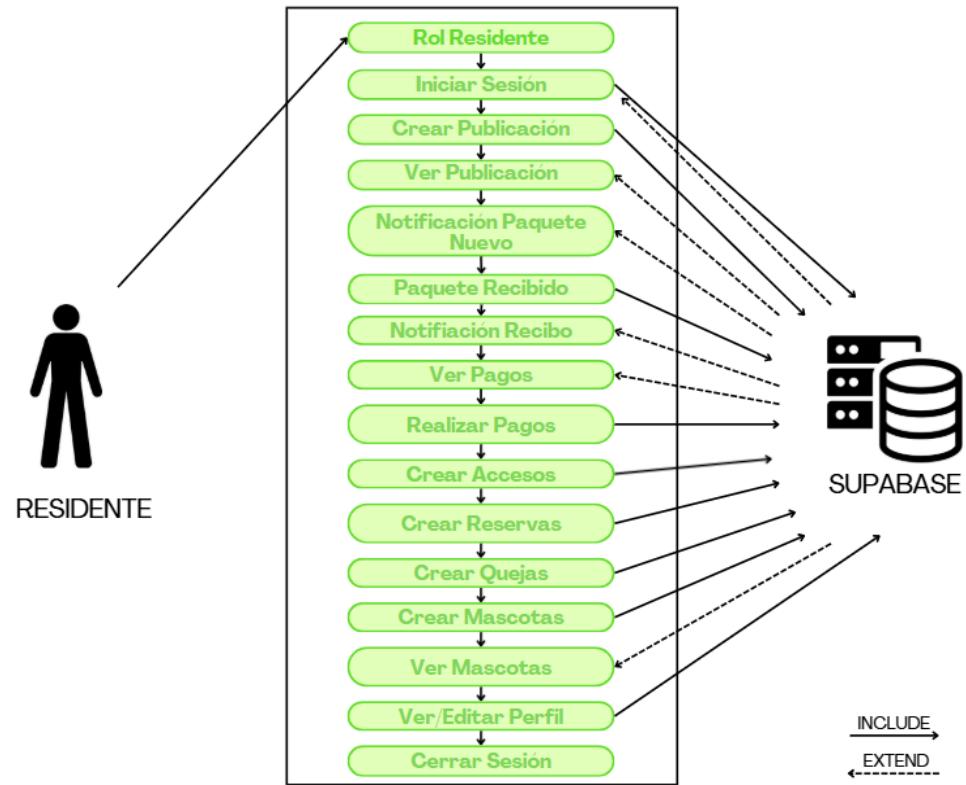


Ilustración 31, Caso de Uso Residente

13.3 Diagramas de Secuencia

Administrador

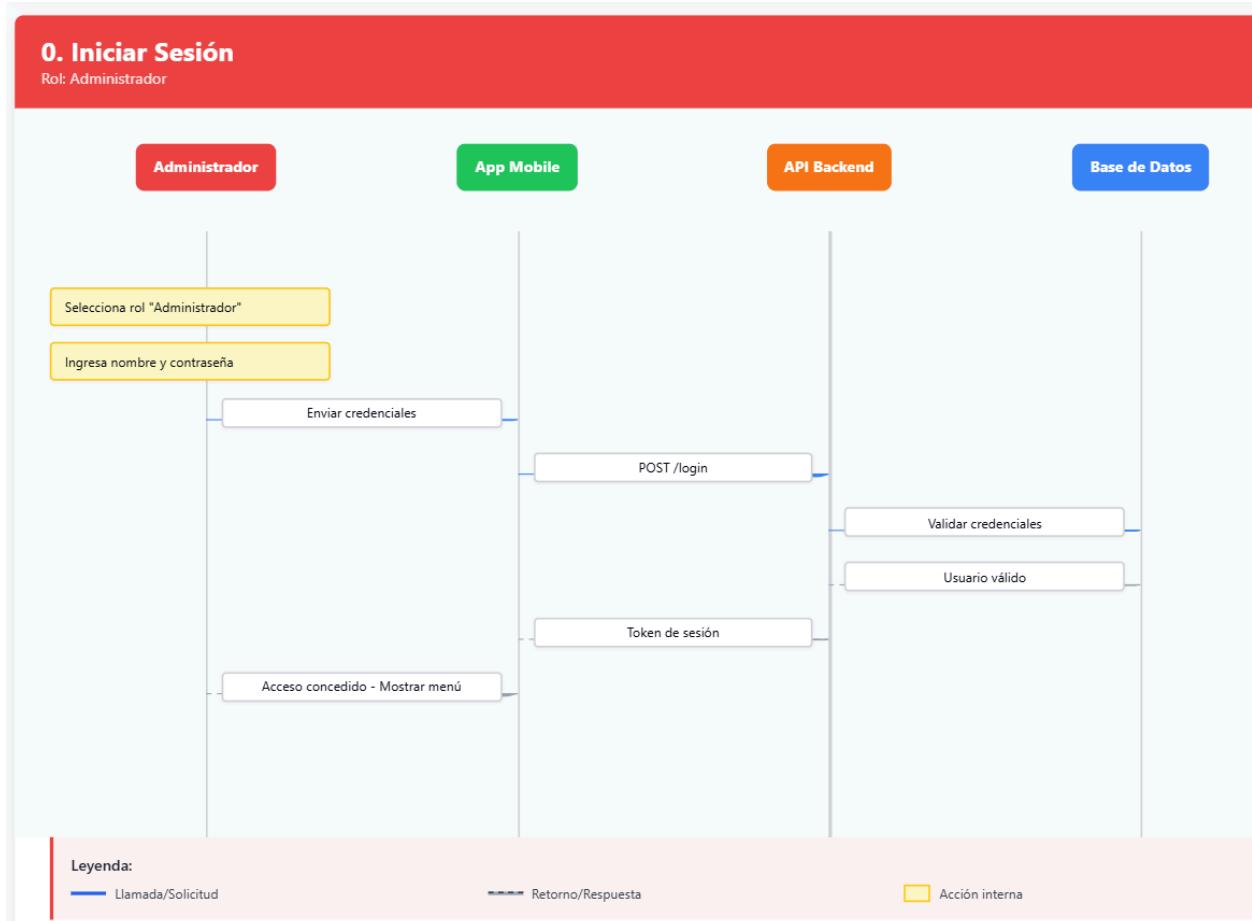


Diagrama 2, Diagrama de secuencia Inicio Sesión Administrador

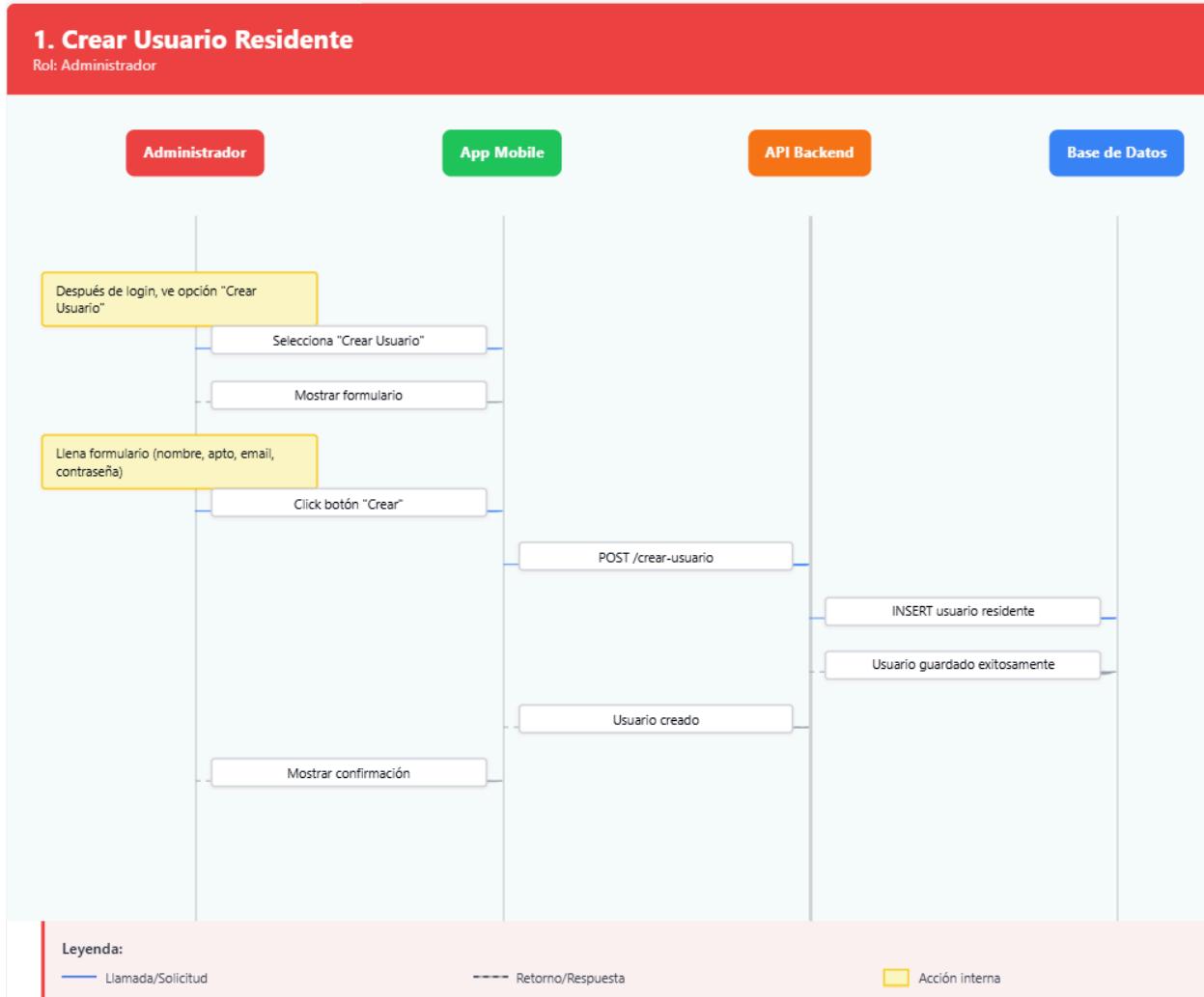


Diagrama 3, Diagrama de secuencia Crear Usuario Administrador

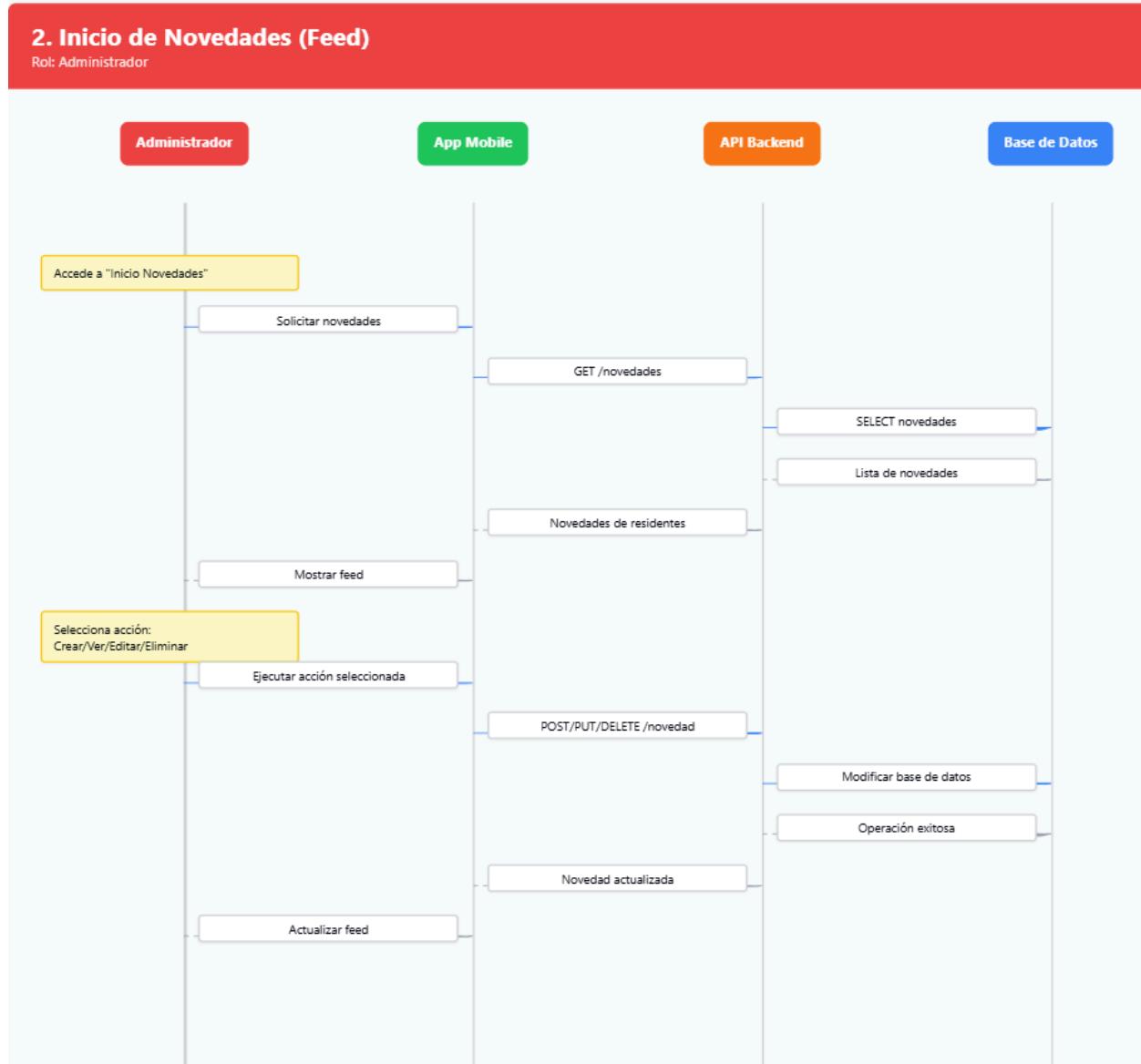


Diagrama 4, Diagrama de secuencia Inicio Novedades Administrador

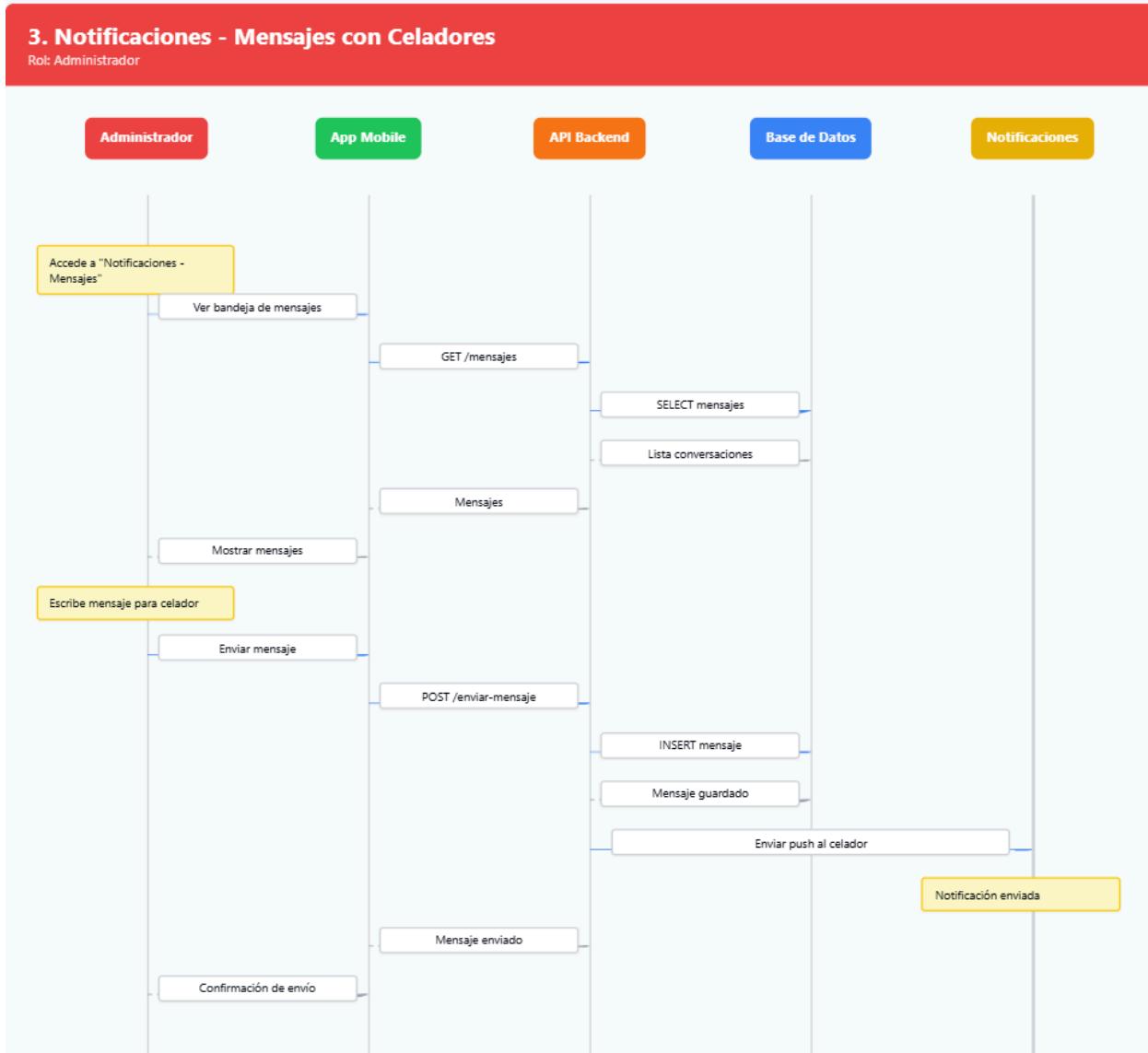


Diagrama 5, Diagrama de secuencia Notificaciones (Mensajes) Administrador

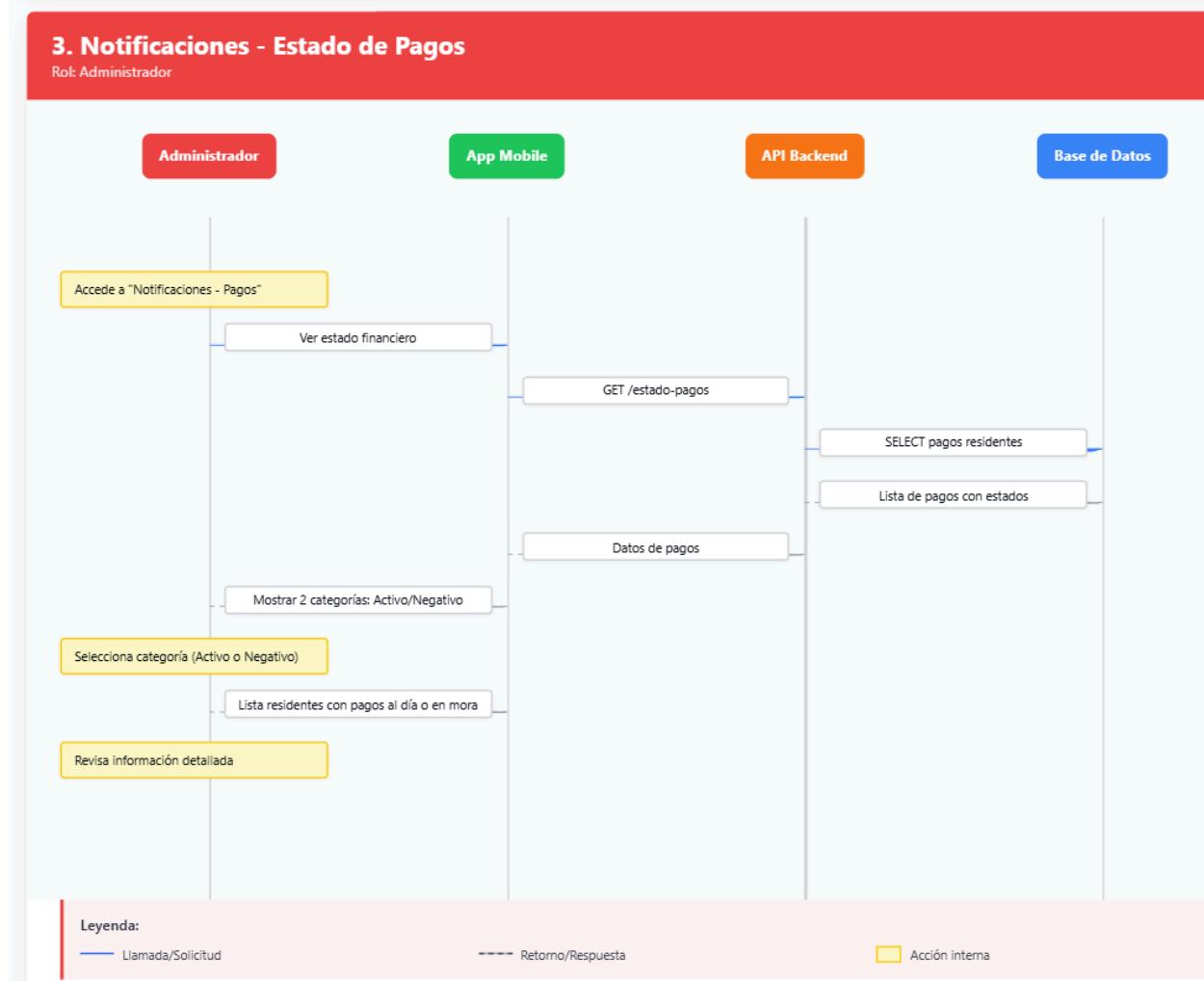


