

# RecVet: Gestión digital de recetas veterinarias

---

Trabajo fin de Máster

**Autora:**

Alma María Mallo Casdelo

Máster Universitario en Ingeniería Informática

Desarrollo de aplicaciones sobre dispositivos móviles

**Consultores:**

Jordi Ceballos Villac

Jordi Almirall López

Enero 2021





Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-sinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)



## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>RecVet: Gestión digital de recetas veterinarias</i>
<b>Nombre del autor:</b>	<i>Alma María Mallo Casdelo</i>
<b>Nombre del consultor/a:</b>	<i>Jordi Ceballos Villac</i> <i>Jordi Almirall López</i>
<b>Nombre del PRA:</b>	<i>Robert Clarisó Viladrosa</i>
<b>Fecha de entrega (mm/aaaa):</b>	01/2021
<b>Titulación:</b>	<i>Máster Universitario en Ingeniería Informática</i>
<b>Área del Trabajo Final:</b>	<i>Desarrollo de aplicaciones sobre dispositivos móviles</i>
<b>Idioma del trabajo:</b>	<i>Español</i>
<b>Palabras clave</b>	<i>Recetas veterinarias, Android, Firebase</i>

### **Resumen del Trabajo (máximo 250 palabras):**

Los tratamientos veterinarios requieren la prescripción de recetas para la adquisición de medicamentos en farmacias. Estas recetas se realizan en papel por lo que es necesario acudir a recogerlas a la clínica veterinaria y llevarlas, posteriormente, a la farmacia. Esta situación se complica en el caso de animales mayores con tratamientos crónicos que requieren la renovación periódica de las recetas. Este trabajo pretende focalizar este problema y ayudar a solucionarlo a través de una herramienta que permite el tratamiento digital de recetas veterinarias.

El sistema desarrollado en este trabajo está formado por una aplicación Android, RecVet. Esta aplicación dispone de perfiles de usuario para propietarios de mascotas y veterinarios, pudiendo enlazarse y colaborar entre ellos. Los propietarios pueden gestionar los datos de sus mascotas y sus tratamientos, así como las recetas de estos. También pueden solicitar recetas a un veterinario con el que se han enlazado previamente y enviar las recetas a una farmacia. Los veterinarios tienen una lista de propietarios con los que están enlazados y pueden enviarles las recetas que estos les soliciten.

En este proyecto se ha utilizado el proceso de Diseño Centrado en el Usuario y metodología ágil con el desarrollo progresivo de productos funcionales situando las necesidades del usuario en el centro de cada etapa.

El desarrollo de la aplicación se ha realizado de forma nativa para Android con Android Studio. Se ha utilizado Firebase como backend, tanto para almacenamiento de información como para la gestión de usuarios.



**Abstract (in English, 250 words or less):**

Veterinary treatments require prescriptions for getting the medicines in pharmacies. These prescriptions are written on paper, so it is necessary to go and get them from the veterinary clinic and take them to the pharmacy. This situation gets hard in the case of older pets with chronic treatments that require the periodic renewal of prescriptions. This work aims to focus on this problem and to help solve it through a tool that allows digital treatment of veterinary prescriptions.

The system developed in this work consists of an Android application, RecVet. This application has user profiles for pet owners and veterinarians, and they can link and collaborate with each other. Owners can manage the data of their pets and their treatments, as well as their prescriptions. They can also request prescriptions from a veterinarian they have previously linked to and send the prescriptions to a pharmacy. Vets have a list of owners they are linked to and they can send the prescriptions they request to them.

In this project User Centered Design and agile methodology are used with the progressive development of functional products placing the user's needs at the center of each stage.

The development of the application has been done natively for Android with Android Studio. Firebase has been used as a backend, both for information storage and user management.



# Índice

1. Introducción .....	5
1.1. Contexto y justificación del Trabajo .....	5
1.2. Objetivos del Trabajo .....	5
1.2.1. Requerimientos funcionales.....	5
1.2.1.1 Incluidos en el alcance del TFM .....	5
1.2.1.2 No Incluidos en el alcance del TFM .....	6
1.2.2. Requerimientos no funcionales.....	7
1.2.2.1 Incluidos en el alcance TFM .....	7
1.2.2.2 No incluidos en el alcance TFM .....	7
1.3. Enfoque y método seguido .....	7
1.4. Planificación del Trabajo .....	8
1.4.1. Diagrama de Gantt .....	8
1.4.2. Tabla de hitos .....	9
1.4.3. Descripción de fases, hitos y actividades .....	9
1.4.3.1 PEC2: Diseño y arquitectura .....	9
1.4.3.2 PEC3: Implementación .....	10
1.4.3.3 Entrega final .....	12
1.5. Análisis de riesgos .....	12
1.5.1. Identificación de los riesgos principales .....	13
1.5.2. Tabla de riesgos .....	13
1.5.3. Medidas para mitigar los riesgos identificados.....	14
1.6. Breve resumen de productos obtenidos.....	14
1.7. Breve descripción del resto de capítulos de la memoria .....	14
2. Diseño centrado en el usuario .....	15
2.1. Usuarios y contexto de uso (Análisis) .....	16
2.1.1. Métodos de indagación seleccionados.....	16
2.1.1.1 Entrevistas en profundidad .....	16
2.1.1.2 Análisis competitivo.....	16
2.1.2. Desarrollo y resultados de los métodos de indagación.....	16
2.1.2.1 Entrevistas en profundidad .....	16
2.1.2.2 Análisis competitivo.....	22
2.1.3. Perfiles de usuario.....	24
2.2. Diseño conceptual.....	25
2.2.1. Personas.....	26
2.2.2. Escenarios de uso .....	27
2.2.3. Flujos de interacción.....	32
2.3. Prototipado .....	33
2.3.1. Sketches escaneados .....	34
2.3.2. Prototipo horizontal de alta fidelidad.....	35
2.4. Evaluación .....	41
2.4.1. Perfil sociodemográfico .....	41
2.4.2. Documento de screening .....	42
2.4.3. Cuestionarios pre-test .....	43
2.4.4. Escenarios y tareas.....	43
2.4.4.1 Escenario para los propietarios .....	44
2.4.4.2 Escenario para los veterinarios .....	44



2.4.5. Cuestionarios post-test.....	45
3. Diseño técnico .....	46
3.1. Casos de uso.....	46
3.2. Modelo de datos .....	58
3.3. Diseño de la arquitectura .....	60
3.3.1. Arquitectura del sistema.....	60
3.3.2. Arquitectura de la aplicación .....	60
4. Implementación .....	61
4.1. Desarrollo .....	61
4.1.1. Herramientas y APIs utilizadas .....	61
4.1.1.1 Android .....	61
4.1.1.2 Firebase.....	62
4.1.1.3 Firebase Open Source.....	62
4.1.2. Decisiones técnicas relevantes.....	62
4.1.2.1 Estructura del proyecto Android .....	62
4.1.2.2 Diseño de las interfaces de usuario .....	63
4.1.2.3 Modelos de las vistas y Live Data .....	64
4.1.2.4 Estructura de la UI: Navigation y Navigation Drawer.....	64
4.1.2.5 Paso de argumentos entre fragmentos con Safe Args.....	68
4.1.2.6 Recycler View para las listas de elementos.....	70
4.1.2.7 Estructura de la base de datos .....	70
4.1.2.8 Acceso a la bd, consultas y modificaciones.....	72
4.1.2.9 Gestión de usuarios con Firebase Authentication. ....	73
4.1.2.10 Tratamiento de imágenes con Firebase Storage.....	73
4.2. Pruebas .....	76
4.3. Funcionamiento de la aplicación .....	78
4.3.1. Funcionalidades comunes.....	78
4.3.2. Funcionalidades del perfil: propietario de mascotas .....	79
4.3.2.1 Inicio, menú y notificaciones.....	79
4.3.2.2 Mascotas con sus tratamientos y recetas .....	80
4.3.2.3 Enlace con veterinario y datos farmacia .....	81
4.3.3. Funcionalidades del perfil: veterinario .....	81
4.3.3.1 Inicio, menú y notificaciones.....	81
4.3.3.2 Propietarios de mascotas, mascotas y recetas.....	81
5. Conclusiones .....	82
5.1. Líneas de trabajo futuro .....	83
6. Glosario.....	84
7. Bibliografía.....	85
8. Anexos .....	86



# 1. Introducción

## 1.1. Contexto y justificación del Trabajo

---

El principal objetivo de este trabajo es el desarrollo de una aplicación para móviles que ayude a la gestión de la salud de las mascotas, centrada en la gestión digital de recetas veterinarias. Aunque, existen aplicaciones que permiten gestionar datos de salud de las mascotas y sus tratamientos, no he encontrado ninguna que permita la gestión de las recetas.

La motivación de esta idea viene a partir de mi situación personal. Tenía dos perros con tratamiento crónico con muchas medicinas y recetas. Para adquirir esas medicinas, muchas de la cuales se adquieren en farmacias, es necesario acudir a la clínica veterinaria a recoger la receta en papel y llevarla después a la farmacia. En algunos casos, no disponen de la medicina con lo que hay que volver más tarde a recogerla. Conozco a más personas en situaciones similares con una o más mascotas con tratamientos crónicos, lo que complica la gestión de los distintos tratamientos.

Esta situación se volvió más complicada durante la etapa de confinamiento a causa del COVID en la que nos encontramos a mediados de 2020, ya que obligaba a acudir de forma presencial en varias ocasiones e intercambiar papeles (las recetas) con otras personas.

Por estos motivos pensé en esta aplicación, con la que, además de facilitar a los propietarios de mascotas la gestión de las recetas de sus mascotas, ayudase a realizar de forma online el proceso de solicitud y envío de recetas a la farmacia, lo que reduciría las visitas presenciales a las imprescindibles.

## 1.2. Objetivos del Trabajo

---

En este apartado se describen los objetivos de este trabajo, indicando a alto nivel los requerimientos funcionales y no funcionales de la aplicación móvil. Se describen los requerimientos incluidos en el alcance de este trabajo fin de máster y también aquellos que no se han incluido.

### 1.2.1. Requerimientos funcionales

#### 1.2.1.1 Incluidos en el alcance del TFM

##### Generales

- RF1: La aplicación dispondrá de funcionalidades para dos perfiles de usuarios: propietarios de mascotas y veterinarios.
- RF2: Enlace de propietarios y veterinarios. El propietario podrá solicitar enlazarse con un veterinario. El enlace se hará efectivo cuando el veterinario acepte la solicitud.



### Específicos del perfil: propietario

- RF3: Gestión de mascotas: nombre, especie, raza, microchip, ...
- RF4: Para cada mascota, se puede gestionar sus tratamientos, es decir, la lista de medicamentos prescritos con sus dosis, duración del tratamiento y recetas asociadas.
- RF5: Solitud de receta a veterinario. Para poder utilizar esta funcionalidad, el veterinario y el propietario deben estar enlazados. El propietario solicitará la receta al veterinario, que recibirá la notificación, realizará la receta y se la enviará al propietario.
- RF6: Recepción de receta de un veterinario: Al recibir una receta en la aplicación, se recibiría una notificación de receta recibida. Se permitirá que el propio usuario adjunte una receta recibida por otros medios para que la aplicación pueda ser útil, aunque el propietario no tenga enlace con ningún veterinario.
- RF7: Consulta de recetas con distintos criterios
- RF8: Envío de las recetas recibidas a la farmacia. Para ello se permitirá guardar los datos de contacto de la farmacia y se guardará un histórico de envíos.

### Específicos del perfil: veterinario

- RF9: Lista de propietarios con sus mascotas. Recetas emitidas a cada mascota.
- RF10: Envío de recetas a los propietarios.
- RF11: Notificaciones: avisos de solicitudes de receta de los propietarios.

#### *1.2.1.2 No Incluidos en el alcance del TFM*

- RF12: Gestión de vacunas/desparasitaciones, incluyendo recordatorios de aviso fecha próxima de vacunación/desparasitación.
- RF13: Gestión de consultas veterinarias:
  - Solicitud de cita desde el perfil de propietario
  - Confirmación de citas por parte de los veterinarios.
  - En el perfil del veterinario se añadiría información de las consultas.
- RF14: Incluir un tercer perfil de usuario: farmacias.
  - Enlace de propietarios con farmacias.
  - Los propietarios enviarían las recetas a través de la aplicación a las farmacias con las que estuvieran enlazados.
  - Las farmacias confirmarían la recepción y enviarían un aviso a los propietarios cuando las medicinas estuviesen disponibles para recoger.
  - Generación de informe con la lista de medicamentos de recetas veterinarias dispensadas para enviar al colegio de farmacéuticos.



### 1.2.2. Requerimientos no funcionales

#### 1.2.2.1 Incluidos en el alcance TFM

- RNF1: La aplicación se podrá ejecutar en dispositivos Android.
- RNF2: Los datos se almacenarán en una base de datos online (Firebase) y, por tanto, es necesaria conexión a Internet para utilizar la aplicación.
- RNF3: La aplicación debe cumplir los estándares de usabilidad.

#### 1.2.2.2 No incluidos en el alcance TFM

- RF4: Almacenamiento offline que permita usar la aplicación sin conexión a internet y se sincronice al recuperar la conexión.
- RF5: Aplicación para dispositivos iOS
- RF6: Aplicación web, de forma que se pudiese utilizar también desde un ordenador.

### 1.3. Enfoque y método seguido

El enfoque elegido es el desarrollo de un producto nuevo, para poder tener control tanto de la calidad del producto como de las tecnologías a utilizar, desde el principio al final del desarrollo.

La plataforma objetivo es Android, la aplicación se dirige a móviles con esta plataforma. Se realizará una aplicación Android nativa realizando el desarrollo en lenguaje Java. Para el almacenamiento de los datos se utilizará la base de datos online Google Firebase. Utilizar una base de datos en la nube permite que un mismo usuario pueda utilizar la aplicación desde distintos dispositivos sin perder ni tener que migrar sus datos.

Se ha combinado la metodología de Diseño Centrado en el Usuario con la metodología de desarrollo ágil que consisten en el desarrollo progresivo de productos funcionales incrementando las funcionalidades en cada etapa.



Ilustración 1: Metodología ágil, ciclo de vida



Para el control de versiones se ha utilizado Git. El código fuente se ha almacenado en GitHub y para el seguimiento de tareas se ha utilizado su gestión de proyectos e “issues”.

## 1.4. Planificación del Trabajo

### 1.4.1. Diagrama de Gantt

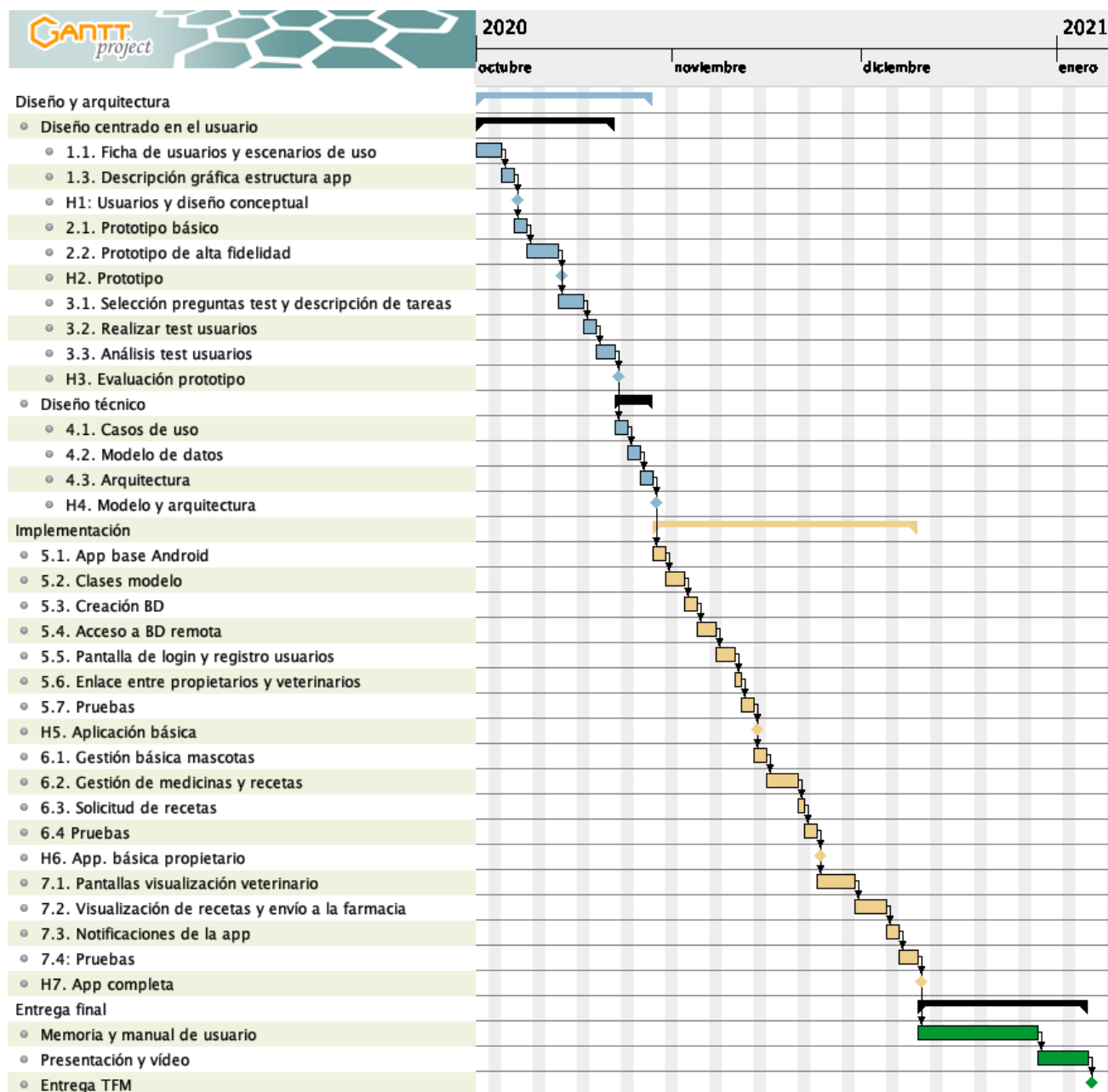


Ilustración 2: Diagrama de Gantt del proyecto



## 1.4.2. Tabla de hitos

Nombre	Duración	Inicio	Fin
<b>PEC2: Diseño y arquitectura</b>	<b>92 h</b>	<b>01/10/2020</b>	<b>28/10/2020</b>
Usuarios y diseño conceptual	20 h	01/10/2020	06/10/2020
Prototipo	26 h	07/10/2020	13/10/2020
Evaluación del prototipo	26 h	14/10/2020	22/10/2020
Arquitectura y modelo de datos	20 h	23/10/2020	28/10/2020
<b>PEC3: Implementación</b>	<b>138 h</b>	<b>29/10/2020</b>	<b>09/12/2020</b>
Aplicación básica con modelo de datos e identificación	48 h	29/10/2020	13/11/2020
Funcionalidad básica para el propietario	38 h	14/11/2020	24/11/2020
Funcionalidad completa de propietario y el veterinario	52 h	25/11/2020	09/12/2020
<b>Entrega final</b>	<b>116 h</b>	<b>10/12/2020</b>	<b>06/01/2021</b>
Escribir la memoria	80 h	10/12/2020	29/12/2020
Realizar la presentación y últimos retoques	36 h	30/12/2020	05/01/2021

**Dedicación:**

Del 1 de octubre al 23 de diciembre la dedicación al trabajo será de 2 horas al día durante los días laborables y 6 horas al día los fines de semana y festivos.

Del 24 de diciembre al 6 de enero la dedicación será de 6 horas diarias excepto los días 25 de diciembre, 1 de enero y 6 de enero.

## 1.4.3. Descripción de fases, hitos y actividades

En este apartado se describe brevemente cada una de las fases e hitos del proyecto. Una tarea transversal incluida en todas las fases es la documentación del trabajo realizado de forma que sólo se tengan que realizar pequeños retoques cuando se integre en la memoria. El tiempo dedicado a esta tarea está ya integrado en la planificación, el tiempo asignado a las actividades incluye esta tarea.

## 1.4.3.1 PEC2: Diseño y arquitectura

Esta etapa, correspondiente a la PEC 2 se centra en el diseño de la aplicación, aplicando las técnicas de diseño centrado en el usuario y realizando también el diseño técnico de la aplicación.

**Hito 1: Usuarios y diseño conceptual**

El objetivo es caracterizar a los usuarios y escenarios de uso, así como comenzar a caracterizar la interfaz de la aplicación.

**Actividades:**

- Actividad 1.1: Ficha de usuarios y escenarios de uso. Duración: 14 h.



- Actividad 1.2: Descripción gráfica de la estructura general de la aplicación. Duración: 6 h.

### Hito 2: Prototipo

A partir de la caracterización de usuarios y escenarios, se realizará un prototipo de la aplicación, realizándolo de forma básica inicialmente hasta realizar un prototipo de alta fidelidad con una herramienta de prototipado (la idea que tengo es utilizar Pencil porque es una herramienta que ya conozco).

Actividades:

- Actividad 2.1: Diseño del prototipo básico en papel. Duración: 6 h.
- Actividad 2.2: Diseño del prototipo de alta fidelidad. Duración: 20 h.

### Hito 3: Evaluación prototipo

Se evaluará la aplicación con usuarios a través del prototipo de alta fidelidad. Para ello, se seleccionarán un conjunto de preguntas y tareas para que realicen los usuarios durante el test. Finalmente se analizarán los resultados del test para realizar una propuesta de mejoras.

Actividades:

- Actividad 3.1: Selección de preguntas para los test de usuario y tareas a realizar por los usuarios en el test. Duración: 12 h.
- Actividad 3.2: Realización de test con usuarios. Duración: 5 h.
- Actividad 3.3: Análisis de test de usuarios, informe de mejoras. Duración: 9 h.

### Hito 4: Arquitectura y modelo de datos

El objetivo de esta fase es realizar el diseño técnico de la aplicación, definiendo principalmente los casos de uso, modelo de datos y arquitectura de la aplicación.

Actividades:

- Actividad 4.1: Definición y diagrama de casos de uso. Duración: 8h.
- Actividad 4.2: Descripción y diagramas del modelo de datos. Duración: 8h.
- Actividad 4.3: Descripción y diagrama de arquitectura. Duración: 4h.

#### 1.4.3.2 PEC3: Implementación

Durante esta fase se realiza la implementación de la aplicación en Android. Se ha planteado como un desarrollo incremental, estableciendo hitos con aplicaciones funcionales con distintos grados de terminación.

Todas las tareas de esta fase incluyen la tarea documentación correspondiente en el código fuente.

### Hito 5: Aplicación básica con modelo de datos e identificación



El objetivo de esta fase es obtener una aplicación Android funcional en la que un usuario se pueda registrar y posteriormente acceder a la aplicación con su perfil. También se podrá realizar el enlace entre propietarios y veterinarios, necesario para añadir funcionalidades en las fases siguientes.

Se establece como requisito previo la creación de la base de datos, así como la implementación del modelo de datos de la aplicación y la conexión y obtención de información de la base de datos, tareas que se realizarán durante esta fase.

Actividades:

- Actividad 5.1: Creación de aplicación base de Android (retoques sobre la base creada en la pec1). Duración: 4 h.
- Actividad 5.2: Implementación de las clases del modelo de datos. Duración: 9 h.
- Actividad 5.3: Creación de base de datos con entidades. Duración: 8 h.
- Actividad 5.4: Conexión de las clases con la base de datos remota y acceso a los datos. Duración: 10 h.
- Actividad 5.5: Implementación de la pantalla de login y registro de usuarios. Duración: 8 h.
- Actividad 5.6: Implementación del “enlace” entre propietarios y veterinarios: Duración: 4 h.
- Actividad 5.7. Pruebas de conexión y obtención de datos de la base de datos y de las funcionalidades implementadas. Duración: 5 h.

### **Hito 6: Aplicación con funcionalidad básica para el propietario**

A partir de la aplicación base, se comenzarán a añadir funcionalidades, comenzando por las funcionalidades básicas para el propietario.

El objetivo de esta fase es una aplicación Android donde un usuario con perfil de propietario pueda, además de acceder a la aplicación:

- Gestionar los datos de sus mascotas incluyendo medicinas y recetas, aún de forma básica.
- Realizar la solicitud de una receta. Dicha solicitud no llegará todavía al perfil del veterinario porque no se ha implementado todavía, quedará almacenada en la base de datos.

Actividades:

- Actividad 6.1: Gestión básica de mascotas. Duración: 10 h.
- Actividad 6.2: Gestión de medicinas y recetas. Duración: 14 h.
- Actividad 6.3: Implementación de la solicitud de recetas. Duración: 8 h.
- Actividad 6.4: Pruebas de las funcionalidades implementadas. Duración: 6 h.

### **Hito 7: Aplicación completa**



Al terminar esta fase se completarán las funcionalidades incluidas en el alcance de este trabajo. El resultado será una aplicación totalmente funcional para los dos perfiles de usuario.

Actividades:

- Actividad 7.1: Implementación de las pantallas de visualización del perfil del veterinario: propietarios, mascotas, recetas solicitadas y envío de recetas a los propietarios. Duración: 20 h.
- Actividad 7.2: Visualización del propietario de las recetas recibidas y envío de recetas a la farmacia. Duración: 12 h.
- Actividad 7.3: Implementación de notificaciones a los usuarios. Duración: 10 h.
- Actividad 7.4: Pruebas de las funcionalidades implementadas y de toda la aplicación en conjunto. Duración: 10 h.

### 1.4.3.3 Entrega final

El objetivo de esta fase es obtener la memoria del trabajo y un manual de usuario de la aplicación.

#### **Hito 9: Escribir la memoria del trabajo y el manual de usuario**

Actividades:

- Actividad 9.1: Unificar todas las notas y documentos parciales que se hayan realizado a lo largo del trabajo para integrarlas en la plantilla de la memoria. Duración: 6 h.
- Actividad 9.2: Revisar y añadir las consideraciones necesarias a los apartados que se han realizado en fases anteriores. Duración: 10 h.
- Actividad 9.3: Escribir el resto de los apartados para completar la memoria y el manual de usuario. Duración: 64 h.

