

Aplicativo web para servicio de cuidado de mascotas

David Ortiz Pérez

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director

Carlos Adolfo Beltrán Castro

Magister en gestión de la tecnología educativa

Tutor

Fernando Antonio Rojas Morales

Magister en ciencias de la computación

Universidad Industrial de Santander

Facultad de ingenierías Físico mecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2026

Tabla de Contenido

	Pág.
Introducción	13
1.Planteamiento y Justificación del Problema	15
2.Objetivos	17
2.1. Objetivo General.....	17
2.2 Objetivos Específicos.....	17
3.Marco de Referencia	18
3.1 Fundamentos Teóricos	18
3.1.1 E-Commerce	18
3.2 Entorno de Desarrollo.....	18
3.2.1 React	18
3.2.2 Javascript.....	19
3.2.3 Json	19
3.2.4 TypeScript.....	19
3.2.5 MongoDB	20
3.2.6 Visual Studio Code	20
3.2.7 Nest	20
3.2.8 Github	21
3.3 Estado del Arte.....	21
3.3.1 Nacional	21
3.3.1.1 Rappi.....	21
3.3.1.2 Laika.	22

3.3.2 Internacional	22
3.3.2.1 Amazon.....	22
3.3.2.2 Mercado Libre.....	22
3.3.2.3 Aliexpress.	23
3.4 Antecedentes del Tema.....	23
3.4.1 Rover.....	24
3.4.2 Petcrew.....	24
3.4.3 Ohana	24
3.4.4 Wag!.....	25
4. Metodología	26
4.1 Metodología Incremental	26
4.1.1 Comunicación	27
4.1.2 Planeación	28
4.1.3 Modelado	28
4.1.4 Construcción	29
4.1.5 Despliegue.....	29
5. Desarrollo del Proyecto.....	31
5.1 Lista de Requerimientos	31
5.1.1 Requerimientos Funcionales.....	31
5.1.2 Requerimientos no Funcionales.....	32
5.2 Usuarios	32
5.3 Diagramas de Casos de Uso.....	33
5.4 Casos de Uso.....	37

5.5 Diagrama de Actividades 49

5.6 Diagramas de Clases 54

5.6.1 Modelo no SQL..... 54

5.7 Arquitectura de software..... 55

5.8 Prototipos 57

5.8.1 Prototipo uno..... 65

5.8.2 Prototipo dos..... 76

5.9 Pruebas de funcionalidad 109

5.9.1 Pruebas primer prototipo..... 109

5.9.2 Pruebas prototipo final..... 110

6.Conclusiones 112

7.Recomendaciones 113

Referencias Bibliográficas 114

Apéndices 116

Lista de Tablas

Tabla 1. Requerimientos funcionales..... 31

Tabla 2. Requerimientos no funcionales..... 32

Tabla 3. Usuarios potenciales. 32

Tabla 4. Caso de uso Registro usuario..... 38

Tabla 5. Caso de uso Inicio de sesión..... 39

Tabla 6. Caso de uso Editar perfil..... 40

Tabla 7. Caso de uso ver servicios..... 40

Tabla 8. Caso de uso Enviar solicitud de servicio. 41

Tabla 9. Caso de uso Aceptar solicitud de servicios..... 42

Tabla 10. Caso de uso Consultar solicitud de servicio. 43

Tabla 11. Caso de uso Cancelar un servicio. 44

Tabla 12. Caso de uso Finalizar un servicio. 45

Tabla 13. Caso de uso Consultar historial de servicios. 46

Tabla 14. Caso de uso Calificar un servicio. 46

Tabla 15. Caso de uso Agregar perfil de mascota. 47

Tabla 16. Caso de uso Editar perfil de mascota..... 48

Tabla 17. Caso de uso Agregar vacunas a mascota. 49

Tabla 18. Pruebas funcionales realizadas al primer prototipo. 109

Tabla 19. Pruebas funcionales realizadas al prototipo final. 110

Lista de Figuras

Figura 1. Metodología incremental. 26

Figura 2. Etapas metodología incremental. 27

Figura 3. Diagrama de caso de uso para el usuario cliente. 34

Figura 4. Diagrama de caso de uso para el usuario cuidador. 35

Figura 5. Diagrama de caso de uso para el usuario veterinario. 36

Figura 6. Diagrama de caso de uso para el usuario (cliente, cuidador y veterinario). 37

Figura 7. Diagrama de actividad iniciar sesión. 50

Figura 8. Diagrama de actividad editar perfil. 51

Figura 9. Diagrama de actividad editar perfil mascota. 52

Figura 10. Diagrama de actividad solicitud se servicio. 53

Figura 11. Modelo no SQL de la base de datos. 54

Figura 12. Diagrama de arquitectura de software. 55

Figura 13. Mockup Pantalla inicial. 58

Figura 14. Mockup Pantalla registro. 59

Figura 15. Mockup Pantalla inicial cliente. 60

Figura 16. Mockup editar perfil. 61

Figura 17. Mockup Mi mascota. 62

Figura 18. Mockup Plan de vacunación. 63

Figura 19. Mockup Historial de servicios. 64

Figura 20. Mockup Servicios activos. 65

Figura 21. Pantalla inicial, prototipo uno. 67

Figura 22. Registro de usuarios.	68
Figura 23. Registro de colaboradores.	69
Figura 24. Página inicial cliente.....	70
Figura 25. Vista ver y editar perfil.....	71
Figura 26. Vista crear perfil mascota.....	72
Figura 27. Vista plan de vacunación mascota.....	73
Figura 28. Vista historial de servicios.....	74
Figura 29. Vista de servicio activo.	75
Figura 30. Vista de inicio.....	77
Figura 31. Vista del formulario de registro para clientes (dueños de mascotas).	78
Figura 32. Vista del formulario de registro para colaboradores (cuidadores o veterinarios).	79
Figura 33. Definición del esquema de Usuario en el backend.....	80
Figura 34. Vista iniciar sesión.	81
Figura 35. Fragmento de código proceso de autenticación.....	82
Figura 36. Archivo de validación de usuarios al iniciar sesión.	83
Figura 37. Archivo de validación de token al inicio con sesión.	84
Figura 38. Archivo configuración tiempo de sesión y token.....	85
Figura 39. Vista de inicio con sesión iniciada (usuario cliente).	86
Figura 40. Vista ver perfil de usuario.	87
Figura 41. Vista de edición de perfil del usuario.....	88
Figura 42. Vista de la sección “Mi mascota”.....	89
Figura 43. Vista mi mascota.	90
Figura 44. Vista edición del perfil de mascota.	91

Figura 45. Vista plan de vacunación de mascota.....	92
Figura 46. Vista agregar vacuna.	93
Figura 47. Vista servicios disponibles.	94
Figura 48. Vista solicitud de servicio.	95
Figura 49. Vista historial de servicios.....	96
Figura 50. Vista cancelar un servicio.....	97
Figura 51. Vista información del colaborador.	98
Figura 52. Vista calificar servicio.	99
Figura 53. Vista historial de servicios.....	100
Figura 54. Vista inicial del panel de colaboradores.....	101
Figura 55. Vista inicial del panel de colaboradores.....	102
Figura 56. Fragmento de código del método FindCollabOrders implementado en el backend.	103
Figura 57. Fragmento de código del método assignorderToColab implementado en el backend	104
Figura 58. Vista perfil de mascota.	105
Figura 59. Vista servicios tomados por el colaborador.....	106
Figura 60. Vista historial de servicios ofrecidos.....	107
Figura 61. Archivo estructura general del proyecto.....	108

Lista de Apéndices

Apéndice A. análisis de acercamiento inicial 116

Apéndice B. Análisis preguntas aplicadas al primer prototipo 124

Apéndice C. Análisis preguntas aplicadas al segundo prototipo 128

Glosario

Aplicativo web: es una aplicación informática accesible mediante un navegador que permite a los usuarios interactuar con diversas funcionalidades sin necesidad de una instalación local.

Comercio electrónico (e-commerce): hace referencia a las transacciones comerciales de productos o servicios que se realizan a través de internet. En otras palabras, corresponde al proceso de compra y venta de bienes o servicios mediante medios electrónicos, lo cual permite que los productos estén disponibles las 24 horas del día.

Framework: es la estructura base utilizada como punto de partida para el desarrollo de un proyecto específico. Constituye un marco de trabajo y un esquema que facilita tanto la implementación como la evolución de una aplicación.

Metodología incremental: es un modelo de desarrollo de software que divide el proceso en iteraciones o incrementos. En cada fase se añaden nuevas funcionalidades las cuales son validadas y mejoradas con cada retroalimentación obtenida.

Prototipo: corresponde a una versión preliminar de un sistema o producto que se utiliza para evaluar su viabilidad, funcionalidad y diseño.

Resumen

Título: Aplicativo web para servicio de cuidado de mascotas*

Autor: David Ortiz Pérez**

Palabras Clave: React, desarrollo de prototipo, modelo de negocio, comercio electrónico, mascotas, cuidado, vacunas.

Descripción: El aumento en la tenencia de mascotas durante y después de la pandemia ha puesto en evidencia una necesidad creciente de contar con soluciones confiables para su cuidado y atención médica. En muchas ocasiones los propietarios se enfrentan a situaciones en las que no pueden brindar atención constante a sus animales, lo que genera preocupación por su bienestar y calidad de vida. Esta realidad ha impulsado el desarrollo de plataformas digitales que permitan conectar de forma segura a cuidadores y veterinarios con los dueños de mascotas. En este contexto, se diseñó y desarrolló un prototipo de aplicación web orientado a facilitar dicho acceso, no solo como respuesta a una necesidad local, sino también como base para una herramienta escalable y adaptable a otros entornos. Para su desarrollo se emplearon tecnologías modernas como React con TypeScript para el desarrollo del frontend y NestJS en la construcción del backend, garantizando así una estructura robusta, modular y eficiente. La metodología aplicada fue incremental, lo que permitió integrar mejoras continuas a lo largo de cada fase del proceso de construcción. En consecuencia, este prototipo representa una propuesta tecnológica que busca contribuir al bienestar animal mediante la integración de herramientas digitales dentro del marco del comercio electrónico y los servicios especializados aplicado en este caso para perros.

* Trabajo de Grado

** Facultad de ingenierías fisicomecánicas. Escuela de ingeniería de sistemas. Ingeniería de sistemas. Director: Carlos Adolfo Beltrán Castro. Magister en gestión de la tecnología educativa.

Abstract

Title: Web Application for Pet Care Services *

Author(s): David Ortiz Perez **

Key Words: React, prototype development, business model, e-commerce, pets, care, vaccinations.

Description: The increase in pet ownership during and after the pandemic has highlighted a growing need for reliable solutions to ensure their care and medical attention. In many cases, many pet owners face situations in which they are unable to provide consistent attention to their animals, raising concerns about their overall well-being and quality of life. This reality has encouraged the development of digital platforms capable of securely connecting pet owners with sitters and veterinarians.

In this context, a web application prototype was designed and developed to facilitate such access, not only as a response to a local need, but also as the foundation for a scalable and adaptable solution that can be implemented in different environments.

Modern technologies were used in its development, including React with TypeScript for frontend development and NestJS for the backend construction, ensuring a robust, modular, and efficient architecture. An incremental methodology was adopted, allowing continuous improvements throughout each stage of the development process. Consequently, this prototype represents a technological proposal aimed at contributing to animal welfare through the integration of digital tools within the framework of e-commerce and specialized services, specifically applied to dog care.