Diagrama 6, Diagrama de secuencia Notificaciones (Pagos) Administrador

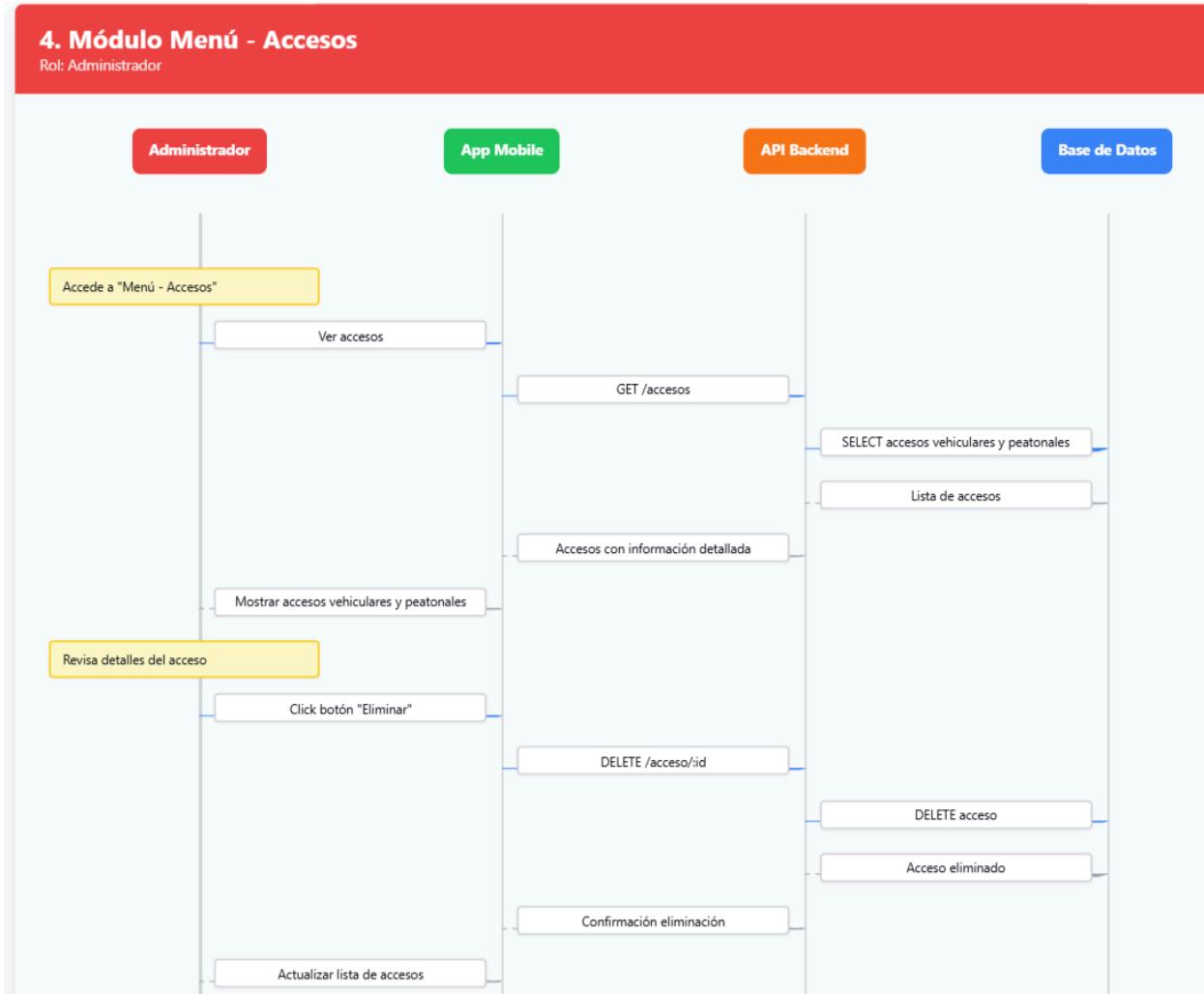


Diagrama 7, Diagrama de secuencia Menú Principal (Accesos) Administrador

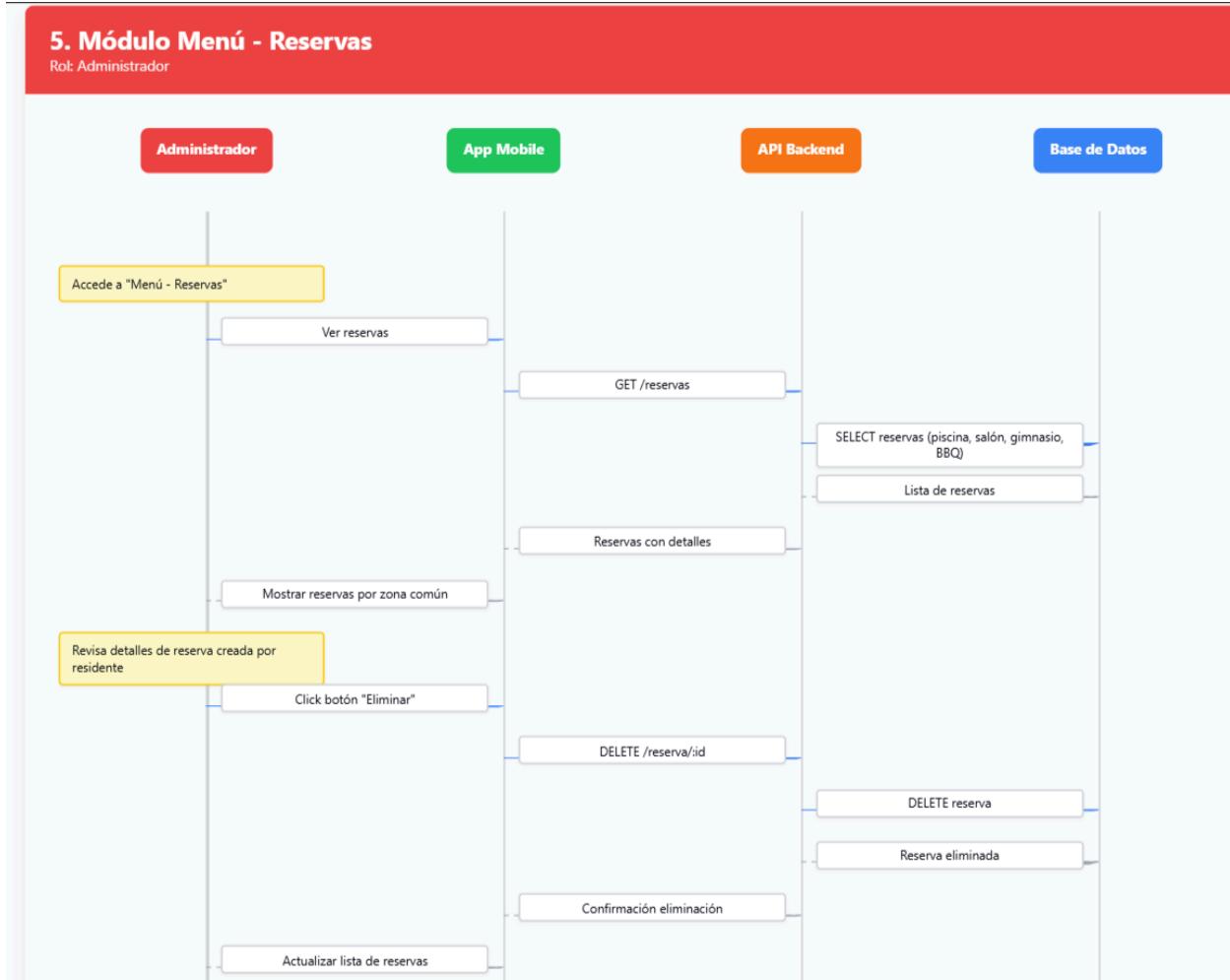


Diagrama 8, Diagrama de secuencia Menú Principal (Reservas) Administrador

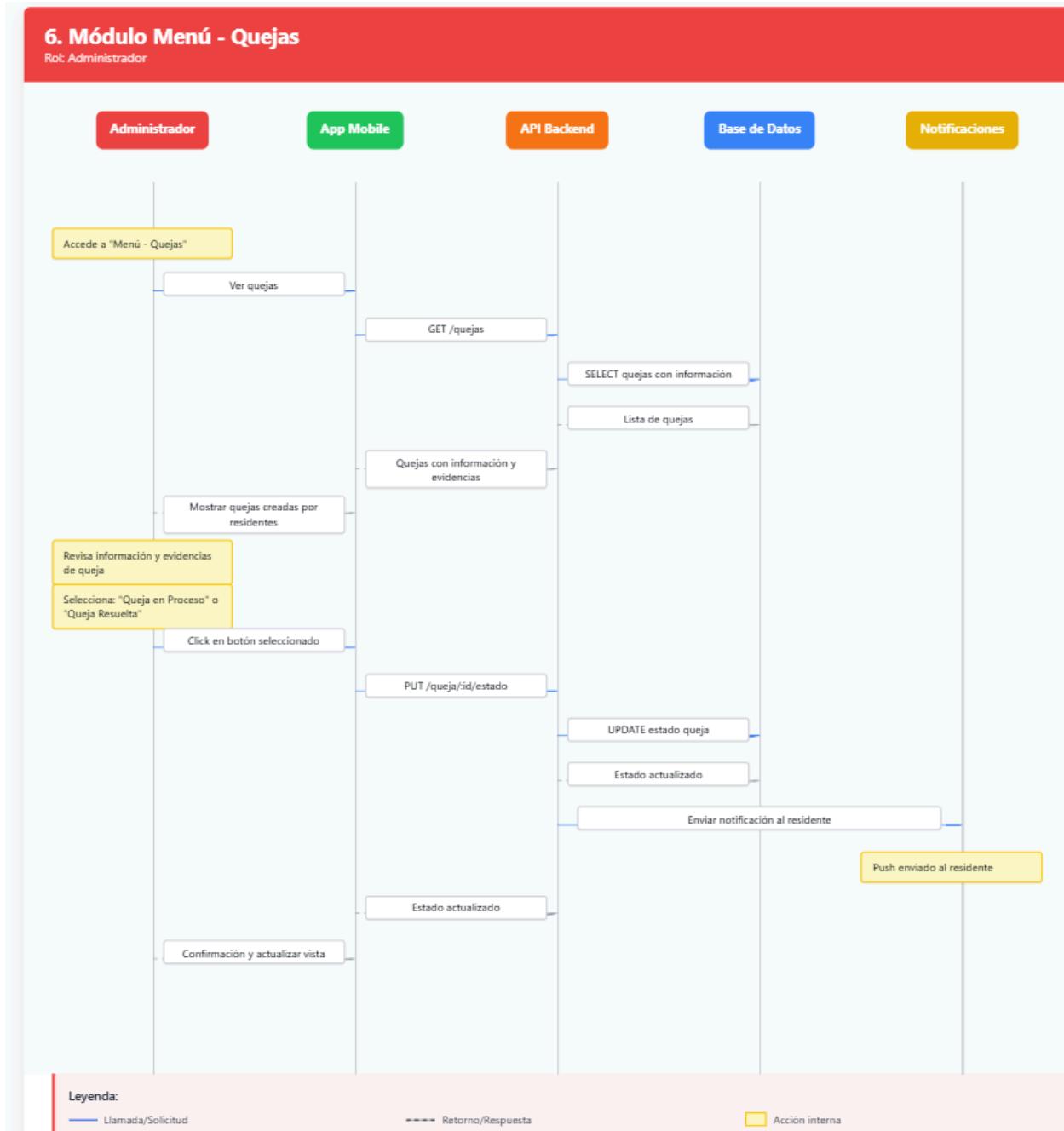


Diagrama 9, Diagrama de secuencia Menú Principal (Quejos) Administrador

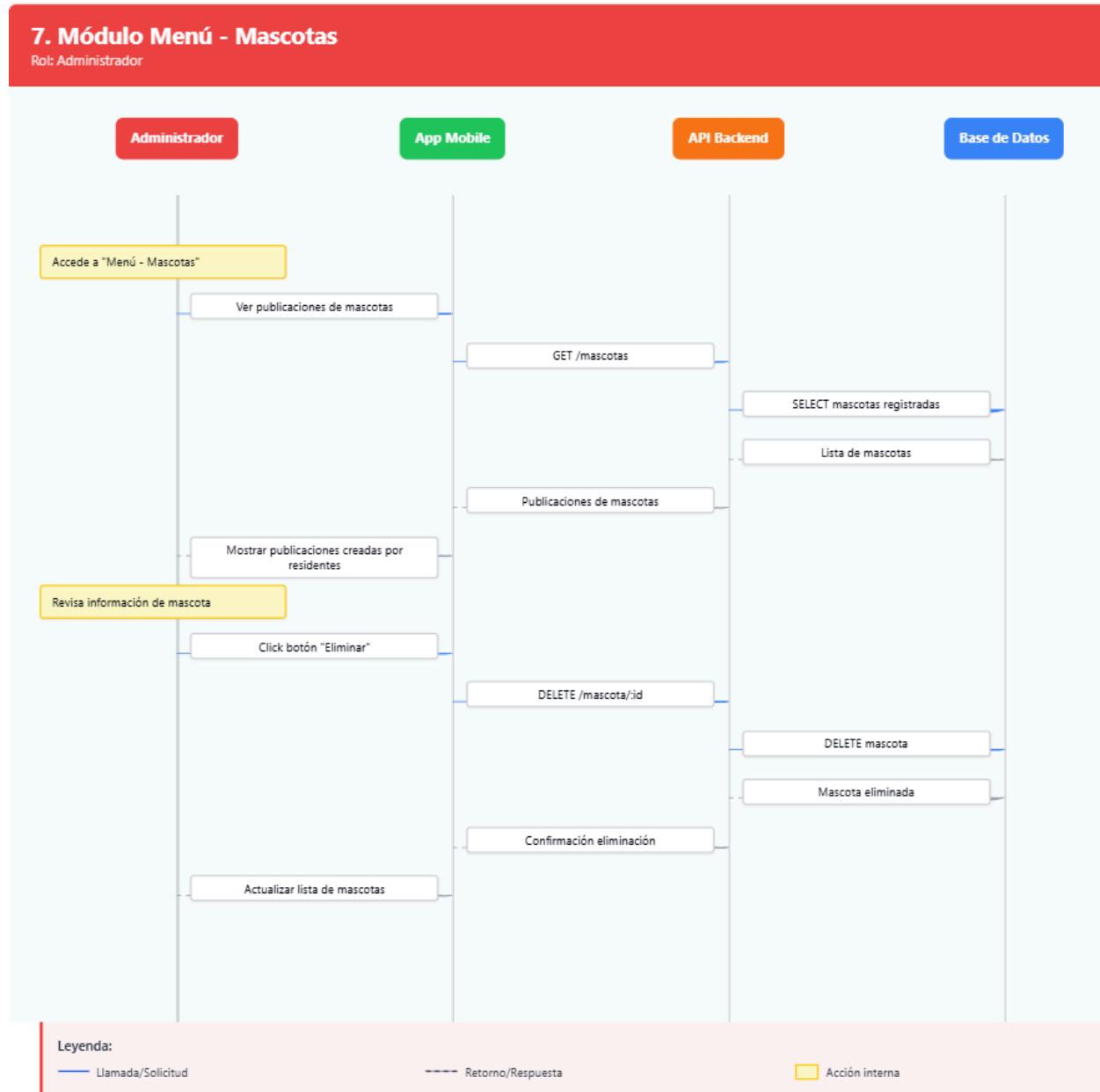


Diagrama 10, Diagrama de secuencia Menú Principal (Mascotas) Administrador

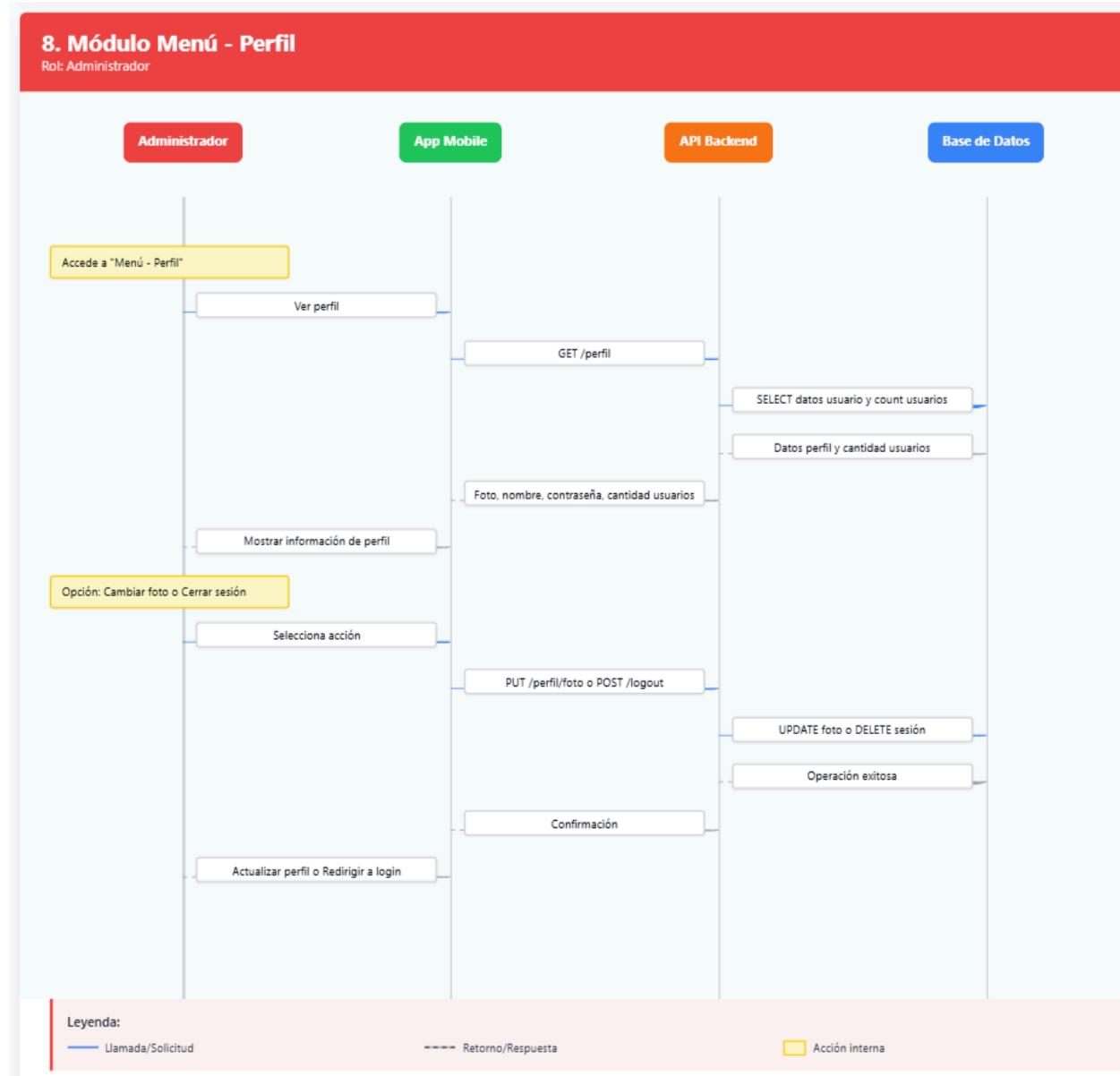
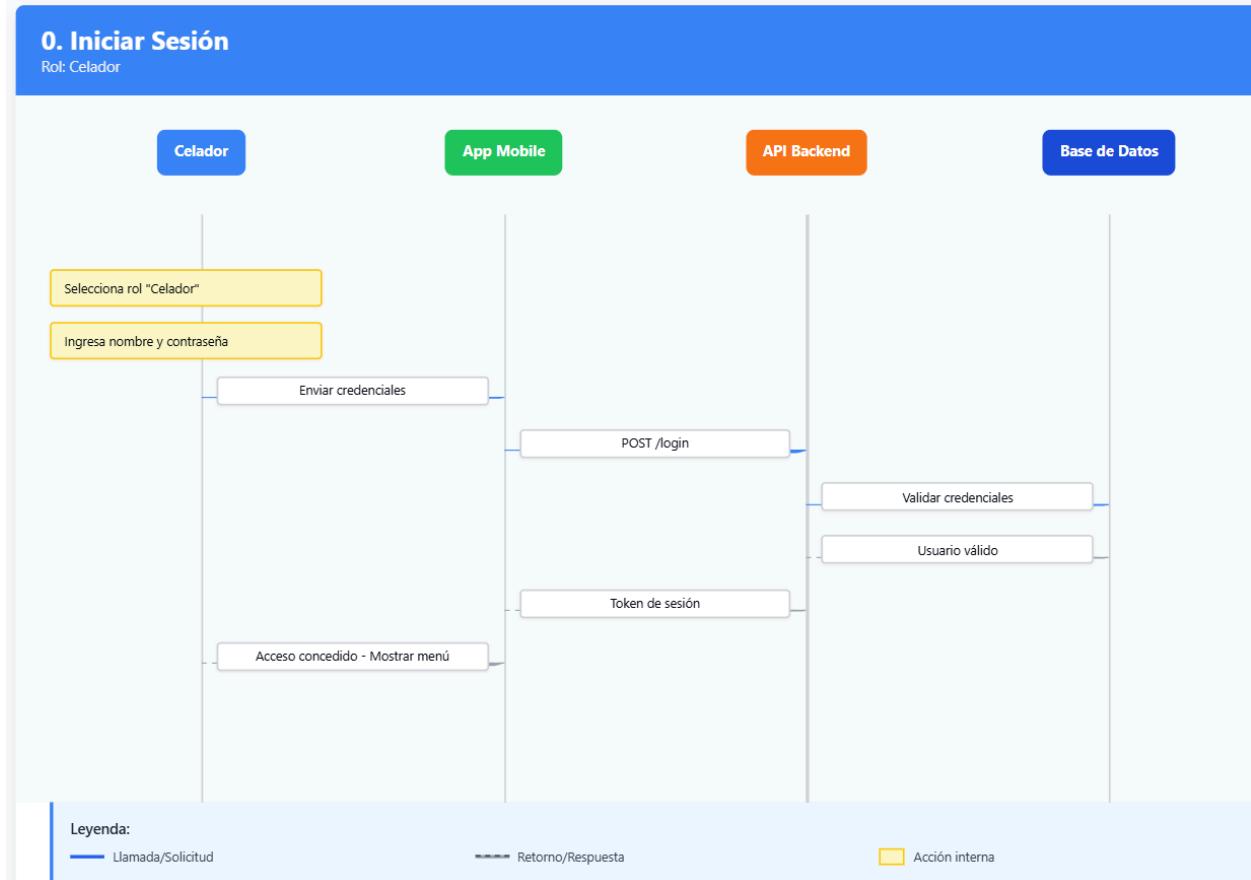