#### **Hito 10: Realizar la presentación y últimos retoques**

El objetivo es obtener el vídeo de la presentación y realizar los últimos retoques del proyecto para entregarlo.

Actividades:

- Actividad 10.1: Preparar presentación con diapositivas. Duración: 16 h.
- Actividad 10.2: Grabar vídeo de la presentación. Duración: 14 h.
- Actividad 10.3: Revisar todo y realizar la entrega. Duración: 6 h.

## 1.5. Análisis de riesgos

---

En este apartado se realiza un pequeño análisis de riesgos, identificando los principales riesgos a los que se enfrenta el proyecto, cuales sería su efecto en caso de que se ocurriesen y cómo podrían mitigarse.



### 1.5.1. Identificación de los riesgos principales

Los principales riesgos en este proyecto son los siguientes:

- R01. Las tecnologías seleccionadas funcionan de forma diferente a la esperada.
- R02. Disponer de menos tiempo del planificado para la realización del TFM.
- R03. Mala planificación de alguna etapa
- R04. Grandes cambios en el diseño

En la tabla siguiente se indica su descripción, causas y consecuencias, así como la probabilidad de ocurrencia de cada uno y el impacto sobre el proyecto en caso de que ocurriese. Y, posteriormente, se indican las acciones a tomar para mitigarlos.

### 1.5.2. Tabla de riesgos

Cód	Nombre	Causa	Descripción	Consecuencia	P	I
R01	Tecnologías funcionan de forma diferente a la esperada	Poca experiencia con las tecnologías a utilizar	Podría ocurrir que ocurriesen imprevistos durante la implementación	Sería necesario invertir más tiempo del planificado	A	A
R02	Disponer de menos tiempo del planificado	Imprevistos no relacionados con el TFM (trabajo, enfermedad, ...)	Debido a que, realizo el TFM al mismo tiempo que trabajo, podría ocurrir que en ocasiones disponga de menos tiempo del previsto. Podría ocurrir lo mismo por causa de una enfermedad.	Retraso en las tareas	M	A
R03	Mala planificación de alguna etapa	Haber subestimado la carga de trabajo	No se han planificado correctamente las tareas de una etapa, el tiempo de dedicación planeado es inferior al necesario	Retraso en la finalización de esa etapa lo que podría implicar la finalización del proyecto fuera de plazo	M	A



R04	Grandes cambios en el diseño	La evaluación con usuarios provoca cambios importantes	Durante la evaluación se detecta que se deben realizar cambios de gran envergadura en el diseño	Se debe rediseñar esa parte de la aplicación	M	M
-----	------------------------------	--	---	--	---	---

Leyenda:

P: Probabilidad de ocurrencia del riesgo - I: Impacto del riesgo

A: Alto – M: Medio – B: Bajo

### 1.5.3. Medidas para mitigar los riesgos identificados

Riesgo	Acción para mitigarlo
R01. Tecnologías funcionan de forma diferente a la esperada	Anticiparme lo máximo posible al problema mirando documentación y realizando pequeñas implementaciones de prueba para comprobar su funcionamiento para saber como realizarlo durante la etapa de implementación.
R02. Disponer de menos tiempo del planificado	Avanzar lo máximo posible en las primeras etapas adelantándose a las fechas de fin de las tareas para ir ganando días de margen por si los necesitase después.
R03. Mala planificación de alguna etapa	Revisar periódicamente la planificación para detectar a tiempo estos errores de planificación y poder ajustar la planificación cuando todavía hay margen de tiempo para hacerlo
R04. Grandes cambios en el diseño	Caracterizar bien a los usuarios y realizar un buen diseño centrado los usuarios y siguiendo las normas de usabilidad

### 1.6. Breve resumen de productos obtenidos

Los entregables finales de este proyecto serán:

- El producto desarrollado, en este caso, una aplicación Android.
- El código fuente del proyecto.
- Un manual de usuario de la aplicación.
- Una memoria describiendo el trabajo realizado.
- Una presentación en vídeo.

### 1.7. Breve descripción del resto de capítulos de la memoria

Los próximos capítulos de la memoria se centrarán en:

- **Análisis y diseño centrado en el usuario:** Se describirá la aplicación del diseño centrado en el usuario en este proyecto.
- **Diseño técnico:** Se describirán el diseño del software desde el punto de vista técnico: modelo de datos, arquitectura, ...
- **Implementación y pruebas:**
  - Se describirán decisiones tomadas en la fase de implementación y pruebas. Esto incluye las herramientas, librerías, ... utilizadas para el desarrollo.



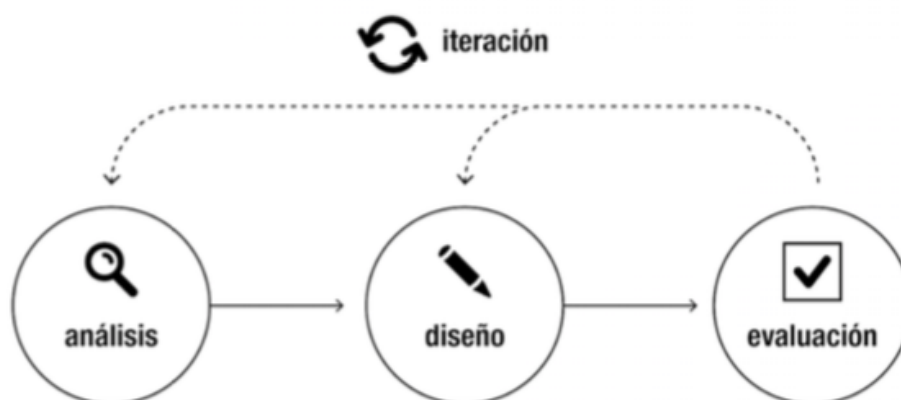
- Se describirá el tipo de pruebas a realizar y las pruebas que se han llevado a cabo.
- También se realizará un análisis de la planificación, analizando las desviaciones si las hubiese.
- **Conclusiones y trabajo futuro:** Análisis del trabajo realizado, describiendo los objetivos alcanzados y aquellos que no se han alcanzado indicando el motivo. Se describirá también cómo se podría continuar el trabajo.
- **Glosario**
- **Bibliografía**

## 2. Diseño centrado en el usuario

El Diseño Centrado en el Usuario (DCU) es una metodología utilizada para crear productos que cumplan las necesidades de los usuarios.

Se trata de que el usuario sea el centro del desarrollo del producto desde el inicio, no sólo para evaluar el resultado. Si el usuario sólo participa en la evaluación, estaríamos desperdiciando mucho trabajo previo que habrá que rehacer si al evaluar el producto no cumple con las necesidades y requerimientos del usuario.

Para llevar a cabo un buen diseño centrado en el usuario, el usuario debe ser el centro del desarrollo desde el comienzo. Tenemos que conocer el contexto en el que se utilizará la aplicación que vamos a desarrollar, conocer a los usuarios que la utilizarán y cuales son sus necesidades. Y, partir de ahí, definir los requerimientos que tendrá la aplicación, diseñarla en base a esos requerimientos y posteriormente evaluar el diseño también con usuarios. Este proceso es iterativo, en caso de no satisfacer las necesidades de los usuarios, debemos volver a fases anteriores y repetir el proceso.



*Ilustración 3: Fases del Diseño Centrado en el Usuario  
(Fuente: Diseño centrado en el Usuario - UOC)*



### 2.1. Usuarios y contexto de uso (Análisis)

---

#### 2.1.1. Métodos de indagación seleccionados

Aunque la aplicación a desarrollar está principalmente orientada a propietarios de mascotas, también se ha contemplado su uso por veterinarios, y por tanto los métodos de indagación deben ir orientados a ambos tipos de usuarios.

He seleccionado los siguientes métodos de indagación:

##### *2.1.1.1 Entrevistas en profundidad*

Utilizo este método de análisis porque las personas con las que hablé no quisieron que las observase y tampoco les gustaba la idea de hacer un diario. Accedieron a la entrevista ya que de esta forma sólo tendrían que dedicar una cantidad determinada de tiempo a esta tarea y después podrían volver a su rutina diaria.

A pesar de que, con este método se pierde la información del contexto, con unas preguntas adecuadas se puede obtener información cualitativa importante para el análisis.

##### *2.1.1.2 Análisis competitivo*

Para complementar la indagación, he pensado que podría ser útil conocer aplicaciones de temática similar, aunque no estén centradas en el ámbito veterinario.

He comenzado realizando una búsqueda de aplicaciones pensadas para propietarios de mascotas pero que contemplasen también a los veterinarios de forma que se pudiese llevar a cabo comunicación entre ellos.

He ampliado un poco la búsqueda incluyendo aquellas en las que, parte de la funcionalidad se incluye en una página web o aplicación de escritorio y no sólo en la aplicación móvil.

He buscado también aplicaciones relacionadas con las recetas electrónicas médicas, ya que no encontrado ninguna que contemple esa posibilidad para las recetas veterinarias y pienso que puede ser interesante ver la forma en que lo gestionan.

#### 2.1.2. Desarrollo y resultados de los métodos de indagación

##### *2.1.2.1 Entrevistas en profundidad*

Para las entrevistas he preparado un guion para los propietarios y otro para los veterinarios.



## Guion para los propietarios

El primer bloque de preguntas sirve para conocer un poco mejor al usuario y el uso que suele hacer del teléfono móvil.

El siguiente bloque, se centra en conocer a sus mascotas y el tercer bloque, trata de conocer como es su rutina de petición/adquisición de recetas en veterinario y farmacia.

El último bloque se centra en el uso de aplicaciones móviles de temática relacionada y conocer su opinión sobre una aplicación con la funcionalidad de la que se quiere desarrollar en este trabajo.

### 1. Conocer al usuario

- Nombre y edad
- Lugar de residencia, profesión y situación laboral
- ¿Cuánto tiempo sueles utilizar el móvil a diario?
- ¿Dónde sueles utilizarlo?
- ¿Qué tipo de aplicaciones sueles utilizar en el móvil?

### 2. Conocer a sus mascotas

- ¿Cuántas mascotas tienes? ¿Podrías indicarme sus nombres, especie y edad?
- ¿Alguna de ellas tiene alguna patología/enfermedad?
- ¿Alguna de ellas toma medicación crónica o durante largos periodos de tiempo?

### 3. Conocer rutina con las recetas/medicaciones

- ¿Adquieres las medicinas directamente en la clínica veterinaria o las adquieres en una farmacia?
- En caso de que sean de farmacia, ¿con qué frecuencia acudes al veterinario a recoger recetas?
- ¿Cómo de fácil o difícil te resulta acercarte en persona a recogerlas? Indica el motivo (falta de tiempo, está lejos y tengo que acudir en coche, horarios poco compatibles/incompatibles con mi trabajo, ...)?
- ¿Le ha pedido alguna vez a tu veterinario/a que te enviase la receta por algún medio electrónico?
- Si la respuesta anterior es sí, ¿te la ha enviado? ¿qué medio ha utilizado (email, whatsapp, ...)?
- ¿Has enviado alguna vez a la farmacia una receta por un medio digital (email, whatsapp, ...)?
- ¿Alguna vez te has olvidado de ir a renovar una receta? ¿Cómo lo has solucionado?
- ¿Te resulta más cómoda la recepción (del veterinario) y envío (a la farmacia) por medio digital o acudir en persona? Si nunca has usado el medio digital, ¿crees que te resultaría más cómodo?

### 4. Uso de aplicaciones

- ¿Conoces alguna aplicación que te permita gestionar los datos de los tratamientos de tu mascota? ¿Las usas? ¿Alguna permite llevar control de las recetas o su gestión digital?



- ¿Utilizas alguna aplicación no específica de recordatorios para no olvidarte de las fechas de renovación de las recetas?
- ¿Encontrarías útil una aplicación que permitiese gestionar de forma digital las recetas de tus mascotas (solicitar renovaciones de recetas a los veterinarios, llevar registro, enviarlas a la farmacia, ...)? ¿La usarías?
- ¿Qué funcionalidades adicionales crees que debería incluir?

### Guion para los veterinarios

El primer bloque de preguntas sirve para conocer un poco mejor al usuario y el uso que suele hacer del teléfono móvil.

El siguiente bloque, se centra en conocer como suele funcionar la emisión de recetas a los propietarios y cual es su relación con ellos.

El último bloque se centra en el uso de aplicaciones móviles de temática relacionada y conocer su opinión sobre una aplicación con la funcionalidad de la que se quiere desarrollar en este trabajo.

#### 1. Conocer al usuario

- Nombre y edad
- ¿Cuánto tiempo sueles utilizar el móvil a diario? ¿Dónde sueles utilizarlo?
- ¿Qué tipo de aplicaciones sueles utilizar en el móvil?
- ¿Sueles utilizar aplicaciones móviles que te ayuden con tu trabajo como veterinario?

#### 2. Conocer la rutina de tratamientos/recetas

- ¿Sueles tratar a muchas mascotas con tratamientos crónicos o de larga duración? ¿Algunos de esos medicamentos se adquieren en las farmacias?
- ¿Cómo es el procedimiento de las recetas de medicamentos que se adquieren en la farmacia?
- ¿Algún propietario te ha pedido que le envíes las recetas por un medio digital en lugar de venir a recogerlas en persona?
- Si la respuesta anterior es sí, ¿lo has hecho?

#### 3. Uso de aplicaciones

- ¿Conoces alguna aplicación móvil que te permita enviar recetas a los propietarios de las mascotas?
- ¿Utilizarías una app para que los propietarios de las mascotas que tratas te soliciten las recetas a través e ella y tu se las envíes también a través de la app?
- Si la respuesta anterior es no, indica el motivo.

### Entrevistas realizadas:



El número de participantes en las entrevistas han sido cinco voluntarios, tres propietarios y dos veterinarios. Los propietarios han sido seleccionados entre conocidos que tienen alguna mascota y los veterinarios son también dos conocidos.

A través de las entrevistas con estas personas se podrá profundizar mejor en los problemas y necesidades de los posibles usuarios de la aplicación.

- **Pedro, hombre de 35 años. Propietario de 2 perros.**

Trabaja como ingeniero industrial en una empresa, trabaja a jornada partida, lo que le viene bien para ir a sacar a sus perros al mediodía, pero le deja muy poco tiempo para hacer recados o tareas fuera del trabajo. Usa el móvil sobre 2-3 horas al día, normalmente en casa y por la calle cuando pasea a sus perros.

Tiene dos perros, ambos de edad avanzada y con enfermedades crónicas por lo que ambos toman medicinas diariamente.

Una parte de estas medicinas las adquiere en la farmacia, con lo que, periódicamente acude al veterinario a por recetas de estas medicinas, normalmente sobre 1 o 2 veces al mes. Eso no siempre es fácil porque su horario de trabajo coincide con el de la clínica y no le gusta pedir horas en el trabajo para ir a buscar recetas, prefiere reservarlas para cuando tiene que llevar a consulta a sus perros.

No usa ninguna aplicación para la gestión de la salud de sus mascotas, pero sí usa recordatorios de Google Calendar desde que un día se olvidó de que tenía que ir a renovar una receta.

Nunca le ha pedido a su veterinario que le envíe las recetas de forma digital, pero piensa que le resultaría más cómodo. Le ha parecido útil tener una aplicación para gestionar las recetas veterinarias de forma digital, ha comentado que le gustaría que las recetas veterinarias funcionasen de forma parecida a las recetas digitales del Sergas (sistema de salud gallego, dónde el médico activa las recetas y con tu tarjeta sanitaria acudes a la farmacia, dónde la leen y te dan las medicinas) aunque entiende no es fácil que se pueda conseguir algo así pronto.

- **María, mujer de 30 años. Propietaria de 1 gata.**

Trabaja como administrativa en la universidad. Usa el móvil sobre 2-3 horas al día, normalmente cuando está en casa y a veces por la calle.

Tiene una gata de 7 años, con un problema renal crónico. Suele comprar las medicinas de su gata directamente en el veterinario, aunque a veces tiene que darle alguna medicina de farmacia.

No utiliza ninguna aplicación para la gestión de salud de su gata ni recordatorios para las citas veterinarias o medicinas.



Nunca le ha pedido su veterinario que le enviase una receta de por medio digital porque no se le ocurrió esa posibilidad.

Sobre utilizar una aplicación para la gestión de las medicinas de su gata y sus recetas, ha dicho que, aunque cree que ahora no le aportaría mucho, la probaría de todas formas porque piensa que quizá le podría ser útil en el futuro si su gata tuviese que tomar más medicinas o si adoptase otro gato. Ha comentado que le gustaría que la aplicación incluyese también vacunas, desparasitaciones y consultas veterinarias, que así le parecería más útil.

- **Manuel, hombre de 50 años. Propietario de 3 perros y dos gatos.**

Es carpintero de profesión, pero está retirado desde hace dos años. Usa el móvil sobre 3-4 horas al día, normalmente en casa y por la calle. Suele utilizar todo tipo de aplicaciones.

Varias de sus mascotas tienen tratamientos crónicos, inicialmente compraba todos en su veterinario, pero su veterinario le dijo que podría comprar alguno en alguna farmacia online especializada en veterinaria porque lo conoce desde hace tiempo y sabe que le cuesta hacer frente a todos los gastos veterinarios que tiene, que le haría las recetas sin problema. Ahora adquiere algunos medicamentos de esta forma y, cuando hace el pedido tiene que enviar antes la receta que le da su veterinario por email a la farmacia online para le permitan adquirir los medicamentos.

Alguna vez se ha olvidado de pedir con tiempo las medicinas en la farmacia online, por suerte la pudo conseguir en su veterinario, pero sí que vería útil tener una aplicación que le ayudase a organizar el pequeño lío que tiene. Desde que se olvidó de esta receta utiliza recordatorios en el móvil, pero encontraría útil una aplicación que le permitiese ver las recetas anteriores, medicinas, ... Le gustaría que incluyese también vacunas, desparasitaciones y consultas veterinarias.

- **Antonio, hombre de 52 años. Veterinario.**

Propietario de su clínica veterinaria desde hace 20 años. Es una clínica pequeña, sólo son él y una auxiliar de veterinaria.

Utiliza el móvil sobre dos horas en casa y en la consulta. Usa todo tipo de aplicaciones, algunas especializadas, de algún instrumento que tiene en la clínica.

Trata a muchas mascotas ancianas con lo que, tiene que hacer recetas con frecuencia sus propietarios. Suele hacer las recetas en papel y el propietario las lleva a la farmacia. Este año algún propietario le ha pedido que le enviase las recetas por email y él lo ha hecho.

Usaría una aplicación como la propuesta para fidelizar a sus clientes, pero tendría que ser sencilla. Le gustaría que estuviese conectada con su



aplicación de gestión de la clínica y le gustaría que existiese un sistema de receta electrónica para los veterinarios.

- **Lucía, mujer de 43 años. Veterinaria.**

Actualmente es propietaria de su clínica, en la que trabajan también dos auxiliares de veterinaria.

Utiliza el móvil sobre 2-3 horas en casa, la calle y la consulta. Usa todo tipo de aplicaciones, pero sobre todo aplicaciones de mensajería. A veces usa alguna aplicación específica, tiene en la clínica alguna máquina con aplicación móvil.

Trata a varias mascotas con enfermedades crónicas y en muchos casos las medicinas que receta se adquieren en las farmacias. En estos casos el procedimiento habitual es que el propietario de las mascotas se acerque a su clínica a recoger las recetas en personas y después acuda a la farmacia con la receta para adquirir el medicamento.

Ha confirmado que, también en su caso ha enviado alguna receta por email a algunos propietarios que se lo han pedido, principalmente durante los últimos meses.

Sobre si usaría una aplicación para enviar las recetas a los propietarios de las mascotas que trata, dice que tendría que mirarla pero que si fuese sencilla de utilizar, la usaría y que, lo ideal sería poder usar esa funcionalidad en su aplicación de gestión de la clínica o que la aplicación móvil se comunicase con su aplicación de alguna forma.

### **Conclusiones extraídas de las entrevistas:**

- El uso del móvil es predominante en casa o por la calle en el caso de los propietarios de mascotas y, en el caso de los veterinarios, a veces también en el trabajo con alguna aplicación específica
- Las recetas digitales les parecen prácticas a todos, pero en la mayoría de los casos no lo había visto como una posibilidad para las recetas veterinarias.
- En general, aunque ninguno de los propietarios entrevistados había utilizado una aplicación de temática similar a la propuesta, suele ser porque no se les había ocurrido, pero en general les parece útil.
- Los veterinarios entrevistados, aunque no se mostraron especialmente interesados en la aplicación, accederían a usarla para fidelizar a sus clientes siempre que fuese muy sencilla de usar.
- En general, la aplicación propuesta, parece que resultaría más útil a propietarios de mascotas que a veterinarios, aunque podría ser útil también para estos si se pudiese comunicar con su aplicación de gestión de la clínica.
- Un sistema de recetas electrónicas sería bien recibido tanto por propietarios de mascotas como veterinarios. Han valorado de forma positiva este trabajo en ese sentido, aunque no se trate realmente de una implementación de receta electrónica sino un sistema que simplemente

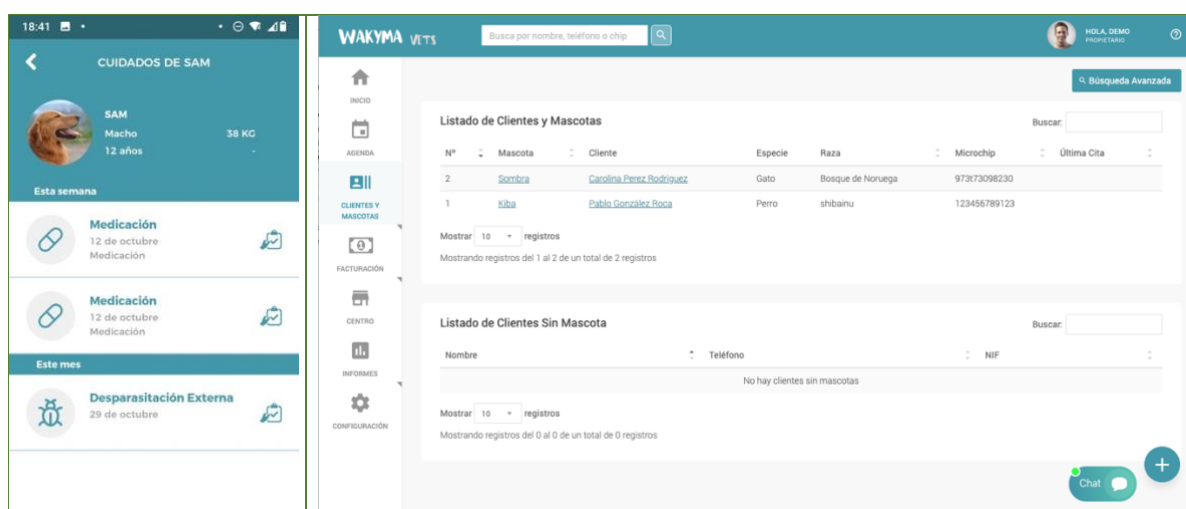


ayuda a realizar las gestiones de las recetas pero que recuerda que la receta electrónica veterinaria sería un objetivo al que llegar algún día.

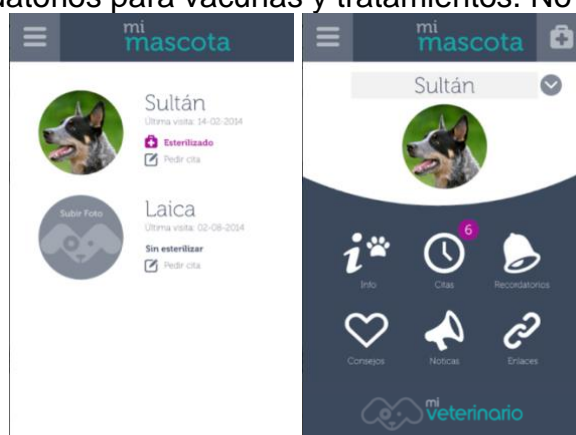
### 2.1.2.2 Análisis competitivo

En la búsqueda de aplicaciones y webs de temáticas similar pensadas para propietarios de mascotas pero que contemplasen también a los veterinarios, he seleccionado las siguientes:

- **Wakyma:** Aplicación móvil para los propietarios de mascotas y aplicación web para los veterinarios. Desde la aplicación móvil permite a los propietarios gestionar los tratamientos de varias mascotas e incluir el código de un veterinario para enlazar su aplicación con ese veterinario. Los veterinarios desde su aplicación web, pueden gestionar su clínica: gestión de propietarios y mascotas, consultas, facturación. Ni la aplicación web ni la móvil contemplan las recetas.



- **Mi veterinario:** Aplicación sólo para propietarios de mascotas clientes de clínicas que tienen el software Winvet. Permite pedir cita para consultas y poner recordatorios para vacunas y tratamientos. No incluye recetas.



En la búsqueda de aplicaciones relacionadas con las recetas electrónicas, he seleccionado las siguientes (ninguna de ellas es para mascotas):



- **Receta Médica:** Pensada para médicos de consultas privadas y sus pacientes. Los médicos pueden enviar recetas digitalmente a sus pacientes a través de esta aplicación. Los pacientes desde su aplicación pueden consultar su lista de recetas y solicitar nuevas recetas al médico.

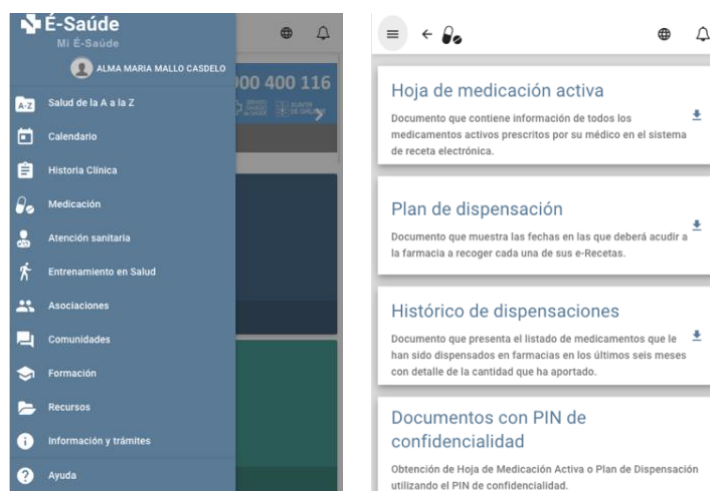


- **Calendario receta electrónica:** Pensada para particulares, para llevar control de las fechas de vencimiento de las recetas electrónicas, dosificación de medicinas y citas médicas. Se introducen manualmente los datos y el sistema envía notificaciones para avisar a usuario en las fechas claves.