* Degree Work

** Faculty of Physics-mechanics Engineering. School of Systems Engineering. Systems Engineering. Advisor: Carlos Adolfo Beltrán Castro. Master in educational technology

Introducción

En la actualidad, las plataformas de comercio digital han tenido una rápida expansión y han sido ampliamente adoptadas por el público. Una plataforma digital es un entorno que permite la interacción entre usuarios, facilitando el acceso a productos, servicios o información a través de dispositivos como computadoras o teléfonos inteligentes. Este crecimiento se debe al alcance y la facilidad que brindan a sus usuarios de acceder a una gran gama de productos y servicios por medio de un dispositivo como computadoras y teléfonos inteligentes. La facilidad de uso y la conveniencia de estas plataformas han transformado el proceso de compra a una experiencia más amigable, eficiente y en muchos casos personalizada para los consumidores. Esto demuestra el potencial de las tecnologías para resolver problemas cotidianos al conectar usuarios con soluciones adaptadas a sus necesidades, en cualquier momento y lugar.

Dado el creciente reconocimiento de estas plataformas y teniendo en cuenta las posibilidades de introducción en el mercado de una herramienta especializada que facilite a los dueños de mascotas la búsqueda de cuidadores y veterinarios confiables, se determina la viabilidad de desarrollar un prototipo web. Un prototipo web es un modelo funcional o una versión preliminar de un sitio web, diseñado para evaluar su estructura, funcionalidad y diseño antes de un despliegue definitivo.

Con el fin de mantener la confiabilidad del prototipo se implementó un mecanismo de calificación mediante una escala de uno a cinco, a través de la cual los usuarios podrán evaluar los

servicios recibidos. Este sistema permite llevar un registro del desempeño de los colaboradores para fortalecer la confianza del prototipo web.

Consiguientemente, el desarrollo de este prototipo busca impactar favorablemente a la población Bumanguesa, brindando una solución accesible y confiable para el cuidado y atención médica de las mascotas caninas. Esta práctica se lleva a cabo con el respaldo de la Escuela de Ingeniería de Sistemas de la Universidad Industrial de Santander, con el fin de ofrecer un servicio eficiente y adaptado a las necesidades de los usuarios.

1. Planteamiento y Justificación del Problema

Durante y después de la pandemia ha habido una tendencia creciente a aceptar nuevas incorporaciones al núcleo familiar, estos fueron las mascotas en 2019 el 47 % de los hogares contaba con una mascota y esta cifra ha aumentado hasta el 57 % en 2024 y de este total el 71 % corresponden a caninos de acuerdo con la empresa de consultoría e investigación Cifras y Conceptos (2024), y puesto que estas pueden ofrecer compañía y apoyo emocional pueden ser muy buenos compañeros que ofrecen cariño y amor incondicional a sus dueños. Esta compañía es muy útil para las personas que tienen dificultades para establecer relaciones sociales y tiene un efecto positivo para la salud física y mental pues se demuestra que los dueños de mascotas tienen menor riesgo a tener enfermedades cardíacas y menos probabilidad de sufrir depresión o ansiedad según un reportaje de la universidad EAN (2023).

En muchas ocasiones los dueños de las mascotas tienen compromisos laborales, viajes o situaciones imprevistas donde se ven obligados a dejar a sus mascotas solas en casa. En estas situaciones las mascotas pueden tender a sufrir de ansiedad, estrés y aburrimiento, lo cual puede repercutir negativamente en su bienestar. Es esencial que en ausencia de sus dueños puedan tener suficiente líquido, comida y ventilación en un ambiente cómodo. En estas circunstancias los dueños se ven obligados a dejar a sus mascotas con algunos vecinos, familiares o amigos. Sin embargo, no siempre es posible encontrar una persona de confianza o con la experiencia necesaria para que se encargue de los cuidados de sus mascotas. Esto ha llevado a que muchas personas contraten guarderías o cuidadores que son conscientes de los cuidados necesarios de las mascotas, encargándose de ellas y asegurando su bienestar.

De la misma manera, cuando una mascota requiere atención médica encontrar un veterinario confiable y disponible puede representar un desafío, especialmente en situaciones de emergencia. Los dueños necesitan la atención de un profesional calificado rápidamente que pueda proporcionar no solo cuidados rutinarios sino también una atención médica especializada en caso de ser requerida.

Tomando en cuenta la problemática anteriormente mencionada se decide crear una solución mediante un prototipo web que permita a los dueños de mascotas caninas encontrar de manera fácil y rápida tanto cuidadores como veterinarios. Los colaboradores estarán capacitados para brindar un servicio de calidad adaptándose a las necesidades específicas de cada una de las mascotas. Este prototipo no solo proporcionará una red de cuidadores que sean amantes de los animales, sino también médicos veterinarios, asegurando así que a las mascotas se les brinde una atención adecuada tanto en términos de cuidados diarios como en atención médica profesional cuando sea requerida.

2. Objetivos

2.1. Objetivo General

Desarrollar un prototipo web para facilitar la contratación de cuidadores y veterinarios calificados en brindar servicios para el cuidado de mascotas en Bucaramanga.

2.2 Objetivos Específicos

1. Identificar los requerimientos funcionales y no funcionales para el desarrollo del prototipo.
2. Diseñar la estructura de datos, arquitectura y las vistas del prototipo.
3. Desarrollar el prototipo web que se ajuste a los requerimientos previamente establecidos.
4. Hacer pruebas funcionales y de usuarios.

2. Marco de Referencia

En esta sección se definen los fundamentos básicos necesarios para ayudar a la comprensión del funcionamiento del prototipo.

3.1 Fundamentos Teóricos

3.1.1 E-Commerce

Este modelo ha venido en aumento los últimos años gracias a la revolución digital y Amazon (s.f.) define el e-commerce como “el comercio de bienes y servicios a través de internet. Una forma de que la gente y vende productos al por menor conectándolos a través de varios canales electrónicos como lo pueden ser sus computadoras o dispositivos móviles”.

Ofrecer soluciones a sus clientes de manera rápida, efectiva y a tan solo un clic de distancia da paso a muchas nuevas oportunidades de emprendimiento, ya que este facilita la diversidad de productos y servicios sin preocuparse por las limitaciones que podría tener un establecimiento físico además de ahorrar a los clientes tiempo, puesto que pueden ver todo el catálogo de manera virtual y solicitar sus servicios directamente desde casa.

3.2 Entorno de Desarrollo

3.2.1 React

React es una biblioteca de JavaScript para renderizar interfaces de usuario (UI, por sus siglas en inglés). La UI se construye a partir de pequeñas unidades de código como botones, texto e imágenes que tienen su propia lógica y apariencia llamadas componentes (React, s.f.).

Fue desarrollada por Facebook en el 2013 con la finalidad de facilitar la creación de componentes reutilizables e interactivos para las interfaces de usuario, actualmente soportado por la misma compañía y la comunidad de software libre en internet. Adicionalmente, también se utiliza en otras plataformas como Netflix, PayPal, Airbnb, Uber, Reddit y Twitter (Nextu, s.f.).

3.2.2 Javascript

JavaScript es un robusto lenguaje de programación que se puede aplicar a un documento HTML y usarse para crear interactividad dinámica en los sitios web. Fue inventado por Brendan Eich, cofundador del proyecto Mozilla Foundation y la Corporación Mozilla.

JavaScript por sí solo es bastante compacto, aunque muy flexible, y los desarrolladores han escrito gran cantidad de herramientas encima del núcleo del lenguaje JavaScript, desbloqueando una gran cantidad de funcionalidad adicional con un mínimo esfuerzo (Mozilla, s.f.).

3.2.3 Json

JavaScript Object Notation (JSON) es un formato basado en texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript. Es comúnmente utilizado para transmitir datos en aplicaciones web (por ejemplo: enviar algunos datos desde el servidor al cliente, así estos datos pueden ser mostrados en páginas web, o viceversa) (Mozilla, s.f.).

3.2.4 TypeScript

TypeScript es un lenguaje de programación de código abierto desarrollado y mantenido por Microsoft. Basado en JavaScript, TypeScript extiende su sintaxis y añade características adicionales que lo hace aún más robusto, coherente y limpio. Esto lo convierte en una opción perfecta para proyectos escalables que se ejecuten tanto del lado del cliente como del servidor (TypeScript, s.f.).

3.2.5 MongoDB

MongoDB es una base de datos NoSQL que utiliza un modelo de datos flexible basado en documentos. Permite la escalabilidad horizontal, facilita operaciones CRUD, y ofrece características como replicación y sharding para manejar grandes volúmenes de datos con escalabilidad de manera eficiente. Algunas de sus características son:

- El modelo de documento se asigna a los objetos en el código de su aplicación para facilitar el trabajo con los datos.
- Almacena datos en documentos flexibles similares a JSON, por lo que los campos pueden variar entre documentos y la estructura de datos puede cambiarse con el tiempo.
- MongoDB es de uso gratuito. Las versiones lanzadas antes del 16 de octubre de 2018 se publican bajo licencia AGPL. Todas las versiones posteriores al 16 de octubre de 2018, incluidos los parches lanzados para versiones anteriores, se publican bajo Licencia pública del lado del servidor (SSPL) v1 (MongoDB, s.f.).

3.2.6 Visual Studio Code

Visual Studio Code es un editor de código gratuito que cuenta con soporte para cualquier lenguaje de programación sin la necesidad de cambiar de editores de código donde una de sus mayores fortalezas además de estar disponible para Windows, macOS y Linux es contar con una amplia cantidad de extensiones y herramientas de todo tipo que ayudan a programar de forma más eficiente (Visual Studio Code, s.f.).

3.2.7 Nest

Nest (NestJS) es un framework para construir aplicaciones Node.js eficientes y escalables del lado del servidor. Está construido con TypeScript y es totalmente compatible con él (aunque sigue permitiendo a los desarrolladores codificar en JavaScript puro) y combina elementos de POO

(Programación Orientada a Objetos), PF (Programación Funcional) y FRP (Programación Reactiva Funcional).

Nest proporciona una arquitectura de aplicaciones lista para usar que permite a los desarrolladores y equipos crear aplicaciones altamente comprobables, escalables, poco acopladas y fáciles de mantener (Nest, s.f.).

3.2.8 Github

GitHub es un sitio “social coding” que permite subir repositorios de código para almacenarlo en el sistema de control de versiones Git. Se puede colaborar en proyectos de código, y el sistema es código abierto por defecto, lo que significa que cualquiera en el mundo puede encontrar tu código en GitHub, usarlo, aprender de él, y mejorarlo (Mozilla, s.f.).

3.3 Estado del Arte

Consecuentemente, algunas de las aplicaciones con mayor importancia en cuanto al sector comercial con énfasis en e-commerce.

3.3.1 Nacional

3.3.1.1 Rappi. Rappi es una compañía multinacional colombiana que actúa como plataforma de intermediación entre varios tipos de usuarios. Fue fundada en Colombia en 2015 por Simón Borrero, Sebastián Mejía y Felipe Villamarín y hoy está presente en 9 países de América Latina (México, Costa Rica, Colombia, Perú, Ecuador, Chile, Argentina, Uruguay y Brasil) y más de 200 ciudades. Tiene oficinas en Bogotá, São Paulo y la Ciudad de México.