Diagrama 11, Diagrama de secuencia Menú Principal (Perfil) Administrador

Celador

*Diagrama 12, Diagrama de secuencia Inicio Sesión Celador*

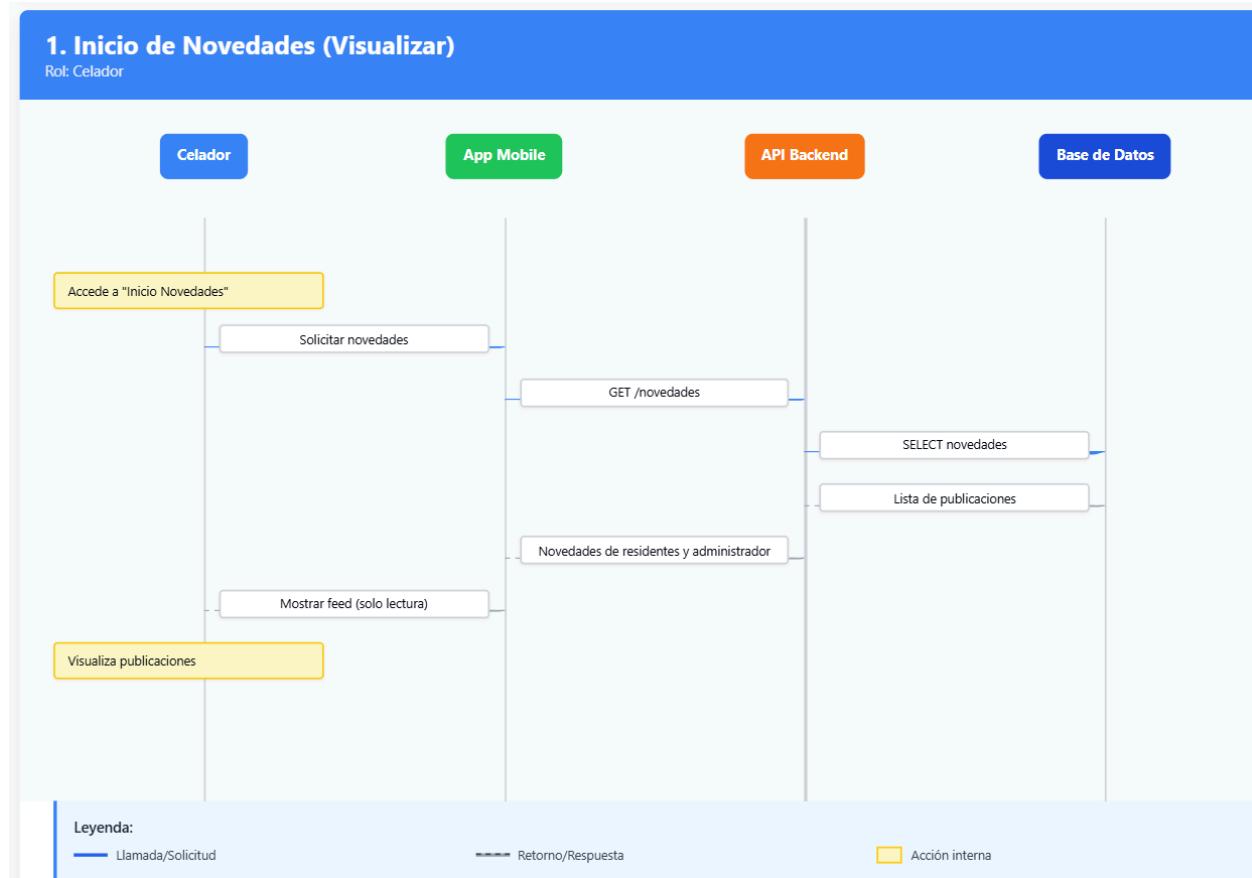


Diagrama 13, Diagrama de secuencia Inicio Novedades Celador

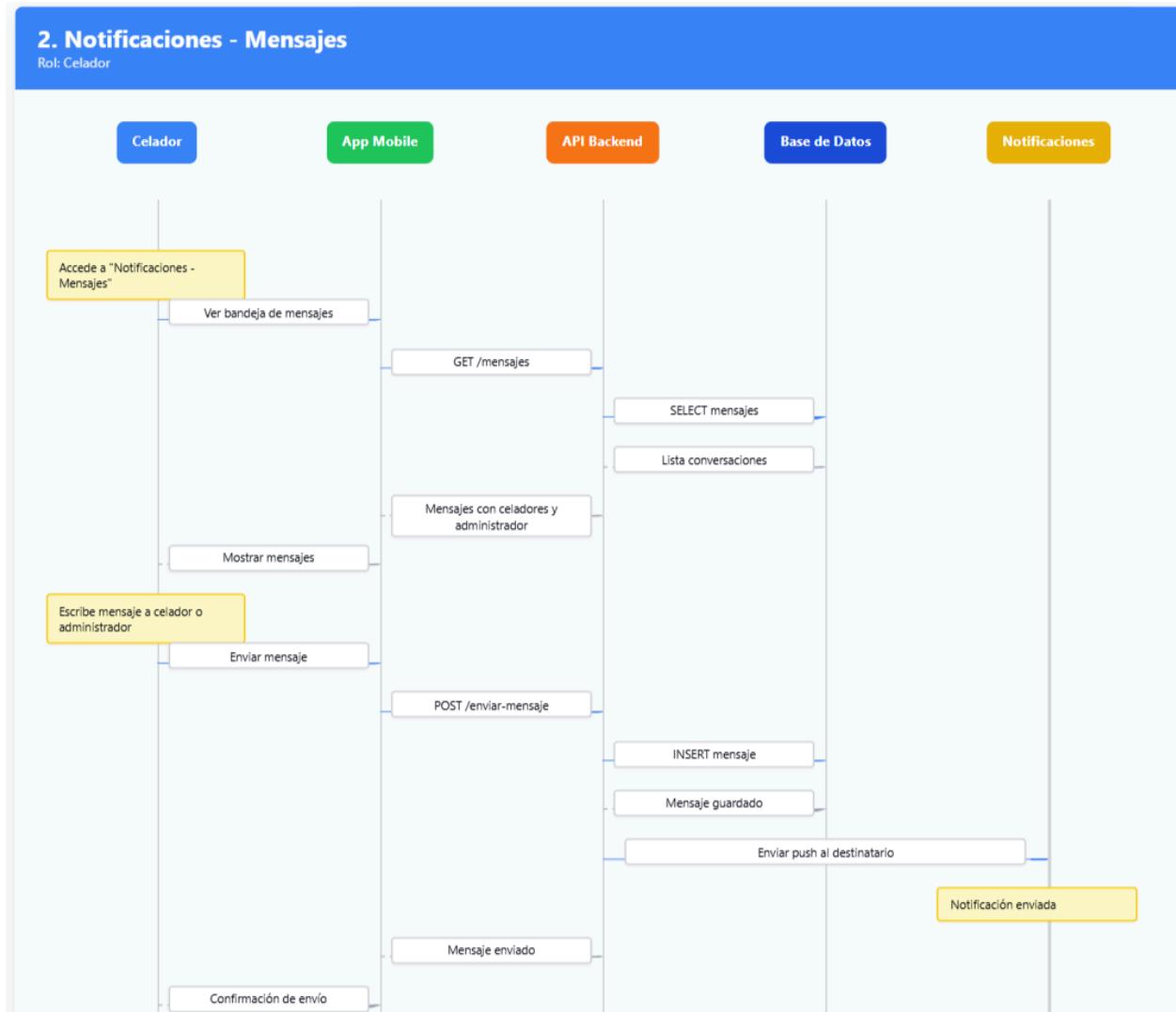


Diagrama 14, Diagrama de secuencia Notificaciones (Mensajes) Celador



Diagrama 15, Diagrama de secuencia Notificaciones (Paquetería) Celador

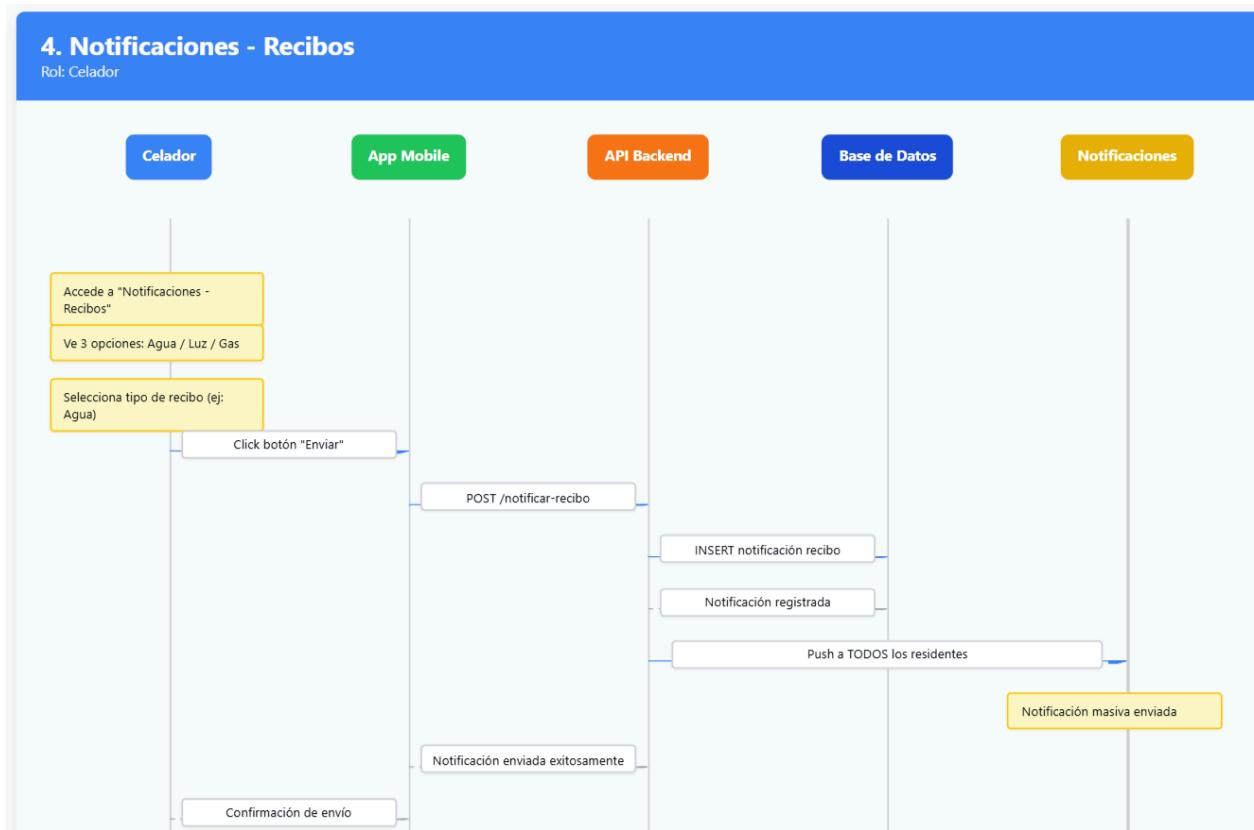


Diagrama 16, Diagrama de secuencia Notificaciones (Recibos) Celador

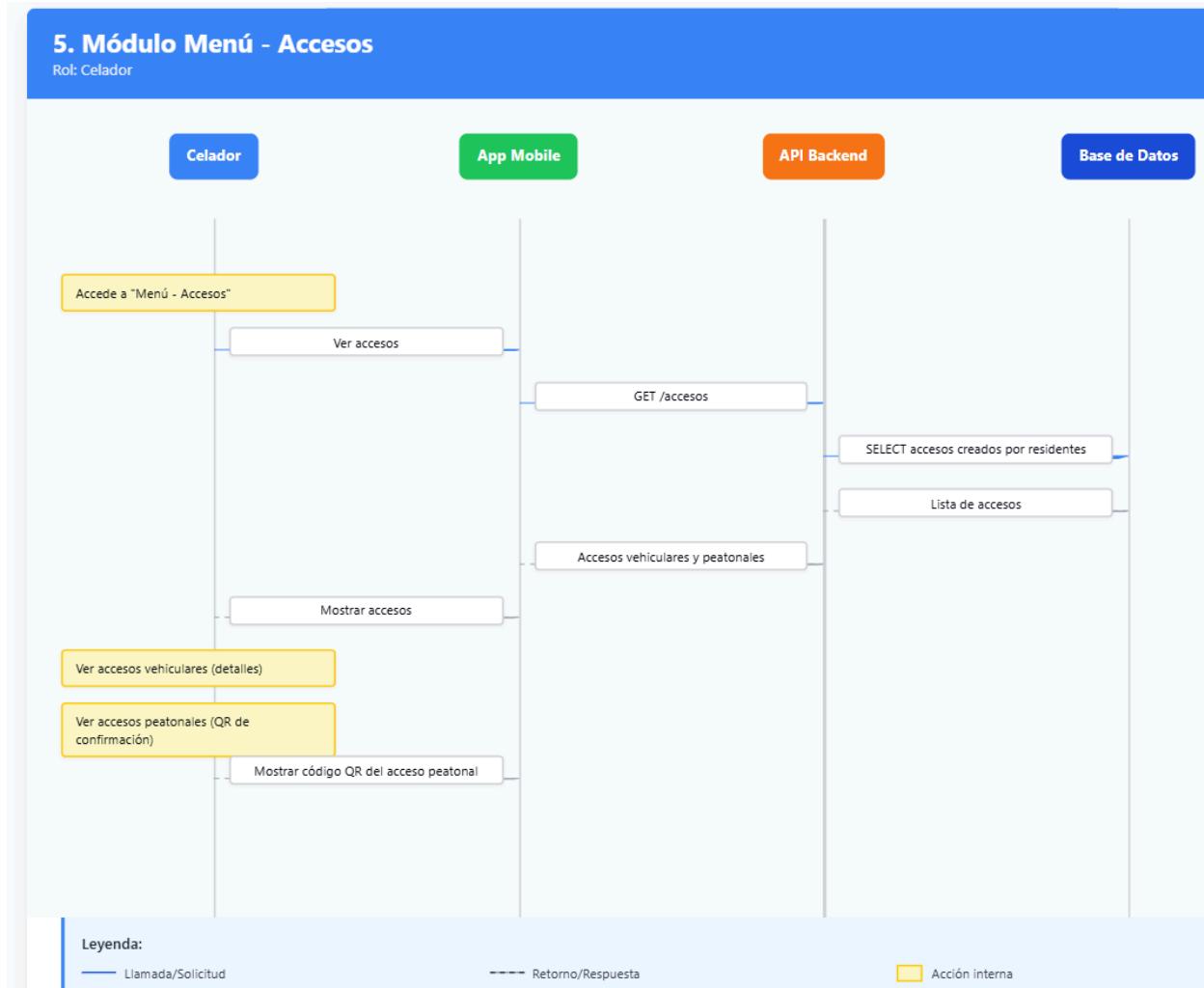


Diagrama 17, Diagrama de secuencia Menú Principal (Accesos) Celador

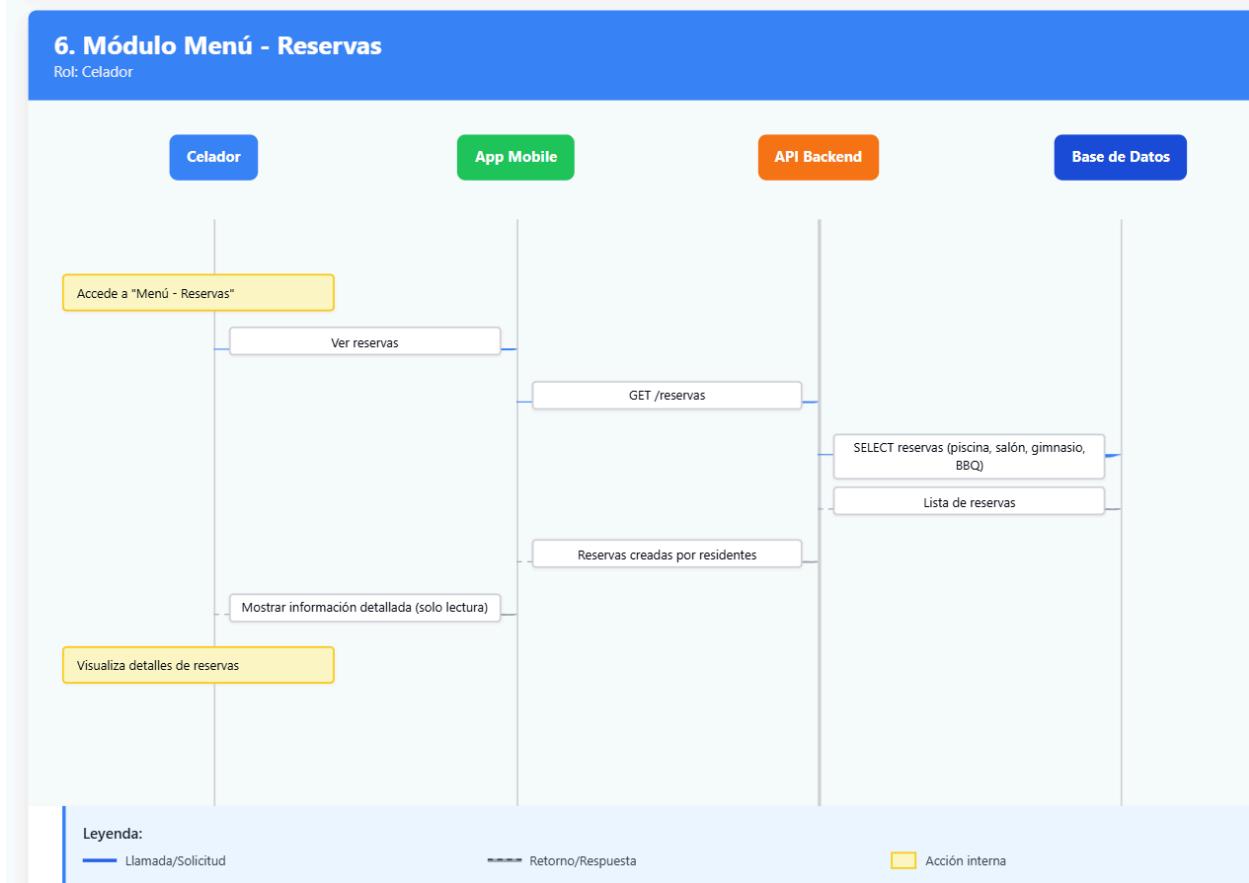