- **E-saúde:** no es aplicación sino una página web, aunque adaptada a distintos tipos de dispositivos. Es la web para pacientes del Sergas, el sistema de salud que tenemos en Galicia. Tiene varios niveles de seguridad para el acceso y, con el nivel más alto (a través de certificado) se pueden consultar el historial médico, incluyendo las pruebas realizadas y las recetas que tiene disponibles el paciente, con las fechas en las que están disponibles.





### 2.1.3. Perfiles de usuario

A continuación, se muestran los perfiles de usuario identificados a partir de la fase de indagación. En cada perfil se indica, sus características, los contextos en los que usará la aplicación, el conjunto de tareas que necesitará para conseguir sus objetivos y un listado de características descubiertas en la fase de indagación que deberían estar presentes en la interfaz.

Usuario: propietario de mascotas con enfermedades	
Características del perfil	<p>Este perfil está formado por personas que tienen una o más mascotas, alguna o varias de ellas con enfermedades que implica que tengan que tomar medicaciones durante largos periodos de tiempo.</p> <p>El rango de edad suele ser una persona entre 18 y 55 años, suele usar el móvil con frecuencia, sobre 2 horas diarias o más.</p> <p>El objetivo fundamental de estos usuarios es encontrar una forma cómoda de gestionar los medicamentos para sus mascotas, ya que, suele ser complicado sobre todo para aquellas personas que tienen varias mascotas.</p>
Contextos	<p>Este grupo de usuarios utilizará con mayor frecuencia la aplicación cuando se encuentre en su casa y en el caso de los propietarios de perros, posiblemente durante el paseo de estos.</p>
Tareas	<ul style="list-style-type: none"> <li>• Registrarse en la aplicación</li> <li>• Iniciar sesión en la aplicación</li> <li>• Añadir mascotas</li> <li>• Añadir tratamientos para una mascota</li> <li>• Enlazar con un veterinario</li> <li>• Solicitar recetas al veterinario</li> <li>• Añadir recetas</li> <li>• Añadir datos de la farmacia</li> </ul>



	<ul style="list-style-type: none"> <li>• Enviar recetas a la farmacia</li> <li>• Consultar recetas por distintos filtros</li> </ul>
Características descubiertas	<ul style="list-style-type: none"> <li>• Las notificaciones son fundamentales, la aplicación debe incluir necesariamente notificaciones de caducidad de las recetas.</li> <li>• Incluir la gestión de vacunas, desparasitaciones y consultas veterinarias con notificaciones sería valorado de forma positiva por muchos usuarios de este grupo (*)</li> <li>• Sería positivo ver un listado de recetas con distintos filtros</li> </ul>

(\*) Incluir la gestión de vacunas, desparasitaciones y consultas veterinarias se había considerado como parte de los requisitos que se escapan del alcance de este trabajo, pero es importante saber que hay usuarios que consideran que tiene sentido incluirlo en la aplicación.


Usuario: veterinario	
Características del perfil	<p>Este perfil está formado por personas que ejercen como veterinarios y tienen clientes con una o varias mascotas con patologías.</p> <p>El rango de edad suele ser una persona mayor de 30 años, suele usar el móvil con frecuencia, sobre 2 horas diarias o más.</p> <p>El objetivo fundamental de estos usuarios es fidelizar a sus clientes, facilitándoles la gestión de las recetas.</p>
Contextos	Este grupo de usuarios utilizará con mayor frecuencia la aplicación cuando se encuentre en su trabajo.
Tareas	<ul style="list-style-type: none"> <li>• Registrarse en la aplicación</li> <li>• Iniciar sesión en la aplicación</li> <li>• Aceptar enlace con propietario</li> <li>• Hacer una receta solicitada por un propietario</li> <li>• Consultar propietarios y mascotas</li> <li>• Consultar las recetas</li> </ul>
Características descubiertas	<ul style="list-style-type: none"> <li>• La interfaz tiene que ser muy sencilla, con pocas opciones</li> <li>• Deberían tener que introducir la menor información posible, sólo la imprescindible.</li> </ul>

## 2.2. Diseño conceptual



### 2.2.1. Personas

Persona 1	
 <p>Sara (<a href="#">fuente</a>)</p>	<p>Sara tiene 35 años, trabaja como informática y sigue estudiando para mejorar su situación laboral. También es voluntaria en una protectora de animales.</p> <p>Trabaja mucho y no dispone de demasiado tiempo libre con lo que intenta optimizar la forma de realizar las tareas diarias para poder cumplir con todas sus obligaciones.</p> <p>Tiene dos perros y acoge temporalmente a perros o gatos de la protectora tras una operación o si necesitan algún cuidado especial cuando está de vacaciones o durante algún fin de semana.</p>
<b>Objetivos</b>	<ul style="list-style-type: none"> <li>• Organizar los tratamientos de sus mascotas</li> <li>• No olvidarse de pedir las recetas con tiempo suficiente</li> </ul>
<b>Comportamientos</b>	<p>Usa el móvil durante varias horas al día.</p> <p>No tiene miedo de probar una aplicación nueva si le resulta útil</p>
<b>Necesidades</b>	<p>Encontrar una forma sencilla y ágil de llevar la gestión de los tratamientos y recetas de las mascotas que tiene a cargo.</p>

Persona 2	
 <p>Luis (<a href="#">fuente</a>)</p>	<p>Luis tiene 45 años, tiene una clínica veterinaria.</p> <p>Ha montado su propia clínica y todavía tiene sólo un empleado con lo que trabaja muchas horas para sacarla adelante y poder atender lo mejor posible a las mascotas que trata. Hace él mismo las urgencias de noche y fines de semana.</p>
<b>Objetivos</b>	<p>Dar el mejor trato posible a sus clientes y conseguir nuevos clientes para que la clínica prospere y así poder contratar más gente y no tener que trabajar tantas horas.</p>
<b>Comportamientos</b>	<p>No tiene miedo de las nuevas tecnologías.</p>



	Utiliza nuevas aplicaciones en el móvil que le puedan resultar útiles tanto en su clínica como en su vida privada.
<b>Necesidades</b>	Encontrar formas de fidelizar clientes y de conseguir otros nuevos.

### 2.2.2. Escenarios de uso

EU1: Dar de alta una mascota	
Perfil de usuario	Propietario de mascotas con enfermedades
Persona	Sara
Contexto	Sara acude a un refugio y adopta a Jack, un perro que llegó ya anciano al refugio y, la protectora para incentivar su adopción ayuda al adoptante con sus gastos veterinarios ya que Jack padece varias enfermedades crónicas. Sara se conmueve con su historia y no duda en adoptarlo.
Objetivo	Dar de alta a Jack en la aplicación para gestionar el tratamiento de sus enfermedades
Tareas	<ul style="list-style-type: none"> <li>• Añadir una mascota</li> </ul>
Necesidades de información	Los datos básicos de Sam: especie, raza, edad, nº de microchip, ...
Funcionalidades	Alta de una mascota
Desarrollo de las tareas	<ol style="list-style-type: none"> <li>1. [Registro en la aplicación]</li> <li>2. Iniciar sesión</li> <li>3. Seleccionar el apartado "Mascotas"</li> <li>4. Añadir una nueva mascota</li> </ol>

EU2: Dar de alta tratamientos de una mascota	
Perfil de usuario	Propietario de mascotas con enfermedades
Persona	Sara
Contexto	Sara ha llevado a Jack a casa, le ha enseñado dónde está su cama, el agua y la comida. Jack que estaba cansado del ajetreo de venir desde el refugio, se acuesta en su cama y se duerme. Sara se sienta en el sillón que hay al lado de su cama y decide revisar todos los tratamientos que debe tomar Jack y ver si tiene todas las medicinas.
Objetivo	Dar de alta los tratamientos de Jack para organizarse y no olvidarse de nada.
Tareas	<ul style="list-style-type: none"> <li>• Añadir tratamiento a una mascota</li> </ul>
Necesidades de información	Nombre del medicamento, dosis y periodicidad de las tomas, duración del tratamiento.



Funcionalidades	<ul style="list-style-type: none"> <li>Alta de tratamiento</li> </ul>
Desarrollo de las tareas	<ol style="list-style-type: none"> <li>[Registro en la aplicación]</li> <li>Iniciar sesión</li> <li>Seleccionar el apartado "Mascotas"</li> <li>[Añadir una nueva mascota]</li> <li>Seleccionar una mascota</li> <li>Añadir tratamiento para esa mascota</li> </ol>

## EU3: Enlazar con un veterinario

Perfil de usuario	Propietario de mascotas con enfermedades
Persona	Sara
Contexto	Sara ha revisado los tratamientos de Jack y se ha dado cuenta que se le está terminando uno de los tratamientos. El veterinario de Jack ya se ha registrado en la aplicación porque ella se lo pidió cuando habló con él la primera vez pero ella olvidó hacer el enlace.
Objetivo	Enlazarse con el veterinario para poder solicitarle recetas de los tratamientos de Sam.
Tareas	<ul style="list-style-type: none"> <li>Enlazarse con un veterinario</li> </ul>
Necesidades de información	Email con el que el veterinario se ha registrado en la aplicación
Funcionalidades	<ul style="list-style-type: none"> <li>Enlazar con un veterinario</li> </ul>
Desarrollo de las tareas	<ol style="list-style-type: none"> <li>[Registro en la aplicación]</li> <li>Iniciar sesión</li> <li>Seleccionar el apartado "Veterinarios"</li> <li>Seleccionar la opción "Enlazar con veterinario"</li> <li>La solicitud se completará cuando el veterinario al acepte, mientras quedará "pendiente"</li> </ol>

## EU4: Solicitar receta

Perfil de usuario	Propietario de mascotas con enfermedades
Persona	Sara
Contexto	Sara ha revisado los tratamientos de Jack y se ha dado cuenta que se le está terminando uno de los tratamientos, así que decide pedir una nueva receta para poder adquirirlo.
Objetivo	Obtener una receta de un medicamento
Tareas	<ul style="list-style-type: none"> <li>Solicitar receta al veterinario</li> </ul>
Necesidades de información	El medicamento del que se solicita la receta, saber acceder a la solicitud de recetas
Funcionalidades	<ul style="list-style-type: none"> <li>Solicitar receta al veterinario</li> </ul>



Desarrollo de las tareas	<ol style="list-style-type: none"> <li>1. [Registro en la aplicación]</li> <li>2. Iniciar sesión</li> <li>3. Seleccionar el apartado “Mascotas”</li> <li>4. Seleccionar una mascota</li> <li>5. Seleccionar un tratamiento de la mascota</li> <li>6. Seleccionar la opción “Añadir receta”</li> <li>7. Elige “Solicitar receta a veterinario”</li> <li>8. Se envía la solicitud de receta al veterinario enlazado (se recibirá una notificación cuando el veterinario envíe la receta)</li> </ol>
--------------------------	---

#### EU5: Adjuntar receta

Perfil de usuario	Propietario de mascotas con enfermedades
Persona	Sara
Contexto	Sara se ha mudado de ciudad y el veterinario al que solía llevar a Jack le queda demasiado lejos con lo que, busca uno más cerca de su casa. Habla con uno que le han recomendado y le gusta como veterinario, así que decide llevar allí a Jack a pesar de que no quiere usar la aplicación. Tras explicarle la situación, el veterinario sigue sin querer usar la aplicación, pero accede a enviarle las recetas por email previa llamada para solicitarlas.
Objetivo	Obtener una receta de un medicamento
Tareas	<ul style="list-style-type: none"> <li>• Adjuntar receta recibida por otros medios</li> </ul>
Necesidades de información	El medicamento del que se solicita la receta, tener la receta, saber añadir la receta.
Funcionalidades	<ul style="list-style-type: none"> <li>• Crear nueva receta adjuntándola</li> </ul>
Desarrollo de las tareas	<ol style="list-style-type: none"> <li>1. [Registro en la aplicación]</li> <li>2. Iniciar sesión</li> <li>3. Seleccionar el apartado “Mascotas”</li> <li>4. Seleccionar una mascota</li> <li>5. Seleccionar un tratamiento de la mascota</li> <li>6. Selecciona añadir una receta</li> <li>7. Rellena los datos de la receta y adjunta una imagen de esta</li> </ol>

#### EU6: Enviar receta a farmacia

Perfil de usuario	Propietario de mascotas con enfermedades
Persona	Sara
Contexto	Sara ha hablado con su farmacia para preguntarles si les puede enviar por email las recetas veterinarias de Jack para no tener que ir varias veces a la farmacia. En la farmacia acceder, le dan



	su email para que pueda enviarles la receta y ellos la avisarán en cuanto.
Objetivo	Obtener una receta de un medicamento
Tareas	<ul style="list-style-type: none"> <li>• Enviar receta a la farmacia</li> </ul>
Necesidades de información	El medicamento del que se solicita la receta, saber acceder a la solicitud de recetas, tener los datos de la farmacia guardados en la aplicación.
Funcionalidades	<ul style="list-style-type: none"> <li>• Enviar receta a la farmacia</li> </ul>
Desarrollo de las tareas	<ol style="list-style-type: none"> <li>1. [Registro en la aplicación]</li> <li>2. Iniciar sesión</li> <li>3. Seleccionar el apartado "Mascotas"</li> <li>4. Seleccionar una mascota</li> <li>5. Seleccionar un tratamiento de la mascota</li> <li>6. Seleccionar la receta activa de ese tratamiento</li> <li>7. Enviar la receta a la farmacia (es necesario que se hayan guardado los datos de la farmacia).</li> </ol>

#### EU7: Consultar recetas

Perfil de usuario	Propietario de mascotas con enfermedades
Persona	Sara
Contexto	Sara tiene un rato libre y decide consultar las recetas para ver el estado en que se encuentran
Objetivo	Consultar el estado de las recetas
Tareas	<ul style="list-style-type: none"> <li>• Consultar recetas</li> </ul>
Necesidades de información	Saber acceder a la consulta de recetas
Funcionalidades	<ul style="list-style-type: none"> <li>• Consultar recetas por distintos filtros</li> </ul>
Desarrollo de las tareas	<ol style="list-style-type: none"> <li>1. [Registro en la aplicación]</li> <li>2. Iniciar sesión</li> <li>3. Seleccionar el apartado "Recetas"</li> <li>4. Seleccionar los filtros para la búsqueda</li> <li>5. Consultar los resultados de la búsqueda</li> </ol>

#### EU8: Consultar notificaciones

Perfil de usuario	Propietario de mascotas con enfermedades
Persona	Sara
Contexto	Sara está dando un paseo con sus perros, se para un rato a descansar y decide consultar la aplicación para ver si tiene notificaciones
Objetivo	Ver si tiene notificaciones importantes
Tareas	- Acceder a la aplicación



	- Consultar las notificaciones
Necesidades de información	Saber iniciar sesión en la aplicación y entender las notificaciones. Datos de la receta.
Funcionalidades	<ul style="list-style-type: none"> <li>Mostrar avisos al usuario</li> </ul>
Desarrollo de las tareas	<ol style="list-style-type: none"> <li>[Registro en la aplicación]</li> <li>Iniciar sesión</li> <li>Pulsar sobre el aviso de notificaciones</li> <li>Consultar las notificaciones</li> </ol>

#### EU9: Enlazar con propietario

Perfil de usuario	Veterinario
Persona	Luis
Contexto	Luis está en consulta y el propietario le habla de la aplicación, le dice que le resultaría muy cómodo si pudiese solicitarle así las recetas. Tras hablar un rato, Luis accede e instala la aplicación para que el propietario pueda conectarse con él.
Objetivo	Enlazarse con propietario
Tareas	<ul style="list-style-type: none"> <li>Acceder a la aplicación</li> <li>Consultar las notificaciones</li> <li>Aceptar solicitud de enlace de propietario</li> </ul>
Necesidades de información	Saber iniciar sesión en la aplicación y entender las notificaciones.
Funcionalidades	<ul style="list-style-type: none"> <li>Mostrar notificaciones al usuario</li> <li>Enlazarse con propietario</li> </ul>
Desarrollo de las tareas	<ol style="list-style-type: none"> <li>[Registro en la aplicación]</li> <li>Iniciar sesión</li> <li>Pulsar sobre el aviso de notificaciones</li> <li>Consultar las notificaciones</li> <li>Selecciona la petición de enlace del propietario</li> <li>Acepta la petición de enlace del propietario</li> </ol>

#### EU10: Hacer receta solicitada

Perfil de usuario	Veterinario
Persona	Luis
Contexto	Luis ha terminado una consulta y faltan aún 15 minutos para la siguiente así que mientras, consulta la aplicación por si ha recibido alguna petición de receta y aprovechar este rato para hacerla.
Objetivo	Hacer receta
Tareas	<ul style="list-style-type: none"> <li>Acceder a la aplicación</li> <li>Consultar las notificaciones</li> </ul>



	- Hacer receta
Necesidades de información	Saber iniciar sesión en la aplicación y entender las notificaciones.
Funcionalidades	<ul style="list-style-type: none"> <li>Mostrar notificaciones al usuario</li> <li>Hacer receta solicitada por propietario</li> </ul>
Desarrollo de las tareas	7. [Registro en la aplicación] 8. Iniciar sesión 9. Pulsar sobre el aviso de notificaciones 10. Consultar las notificaciones 11. Si tiene solicitud de receta, hace la receta

(\*) Las funcionalidades del veterinario en esta versión cumplen sólo la función de asistir al propietario para el envío de recetas. Hay funcionalidades que sería interesante incluir, pero se escapan del alcance de este trabajo. Una forma de añadir funcionalidades sin que supusiese trabajo extra para el veterinario sería comunicar la aplicación con su software de gestión de la clínica.

### 2.2.3. Flujos de interacción

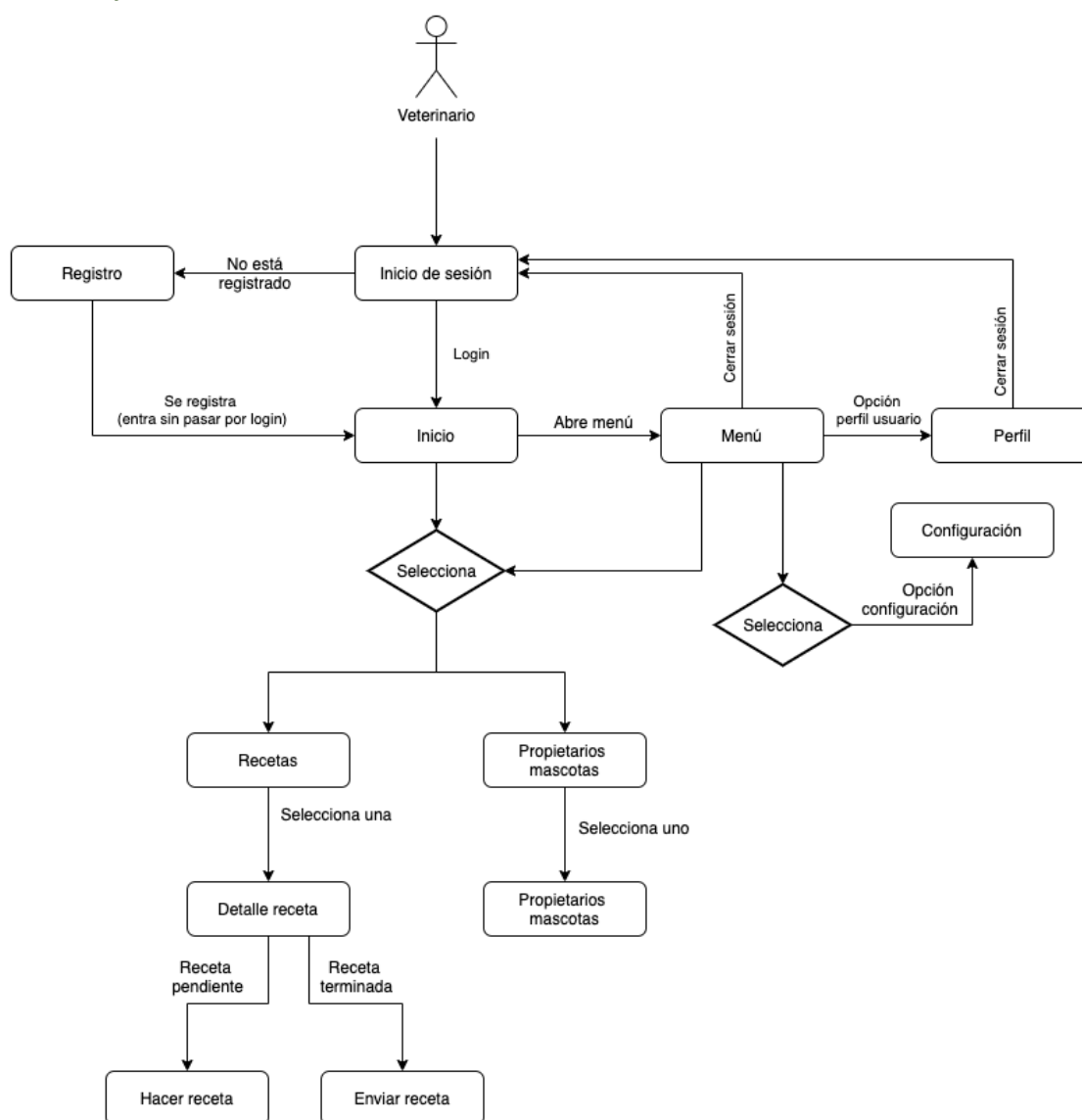


Ilustración 4: Flujos de interacción del veterinario



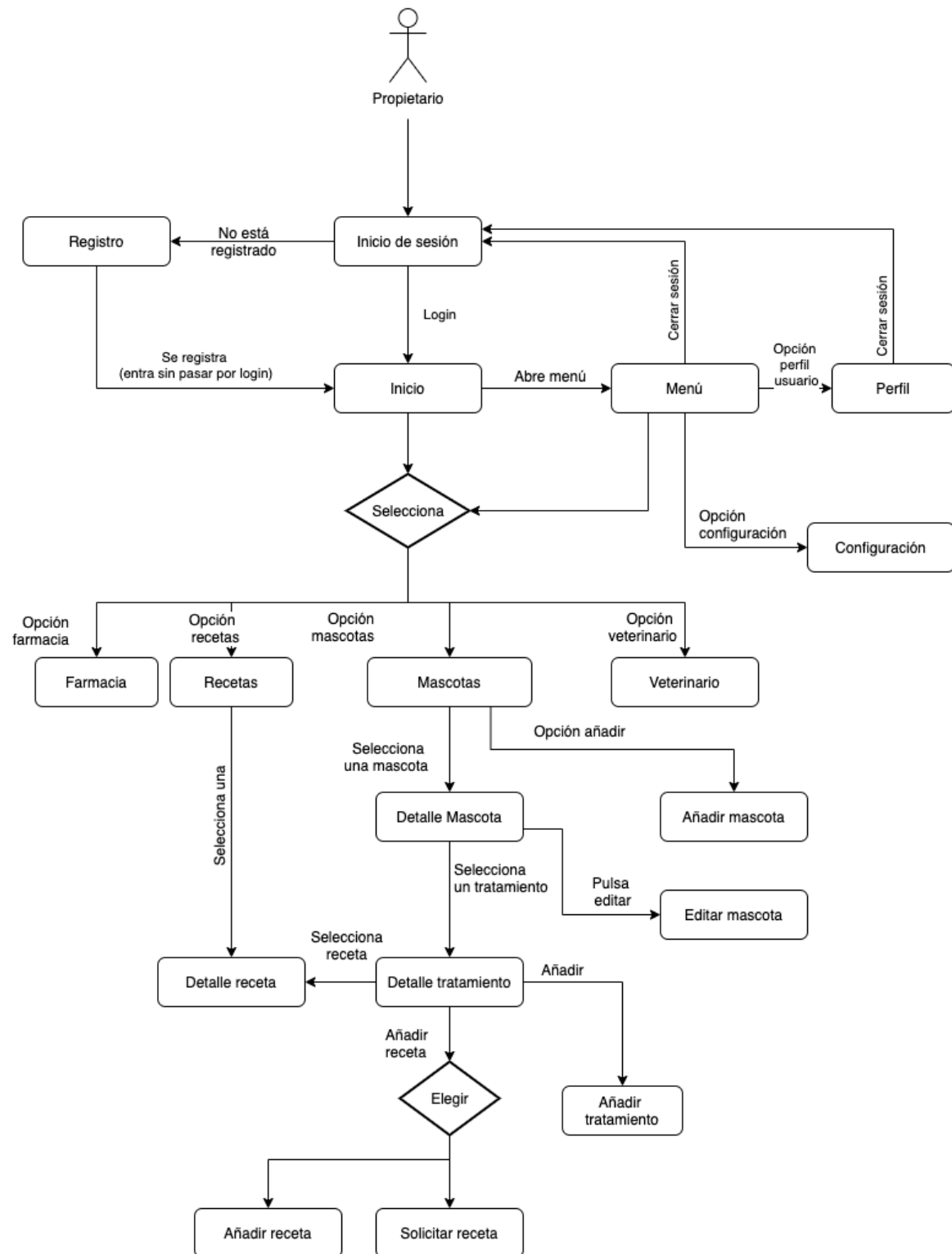


Ilustración 5: Flujos de interacción del propietario de mascotas

### 2.3. Prototipado

En esta tercera fase se realiza el prototipo a partir de los flujos de interacción, primero de forma esquemática a mano alzada y, a partir de estos esquemas se

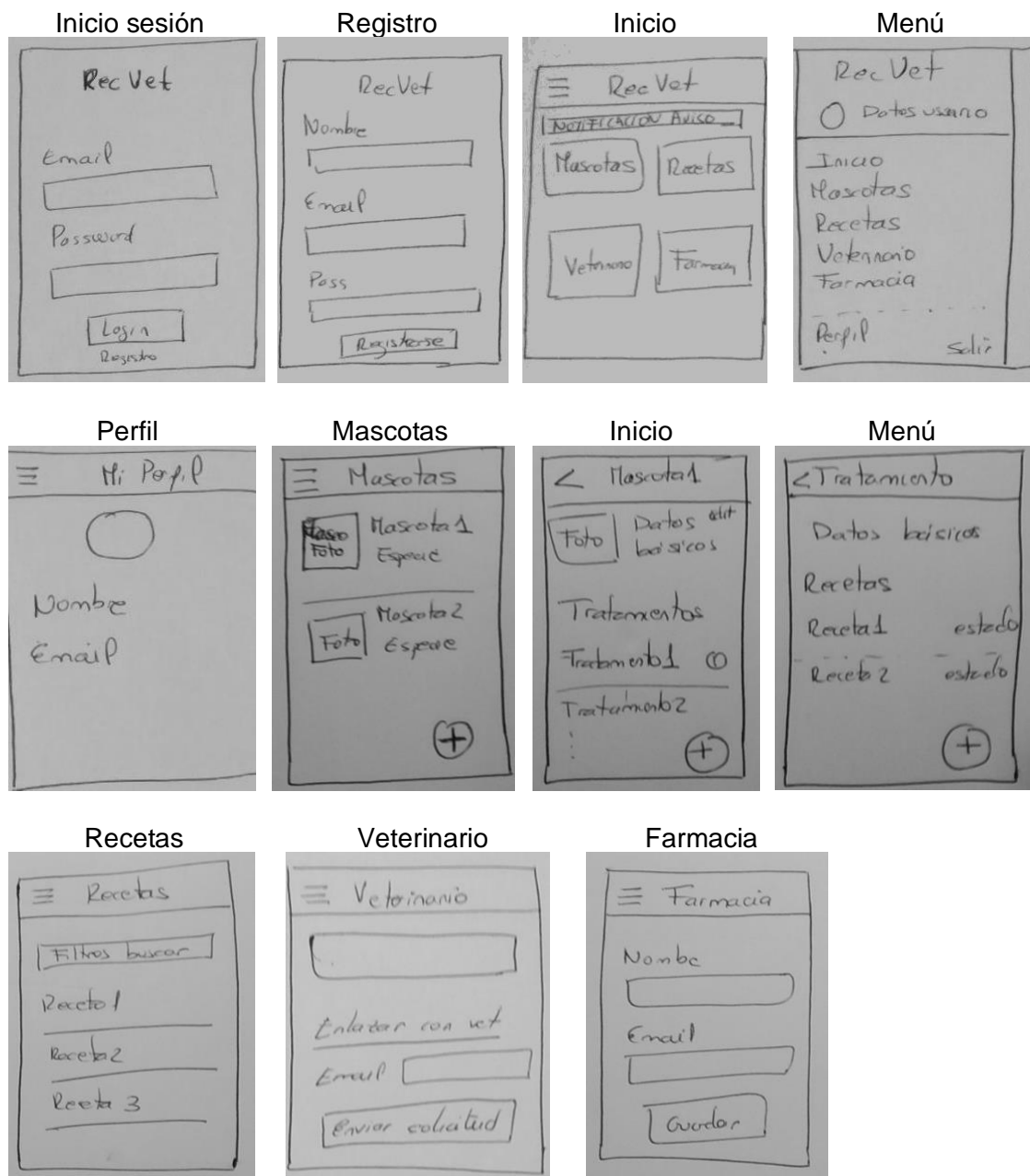


crear un prototipo horizontal de alta fidelidad que nos permitirá evaluar los modelos diseñados.

