

GoodWeight, Cuida de los que te acompañan.

Antonio Gómez Martínez, 48791884Y

Universidad de Alicante

Trabajo de fin de grado

Jaume Aragonés Ferrero

• De - de 2024

El por qué y para qué

Agradecimientos

Índice:

Índice de tablas:

Índice de ilustraciones:

1. Introducción:

El proyecto que se presenta a continuación es una aplicación a la que he decidido llamar *GoodWeight*, la cual se enfoca en la creación de una aplicación para dispositivos móviles con el reto de utilizar tecnologías que apenas o nunca han sido vistas.

Durante este proceso, no solo se pretende aprender estas nuevas tecnologías e intentar dominarlas sino que también se pretende aprender el trabajo real que conlleva la creación de una aplicación multiplataforma.

Para la creación de esta aplicación se pretendía buscar algo que se pudiese llevar a cabo como TFG para obtener un poco de guía pero además qué pudiese tener un significado más allá del TFG. El objetivo era crear la aplicación y llegar a publicarla en cierto momento.

Durante esta memoria se explicará todo el proceso desde que comenzó la idea a como se ha desarrollado. Pasaremos por el comienzo de la idea, el prototipado, selección de tecnologías, desarrollo y problemáticas que han surgido a lo largo de esta.

Antes de comenzar a nada, gracias a la certeza del objetivo de la aplicación pude sacar en claro las necesidades de la aplicación y lo que realmente iba a necesitar hacer.

2. Justificación:

Mi decisión de aventurarme en el uso de tecnologías desconocidas estuvo motivada principalmente por el deseo de no sólo aprovechar una oportunidad única para aprender un nuevo framework y diversas herramientas para el desarrollo del backend, sino también para aplicar y expandir conceptos previamente adquiridos durante mi formación académica. Este enfoque me permitió tanto reforzar mi entendimiento teórico, como trasladar esos conocimientos a prácticas reales y concretas.

Al enfrentarme a este desafío, buscaba una forma de integrar teoría y práctica de manera que pudiera realizar un trabajo que fuese innovador y distintivo, pero que al mismo tiempo estuviera firmemente arraigado en los principios y las metodologías estudiadas en la Universidad de Alicante.

Además, este proyecto es una excelente oportunidad para demostrar mi capacidad de adaptación y aprendizaje autónomo, características altamente valoradas en el campo de la tecnología. Al sumergirme en herramientas y frameworks no familiares, he podido no solo incrementar mi competencia técnica, sino también desarrollar habilidades críticas de resolución de problemas y pensamiento innovador, fundamentales para mi desarrollo profesional futuro.

Por último, enfrentar este tipo de desafíos me ha permitido establecer una conexión más profunda entre los conocimientos teóricos proporcionados por mi educación y su aplicación en escenarios del mundo real, lo cual es esencial para cualquier profesional del ámbito tecnológico que aspire a tener un impacto significativo en su campo.

3. Objetivos

- **Cuidado de tu mascota:** Se busca desarrollar una aplicación que responda a la escasez de aplicaciones de seguimiento para mascotas en el mercado actual. Observando esta carencia, se ha visto una gran oportunidad para crear una herramienta que no solo cumpla con una necesidad específica de los usuarios, sino que también ofrezca una nueva funcionalidad en un sector que aún tiene mucho por explorar. Esta aplicación aspira a ser un recurso valioso para los dueños de mascotas que deseen mantener un seguimiento detallado y efectivo del bienestar y la actividad de sus animales.
- **Entender cómo aprender:** Mi objetivo es dominar el uso de tecnologías que, aunque comparten fundamentos con otras ya conocidas, presentan diferencias significativas en su implementación y aplicación. Este aprendizaje no solo me permitirá adquirir nuevas habilidades tecnológicas, sino que también facilitará mi capacidad para enfrentarme a nuevos desafíos. Entender cómo abordar y resolver problemas en este contexto será crucial para mi desarrollo profesional continuo.
- **Facilidad de uso:** La elección de React Native se ha basado en su notable facilidad de personalización. Siempre he tenido interés en el diseño de interfaces (UX/UI) y considero que esta tecnología es ideal para avanzar en este campo. React Native permite una adaptación rápida y efectiva a las necesidades específicas del proyecto, lo que me ha motivado a profundizar en su estudio y aplicación práctica, buscando siempre optimizar la experiencia del usuario final.

- **Entender tu proyecto:** Uno de mis principales objetivos ha sido comprender a fondo el proyecto en el que trabajo. He intentado desarrollar un proyecto lo más pulcro posible, que siga muchas de las buenas prácticas de programación que aprendí en la Universidad de Alicante. Este enfoque no solo mejora la calidad del software, sino que también asegura su mantenibilidad y escalabilidad a largo plazo, aspectos esenciales para el éxito de cualquier desarrollo tecnológico.

4. Estudio de Mercado:

4.1. Análisis

En España el **número de mascotas** ha experimentado un crecimiento significativo. Se estima que hay más de 30 millones de mascotas, de las cuales aproximadamente 9,3 millones son perros¹⁴. Además, se ha observado un aumento en el número de mascotas en los hogares españoles, con más del 42% de los hogares teniendo al menos una mascota.

La humanización de las mascotas en España ha ido más allá del cuidado diario de las mascotas y se ha expandido para incluir servicios no sólo en la vida del animal, sino también en la del tutor..

Para concluir esta introducción, se ha de recalcar que **el mercado de seguros** para mascotas está en auge. Desde 2021, el número de perros y gatos en los hogares españoles ha crecido en más de 2 millones, hasta alcanzar los 15,3 millones. Además, la nueva ley de bienestar animal introduce cambios importantes en las responsabilidades de los propietarios. Para el análisis del mercado que rodea a esta aplicación vamos a centrarnos en las funcionalidades que ofrecen al usuario y cómo podemos cubrir las necesidades y debilidades de nuestros competidores.

Existen muchas aplicaciones para el **cuidado de tu mascota**, pero no todas se centran en lo mismo que *GoodWeight*.

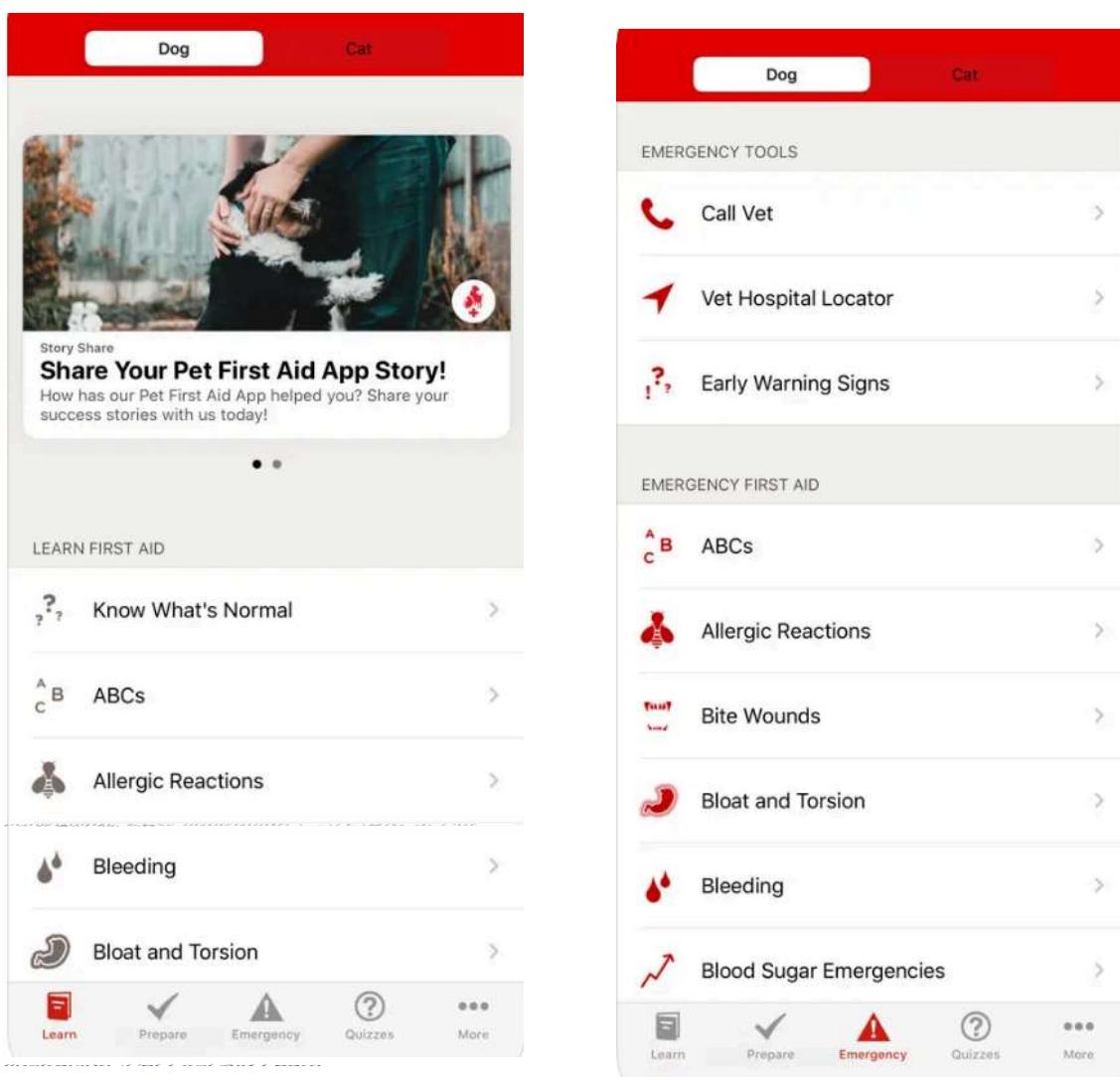
Estas aplicaciones ayudan a los dueños de mascotas a cuidar de la salud y actividad de sus acompañantes. Ofrecen funcionalidades como el control de la medicación, recordatorios de citas con el veterinario e incluso consultas por chat. Con *GoodWeight* lo que se pretende es cubrir las necesidades básicas de los usuarios y añadir ciertas funcionalidades que no se encuentren en ninguna otra aplicación. Algunas de estas son por ejemplo el cuidado de la dieta del perro, recomendando según la raza, tamaño y género del animal un peso y una dieta óptima para este, por otro lado tenemos el

seguimiento de los medicamentos y vacunas del animal, además de ciertos factores importantes sobre el animal.

A continuación, analizaremos todas estas aplicaciones que nos pueden ayudar a entender el mercado en el que estamos trabajando. Las aplicaciones que se estudiarán serán Pet First Aid¹, DogLog², 11Pets³, Pet Coach⁴ y por último MyOllie⁵.

Pet First Aid:

Esta aplicación está creada por *American Red Cross*, y está enfocada en las emergencias que pueden ocurrir con tu mascota. Ofrece videos de cómo socorrer a esta, tips para ayudar con su dieta e incluso test para prepararte ante situaciones adversas. A pesar de que no se centra en lo mismo que *GoodWeight*, nos ayuda a entender que las preocupaciones de los dueños sobre sus mascotas no solo son dietéticas.



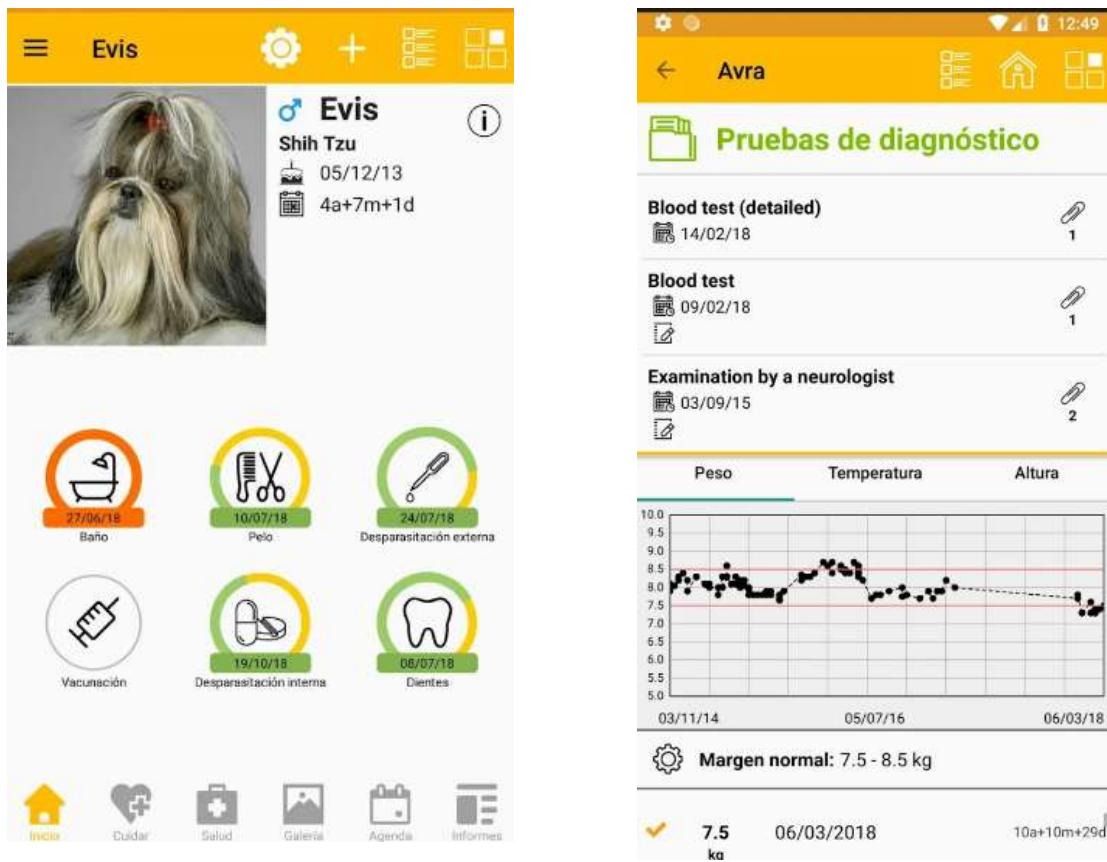
Dog Log App:

Esta aplicación fue diseñada para seguir y coordinar varios aspectos de la vida de un perro. Facilita el compartir las actividades del animal con el resto de la familia o aquellos implicados en el cuidado de este. Esta ayuda a crear eventos, recordatorios para tareas, medicinas y citas, y además analiza la rutina del animal. Además, promueve el uso social de la aplicación para dar “Likes” y “Comentar” en las publicaciones de otros dueños. La funcionalidad principal de esta es llevar al día todo lo que se hace con tu mascota, es decir, la comida que le das, y todo lo relacionado con su comportamiento.



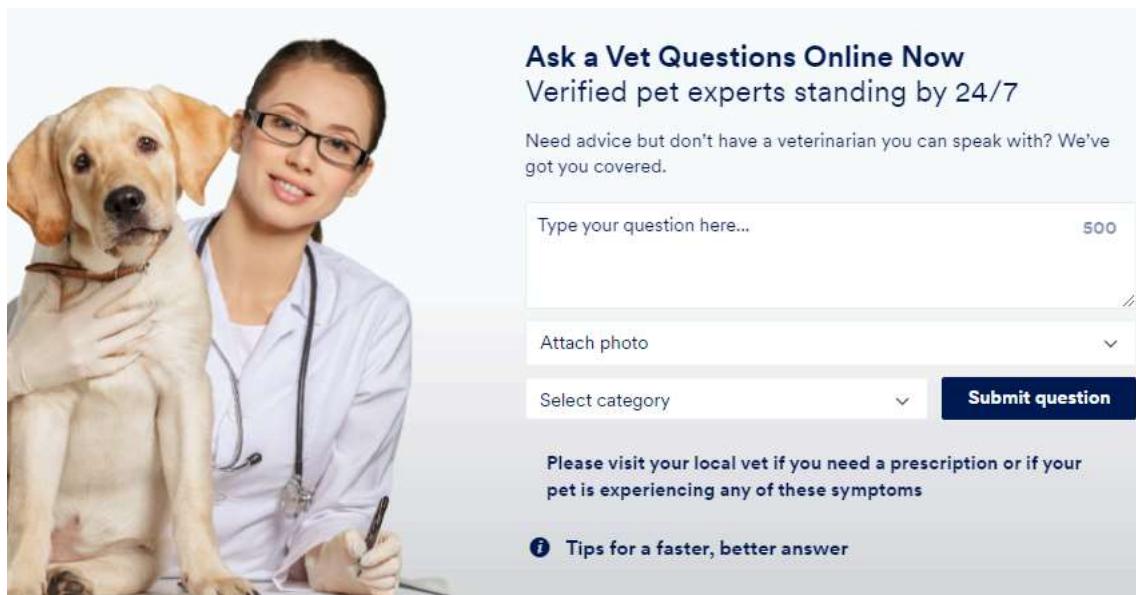
11Pets:

Esta aplicación lleva todo lo relacionado con la salud del animal, desde vacunas, medicaciones y alergias hasta las visitas al veterinario. Los usuarios pueden crear recordatorios para los medicamentos o tratamientos que el animal tenga. Esta es una de las aplicaciones con más información que hay ahora mismo en el mercado. A pesar de no ser muy grande, esta tiene más de medio millón de descargas y sus funcionalidades a pesar de ser específicas para la salud gustan a muchos usuarios.



Pet Coach:

Por último, tenemos pet coach, esta aplicación no solo te deja llevar la información médica de tu mascota, sino que también te conecta con profesionales para un chat en vivo o para consejos por video. A pesar de que da mucha comodidad no siempre es bueno el confiar en una consulta por video, aunque está bien para ciertas ocasiones.