Rappi ha sido definida como la SuperApp Latinoamericana y una de las empresas con un mayor crecimiento en la región. Sus fundadores fueron incluidos en la lista Bloomberg 50 de 2019,

reconociéndolos como emprendedores y líderes que han cambiado el panorama mundial de los negocios de maneras medibles en el último año (Azevedo, 2019).

3.3.1.2 Laika. Laika es la tienda online más completa para mascotas creada por pet lovers para pet lovers! Somos la tienda en línea de mascotas con mayor variedad de productos y servicios a domicilio en Colombia, contamos con más de 4.000 productos (alimentos, dietas naturales, snacks, medicinas, higiene y accesorios) y más de 17 servicios como (medicina veterinaria prepagada, veterinario a domicilio, vacunación, baños, funeraria, entrenador, gaticuidadores, información de colegios, fundaciones, entre otros. (Laika, s.f.).

3.3.2 Internacional

3.3.2.1 Amazon. Amazon, es una compañía estadounidense de comercio electrónico y servicios de computación en la nube a todos los niveles con sede en la ciudad estadounidense de Seattle; Su lema es “De la A a la Z” (Mateos, 2018.).

En la actualidad está totalmente diversificada y catalogada en diferentes líneas de productos, ofreciendo DVD, CD de música, software, videojuegos, electrónica, ropa, muebles, comida, libros, etc. Es la marca de venta al por menor más valiosa del mundo, según el índice BrandZ.

3.3.2.2 Mercado Libre. Mercado Libre es una empresa multinacional de origen argentino con sede en Montevideo, Uruguay y dedicada al comercio electrónico en Latinoamérica. Fundada en 1999, cuenta con operaciones en Argentina, Bolivia, Brasil, Chile, Colombia, Costa Rica, Ecuador, El Salvador, Guatemala, Honduras, México, Nicaragua, Panamá, Paraguay, Perú, República Dominicana, Uruguay y Venezuela. En 2022, ingresó en el ranking de las 100 marcas más valiosas del mundo, según Most Valuable Global Brands Kantar BrandZ, siendo la primera

compañía argentina en hacerlo. En 2023, la revista estadounidense Time incluyó a Mercado Libre en el listado de las 100 empresas más influyentes del mundo (Mercado Libre, s.f.).

Además de la plataforma de comercio electrónico Marketplace, Mercado Libre tiene otras cinco divisiones de negocio: Mercado Pago, Mercado Shops, Mercado Libre Publicidad y Mercado Envíos, ofrece soluciones para que tanto individuos como empresas puedan comprar, vender, anunciar, enviar y pagar bienes y servicios por Internet. Adicionalmente, Mercado Libre cuenta con aplicaciones y consultoras certificadas que permiten ampliar las funcionalidades existentes en este Marketplace.

3.3.2.3 Aliexpress. AliExpress representa una plataforma de comercio electrónico china que ofrece la venta al por menor o al por mayor a costos muy accesibles, con la posibilidad de realizar envíos a prácticamente cualquier parte del mundo. Es importante señalar que AliExpress no se clasifica como una tienda en línea convencional; más bien, opera como un intermediario que conecta a los vendedores o proveedores con los compradores.

Además, AliExpress asume la responsabilidad de supervisar las normas establecidas para los vendedores, garantizar la protección del comprador y actuar como intermediario en caso de cualquier inconveniente que pueda surgir (Vásquez, 2020.).

3.4 Antecedentes del Tema

El incremento en la integración de la tecnología en la rutina diaria de las personas ha favorecido la expansión de los mercados digitales, es así como en las diversas aplicaciones cotidianas surgen constantemente nuevas marcas, promociones, eventos y otros elementos que captan la atención de los usuarios. La mayoría de las empresas que ofrecen servicios han optado por incorporar herramientas tecnológicas para acercarse más a sus clientes, generando así nuevas

oportunidades de negocio para los proveedores de servicios, esto inspira la exploración de nuevas ideas de negocio para impactar a las personas interesadas. Actualmente, en Colombia, son escasas las empresas nacionales que utilizan servicios tecnológicos para la contratación de personal capacitado en el cuidado de mascotas, siendo este un mercado en auge.

3.4.1 Rover

Rover es una app enfocada en cuidado de mascotas y paseo de perros. Donde se encuentran cuidadores de mascotas de confianza en tu zona que ofrecen alojamiento, guardería de día o paseos. La app Rover ha sido creada por y para amantes de los perros (The Dog People TM). Mediante la app puedes recibir fotos y noticias de tu mascota, un registro mediante GPS del paseo de tu perro, comunicarte fácilmente con tu cuidador, reservar y pagar de forma segura o, si eres cuidador, gestionar tus reservas y pagos cómodamente (Rover, s.f.).

3.4.2 Petcrew

Petcrew es una plataforma web colombiana creada en el año 2017 gracias a un concurso de emprendimiento realizado por Mintic, que permite el servicio de hospedaje y paseos de perros en la ciudad de Bogotá, Cali y Medellín, con precios preestablecidos por la misma aplicación donde el usuario escoge quién quiere que le ofrezca el servicio y el cuidador confirma la solicitud donde los dos coordinaran las horas de entrega y recogida (PetCrew, s.f.).

3.4.3 Ohana

Es un sitio web donde se define el “Pet Sitter” como el acto de cuidar a un animal de compañía en su propio hogar mientras el propietario está ausente. Se usa más para el cuidado de felinos. En el caso del cuidado de caninos, se puede utilizar cuando el animal de compañía no se puede trasladar a una guardería para perros, tales como celo, sin castrar, menor edad de la permitida.

Como lo indica la definición anterior, los cuidadores se preocupan por los animales de compañía en los hogares de los animales. Los tutores de animales de compañía a menudo utilizan el servicio de cuidado de animales de compañía cuando se van de vacaciones, viajan por negocios, trabajan largas horas o cuando están incapacitados por algún motivo para cuidar a su animal de compañía (Ohana, s.f.). Actualmente, solo opera en la ciudad de Medellín.

3.4.4 Wag!

Wag! Fundada en 2014, es una aplicación diseñada para la solicitud de servicios de paseo de perros, ofreciendo a las personas la posibilidad de reservar paseadores de perros para salidas ocasionales o citas programadas. Promocionado como el “Uber para perros”, La plataforma web y la aplicación facilitan a los dueños de perros la búsqueda de cuidadores cualificados para sus mascotas. Además, esta aplicación ha brindado a individuos apasionados por los perros la oportunidad de generar ingresos adicionales al desempeñarse como paseadores de perros, siendo una de las usadas en los estados unidos (Wag, s.f.).

4. Metodología

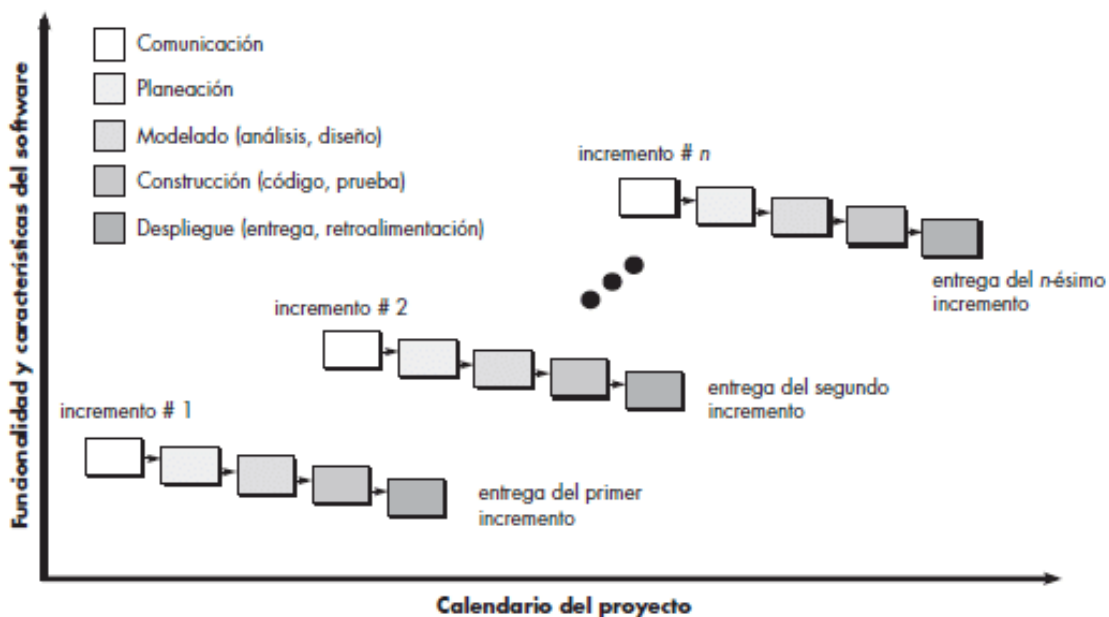
La ejecución de este proyecto abarcará diversas etapas que serán planificadas, ajustadas y evaluadas mediante una metodología adaptada a las especificidades que demanda este tipo de proyecto, además de considerar la gestión eficiente del tiempo.

4.1 Metodología Incremental

Para este proyecto se utilizó la metodología incremental propuesta por Harlan Mills, la cual se concentra en abordar un conjunto inicial de requisitos, añadiendo progresivamente nuevas funcionalidades en versiones posteriores hasta obtener el producto final viable, como se ilustra en la siguiente imagen:

Figura 1

Metodología incremental.

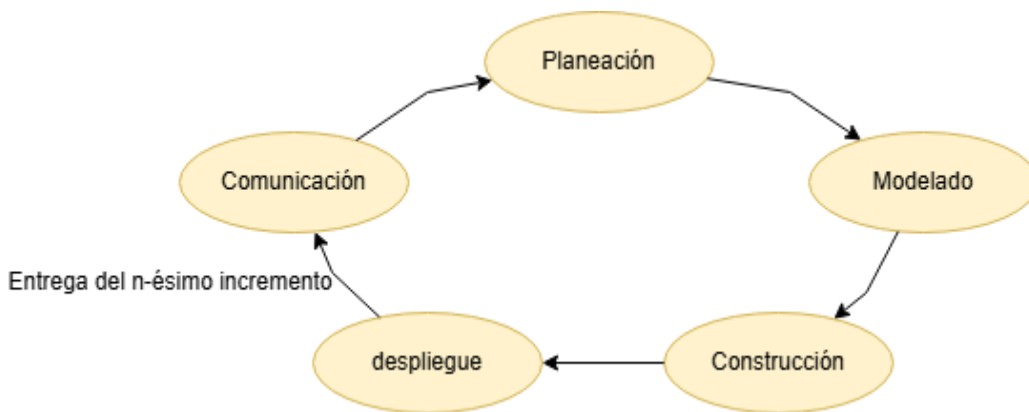


Nota. El gráfico representa el desarrollo de software mediante el modelo incremental. Tomado de R. Pressman, Ingeniería del software: Un enfoque Práctico. 7 Ed. McGraw Hill, 2010 (Pressman, 2010).

Este ciclo iterativo se enfoca en cinco componentes esenciales: comunicación, planeación, modelado, construcción de código y ejecución de pruebas, adaptándose a los incrementos necesarios en cada avance del proyecto. Por esta razón, las cinco fases mencionadas se abordan de manera conjunta y se iteran progresivamente en cada etapa de desarrollo como se describe a continuación:

Figura 2

Etapas metodología incremental.