### 2.3.1. Sketches escaneados

En este apartado se incluyen el prototipo de baja fidelidad realizado a mano alzada que representa el diseño de la aplicación de forma esquemática.

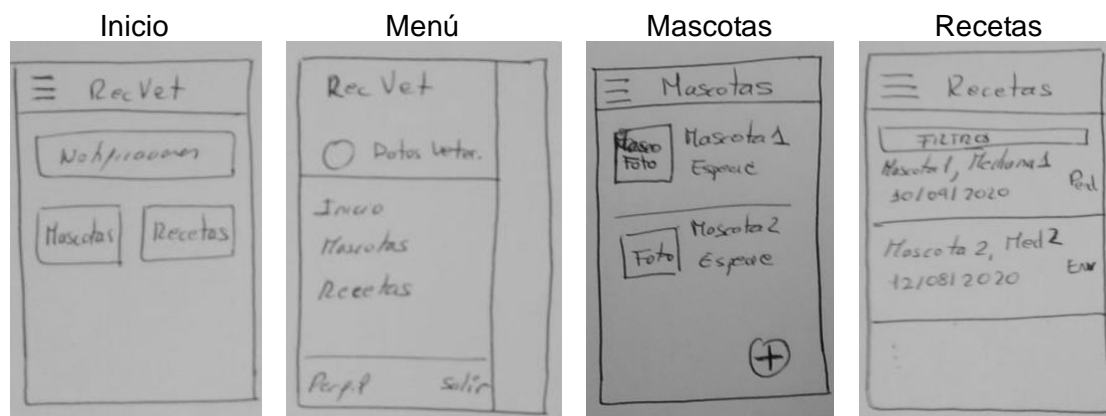
#### Pantallas propietario



#### Pantallas veterinario

A continuación, se muestran los sketches de las pantallas que varían para el veterinario:





### 2.3.2. Prototipo horizontal de alta fidelidad

En las siguientes imágenes se muestran el diseño detallado de las pantallas de la aplicación, comenzando con las pantallas del perfil del propietario y, a continuación, las del perfil del veterinario (se adjuntan con este documento dos HTML con una versión interactiva de este prototipo, una para el perfil de propietario de mascotas y otra para el perfil de veterinario).

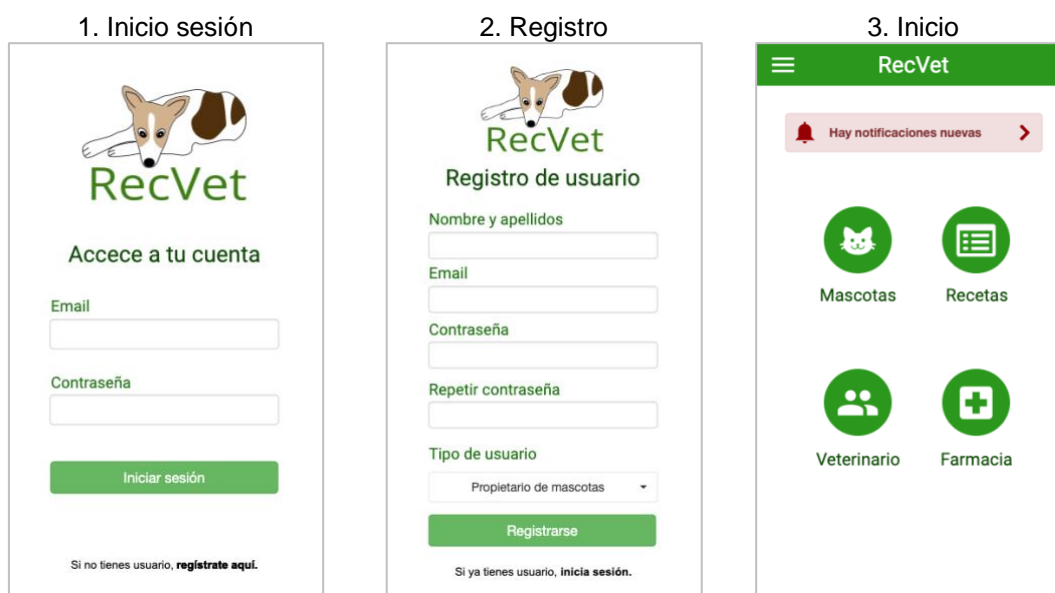
Breve descripción de las pantallas del perfil de propietario de mascotas:

1. **Inicio sesión:** pantalla inicial a través de la cual el usuario introduce sus credenciales para acceder a la aplicación.
2. **Registro:** A través de esta pantalla, un usuario puede registrarse en la aplicación. Desde la pantalla de inicio de sesión se puede acceder a esta pantalla para que, las personas que todavía no son usuarios de la aplicación puedan acceder a ella y darse de alta.
3. **Inicio:** Pantalla principal para el propietario de mascotas. Desde ella tiene acceso a las principales funcionalidades de la aplicación. En esta pantalla se muestra también un aviso si hay notificaciones nuevas sin leer (recetas a punto de caducar, recetas recibidas del veterinario, ...)
4. **Menú:** Menú de la aplicación, al que se puede acceder desde todas las pantallas principales. Desde el menú se da acceso, tanto a las principales funcionalidades de la aplicación, así como a notificaciones, perfil del usuario. También se puede cerrar la sesión.
5. **Perfil:** Datos de la cuenta de usuario
6. **Configuración:** Configuración de las notificaciones, permite activarlas y desactivarlas.
7. **Mascotas:** Pantalla con la lista de mascotas dadas de alta y desde la que se pueden dar de alta nuevas mascotas.
8. **Añadir una mascota:** Pantalla con el formulario para añadir una nueva mascota en la aplicación.
9. **Detalle de mascota:** Pantalla de consulta de los datos de una mascota y lista de sus tratamientos. Desde esta pantalla se pueden dar de alta nuevos tratamientos.



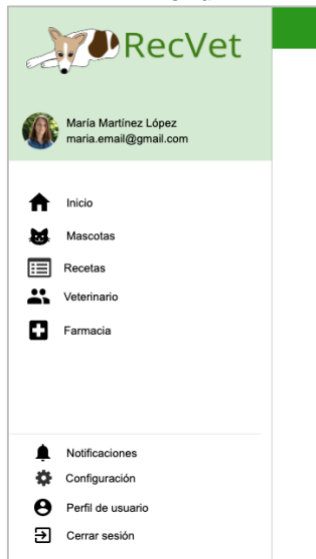
10. **Editar mascota:** Pantalla desde la que se pueden modificar los datos básicos de la mascota.
11. **Añadir tratamiento:** Pantalla con el formulario para añadir un nuevo tratamiento. Los campos serán de selección en los casos que sea posible porque se puede crear una lista de selección de los valores, dejando la introducción de texto sólo para aquellos que sea imprescindible como el nombre del medicamento, dosis y comentarios.
12. **Detalle de tratamiento:** Pantalla con el detalle de un tratamiento y las recetas que tiene asociadas. Desde esta pantalla se pueden añadir recetas.
13. **Editar tratamiento:** Desde esta pantalla se pueden editar los datos de un tratamiento.
14. **Nueva receta:** Al pulsar nueva receta desde la pantalla de un tratamiento, se elige si se va a solicitar la receta a un veterinario o si se adjuntará una receta recibida desde fuera de la aplicación. Si se selecciona la opción de solicitarla al veterinario, la app enviaría la solicitud al veterinario.
15. **Añadir receta:** Pantalla con el formulario para añadir una nueva receta que se ha recibido por otros medios.
16. **Veterinario:** Desde esta pantalla se puede enviar una solicitud de enlace a un veterinario o borrar el enlace con el veterinario que estaba enlazado.
17. **Farmacia:** Pantalla para introducir los datos de la farmacia.
18. **Recetas:** Desde esta pantalla se podrán consultar las recetas por varios filtros.
19. **Detalle receta:** Pantalla con la información detallada de una receta.
20. **Notificaciones:** Pantalla con la lista de notificaciones recibidas.

Diseño de las pantallas del perfil propietario de mascota:

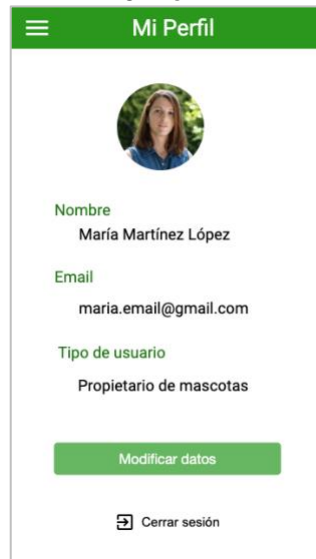




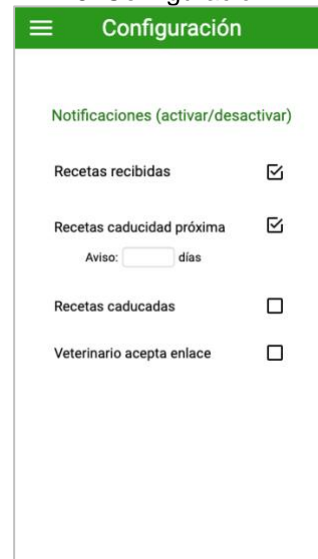
4. Menú



5. Perfil



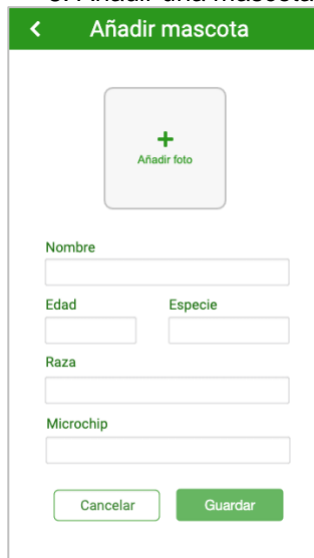
6. Configuración



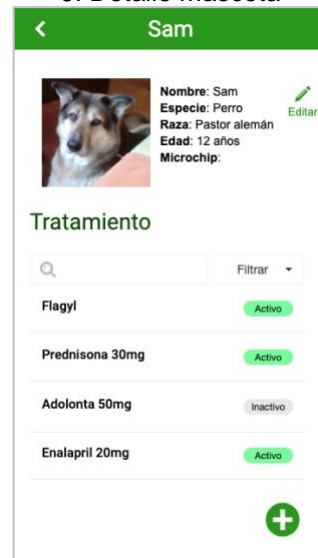
7. Mascotas



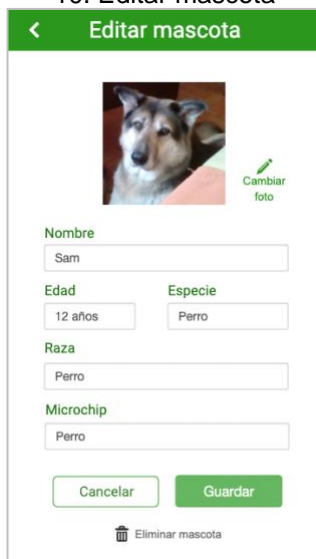
8. Añadir una mascota



9. Detalle mascota



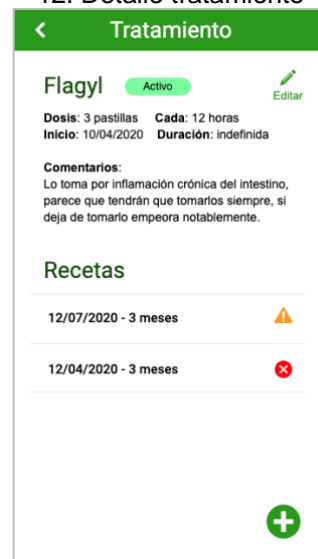
10. Editar mascota



11. Añadir tratamiento



12. Detalle tratamiento





13. Editar tratamiento

**< Editar tratamiento**

Nombre medicamento  
Flagyl

Dosis: 3 pastillas Cada: 12 horas  
Inicio: 10/06/2020 Duración: Indefinida

Comentarios

☒ Tratamiento activo

Cancelar Guardar

Eliminar tratamiento

14. Nueva receta

**< Tratamiento**

Flagyl

Dosis: 3 pastillas Cada: 12 horas  
Inicio: 10/04/2020 Duración: Indefinida

Comentarios:  
Lo toma por inflamación crónica del intestino, parece que tendrán que tomarlos siempre, si deja de tomarlo empeora notablemente.

Añadir receta

Seleccione cómo quiere añadir la receta:

Solicitar receta a veterinario

Adjuntar una receta

15. Añadir receta

**< Añadir receta**

Flagyl

Dosis: 3 pastillas Cada: 12 horas  
Inicio: 10/04/2020 Duración: Indefinida

Comentarios:  
Lo toma por inflamación crónica del intestino, parece que tendrán que tomarlos siempre, si deja de tomarlo empeora notablemente.

Receta

Fecha: 25/10/2020 Duración: 3 meses

Cancelar Guardar

16. Enlace veterinario

**≡ Veterinario**

No tienes enlace con el veterinario

Introduce el email del veterinario para enviarle la solicitud de enlace

Email del veterinario  
info@miveterinario.com

Enviar solicitud de enlace

17. Datos farmacia

**≡ Farmacia**

Nombre farmacia

Email

Cancelar Guardar

18. Recetas

**≡ Recetas**

Consultar recetas

Filtrar

Resultados

Sam, Flagyl, 12/07/2020

Sam: Prednisona 30mg, 12/05/2020

19. Detalle receta

**< Receta**

Detalles receta

Mascota: Sam  
Tratamiento: Flagyl

Fecha: 25/10/2020 Duración: 3 meses

Recibida de: Clínica MyPets  
Fecha: 25/10/2020

Enviada a: Farmacia López  
Fecha: 26/10/2020

Imagen receta

20. Notificaciones

**≡ Notificaciones**

Consultar notificaciones

Filtrar

15/10/2020 Receta recibida

01/10/2020 Receta caducada

25/09/2020 Receta caducidad próxima

Breve descripción de las pantallas del perfil de veterinario:



1. **Inicio sesión:** pantalla inicial a través de la cual el usuario introduce sus credenciales para acceder a la aplicación.
2. **Registro:** A través de esta pantalla, un usuario puede registrarse en la aplicación. Desde a pantalla de inicio de sesión se puede acceder a esta pantalla para que, las personas que todavía no son usuarios de la aplicación puedan acceder a ella y darse de alta.
3. **Inicio:** Pantalla principal para el veterinario. Desde ella tiene acceso a las principales funcionalidades de la aplicación. En esta pantalla se muestra también un aviso si hay notificaciones nuevas sin leer (solicitud de enlace recibido, solicitud de recetas recibida, ...)
4. **Menú:** Menú de la aplicación, al que se puede acceder desde todas las pantallas principales. Desde el menú se da acceso, tanto a las principales funcionalidades de la aplicación, así como a notificaciones, perfil del usuario. También se puede cerrar la sesión.
5. **Perfil:** Datos de la cuenta de usuario
6. **Propietarios:** Pantalla con la lista de propietarios de las mascotas.
7. **Mascotas:** Pantalla con la lista de mascotas de un propietario.
8. **Recetas:** Desde esta pantalla se podrán consultar las recetas por varios filtros.
9. **Hacer receta:** Desde esta pantalla se hacen las recetas que solicita el propietario.
10. **Detalle receta:** Pantalla con la información detallada de una receta. Desde aquí se puede enviar al propietario.
11. **Notificaciones:** Pantalla con la lista de notificaciones recibidas.

Diseño de las pantallas del perfil propietario de mascota:

1. Inicio sesión

2. Registro

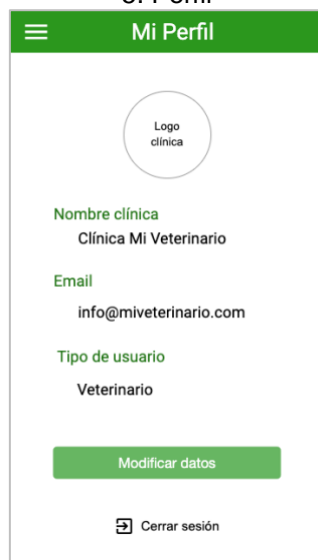
3. Inicio



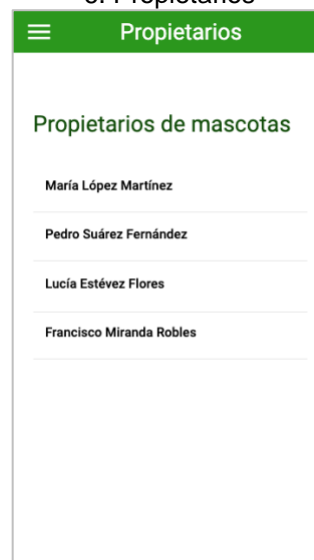
4. Menú



5. Perfil



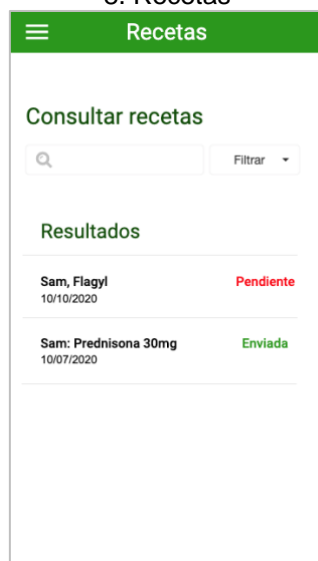
6. Propietarios



7. Mascotas



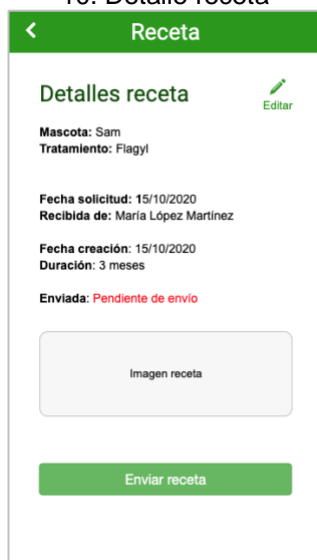
8. Recetas



9. Hacer receta



10. Detalle receta



11. Notificaciones





### 2.4. Evaluación

---

Una vez realizado el diseño de la aplicación, el siguiente paso es evaluar el diseño y para ello se realizará un test de usuarios, en el cual, los usuarios, realizarán una serie de tareas establecidas utilizando el prototipo interactivo de alta fidelidad mientras son observados para evaluar cómo utilizar la aplicación y detectar así formas de mejorar el diseño.

Por tanto, el objetivo general de este test con usuarios es evaluar la experiencia del usuario con el prototipo de la aplicación. Como se trata de un prototipo, aunque interactivo, realizado con una herramienta de diseño, no tiene funcionalidad real y, por tanto, debemos ser conscientes que no se podrán analizar en detalle la totalidad de las funciones.

De todas formas, dado que se puede navegar entre las distintas pantallas y ver contenido en ellas, podemos plantear los siguientes objetivos parciales comprobar si:

- Ver si la navegación es eficaz y eficiente: queremos saber si los usuarios encuentran lo que buscan fácilmente o por lo contrario les resulta difícil.
- El contenido que presentan las pantallas es útil y práctico: queremos saber si la información que se presenta en la aplicación está organizada de forma clara y permite que los usuarios la encuentren sin dificultad. Además, queremos saber si la información que se muestra es la que quiere el usuario.
- La claridad en la presentación de los contenidos: queremos saber si el diseño de la aplicación ayuda al usuario y hace más sencillo el uso de la aplicación o por lo contrario lo distrae.

Además, se plantea otro objetivo: “Conseguir que los usuarios realicen las tareas que les proponemos”. A través de las tareas realizadas obtendremos información importante sobre las impresiones de los usuarios sobre el diseño que hemos realizado que nos ayudarán a mejorarlo.

#### 2.4.1. Perfil sociodemográfico

Antes de comenzar el test de usuarios, debemos establecer un perfil sociodemográfico de los usuarios a los que haremos el test. Definiremos este perfil a partir de las características extraídas de los métodos de indagación de usuarios.

El perfil que definimos para los usuarios a los que haremos el test, se define a continuación. En este caso, definiremos un perfil para los propietarios de mascotas y otro para los veterinarios, ya que los tests serán distintos.

Perfil sociodemográfico para los propietarios de mascotas:

- Personas de cualquier género entre 18 y 55 años
- Tienen una o varias mascotas y al menos una de ellas tiene alguna patología



- Suelen utilizar aplicaciones móviles, al menos las básicas de mensajería instantánea, es decir, usan los dispositivos móviles con cierta facilidad.

Perfil sociodemográfico para los veterinarios:

- Personas de cualquier género de 30 años o más
- Son veterinarios de mascotas
- Trabajan por cuenta propia, es decir, tienen una clínica veterinaria o atienden a domicilio, pero no son empleados de otro veterinario.
- Suelen utilizar aplicaciones móviles, al menos las básicas de mensajería instantánea, es decir, usan los dispositivos móviles con cierta facilidad.

### 2.4.2. Documento de screening

Se trata de un cuestionario que nos permitirá seleccionar a los usuarios que encajan en el perfil que hemos definido.

Documento de screening para los propietarios:

#### **PREGUNTAS GENERALES**

1. ¿Es usted hombre o mujer?
2. Lugar de residencia
3. ¿Cual de los siguientes rangos de edades describe mejor su edad?
  - a) De 18 a 24
  - b) De 25 a 44
  - c) De 45 a 55
  - d) Más de 55

#### **MASCOTAS**

1. ¿Tiene mascotas?
2. Si tiene mascotas, ¿cuántas tiene y de que especie son?
3. Si tiene mascotas, ¿alguna de ellas tiene alguna enfermedad?

#### **USO DEL MÓVIL**

1. ¿Tiene un smartphone?
2. Para qué suele utilizarlo (puede seleccionar varias opciones):
  - a) Para llamar
  - b) Mensajes instantáneos (whatsapp, ...)
  - c) Email
  - d) Todo tipo de aplicaciones

Documento de screening para los veterinarios:

#### **PREGUNTAS GENERALES**

1. ¿Es usted hombre o mujer?
2. Lugar de residencia
3. ¿Cual de los siguientes rangos de edades describe mejor su edad?
  - a) De 18 a 29
  - b) De 30 a 44
  - c) De 45 a 55
  - d) Más de 55

#### **SITUACIÓN PROFESIONAL**

1. ¿Es veterinario de mascotas?
2. Elija la opción que mejor encaje en su situación actual:



- a) Trabajo por cuenta ajena
- b) Tengo mi propia clínica

### USO DEL MÓVIL

1. ¿Tiene un smartphone?
2. Para qué suele utilizarlo (puede seleccionar varias opciones):
  - a) Para llamar
  - b) Mensajes instantáneos (whatsapp, ...)
  - c) Email
  - d) Todo tipo de aplicaciones

### 2.4.3. Cuestionarios pre-test

Una vez seleccionados los usuarios que encajan en el perfil a través de los documentos de screening, se realiza el test a los usuarios seleccionados.