Diagrama 18, Diagrama de secuencia Menú Principal (Reservas) Celador

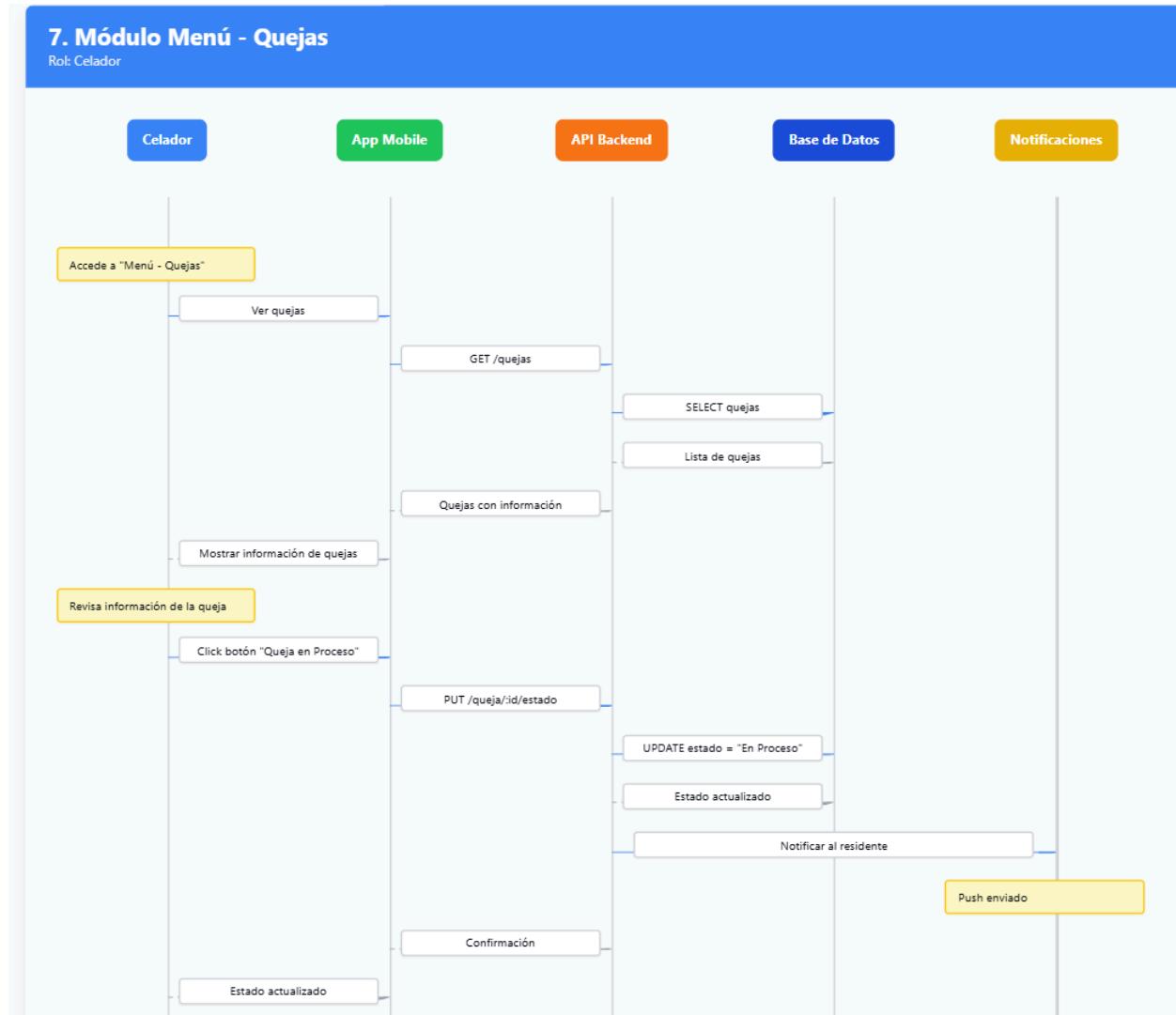


Diagrama 19, Diagrama de secuencia Menú Principal (Quejas) Celador

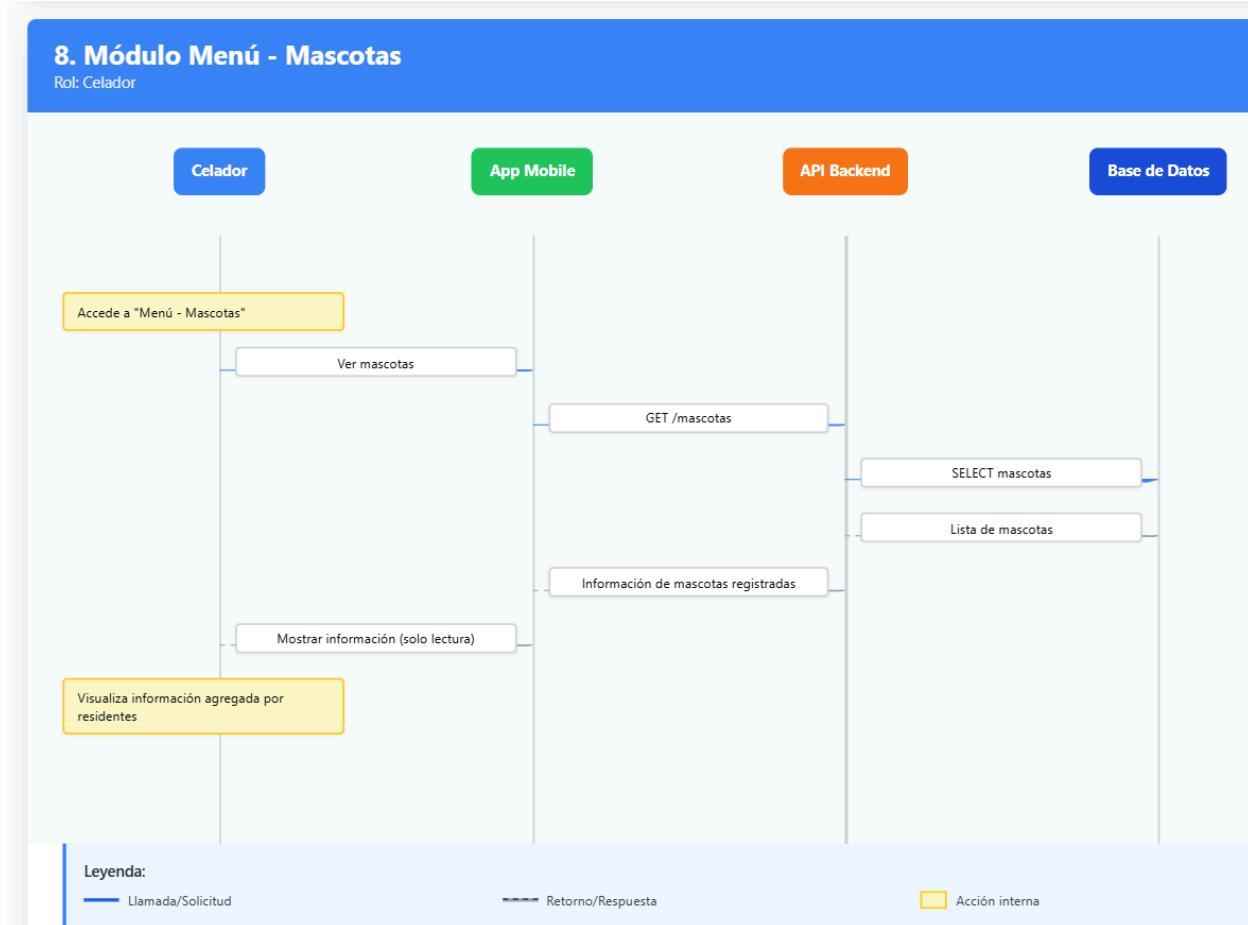


Diagrama 20, Diagrama de secuencia Menú Principal (Mascotas) Celador

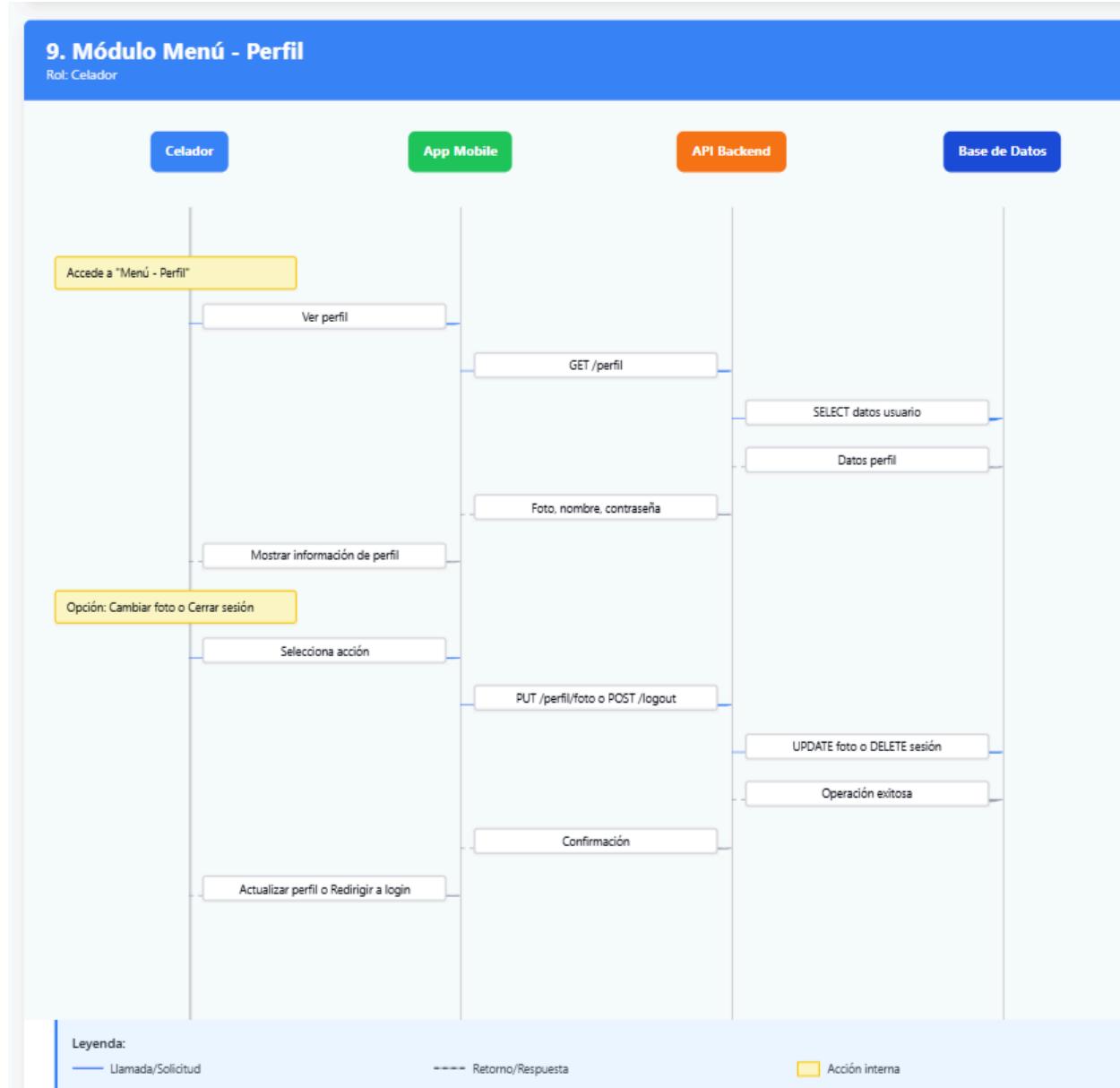


Diagrama 21, Diagrama de secuencia Menú Principal (Perfil) Celador

Residente

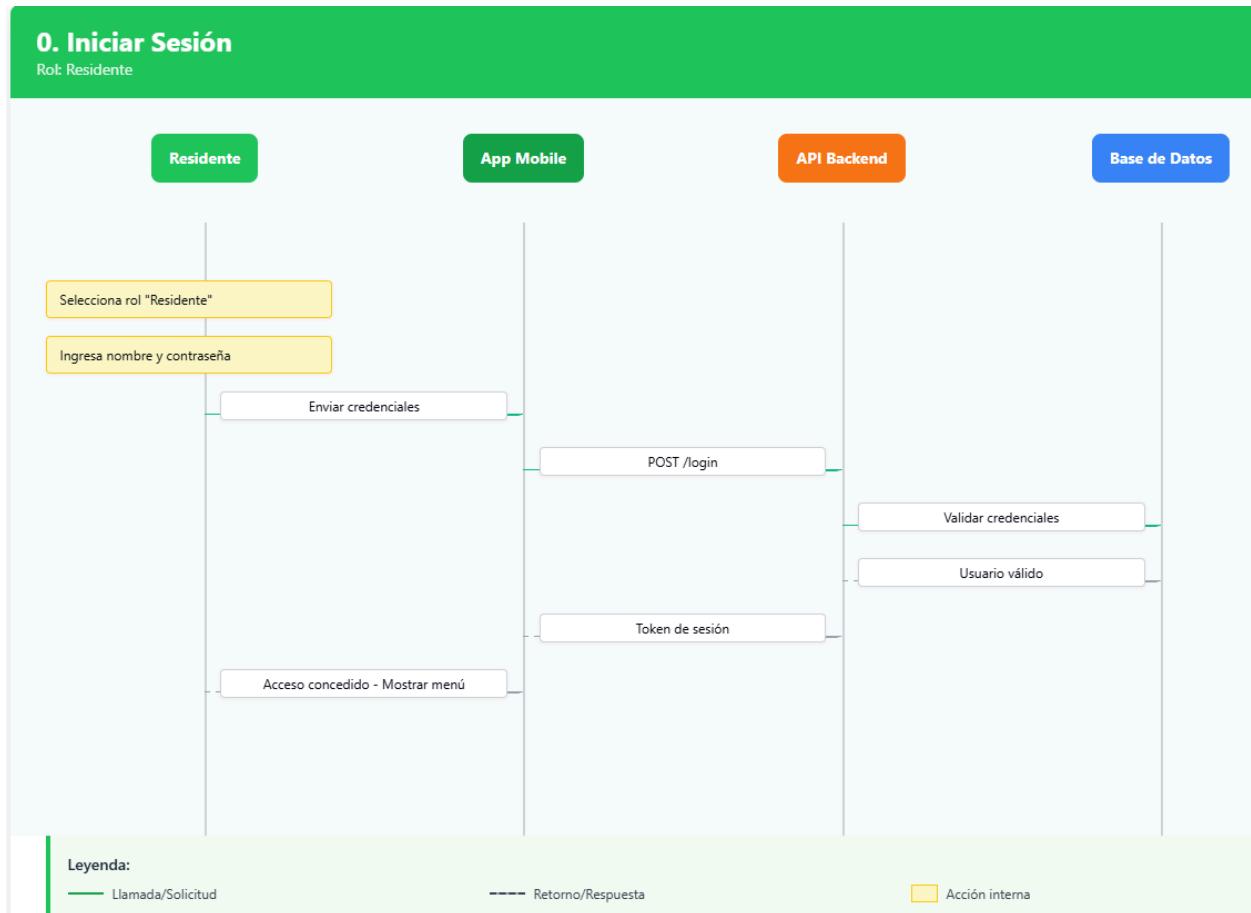


Diagrama 22, Diagrama de secuencia Inicio Sesión Residente

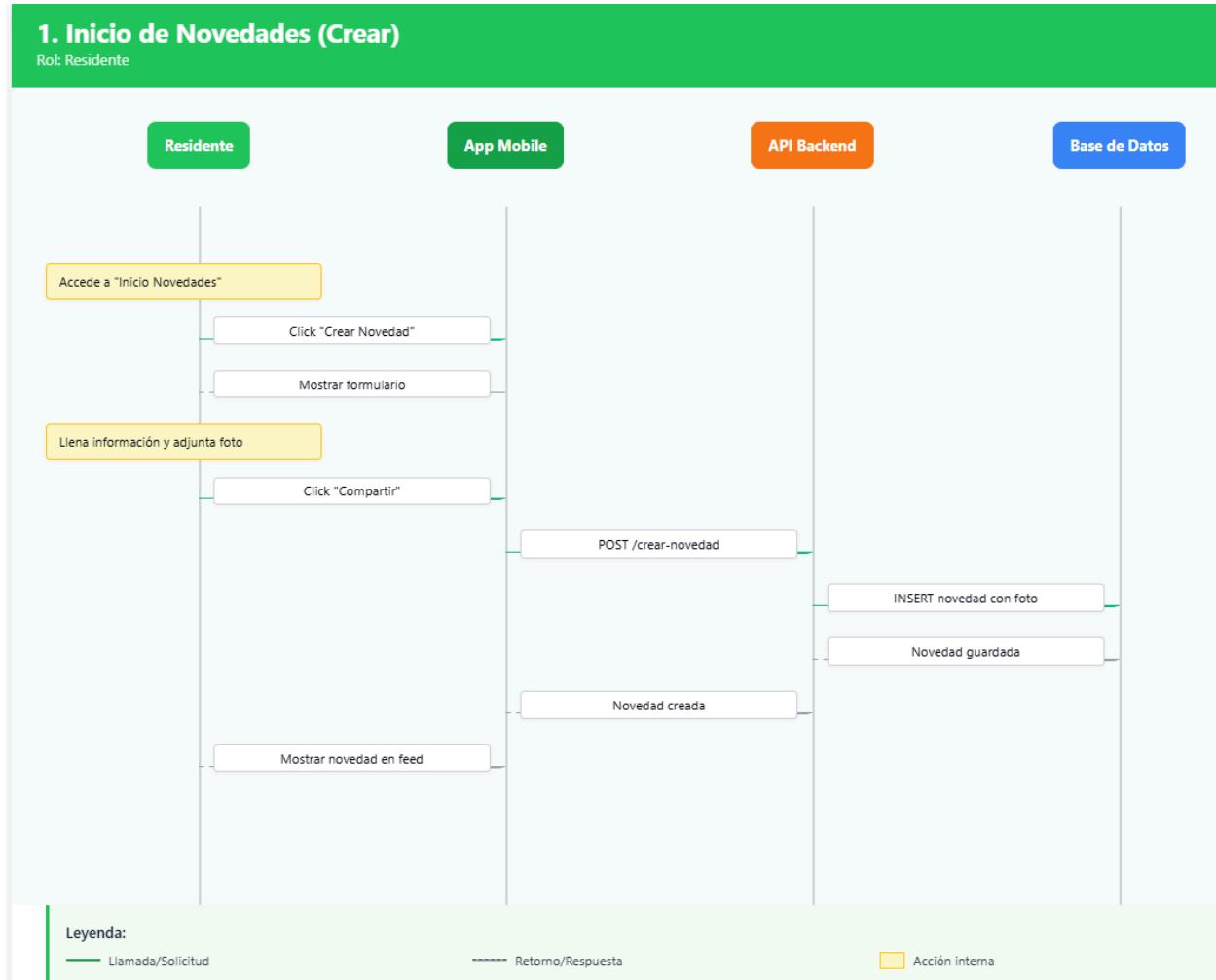


Diagrama 23, Diagrama de secuencia Inicio Novedades Residente

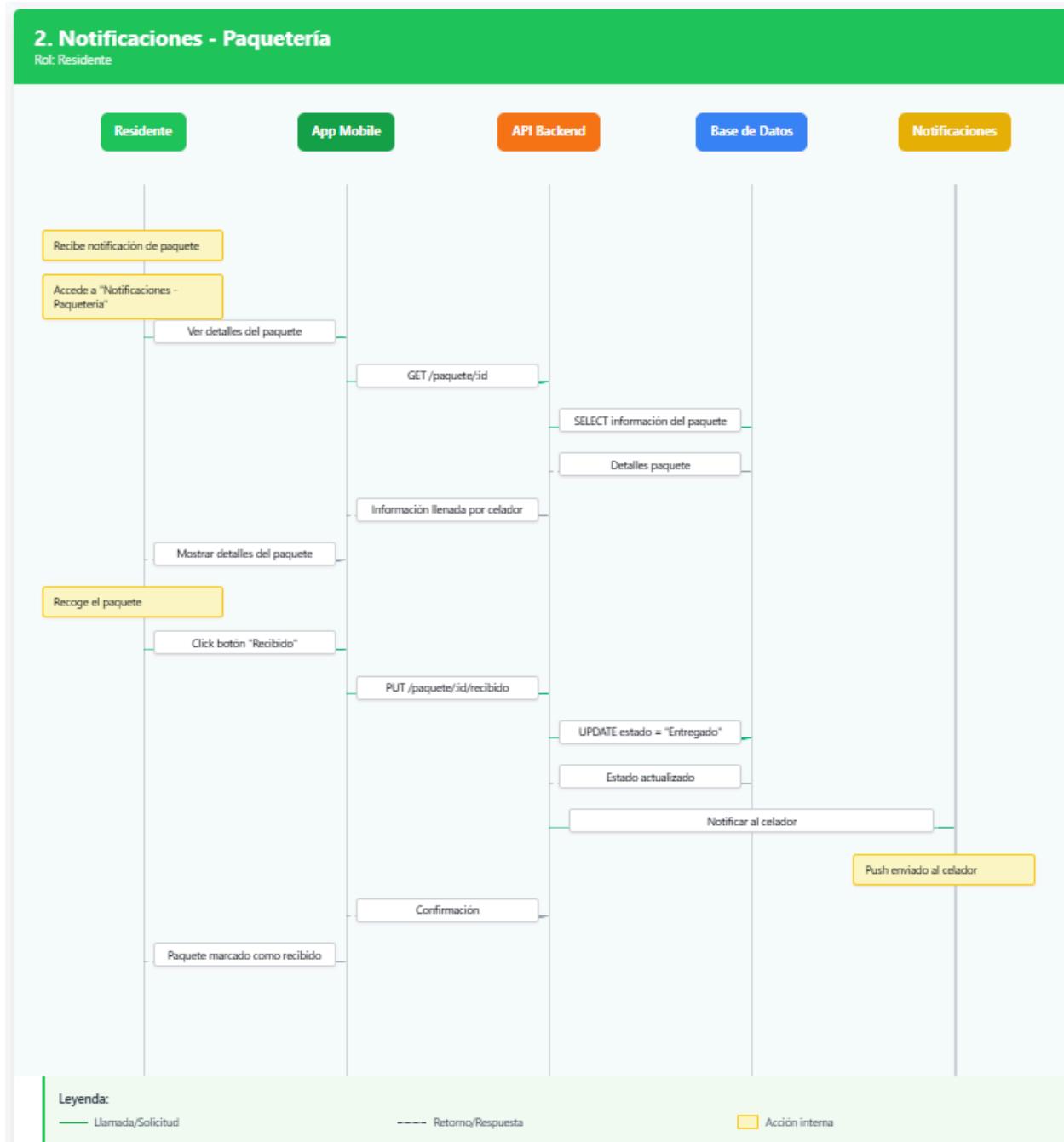