Nota. El gráfico representa las fases del desarrollo por las cuales se seguirá mejorando hasta tener una entrega viable.

4.1.1 Comunicación

En la primera etapa se definió la idea principal del prototipo web, orientada en ofrecer servicios de cuidado y atención veterinaria para mascotas caninas. Posteriormente, durante la fase de despliegue del primer prototipo, se formularon preguntas a un grupo de cinco personas pertenecientes a la comunidad universitaria de la Universidad Industrial de Santander, pertenecientes a diferentes estratos socioeconómicos con el propósito de conocer aspectos básicos de su perfil, como la edad, la ocupación y si poseían mascotas. Estas personas fueron seleccionadas

con el fin de obtener una perspectiva del contexto urbano, en el marco de una práctica académica desarrollada en la Universidad Industrial de Santander. Esta información permitió comprender mejor las necesidades de los posibles usuarios y considerar sus comentarios para el desarrollo y mejora del segundo prototipo.

Esto se refleja en las preguntas aplicadas durante la evaluación focal del primer y segundo prototipo, las cuales se presentan en los Apéndices A y B.

4.1.2 Planeación

Es fundamental comprender qué se va a desarrollar, el propósito del prototipo, así como especificar los requerimientos y elementos necesarios. También resulta esencial analizar el impacto que el proyecto puede generar en la sociedad, además de realizar un estudio del framework React para el desarrollo del frontend y de NestJS para la implementación del backend, junto con los diseños de las interfaces de usuario.

4.1.3 Modelado

Una de las características principales de cualquier plataforma es su usabilidad, por esta razón, en el primer diseño se priorizó la funcionalidad y, a partir del concepto inicial de los servicios a ofrecer, se elaboraron los diagramas de clases, de flujo y de casos de uso, avanzando así hacia la siguiente fase del desarrollo.

En el caso del segundo prototipo, se tomaron en consideración los resultados obtenidos de las preguntas de acercamiento inicial indicadas en el apéndice A y las preguntas de evaluación aplicadas al primer prototipo en el apéndice B. Con base en la información recolectada en el

apéndice B se propuso ajustar los servicios planteados inicialmente y extender los diagramas de casos de uso.

4.1.4 Construcción

Esta fase se centró en la implementación y construcción del código funcional de la aplicación, centrándose en la creación de la interfaz de usuario, basado en lo realizado en los pasos anteriores, donde se elaboran los diagramas de clases y el diseño de la base de datos. Con el objetivo de obtener un producto funcional, con bases para la creación de una versión final que cumpla los objetivos establecidos.

Se definió la estructura de la base de datos del prototipo, lo que permitió avanzar en el desarrollo de la interfaz de usuario y de funcionalidades como el registro, la solicitud y aceptación de servicios, entre otras. Para la entrega del prototipo final, se ajustaron las interfaces de usuario y se añadieron nuevas funciones, el seguimiento del estado de solicitudes y la implementación de un historial. Finalmente, se toma en consideración el rol de administrador, que permite modificar las descripciones de los servicios existentes y gestionar su creación o eliminación según sea necesario.

4.1.5 Despliegue

Esta fase final, el producto funcional es sometido a una serie de pruebas con el objetivo de recopilar evaluaciones del grupo piloto de usuarios. Todas las opiniones, recomendaciones, críticas y mejoras por parte de los usuarios serán tomadas en cuenta para la siguiente iteración. Posteriormente, se procede a realizar la siguiente iteración para cada una de las fases hasta obtener un prototipo funcional viable que pueda cumplir los requerimientos.

Ambos prototipos fueron sometidos a pruebas funcionales realizadas por el programador, y con la colaboración del grupo de usuarios se recolectó información, que se encuentra en el Apéndice B, para mejorar y agregar funcionalidades. Al completar el prototipo final, se aplicaron nuevas preguntas a una muestra más amplia conformada por 30 personas residentes en la ciudad de Bucaramanga, seleccionadas bajo criterios de diversidad socioeconómicas. Las personas seleccionadas para esta muestra pertenecían a los barrios de San Alonso, Álvarez, Antonia Santos y Cabecera, entre las cuales se encontraban un médico veterinario, doce cuidadores y diecisiete propietarios de mascotas. Los resultados de esta fase, presentados en el Apéndice C, ofrecen información valiosa que puede ser considerada para el desarrollo de futuras versiones del prototipo web. Los resultados obtenidos en esta evaluación permitieron identificar oportunidades de mejora en la interfaz y en la experiencia de usuario, consolidando el prototipo como una base sólida para futuras iteraciones y posibles implementaciones a mayor escala.

5. Desarrollo del Proyecto

5.1 Lista de Requerimientos

5.1.1 Requerimientos Funcionales

Los requerimientos funcionales especifican las actividades que el sistema debe llevar a cabo, es decir, las funciones generales que el software debe ofrecer. Los requerimientos funcionales de este proyecto se especifican en la siguiente tabla:

Tabla 1

Requerimientos funcionales.

Identificador	Nombre del requerimiento
RF00	Registro de usuario
RF01	Iniciar sesión
RF02	Cerrar sesión
RF03	Ver perfil
RF04	Editar perfil
RF05	Ver servicios
RF06	Solicitar un servicio
RF07	Agregar calificaciones
RF08	Cancelar un servicio
RF09	Ver historial de servicios
RF10	Ver perfil mascota

RF11	Modificar perfil mascota
RF12	Agregar vacunas a mascota

5.1.2 Requerimientos no Funcionales

Los requerimientos no funcionales son aquellos que representan las características generales del sistema con los que se puede valorar la operación del software. Los requerimientos no funcionales de este proyecto son los siguientes:

Tabla 2

Requerimientos no funcionales.

Identificador	Nombre del requerimiento
RNF00	Diseño de interfaz de usuario
RNF01	Tener buena visibilidad en navegadores web
RNF02	Integridad de los datos almacenados, confiables y exactos
RNF03	Tiempos de respuesta cortos

5.2 Usuarios

Tabla 3

Usuarios potenciales.

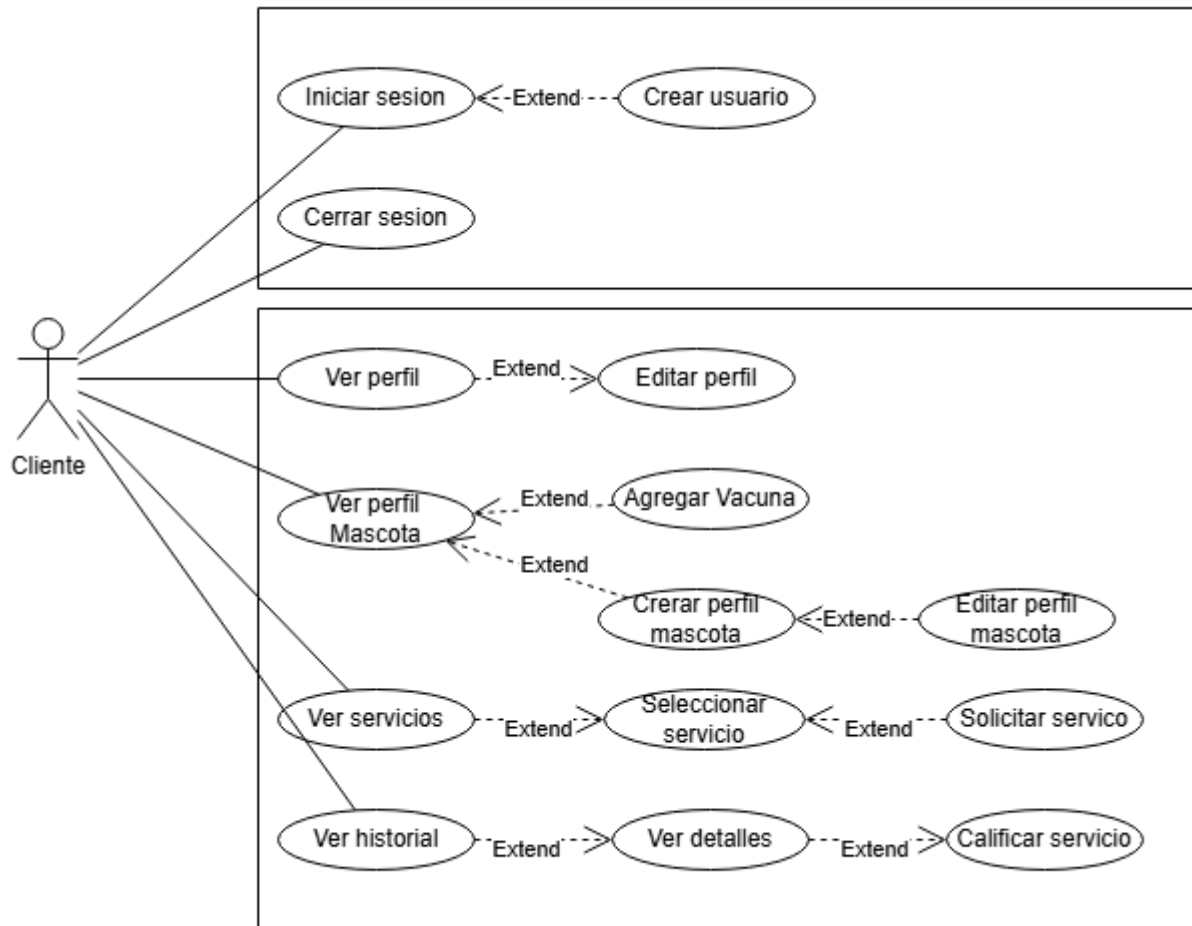
Identificador	Usuarios
UP00	Clientes
UP01	Cuidadores mayores de edad
UP02	Veterinarios

5.3 Diagramas de Casos de Uso

Un diagrama de caso de uso se emplea para el análisis y diseño del sistema, además de representar su funcionalidad, esclareciendo qué acciones realizará el sistema. Están enfocados en describir el comportamiento del sistema desde una perspectiva externa.

Figura 3

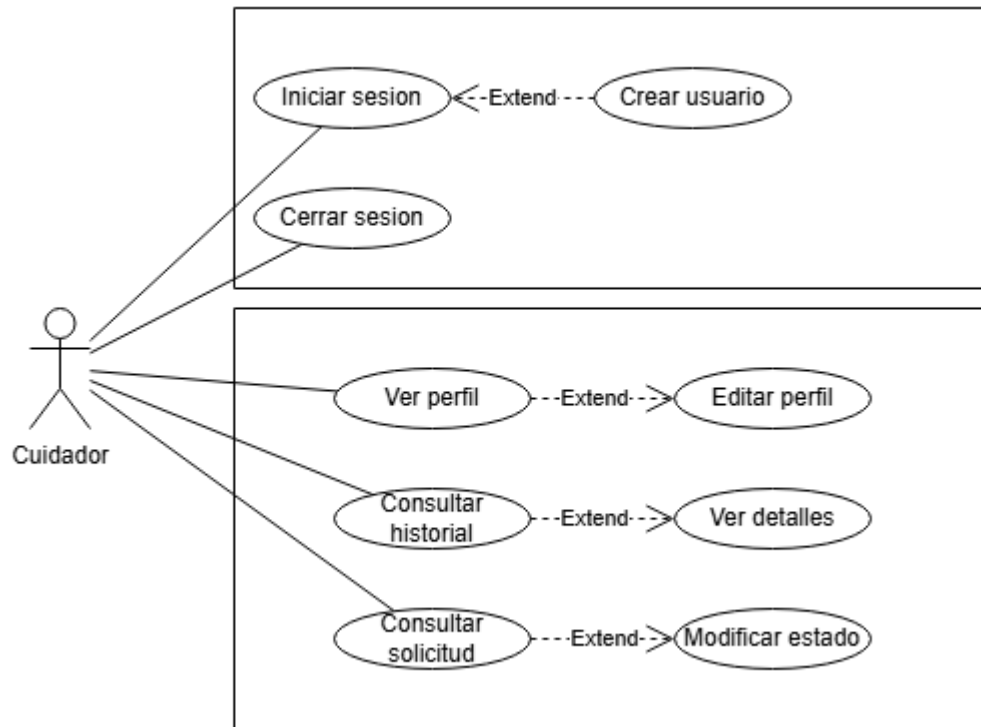
Diagrama de caso de uso para el usuario cliente.