A screenshot of the Pet Coach app's frequently asked questions section. It shows a list of categories with icons: Health (a plus sign), Behavior (a person icon), Nutrition (a bowl icon), Care (a stethoscope icon), and Breeds (a dog head icon). Next to each category is a question. The questions are: "My pet has diarrhea. What are the possible causes and treatments?", "My dog drinks a lot of water, should I worry?", "My cat is very lethargic lately, is he sick?", "My dog seems to be in pain when urinating. Why and what should I do?", "What is kennel cough and how can it be treated?", and "Is it safe to give Tylenol to my pet?". At the bottom, there is a search bar with the placeholder "Search pet content..." and a "Search" button.

Ollie - Human Grade Dog Food:

Permite gestionar una suscripción de comida para perros, donde puedes seleccionar nuevas recetas, cambiar la fecha de entrega, actualizar la frecuencia de entrega, y manejar otros aspectos relacionados con la suscripción de tu perro. También proporciona notificaciones sobre el envío y entrega de tu caja de comida, así como sobre nuevos productos lanzados.

Como es obvio esta aplicación no se centra en el seguimiento de la salud de tu mascota, sino más bien en la venta de una comida saludable para tu acompañante.



Ahora se mostrará una tabla con las funcionalidades más relacionadas con nuestra aplicación y si son incorporadas por las aplicaciones estudiadas.

4.2. Características y funcionalidades

Características	Pet First Aid	DogLogs	11Pet	Pet Coach	Ollie	GoodWeight
Seguimiento de salud	✓	✓	✓	✗	✗	✓
Seguimiento Vacunas	✗	✗	✓	✗	✗	✓
Gestión info médica	✓	✗	✓	✗	✗	✓
Historial veterinario	✗	✗	✓	✓	✗	✓
Control sobre la comida	✗	✗	✗	✗	✗	✓
Compra de comida desde la aplicación	✗	✗	✗	✗	✓	✗ *

*Se ha estudiado cómo funcionalidad de futuro

Funcionalidades básicas de la aplicación:

Funcionalidad	Descripción
Perfil integral del animal	Esta función concentra todos los detalles vitales de su mascota en un interfaz unificado. Facilita el registro y consulta de información: raza, peso, sexo, fotografía actualizada, identificación por chip, estado de esterilización y desparasitación, así como su historial médico y medicaciones actuales y pasadas.
Registro de vacunación	Gestionar con facilidad el calendario de vacunación de su mascota. Esta herramienta permite agregar, actualizar o eliminar registros de vacunas.
Administración de tratamientos médicos	Esta función permite administrar los tratamientos médicos de su mascota. Se podrá añadir o modificar la información de medicamentos y dosis, también habrá recordatorios para asegurar el seguimiento tratamiento.
Centro de asistencia y recursos	Se proporcionará una pantalla dedicada para soporte y asesoramiento, con consejos nutricionales para optimizar la dieta de su mascota, un apartado de preguntas frecuentes y una red de contacto con veterinarios locales para consultas y asesoramiento profesional.
Monitor de salud y control de peso	Esta sección ofrece un seguimiento exhaustivo de la salud de su mascota, centralizando datos clave como el peso actual, la dieta actual y las recomendaciones nutricionales personalizadas según su estilo de vida. Se mostrarán todos los alimentos sugeridos para mejorar al máximo el estilo de vida y salud.

4.3. Lean Canvas Model:

Una forma para ver si tus ideas son tangibles y posibles es usar el modelo Lean Canvas¹². De esta forma se elimina lo que realmente no se necesita del documento, haciendo que la obtención de ideas sea más veloz y limpia. Sin tener que llenar documentos que realmente nadie se va a leer y pueden llegar a ser una pérdida de tiempo.

Lean canvas se divide en 9 bloques que pueden ser expandidos o reducidos según las necesidades de cada empresa. Estos bloques nos ayudan a seguir un proceso mental sobre el producto y el propio mercado.

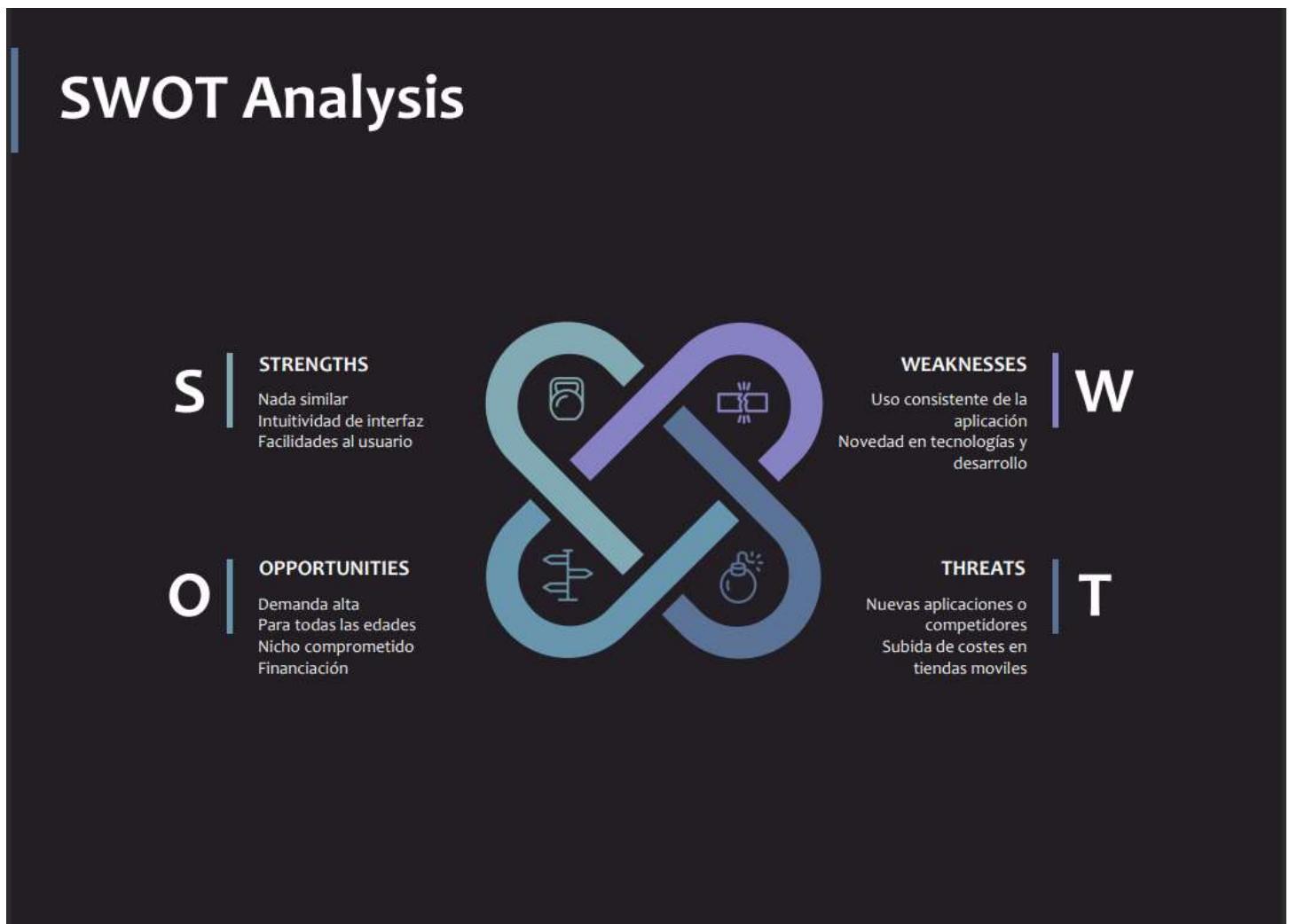
Lean Canvas Model



Problem Hard to find relevant information	Solution Create a technology that could analyze user searches fast	Unique Selling Proposition Quick search results; more relevant content	Unfair Advantage Google's PageRank algorithms	Customer Segments Web users
Existing Alternatives Yahoo, Altavista, Excite	Key Metrics Number of search queries; percentage of users who found what they wanted so fast	High-Level Concept Quick and reliable web searcht	Channels User intents	Early Adopters Stanford graduates
Cost Structure Development, hosting		Revenue Stream Investment, advertising revenues		

4.4. SWOT Analysis:

Conocido como el análisis DAFO¹³ en español nos ayuda a analizar las fortalezas, debilidades, oportunidades y amenazas que rodean a nuestra empresa o proyecto.



4.5. Conclusión:

Con el lean canvas podemos dejar claras cuáles son nuestras necesidades tanto en la parte de desarrollo como en la parte del mercado.

A pesar de que mostramos un gran potencial como idea y proyecto, el cual se encuentra en un mercado en auge que puede llegar a crecer de forma exponencial, tenemos una serie de debilidades y amenazas que pueden complicar el lanzamiento de la aplicación o el mantenimiento de clientes una vez lanzada. Siempre pueden llegar competidores que hagan lo mismo que tú pero sean más atractivos en algo.

Para conseguir que las debilidades no venzan sobre las fortalezas, se deben abordar estas con cuidado, determinación y precisión, dejando así el mínimo espacio posible para que surjan errores y problemas.

5. Tecnologías y herramientas utilizadas

5.1. React Native

Este framework desarrollado por Meta se basa en React.

Actualmente es una herramienta muy útil para el desarrollo de aplicaciones multiplataforma ya que permite emular ambos sistemas operativos.

Además de que permite modularizar bajo Swift o Java, nos proporciona una gran cantidad de librerías y formas de expandir nuestra aplicación hacia horizontes que no se habían contemplado.¹⁵

Por esta razón escogí React Native frente a Ionic o Flutter

5.2. Firebase

A pesar de que el uso al principio fue complicado (debido a sus cambios constantes con otras versiones), llegó a proporcionar una cantidad de herramientas que facilitó el desarrollo y distribución de la aplicación.

Se utilizaron las herramientas *Cloud Firestore*, *Authentication* y *Storage* las cuales proporcionaron una base de datos en tiempo real, una forma de autenticación e inicio de sesión segura y almacenamiento en la nube de archivos que podían serme útiles.¹⁶

Estas han sido las razones por la que me he decantado por *Firebase* en lugar de una base de datos tradicional

5.3. Github

Github es una herramienta que ya la tengo muy conocida y sabida, siempre proporciona un poco de cordura dentro del caos. Varias veces me ha ayudado a recuperar versiones antiguas de código, el cual se sabía que era y que tenía gracias a los Issues y a todos los pull request.¹⁷

Esta ha sido la principal razón por la que me he decantado a usar Github y no por ejemplo, DevOps de Azure.

5.4. Trello

Trello es una herramienta que permite distribuir las cargas y los sprints de trabajo según sean considerados. Usa tarjetas como en el método Kanban y a pesar de tener múltiples competidores que puede o no que sean mejores me decidí por este por que ya lo había utilizado anteriormente y conocía la mayor parte de esta.¹⁸

Podría haber utilizado Atlassian o Notion que también han sido utilizadas y son realmente útiles

5.5. NodeJS

React Native trabaja con NodeJS, este funciona en el servidor y además, en tiempo de ejecución funciona bajo JavaScript. NodeJS es elegido por una gran cantidad de Frameworks por qué permite programar tanto en cliente como en servidor con el mismo lenguaje de programación. A pesar de que se puede utilizar TypeScript también, al ser una opción ya conocida me he decantado por esta.¹⁹

Además cuenta con su propio gestor de paquetes (NPM ) el cual ayuda a gestionar todas las librerías y dependencias de nuestro proyecto.

5.6. Loop11

Esta herramienta nos proporciona una forma de testear un prototipo que les proporcionamos a los usuarios. Gracias a Loop 11 he podido sacar en claro qué necesitaba mi aplicación y cuáles eran sus virtudes. Esta aplicación me proporcionó una cantidad de perspectivas que no podría haber obtenido de ninguna otra forma, desde datos precisos sobre el prototipo, hasta comentarios de los propios usuarios.²⁰

Era de las pocas aplicaciones que conocía que supiese cómo funcionaban, así que fue una opción bastante clara cuando tuve que testear el prototipo.

5.7. JustInMind

El prototipo fue creado con JustInMind. A pesar de conocer otras aplicaciones que realmente ofrecen más funcionalidades y son más completas (como por ejemplo Figma) me decidí a utilizar JustInMind por que la Universidad de Alicante me proporcionó una licencia para una de las asignaturas del año previo, aproveché que ya conocía la aplicación y sus funcionalidades para crear mi prototipo con este. A pesar de que no habría podido modificar el prototipo ya

que no tenía licencia, los resultados del testing no dieron motivo a modificarlo pero si a añadir ciertas funcionalidades.²¹

5.8. VSCode

VisualStudioCode es un IDE que durante mucho tiempo he estado utilizándolo. Desde antes y durante mis años en la universidad ha sido el que más opciones de personalización me ha otorgado y el que más extensiones tiene para poder mejorar en gran medida la forma de programar y la rapidez y efectividad. Además para hacer varios scripts que más tarde comentaré me resulta realmente fácil y rápido de usar.²²

5.9. XCode

Para la emulación de mi aplicación en IOS he tenido que utilizar XCode. No he tenido otra opción ya que es la única forma de ejecutar de forma casi nativa la aplicación en IOS. A pesar de que no he programado en XCode ya que utilizaba VSCode me ha permitido con mucha facilidad probar y testear mi aplicación.²³

5.10. Diagrams

Con diagrams he podido crear todo tipo de diagramas que han servido para ilustrar mis ideas de mejor forma sin tener que complicarme demasiado la vida.²⁴

5.11. OpenAI ChatGPT API

Por último ChatGPT, gracias a las tecnologías ofrecidas por OpenAI he podido crear un chatbot con un entrenamiento personalizado. Más adelante se indaga sobre el desarrollo de este Chatbot.

La idea principal de crear un chatbot sin tener que recurrir a la programación de este era utilizar el API de OpenAI. De esta manera podría acceder al famoso ChatGPT que tan admirado es y entrenarlo para mi aplicación.

A pesar de que no se si acabará en la versión final de cara al público (ya que el gasto por usuario puede llegar a ser muy elevado si no está correctamente controlado) me ha servido para aprender sobre el entrenamiento de una inteligencia artificial que actualmente es de las más reconocidas.

Gracias a su gran comunidad he podido indagar en la propia creación de ChatGPT, su entrenamiento como LLM y por supuesto su utilización para fines personales. Esta idea la he considerado fantástica debido a lo que me ha llegado a proporcionar.²⁵

6. Planificación temporal y desarrollo

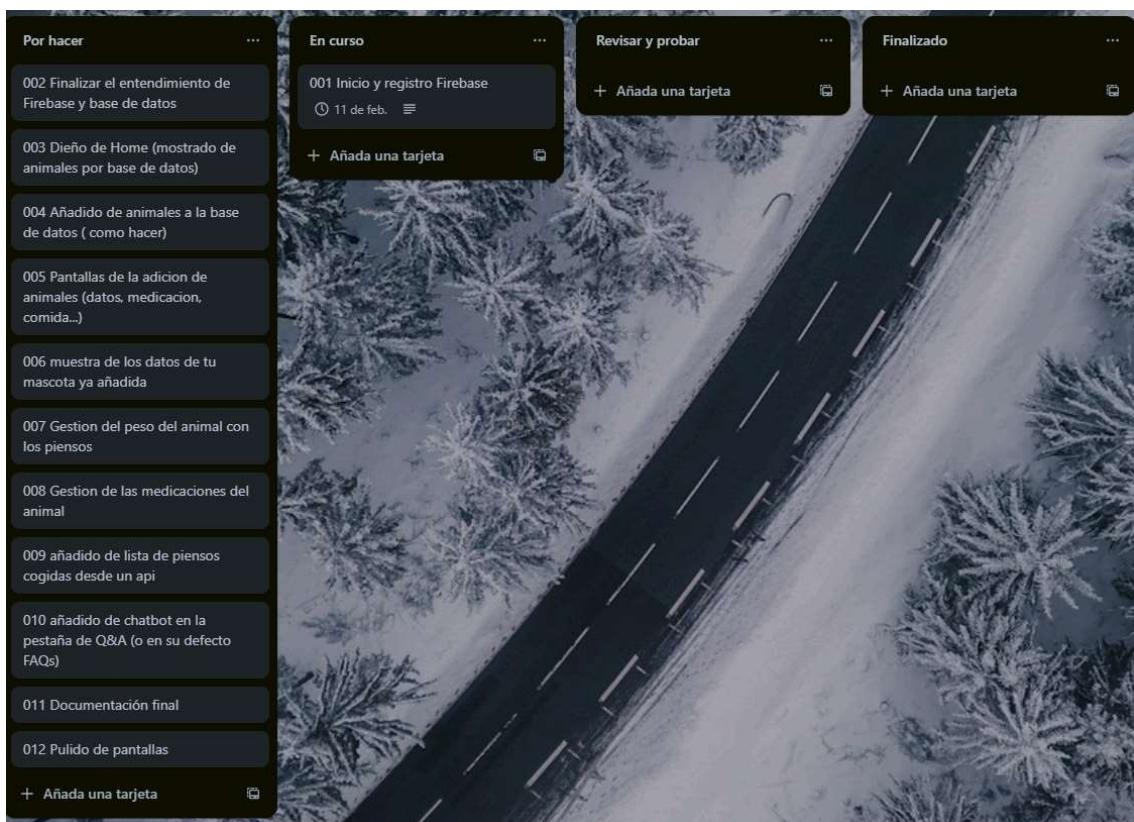
Para el desarrollo de un proyecto de semejantes magnitudes, se tiene que tomar tiempo no solo en el desarrollo del propio proyecto sino también en la planificación de este y en cómo se debe estructurar.

Es importante tomar en cuenta los tiempos esperados para el desarrollo de una característica y los tiempos que realmente te pueden llegar a tomar.