Diagrama 24, Diagrama de secuencia Notificaciones (Paquetería) Residente

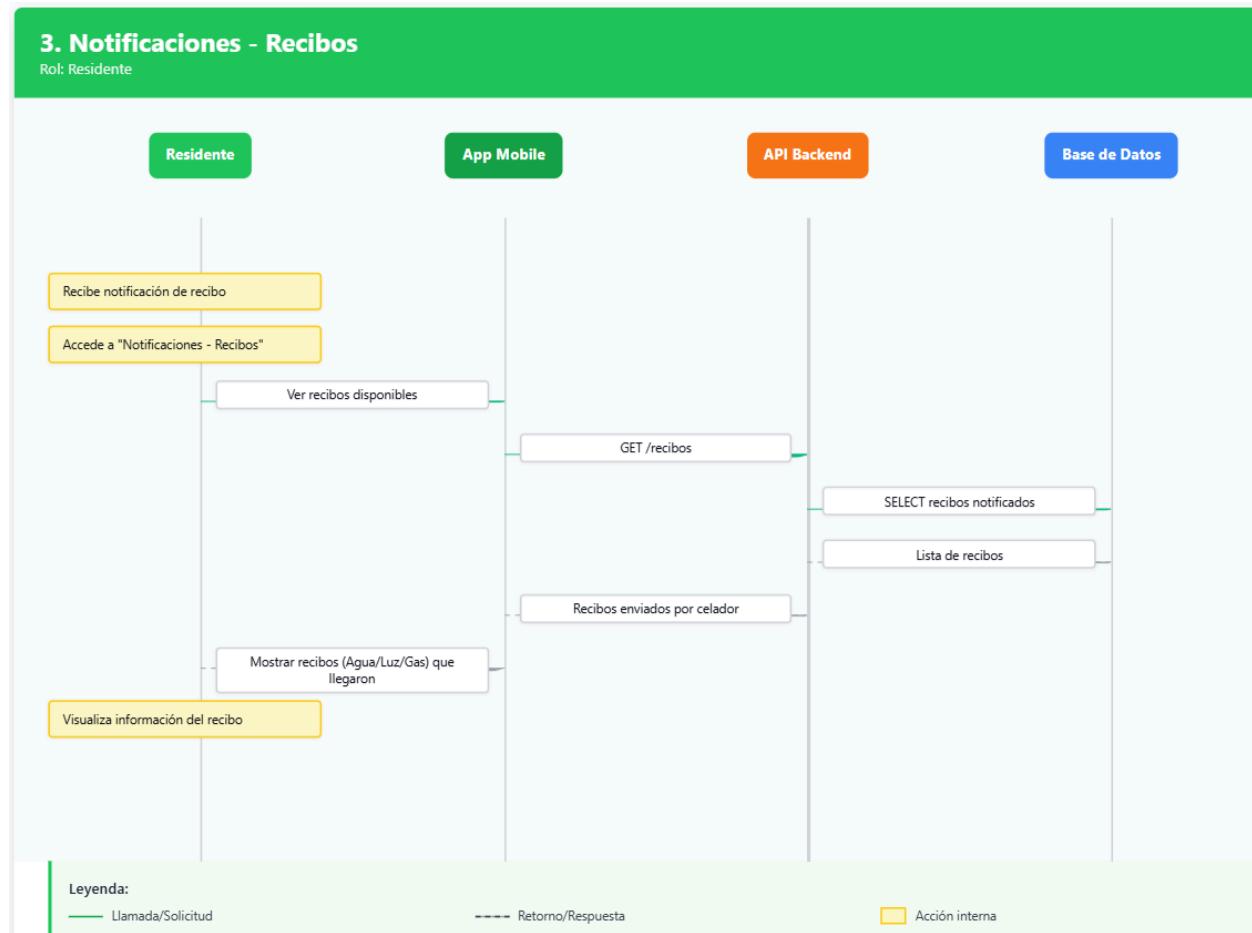


Diagrama 25, Diagrama de secuencia Notificaciones (Recibos) Residente

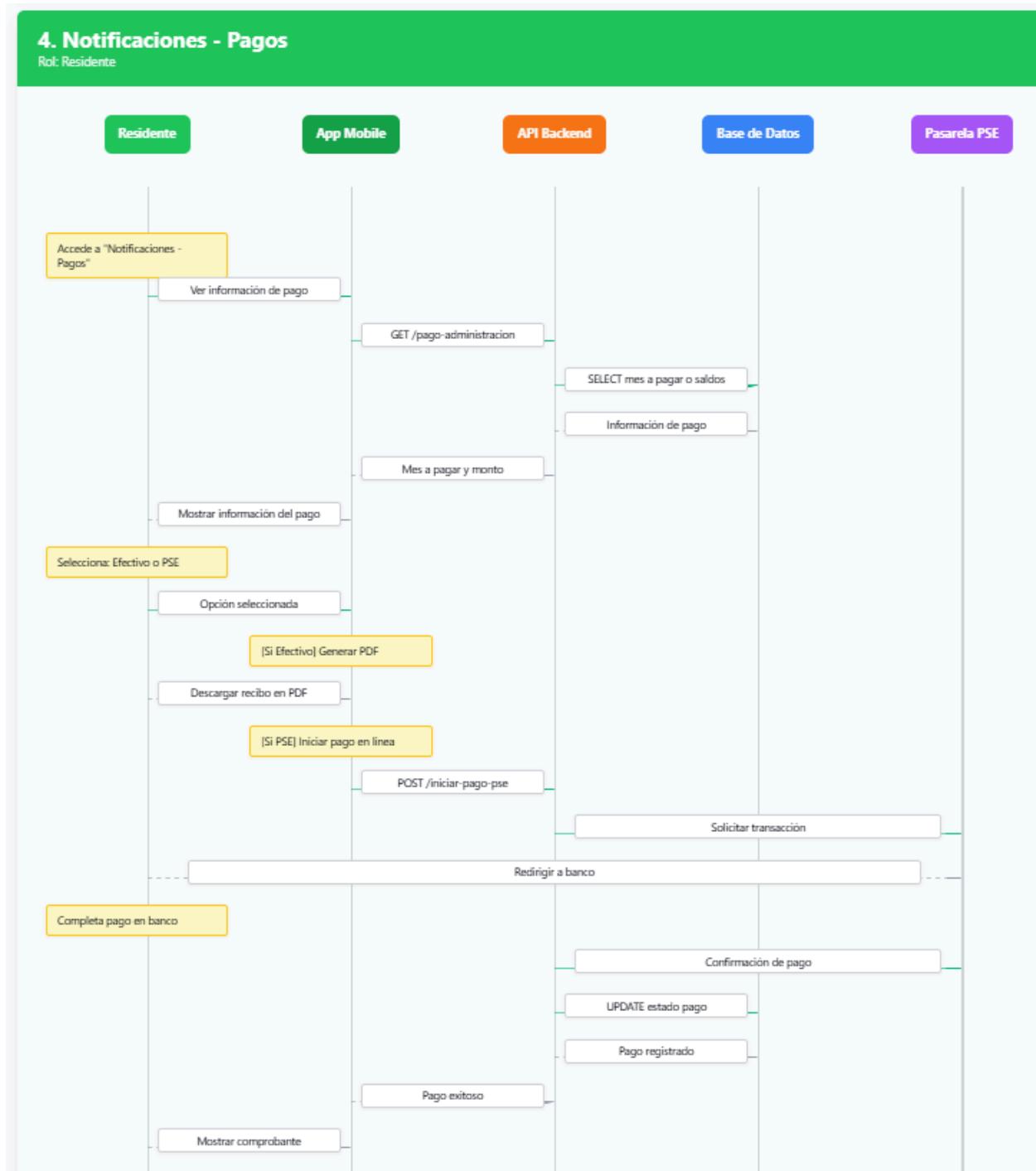


Diagrama 26, Diagrama de secuencia Notificaciones (Pagos) Residente

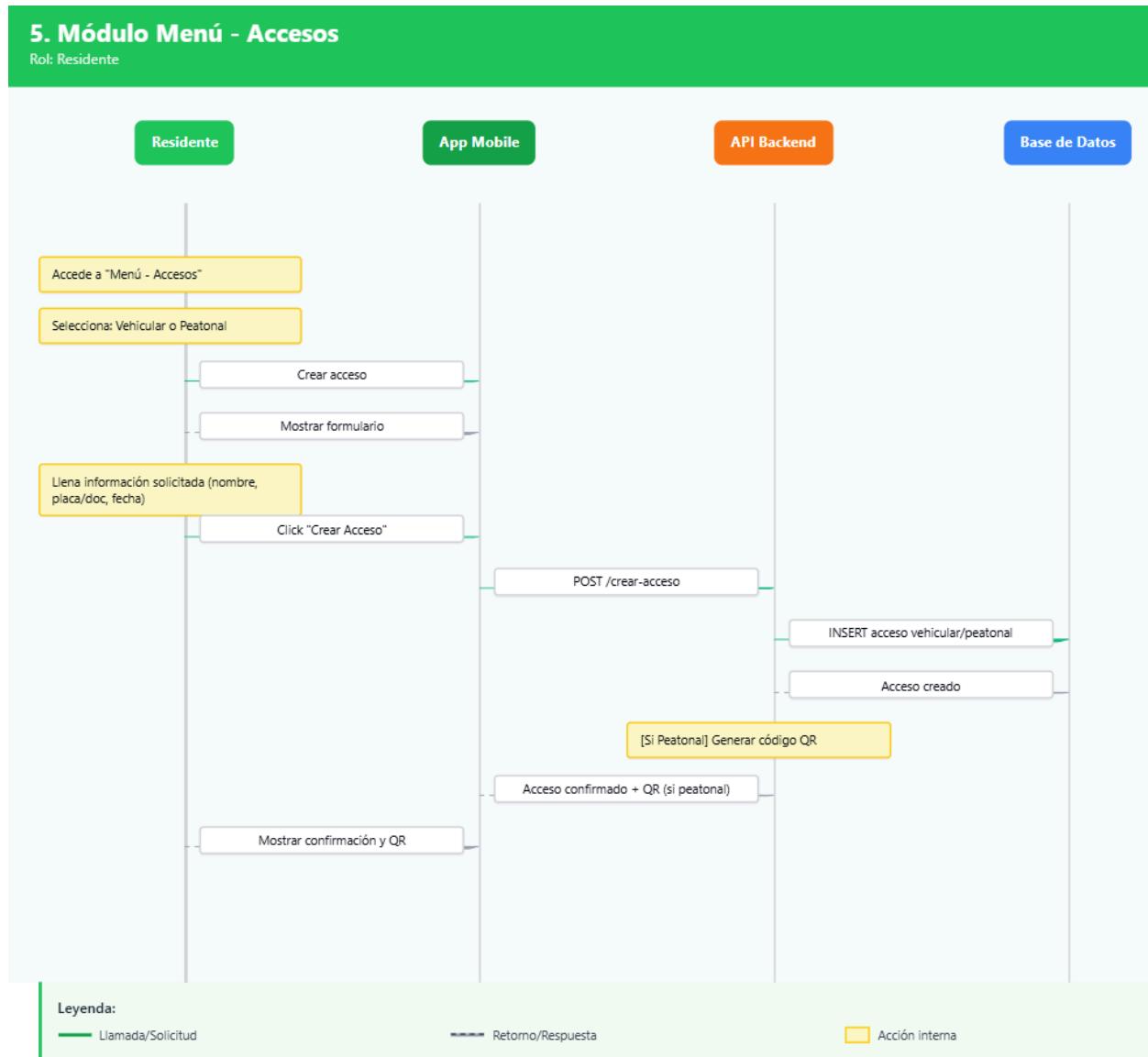


Diagrama 27, Diagrama de secuencia Menú Principal (Accesos) Residente

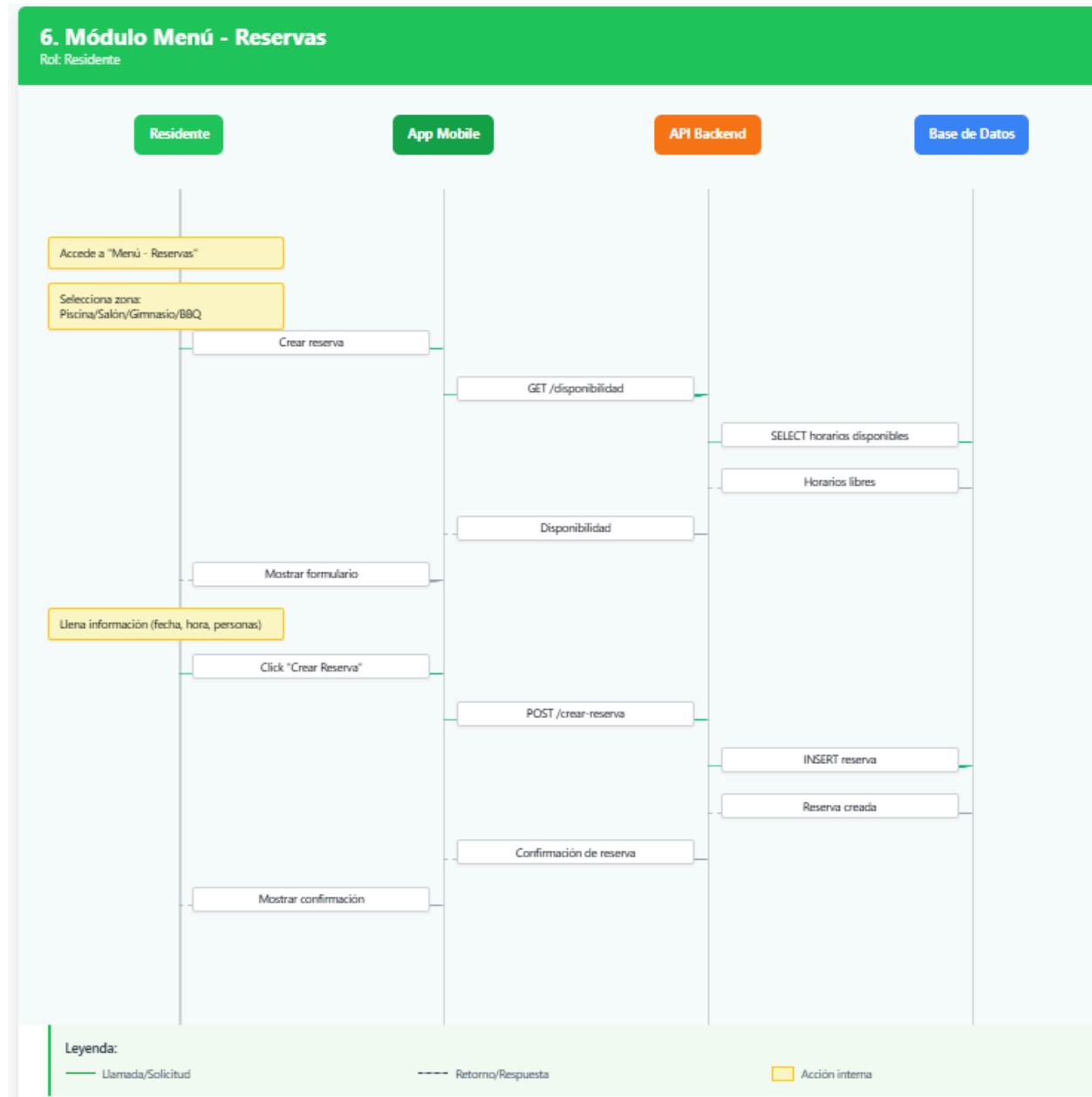


Diagrama 28, Diagrama de secuencia Menú Principal (Reservas) Residente

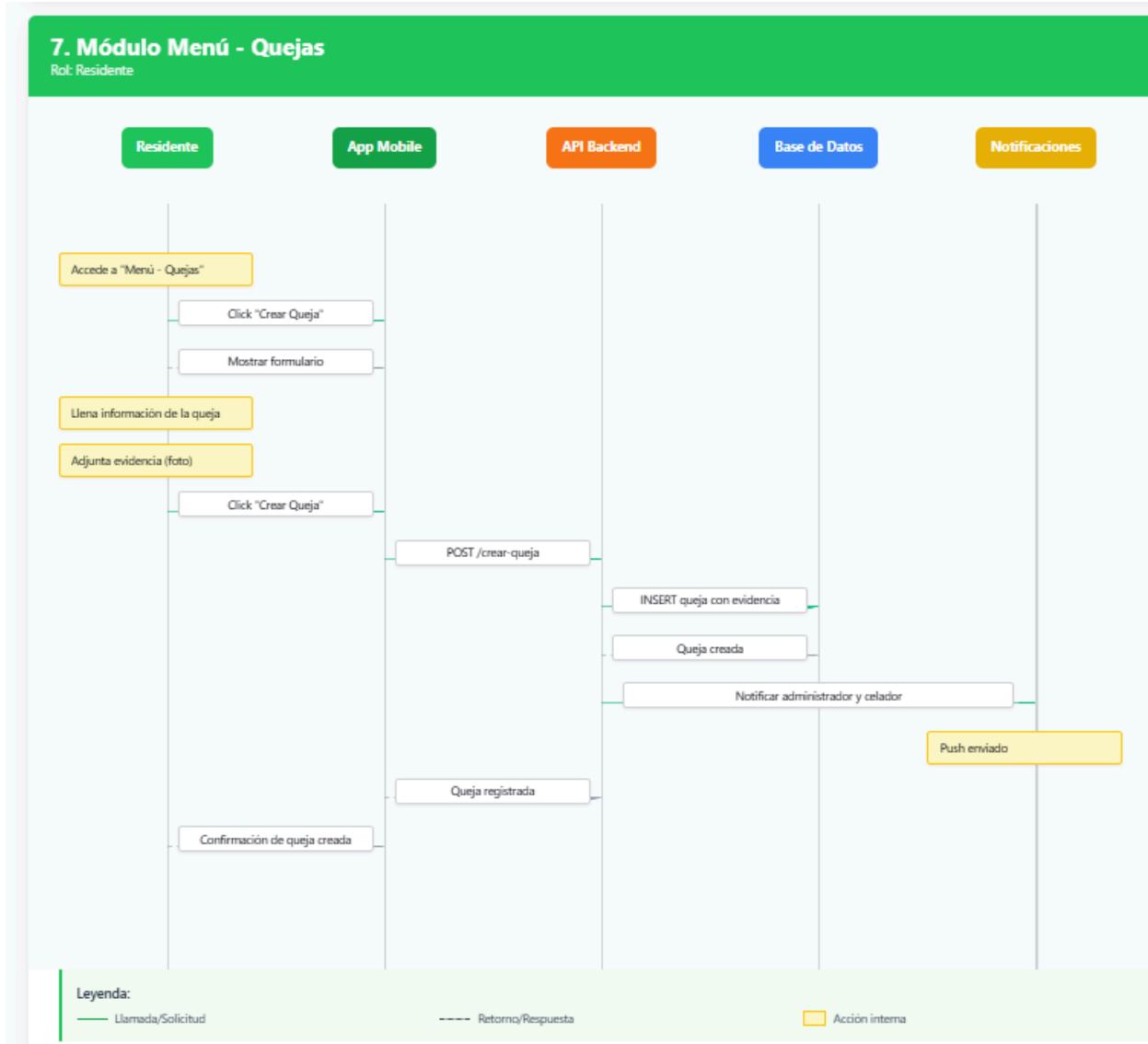


Diagrama 29, Diagrama de secuencia Menú Principal (Quejas) Residente

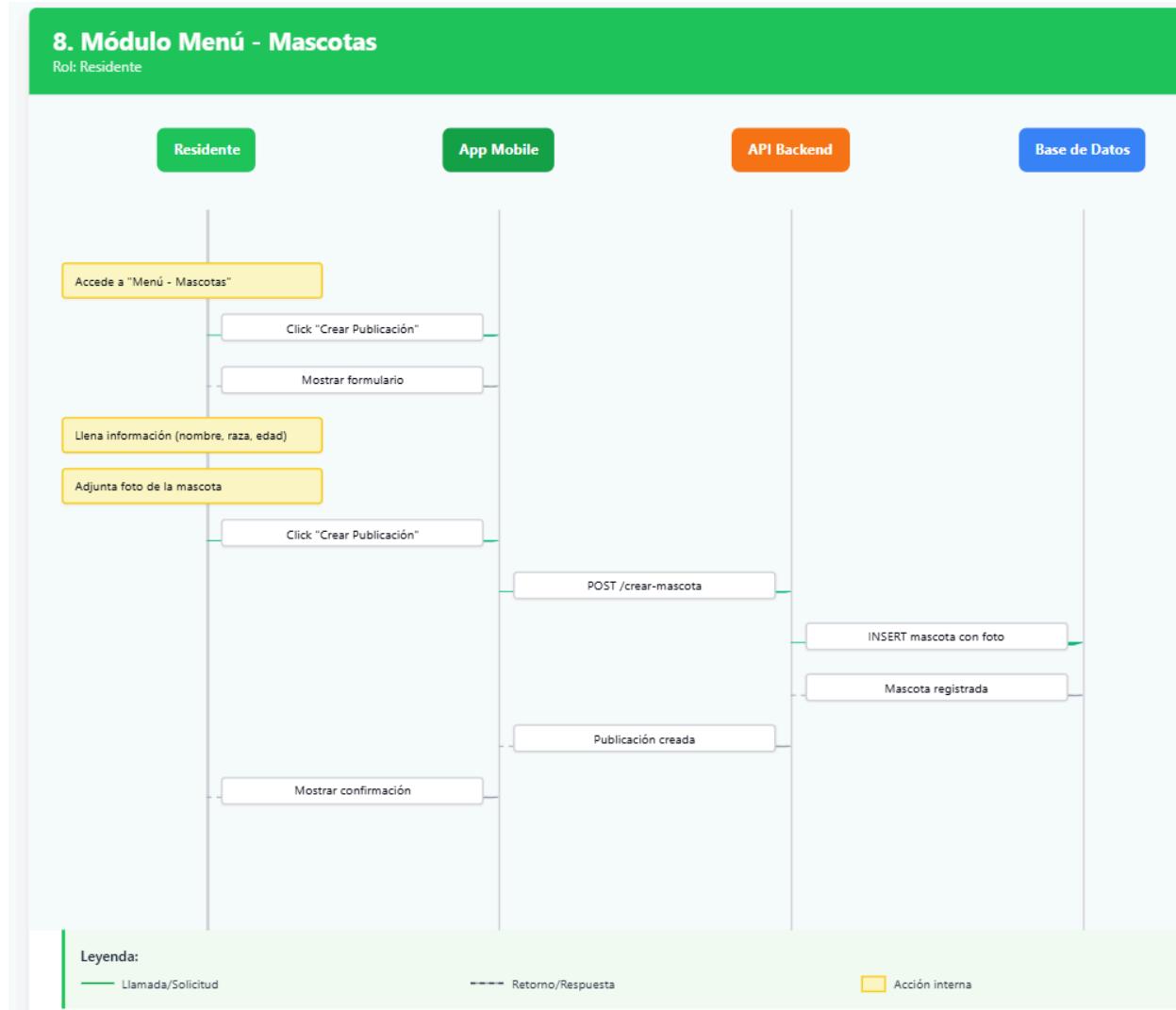