Nota. El diagrama representa el caso de uso para los usuarios registrados con el rol de cliente.

Figura 4

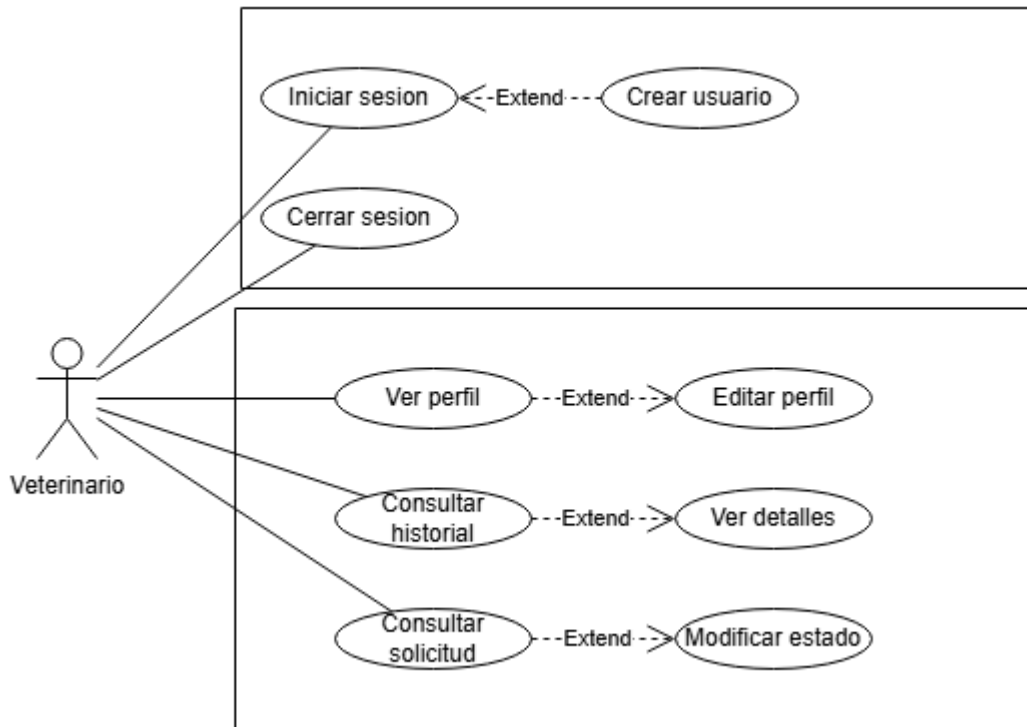
Diagrama de caso de uso para el usuario cuidador.



Nota. El diagrama representa el caso de uso para los usuarios registrados con el rol de Cuidador.

Figura 5

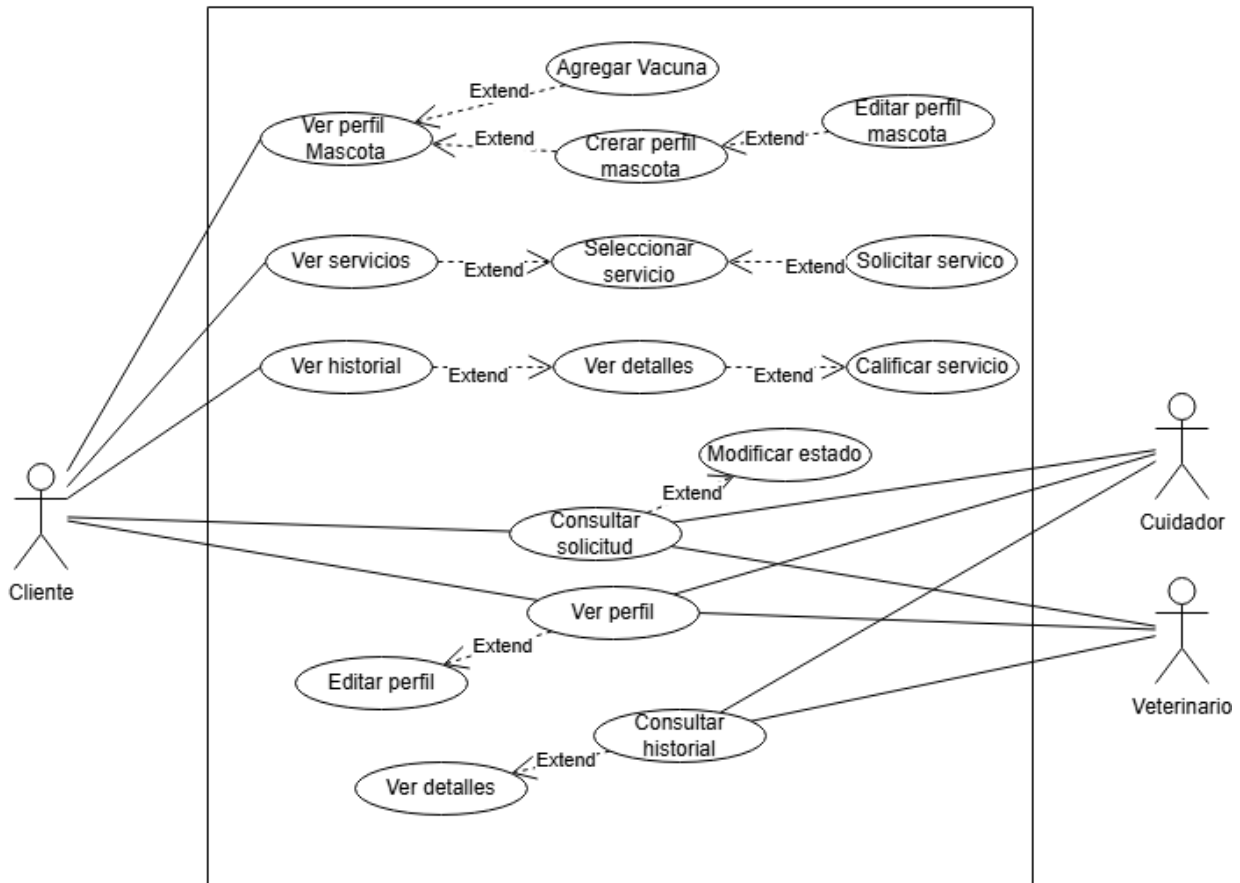
Diagrama de caso de uso para el usuario veterinario.



Nota. El diagrama representa el caso de uso para los usuarios registrados con el rol de Veterinario.

Figura 6

Diagrama de caso de uso para el usuario (cliente, cuidador y veterinario).



Nota. El diagrama representa el caso de uso general donde interactúan los tres tipos de usuarios del prototipo.

5.4 Casos de Uso

Los casos de uso ofrecen una estructura para expresar los requisitos funcionales dentro del contexto del sistema. Reflejan cómo se utiliza el software y los resultados esperados, esclareciendo qué acciones realizará el sistema.

En este proyecto, se desarrollaron los siguientes casos de uso para ilustrar la funcionalidad del software y cómo interactúan los diferentes actores con él.

Tabla 4

Caso de uso Registro usuario.

Caso de uso: Registro de usuario
Descripción: Realizar registro como nuevo usuario y permite hacer uso de todas las funcionalidades que se ofrecen
Actores: Usuario (Clientes – Cuidadores – Veterinario)
Precondiciones: El usuario debe ser mayor de edad, disponer de un correo electrónico y número telefónico para proceder con el registro
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario entra mediante un navegador al prototipo web 2. El usuario selecciona la opción de registro 3. Selecciona su rol (Clientes – Cuidadores – Veterinario) 4. El usuario ingresa los datos solicitados aceptando los términos y condiciones de uso del prototipo 5. El usuario oprime el botón registrarme notificando un registro satisfactorio
Flujo alternativo:
<ol style="list-style-type: none"> 1. Se notifica al usuario que los datos suministrados no son válidos, están incompletos o faltan campos por llenar, permitiendo corregir o agregar datos faltantes para completar el registro
Postcondiciones: El usuario se registra y se redirige a la página de inicio
<i>Nota.</i> Esta tabla hace referencia al proceso de registro en el prototipo web.

Tabla 5

Caso de uso Inicio de sesión.

Caso de uso: Inicio de sesión
Descripción: Permite al usuario ingresar al prototipo web mediante una autenticación
Actores: Usuario (Clientes – Cuidadores - Veterinario)
Precondiciones: El usuario ya debe estar registrado en el sistema contando con sus credenciales de acceso
Flujo normal: <ol style="list-style-type: none"> 1. El usuario entra mediante un navegador al prototipo web 2. El usuario ingresa sus credenciales de acceso 3. El usuario oprime el botón iniciar sesión 4. El sistema realiza la verificación de usuario 5. El usuario ingresa al prototipo web
Flujo alternativo: <ol style="list-style-type: none"> 1. Se notifica al usuario que los datos suministrados no son válidos, están incompletos o faltan campos por llenar, permitiendo corregir o agregar datos faltantes para completar el registro 2. Se notifica al usuario y se da la opción de ingresar las credenciales u olvido su contraseña en caso de haberla olvidado
Postcondiciones: El usuario ingresa y se redirige a la página de inicio
<i>Nota.</i> Esta tabla hace referencia al proceso de inicio de sesión en el prototipo web.

Tabla 6

Caso de uso Editar perfil.

Caso de uso: Editar perfil
Descripción: Permite al usuario editar la información de su perfil
Actores: Usuario (Clientes – Cuidadores - Veterinario)
Precondiciones: El usuario ya ha iniciado sesión en el prototipo
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario cliquea el botón ‘Perfil’ en la barra lateral izquierda 2. El sistema muestra la información del perfil 3. El usuario cliquea el botón ‘Modificar perfil’ 4. El usuario ingresa los datos que quiere cambiar de su perfil 5. El usuario confirma los cambios realizados 6. Se realiza la actualización en la base de datos
Flujo alternativo:
<ol style="list-style-type: none"> 1. Se notifica al usuario que los datos suministrados no son válidos, están incompletos o faltan campos por llenar, permitiendo corregir o agregar datos faltantes.
Postcondiciones: El usuario observa los cambios realizados
<i>Nota.</i> Esta tabla hace referencia al proceso de edición de datos personales por parte del usuario.

Tabla 7

Caso de uso ver servicios.