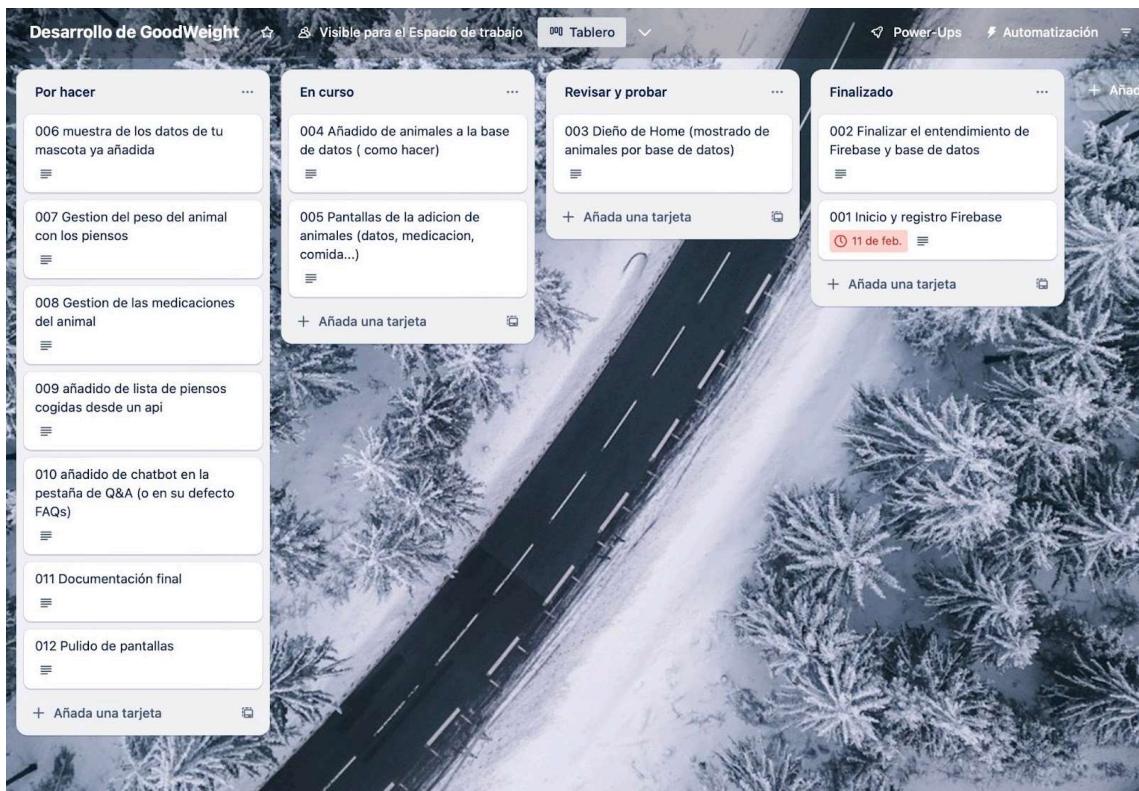
Para ello y como ya he comentado, he utilizado trello y github.

Para ello saqué en claro basándose en la siguiente tabla de tiempos (esperada) por meses

Como resultado obtuve 12 historias de usuario, las cuales fueron las siguientes y fueron evolucionando según iba completando



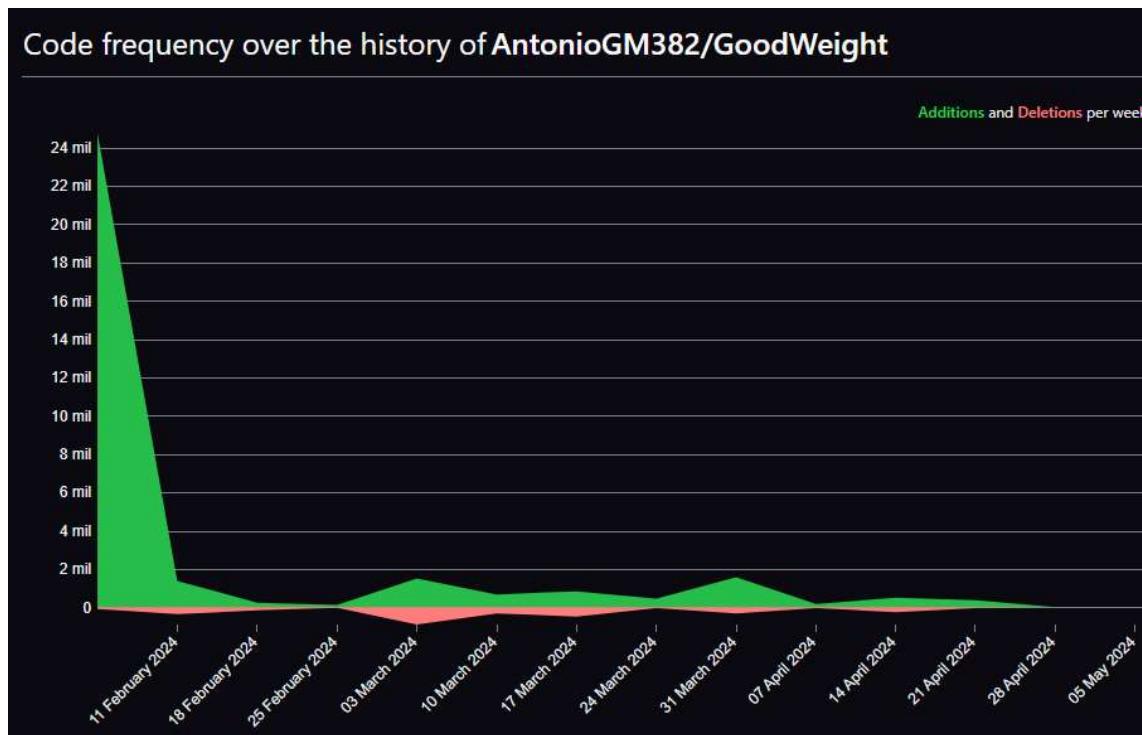
Ya en la fase de desarrollo, fui determinando las tareas por semanas.



Realizar un seguimiento lo más detallado posible es clave en el desarrollo de aplicaciones o software en general, más aún si te encuentras solo en el proyecto, ya que debes seguir unas pautas, y debes tener claro cuánto llevas, cuánto te queda y cuándo acabarás aproximadamente. Por eso también, salían nuevas historias de Usuario con cosas que no llegan a convencer, aún después de haber sido hechas.

6.1. Evolución de commits

A continuación se muestra tanto la evolución de commits como la contribución total en el proyecto:



Se puede observar que hasta finales de enero no se comenzó a subir contenido a Github, ya que hasta entonces, entre la creación de varios proyectos para aprender como funcionaba realmente ReactNative y la creación del prototipo con su división de pantallas y estudio de usabilidad, fue en enero cuando comencé realmente la programación de *GoodWeight*.

6.2. Historias de Usuario

A continuación se mostrarán todas las historias del usuario, las cuales irían cambiando según el avance del proyecto

Título	Historia de usuario	Detalles	COS (Condiciones de satisfacción)
001 Inicio y registro Firebase	<p>Crear con firebase un formulario de registro y de login para la aplicación.</p> <p>Autenticación con servicios externos (Google, Microsoft ...)</p> <p>Autenticación mail y contraseña</p>	<p>Página principal de comenzar → luego manda a login y abajo tenemos registrar</p> <p>Dentro de login se puede volver a la pantalla de registrar y además recuperar la contraseña</p>	<p>estar registrado → sino registrar con mail y contraseña y nombre válido</p> <p>una vez registrado iniciar sesión con los datos correctos</p>
002 Finalizar el entendimiento de Firebase y base de datos	<p>Crear consultas a la base de datos satisfactoriamente</p> <p>Conectar usuarios que se registren con la base de datos</p>	<p>Cuando se registre un usuario conectarlo con la base de datos para que a la hora de crear una mascota se pueda conectar con el usuario</p>	<p>Crear mascota/registrar un usuario</p>

		también	
003 Diseño de Home (mostrado de animales por base de datos)	Crear un home con las mascotas del usuario Recuperar mascotas de la base de datos y toda su información	Recuperar todos los datos de la mascota del usuario para mostrarlo por pantalla Posibilidad de crear una mascota con él + (funcionalidad hecha más adelante)	estar registrado Crear una mascota o tener una mascota
004 Añadido de animales a la base de datos (como hacer)	Crear mascotas en la base de datos desde la pantalla de creación	Añadir datos básicos (los cuales se tienen que llenar todos) Añadir datos sobre la medicación (si es que posee alguna medicación) Añadir datos sobre la alimentación para poder hacer un correcto tracking del estilo de vida	Estar registrado y logueado y querer añadir una mascota
005 Pantallas de adición de animales (datos, medicación, comida...)	pantallas individuales de la adición de mascota Datos Medicación Comida	descritos previamente	Rellenar todos los datos del animal Foto y medicación opcional
006 muestra de los datos de tu mascota ya añadida	Una vez se haya añadido la mascota que se muestre desde el	Debe mostrar también detalles sobre la mascota como id /genero	Tener una mascota creada

	<p>home</p> <p>Al clicar dentro que te muestre Un resumen de la mascota</p>	etc etc Debe mostrar los dos botones de medicación y peso que serán implementados más adelante	
007 Gestión del peso del animal con los piensos	<p>Una vez se haya añadido la mascota que se muestre desde el home</p> <p>Al clicar dentro que te muestre Un resumen de la mascota → En este caso al clickar en peso poder ver más detalles del peso del animal</p>	Debe mostrar una gráfica del peso por meses, el peso actual, para meter nuevo peso, el pienso actual, el pienso recomendado, el estilo de vida y un resumen total de la salud del perro Además debe mostrar la cantidad de pienso recomendada y para buscar más piensos y consejos	Tener una mascota creada
008 Gestión de las medicaciones del animal	<p>Una vez se haya añadido la mascota que se muestre desde el home</p> <p>Al clicar dentro que te muestre Un resumen de la mascota → en este caso que se muestre la medicación actual y pasada del animal</p>	Mostrar todas las medicaciones actuales del animal Mostrar con un botón de historial las medicaciones pasadas Mostrar la dosis, si quiere recordatorio, el tiempo restante que le queda y la posibilidad de añadir otro medicamento	Tener una mascota creada

009 añadido de lista de piensos cogidas desde un api	<p>Mostrar los piensos actuales en todas las tiendas de españa por ejemplo</p> <p>Se deben poder añadir a favoritos los piensos</p>	<p>Debe mostrar todos los piensos además de un buscador con filtro por animal, tamaño etc etc</p>	Estar logeado
010 añadido de chatbot en la pestaña de Q&A (o en su defecto FAQs)	<p>Chatbot ya creado → ver como implementar en el react server</p> <p>entrenar para específicamente cuidado de animales</p> <p>Si no es factible por precios → FAQs</p>	<p>Debe responder preguntas que simplemente estén relacionadas con la alimentación/cuidado de las mascotas</p> <p>Se deben controlar los precios para que sea viable en caso de que haya muchos usuarios</p> <p>Interfaz con react native bonita sencilla de hacer</p>	Estar logueado
011 Documentación final	DOCUMENTAR	DOCUMENTAR	DOCUMENTAR
012 Pulido de pantallas	DISEÑAR	DISEÑAR	DISEÑAR

6.3. Pull Requests:

En caso de que el Issue tuviese que ver con el código subido actualmente se creaba un “*Pull Request*” el cual contenía el código recientemente creado y subido además del commit para así llevar un cierto orden. Los “*Pull Request*” venían dados a partir de los issues creados al comienzo.

A screenshot of a GitHub pull request list. At the top, it shows 6 Open and 4 Closed pull requests. The first three are listed:

- 010 añadido de chatbot en la pestaña de Q&A (o en su defecto FAQs) #11 opened on Mar 14 by AntonioGM382
- 009 añadido de lista de piensos cogidas desde un api #10 opened on Mar 14 by AntonioGM382
- 008 Gestión de las medicaciones del animal #9 opened on Mar 14 by AntonioGM382

A screenshot of a GitHub pull request list. At the top, it shows 0 Open and 1 Closed pull request. The one listed is:

- 007 Gestión del peso del animal con los piensos #8 opened on Mar 14 by AntonioGM382

A screenshot of a GitHub pull request list. At the top, it shows 0 Open and 1 Closed pull request. The closed pull request is listed:

- Guarda en BD Usuarios junto con authentication #3 by AntonioGM382 was merged on Feb 13 ➔ Inicio de aplicaci...

006 Mostrado de datos de la mascota ya añadida #7

 Closed AntonioGM382 opened this issue on Mar 14 · 1 comment



AntonioGM382 commented on Mar 14

Historia de Usuario

Una vez se haya añadido la mascota que se muestre desde el home

Al clickar dentro que te muestre Un resumen de la mascota

Detalles

Debe mostrar también detalles sobre la mascota como id /genero etc etc

Debe mostrar los dos botones de medicacion y peso que serán implementados más adelante

Condiciones de satisfaccion

Tener una mascota creada



AntonioGM382 added the [006 Mostrado de datos de la mascota ya añadida](#) label on Mar 14



AntonioGM382 self-assigned this on Mar 14



AntonioGM382 commented on Mar 14

Muestra los animales: [f5f9cbb](#)

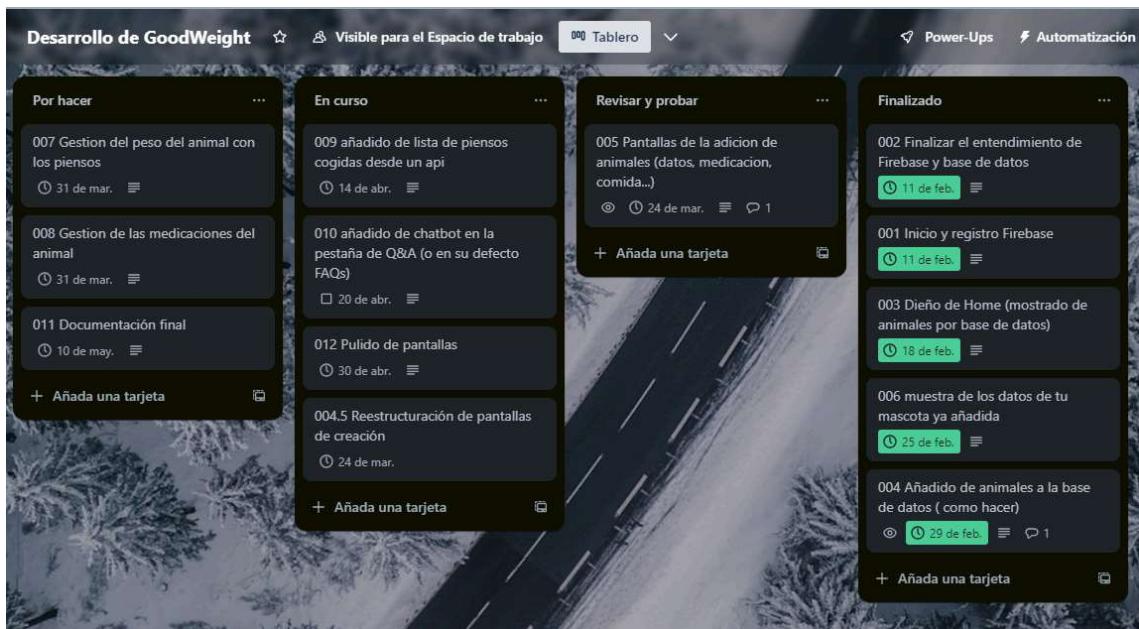
Animal en específico: [f5f9cbb](#)

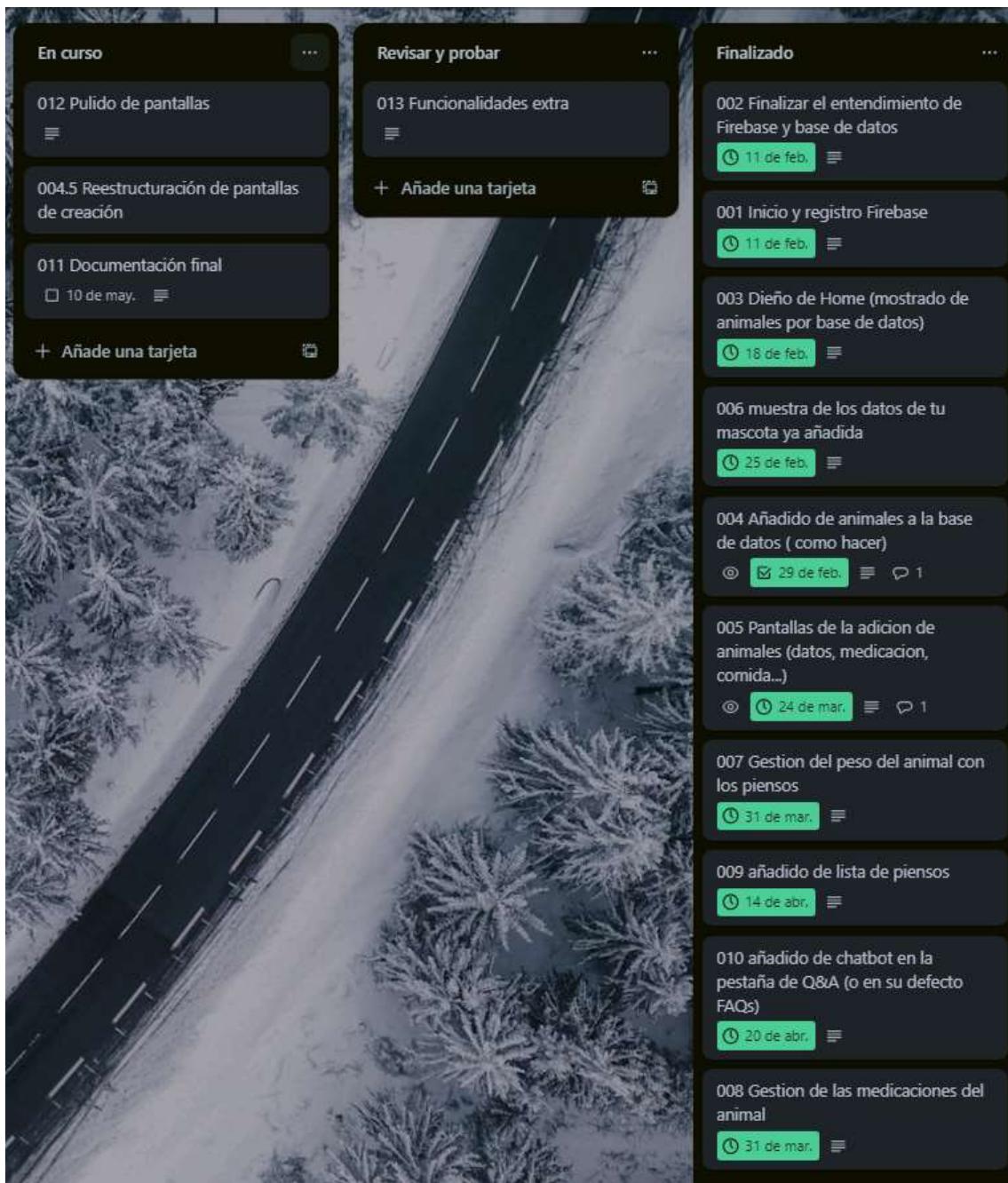


AntonioGM382 closed this as [completed](#) on Mar 14

6.4. Finalización del Trello:

El trello se iba completando y se iban cambiando fechas según se acababan cosas, ya que era más sencillo ir saltando entre tareas si no conseguía avanzar, que estar trabajando en la misma tarea durante mucho tiempo cuando ese microproyecto puede dejarse en barbecho y entrar con otro desde el inicio donde se pudiese avanzar más.