Diagrama 30, Diagrama de secuencia Menú Principal (Mascotas) Residente



Diagrama 31, Diagrama de secuencia Menú Principal (Perfil) Residente

13.4 Diagramas de Estado

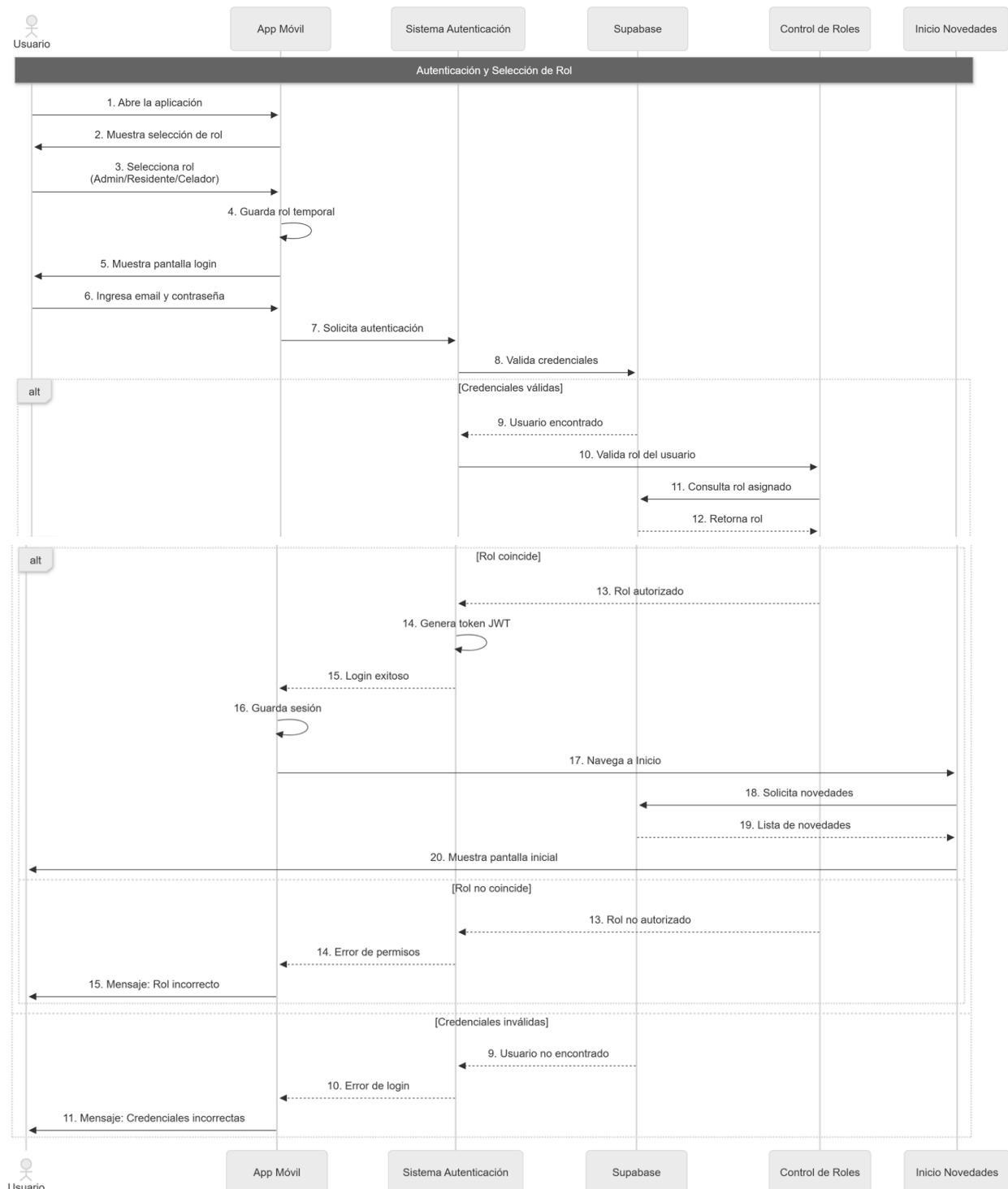


Diagrama 32, Diagrama de Estado Autenticación y Selección Rol

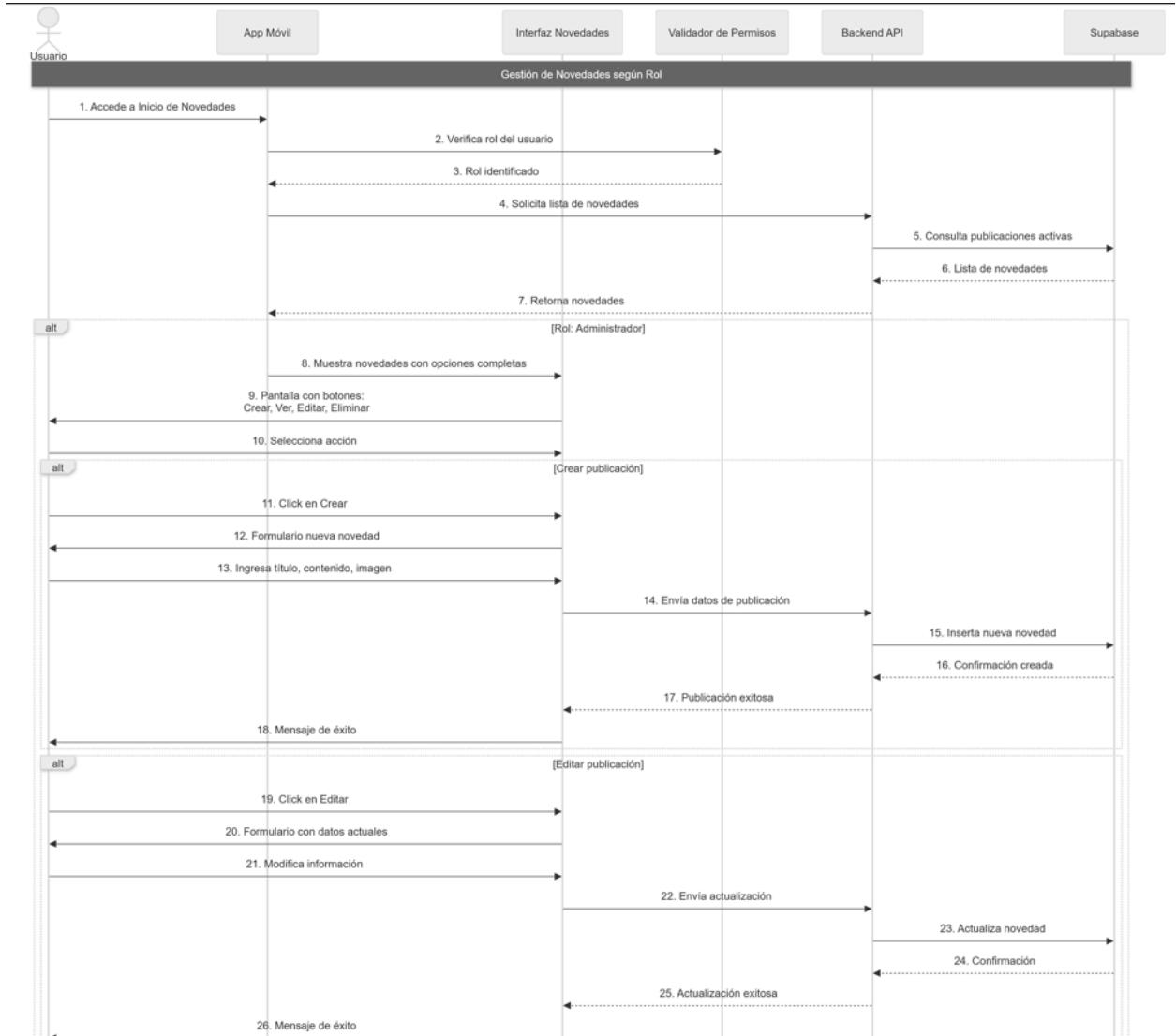


Diagrama 33, Diagrama de Estado Inicio Novedades Crear y Editar Publicación Administrador

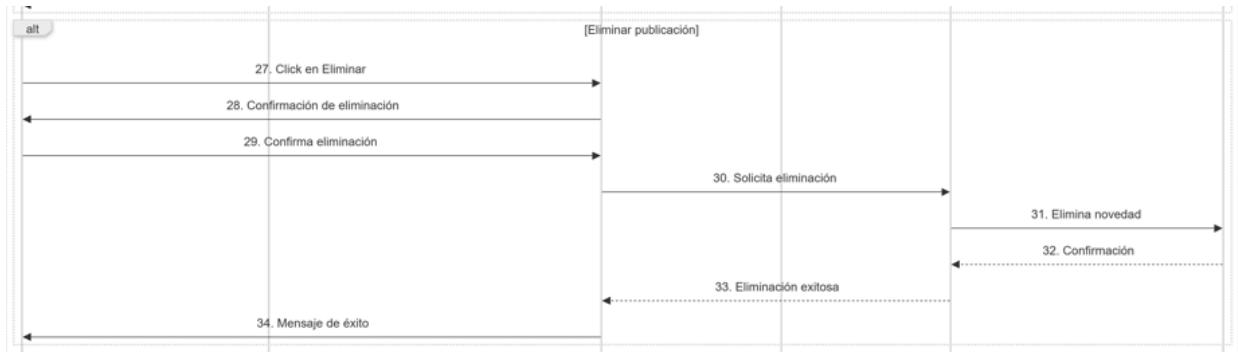


Diagrama 34, Diagrama de Estado Inicio Novedades Eliminar Publicación Administrador

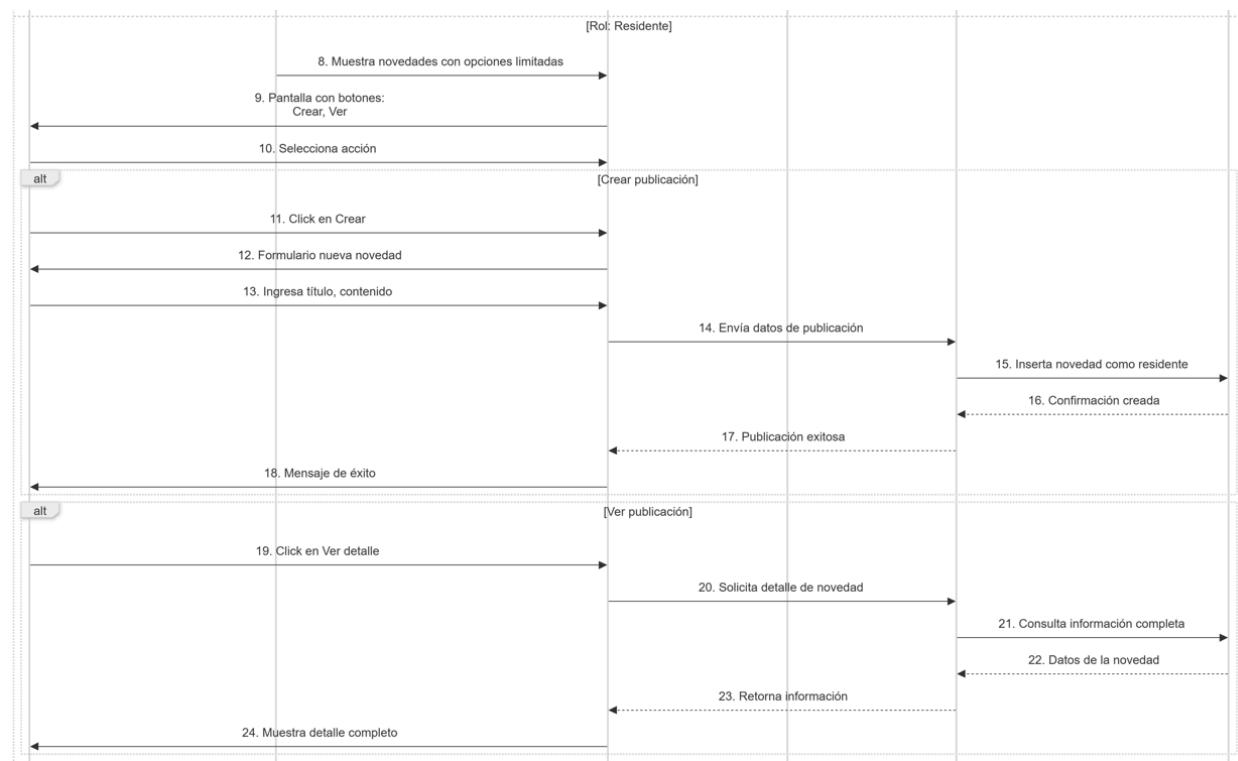


Diagrama 35, Diagrama de Estado Inicio Novedades Crear y Ver Publicación Residente