Antes de realizar las tareas propias de la evaluación de la aplicación, se realiza un cuestionario que nos permite conocer mejor al usuario y que nos permitirá analizar mejor los resultados. A continuación, los cuestionarios pre-test para los propietarios y para los veterinarios.

#### Cuestionario pre-test para los propietarios de mascotas:

1. Género:
2. Lugar de residencia:
3. Edad:
4. Profesión:
4. ¿Cuántas mascotas tiene?
5. ¿Cuáles son sus nombres y de que especie es cada una?
3. ¿Cuántas horas al día suele utilizar el móvil?
4. ¿Qué tipo de aplicaciones suele utilizar?
5. ¿Ha utilizado alguna vez alguna aplicación relacionada con la gestión de la salud de sus mascotas?
6. ¿Ha utilizado alguna vez alguna aplicación para gestionar su salud?

#### Cuestionario pre-test para los veterinarios:

- Género:
- Lugar de residencia:
- Edad:
- ¿Suele tener clientes con varias mascotas?
- ¿Qué porcentaje de las mascotas que trata tienen alguna enfermedad crónica?
- ¿Trata a muchas mascotas ancianas?
- ¿Suele recetar muchas medicinas de farmacia?
- ¿Suele utilizar aplicaciones de temática veterinaria en el móvil?

### 2.4.4. Escenarios y tareas

En este apartado se describen los escenarios y las tareas a realizar durante el test. Las tareas no abarcan todas las funcionalidades de la aplicación, pero abarcan las principales.



### 2.4.4.1 Escenario para los propietarios

Tu pareja y tu habéis pensado en tener un perro y después de pensarlo con calma, estáis seguros de que queréis adoptar un perro en una protectora, así que, llamáis a una protectora que está cerca de la zona donde vivís y os dicen que acudáis al refugio a conocer a los perros.

Cuando llegáis allí, un perro muy gracioso se acerca a saludaros y no podéis evitar enamoraros de él así que, le decís a la chica que realiza las adopciones que os gustaría adoptarlo. Ella se alegra mucho pero os avisa de que Pluto (así se llama el perro) tiene una enfermedad cardíaca crónica, que es leve pero que tiene que tomar una pastilla a diario probablemente el resto de su vida. No os importa y seguís adelante con la adopción.

Tareas a realizar:

- Tarea 1: Añade a Pluto como mascota en la aplicación.
- Tarea 2: Añadir el tratamiento de Pluto. Se trata de 1 pastilla de Enalapril 20mg cada 24 horas.

**Continuación escenario - supuesto 1:** Hablas con tu veterinario y accede a instalar la aplicación.

Tareas a realizar:

- Tarea 3: Enlaza tu usuario con el del veterinario. Su email es: [info@mypet.com](mailto:info@mypet.com)
- Tarea 4: Solicita una receta para Pluto de Enalapril 20mg a tu veterinario

**Continuación escenario - supuesto 2:** Tu veterinario no quiere usar la aplicación, pero accede a enviártelas por email y quieres tenerlas guardadas en la aplicación para tener un histórico y poder consultarlas posteriormente.

Tareas a realizar:

- Tarea 5: Añade la receta que te ha enviado tu veterinario a la aplicación. Vamos a suponer que has guardado ya el archivo con la receta y sólo tendrías que adjuntarlo. La fecha de la receta es el día de hoy y la duración de la receta son 3 meses.

### 2.4.4.2 Escenario para los veterinarios

Uno de tus mejores clientes te pide que instales la aplicación para poder pedirte las recetas a través de ella y que tu se las envíes así también. Acceder a hacerlo, instalas la aplicación y te registras. Al poco tiempo, tu cliente te envía una solicitud de enlace.

Tareas a realizar:

- Tarea 1: Consulta las notificaciones.
- Tarea 2: Acepta la solicitud de enlace que te ha enviado tu cliente.



**Continuación del escenario:** Unos días más tarde recibes una nueva notificación, tu cliente te solicita una receta.

Tareas a realizar:

- Tarea 3: Haz la receta y envíasela a tu cliente.

**Continuación del escenario:** Tu cliente ha hablado con otros y te han llegado varias solicitudes a la aplicación, las has aceptado todas.

Tareas a realizar:

- Tarea 4: Consulta la lista de clientes y mascotas que tienes en la aplicación.

#### 2.4.5. Cuestionarios post-test

Al finalizar el test, daremos este cuestionario a los usuarios para que valoren como ha sido su experiencia.

En este caso, el formulario puede ser común para el propietario de mascotas y para el veterinario ya que hay preguntas específicas de cada perfil. Aunque, sí habrá que analizar los resultados separando los post-test de ambos perfiles ya que, las funcionalidades de la aplicación son distintas y es muy posible que haya opiniones también muy distintas.

1. De forma general, ¿crees que es fácil utilizar la aplicación?	Poco	1	2	3	4	5	Mucho
2. Tu impresión general de la aplicación es	Pésima	1	2	3	4	5	Excelente
3. ¿Has sabido en qué pantalla estabas en todo momento?	Nunca	1	2	3	4	5	Siempre
4. ¿Consideras que la interfaz gráfica de la aplicación te ha ayudado a encontrar lo que buscabas?	Nunca	1	2	3	4	5	Siempre
5. ¿Cómo ha sido el desarrollo de las tareas?	Fácil	1	2	3	4	5	Imposible
6. ¿Cuál te ha resultado más difícil? Explica el motivo.							
7. ¿Qué es lo que más te ha gustado de la aplicación?							
8. ¿Qué es lo que menos te ha gustado de la aplicación?							
9. ¿Qué cambiarías? (Puede ser añadir/modificar/eliminar algo)							
10. ¿Qué te ha parecido el diseño gráfico?							
11. ¿Utilizarías la aplicación de forma habitual?							



### 3. Diseño técnico

Una vez terminada la fase de la DCU, se procede a realizar el diseño técnico de la aplicación. Los escenarios y prototipos diseñados servirán de punto de partida para la definición de los casos de uso de la aplicación. A continuación, se realizará el diseño de la arquitectura del sistema.

#### 3.1. Casos de uso

En este apartado se realiza la definición formal de los casos de uso que servirán para determinar las funcionalidades de la aplicación.

Antes de comenzar la definición de cada caso de uso, definimos los actores del sistema:

Identificador	ACT-001
Nombre	Propietario
Descripción	Este actor representa a los usuarios con el perfil “propietario de mascotas” que realizarán la interacción con la aplicación a través de un dispositivo móvil Android y accederán a los casos de uso de los propietarios de mascotas

Identificador	ACT-002
Nombre	Veterinario
Descripción	Este actor representa a los usuarios con el perfil “veterinario” que realizarán la interacción con la aplicación a través de un dispositivo móvil y accederán a los casos de uso de los

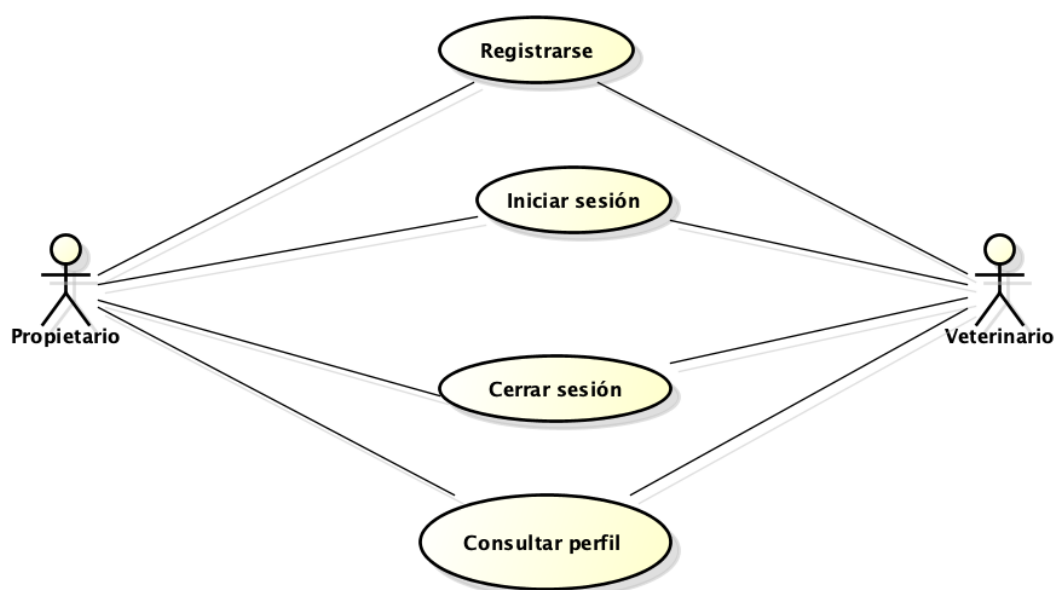


Ilustración 6: Casos de uso comunes



**Casos de uso comunes a ambos actores:**

Identificador	CU-001
Nombre	Registrarse
Prioridad	Normal
Descripción	Permite a un usuario registrarse en el sistema, crea una cuenta para poder acceder
Actores	Propietario, Veterinario
Precondiciones	El usuario no se ha registrado con ese email
Iniciado por	Usuario
Flujo	<p><b>Principal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación</li> <li>2. Accede a la pantalla de registro</li> <li>3. Introduce los datos de registro en el formulario y pulsa el botón de registro: email, contraseña, tipo de usuario (propietario o veterinario)</li> <li>4. El sistema verifica los datos, todo correcto</li> <li>5. El sistema crea la cuenta del usuario con los datos introducidos.</li> </ol> <p><b>Alternativos:</b></p> <ol style="list-style-type: none"> <li>4a. El sistema detecta que ya existe una cuenta de usuario con ese email y muestra un error al usuario.</li> <li>4b. El usuario no rellena todos los campos. El sistema muestra un mensaje de error al usuario.</li> </ol>
Postcondiciones	Al finalizar el proceso el usuario quedará registrado en el sistema
Notas	Al finalizar el registro el sistema inicia sesión con lo que el usuario accederá directamente a la aplicación si pasar por el inicio de sesión

Identificador	CU-002
Nombre	Iniciar sesión
Prioridad	Normal
Descripción	Permite a un usuario acceder a la aplicación
Actores	Propietario, Veterinario
Precondiciones	El usuario se ha registrado previamente con ese email
Iniciado por	Usuario
Flujo	<p><b>Principal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación</li> <li>2. Accede a la pantalla de Inicio de sesión</li> <li>3. Introduce el email y contraseña con los que se ha registrado</li> <li>4. Los datos introducidos son correctos y se accede a la aplicación.</li> </ol> <p><b>Alternativos:</b></p> <ol style="list-style-type: none"> <li>4a. No existe ningún usuario con ese email, se muestra un mensaje de error.</li> </ol>



	4b. El usuario existe con ese email pero la contraseña introducida es incorrecta. Se muestra un mensaje de error..
Postcondiciones	Al finalizar el proceso el usuario accede a la aplicación

Identificador	CU-003
Nombre	Consultar perfil de usuario
Prioridad	Baja
Descripción	Permite a un usuario consultar sus datos
Actores	Propietario, Veterinario
Precondiciones	El usuario ha iniciado sesión
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> <li>1. El usuario accede al menú</li> <li>2. Selecciona la opción "Perfil de usuario"</li> <li>3. El sistema muestra la pantalla con los datos del usuario</li> </ol>
Postcondiciones	El usuario visualiza los datos de su perfil

Identificador	CU-004
Nombre	Cerrar sesión
Prioridad	Normal
Descripción	Permite a un usuario cerrar la sesión de la aplicación
Actores	Propietario, Veterinario
Precondiciones	El usuario ha iniciado sesión
Iniciado por	Usuario
Flujo	<b>Principal:</b> <ol style="list-style-type: none"> <li>1. El usuario accede al menú</li> <li>2. Selecciona la opción "Cerrar sesión"</li> <li>3. El sistema cierra la sesión del usuario y muestra la pantalla de login</li> </ol>
Postcondiciones	El usuario ha salido de su sesión en la aplicación

### Casos de uso del actor "Propietario":

Identificador	CU-005
Nombre	Consultar lista de mascotas
Prioridad	Alta
Descripción	Permite a un usuario consultar la lista de mascotas que ha añadido en la aplicación
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión
Iniciado por	Usuario
Flujo	<b>Principal:</b> <ol style="list-style-type: none"> <li>1. El usuario accede al apartado "Mascotas" desde el menú o desde la pantalla principal.</li> <li>2. El sistema muestra la pantalla con la lista de mascotas o sin ninguna mascota si no e ha añadido ninguna todavía</li> </ol>
Postcondiciones	El usuario visualiza la pantalla de mascotas



Identificador	CU-006
Nombre	Consultar el detalle de una mascota
Prioridad	Alta
Descripción	Permite a un usuario consultar los datos de una mascota y la lista de sus tratamientos
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión y ha accedido a la pantalla con la lista de mascotas.
Iniciado por	Usuario
Flujo	<b>Principal:</b> <ol style="list-style-type: none"> <li>1. El usuario pulsa sobre una mascota de la lista de mascotas</li> <li>2. El sistema muestra la pantalla de esa mascota con su información básica y la lista de tratamientos.</li> </ol>
Postcondiciones	El usuario visualiza la pantalla de detalle de una mascota

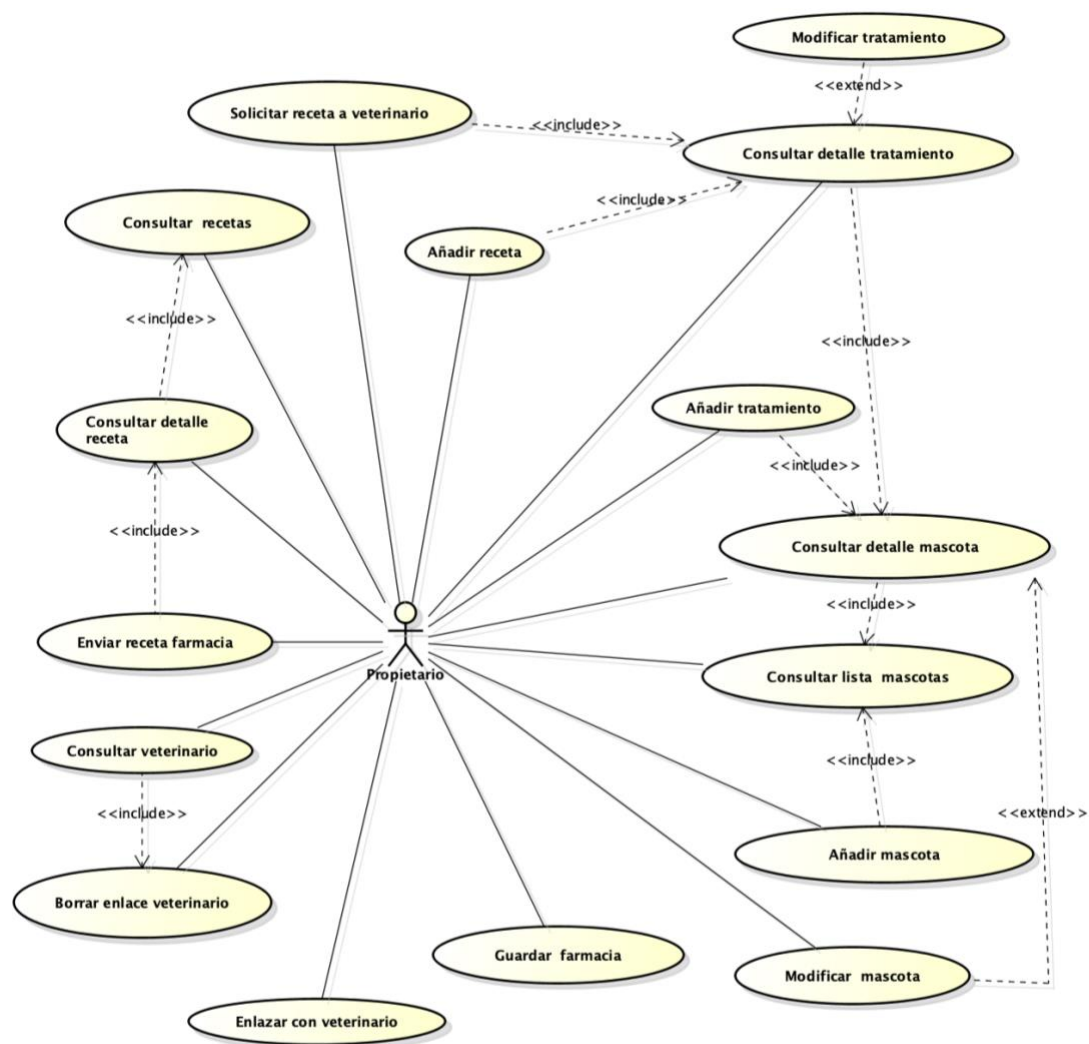


Ilustración 7: Casos de uso del propietario



Identificador	CU-007
Nombre	Añadir una mascota
Prioridad	Alta
Descripción	Permite a un usuario añadir una nueva mascota
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión y ha accedido a la pantalla con la lista de mascotas.
Iniciado por	Usuario
Flujo	<p><b>Principal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario pulsa sobre el icono de añadir mascota</li> <li>2. El sistema muestra el formulario para añadir una nueva mascota.</li> <li>3. El usuario introduce los datos</li> <li>4. El sistema comprueba los datos, son correctos.</li> <li>5. El sistema añade la nueva mascota.</li> </ol> <p><b>Alternativo:</b></p> <p>4b. El usuario no introduce todos los datos obligatorios, el sistema muestra un error</p>
Postcondiciones	Se añade una nueva mascota en el sistema

Identificador	CU-008
Nombre	Modificar los datos de una mascota
Prioridad	Alta
Descripción	Permite a un usuario editar los datos básicos una nueva mascota
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión y ha accedido a la pantalla de detalle de una mascota Existe al menos una mascota en el sistema
Iniciado por	Usuario
Flujo	<p><b>Principal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario pulsa sobre el icono de editar mascota</li> <li>2. El sistema muestra el formulario para modificar los datos de la mascota.</li> <li>3. El usuario introduce los datos.</li> <li>4. El sistema comprueba los datos, son correctos.</li> <li>5. El sistema modifica los datos de la mascota.</li> </ol> <p><b>Alternativo:</b></p> <p>4b. El usuario no introduce todos los datos obligatorios, el sistema muestra un error</p>
Postcondiciones	Los datos de la mascota se modifican

Identificador	CU-009
Nombre	Añadir tratamiento a una mascota
Prioridad	Alta
Descripción	Permite a un usuario añadir un nuevo tratamiento a una mascota



Actores	Propietario
Precondiciones	El usuario ha iniciado sesión y ha accedido a la pantalla de detalle de una mascota Existe al menos una mascota en el sistema.
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario pulsa sobre el icono de añadir tratamiento 2. El sistema muestra el formulario para añadir los datos de un nuevo tratamiento 3. El usuario introduce los datos. 4. El sistema comprueba los datos, son correctos. 5. El sistema añade el tratamiento <b>Alternativo:</b> 4b. El usuario no introduce todos los datos obligatorios, el sistema muestra un error
Postcondiciones	La mascota tiene un nuevo tratamiento

Identificador	CU-010
Nombre	Consultar detalles de un tratamiento
Prioridad	Alta
Descripción	Permite a un usuario consultar los detalles de un tratamiento incluyendo la lista de recetas asociadas
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión y ha accedido a la pantalla de detalle de una mascota Existe al menos una mascota con un tratamiento en el sistema
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario pulsa sobre uno de los tratamientos 2. El sistema muestra la pantalla con los detalles de ese tratamiento.
Postcondiciones	El usuario visualiza la pantalla de detalle de un tratamiento

Identificador	CU-011
Nombre	Modificar un tratamiento
Prioridad	Alta
Descripción	Permite a un usuario modificar los datos de un tratamiento
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión y ha accedido a la pantalla de detalle de un tratamiento. Existe al menos una mascota con un tratamiento en el sistema
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario pulsa sobre el icono de editar los datos del tratamiento 2. El sistema muestra la pantalla de edición del tratamiento



	<p>3. El sistema modifica los datos del tratamiento y pulsa aceptar.</p> <p>4. El sistema comprueba los datos, todo correcto.</p> <p>5. El sistema modifica los datos del tratamiento.</p> <p><b>Alternativa:</b></p> <p>4b. El sistema comprueba los datos, el usuario ha modificado el nombre del tratamiento. El sistema comprueba si hay recetas asociadas y, en ese caso, muestra un aviso al usuario indicando que, salvo para corrección de errores tipográficos, no debería cambiar el nombre del medicamento.</p>
Postcondiciones	El usuario visualiza la pantalla de detalle de un tratamiento
Notas	<p>Se permite la edición del nombre del medicamento para la corrección de errores tipográficos.</p> <p>Una solución a este problema sería que los nombres de los medicamentos se seleccionasen en lugar de introducirlos, pero eso implicaría tener una tabla de medicamentos o leerlos de una base de datos externa, algo que se escapa del alcance de este trabajo.</p>

Identificador	CU-012
Nombre	Solicitar una receta a un veterinario
Prioridad	Alta
Descripción	Permite a un usuario solicitar una receta para un tratamiento
Actores	Propietario
Precondiciones	<p>Existe al menos una mascota con un tratamiento</p> <p>El usuario ha iniciado sesión y ha accedido a la pantalla de detalle de un tratamiento.</p> <p>El usuario está enlazado con un propietario.</p>
Iniciado por	Usuario
Flujo	<p><b>Principal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario pulsa sobre el icono de añadir receta</li> <li>2. El usuario selecciona la opción de solicitar receta a un veterinario</li> <li>3. El sistema realiza la petición de la receta al veterinario.</li> </ol>
Postcondiciones	Se crea una solicitud de receta, que queda pendiente de recibir la receta creada por el veterinario.

Identificador	CU-013
Nombre	Añadir una receta
Prioridad	Alta
Descripción	Permite a un usuario añadir una receta recibida por otros medios
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión y ha accedido a la pantalla de detalle de un tratamiento.



	Existe al menos una mascota con un tratamiento en el sistema.
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario pulsa sobre el icono de añadir receta 2. El usuario selecciona la opción de solicitar receta a un veterinario 3. El sistema realiza la petición de la receta al veterinario.
Postcondiciones	Se crea una solicitud de receta, que queda pendiente de recibir la receta creada por el veterinario.

Identificador	CU-014
Nombre	Consultar los detalles de una receta
Prioridad	Alta
Descripción	Permite a un usuario consultar los detalles de una receta
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión y ha accedido a la pantalla de detalle de un tratamiento. Existe al menos una mascota con un tratamiento y una receta en el sistema.
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario pulsa sobre una receta del tratamiento 2. El sistema muestra la pantalla de detalles de la receta.
Postcondiciones	El usuario visualiza la pantalla de detalle de la receta.

Identificador	CU-015
Nombre	Enviar una receta a la farmacia
Prioridad	Alta
Descripción	Permite a un usuario enviar una receta a la farmacia
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión y ha accedido a la pantalla de detalle de un tratamiento. Existe al menos una mascota con un tratamiento y una receta en el sistema. Se han añadido los datos de la farmacia en el sistema.
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario pulsa sobre una receta del tratamiento 2. El sistema muestra la pantalla de detalles de la receta. 3. El usuario pulsa sobre el botón "Enviar receta a farmacia" 4. El sistema envía la receta por email a la farmacia y registra el envío en la base de datos.  <b>Alternativos:</b> 3a. El botón "Enviar receta a farmacia" no está activado porque la receta ya ha sido enviada, está caducada o está todavía pendiente de creación por el veterinario.



Postcondiciones	La receta se envía por email a la farmacia y queda registrado el envío en el sistema.
-----------------	---

Identificador	CU-016
Nombre	Consulta de recetas por filtros
Prioridad	Normal
Descripción	Permite a un usuario consultar recetas por distintos filtros
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión
Iniciado por	Usuario
Flujo	<b>Principal:</b> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “recetas” a través del menú o de la pantalla de inicio</li> <li>2. El sistema muestra la pantalla de consulta de recetas.</li> <li>3. El usuario introduce los filtros de búsqueda y pulsa buscar</li> <li>4. El sistema muestra las recetas que encajan en los filtros o ninguna si no encaja ninguna</li> </ol>
Postcondiciones	El usuario visualiza la pantalla de consulta de recetas.

Identificador	CU-017
Nombre	Consultar datos del veterinario
Prioridad	Alta
Descripción	Permite a un usuario consultar los datos de un veterinario
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión
Iniciado por	Usuario
Flujo	<b>Principal:</b> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “veterinario” a través del menú o de la pantalla de inicio</li> <li>2. El sistema muestra la pantalla de datos del veterinario.</li> </ol>
Postcondiciones	El usuario visualiza los datos del veterinario

Identificador	CU-018
Nombre	Enlazar con veterinario
Prioridad	Alta
Descripción	Permite a un usuario enviar una solicitud de enlace a un veterinario
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión
Iniciado por	Usuario
Flujo	<b>Principal:</b> <ol style="list-style-type: none"> <li>3. El usuario selecciona la opción “veterinario” a través del menú o de la pantalla de inicio</li> <li>4. El sistema muestra la pantalla de datos del veterinario.</li> <li>5. El usuario introduce el email del veterinario y pulsa el botón e enviar solicitud de enlace.</li> <li>6. El sistema envía la solicitud.</li> </ol>



	<b>Alternativo:</b> 4b. El email que introduce el propietario no existe como usuario veterinario del sistema. Se muestra un mensaje de error al propietario.
Postcondiciones	Queda registrada una solicitud de enlace con un veterinario, queda pendiente hasta que el veterinario al confirme.