Caso de uso: Ver servicios
Descripción: Permite al usuario consultar los diferentes tipos de servicios
Actores: Usuario (Clientes)
Precondiciones: El usuario ya ha iniciado sesión en el prototipo
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario observa los servicios en la página principal del prototipo 2. El usuario selecciona un servicio para observar más información
Flujo alternativo:

-
1. Por errores de conexión el tiempo de respuesta es tardío
-

Postcondiciones: El usuario selecciona el servicio que desea solicitar

Nota. Esta tabla permite ver el flujo por parte del usuario para consultar los servicios.

Tabla 8

Caso de uso Enviar solicitud de servicio.

Caso de uso: Enviar solicitud de servicio

Descripción: Permite al usuario realiza una solicitud de servicio

Actores: Usuario (Clientes)

Precondiciones: El usuario ya ha iniciado sesión en el prototipo, ha buscado los tipos de servicio y consultado sus detalles

Flujo normal:

1. El usuario observa los detalles del servicio a solicitar
 2. El usuario selecciona cliqueea ‘solicitar’
 3. Posteriormente, se solicitan los campos requeridos para la solicitud
 4. El usuario llena los campos
 5. El usuario define fecha y hora del servicio
 6. El usuario ingresa la dirección donde se llevará a cabo el servicio
 7. El usuario puede agregar información adicional dentro del campo ‘Observaciones’
 8. El sistema muestra un resumen del servicio a solicitar
 9. El usuario cliqueea en ‘Solicitar’
-

Flujo alternativo:

1. Se notifica al usuario que los datos suministrados no son válidos, están incompletos o faltan campos por llenar, permitiendo corregir o agregar datos faltantes.
-

Postcondiciones: El usuario puede consultar la solicitud, en la barra lateral ‘Servicios activos’

Nota. Esta tabla indica el proceso de solicitud de un servicio.

Tabla 9

Caso de uso Aceptar solicitud de servicios.

Caso de uso: Aceptar solicitud de servicios
Descripción: Permite al usuario realiza una solicitud de servicios
Actores: Usuario (cuidadores - Veterinario)
Precondiciones: El usuario ya ha iniciado sesión en el prototipo, en la barra lateral de la página principal en la opción ‘Servicios disponibles’ encontrará la lista de solicitudes que aún no han sido aceptadas
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario cliquea en la lista de ‘servicios disponibles’ en la barra lateral izquierda 2. El usuario selecciona el servicio que desea aceptar 3. El sistema le muestra al usuario la información del servicio seleccionado 4. El usuario cliquea ‘Aceptar’ 5. La solicitud es aprobada y el sistema la agrega a la sección de servicios activos
Flujo alternativo:
<ol style="list-style-type: none"> 6. La conexión es tardía, el usuario debe esperar a que su solicitud sea procesada 7. El usuario puede seguir consultado las solicitudes de servicio
Postcondiciones: El usuario puede consultar la solicitud en la barra lateral ‘Servicios activos’

Nota. Esta tabla permite ver el proceso de aceptación de servicios.

Tabla 10

Caso de uso Consultar solicitud de servicio.

Caso de uso: Consultar solicitud de servicio
Descripción: Permite al usuario consultar los detalles del servicio
Actores: Usuario (Clientes – Cuidadores - Veterinario)
Precondiciones: El usuario ya ha iniciado sesión en el prototipo y previamente haber hecho una solicitud o haber aceptado una solicitud de servicio
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario cliquea el botón ‘servicios activos’ en la barra lateral izquierda 2. El usuario busca la solicitud que desea consultar 3. El usuario cliquea en ‘Mas información’ 4. El sistema muestra un resumen de la solicitud
Flujo alternativo:
<ol style="list-style-type: none"> 1. El usuario puede modificar el estado de la solicitud del servicio aceptado
Postcondiciones: El usuario puede volver a consultar otras solicitudes
<i>Nota.</i> Esta tabla hace referencia al proceso para consultar los detalles de las solicitudes de servicios.

Tabla 11

Caso de uso Cancelar un servicio.

Caso de uso: Cancelar un servicio
Descripción: Permite al usuario cancelar el servicio solicitado
Actores: Usuario (Clientes)
Precondiciones: El usuario ya ha realizado una solicitud de servicio y aún no ha sido aceptada
Flujo normal: <ol style="list-style-type: none"> 1. El usuario cliquea el botón ‘servicios activos’ en la barra lateral izquierda 2. El usuario busca la solicitud que desea cancelar 3. El usuario cliquea en ‘Cancelar’ 4. El sistema muestra un resumen de la solicitud 5. El usuario cliquea en cancelar solicitud
Flujo alternativo: <ol style="list-style-type: none"> 1. El usuario puede volver a solicitar un servicio
Postcondiciones: El usuario puede ver el servicio cancelado en ‘Historial de solicitudes’
<i>Nota.</i> Esta tabla hace referencia al proceso para cancelar una solicitud de servicio que aún no ha sido aceptada.

Tabla 12

Caso de uso Finalizar un servicio.

Caso de uso: Finalizar un servicio
Descripción: Permite al usuario finalizar el servicio aceptado
Actores: Usuario (cuidadores - veterinario)
Precondiciones: El usuario ya ha aceptado una solicitud de servicio y la ha realizado satisfactoriamente
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario cliquea el botón ‘servicios activos’ en la barra lateral izquierda 2. El usuario busca la solicitud que desea finalizar 3. El usuario cliquea en ‘Mas información’ 4. El sistema muestra un resumen de la solicitud 5. El usuario puede agregar historia médica 6. El usuario cliquea en finalizar solicitud
Flujo alternativo:
<ol style="list-style-type: none"> 1. El usuario puede aceptar más solicitudes de servicio
Postcondiciones: El usuario puede ver el servicio finalizado en ‘Historial de solicitudes’
<i>Nota.</i> Esta tabla hace referencia al proceso para finalizar una solicitud de servicio.

Tabla 13

Caso de uso Consultar historial de servicios.

Caso de uso: Consultar historial de servicios
Descripción: Permite al usuario consultar el historial de servicios
Actores: Usuario (clientes - cuidadores - veterinario)
Precondiciones: El usuario ya ha iniciado sesión y el servicio solicitado ya ha sido finalizado
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario cliquea el botón ‘Historial de servicios’ en la barra lateral izquierda 2. El usuario busca la solicitud que desea consultar 3. El usuario cliquea en ‘Mas información’ 4. El sistema muestra un resumen de la solicitud
Flujo alternativo:
<ol style="list-style-type: none"> 1. El usuario puede realizar más solicitudes de servicio
Postcondiciones: El usuario puede consultar otro servicio

Nota. Esta tabla hace referencia al proceso para consultar una solicitud de servicio.

Tabla 14

Caso de uso Calificar un servicio.

Caso de uso: Calificar un servicio
Descripción: Permite al usuario calificar el servicio recibido por un cuidador o veterinario
Actores: Usuario (clientes)
Precondiciones: El usuario ya ha iniciado sesión y el servicio solicitado ya ha sido finalizado
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario cliquea el botón ‘Historial de servicios’ en la barra lateral izquierda 2. El usuario busca la solicitud que desea consultar 3. El usuario cliquea en ‘Mas información’ 4. El sistema muestra un resumen de la solicitud 5. El usuario cliquea la opción ‘Calificar’
Flujo alternativo:

-
1. El usuario puede realizar más solicitudes de servicio
-

Postcondiciones: El usuario puede consultar otro servicio

Nota. Esta tabla hace referencia al proceso para calificar una solicitud de servicio.

Tabla 15

Caso de uso Agregar perfil de mascota.

Caso de uso: Agregar perfil de mascota

Descripción: Permite al usuario agregar información sobre su mascota

Actores: Usuario (Clientes)

Precondiciones: El usuario ya ha iniciado sesión en el prototipo

Flujo normal:

1. El usuario cliquea el botón ‘Mi mascota’ en la barra lateral izquierda
 2. El usuario cliquea en agregar mascota
 3. A continuación, el sistema muestra los campos necesarios para identificar una mascota
 4. El usuario llena los campos
 5. El usuario cliquea en ‘Añadir mascota’
-

Flujo alternativo:

1. La conexión es tardía, el usuario debe esperar a que su solicitud sea procesada
-

Postcondiciones: El usuario puede modificar el perfil de su mascota en la barra lateral ‘Mi mascota’

Nota. Esta tabla hace referencia al proceso para añadir una mascota.

Tabla 16

Caso de uso Editar perfil de mascota.

Caso de uso: Editar perfil de mascota
Descripción: Permite al usuario editar información sobre su mascota
Actores: Usuario (Clientes)
Precondiciones: El usuario ya ha iniciado sesión en el prototipo
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario cliquea el botón ‘Mi mascota’ en la barra lateral izquierda 2. El sistema muestra la información de la mascota 3. El usuario cliquea en modificar perfil 4. El usuario modifica los campos 5. El usuario confirma las modificaciones 6. Se realiza la actualización en la base de datos
Flujo alternativo:
<ol style="list-style-type: none"> 7. Se le notifica el usuario que los datos suministrados no son válidos, están incompletos o faltan campos por llenar, permitiendo corregir o agregar datos faltantes
Postcondiciones: El usuario observa los cambios realizados
<i>Nota.</i> Esta tabla hace referencia al proceso para modificar el perfil de una mascota.

Tabla 17

Caso de uso Agregar vacunas a mascota.

Caso de uso: Agregar vacunas a mascota
Descripción: Permite al usuario agregar vacunas a una mascota
Actores: Usuario (Clientes)
Precondiciones: El usuario ya ha iniciado sesión en el prototipo
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario cliquea el botón ‘Mi mascota’ en la barra lateral izquierda 2. El sistema muestra la información de la mascota 3. El usuario cliquea en ‘Plan de vacunación’ 4. El usuario cliquea en ‘Añadir vacuna’ 5. El usuario llena los campos 6. El usuario cliquea en ‘Agregar vacuna’
Flujo alternativo:
<ol style="list-style-type: none"> 1. La conexión es tardía, el usuario debe esperar a que su solicitud sea procesada
Postcondiciones: El usuario observa los cambios realizados

Nota. Esta tabla hace referencia al proceso para agregar vacunas a una mascota.