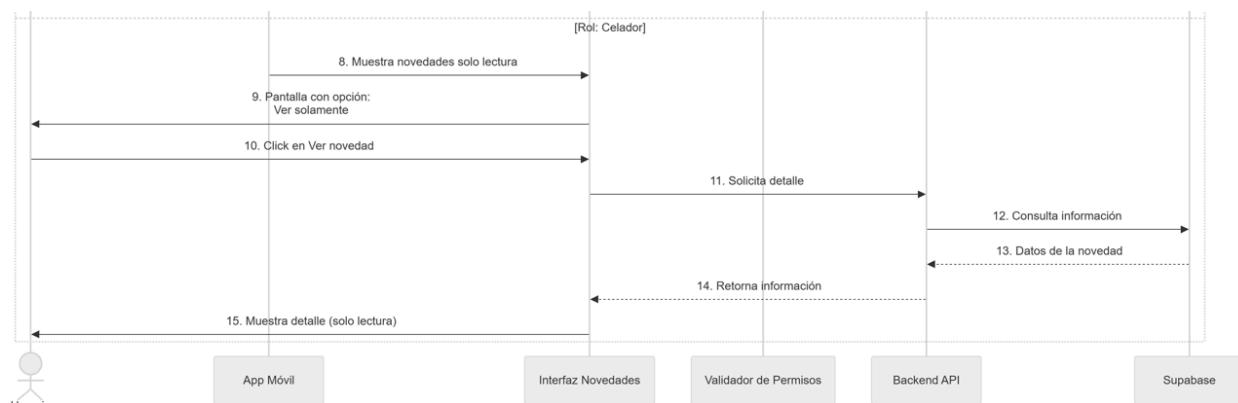


Diagrama 36, Diagrama de Estado Inicio Novedades Ver Publicación Celador

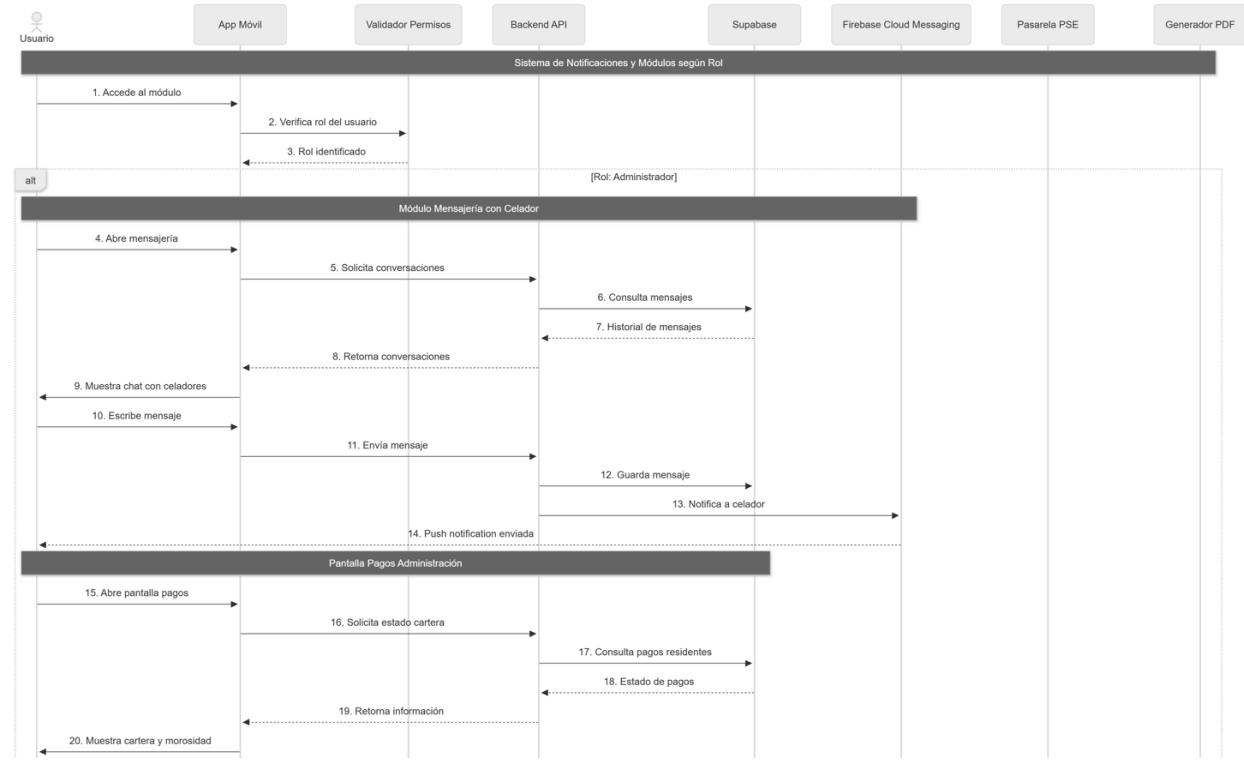


Diagrama 37, Diagrama de Estado Notificaciones (Mensajes y Pagos) Administrador

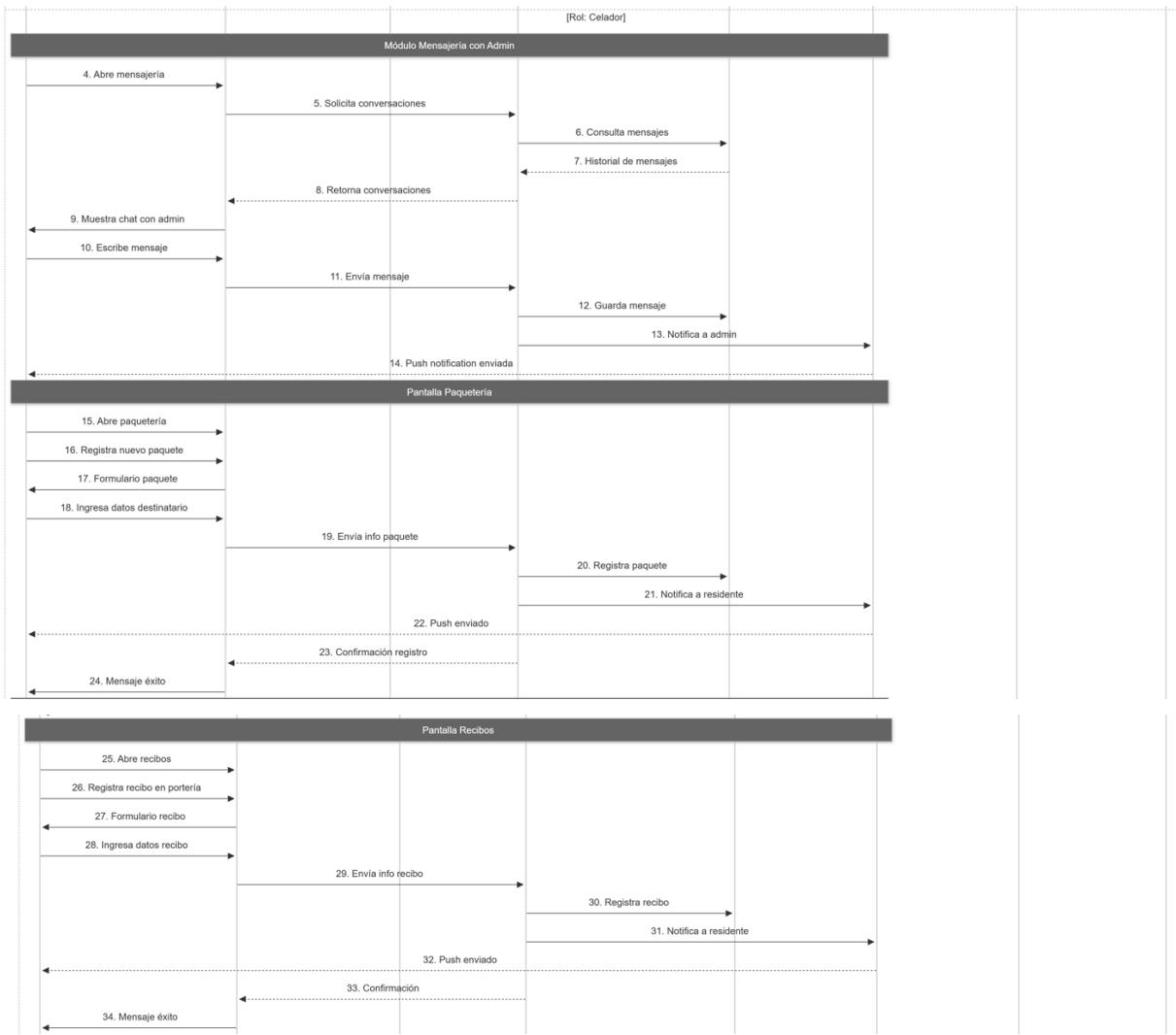


Diagrama 38, Diagrama de Estado Notificaciones (Mensajes, Paquetería y Recibos) Celador

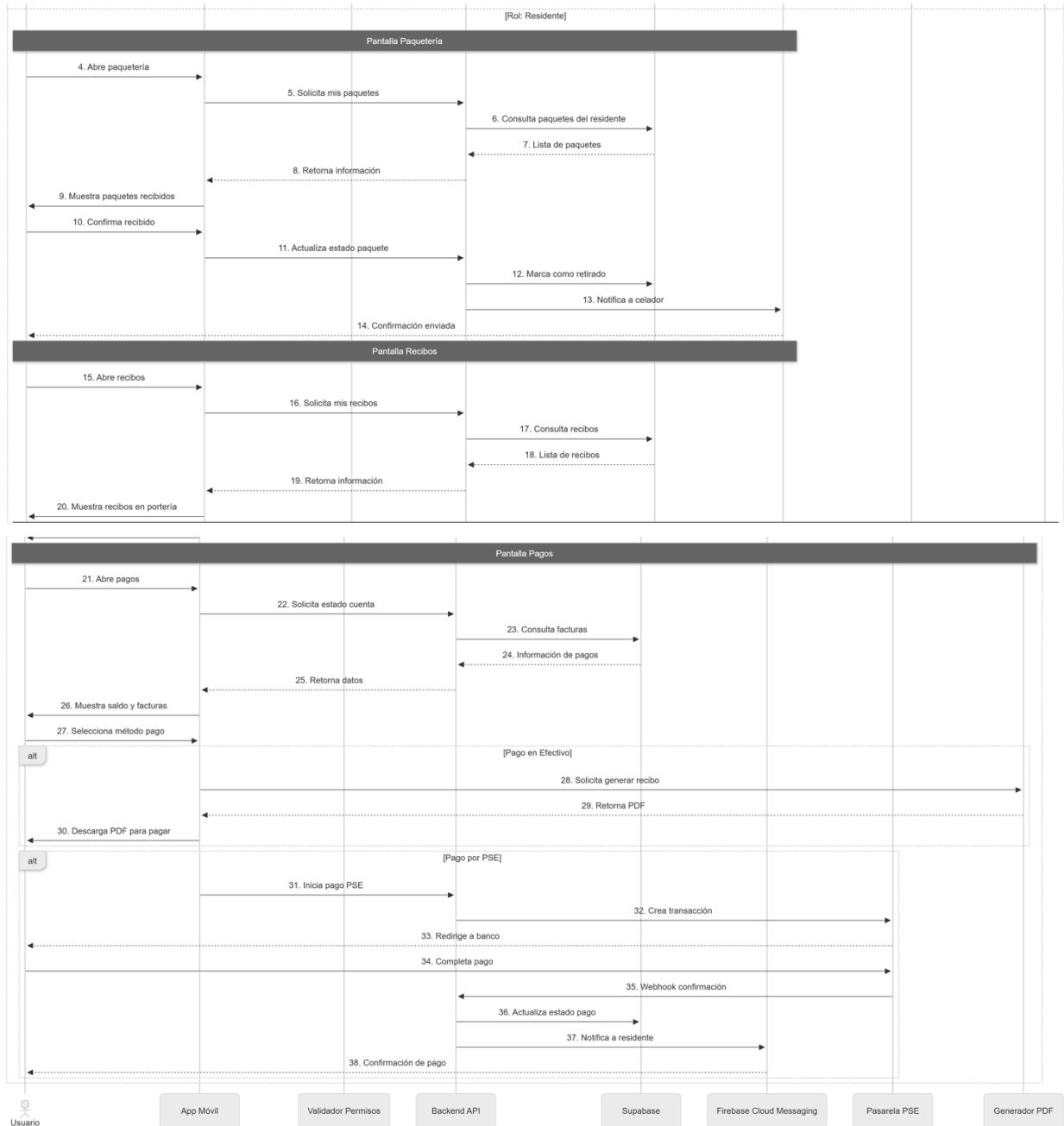


Diagrama 39, Diagrama de Estado Notificaciones (Paquetería, Recibos y Pagos) Residente

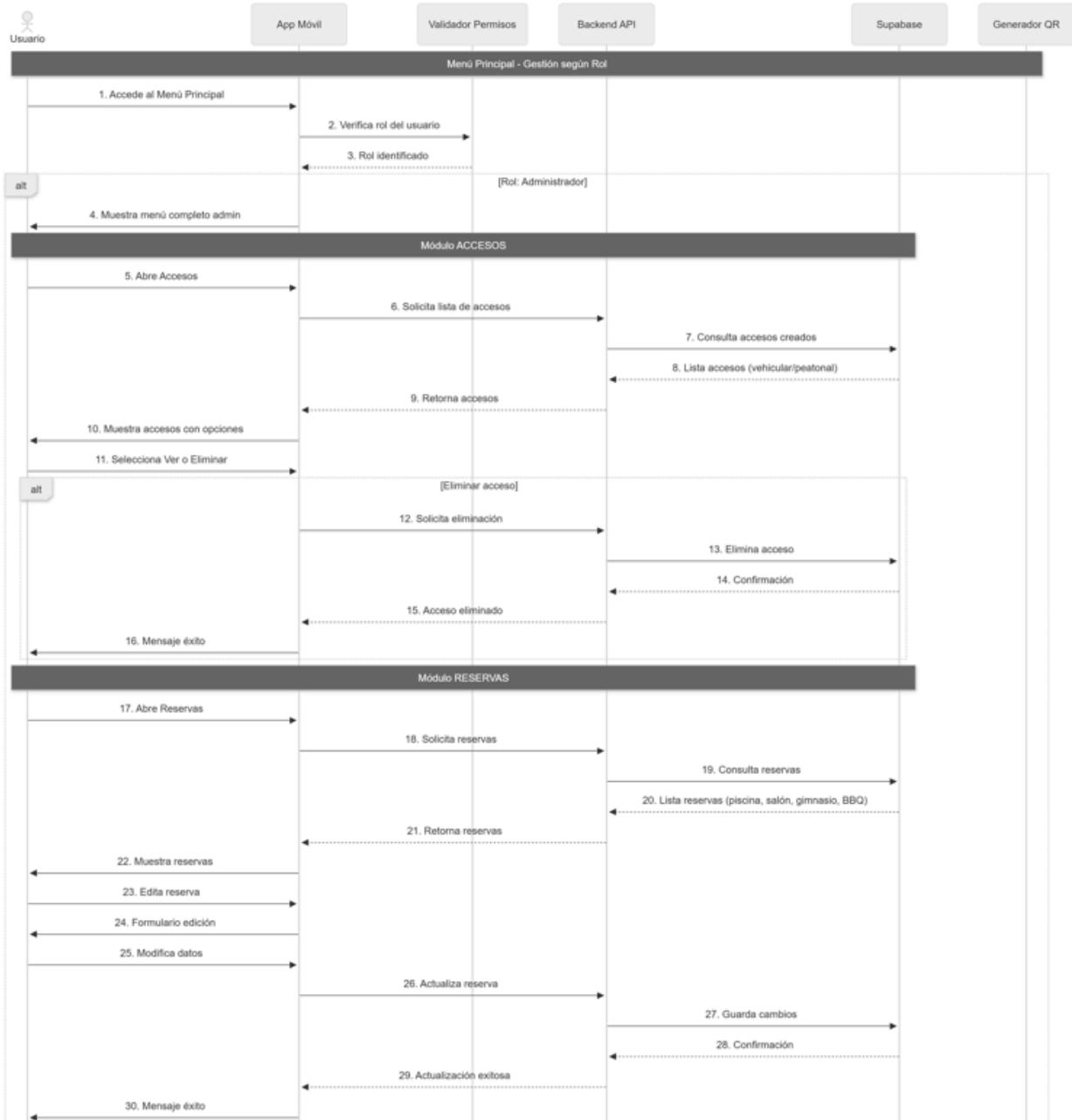


Diagrama 40, Diagrama de Estado Menú Principal (Accesos y Reservas) Administrador