Identificador	CU-019
Nombre	Borrar enlace con veterinario
Prioridad	Normal
Descripción	Permite a un usuario borrar el enlace con un veterinario
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión El usuario está enlazado con un veterinario
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario selecciona la opción “veterinario” a través del menú o de la pantalla de inicio 2. El sistema muestra la pantalla de datos del veterinario. 3. El usuario pulsa sobre “Borrar enlace”. 4. El enlace se borra.
Postcondiciones	El enlace con el veterinario queda eliminado

Identificador	CU-020
Nombre	Introducir datos de la farmacia
Prioridad	Normal
Descripción	Permite a un usuario guardar datos de una farmacia para poder enviarle las recetas
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario selecciona la opción “farmacia” a través del menú o de la pantalla de inicio 2. El sistema muestra el formulario para añadir la farmacia. 3. El usuario introduce los datos y pulsa guardar. 4. El sistema guarda los datos  <b>Alternativo:</b> 4b. Si el usuario no introduce los datos obligatorios, el sistema muestra un mensaje de error.
Postcondiciones	Los datos de la farmacia quedan guardados en el sistema.

Identificador	CU-021
Nombre	Modificar datos de la farmacia
Prioridad	Normal
Descripción	Permite a un usuario modificar los datos de la farmacia
Actores	Propietario
Precondiciones	El usuario ha iniciado sesión



Iniciado por	Usuario
Flujo	<b>Principal:</b> <ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “farmacia” a través del menú o de la pantalla de inicio</li> <li>2. El sistema muestra el formulario para añadir la farmacia.</li> <li>3. El usuario introduce los datos y pulsa guardar.</li> <li>4. El sistema guarda lo datos</li> </ol> <b>Alternativo:</b> <ol style="list-style-type: none"> <li>4b. Si el usuario no introduce los datos obligatorios, el sistema muestra un mensaje de error.</li> </ol>
Postcondiciones	Los datos de la farmacia quedan guardados en el sistema.

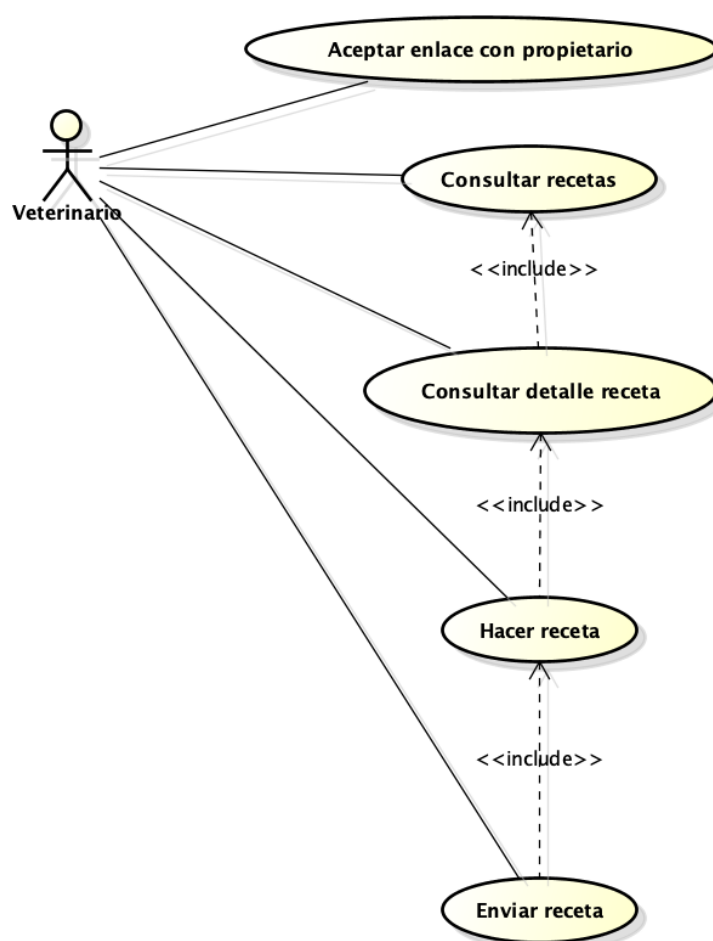


Ilustración 8: Casos de uso del veterinario

### Casos de uso del actor “Veterinario”:

Identificador	CU-021
Nombre	Aceptar enlace de propietario
Prioridad	Alta
Descripción	Permite a veterinario enlazarse con un propietario aceptando su solicitud de enlace



Actores	Veterinario
Precondiciones	El usuario ha iniciado sesión
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario recibe la notificación de solicitud de enlace 2. Pulsa sobre ella y acepta la solicitud <b>Alternativo:</b> 2b. El veterinario rechaza la solicitud de enlace
Postcondiciones	El veterinario y el propietario quedan enlazados

Identificador	CU-022
Nombre	Hacer receta solicitada
Prioridad	Alta
Descripción	Permite a veterinario hacer una receta que le ha solicitado un propietario
Actores	Veterinario
Precondiciones	El usuario ha iniciado sesión El propietario y el veterinario están enlazados El propietario ha enviado una solicitud de receta al veterinario
Iniciado por	Sistema
Flujo	<b>Principal:</b> 1. El usuario recibe la notificación de solicitud de receta 2. Pulsa sobre ella y pulsa sobre "Hacer receta" 3. Rellena los datos de la receta y la guarda <b>Alternativo:</b> 3b. El veterinario no introduce todos los datos y el sistema muestra un mensaje de error
Postcondiciones	La receta queda guardada

Identificador	CU-023
Nombre	Consulta de recetas
Prioridad	Normal
Descripción	Permite a veterinario consultar recetas por varios filtros
Actores	Veterinario
Precondiciones	El usuario ha iniciado sesión
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario accede a la opción "recetas" desde el menú o desde la página de inicio. 2. El sistema muestra la pantalla de recetas 3. El usuario selecciona los filtros y pulsa buscar 4. El sistema muestra la lista de recetas
Postcondiciones	El usuario consulta la lista de recetas

Identificador	CU-024
Nombre	Consultar detalle de una receta
Prioridad	Alta
Descripción	Permite a veterinario consultar recetas por varios filtros



Actores	Veterinario
Precondiciones	El usuario ha iniciado sesión El usuario ha consultado la lista de recetas
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario pulsa sobre la receta 2. El sistema muestra los detalles de la receta
Postcondiciones	El usuario visualiza los detalles de la receta

Identificador	CU-025
Nombre	Enviar receta a propietario
Prioridad	Alta
Descripción	Permite a veterinario una receta al propietario
Actores	Veterinario
Precondiciones	El usuario ha iniciado sesión El propietario y el veterinario están enlazados El propietario ha enviado una solicitud de receta al veterinario El veterinario ha creado la receta El veterinario consulta la receta
Iniciado por	Usuario
Flujo	<b>Principal:</b> 1. El usuario pulsa sobre el botón “Enviar receta” 2. El sistema envía la receta al propietario
Postcondiciones	La receta se envía al propietario

### 3.2. Modelo de datos

El modelo de datos del sistema está formado por las siguientes clases y atributos:

- **Usuario:** representa un usuario de la aplicación, puede ser de dos tipos: Propietario o Veterinario.
- **Mascota:** representa una mascota, los atributos son sus datos básicos. Para simplificar, los datos de especie y raza se definirán como enumerados definiendo un “Otros” para los casos no contemplados.
- **Tratamiento:** representa un tratamiento de una mascota, es decir, una medicación administrada con una dosis determinada durante un tiempo determinado. Los tipos de datos de los atributos “dosis”, “administración” y “duración” se han puesto como String para simplificar dado que, en esta versión, pero lo ideal sería definir los posibles valores de cada uno indicando unidades, etc.
- **Receta:** representa una receta de un tratamiento. Puede pasar por distintos estados: solicitada (al veterinario), creada, caducada, ....
- **RecetaSolicitadaVeterinario:** Representa una receta solicitada a un veterinario. Tiene como atributos las fechas de los distintos estados por los que pasa (solicitada por el propietario, creada por el veterinario, enviada al propietario). Se han establecido así las fechas por simplificar,



lo ideal sería que el estado tuviese entidad propia de forma que se podrían añadir más estados en el futuro si fuese necesario sin modificar el modelo de datos de la aplicación.

- **RecetaEnviadaFarmacia:** Representa una receta enviada a una farmacia, tiene como atributo la fecha de envío.
- **Farmacia:** Representa una farmacia, sólo puede haber una farmacia activa en cada momento (decisión tomada para simplificar la funcionalidad).

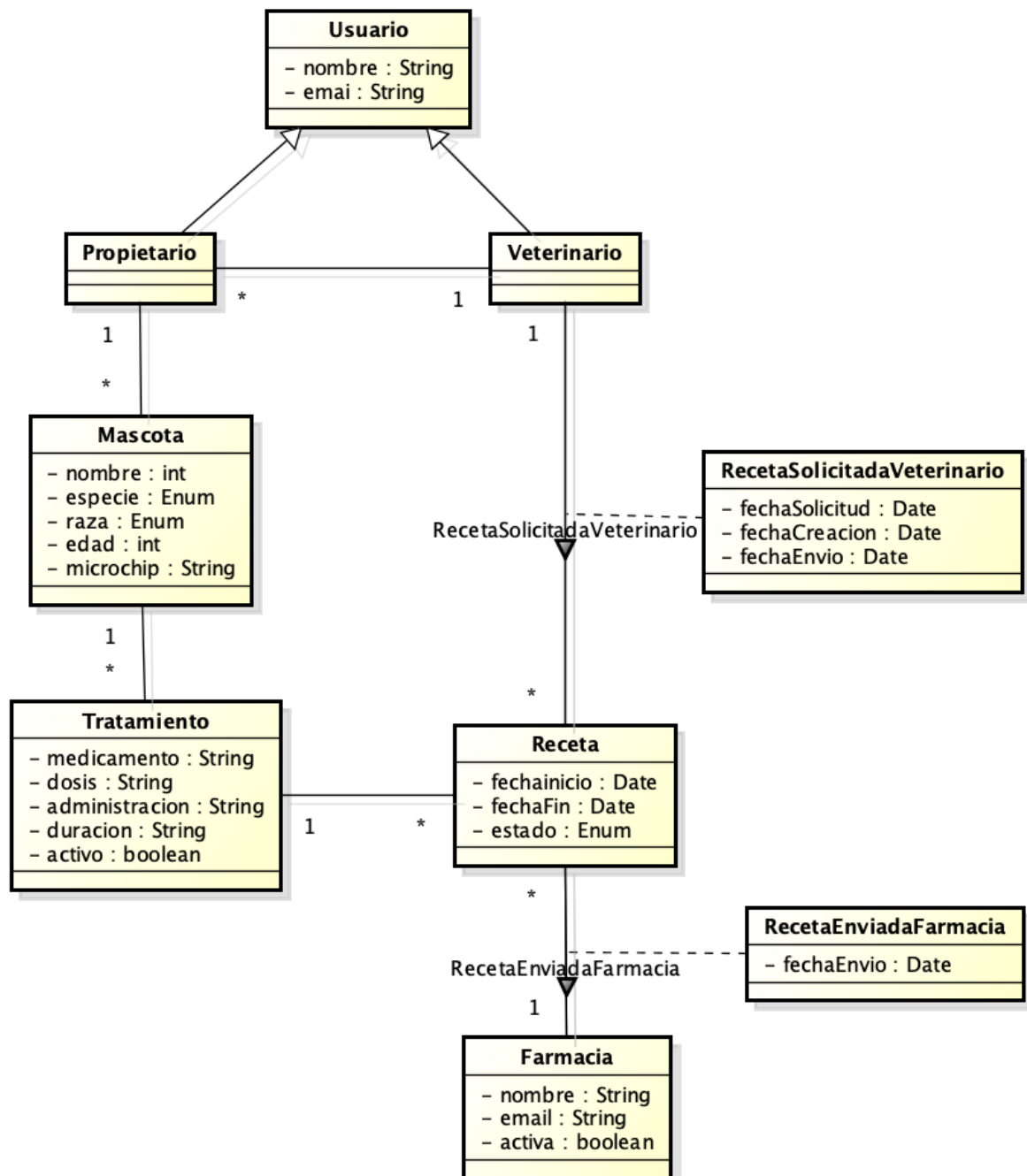


Ilustración 9: Diagrama de clases



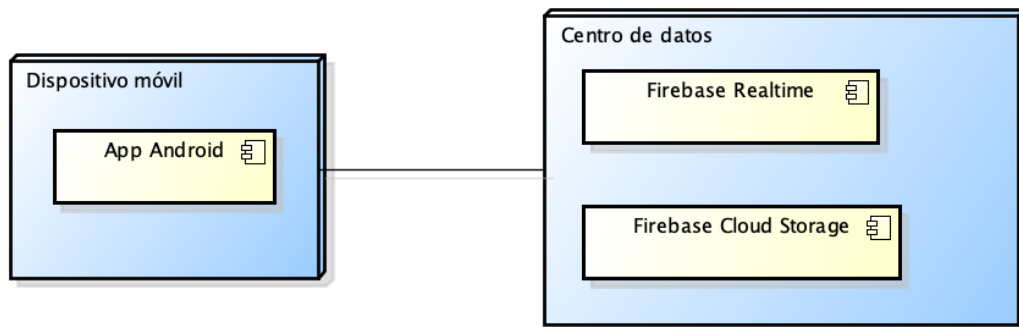
No se ha realizado un diagrama de base de datos ER, dado que la base de datos que se utilizará en la implementación es Firebase realtime, una base de datos NoSQL, que no sigue el modelo entidad-relación, sino que almacena los datos a través de nodos jerárquicos en archivos JSON.

### 3.3. Diseño de la arquitectura

#### 3.3.1. Arquitectura del sistema

El sistema sigue una arquitectura cliente-servidor. Toda la información se almacena en un servidor de bases de datos en la nube, que actúa como servidor, al que los clientes Android con la aplicación se conectan para acceder a la información.

A continuación, se muestra el diagrama de despliegue con los nodos y componentes del sistema:



*Ilustración 10: Diagrama de despliegue*

#### 3.3.2. Arquitectura de la aplicación

En la aplicación Android se utilizará el paradigma MVC (Modelo-Vista-Controlador) que se caracteriza por separar los datos de la aplicación (Modelo), la interfaz de usuario (Vista) y la lógica de negocio (Controlador).

Así, la aplicación Android a desarrollar, estará formada por:

- **Modelo:** las entidades del modelo de datos
- **Vista:** las distintas Activity en formato xml que definen la interfaz de usuario de esa pantalla.
- **Controlador:** las clases Java correspondientes a cada Activity y que controlan la lógica de los eventos recibidos en los elementos de la vista, así como el envío de los datos del modelo para que sean mostrados en la interfaz.



## 4. Implementación

### 4.1. Desarrollo

En este apartado se describirán las herramientas y librerías utilizadas durante la fase de implementación, así como las decisiones técnicas relevantes que se han tomado a lo largo de esta fase.

#### 4.1.1. Herramientas y APIs utilizadas

##### 4.1.1.1 Android

La aplicación desarrollada en este trabajo se ha realizado para Android, una plataforma basada en Linux y cuya arquitectura es la siguiente (Ilustración 11).

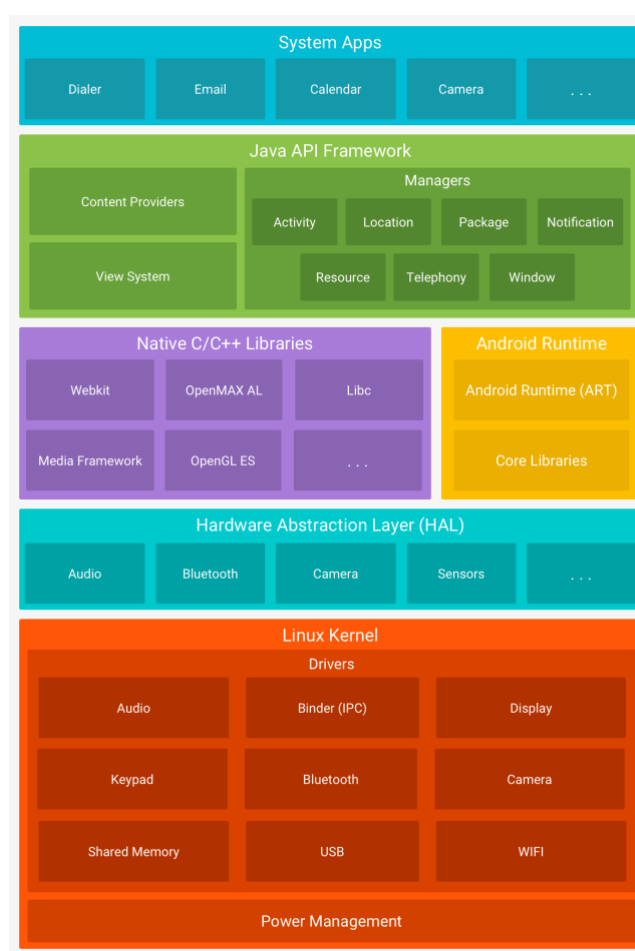


Ilustración 11: Arquitectura de Android

(Fuente: <https://developer.android.com/guide/platform>)

La aplicación se ha desarrollado para Android de forma nativa, utilizando Java como lenguaje de programación. Como entorno de desarrollo se ha utilizado Android Studio, es el IDE oficial para el desarrollo de Android e incluye todas las utilidades necesarias para realizar el desarrollo: instalación de los SDK para las distintas versiones de Android, creación de dispositivos virtuales para la



ejecución de las aplicaciones, conexión con dispositivos físicos, depuración en dispositivos físicos y/o virtuales, ...

### 4.1.1.2 *Firebase*

Como sistema de almacenamiento e identificación de usuarios, se ha utilizado Firebase, plataforma en la nube proporcionada por Google para facilitar el desarrollo de aplicaciones que ofrece gran cantidad de servicios a través de distintos planes de suscripción. Para el desarrollo de este trabajo se ha utilizado el plan básico (Spark) y de las utilidades disponibles para este plan, se ha utilizado **Firebase Realtime** como base de datos, **Firebase Storage** para el almacenamiento de archivos y **Firebase Authentication** para el registro y acceso de los usuarios de la aplicación.

- **Firebase Realtime Database:** base datos cloud NoSQL. Almacena la información en formato JSON y sincroniza la información en tiempo real a los clientes conectados a ella.
- **Firebase Storage:** sistema de almacenamiento de archivos sobre Google Cloud Storage, permite el acceso a los archivos desde Firebase y Google Cloud. Proporciona un SDK para cargar y descargar los archivos desde los clientes.
- **Firebase Authentication:** sistema de identificación y acceso que proporciona servicios de backend para la gestión de los usuarios a través de un SDK.

### 4.1.1.3 *Firebase Open Source*

Además de las herramientas de la plataforma Firebase, existe una comunidad de desarrolladores, Firebase Open Source, que proporcionan herramientas útiles para el uso de ciertos servicios de Firebase.

Una de estas herramientas es Firebase UI, que permite descargar de forma sencilla una imagen almacenada en Firebase Storage y mostrarla en la interfaz con la herramienta Glide.

## 4.1.2. Decisiones técnicas relevantes

### 4.1.2.1 *Estructura del proyecto Android*

La aplicación se ha desarrollado de forma nativa en Android, para dispositivos con Android 8.0 o posterior (API 26).

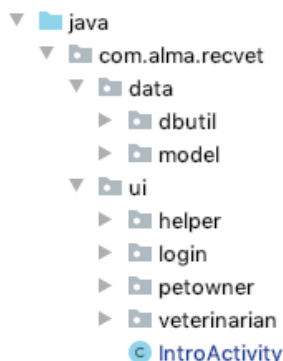
El proyecto está estructurado en paquetes, comenzando por: *com.alma.recvet*. Sobre este paquete base se ha creado la estructura de paquetes de la aplicación, realizando una primera división entre datos e interfaz de usuario:

- **data:** Contiene las clases relacionadas con el modelo y la base de datos.
- **ui:** Contiene las clases de la interfaz de usuario, Activities y Fragment. Dado que en la aplicación existen dos perfiles de usuario, se ha realizado una organización en paquetes de las Activities y Fragments utilizadas por cada perfil. El primer nivel "ui" sólo contiene la IntroActivity, splash screen que se



muestra siempre que se inicia la aplicación, el resto de las clases de la UI están organizadas en distintos paquetes:

- helper: clases de apoyo a la UI
- login: Activies y clases para el inicio de sesión y registro de usuario
- petowner y veterinarian: Contiene las MainActiviy de cada perfil de usuario y los paquetes en los que se organizan el resto de las pantallas (todas Fragmentos) del perfil.



*Ilustración 12: Estructura de paquetes.*

#### 4.1.2.2 Diseño de las interfaces de usuario

El diseño de las interfaces de usuario se ha realizado mayoritariamente con ConstraintLayout en los layouts contenedores principales y en casos puntuales se ha utilizado LinearLayout para algún elemento interior concreto.

ConstraintLayout permite diseñar las interfaces a partir de restricciones/relaciones entre los distintos elementos, llamadas “constraint”, que permite establecer su ubicación en la pantalla de forma relativa con lo que la interfaz se adapta a distintos tamaños de pantalla.

Se ha añadido un ScrollView sobre el que se construye el Constraint Layout para que, en los casos en los que, la pantalla no tenga el tamaño mínimo suficiente para contener todos los elementos, se podrá acceder a ellos a través de un gesto de scroll sobre la pantalla. Esto podría ocurrir, por ejemplo, si el usuario girase la pantalla y la colocase de forma horizontal. Otras formas de solucionar este problema podrían haber sido, impedir el giro de la pantalla forzando a que se muestre siempre en vertical o realizar un diseño adicional para la posición horizontal de la pantalla.

Para los tamaños de los elementos, salvo excepciones, se ha optado por:

- Match parent y wrap content: para los contenedores, de forma que se adaptan al elemento padre o al contenido respectivamente.
- 0dp: para ciertos elementos. Significa que, el tamaño del elemento se calcula en base a los constraint que se le han aplicado.
- Para los casos en los que se han fijado tamaños concretos, se han utilizado píxeles independientes de la densidad (dp) en las vistas y píxeles independientes de la escala (sp) en el texto.



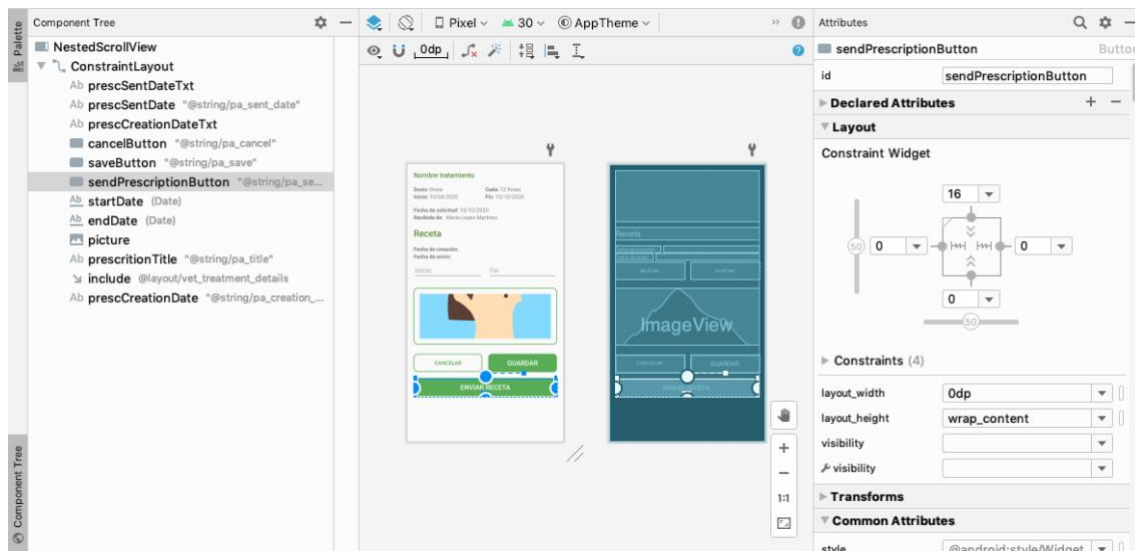


Ilustración 13: Diseño de layout con ConstraintLayout y Scroll View

### 4.1.2.3 Modelos de las vistas y Live Data

Se ha utilizado ViewModel y Live Data en las Activities y Fragments de la aplicación con el fin de gestionar los datos utilizados en la UI a través de los ciclos de vida. A través del uso de Live Data se pueden observar los cambios en los datos y actuar en base a esos cambios.

De forma general, para cada Activity/Fragment, se han implementado las siguientes clases:

- **Activity/Fragment:** desde la que se observan los datos live y se actúa en base a esos cambios.
- **View Model:** almacena y administra los datos utilizados en la Activity/Fragment.
- **Estado del formulario:** en los casos de los formularios, en esa clase se mantiene el estado de validación de los campos de los formularios, de forma que, cada vez que el usuario escribe sobre un campo de formulario se actualiza esta clase que a su vez es observada y, en base a estos cambios, se muestra un error si el campo no cumple los criterios de validación.
- **ActionResult:** A través de esta clase se recibe en la UI el resultado de acciones realizadas, indicando si ha tenido éxito o se ha producido un error y, en caso de éxito contiene los datos con el resultado de la acción. Inicialmente cada formulario tenía su clase Result pero, finalmente se implementó una clase genérica ActionResult para poder utilizarla en todos los casos indicando la clase concreta de uso en cada formulario.

### 4.1.2.4 Estructura de la UI: Navigation y Navigation Drawer

La interfaz de usuario está estructurada en base al componente Navigation mediante el uso de Fragments. Sólo las primeras pantallas (intro, login, register y main) son Activities, el resto de las pantallas se construyen con Fragmentos sobre la MainActivity y el componente Navigation.

El componente Navigation consta de tres elementos principales:



- Navigation graph: El gráfico de navegación es un recurso XML que puede verse y editarse también de forma gráfica (Ilustración 15) en el que se define la navegación entre las distintas pantallas, a través de acciones que llevan a “destinos”.
- NavHost: Contenedor donde se muestran los destinos del gráfico de navegación. Se ha utilizado la implementación predeterminada NavHostFragment que muestra destinos de Fragmentos (que son los utilizados en esta aplicación).
- NavController: Administra la navegación de la aplicación para un NavHost, a través de él podemos navegar siguiendo la pila de llamadas o ir a un destino específico.

Esta aplicación tiene dos gráficos de navegación, uno para el propietario de mascotas (Ilustración 14) y otro para el veterinario (Ilustración 15).