7. Análisis en profundidad

7.1. Requisitos funcionales

Los requisitos funcionales de una aplicación son claves para el desarrollo final de esta.

Se miraron varios requisitos que se introdujeron en las historias de usuario para poder conocer realmente si era posible y cuales eran las restricciones que se deberían de imponer para el uso de ese requisito.

7.1.1. Usuario no registrado

Para el usuario no registrado los requisitos que tendría serían básicos ya que el plan es que no se pueda entrar a la plataforma a no ser que exista un usuario donde añadir los animales.

Requisito Funcional	Nombre	Descripción
7.1.1.1	Registrar	Crear un usuario en la base de datos

7.1.2. Usuario Registrado -> Unico usuario

Requisito Funcional	Nombre	Descripción
7.1.2.1	Iniciar Sesión	Loguearse en la cuenta mediante correo y contraseña
7.1.2.2	Cerrar Sesión	Salirse de la cuenta actual
7.1.2.3	Eliminar cuenta	Salirse de la sesión y borrar la cuenta y todos los datos pertenecientes a esta, animales y los datos pertenecientes a estos
7.1.2.4	Restablecer contraseña	En caso de olvidar usuario o contraseña, poder restablecer las credenciales de inicio de sesión
7.1.2.5	Creación del animal	Posibilidad de añadir un animal con sus datos, alimentación, medicación e historial medico y vacunaciones
7.1.2.6	Edición del animal	Posibilidad de cambiar cualquier dato del animal.
7.1.2.7	Eliminación del animal	Posibilidad de eliminar cualquier animal
7.1.2.8	Consultar animal	Consultar todos los datos actuales del animal
7.1.2.9	Añadir peso	Añadir peso para las mediciones de este
7.1.2.10	Añadir medicación	Añadir medicación y recordatorios a un animal dado

7.1.2.11	Añadir visita veterinario	Añadir la fecha en la que un animal fue llevado al veterinario
7.1.2.12	Añadir vacunación	Añadir la vacunación de un animal de una vacuna dada en una cierta fecha
7.1.2.13	ChatBot	Posibilidad de preguntar al chatbot
7.1.2.14	Ver piensos	Consultar piensos disponibles

7.2. Requisitos no funcionales

Requisito No Funcional	Nombre	Descripción
7.2.1	Encriptación de datos	Todos los datos (desde usuario hasta los propios del animal) estarán almacenados en firebase y las contraseñas encriptadas
7.2.2	Varios SO	El hecho de utilizar React native facilita el uso de la aplicación tanto en sistemas Android como en iOS
7.2.3	Velocidad	React Native proporciona gran velocidad de cargado si no se satura con animaciones, e incluso llega a ser mayor en iOS, que es el enfoque principal.
7.2.4	Múltiples formas de registrarse	Gracias a Firebase podemos implementar múltiples formas de crear una cuenta en la aplicación, desde Google, Facebook, X (Twitter) o hasta Apple.
7.2.3	Resiliencia	Gracias al control de errores cuando sucede un error en el sistema o en la

		base de datos las recuperación es casi inmediata
7.2.4	Diseño de cara al SO	Gracias a react se puede crear un diseño que cuadre más con el propio sistema operativo
7.2.5	Mantenimiento y proyecto de futuro	Ya se han pensado decenas de nuevas funcionalidades que quieren introducirse en la aplicación en un futuro, proporcionando más utilidades y contenidos al usuario

8. Testeo de la interfaz de usuario

Para obtener una buena interfaz, que sea llevable, intuitiva y sencilla, no solo vale con crear lo que uno piense que es lo mejor. Se debe testear si está interfaz y funcionalidades pensadas para la aplicación realmente funcionan. Por ello a continuación, se indicará el proceso seguido para el testeo de esta.

8.1. User Testing Escenarios:

Para el testeo de la aplicación se crearon 5 escenarios que más tarde servirían para crear las tareas que realizarán los usuarios.

- **Escenario A)**

- Tarea: Consultar datos de su perro toby
- Persona: Juan Martínez
- Descripción: Juan está en el veterinario ahora mismo y no recuerda con certeza si su perro, Toby, estaba vacunado, la edad exacta o incluso si estaba esterilizado. Juan se mete en la aplicación y gracias al almacenamiento de datos sobre su mascota en la aplicación puede dar todos los datos sobre Toby.

- **Escenario B)**

- Tarea: Añadir medicación a su mascota
- Persona: Luis Segura
- Descripción: Luis acaba de recibir el tratamiento que le tiene que dar a su perro el cual sufre una gastroenteritis, añade la medicación en el apartado de Toby con la dosis y a la hora que se lo tiene que dar para que así quede

registrado en la aplicación para la próxima visita al veterinario y además la aplicación le recordará la hora para dárselo a Toby

- **Escenario C)**

- Tarea: Controlar el peso de toby
- Persona: María Luisa Pérez
- Descripción: María ha empezado a cuidar de su perro el cual es un pug y tiene sobrepeso. Toby está registrado en la aplicación ya pero quiere ver como evoluciona el peso de este. Lo primero que hace es registrar el peso de su mascota, a continuación introduce la frecuencia de ejercicio de Toby y ve el pienso recomendado para dárselo a Toby para que adelgace.

- **Escenario D)**

- Tarea: Añadir una mascota
- Persona: Josefa García
- Descripción: Josefa adoptó un nuevo perrete hace una semana y para llevar todo al día sobre él lo añade a la aplicación junto a toda la información que ya sabe para que no se le olvide. De esta forma puede llevar un control estricto de su nuevo amigo.

- **Escenario E)**

- Tarea: Seleccionar nuevo pienso
- Persona: Eduardo Sempere
- Descripción: Eduardo lleva controlando el peso de su perrita durante ya 3 meses, pero no mejora así que decide comprar un nuevo pienso recomendado por la aplicación, el cual esta basado en la raza, tamaño, sexo y por su puesto si está esterilizado o no. Un mes después

ve una gran mejoría en su acompañante, como ha recuperado ese estado de salud que tenía previamente.

8.2. Tareas asociadas a los escenarios:

A partir de estos escenarios, sacamos las tareas siguientes:

- Tarea: Consultar datos de su perro toby
 - Para ello después de dar al botón de comenzar, tiene que registrarse/loguearse y ver sus mascotas, una vez hecho accede a Toby y aparecerán los datos de su mascota, como puede ser la raza, peso, tamaño de la mascota etc
- Tarea: Añadir medicación a su mascota
 - Para ello después de dar al botón de comenzar, tiene que registrarse/loguearse y ver sus mascotas, una vez hecho accede a Toby y aparecerán dos botones, "peso" y "medicación", al pinchar en medicación podrá ver los medicamentos de su mascota y justo debajo podrá "Añadir nuevo medicamento"
- Tarea: Controlar el peso de toby
 - Para ello después de dar al botón de comenzar, tiene que registrarse/loguearse y ver sus mascotas, una vez hecho accede a Toby y aparecerá mucha información en la pantalla de Toby, tendrá que acceder al apartado de medicaciones, al entrar podrá ver los medicamentos de su mascota y podrá añadir el medicamento que quiera para su mascota.
- Tarea: Añadir una mascota

- Para ello después de dar al botón de comenzar, tiene que registrarse/loguearse y ver sus mascotas, una vez hecho accede al inicio de sus mascotas tendrá que añadir una nueva mascota. Ahí podrá poner una foto de su mascota, añadir los datos relevantes, la medicación en caso de actualmente estar medicado y la alimentación que lleva su nuevo compañero, ahí podrá añadir el tipo de dieta, su pienso y además su ejercicio físico. Una vez hecho esto guardará los datos de este.
- Tarea: Seleccionar nuevo pienso
 - Para ello después de dar al botón de comenzar, tiene que registrarse/loguearse y ver sus mascotas, una vez hecho accede a Toby y tendrá que acceder a peso como ya ha hecho previamente. Una vez ahí tendrá que ver más piensos que puedan adaptarse a su mascota y encontrar el ideal.

8.3. Formularios de Evaluación e instrumentos usados

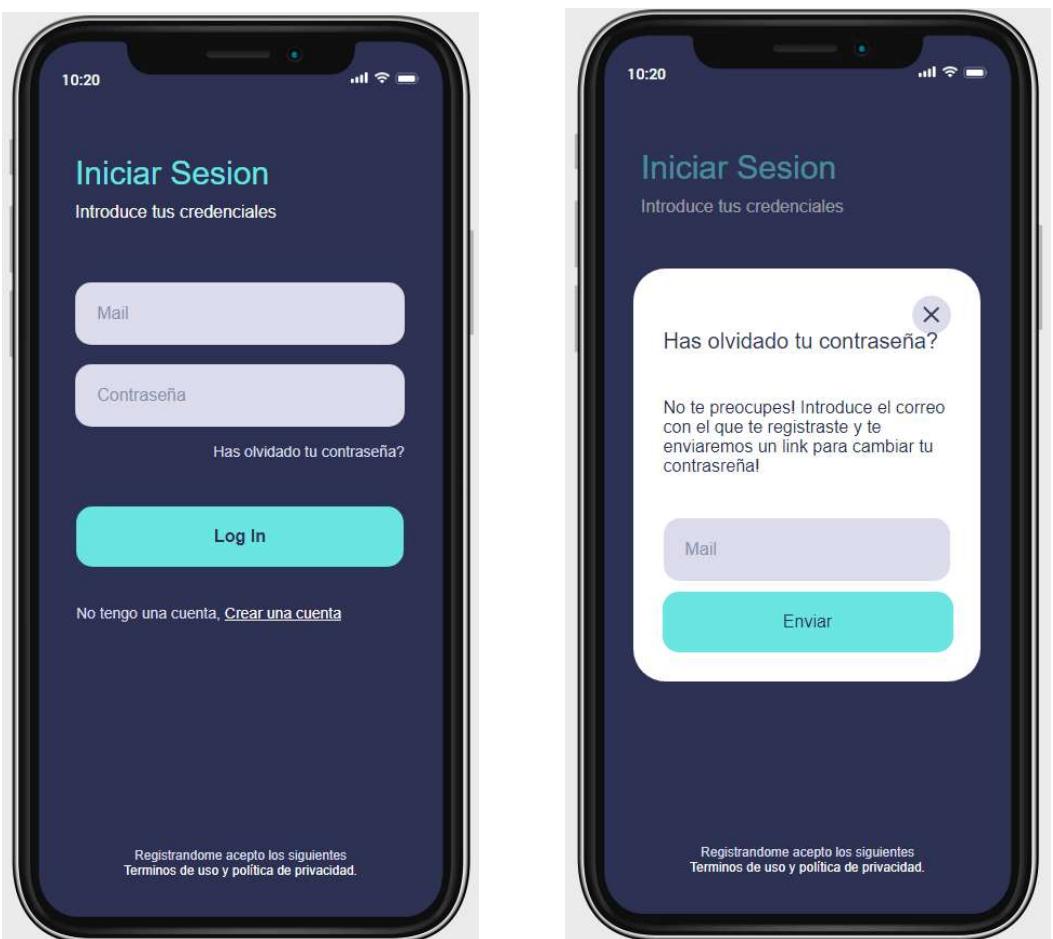
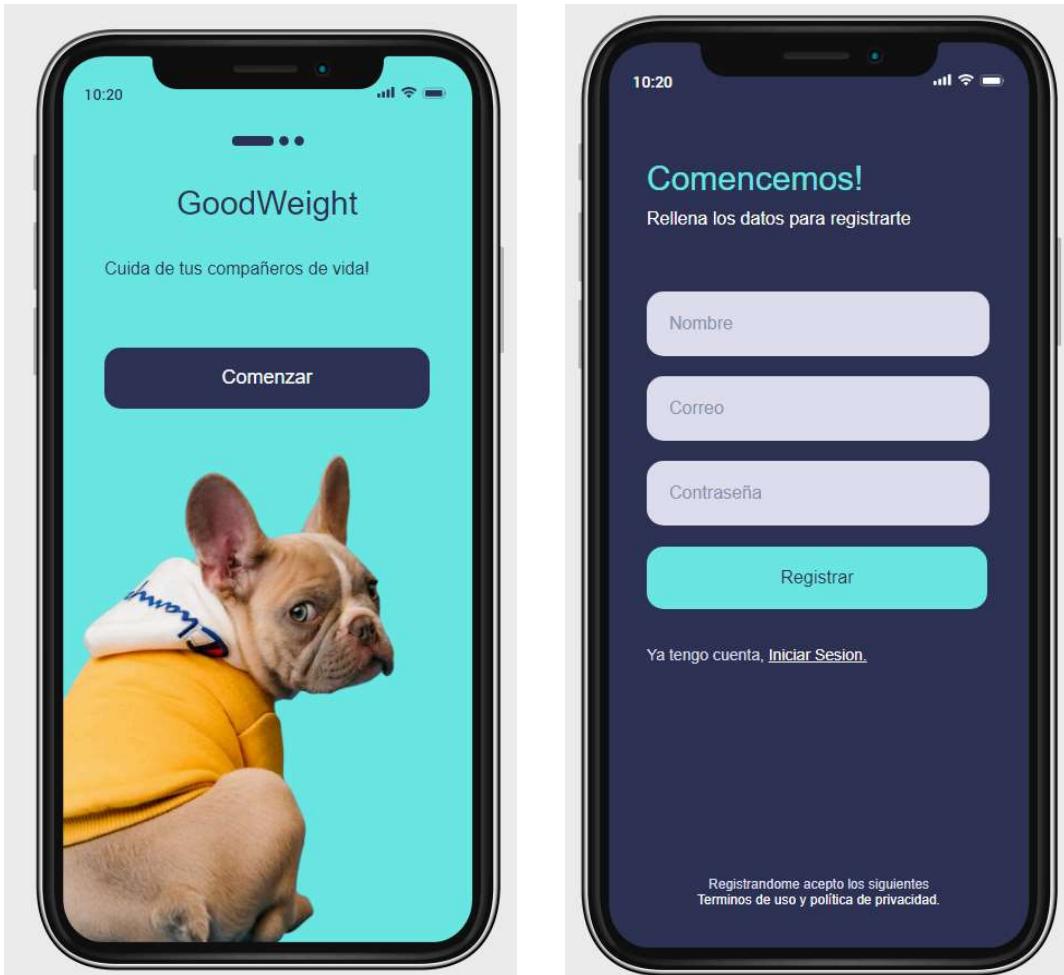
Para comprender mejor a los participantes y su familiaridad con aplicaciones de naturaleza similar, se utilizó un breve cuestionario de perfil de personalidad. Este enfoque fue esencial para recabar información contextual sobre los usuarios.