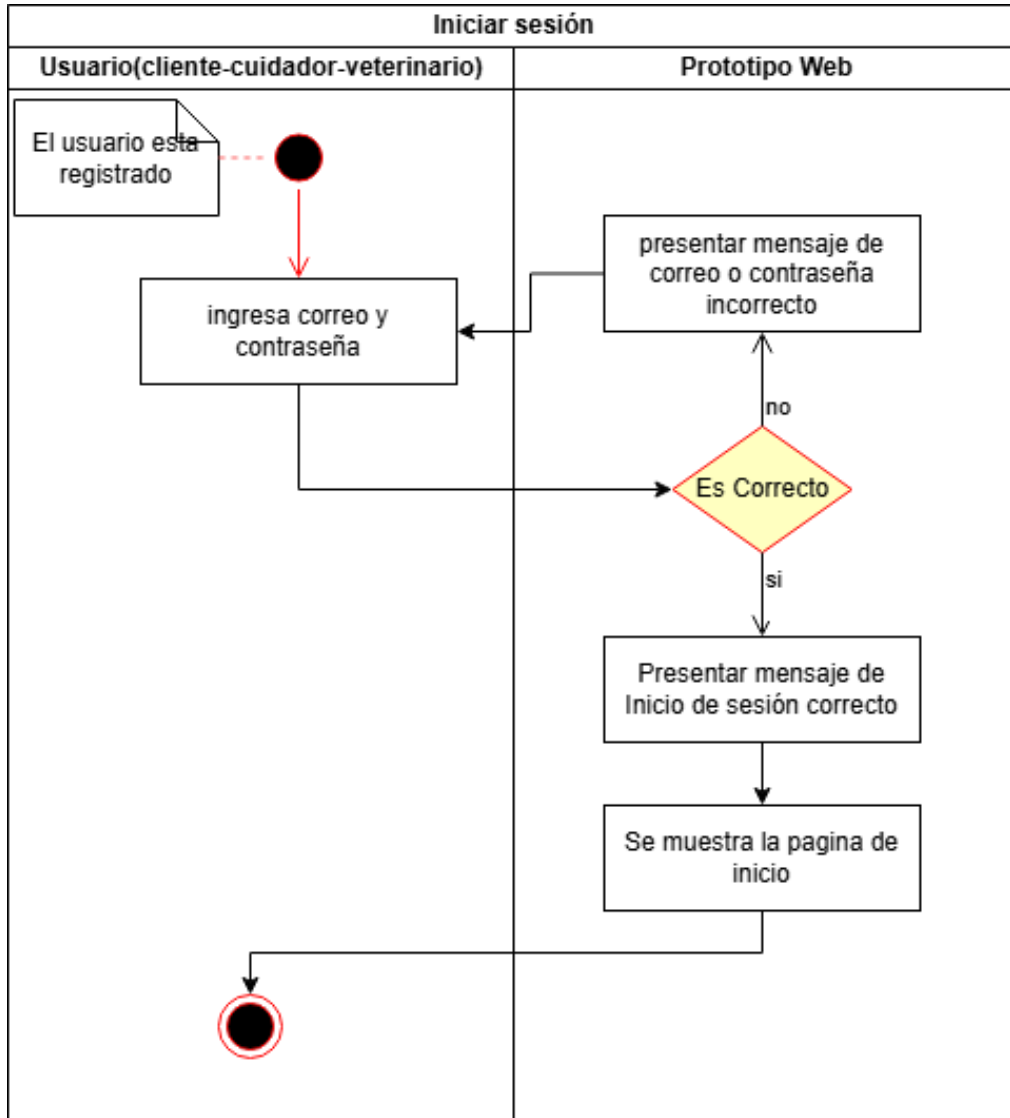
5.5 Diagrama de Actividades

El diagrama de actividades es una representación gráfica de un proceso, que ilustra cómo los eventos de un caso de uso de un sistema se conectan mostrando cómo se realizan y en qué orden.

A continuación, se presentan los principales diagramas de actividades desarrollados para este proyecto.

Figura 7

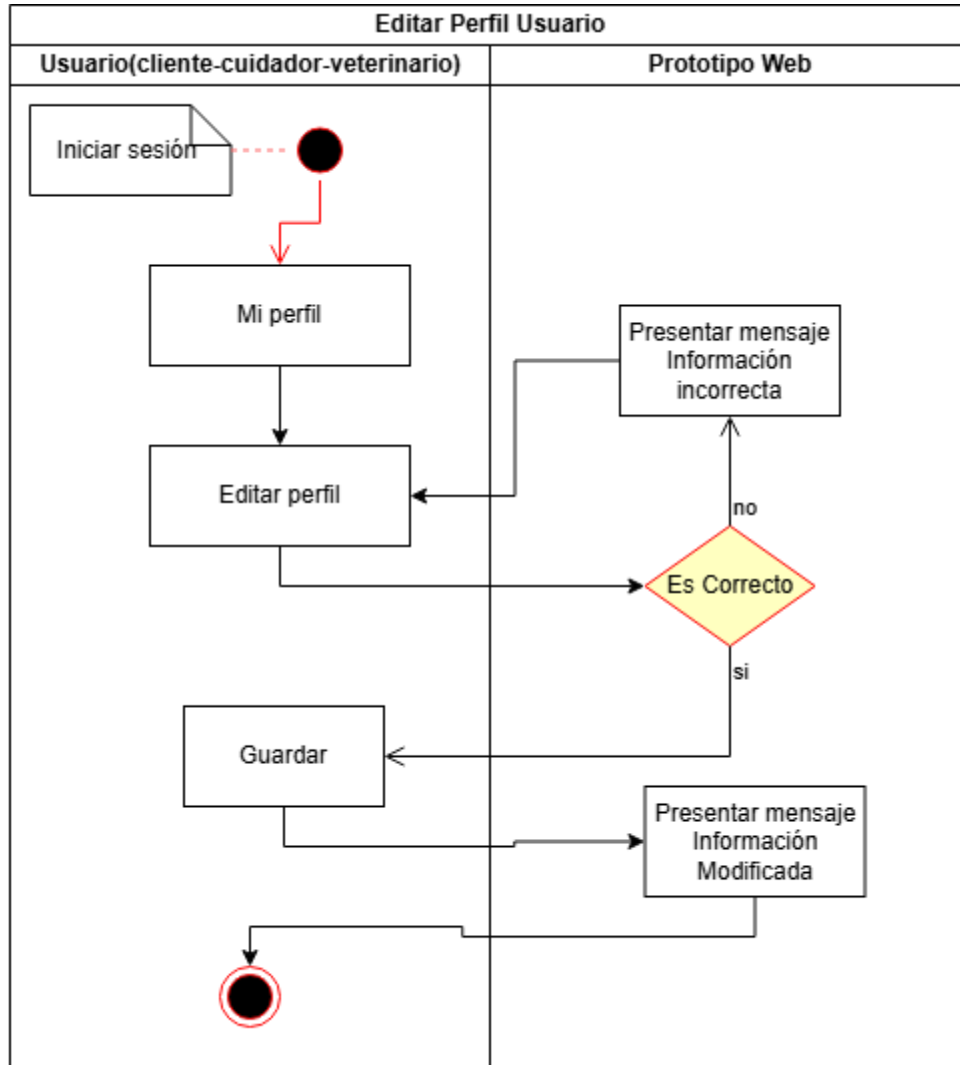
Diagrama de actividad iniciar sesión.



Nota. Este diagrama representa la secuencia para iniciar sesión en el sistema.

Figura 8

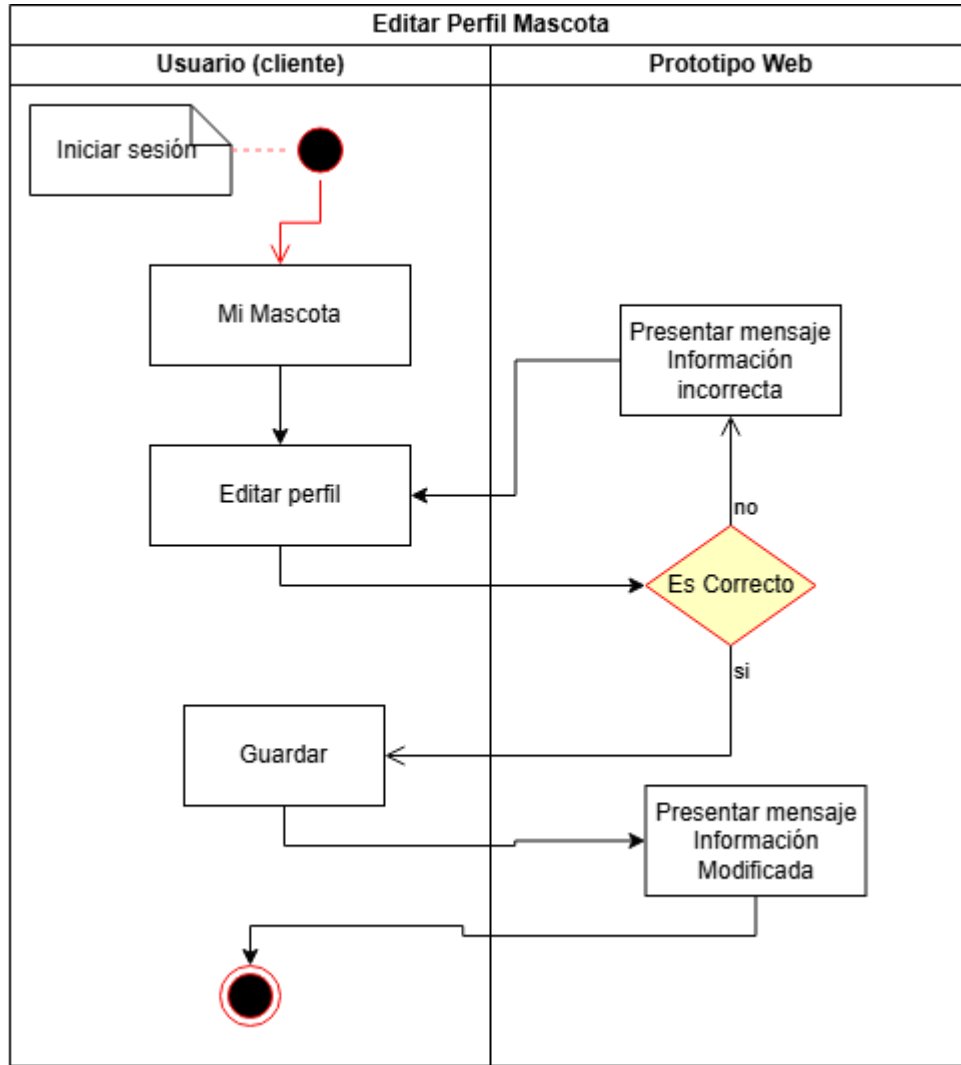
Diagrama de actividad editar perfil.



Nota. Este diagrama representa la secuencia para editar la información en el perfil del usuario.