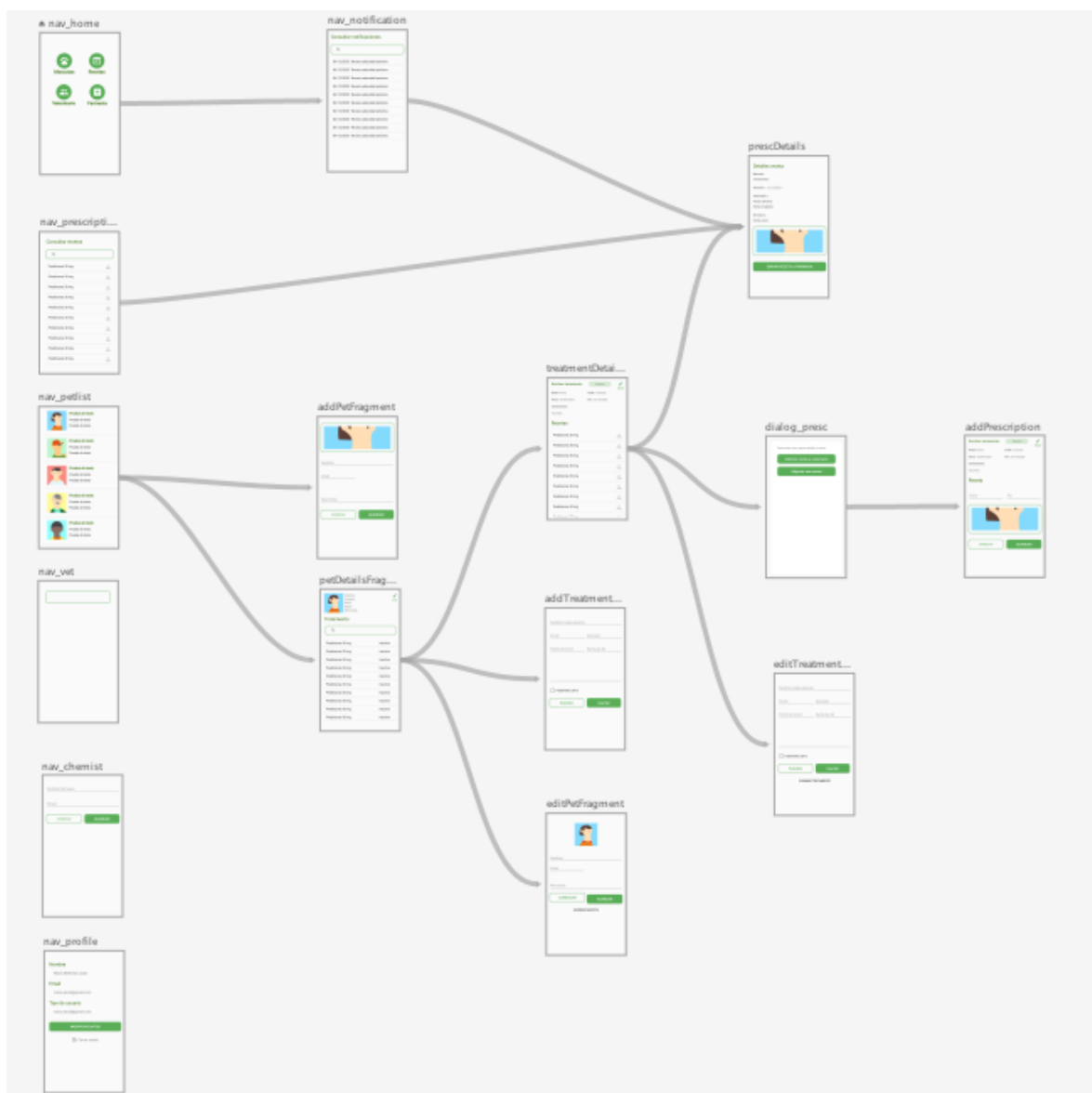


Ilustración 14: Gráfica de navegación (visual) del propietario de mascotas



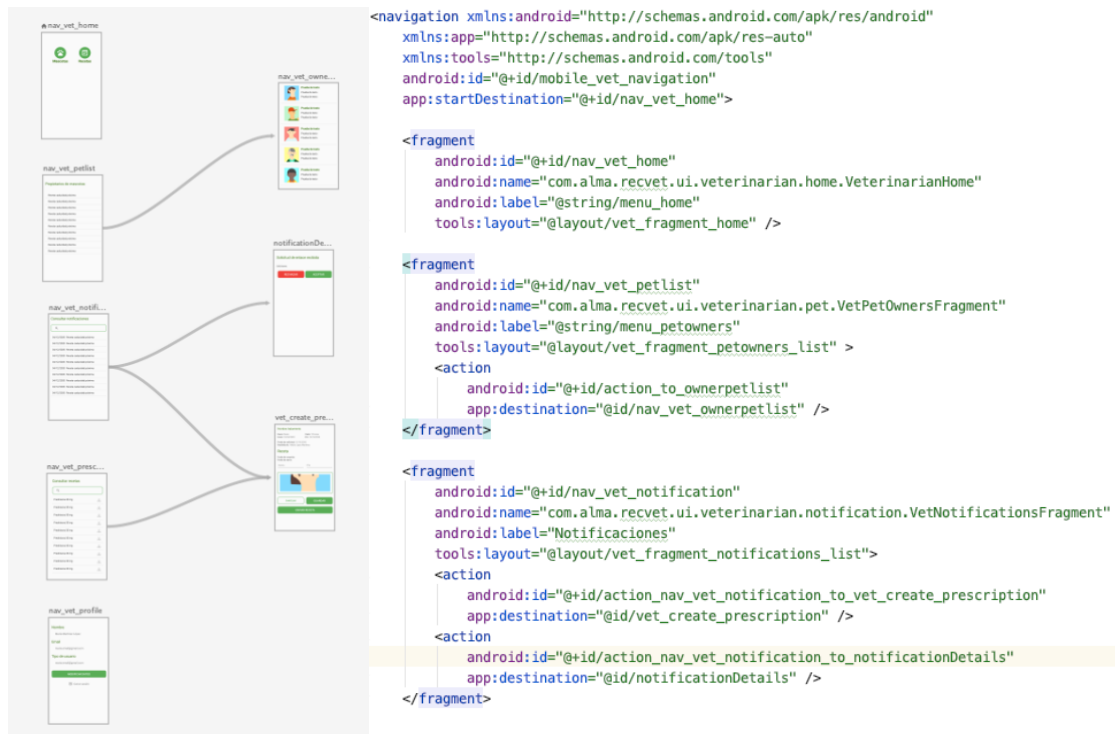


Ilustración 15: Gráfica de navegación del veterinario

El componente Navigation se usa en conjunto con el Navigation Drawer, controlado completamente en la actividad principal (Main Activity para el propietario de mascotas y VetMainActivity para el veterinario).

Al conectar el Navigation Controller con el Navigation Drawer informamos al Navigation Controller de cuales son los fragmentos de nivel superior definidos en la configuración del AppBar.

```
mAppBarConfiguration = new AppBarConfiguration.Builder(
    R.id.nav_home, R.id.nav_petlist, R.id.nav_prescription, R.id.nav_chemist,
    R.id.nav_vet, R.id.nav_notification, R.id.nav_profile)
    .setOpenableLayout(drawer)
    .build();
```

Ilustración 16: Configuración de los fragmentos de nivel superior

A través de esta configuración de los elementos de navegación, se navega a los elementos de primer nivel desde el drawer o a través del NavController si se requiere esta navegación desde otros fragmentos (por ejemplo, desde Home se accede a pantallas de primer nivel). Y, la navegación hacia los destinos que no se encuentran en el nivel superior, se realiza a través del NavController:

```
NavController navController = Navigation.findNavController(getActivity(), R.id.nav_host_fragment);
navController.navigate(R.id.addPrescription);
```

Y, si queremos volver a una pantalla anterior en la pila, podemos hacerlo así:

```
NavController navController = Navigation.findNavController(getActivity(), R.id.nav_host_fragment);
navController.popBackStack();
```



Definiendo de esta forma los destinos de nivel superior, la toolbar de la aplicación cambiará si el fragmento actual es un destino de primer nivel o no, mostrando el icono de acceso al menú en el caso de los destinos de primer nivel y una fecha para volver al destino anterior en el resto de los casos.

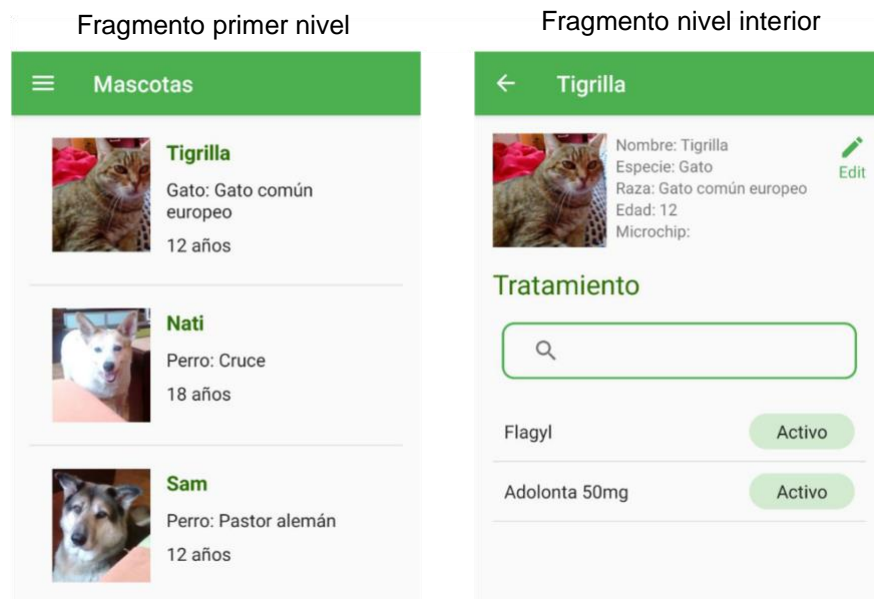


Ilustración 17: Toolbar en fragmentos de primer nivel y nivel interior

También se controla desde la MainActivity el botón flotante utilizado en distintas pantallas para acceder a los formularios de añadir mascotas, tratamientos, recetas. Para seleccionar en que pantallas se muestra y la acción a realizar en cada caso, se añade un Listener en la activity donde se comprueba, a través del NavController, el destino cargado actualmente.

```
navController.addOnDestinationChangeListener(new NavController.OnDestinationChangeListener() {
    @Override
    public void onDestinationChanged(@NonNull NavController controller,
                                     @NonNull NavDestination destination,
                                     @Nullable Bundle arguments) {
        Set<Integer> fragmentsWithFab = new HashSet<>() {
            Arrays.asList(R.id.nav_petlist,
                          R.id.petDetailsFragment, R.id.treatmentDetailsFragment));

        if (fragmentsWithFab.contains(destination.getId())) {
            fab.show();
        } else {
            fab.hide();
        }
    }
});
```

Ilustración 18: Selección de fragmentos con botón flotante

Y, en base al fragmento actual cargado, se selecciona la acción a ejecutar en caso de que el usuario pulse el botón:



```

fab.setOnClickListener((view) -> {
    switch (navController.getCurrentDestination().getId()) {
        case R.id.nav_petlist: {
            navController.navigate(R.id.addPetFragment);
            break;
        }
        case R.id.petDetailsFragment: {
            navController.navigate(R.id.addTreatmentFragment);
            break;
        }
        case R.id.treatmentDetailsFragment: {
            navController.navigate(R.id.dialog_presc);
            break;
        }
    }
});

```

Ilustración 19: Navegación desde botón flotante en MainActivity

#### 4.1.2.5 Paso de argumentos entre fragmentos con Safe Args

A través del componente Navigation se pueden pasar datos entre fragmentos. Para ello, se definen en el gráfico de navegación, los argumentos que recibirá el destino.

Se permite el paso de argumentos de distintos tipos incluyendo objetos siempre que la clase sea `Serializable` o `Parcelable`. De todas formas, no se recomienda pasar de esta forma objetos complejos con gran cantidad de información y usar en ese caso un `ViewModel` compartido entre varios fragmentos.

Se recomienda el uso del complemento de Gradle “Safe Args” para el paso de argumentos. De esta forma, al compilar se generan clases que permiten la navegación asegurando los tipos.

En esta aplicación se han utilizado `SafeArgs` para el paso de argumentos a los destinos definidos en las rutas del gráfico de navegación. Se han pasado tipos simples y objetos ligeros y, en los casos en los que se necesitaba objetos más pesados u objetos de distintos fragmentos previamente cargados, se ha utilizado un `ViewModel` compartido.

Cuando se utiliza `SafeArgs`, al compilar se generan las siguientes clases y métodos para cada acción y destino de recepción definido en el gráfico de navegación:

- Se crea una clase por cada destino en el que se origina una acción. El nombre de esta clase es el nombre del destino de origen, unido a la palabra “Directions”. Por ejemplo, en el caso de la acción marcada en la Ilustración 20, la clase generada se llamaría `SpecifyAmountFragmentDirections`. Esa clase tiene un método para cada acción definida.
- Para cada acción a través de la que se pasará el argumento, se crea una clase interna. Por ejemplo, en el caso de la acción marcada en la Ilustración 20, la acción se llama `editTreatmentFragmentAction`, la clase



generada se llamará `EditTreatmentFragmentAction`. En la Ilustración 22 se muestra un ejemplo de navegación utilizando estas clases.

- Para el destino de la acción se crea también una clase. El nombre de esta clase es el nombre del destino, unido a la palabra "Args". Por ejemplo, en el caso del destino de la acción anterior, el parámetro de define en el xml (Ilustración 21) y se crea la clase `EditTreatmentFragmentArgs`. Los argumentos e recuperan a través del método `fromBundle()` de esta clase. En la Ilustración 23 se muestra un ejemplo de recuperación de parámetros.

```
<fragment
    android:id="@+id/treatmentDetailsFragment"
    android:name="com.alma.recvet.ui.petowner.pets.treatment.TreatmentDetailsFragment"
    android:label="Detalles Tratamiento"
    tools:layout="@layout/fragment_treatment_details_list" >

    <argument
        android:name="selectedTreatment"
        app:argType="com.alma.recvet.data.model.Treatment"
        app:nullable="true"
        android:defaultValue="@null" />

    <action
        android:id="@+id/addPrescriptionDialogAction"
        app:destination="@id/dialog_presc" />
    <action
        android:id="@+id/prescDetailsAction"
        app:destination="@id/prescDetails" />
    <action
        android:id="@+id/editTreatmentFragmentAction"
        app:destination="@id/editTreatmentFragment" />
</fragment>
```

Ilustración 20: Ejemplo de definición acción.

```
<fragment
    android:id="@+id/editTreatmentFragment"
    android:name="com.alma.recvet.ui.petowner.pets.treatment.EditTreatmentFragment"
    android:label="Editar tratamiento"
    tools:layout="@layout/edit_treatment_fragment" >

    <argument
        android:name="selectedTreatment"
        app:argType="com.alma.recvet.data.model.Treatment"
        app:nullable="true"
        android:defaultValue="@null" />
</fragment>
```

Ilustración 21: Definición de parámetro en destino

```
NavController navController = Navigation.findNavController(getActivity(),
    R.id.nav_host_fragment);

TreatmentDetailsFragmentDirections.EditTreatmentFragmentAction action =
    TreatmentDetailsFragmentDirections.editTreatmentFragmentAction();
action.setSelectedTreatment(viewModel.getSelectedTreatment());
navController.navigate(action);
```

Ilustración 22: Navegación utilizando acción con Safe Args



```
viewModel.setTreatment(EditTreatmentFragmentArgs
    .fromBundle(getArguments()).getSelectedTreatment());
```

*Ilustración 23: Obtención de parámetros en destino*

#### 4.1.2.6 Recycler View para las listas de elementos

Para mostrar las listas de elementos en la interfaz se han utilizado Recycler Views con Custom Adapters. Se ha implementado un click listener para detectar el click sobre los ítems de la lista.

Se ha combinado el uso de Recycler View con Search View para realizar búsquedas sobre la lista a través de la implementación de la función `getFilter()` de la interfaz `Filterable` en los `CustomAdapter`.

#### 4.1.2.7 Estructura de la base de datos

La base de datos utilizada es Firebase Realtime, base datos NoSQL en la que los datos se almacenan de forma jerárquica en formato JSON.



*Ilustración 24: Estructura general de la base de datos*

Para el diseño de la estructura de los datos se ha tenido en cuenta la forma en la que se utilizarían los datos para mostrarlos en la UI pero sin que complicase excesivamente las operaciones de escritura. Siguiendo las recomendaciones de la documentación de Firebase, se ha realizado una desnormalización de los datos, duplicando cierta información para agilizar su obtención desde la UI. En estos casos, se ha tenido en cuenta dicha duplicidad en la actualización borrado de las entidades principales de estos datos duplicados, llevando a cabo, en estos



casos, actualizaciones multi-registro que han permitido mantener la consistencia de la información almacenada en la base de datos.



Ilustración 25: Estructura de los nodos "Users" y "Vets".



Ilustración 26: Estructura del nodo "Prescriptions".

La estructura de la base de datos (Ilustración 24), se basa en cuatro nodos principales:

- **Petowners:** propietarios de mascotas. Cada uno de ellos se identifica por un ID único que se corresponde con el ID del usuario creado para él. Este nodo, puede contener otros nodos (Ilustración 27) como son:
  - **Pets:** nodo contendor que contiene la lista de mascotas del petowner. Cada una se identifica por un ID único que se crea en el momento que se añade una mascota. Cada nodo mascota puede a su vez contener un nodo contendor "**Treaments**" que contiene los tratamientos de una mascota, cada uno identificado por un ID único.
  - **Notifications:** lista de notificaciones del petowner.
- **Prescriptions:** lista de recetas (Ilustración 26), cada una con un ID único que se genera durante su creación. Aunque las recetas son de un tratamiento de una mascota, se han ubicado en el nivel superior porque se necesita un listado completo de las mismas tanto desde las pantallas del propietario como desde el veterinario. Para poder seleccionarlas por distintos criterios se han creado campos con información duplicada de propietarios, mascotas y tratamientos.
- **Users:** lista de usuarios (Ilustración 25), son los usuarios registrados en el sistema de autenticación de Firebase sólo que aquí se almacena el tipo de usuario (propietario de mascotas o veterinario) para mostrar la interfaz de usuario correspondiente.



- **Vets:** lista de veterinarios (Ilustración 25), cada uno de ellos se identifica por un ID único que se corresponde con el ID del usuario creado para él. Dentro de este nodo se almacenan las notificaciones del veterinario.



Ilustración 27: Estructura del nodo "Petowners"

#### 4.1.2.8 Acceso a la bd, consultas y modificaciones

Para poder acceder a Firebase, es necesario establecer reglas para la lectura y escritura. En este caso se ha optado por permitir que sólo los usuarios registrados puedan acceder a la base de datos para realizar operaciones de lectura/escritura. Las reglas establecidas son las siguientes:

```

"rules": {
  ".read": "auth.uid != null",
  ".write": "auth.uid != null"
}

```

Ilustración 28: Reglas de acceso a Firebase

Para los accesos a la base de datos es necesario obtener una instancia de DatabaseReference. Se ha creado una clase instancia única donde se obtiene esa referencia y al mismo tiempo actúa de fachada para llamar a los métodos de acceso creados en distintas clases Repository para la estructuración de las consultas.



Para obtener los resultados de las consultas, tanto de lectura como escritura, se utilizan listeners. En el caso de las consultas para obtener información de la base de datos, en esta aplicación se ha utilizado el método “addListenerForSingleValueEvent” para agregar un “ValueEventListener” que llama al método “onDataChange” cuando se obtienen los datos. Se utiliza este listener porque se quiere obtener el valor instantáneo de la base de datos en el momento que se realiza la consulta.

En el caso de las inserciones de nuevos datos y las modificaciones de datos existentes se han utilizado los métodos “setValue” y “updateChildren” respectivamente. Este último método permite la actualización simultánea de varios hijos al mismo tiempo, modificando sólo los valores de los campos que se le pasan. Esto ha resultado muy útil para mantener la consistencia de los datos desnormalizados.

Para obtener y enviar a la UI los resultados de las consultas se ha implementado un callback y una clase genérica Result que se instancia con la clase necesaria en cada momento pudiendo usarla en todos los casos.

#### *4.1.2.9 Gestión de usuarios con Firebase Authentication.*

La gestión de usuario en la aplicación se realiza a través del sistema Firebase Authentication que proporciona métodos para realizar las gestiones de usuario habituales:

- **Registrar un usuario:** El registro del usuario se realiza a través del método “createUserWithEmailAndPassword” que permite registrar un usuario a través de un email y contraseña. El sistema comprueba si el usuario ya está registrado y devuelve una excepción que es capturada en la aplicación para mostrar un mensaje de aviso al usuario que se está intentando registrar. Tras el registro, se obtiene el identificador único del nuevo usuario. Con este identificador se añade el usuario en la base de datos Firebase Realtime con la información adicional necesaria para la aplicación (tipo de usuario: propietario de mascotas o veterinario).
- **Realizar un inicio de sesión:** El inicio de la sesión de usuario se realiza a través el método “signInWithEmailAndPassword”.
- **Obtener el token de la sesión actual:** Cuando un usuario inicia sesión, esta se mantiene activa durante un tiempo salvo que el usuario realice un logout. Cuando se inicia la aplicación, se comprueba si está todavía activa la sesión del usuario y en este caso no se piden de nuevo las credenciales y se accede directamente al home. Esta comprobación se realiza en al IntroActivity, de forma que, si el token está activo todavía, se envía al home del usuario y sino, se envía al formulario de inicio de sesión.
- **Cerrar la sesión:** Se utiliza el método “signOut()” de Firebase Authentication para cerrar la sesión del usuario.
- **Restaurar la contraseña:** A través del método “sendPasswordResetEmail” se permite la restauración de la contraseña del usuario en caso de que no la recuerde. Para ello, este método envía un email al usuario a la cuenta que ha registrado para que pueda restaurar su contraseña.

#### *4.1.2.10 Tratamiento de imágenes con Firebase Storage*



El almacenamiento de las imágenes se realiza en Firebase Storage, guardando en Firebase Realtime el path para acceder a la imagen.

```
public void addPetImage(Context context, Uri imageUri, Pet pet) {
    Bitmap bitmap = ImageHelper.reduceImage(imageUri, context);
    byte[] imageBytes = ImageHelper.imageToCompressedJpg(bitmap);

    DatabaseApi.getInstance().uploadPetPhoto(imageBytes, pet, new IApiCallback() {
        @Override
        public void notify(Result result) {
            if (result instanceof Result.Success) {
                Pet data = ((Result.Success<Pet>) result).getData();
                addPetImageResult.setValue(new AddPetResult(data));
            } else {
                addPetImageResult.setValue(new AddPetResult("Error añadiendo la imagen"));
            }
        }
    });
}
```

*Ilustración 29: Obtención de datos de la imagen*

En los casos donde se pueden adjuntar imágenes (mascotas y recetas), se permite el acceso a la galería del teléfono a través de un Intent que accede a la galería y que devuelve la uri de la imagen como resultado. La imagen obtenida de la galería se procesa para reducir sus dimensiones y comprimirla almacenando el resultado en un array de bytes (Ilustración 29) y se llama al método que guarda la imagen en Firebase Storage (Ilustración 30).

El procesamiento de la imagen se realiza con una clase creada para este propósito, *ImageHelper*, en la que se configura las dimensiones máximas de la imagen resultado del procesado y la calidad que se aplica a la compresión. Estos valores se definen como constantes en esta versión pero podrían incluirse dentro de la configuración para que el usuario pudiese cambiar estos valores establecidos por defecto.

Para la descarga de las imágenes se ha utilizado Firebase UI y la librería Glide, que permite descargar directamente las imágenes y mostrarlas en un componente ImageView de la vista.

Para ello es necesario incluir las dependencias de Glide y Firebase UI en el proyecto y compilarlo para poder registrar el módulo de Glide (Ilustración 31) que permite la descarga directa de las imágenes desde Firebase Storage.



```
public static void uploadPetPhoto(DatabaseReference databaseReference,
                                final StorageReference mStorageRef, byte[] data,
                                final Pet pet, final IApiCallback callback) {
    final String imageId = databaseReference.push().getKey();
    StorageReference photoRef = mStorageRef.child(ST_PETS).child(imageId + DatabaseApi.IMG_SUFFIX);
    photoRef.putBytes(data)
        .addOnSuccessListener((OnSuccessListener) (taskSnapshot) -> {
            String imgPath = mStorageRef.toString() +
                DatabaseApi.PATH_SEP +
                taskSnapshot.getMetadata().getPath();
            pet.setImage(imgPath);
            callback.notify(new Result.Success<>(pet));
        })
        .addOnFailureListener((exception) -> {
            callback.notify(new Result.Error(new IOException("Error saving pet photo", exception)));
        });
}
```

Ilustración 30: Guardar la imagen en Firebase Storage

```
@GlideModule
public class MyAppGlideModule extends AppGlideModule {

    @Override
    public void registerComponents(Context context, Glide glide, Registry registry) {
        // Register FirebaseImageLoader to handle StorageReference
        registry.append(StorageReference.class, InputStream.class,
            new FirebaseImageLoader.Factory());
    }
}
```

Ilustración 31: Módulo de Glide para descarga imágenes

Una vez que se ha registrado el módulo, podemos utilizar el siguiente método (Ilustración 32) para mostrar las imágenes directamente en la UI.

```
public void downloadDirect(Context context, StorageReference imageRef, ImageView imageView) {
    try {
        if (imageRef != null) {
            // Download directly from StorageReference using Glide
            GlideApp.with(context) GlideRequests
                .load(imageRef) GlideRequest<Drawable>
                .centerCrop()
                .transition(DrawableTransitionOptions.withCrossFade())
                .into(imageView);
        } else {
            Log.e(TAG, msg: "Null image storage reference!");
        }
    } catch (Exception ex){
        Log.e(TAG, ex.toString());
    }
}
```

Ilustración 32: Descarga directa a un ImageView con Glide



## 4.2. Pruebas

A lo largo de la implementación se realizaron pruebas de las funcionalidades que se iban implementando. Se realizaron de forma manual, utilizando checklist con los distintos test a realizar. Se utilizó la herramienta de Debugger del Android Studio para localizar y corregir los errores encontrados. Estas pruebas se han realizado mayoritariamente en un dispositivo físico, Xiaomi Mi A2 Lite y en un emulador de dicho dispositivo.