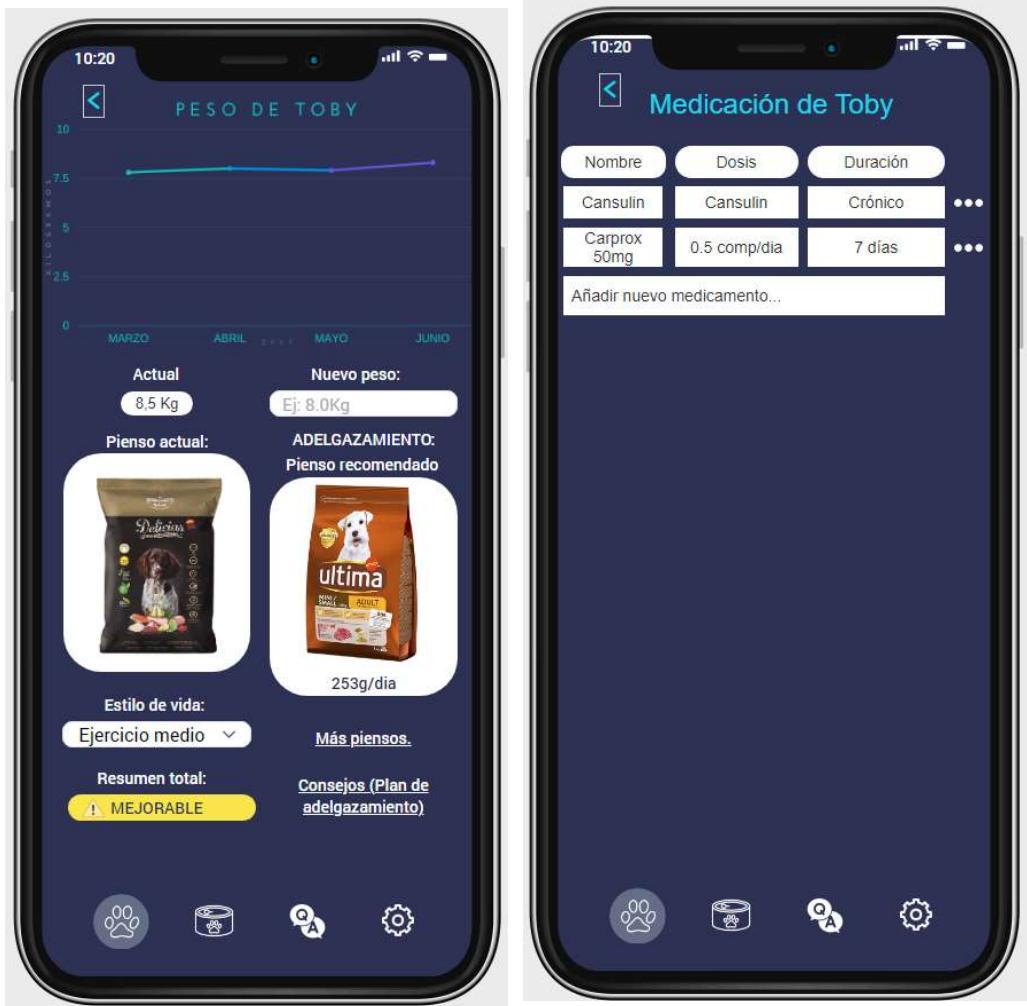
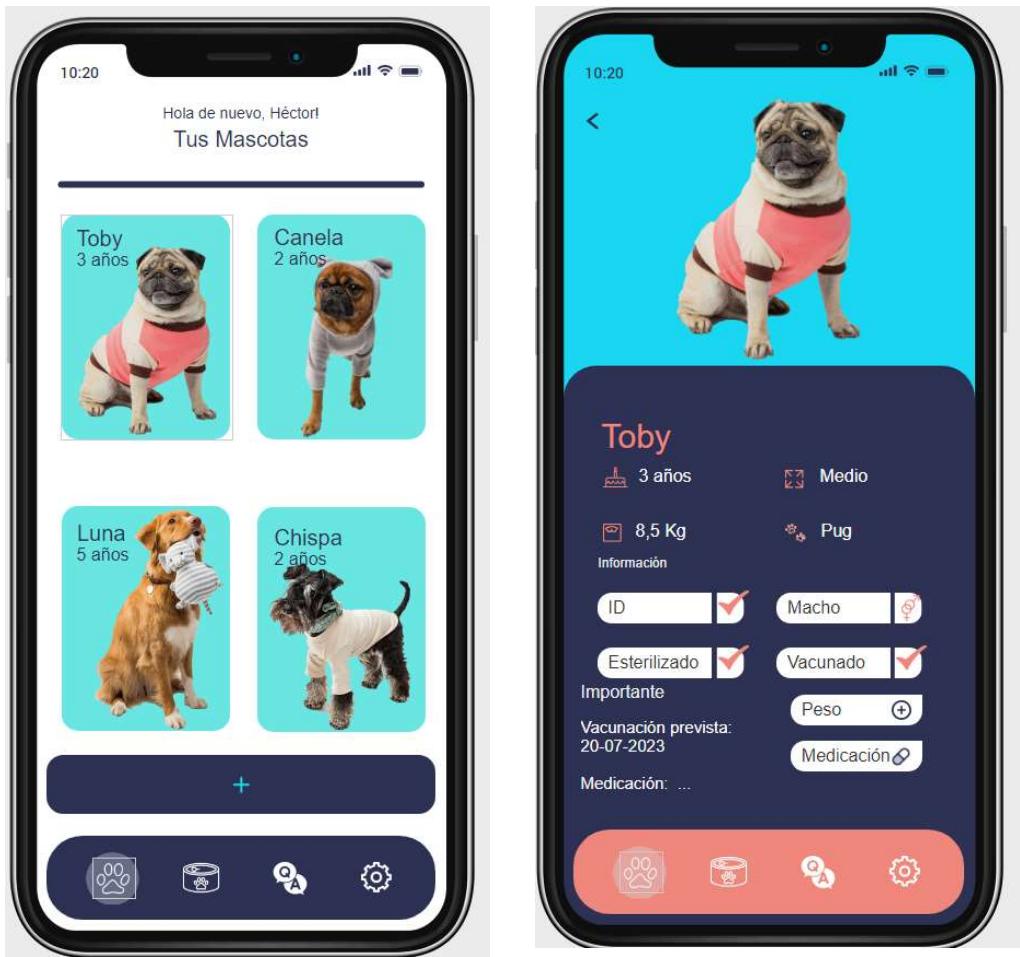
Para la realización de las pruebas de usabilidad se optó por el uso de Loop 11. Esta herramienta facilitó la realización de pruebas interactivas con los usuarios, permitiendo recoger tanto datos cuantitativos como cualitativos acerca de su experiencia con la interfaz.

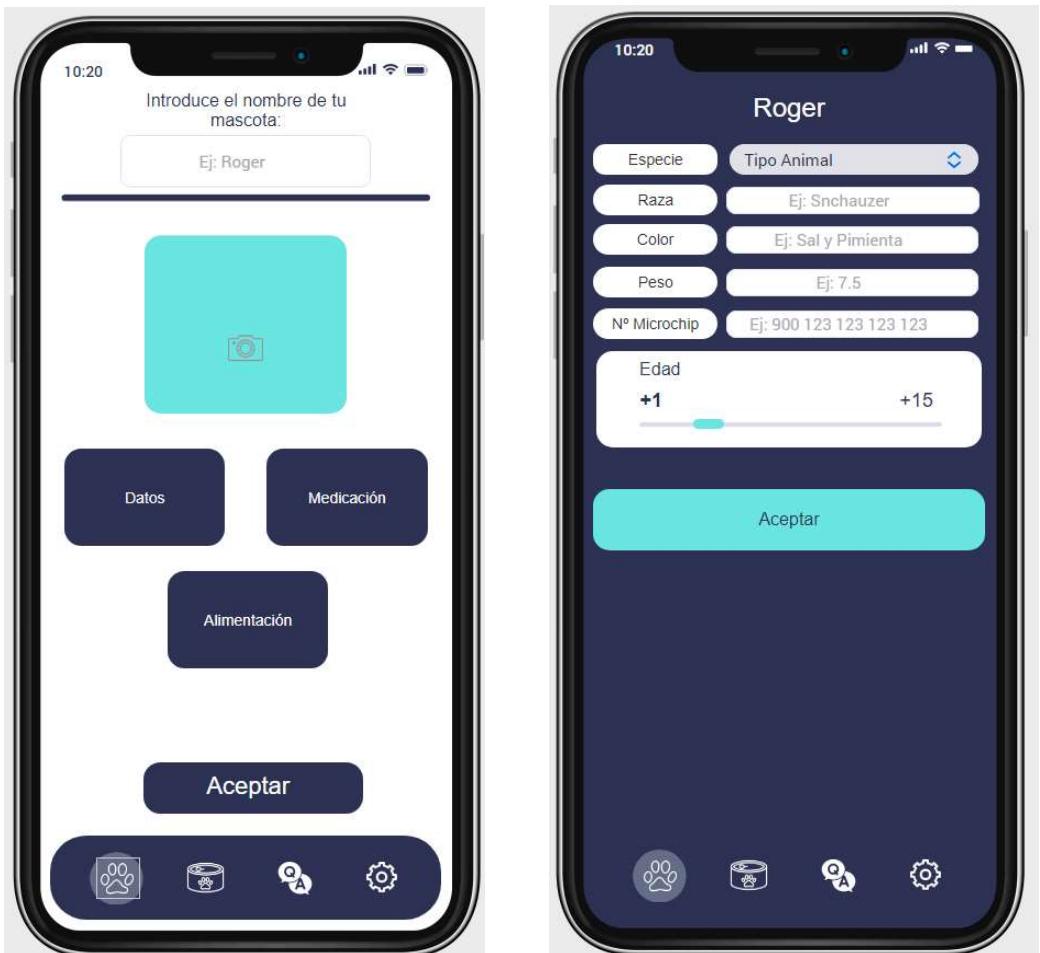
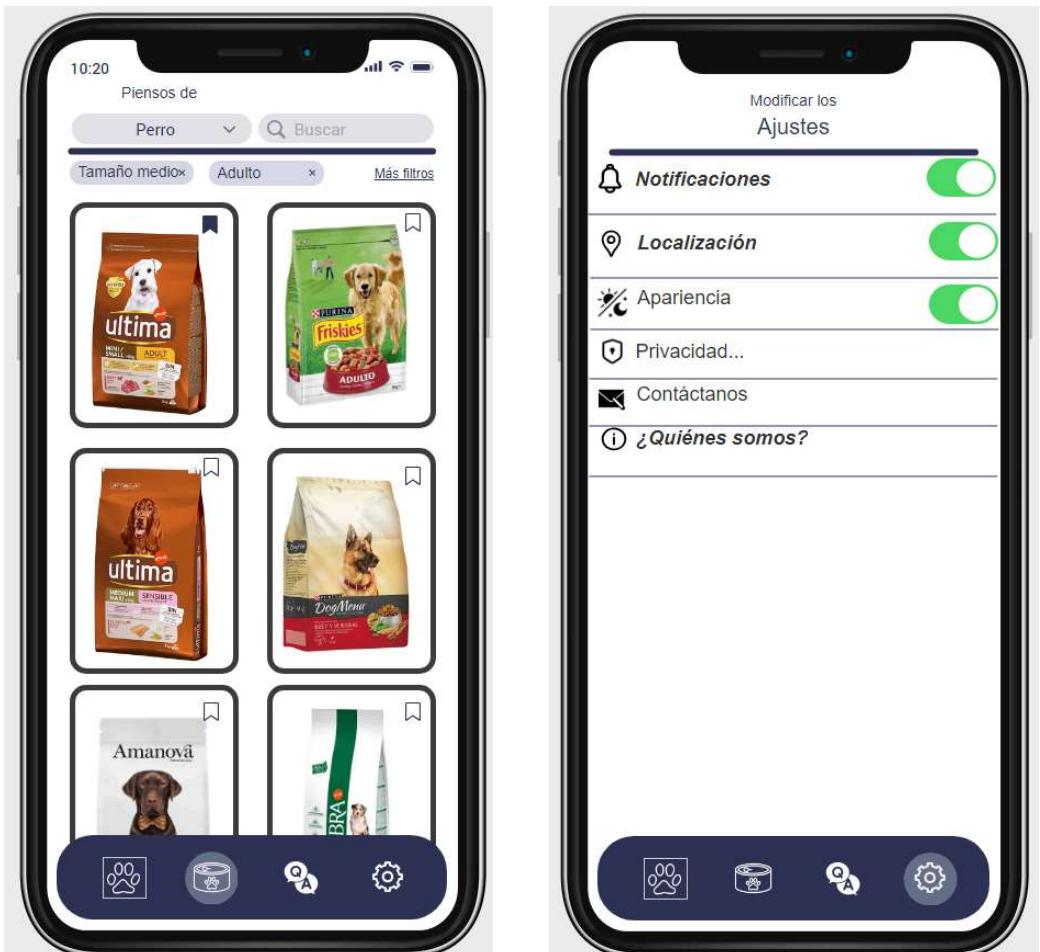
8.4. Criterios de Usabilidad Aplicados

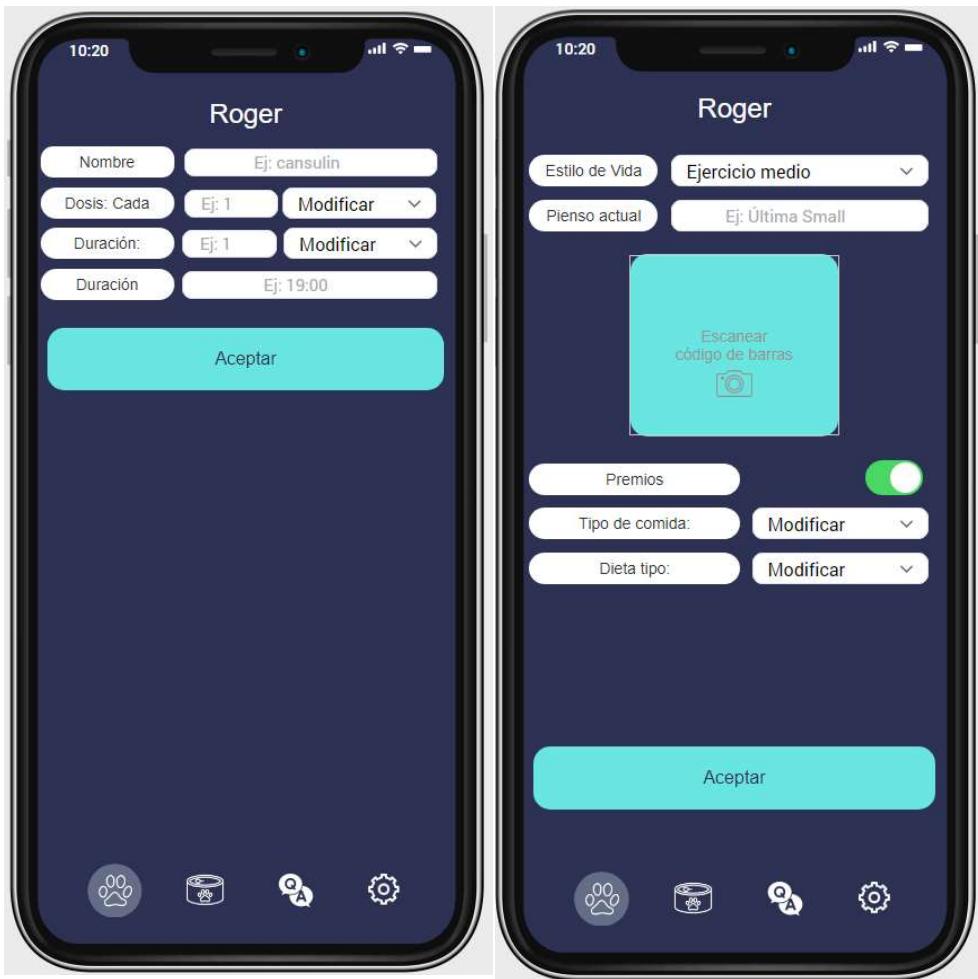
Las métricas de usabilidad se determinaron mediante el análisis de las tareas ejecutadas por los usuarios. Se tuvieron en cuenta factores como el tiempo de ejecución de cada tarea, la frecuencia de errores, y el grado de satisfacción subjetiva reportado por los participantes. Estos indicadores fueron clave para detectar áreas problemáticas y proponer mejoras orientadas a optimizar la usabilidad de la aplicación.

8.5. Pantallas del prototipo testeado:







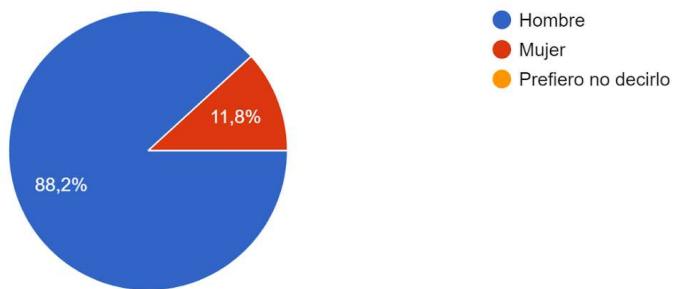


8.6. Resultados y conclusiones del UT sobre la UX:

Los participantes:

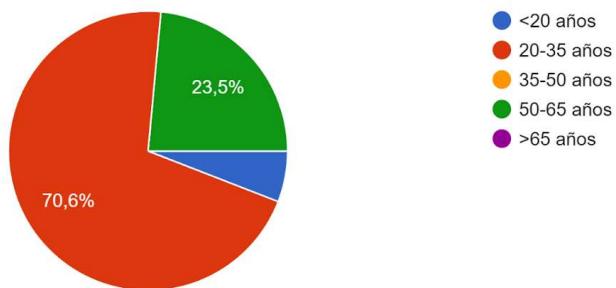
Género

17 respuestas



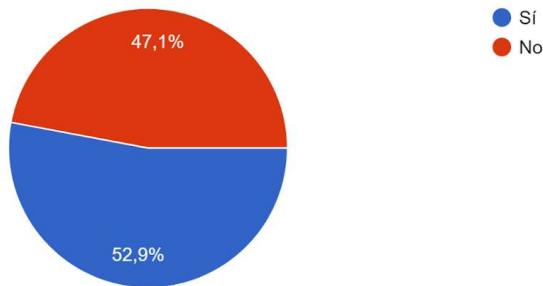
Edad

17 respuestas



Tienes mascotas?