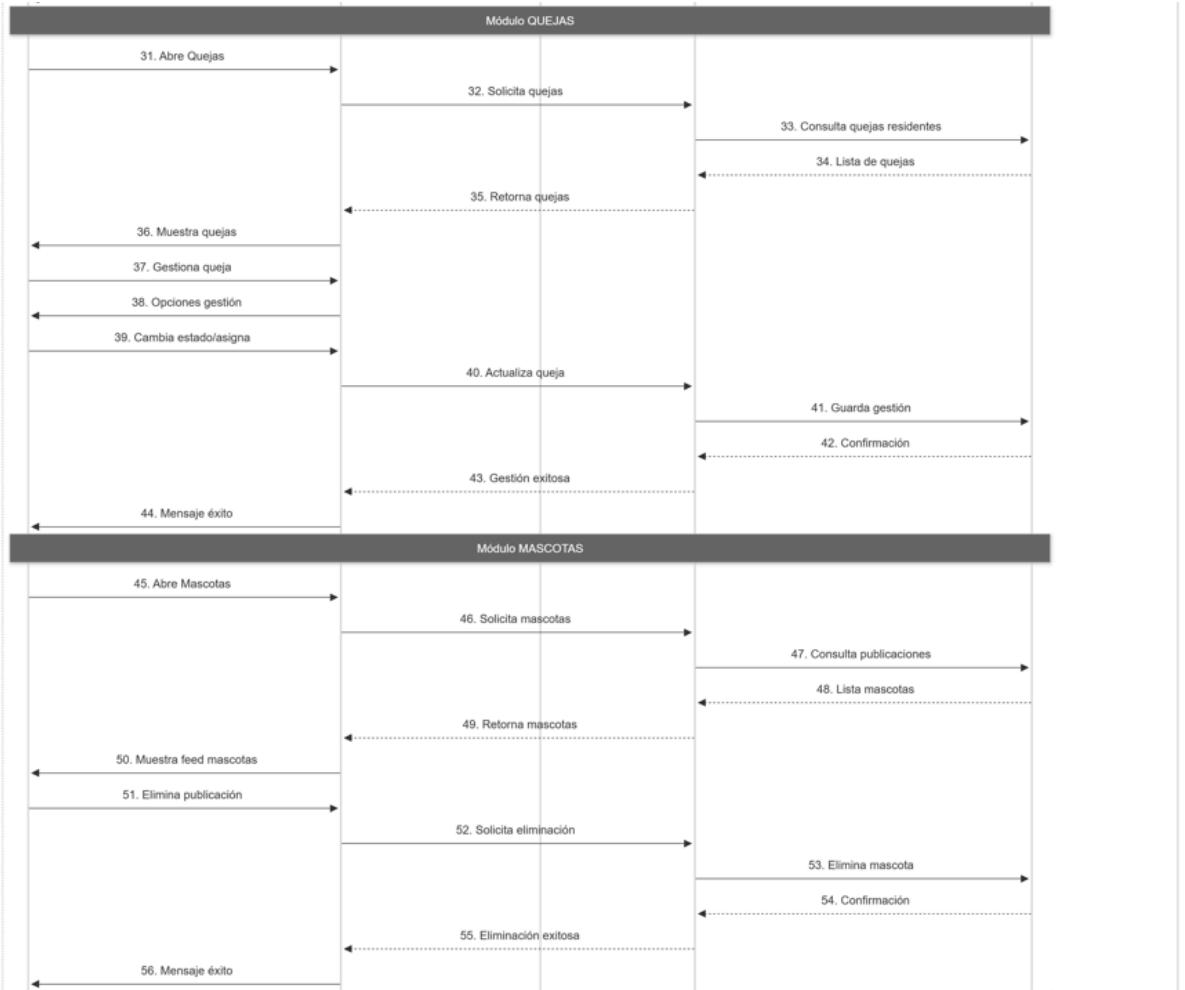


Diagrama 41, Diagrama de Estado Menú Principal (Quejas y Mascotas) Administrador

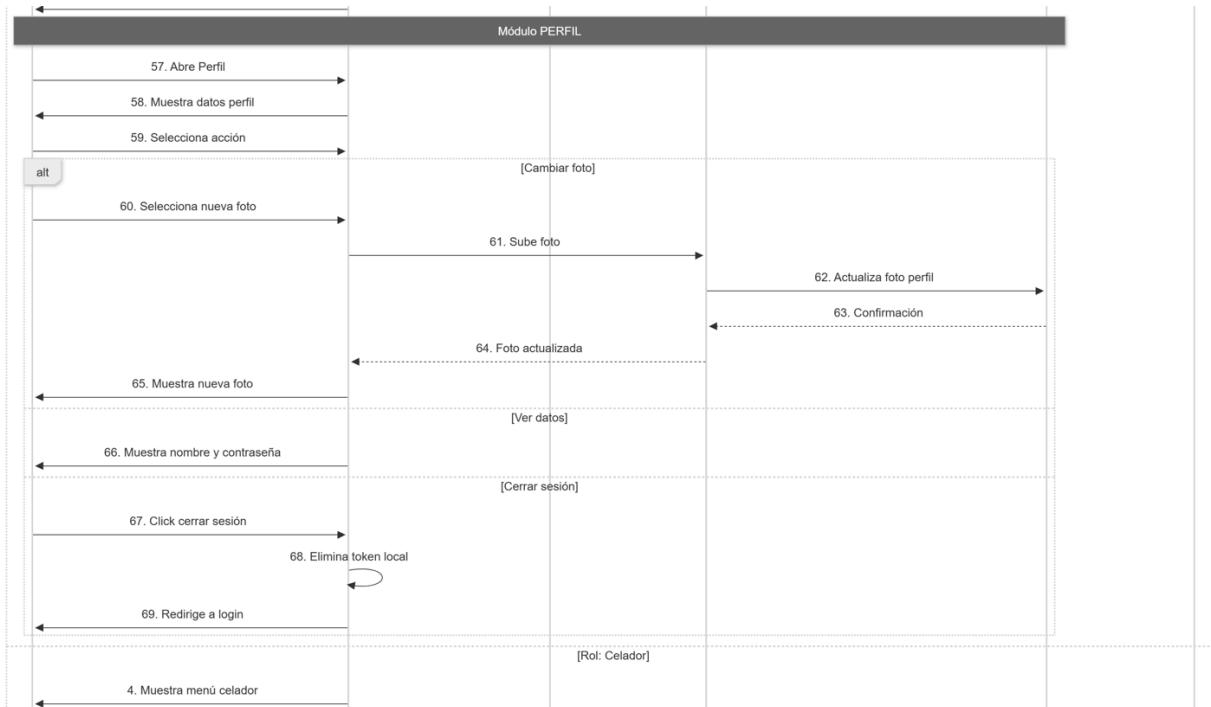


Diagrama 42, Diagrama de Estado Menú Principal (Perfil) Administrador

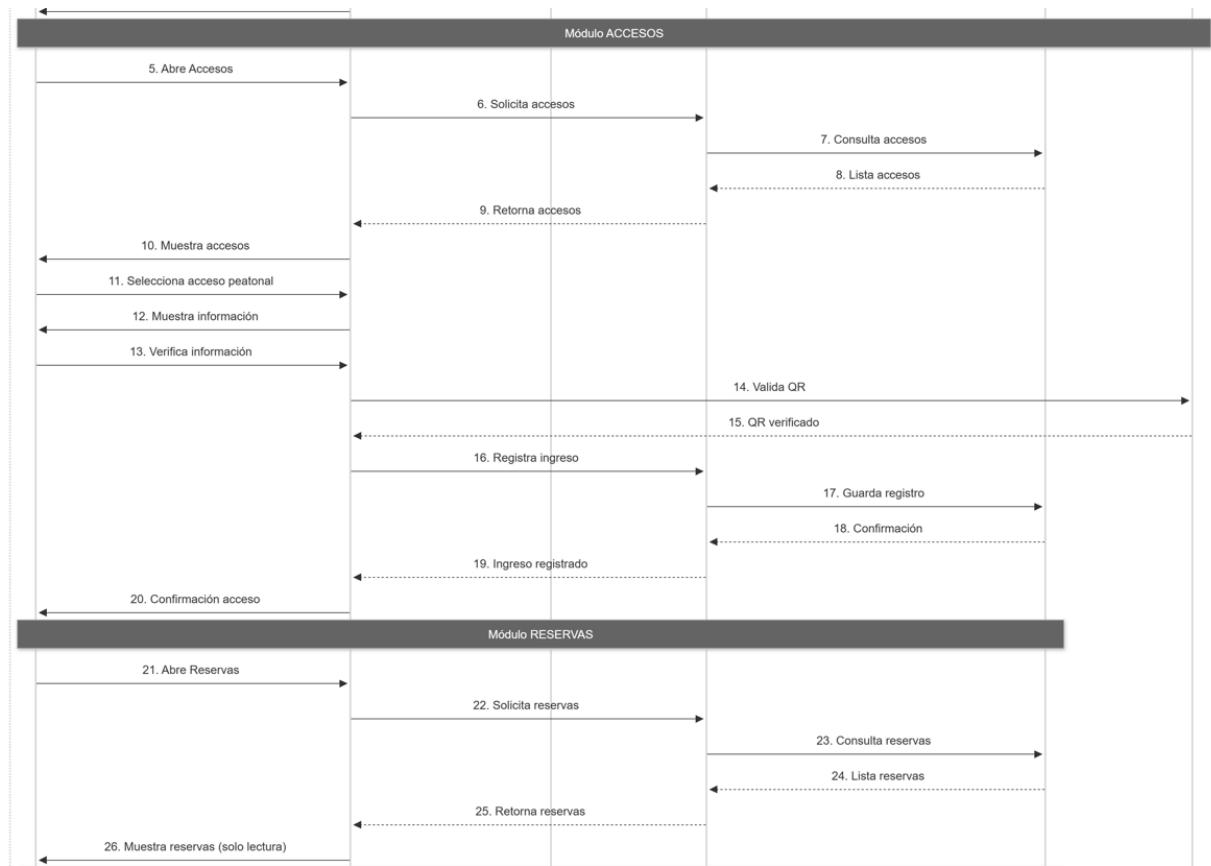


Diagrama 43, Diagrama de Estado Menú Principal (Accesos, Reservas) Celador

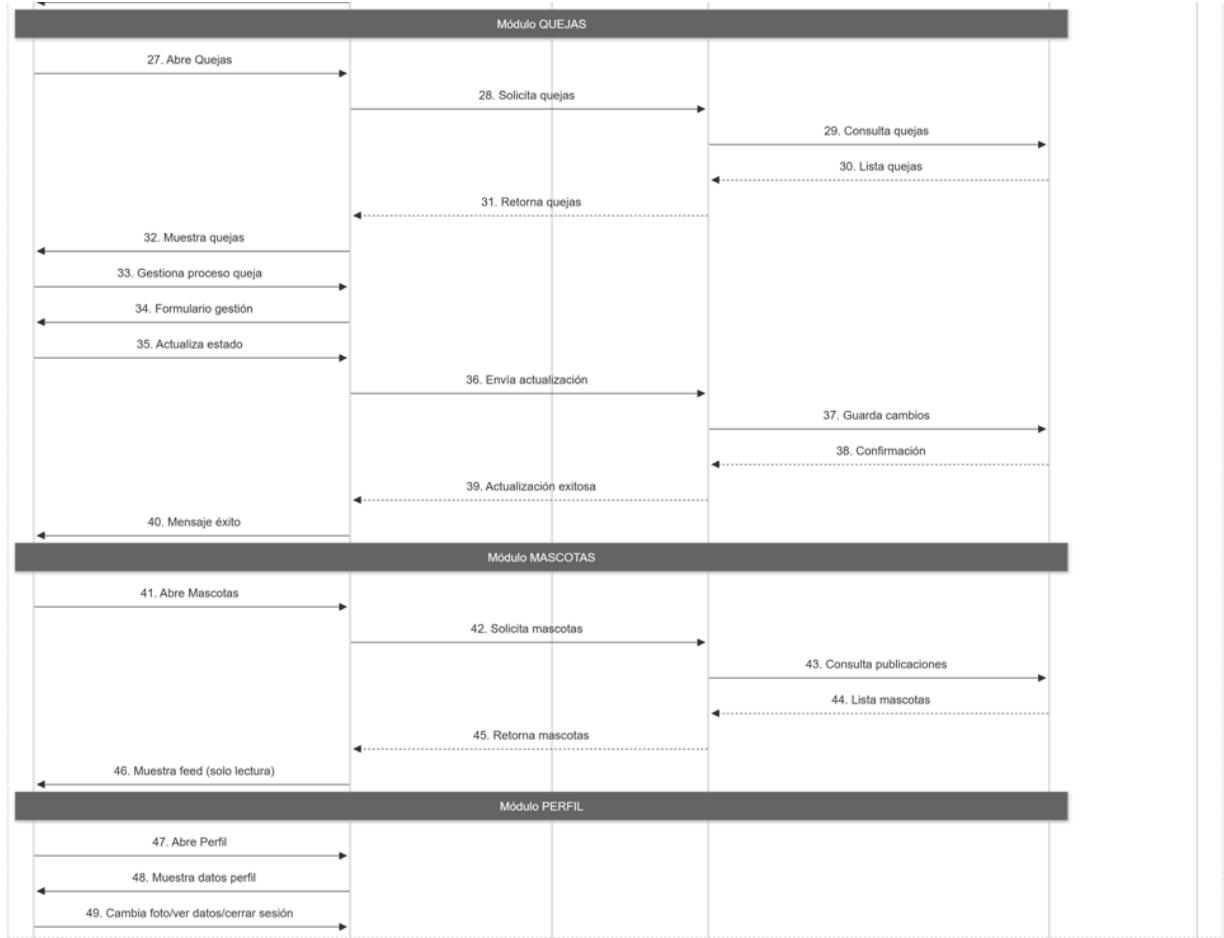


Diagrama 44, Diagrama de Estado Menú Principal (Quejas, Mascotas y Perfil) Celador

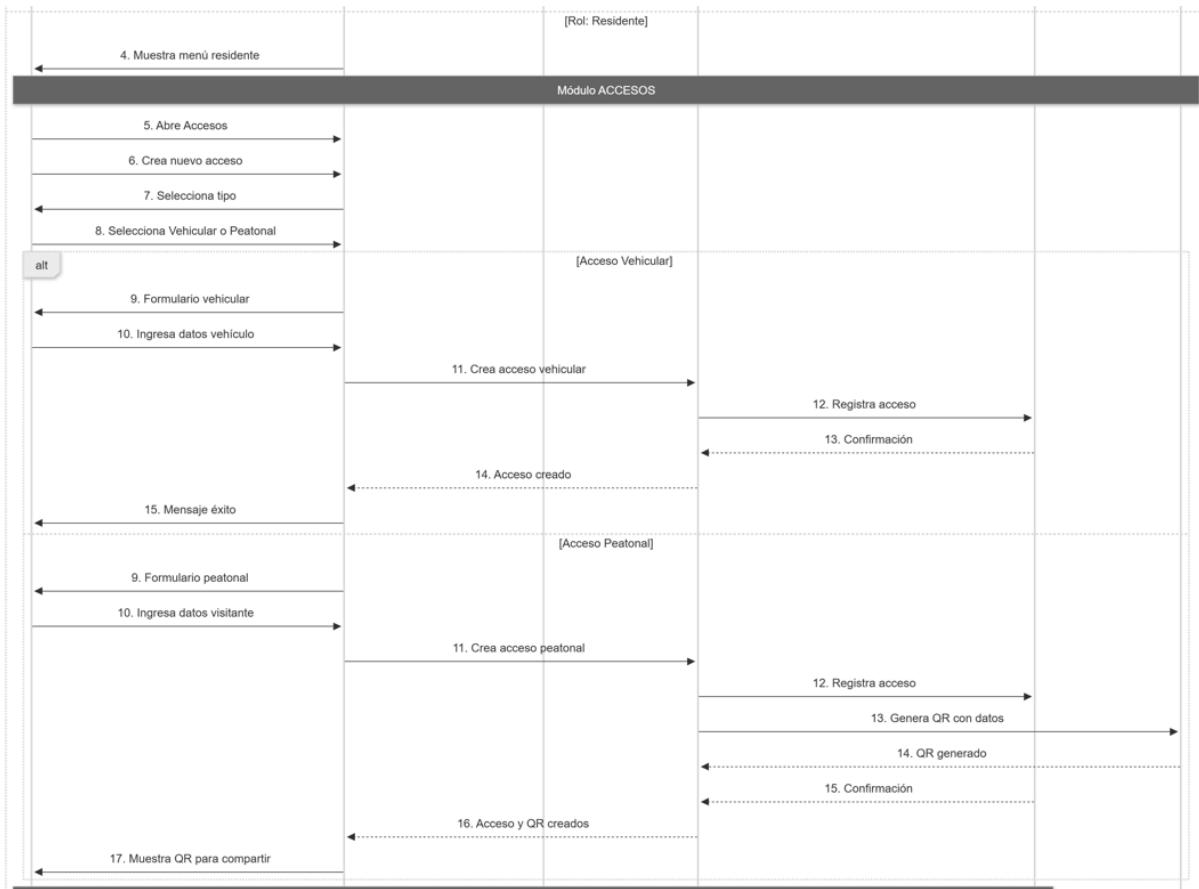


Diagrama 45, Diagrama de Estado Menú Principal (Accesos) Residente

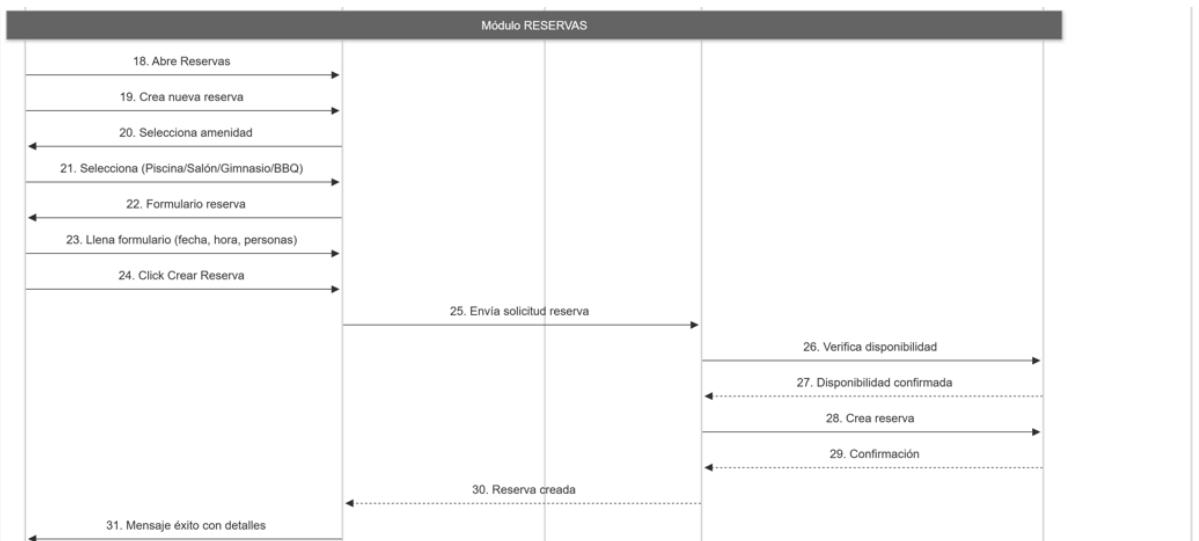


Diagrama 46, Diagrama de Estado Menú Principal (Reservas) Residente

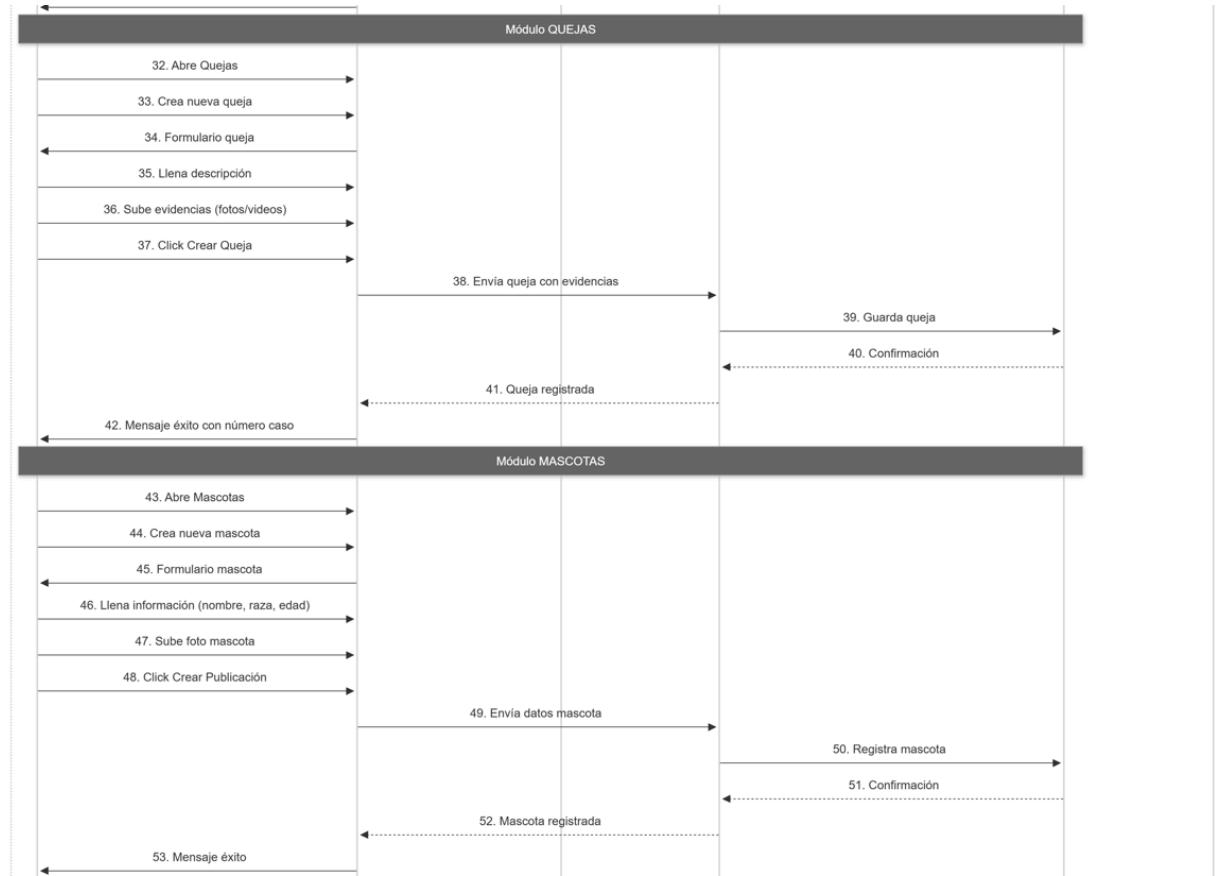


Diagrama 47, Diagrama de Estado Menú Principal (Quejas y Mascotas) Residente

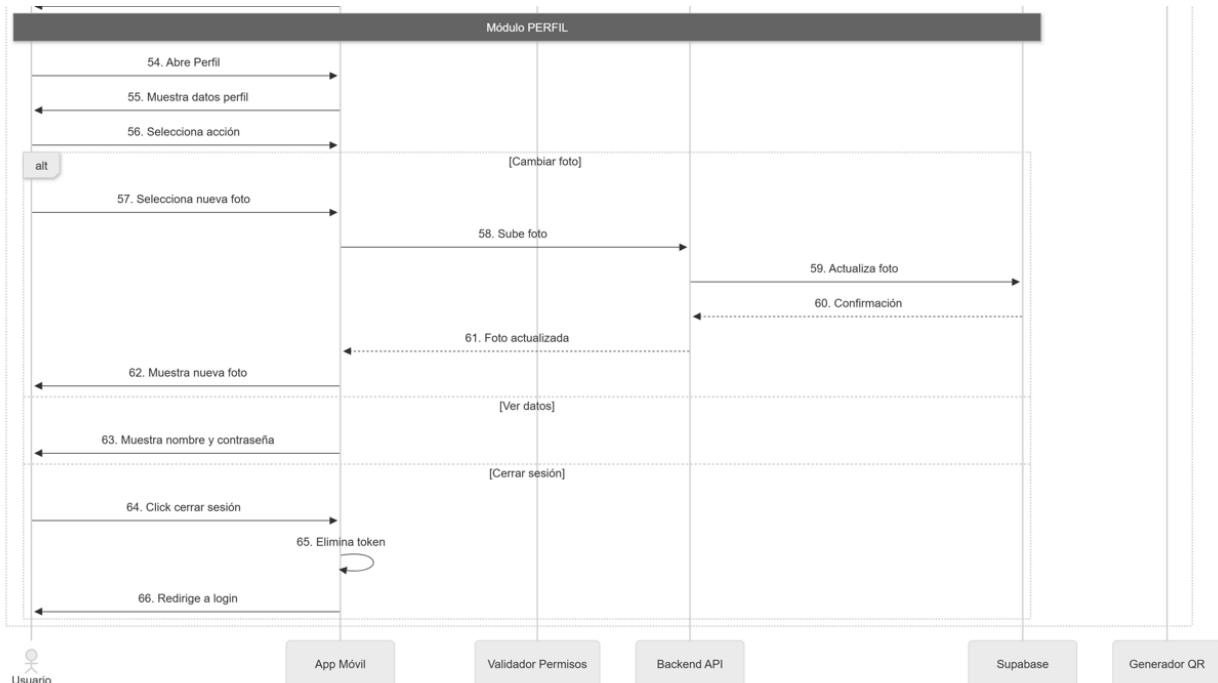


Diagrama 48, Diagrama de Estado Menú Principal (Perfil) Residente

14 Conclusiones

En el desarrollo de la aplicación móvil para la gestión del conjunto residencial Hacienda San Rafael se logró evidenciar una solución tecnológica integral que responde a las necesidades de comunicación, control y seguridad de la comunidad. La adopción de una arquitectura modular, junto con el uso de tecnologías modernas como Kotlin, Spring Boot y Supabase, permitió diseñar un sistema sólido, escalable y mantenable, que optimiza los procesos de inicio de novedades, notificaciones y el menú principal.

El modelado de la base de datos y la implementación de componentes clave, como el control de accesos mediante códigos QR, el módulo de notificaciones en tiempo real y la integración de pagos en línea, fortalecieron la confiabilidad y trazabilidad de la información. De igual manera, el análisis de riesgos, el marco legal y la consideración de normas de seguridad y protección de datos garantizan que la aplicación cumpla con los estándares de la ingeniería de software y las regulaciones vigentes.

La planificación en fases y la estimación de recursos evidencian un ciclo de desarrollo organizado, alineado con la metodología scrum que facilita la iteración y el control de calidad. Finalmente, este proyecto demuestra que la integración de buenas prácticas de ingeniería, un enfoque modular y un análisis contextual del entorno son factores determinantes para implementar soluciones tecnológicas que generan mayor eficiencia, seguridad y calidad de vida en las comunidades residenciales.

14.1 Recomendaciones

- Priorizar la mitigación de los riesgos de nivel Crítico y Alto, seguidos de los riesgos de nivel Medio. Los riesgos de nivel Bajo deben monitorearse pero requieren menor atención inmediata.
- Organizar jornadas de inducción para celadores, administración y residentes, explicando el uso de la aplicación y resolviendo dudas sobre funcionalidades.

- Definir un reglamento interno sobre cómo se gestionan las autorizaciones de visitas, reservas, quejas y pagos dentro de la plataforma.
- Asegurarse de que el manejo de la información cumpla con la **Ley 1581 de 2012** (protección de datos personales) y políticas de confidencialidad. Los residentes deben autorizar expresamente el uso de sus datos en la app.
- Preparar un procedimiento alterno en caso de fallas técnicas (ej. red caída en portería). El celador debe contar con una opción offline de validación temporal de accesos, sincronizando datos cuando vuelva la conexión.
- Establecer un plan de soporte técnico y mantenimiento preventivo. Se recomienda designar a un responsable interno que supervise las métricas (fallas de acceso, tiempos de carga, entregas de paquetes) y coordine con el equipo de desarrollo las mejoras.
- Considerar la aplicación como un sistema en evolución. Se recomienda planear actualizaciones periódicas para incluir nuevas funcionalidades, mejorar la experiencia del usuario y atender necesidades emergentes de la comunidad.

Referencias

- Universidad de los Andes. (2022). *Neivor: Software para la gestión de conjuntos residenciales en Colombia*. Recuperado de: <https://www.neivor.com>
- Kovive360. (2021). *Cohabit: plataforma de digitalización para propiedad horizontal*. Recuperado de: <https://www.kovive360.com>
- Mejía, J. (2017). *Webdificio: Plataforma digital para gestión de edificios y conjuntos residenciales en Colombia*. Bogotá: Revista Digital de Tecnología.
- Pressman, R. S., & Maxim, B. R. (2020). *Ingeniería del Software: Un enfoque práctico*. McGraw-Hill.
- Sommerville, I. (2019). *Software Engineering (10th ed.)*. Pearson Education.
- ISO/IEC. (2018). *ISO/IEC 25010: Systems and Software Quality Requirements and Evaluation (SQuaRE)*.
- ISO/IEC. (2013). *ISO/IEC 27001: Information Security Management Systems*.
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum*. Scrum.org.
- Ministerio TIC Colombia. (2022). *Política de Seguridad Digital y Transformación TIC en la Propiedad Horizontal*. Bogotá D.C.
- Android Developers. (2024). *Material Design 3 Guidelines*. Recuperado de: <https://developer.android.com>
- Supabase Inc. (2024). *Supabase Documentation*. Recuperado de: <https://supabase.com/docs>
- JetBrains. (2024). *Spring Boot Framework Overview*. Recuperado de: <https://spring.io/projects/spring-boot>