A continuación, se indican los checklist más relevantes de las pruebas:

General	
Pantalla intro	<ul style="list-style-type: none"> <li>• Muestra la pantalla</li> <li>• Verifica el token del usuario</li> </ul>
Inicio sesión	<ul style="list-style-type: none"> <li>• Comprueba las credenciales usuario</li> <li>• Recupera contraseña</li> </ul>
Registro	<ul style="list-style-type: none"> <li>• Crea nuevo usuario</li> <li>• Devuelve error si el usuario ya está registrado</li> </ul>
Perfil de usuario	<ul style="list-style-type: none"> <li>• Muestra los datos del usuario</li> <li>• Permite añadir foto del usuario</li> <li>• Se puede cerrar sesión</li> </ul>
Inicio	<ul style="list-style-type: none"> <li>• Muestra la pantalla de inicio</li> <li>• Accede directamente si se conserva el token</li> <li>• Se muestran las opciones adecuadas para el perfil de usuario y llevan a las pantallas adecuadas.</li> <li>• Se muestra el panel de notificaciones si hay notificaciones nuevas.</li> </ul>
Menú	<ul style="list-style-type: none"> <li>• Se muestra el menú correspondiente al perfil de usuario y las opciones del menú llevan a las pantallas adecuadas.</li> </ul>

Específico propietario de mascotas	
Añadir mascota, añadir tratamiento, añadir receta	<ul style="list-style-type: none"> <li>• Validación de campos</li> <li>• Si se selecciona una foto, la guarda en Firebase Storage</li> <li>• Guarda el elemento correctamente en Firebase database</li> <li>• Vuelve a la lista de elementos al terminar de añadirlo</li> </ul>
Lista de Mascotas, lista de tratamientos	<ul style="list-style-type: none"> <li>• Muestra la lista de elementos y sus estados correctamente</li> <li>• Permite añadir uno nuevo</li> <li>• Al pulsar sobre el elemento de la lista, muestra detalles de este.</li> </ul>
Recetas	<ul style="list-style-type: none"> <li>• Añade correctamente una receta</li> </ul>



	<ul style="list-style-type: none"> <li>• Permite solicitar una receta al veterinario si hay enlace con un veterinario</li> <li>• Si no hay enlace, no permite solicitar receta</li> </ul>
Consultar recetas	<ul style="list-style-type: none"> <li>• Muestra la lista de recetas</li> <li>• Filtra correctamente las recetas según el filtro</li> </ul>
Detalle mascota	<ul style="list-style-type: none"> <li>• Muestra detalles de la mascota</li> <li>• Permite editar los datos de la mascota</li> <li>• Muestra la lista de tratamientos de la mascota</li> <li>• Permite añadir un nuevo tratamiento</li> </ul>
Detalle tratamiento	<ul style="list-style-type: none"> <li>• Muestra detalles del tratamiento</li> <li>• Permite editar los datos</li> <li>• Muestra la lista de recetas</li> <li>• Permite añadir una receta</li> </ul>
Detalle receta	<ul style="list-style-type: none"> <li>• Muestra detalles de la receta</li> <li>• Se envía la receta a la farmacia por email correctamente si hay datos de una farmacia.</li> <li>• Se guardan los datos de envío y de la farmacia en la receta.</li> </ul>
Notificaciones	<ul style="list-style-type: none"> <li>• Se muestran las notificaciones nuevas en la pantalla de inicio</li> <li>• Se reciben notificaciones cuando se recibe una receta de un veterinario o una receta está a punto de caducar</li> <li>• Cuando se lee la notificación se marca como leída</li> </ul>
Veterinario	<ul style="list-style-type: none"> <li>• Se envía la solicitud de enlace correctamente</li> <li>• Se cancela la solicitud de enlace</li> <li>• Se cambia el estado de la solicitud a aceptado cuando el veterinario la confirma</li> </ul>
Farmacia	<ul style="list-style-type: none"> <li>• Se guardan correctamente los datos de la farmacia.</li> </ul>

Específico veterinario	
Propietarios y mascotas	<ul style="list-style-type: none"> <li>• Muestra correctamente los propietarios con los que está enlazado</li> <li>• Muestra correctamente las mascotas de cada propietario</li> </ul>
Recetas	<ul style="list-style-type: none"> <li>• Muestra la lista de recetas con su estado</li> <li>• Filtras las recetas correctamente</li> <li>• Al pulsar sobre una receta accede a la pantalla de detalles</li> </ul>
Detalles receta	<ul style="list-style-type: none"> <li>• Permite establecer fechas y adjuntar imagen</li> <li>• Guardar los datos y marca la fecha de creación</li> </ul>



	<ul style="list-style-type: none"> <li>• Permite enviar la receta al propietario</li> <li>• Guarda la fecha de envío al enviar la receta</li> </ul>
Notificaciones	<ul style="list-style-type: none"> <li>• Se reciben correctamente las notificaciones de recetas y de enlaces con el propietario</li> <li>• Al pulsar sobre la notificación se puede ver el detalle</li> </ul>
Enlace con propietario	<ul style="list-style-type: none"> <li>• Se puede aceptar o rechazar el enlace y ambas funciona correctamente.</li> </ul>

Algunos de los errores detectados con estas pruebas fueron los siguientes:

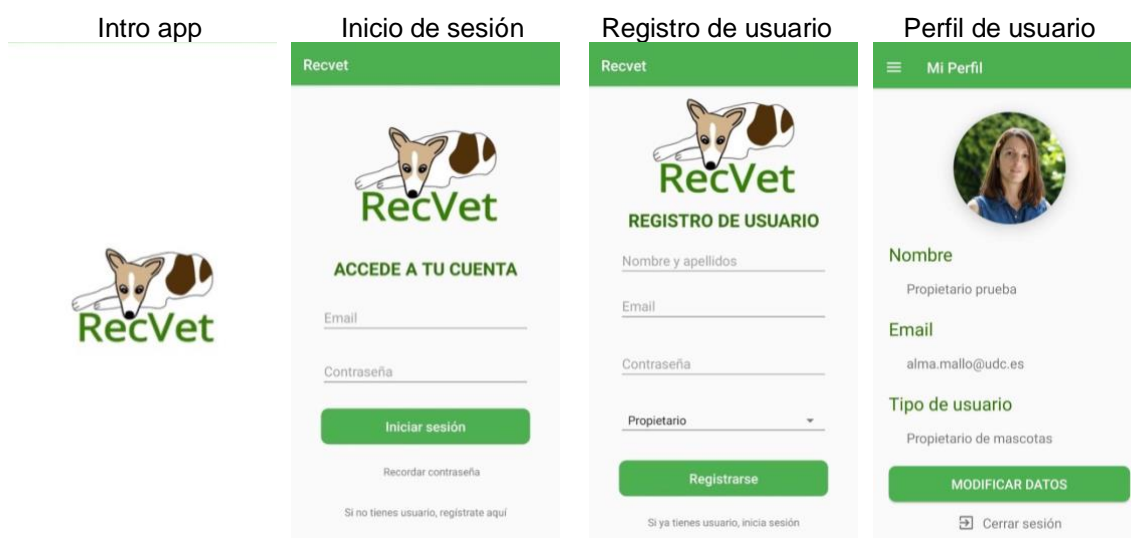
- No se comprobaba que existiese el usuario en el registro de usuario.
- Tras añadir una mascota, tratamiento o receta, al pulsar el botón guardar, la aplicación no volvía a la lista de elementos, se quedaba en la misma pantalla con lo que si se pulsaba el botón de nuevo se añadía de nuevo el mismo elemento.
- No se guardaban correctamente todos los datos de los tratamientos, se escribía en varios campos de texto el mismo valor (problema de haber utilizado copiar/pegar durante el desarrollo).
- En la edición de mascotas, si se cambiaba la imagen, no se borraba la imagen anterior de Firebase Storage, sólo se añadía la nueva.
- En el envío de recetas por email a la farmacia, al adjuntar la imagen de la receta el sistema devolvía un error. El motivo se debía a que es necesario definir un FileProvider para poder compartir una imagen con otra aplicación, como la aplicación de mail en este caso.

### 4.3. Funcionamiento de la aplicación

A continuación, se incluyen capturas de ejecución de la aplicación con las funcionalidades descritas en los casos de uso.

#### 4.3.1. Funcionalidades comunes

Las funcionalidades relacionadas con la gestión de usuarios, así como la pantalla de introducción para la aplicación, son comunes para los dos perfiles de usuario.





### Comentarios:

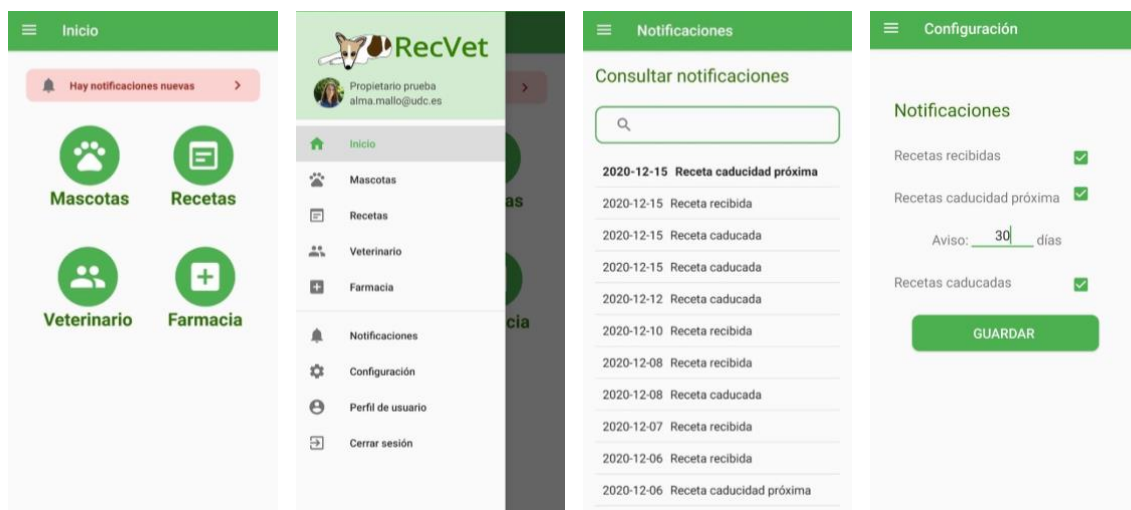
- Tras el registro del usuario se realiza de forma automática el inicio de sesión.
- La funcionalidad de cerrar sesión está accesible desde la pantalla de perfil de usuario y desde el menú de la aplicación de cada uno de los perfiles de usuario.

La funcionalidad de recordar contraseña envía un email a través de los servicios proporcionados por Firebase al usuario para que pueda restaurar la contraseña:



### 4.3.2. Funcionalidades del perfil: propietario de mascotas

#### 4.3.2.1 Inicio, menú y notificaciones




Comentarios: El panel de notificaciones de la pantalla principal sólo aparece cuando hay nuevas notificaciones que mostrar. Si no hay nuevas notificaciones, se pueden consultar las notificaciones existentes desde el apartado del menú "Notificaciones".




## 4.3.2 Mascotas con sus tratamientos y recetas


Mascotas



**Tigrilla**  
Gato: Gato común europeo  
12 años



**Nati**  
Perro: Cruce  
18 años



**Sam**  
Perro: Pastor alemán  
12 años

+

Añadir mascota

Nombre

Edad

Perro

Raza


Microchip

CANCELAR

GUARDAR

+

Sam



Nombre: Sam  
Especie: Perro  
Raza: Pastor alemán  
Edad: 12  
Microchip:

Edit

Tratamiento

Flagyl 250 mg

Activo

Adolonta 50 mg

Activo

Prednisona 30 mg

Activo

Enalapril 20 mg

Activo

Kefloridina 500 mg


Activo

Cognidine

Activo

+

Editar mascota



Sam

12

Perro

Pastor alemán

Microchip

CANCELAR

GUARDAR

ELIMINAR MASCOTA

Añadir tratamiento

Nombre medicamento

Dosis

Intervalo

Fecha de inicio

Fecha de fin

☐ Tratamiento activo

Cancelar

Guardar

Detalles Tratamiento

Flagyl 250 mg

Activo

Edit

Dosis: 3 comprimidos Cada: 12 horas

Inicio: 01/10/2020 Fin: 10/12/2020

Comentarios:

Lo toma por inflamación crónica del intestino.

Recetas

10/12/2020 - 10/03/2021

01/10/2020 - 13/12/2020

+

Editar tratamiento

Flagyl 250 mg

3 comprimidos

12 horas

01/10/2020

10/12/2020

Lo toma por inflamación crónica del intestino.

☒ Tratamiento activo

Cancelar

Guardar

ELIMINAR TRATAMIENTO

Detalles Tratamiento

Flagyl 250 mg

Activo

Edit

Dosis: 3 comprimidos Cada: 12 horas

Inicio: 01/10/2020 Fin: 01/10/2020

Comentarios:

Seleccione cómo quiere añadir la receta

Solicitar receta a veterinario

Adjuntar una receta

Añadir receta

Flagyl 250 mg

Dosis: 3 comprimidos Cada: 12 horas

Inicio: 10/12/2020 Fin: 31/03/2021

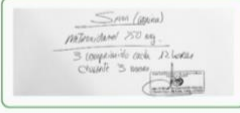
Comentarios:

Lo toma por inflamación crónica del intestino.

Receta

10/12/2020

10/3/2021



CANCELAR

GUARDAR

Recetas

Consultar recetas

Tigrilla, Flagyl, 01/01/2021

Tigrilla, Flagyl, 15/12/2020

Sam, Flagyl 250 mg, 10/12/2020

Tigrilla, Flagyl, 08/12/2020

Tigrilla, Adolonta 50mg, 08/12/2020

Tigrilla, Flagyl, 03/12/2020

Tigrilla, Flagyl, 01/12/2020

Tigrilla, Flagyl, 01/12/2020

Sam, Flagyl 250 mg, 01/10/2020

Receta

Detalles receta

Mascota: Tigrilla  
Tratamiento: Flagyl  
Duración: 15/12/2020 - 15/01/2021  
Solicitada a: Veterinario prueba  
Fecha solicitud: 15/12/2020  
Fecha recepción: 15/12/2020  
Enviada a:  
Fecha envío:

ENVIAR RECETA A FARMACIA



## 4.3.2.3 Enlace con veterinario y datos farmacia

The first screen, titled 'Veterinario', shows a message 'No tienes enlace con el veterinario' and a prompt to 'Introduce el email del veterinario para enviarle la solicitud de enlace'. It includes an 'Email' input field and an 'Enviar solicitud de enlace' button.

The second screen, also titled 'Veterinario', shows a confirmation box for 'Veterinario prueba' with email 'alma.mallo@mintforpeople.com' (Estado: accepted) and a 'Cancelar enlace' button.

The third screen, titled 'Farmacia', shows a form for 'Farmacia de prueba' with the email 'almamallo@gmail.com' and 'CANCELAR' and 'GUARDAR' buttons.

## 4.3.3. Funcionalidades del perfil: veterinario

### 4.3.3.1 Inicio, menú y notificaciones

The first screen, 'Inicio', shows a notification banner 'Hay notificaciones nuevas' and two main buttons: 'Mascotas' and 'Recetas'.

The second screen shows the 'RecVet' profile for 'Veterinario prueba' with a menu: Inicio, Mascotas, Recetas, Notificaciones, Perfil de usuario (selected), and Cerrar sesión.

The third screen, 'Notificaciones', shows a list of notifications under 'Consultar notificaciones', including 'Solicitud de enlace recibido' and 'Solicitud de receta recibida' with dates.

The fourth screen, 'Notificación', shows details for a 'Solicitud de enlace recibida' from 'Alma UOC' with 'RECHAZAR' and 'ACEPTAR' buttons.

### 4.3.3.2 Propietarios de mascotas, mascotas y recetas

The first screen, 'Propietarios', shows a list of 'Propietarios de mascotas' including 'Alma UOC' and 'Propietario prueba'.

The second screen, 'Mascotas', shows a list of pets: 'Tigrilla' (Gato: Gato común europeo, 12 años) and 'Sam' (Perro: Pastor alemán, 12 años).

The third screen, 'Recetas', shows a list of prescriptions under 'Consultar recetas', including 'Tigrilla, Flagyl, 08/12/2020 Enviada' and 'Tigrilla, Flagyl, 15/12/2020 Enviada'.

The fourth screen, 'Añadir receta', shows a form to add a prescription for 'Flagyl' with dosage, frequency, start/end dates, and a 'Receta' section with creation and sending dates. It includes 'CANCELAR', 'GUARDAR', and 'ENVIAR RECETA' buttons.



## 5. Conclusiones

Tras finalizar este Trabajo Fin de Máster, en este apartado procedo a analizar el trabajo realizado y los resultados obtenidos.

Desde el punto de vista del aprendizaje, este trabajo me ha permitido ampliar y afianzar los conocimientos adquiridos en el máster. Creo que he mejorado no sólo en mis habilidades en desarrollo de software (aprendizaje en desarrollo de aplicaciones Android y uso de base de datos NoSQL) sino también en planificación y diseño, etapas a veces olvidadas o menos valoradas. Considero importante destacar su importancia ya que son fundamentales para llevar a cabo un proyecto de software de calidad.

Desde el punto de vista personal, el desarrollo de este proyecto me ha resultado duro y gratificante al mismo tiempo, me ha supuesto un gran esfuerzo y he tenido que enfrentarme a circunstancias personales y laborales que me llevaron a desviarme de la planificación durante la etapa de implementación. Pude corregir posteriormente estas desviaciones con disciplina y esfuerzo, dedicando más horas de las previstas para volver a encaminar el proyecto y poder alcanzar los objetivos planteados.

La metodología utilizada creo que ha sido la adecuada. El desarrollo de versiones incrementales de productos funcionales con el usuario como centro del proceso me ha servido para focalizar mejor los objetivos a corto plazo y me ha ayudado a reencaminar el proyecto tras las desviaciones sufridas.

Respecto a la fase de diseño, considero que la metodología de Diseño Centrado en el Usuario utilizada ha sido imprescindible para llevar a cabo los objetivos planteados. Un buen estudio del contexto de uso de la aplicación y el diseño de prototipos de alta fidelidad permite por una parte enfocar el proyecto hacia el contexto en el que se utilizará y, por otro lado, detectar defectos y desviaciones en fases tempranas, facilitando su corrección.

Durante la fase de implementación, he seguido las recomendaciones de la documentación oficial de Android, utilizando las nuevas herramientas que permiten estructurar la navegación de la aplicación de forma más eficiente y que facilitan su mantenimiento. Me ha resultado difícil adentrarme en el ecosistema Android, pero fue algo que me propuse como un reto personal y fue uno de los motivos que me llevó a elegir el área de “Desarrollo de aplicaciones sobre dispositivos móviles” para la realización del Trabajo Fin de Máster.

El uso de Firebase Realtime como base de datos me ha permitido utilizar una base de datos NoSQL por primera vez en el desarrollo de una aplicación. Esto supuso un reto ya que el modelo de datos utilizado es completamente distinto al que se utiliza en las bases de datos relacionales y realizar la desnormalización de los datos para adaptarlos a la forma de acceso desde la interfaz de usuario me ha costado más de lo que pensé en un principio. Pero lo valoro como un proceso enriquecedor y creo que me ha ayudado a comprender mejor la potencia y posibilidades de este tipo de base de datos.



Desde el punto de vista del grado de realización de los objetivos, en mi opinión el proyecto ha concluido con una aplicación funcional, visualmente agradable, de uso sencillo y que cumple con las funcionalidades descritas en los requisitos.

A pesar de ello, disponiendo del tiempo necesario, se podrían realizar mejoras que aportarían valor al proyecto. En el siguiente apartado, se describen estas líneas de trabajo futuro.

### 5.1. Líneas de trabajo futuro

---

El alcance de este proyecto se ha ajustado al tiempo establecido para realizarlo con lo que, se han limitado las funcionalidades de la aplicación para que fuese posible realizarlas en los plazos establecidos.

Por tanto, han quedado abiertas varias líneas de trabajo que aportarían gran valor al proyecto:

- Notificaciones integradas en el sistema que se reciban también cuando la aplicación no se está ejecutando.
- Añadir gestión de vacunas y desparasitaciones, una funcionalidad que han valorado positivamente los usuarios con los que se ha hablado.
- Gestión de consultas veterinarias, permitiendo solicitar citas desde el perfil del propietario y confirmación de éstas desde el perfil del veterinario.
- Añadir la creación de recetas oficiales por los veterinarios. Para ello sería necesario incluir firma electrónica y generar la receta en el formato oficial que se establece en el BOE-A-2012-15711 [5].
- Incluir un tercer perfil de usuario para las farmacias.
  - Enlace de propietarios con farmacias (del mismo modo que con los veterinarios).
  - Los propietarios enviarían las recetas a través de la aplicación a las farmacias con las que tuvieran enlace.
  - Las farmacias confirmarían la recepción y enviarían un aviso a los propietarios cuando las medicinas estuviesen disponibles para recoger.
  - Generación de informe con la lista de medicamentos de recetas veterinarias dispensadas para enviar al colegio de farmacéuticos.
- Almacenamiento offline que permita usar la aplicación sin conexión a internet y que se sincronizase con el almacenamiento online cuando el dispositivo volviese a tener conexión.
- Soporte para múltiples idiomas.
- Desarrollo de la aplicación para otros sistemas móviles, principalmente para dispositivos iOS.
- Desarrollo de aplicación web, de forma que se pudiese utilizar también desde un ordenador.



## 6. Glosario

**API:** Abreviatura utilizada para “Application Programming Interface” (Interfaz de Programación de Aplicaciones). Se trata de un conjunto de funciones/métodos que realiza el papel de intermediario en la comunicación entre distintas aplicaciones. Esta capa de abstracción permite que las aplicaciones se comuniquen entre sí a través de estos métodos sin conocer detalles de su implementación.

**App:** Abreviatura utilizada para referirse a una aplicación software, utilizada de forma habitual para las aplicaciones móviles.

**Backend:** Capa de una aplicación software de acceso a datos y que contiene la lógica de la aplicación de acceso a esos datos. Es una capa no accesible directamente por los usuarios de la aplicación.

**Desnormalización:** Proceso por el cual se duplica información en ciertos puntos en las bases de datos NoSQL con el objetivo de optimizar el rendimiento en las consultas. [13]

**DCU:** Abreviatura utilizada para referirse a la metodología de Diseño Centrado en el Usuario [1].

**Firebase:** Servicio de backed de google para el desarrollo de aplicaciones web y móviles.

**Intent:** objeto de mensajería que facilita la comunicación entre distintos componentes. Es un mecanismo proporcionado por Android a través del cual se puede solicitar acciones a otros componentes intercambiar información con ellos.

**SDK:** Abreviatura utilizada para “Software development Kit” (Kit de Desarrollo de Software). Se trata de un conjunto de herramientas para desarrollar aplicaciones para un hardware y/o software específico o en un determinado lenguaje de programación.

**UI:** Abreviatura utilizada para “User Interface” (Interfaz de usuario). Son los elementos de la aplicación con los que interacciona el usuario, en este caso, las pantallas de la aplicación con sus campos de texto, botones, ...



## 7. Bibliografía

1. Documentación Diseño Centrado en el Usuario – UOC, Muriel Garreta Domingo, Enric Mor Pera, Universitat Oberta de Catalunya, 2017.
2. Documentación Ingeniería de la Usabilidad – UOC, Amaia Calvo-Fernández, Rodríguez Sergio Ortega Santamaría, Alicia Valls Saez, Mònica Zapata Lluch, Joan Vinyets i Rejón, Universitat Oberta de Catalunya, 2015
3. Documentación DCU para dispositivos móviles – UOC, Jordi Almirall López, Univeritat Oberta de Catalunya, 2013
4. The UX Book, Process and Guidelines for Ensuring a Quality User Experience, Rex Harston, Pardha S. Pyla, Elsevier, USA, 2012
5. Real Decreto 1675/2012, de 14 de diciembre, por el que se regulan las recetas oficiales y los requisitos especiales de prescripción y dispensación de estupefacientes para uso humano y veterinario. Ministerio de la Presidencia «BOE» núm. 313, de 29 de diciembre de 2012 Referencia: BOE-A-2012-15711.
6. Interaction Design Foundation, <https://www.interaction-design.org/> (Oct-Nov 2020)
7. Android Developer Fundamentals Course, <https://google-developer-training.github.io/android-developer-fundamentals-course-concepts-v2/>, (Oct-Nov 2020)
8. Android Developer Advanced Course, <https://google-developer-training.github.io/android-developer-advanced-course-concepts/>, (Oct-Nov 2020)
9. Android Developers, <https://developer.android.com/>. (Oct-Dic 2020)
10. StackOverFlow, <https://stackoverflow.com> (Oct-Dic 2020)
11. Firebase documentation, <https://firebase.google.com> (Oct-Nov 2020)
12. Firebase Open Source documentation. <https://firebaseopensource.com/> (Oct-Nov 2020)
13. Denormalization is normal with the Firebase Database. <https://www.youtube.com/watch?v=vKqXSZLLnHA> (Oct-Nov 2020)
14. Navigation Component for Android Part 2: Graphs and Deep Links, Meng Taing. <https://www.raywenderlich.com/4332831-navigation-component-for-android-part-2-graphs-and-deep-links> (Nov-Dic 2020)
15. Learn how to use fileprovider in Android with example. <https://techenum.com/learn-how-to-use-fileprovider-in-android-with-example/>, (Dic 2020)
16. Wikipedia, <https://www.wikipedia.org/>, (Sep-Dic 2020)
17. Google Play, <https://play.google.com/store/apps> (Sep-Oct, Nov 2020)



## 8. Anexos

En este capítulo se indican los apartados que son demasiado extensos para incluir dentro de la memoria y tienen un carácter autocontenido.

En el caso de este trabajo, se incluye como documento anexo a esta memoria el manual de usuario de la aplicación.