17 respuestas

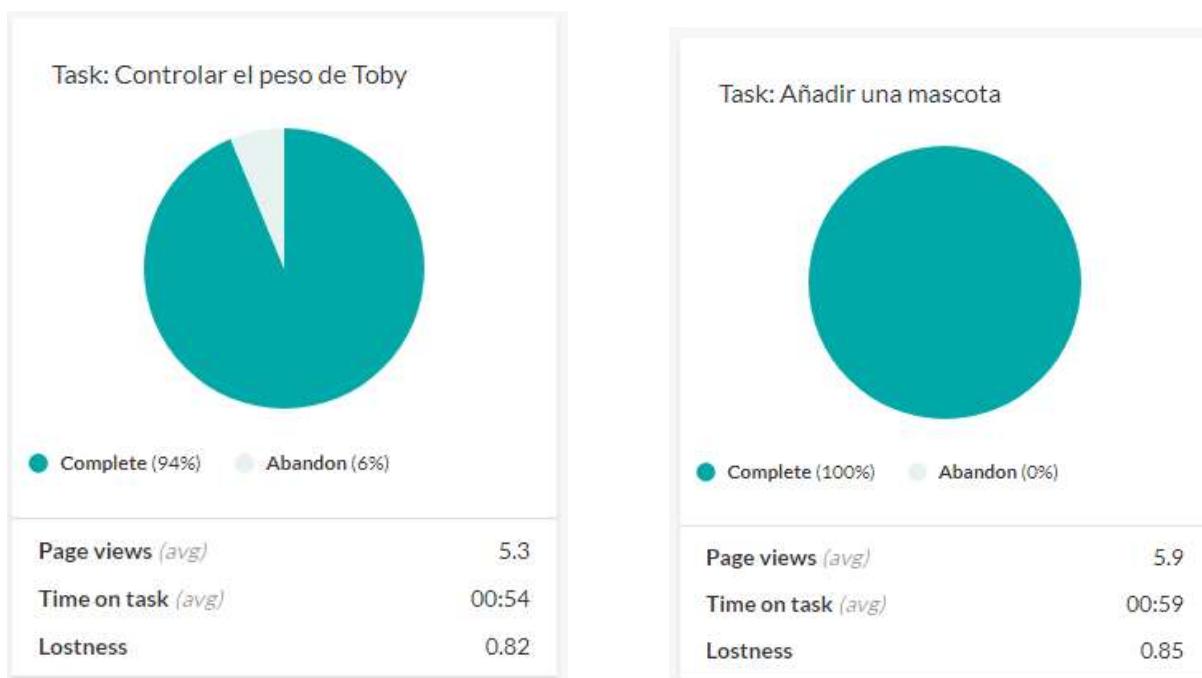


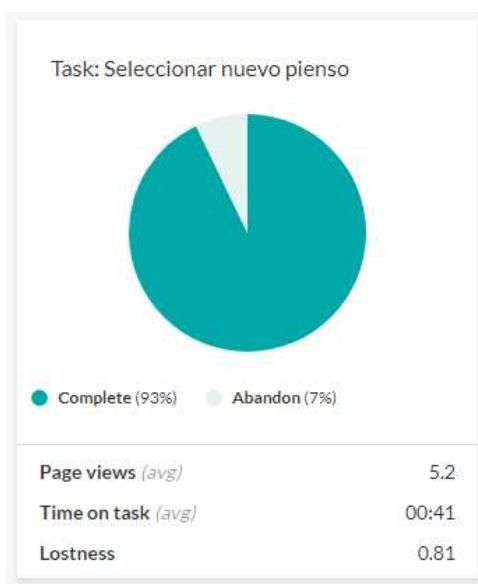
8.7. Resumen inicial

Este informe evalúa la usabilidad del prototipo de la aplicación *GoodWeight*, diseñada para el seguimiento de la salud de mascotas. Se centró en la facilidad de uso, claridad de la interfaz, y eficiencia de las tareas clave. A continuación se muestra la tasa de abandono de los usuarios,



que se puede ver que es mínima.





También se puede observar que el tiempo medio de las tareas propuestas no se excede del minuto y 20 segundos en ninguno de los casos.

Esta revisión inicial de los resultados nos da a entender que el testeo ha sido exitoso y a pesar de que puede haber algún resultado impreciso, el total sabemos que es cercano a la realidad.

8.8. Aspectos Positivos

- Sencillez y Facilidad de Uso: Varios participantes (20, 11, 8, 5, 4, 1) destacaron la sencillez y facilidad de uso de la app.
- Claridad de la Interfaz: Algunos participantes (17, 1) apreciaron la claridad de la interfaz y sus apartados.
- Diseño Amigable: La interfaz amigable fue mencionada por los participantes 9 y 3.
- Intuitividad: La app fue considerada intuitiva por el participante 4.
- Paleta de Colores Atractiva: El participante 2 elogió la paleta de colores de la app.
- Funciones Específicas: El participante 14 valoró positivamente la lista de mascotas con imágenes y la vista de opciones de peso.

8.9. Aspectos Negativos

- Iconografía Confusa: El participante 20 mencionó que la iconografía a veces no es clara.
- Inconsistencia en Apartados: El participante 14 sintió que el apartado de nueva mascota no está al mismo nivel que el resto.

El resto de aspectos negativos fue por la propia aplicación de testing que en ocasiones no funcionaba correctamente.

8.10. Sugerencias de Mejora

- Ajustes de Tamaño en la Interfaz: El participante 20 sugirió hacer algunos elementos más grandes para facilitar su uso.
- Integración con Profesionales de la Salud: El participante 17 propuso que los profesionales de la salud puedan introducir tratamientos en la app.
- Mejoras en la Navegación y Botones: Los participantes 14 y 3 sugieren integrar botones y mejorar la navegación.
- Incrementar la Accesibilidad: El participante 4 sugirió hacer la app más accesible.
- Mejorar el Apartado de Información del Perro: Sugerido por el participante 2.

8.11. Hallazgos Principales

- Navegación Intuitiva: La mayoría de los usuarios encontraron la navegación clara y sencilla.
- Claridad en las Tareas: Las instrucciones para realizar tareas como agregar un medicamento o controlar el peso de la mascota fueron claras para la mayoría de los usuarios.
- Eficiencia en la Realización de Tareas: Las tareas se completaron en un tiempo razonable, aunque algunos usuarios experimentaron dificultades menores que aumentaron el tiempo de tarea.
- Interfaz Visual: La interfaz fue bien recibida, con elementos visuales claros y atractivos.

8.12. Conclusión del Testing

El prototipo de GoodWeight muestra una usabilidad generalmente alta, con una interfaz intuitiva y tareas claras. Los usuarios aprecian la sencillez, facilidad de uso, y la interfaz amigable de GoodWeight. Sin embargo, señalan áreas de mejora como la claridad de la iconografía, la eficiencia de carga, y la funcionalidad en ciertos apartados. Las sugerencias de los usuarios ofrecen valiosas perspectivas para futuras mejoras y desarrollos de la aplicación. Tras una investigación sobre los datos comúnmente obtenidos en tests de Interfaz de usuario (UX) ^{6 7 8} podemos dejar en claro lo siguiente:

8.12.1. Tasa de finalización ^{9 10}:

La puntuación escala de usabilidad del sistema (SUS) fue de 87, lo que indica un alto nivel de usabilidad.

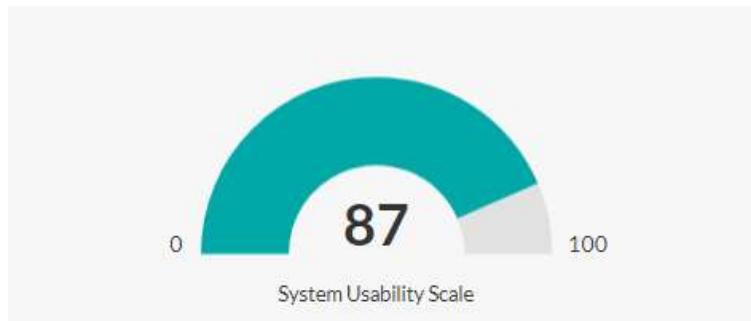
El estándar de la industria para las tasas de finalización de tareas en UX suele ser alto. No es sólo la tasa de finalización lo que importa, sino también la confianza en la finalización. Las altas tasas de finalización de tareas a menudo se asocian con una buena usabilidad, lo que puede reflejarse en las puntuaciones de la Escala de usabilidad del sistema (SUS). Una tasa de finalización del 100% puede asociarse con una puntuación SUS de 80 o más, lo que se considera bueno.

8.12.2. SUS (Escala de usabilidad del sistema) ¹⁰:

La Puntuación Neta del Promotor (NPS) fue de 58, con la mayoría de los participantes siendo promotores.

La puntuación SUS promedio es 68, que se considera el punto medio de usabilidad. Una puntuación SUS de 68 coloca a un sistema en el

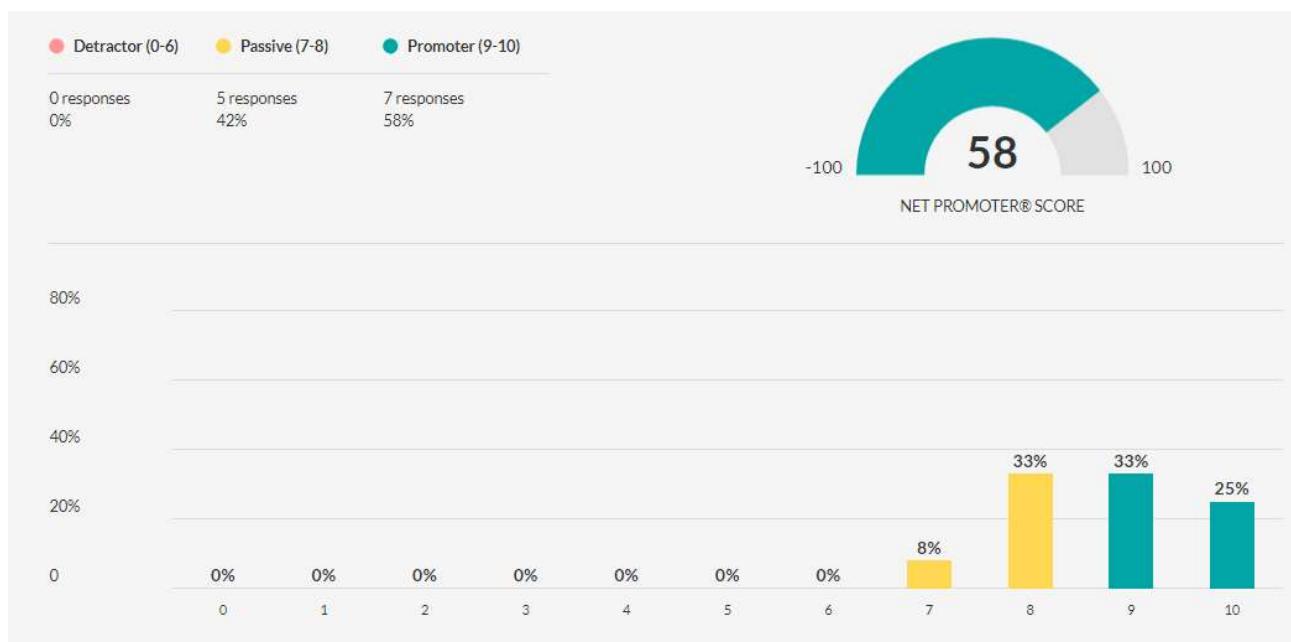
percentil 50, lo que significa que es mejor o igual al 50% de todos los demás sistemas. Las puntuaciones superiores a 80 se consideran buenas, lo que refleja un sistema fácil de usar y utilizable.



8.12.3. NPS (puntuación neta del promotor):

La puntuación media de satisfacción para la aplicación fue de 83 sobre 100.

El Net Promoter Score es una medida de la lealtad del cliente y puede oscilar entre -100 y 100. Si bien no es una medida directa de usabilidad, puede reflejar la experiencia y satisfacción general del usuario. Un NPS promedio puede variar significativamente entre industrias, pero un



NPS positivo (superior a 0) es generalmente bueno y una puntuación superior a 50 se considera excelente.

8.12.4. Pérdida (Lostness)¹¹ :

La pérdida mide la frecuencia con la que los usuarios se pierden al navegar por un sistema. Se calcula utilizando la cantidad de pantallas diferentes visitadas (N), la cantidad total de pantallas visitadas (S) y la cantidad mínima de pantallas que se deben visitar para completar una tarea (R).

$$L = \sqrt{(N/S - 1)^2 + (R/N - 1)^2}$$

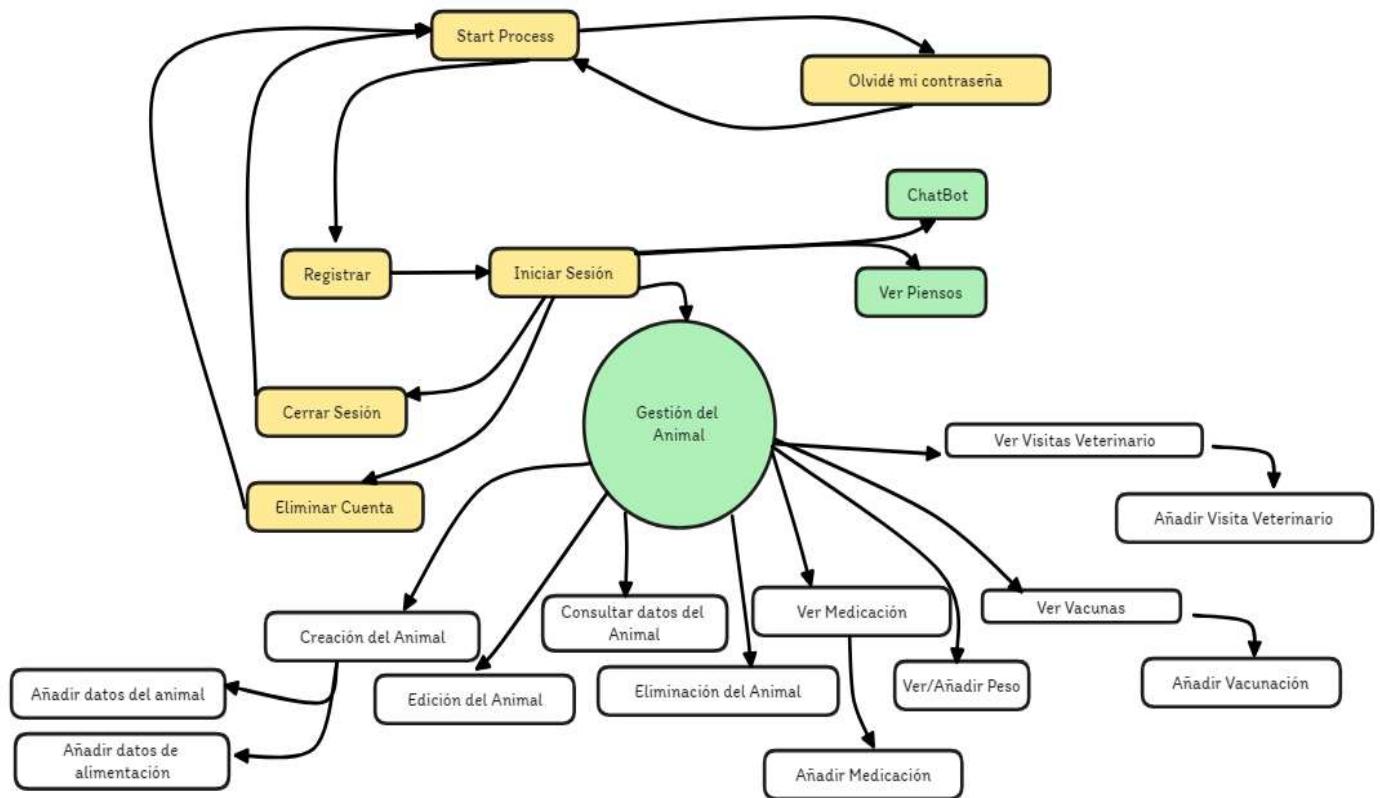
Las puntuaciones más bajas de pérdida son mejores, lo que indica que los usuarios encuentran su camino de manera más eficiente. No existe un número estándar industrial específico para pérdida, pero el objetivo es tener un valor más bajo, lo que indica que los usuarios navegan por el sistema con menos confusión y redundancia.

Dado que la puntuación de pérdida es 0,84, está en el extremo superior, lo que indica que los usuarios están experimentando dificultad para navegar dentro del sistema. No toman los caminos más directos para completar sus tareas y probablemente tengan problemas para encontrar lo que necesitan, esto será solucionado en el prototipo final.

En general, al comparar los resultados totales y finales con los estándares de la industria, parece que son bastante buenos. La tasa de finalización es excelente, la puntuación SUS está muy por encima del promedio, el NPS es positivo y la baja tasa de abandono sugiere que los usuarios no se pierden con frecuencia.

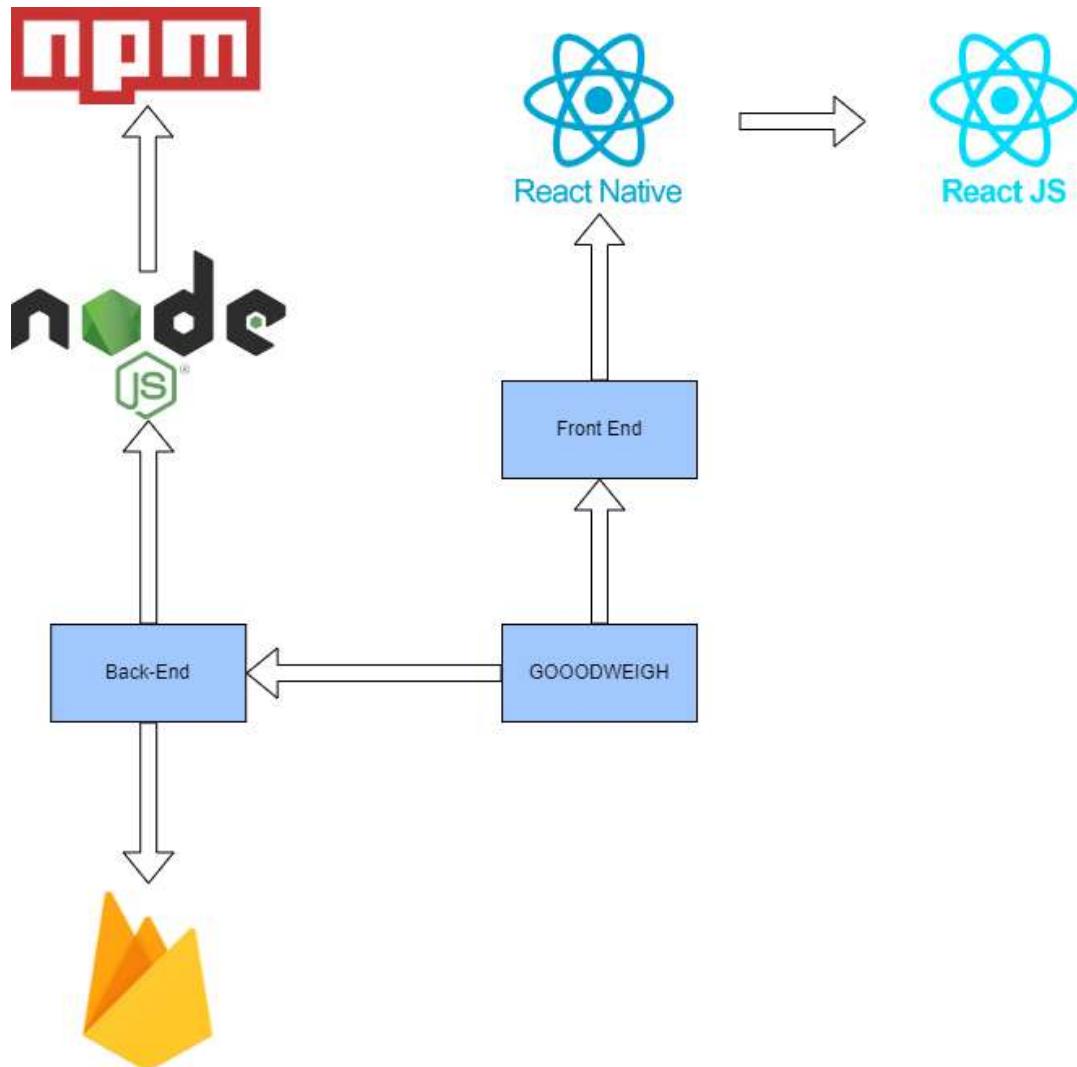
9. Diseño de la aplicación

9.1. Flujo de uso:



En el flujo de uso mostrado podemos observar cómo seguiría el usuario el uso de la aplicación. Podemos ver en amarillo los apartados de gestión de la cuenta y en verde lo que el usuario puede realizar dentro de la aplicación.

9.2. Arquitectura:



En esta figura se muestra cuales son las dependencias de tanto el *Front End* como del *Back End*, el front end ha sido desarrollado como ya he dicho en React Native, mientras que el back end a pesar de que las funciones se programan en React Native, realmente es Firebase y es donde se realiza el control de usuario, de animales y de todos los datos que rodean al usuario.

Ambos utilizan NodeJS que a su vez dependen del propio NPM, que es quien realmente gestiona los paquetes subidos por terceros.

10. Distribución de la Base de Datos

La base de datos está basada en un diagrama entidad-relación (ER).

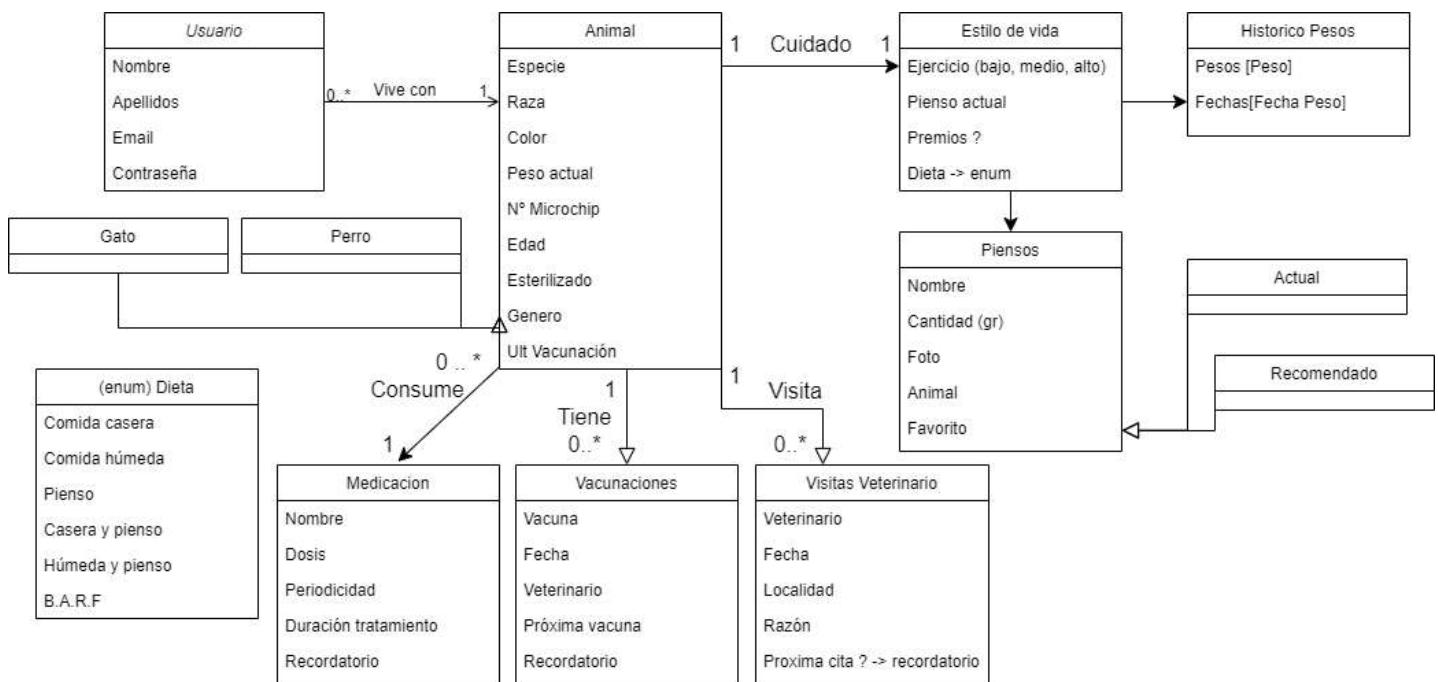
Inicialmente, la aplicación se diseñó utilizando un modelo entidad-relación, el cual es muy usado. A pesar de eso, se adaptó a una base de datos no relacional, la cual es usada por Firebase, concretamente utiliza NoSQL.

Impacto de Firebase en el diseño: Esquema NoSQL: el esquema flexible de Firebase permite anidar datos relacionados directamente dentro de los documentos principales, como se muestra en la estructura. Esto permite que no haya que realizar uniones extensas y proporciona un acceso más rápido.

Subcolecciones: La posibilidad de utilizar subcolecciones para relaciones anidadas las cuales han sido altamente utilizadas para relacionar las “tablas” entre sí.

Límites de tamaño del documento: cada documento puede tener hasta 1 MB de datos, aún así no se ha superado en ninguno de los documentos.

Funciones en tiempo real: La sincronización en tiempo real de Firestore me ha servido para implementar funciones como seguimiento de peso en vivo o añadido de cualquier dato del animal.



11. Implementación

11.1. Dependencias y librerías utilizadas



- 11.1.1. Firebase: Añade funcionalidades de backend como autenticación, base de datos y almacenamiento.
 - @firebase/app (^0.9.27):
 - Propósito: Provee la funcionalidad principal para inicializar Firebase en una aplicación web o móvil. Este módulo incluye el objeto firebase.
 - Usos Comunes: Inicializar Firebase con la configuración del proyecto.
 - @firebase/auth (^1.6.1):
 - Propósito: Implementa los servicios de autenticación de Firebase, permitiendo a los desarrolladores añadir inicio de sesión y autenticación a sus aplicaciones.
 - Usos Comunes: Autenticación con correo electrónico/contraseña, Google, Facebook, etc.
 - @firebase/firestore (^4.4.3):
 - Propósito: Proporciona acceso a Cloud Firestore, una base de datos NoSQL en tiempo real de Firebase.
 - Usos Comunes: Almacenar y sincronizar datos de la aplicación en tiempo real.
 - firebase (^10.8.1):
 - Propósito: Proporciona el SDK principal de Firebase que incluye todas las funcionalidades.
 - Usos Comunes: Incluir funcionalidades de Firebase como base de datos, autenticación, etc.



11.1.2. React Navigation: Una solución completa para gestionar la navegación en aplicaciones móviles.

- `@react-native-async-storage/async-storage (^1.21.0)`:
 - Propósito: Proporciona una solución de almacenamiento de datos clave-valor para aplicaciones React Native.
 - Usos Comunes: Almacenar tokens de autenticación, configuraciones de usuario, etc.
- `@react-native-community/datetimepicker (^7.6.3)`:
 - Propósito: Proporciona un selector de fecha/hora nativo para React Native.
 - Usos Comunes: Seleccionar fechas y horas en formularios y configuraciones.
- `@react-native-community/slider (^4.4.2)`:
 - Propósito: Un componente deslizante (slider) nativo para React Native.
 - Usos Comunes: Ajustar valores en un rango, como el volumen o el brillo, en mi caso la edad.
- `@react-native-firebase/app (^19.0.0)`:
 - Propósito: Inicializa y gestiona la conexión con Firebase específicamente para aplicaciones React Native.
 - Usos Comunes: Iniciar Firebase en aplicaciones móviles.
- `@react-native-firebase/auth (^19.0.0)`:
 - Propósito: Proporciona los servicios de autenticación de Firebase adaptados para React Native.
 - Usos Comunes: Añadir autenticación de usuarios con diferentes proveedores.
- `@react-native-firebase/storage (^19.0.0)`:
 - Propósito: Gestiona el almacenamiento de archivos en Firebase Storage para aplicaciones React Native.

- Usos Comunes: Cargar y descargar imágenes, videos, documentos, etc.
- @react-native-picker/picker (^2.6.1):
 - Propósito: Un componente selector (picker) nativo para React Native.
 - Usos Comunes: Seleccionar valores de una lista desplegable, en mi caso, razas, piensos, tipo animal etc.
- @react-navigation/bottom-tabs (^6.5.12):
 - Propósito: Proporciona una navegación basada en pestañas inferiores (bottom tabs).
 - Usos Comunes: Navegación entre diferentes pantallas usando pestañas inferiores, así se creó el NavBar de la aplicación.
- @react-navigation/native (^6.1.17):
 - Propósito: Implementa la estructura básica de navegación para aplicaciones React Native.
 - Usos Comunes: Navegar entre pantallas, manejo de rutas, etc.
- @react-navigation/native-stack (^6.9.17):
 - Propósito: Añade un stack de navegación nativa a la navegación de React Native.
 - Usos Comunes: Navegación entre pantallas usando una pila de navegación.
- react (18.2.0):
 - Propósito: La biblioteca principal de React.
 - Usos Comunes: Crear componentes y gestionar el estado en una aplicación React Native.
- react-native (^0.73.5):
 - Propósito: El framework principal de React Native para desarrollar aplicaciones móviles.

- Usos Comunes: Crear aplicaciones móviles para iOS y Android.
- react-native-chart-kit (^6.12.0):
 - Propósito: Una biblioteca para crear gráficos en aplicaciones React Native.
 - Usos Comunes: Crear gráficos de barras, líneas, circulares, etc, en mi caso, la gráfica del peso del animal.
- react-native-image-picker (^7.1.0):
 - Propósito: Proporciona una interfaz para seleccionar imágenes y videos desde la galería o cámara.
 - Usos Comunes: Capturar imágenes o videos o seleccionar de la galería, en mi caso, para subir la foto de la mascota.
- react-native-keyboard-aware-scroll-view (^0.9.5):
 - Propósito: Gestiona el comportamiento del teclado para evitar que este cubra las vistas.
 - Usos Comunes: Evitar que el teclado cubra campos de texto u otros elementos importantes.
- react-native-safe-area-context (^4.8.2):
 - Propósito: Proporciona un contexto para gestionar áreas seguras en dispositivos con notch o pantalla curva.
 - Usos Comunes: Ajustar la interfaz de usuario para respetar las áreas seguras.
- react-native-screens (^3.29.0):
 - Propósito: Mejora la gestión de la navegación utilizando vistas nativas.
 - Usos Comunes: Acelerar la navegación entre pantallas en aplicaciones React Native.
 - Dependencias de Desarrollo



11.1.3. axios (^1.6.8):

- Propósito: Una biblioteca de clientes HTTP basada en promesas.
- Usos Comunes: Realizar solicitudes HTTP a una API, manejar respuestas, etc.



- expo (~50.0.6):

- Propósito: Proporciona el SDK principal para las aplicaciones desarrolladas con Expo.
- Usos Comunes: Acceder a componentes y APIs de Expo, como la cámara, notificaciones, etc.

- expo-blur (^12.9.2):

- Propósito: Proporciona vistas difuminadas utilizando la API nativa de desenfoque.
- Usos Comunes: Aplicar un efecto borroso sobre otras vistas.

- expo-status-bar (~1.11.1):

- Propósito: Proporciona un componente para personalizar la barra de estado en aplicaciones Expo.
- Usos Comunes: Cambiar el color, estilo o visibilidad de la barra de estado.



- @babel/core (^7.20.0):

- Propósito: Proporciona el compilador de Babel para transformar el código JavaScript.
- Usos Comunes: Convertir código moderno a versiones compatibles para diferentes navegadores y entornos.
- Observaciones Generales:

- Expo SDK: Proporciona un conjunto de herramientas integradas para el desarrollo de aplicaciones móviles sin necesidad de configuraciones complejas.

12. Usabilidad y desarrollo

12.1. Motivación y el trasfondo de GoodWeight

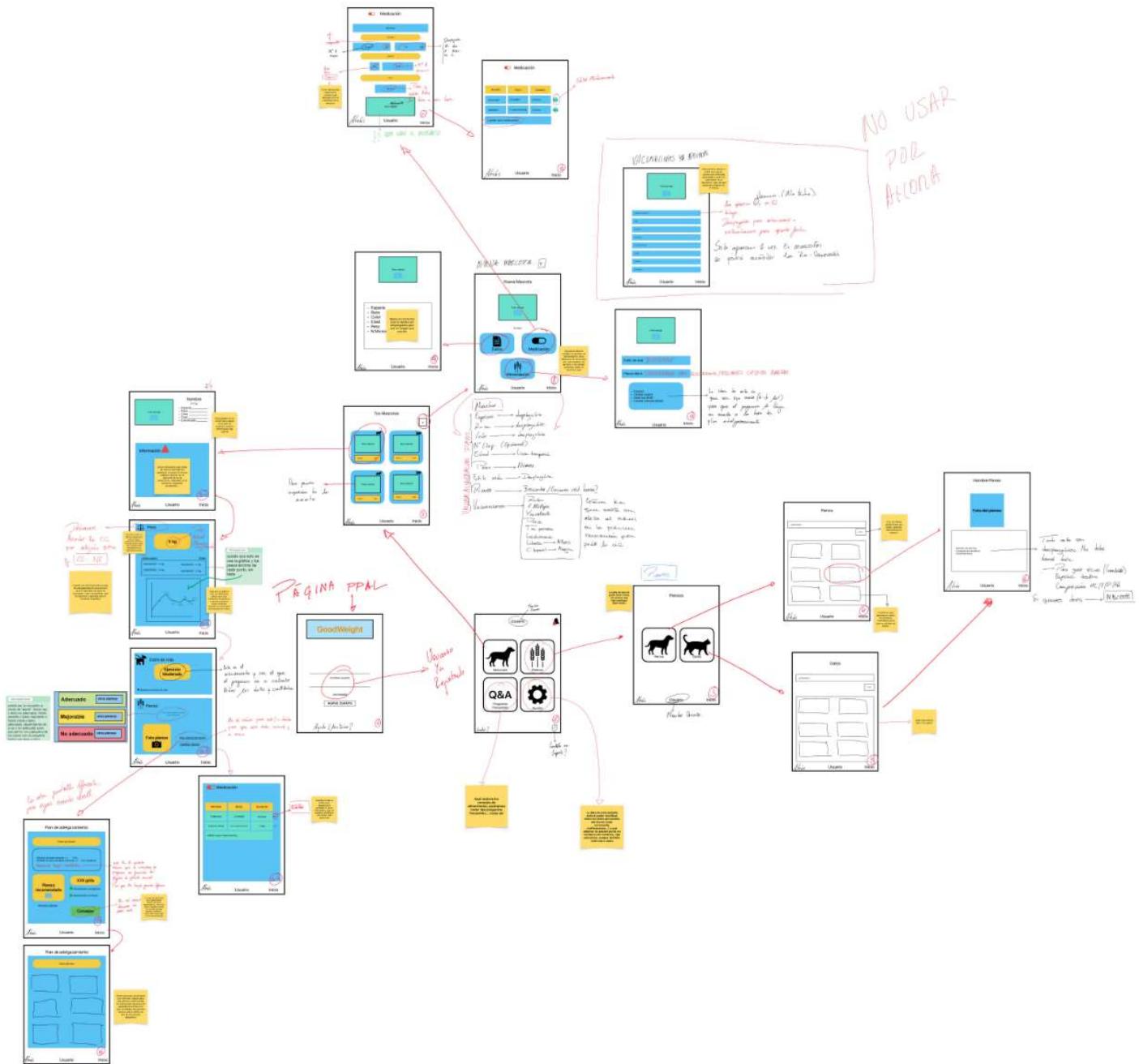
GoodWeight surgió a partir de una observación realizada por un compañero veterinario durante sus prácticas y trabajo en diversas clínicas y hospitales veterinarios. Él se percató de que numerosos dueños de mascotas enfrentaban dificultades al gestionar el cuidado de estos animales, particularmente en lo referente a su nutrición. Ante esta problemática, se puso en contacto conmigo para proponerme liderar un proyecto que ofreciera soluciones a estos desafíos.

Héctor, que así se llama mi compañero, me presentó la idea inicial de desarrollar una aplicación. Aunque en un principio el proyecto parecía extremadamente complejo para alguien que aún estaba finalizando sus estudios en ingeniería informática, el concepto pronto comenzó a tomar forma. El diseño inicial incluía numerosas pantallas, un esquema que, tras varias iteraciones y mejoras, evolucionaría hasta convertirse en lo que hoy conocemos como *GoodWeight*.

Después de desarrollar un prototipo funcional, decidimos presentar nuestro proyecto en diversos concursos, logrando un éxito notable. Participamos y ganamos en dos competiciones importantes: el **Santander XPlorer 1st Edition 2023** y otro concurso organizado por la **Xunta de Galicia**. Nuestro triunfo en estos eventos, basado en la presentación de un plan de negocio claro y el prototipo, nos animó a continuar con el desarrollo de la aplicación.

Inspirado por estos logros, decidí que *GoodWeight* sería el proyecto ideal para mi trabajo de fin de grado, dada la coincidencia temática y la oportunidad de aplicar lo aprendido durante mi carrera.

Tras meses de dedicación y esfuerzo intensivo, hemos logrado transformar esa idea original en la versión actual de *GoodWeight*, una solución innovadora para la gestión nutricional de mascotas.



12.2. Desarrollo:

Uno de los principales retos al comenzar a trabajar con una nueva tecnología fue la complejidad inicial que todo parecía tener, haciendo que tareas aparentemente simples, como crear una barra de navegación (NavBar), me llevaran varios días. Comencé desarrollando las funcionalidades de inicio de sesión y registro, lo que me permitió una introducción básica al uso de Firebase. A continuación, como ya mencioné, implementé el NavBar, que aunque inicialmente llevaba a pantallas vacías, ya era funcional.

A medida que fui completando y detallando las pantallas, el proceso se hizo más fluido, al menos hasta que surgían nuevos obstáculos. Uno de los desafíos recurrentes fue el estilizado de las pantallas para lograr un diseño atractivo y funcional. Para ello, utilicé elementos que proporciona React Native, los cuales, a pesar de su utilidad, presentan una curva de aprendizaje no trivial, especialmente en la estilización. Por ejemplo, al trabajar en la funcionalidad para renderizar tarjetas interactivas que llevan a la información detallada de cada animal al hacer clic en ellas, encontré dificultades significativas.

Por ejemplo para renderizar por tarjetas y que llevase a cada animal a la hora de clicar:

```
const renderPet = ({ item }) => {

  let petImage = placeholderOtherImage; // Default image

  if (item.photo) {

    petImage = { uri: item.photo };

  } else {

    switch (item.species) {

      case 'Perro':

        petImage = placeholderDogImage;

        break;

      case 'Gato':

        petImage = placeholderCatImage;

        break;

    }

  }

  return (
    <TouchableOpacity style={styles.petCard} onPress={() =>
navigation.navigate('HomeAnimal', { petId: item.id })}>
      <Image source={petImage} style={styles.petImage} />
      <Text style={styles.petName}>{item.name}</Text>
      <Text style={styles.petAge}>{item.age}</Text>
    </TouchableOpacity>
  );
};
```

La pantalla de inicio del animal en la aplicación resultó ser uno de los componentes más desafiantes de desarrollar, a pesar de su aparente simplicidad visual. El objetivo era presentar toda la información relevante de manera clara y accesible, lo que implicó una considerable carga de trabajo detrás de las escenas.

Para empezar, fue necesario introducir una extensa cantidad de datos sobre las razas de animales. En concreto, incorporé datos de aproximadamente 50 razas de perros y 30 de gatos. Cada entrada no solo necesitaba incluir el nombre de la raza, sino también detalles específicos como el tamaño y el peso máximo y mínimo ideal tanto para machos como para hembras²⁶. Extraje los datos necesarios de un sitio web que contenía las especificaciones deseadas, lo que me permitió obtener la información de manera estructurada y precisa.

Una vez recopilados los datos, el siguiente paso fue limpiarlos y estructurarlos adecuadamente para su uso en la aplicación. Desarrollé un script que automatiza este proceso, eliminando cualquier inconsistencia y preparando los datos para su integración final. Posteriormente, cargué toda esta información en Firebase, donde se almacenaría y gestionaría de forma segura.

Además, con vistas a futuras ampliaciones de la aplicación, también recolecté información detallada sobre diferentes tipos de piensos. Esto incluyó no solo imágenes, títulos y descripciones, sino también los ingredientes de cada producto. Mi objetivo a largo plazo es poder analizar y presentar los nutrientes específicos de cada pienso, facilitando así recomendaciones personalizadas basadas en las necesidades dietéticas de cada animal. Este paso adicional no solo mejoró la funcionalidad actual de

la aplicación, sino que también sentó las bases para futuras mejoras y características.

Este enfoque detallado y minucioso en la fase de desarrollo garantiza que la pantalla de inicio del animal no solo cumpliera con su función informativa básica, sino que también se preparará para futuras expansiones y mejoras en la aplicación.

Estos fueron los resultados en la base de datos:

The screenshot displays two tables from a database interface:

- Breeds Table:**
 - Header: Perro > species > Perro > breeds > Airedaleterrier
 - Columns: breeds, breeds, Airedaleterrier
 - Actions: + Iniciar colección, + Agregar documento, + Iniciar colección, + Agregar campo
 - Data:

breeds	Galgoafgano	maxIdealFemale: 19
food	Galgoespañol	maxIdealMale: 25
	Galgoinglés	minIdealFemale: 19
	Galgorusoborzoi	minIdealMale: 25
	Goldenretriever	name: "Airedale terrier"
	Gordonsetter	size: "Grande"
	Grandanés	
- Food Table:**
 - Header: Perro > species > Perro > food > ACANAADULTA...
 - Columns: Perro, food, ACANAADULTALLBREEDSGRASSFEDCORDEROYMANZANAPIENSOPARAPERROS...
 - Actions: + Iniciar colección, + Agregar documento, + Iniciar colección, + Agregar campo
 - Data:

breeds	ACANAADULTALLBREEDSGRASSFEDCORDEROYMANZANAPIENSOPARAPERROS...
food	ACANAADULTLARGEPIENSOPARAPERROS
	ACANAADULTLIGHTPIENSOPARAPERROS
	ACANAADULTSMALLBREEDPIENSOPARAPERROS
	ACANACEROCONCALABAZAPIENSOPARAPERROS
	ACANACLASSICSPRAIRIEPOULTRYPOLLOPIEN...
	ACANACLASSICSWILDCOASTPIECPESCADOPIENOP...
	ACANAFREERUNPATOPIENSOPARAPERROS
	ACANAPACIFICAPIENSOPARAPERROS
	ACANAPUPPYLARGEPIENSOPARAPERROS
	ACANAPUPPYMEDIUMPIENSOPARAPERROS
	ACANAPUPPYSMALLBREEDPIENSOPARAPERROS
	ACANARANCHLANDSCARNESROJASPIENSOPARA...
	ACANASENIORPIENSOPARAPERROS
	ACANASPORTAGILITYPIENSOPARAPERROS
	ACANASPORTANTIDOTEPIENSOPARAPERROS

Con esto ya tenía una base de datos poblada para el uso del usuario. Gracias a los datos que tenía en la base de datos, pude crear la pantalla de mostrado de piensos, junto con un filtro y un buscador además de la del home del animal:



Para finalizar el desarrollo de la aplicación, incorporé un chatbot que interactúa con los usuarios mediante la tecnología de OpenAI, específicamente usando la API de ChatGPT. Este componente fue posible gracias a la valiosa ayuda de tutoriales y repositorios en GitHub, que ofrecieron guías detalladas sobre qué librerías utilizar, cómo tokenizar los datos y cómo gestionar el envío y la recepción de mensajes de los usuarios.

Inicialmente, mi enfoque para el manejo del contexto en las conversaciones era bastante básico: simplemente enviaba el mensaje anterior del usuario como contexto para la nueva interacción. Sin embargo, con el tiempo, y gracias a la exploración de más recursos, descubrí el modelo de embebimiento text-embedding-ada-002 de OpenAI. Este modelo me permitió tokenizar y embeber el contenido del contexto de manera más eficiente, lo que no solo mejoraba la calidad de las interacciones con el chatbot, sino que también reducía los costos operativos, un aspecto crucial para la escalabilidad del proyecto.

Implementé este sistema mediante varios archivos clave:

- **sourceData.txt**: donde se introduce el contexto necesario para las conversaciones.
- **embedding.js**: se utiliza para tokenizar y embeber el texto antes de enviarlo a OpenAI.
- **completion.js**: este archivo procesa la información recibida y genera las respuestas a las preguntas del usuario.

Además, integré todos estos componentes con un servidor y una pantalla dedicada en la aplicación, teniendo al final un chatbot que dispone de aproximadamente 600 líneas de información. Este bot ha demostrado ser capaz de responder preguntas de manera óptima y consistente.

No obstante, a pesar de los avances técnicos y la eficacia del bot, una de las consideraciones finales y más desafiantes antes de lanzar el producto al mercado es la evaluación de los costos finales. La implementación de tecnologías de inteligencia artificial, especialmente en un formato interactivo como un chatbot, puede implicar costos operativos significativos. Por lo tanto, es crucial realizar un análisis detallado de estos costos para asegurar que la solución sea económicamente viable a largo plazo y determinar la estrategia de precios más adecuada para el servicio.

12.3. Usuarios que han probado la aplicación

12.3.1. Tareas realizadas en el testeo

12.3.2. Resultados del testeo

12.3.3. Conclusiones

13. Seguridad y administración de los datos del usuario

En una aplicación como GoodWeight, la seguridad y la administración de datos de usuario son aspectos fundamentales que no deben ser subestimados. Aunque el uso de servicios de Google, como Firebase, proporciona una capa robusta de seguridad garantizando la integridad y protección de los datos almacenados, es crucial tener en cuenta que ningún sistema es completamente infalible y puede ser susceptible a vulnerabilidades.

13.1. Encriptación y autenticación en Firebase:

En *GoodWeight*, la seguridad de las credenciales de usuario es una prioridad. Por esto, las contraseñas y los nombres de usuario se gestionan a través de la sección "Authentication" de Firebase, donde las contraseñas son encriptadas. Este proceso asegura que las contraseñas no sean visibles ni siquiera para los administradores del sistema, y sólo el usuario propietario tiene la capacidad de acceder a su cuenta con sus credenciales. Además, para realizar cualquier operación que involucre acceso a la base de datos desde el código, se requiere una clave secreta que solo los administradores pueden visualizar, reforzando aún más la seguridad.

13.2. Reglas de Firebase:

Para administrar el acceso y las interacciones dentro de la aplicación, Firebase ofrece un conjunto de reglas configurables que permiten al administrador conceder o restringir el acceso según el rol o el nivel del

usuario. En el caso de GoodWeight, al contar con un único tipo de usuario, la configuración de estas reglas se simplifica considerablemente. Estas son algunas de las reglas implementadas:

- **Acceso a datos personales y de mascotas:** Se establecen políticas estrictas para asegurar que los usuarios solo tengan acceso a sus propios animales y datos personales. Esto se logra mediante reglas que identifican y verifican la identidad del usuario antes de permitir el acceso a la información.
- **Permisos de lectura y escritura:** Aunque los usuarios tienen acceso completo (lectura y escritura) a sus propios datos, los permisos para acceder a información común o compartida se limitan únicamente a la lectura. Esto significa que los usuarios pueden ver, pero no modificar, los datos que son de acceso común.

Aquí hay un ejemplo de cómo podrían estructurarse estas reglas en Firebase:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

    // Anyone can read any document in the database
    match /{document=**} {
      allow read, write: if true;
    }

    // Users can read and write their own document in the 'users' collection
    match /users/{userId}/{document=**} {
      allow read, write: if request.auth != null & request.auth.uid == userId;
    }

    // Users can read and write within their own 'animals' subcollection
    match /users/{userId}/animals/{animalId} {
      allow read, write: if request.auth != null & request.auth.uid == userId;
    }
  }
}
```

Estas medidas son esenciales para proteger la privacidad de los usuarios y garantizar la seguridad de los datos en la aplicación, equilibrando entre accesibilidad y protección.

13.3. Uso de Authentication en firebase

El uso de **Authentication en Firebase** es un aspecto crucial para garantizar la seguridad y la privacidad en *GoodWeight*. Esta plataforma de desarrollo proporciona una manera eficiente y segura de manejar la autenticación de usuarios, eliminando la necesidad de encriptar manualmente las contraseñas gracias a sus librerías integradas. Al registrar a los usuarios, sus contraseñas son automáticamente encriptadas antes de ser almacenadas en la base de datos, asegurando que incluso en caso de un acceso no autorizado a la base de datos, la información de las contraseñas permanezca protegida.

Lista de usuarios registrados				
Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
antonio@a.es		17 abr 2024	17 abr 2024	UmuxNLCxV7XiNb14iSrKnszT...
antonio@ua.es		14 abr 2024	14 abr 2024	uetKK07FQte09RY8SsZVSI3s...
antonio@gmail.com		7 mar 2024	10 mar 2024	jgsq2iTdiiPAkR2saQtytHeWCjx1
a@a.es		6 mar 2024	19 abr 2024	SdZxShLQaZVtqoaUAmFZ9z0...

13.4. Datos del usuario y como se han tratado

En cuanto al **tratamiento de los datos del usuario**, este se realiza conforme a la normativa vigente de la Unión Europea. Durante el proceso de registro e inicio de sesión, se proporcionan indicaciones claras sobre la política de privacidad, ofreciendo a los usuarios la oportunidad de entender y aceptar cómo serán utilizados y protegidos sus datos personales. Esto no solo es fundamental para cumplir con las leyes de protección de datos, sino que también fortalece la confianza del usuario en la aplicación.

Además, los datos interactuados con el chatbot, que utiliza la tecnología de OpenAI, son gestionados según los términos y condiciones de OpenAI. Esto significa que, aunque la entrada de datos pueda ser controlada por *GoodWeight*, la información procesada y almacenada por el chatbot está bajo la jurisdicción de OpenAI, lo que implica que la manipulación y almacenamiento de dichos datos se hacen conforme a otra serie de políticas de privacidad y seguridad.

Por otro lado, respecto a la **seguridad de los datos almacenados**, estos se mantienen en Firebase donde solo yo, como administrador, tengo acceso a ellos mediante claves seguras y procedimientos de autenticación rigurosos. Este nivel de control ayuda a prevenir accesos no autorizados y garantiza que los datos sensibles de los usuarios se manejen de manera segura.

Esta estructura de seguridad y privacidad no solo cumple con las regulaciones legales, sino que también ofrece a los usuarios una garantía de que sus datos personales están bien protegidos, lo cual es esencial para el mantenimiento de la confianza y la integridad del sistema en *GoodWeight*.

14. Comparativas finales

Para las comparaciones finales,

15. Conclusiones

16. Ampliación y proyecto de futuro

Referencias (ctrl+. ctrl+,)

¹ <https://apps.apple.com/us/app/pet-first-aid/id780415389>

² <https://www.doglogapp.com/>

³

<https://play.google.com/store/apps/details?id=com.m11pets.elevenpets&hl=es&gl=US>

⁴ <https://www.petcoach.co/>

⁵ <https://www.myollie.com/>

⁶ <https://www.nngroup.com/articles/measuring-perceived-usability/>

⁷

<https://www.nngroup.com/articles/success-rate-the-simplest-usability-metric/>

⁸

<https://measuringu.com/wp-content/uploads/2022/01/Benchmark-Testing.pdf>

⁹ <https://measuringu.com/task-based-metrics/>

¹⁰ <https://measuringu.com/ux-benchmarks/>

¹¹ <https://www.shanedoyle.io/post/ux-design-metric-to-measure-lostness>

¹²

https://medium.com/@steve_mullen/an-introduction-to-lean-canvas-5c17c469d3e0

¹³ [https://www.investopedia.com/terms/s/swot.asp#:~:text=SWOT%20\(strengths%2C%20weaknesses%2C%20opportunities,as%20current%20and%20future%20potential.](https://www.investopedia.com/terms/s/swot.asp#:~:text=SWOT%20(strengths%2C%20weaknesses%2C%20opportunities,as%20current%20and%20future%20potential.)

¹⁴ <https://www.anfaac.org/datos-sectoriales/>

¹⁵ <https://reactnative.dev/>

¹⁶ <https://firebase.google.com/?hl=es>

¹⁷ <https://github.com/>

¹⁸ <https://trello.com/es>

¹⁹ <https://nodejs.org/en>

²⁰ <https://www.loop11.com/>

²¹ <https://www.justinmind.com/>

²² <https://code.visualstudio.com/>

²³ <https://developer.apple.com/xcode/>

²⁴<https://chromewebstore.google.com/detail/diagramsnet/onlkggianjhjenigcpigpjehhpplldkc?hl=es-419>

²⁵ <https://openai.com/chatgpt>

²⁶ <https://petyzoo.com/razas-de-perros/>

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46