Figura 9

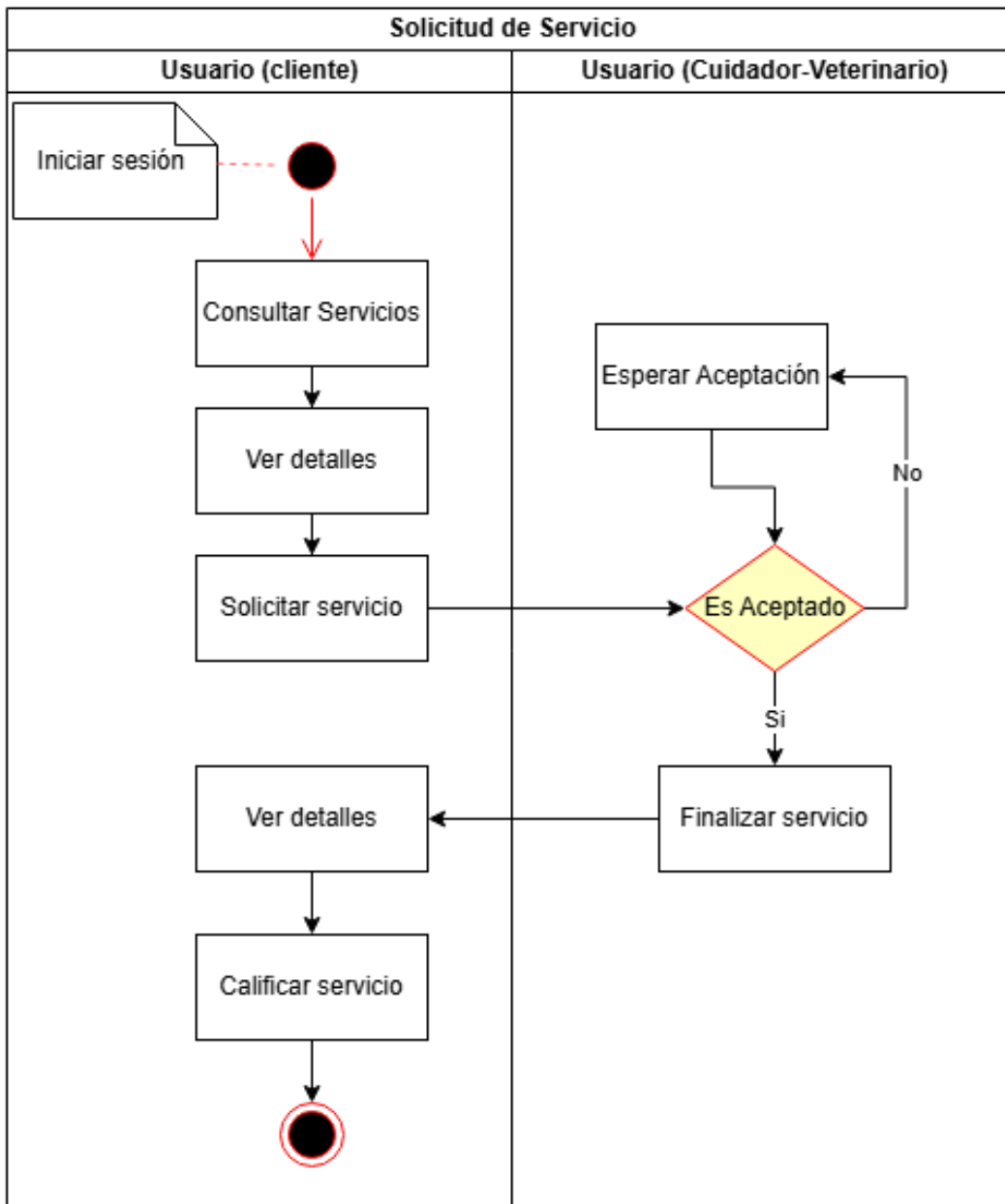
Diagrama de actividad editar perfil mascota.



Nota. Este diagrama representa la secuencia para editar la información en el perfil de mascota.

Figura 10

Diagrama de actividad solicitud se servicio.



Nota. Este diagrama representa la secuencia para solicitar un servicio.

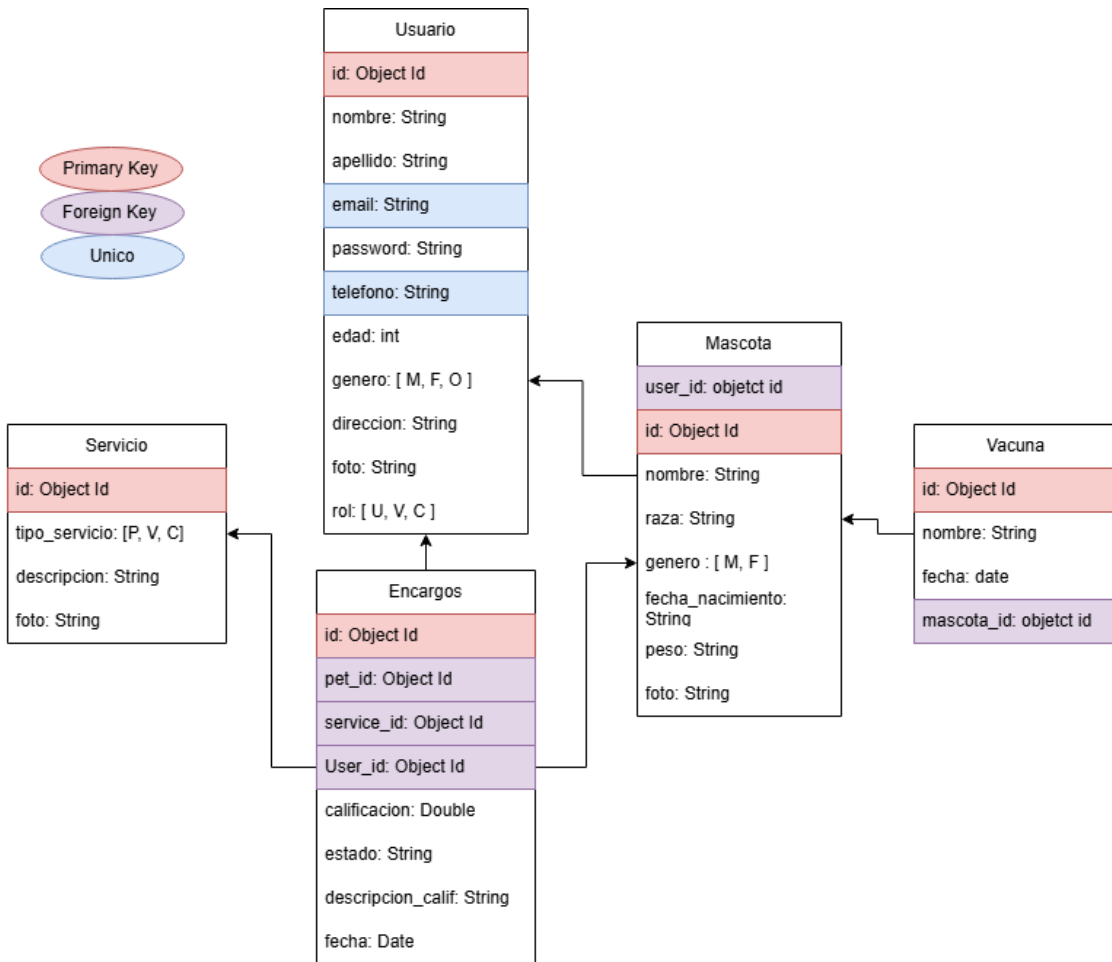
5.6 Diagramas de Clases

El diagrama de clases se usa para describir la estructura del sistema.

5.6.1 Modelo no SQL

Figura 11

Modelo no SQL de la base de datos.



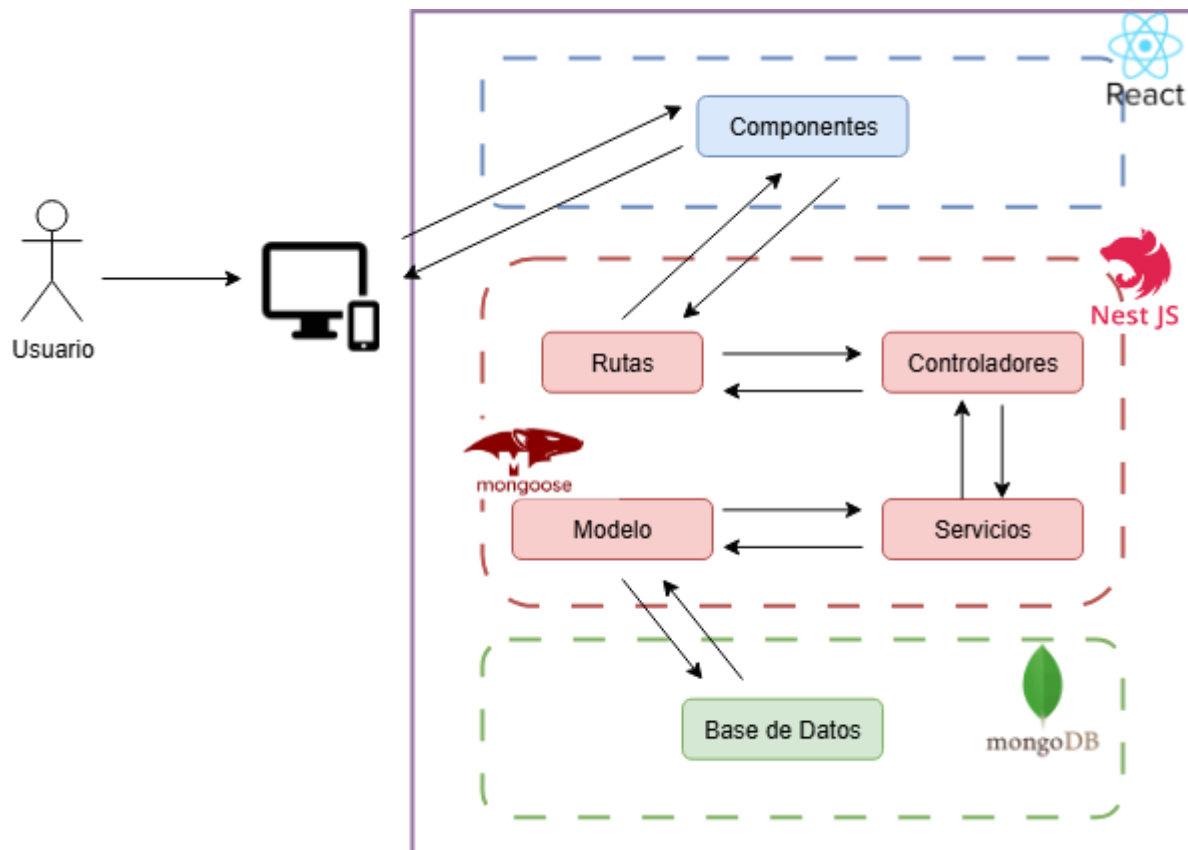
Nota. Aunque en el desarrollo del proyecto se implementó una base de datos NoSQL, este modelo relacional se presenta únicamente para facilitar la comprensión de la estructura general de los datos y las relaciones entre las entidades.

5.7 Arquitectura de software

A continuación, se presenta el diseño de la arquitectura utilizada para el desarrollo del proyecto. En este esquema se evidencian los componentes que conforman el prototipo web y la manera en que se relacionan entre sí a través del flujo de datos y los procesos involucrados.

Figura 12

Diagrama de arquitectura de software.



Como se observa en la Figura 12, el proceso inicia cuando un usuario accede al prototipo mediante un navegador web, donde interactúa directamente con la interfaz gráfica desarrollada en React, desde donde realiza diversas acciones tales como el inicio de sesión, el registro de usuarios, la solicitud de servicios. Estas acciones generan peticiones HTTP (GET, POST, PUT o DELETE)

que son enviadas hacia el servidor en formato JSON. El Frontend se encarga de capturar los datos introducidos por el usuario, validar la información en el cliente y presentar las respuestas visualmente una vez procesadas.

Posteriormente, las solicitudes llegan a la capa de Backend, implementada con el framework NestJS, la cual está estructurada en rutas, controladores, servicios y modelos.

Las rutas definen los puntos de acceso o endpoints y son las encargadas de recibir las peticiones provenientes del frontend. Estas rutas redirigen las solicitudes hacia los controladores, los cuales interpretan los datos recibidos y determinan qué proceso debe ejecutarse. Los controladores gestionan la lógica de comunicación entre la interfaz y la capa de negocio implementada a través de los servicios.

Los servicios constituyen el núcleo de la lógica de negocio. En ellos se implementan las validaciones, reglas operativas y transformaciones de datos necesarias antes de interactuar con la base de datos. Una vez que el servicio requiere acceder o modificar información persistente, se comunica con el modelo de datos, definido mediante Mongoose, una librería que facilita la integración con la base de datos MongoDB.

El modelo traduce las operaciones de negocio en instrucciones de manipulación de datos (como creación, lectura, actualización o eliminación). Dichas instrucciones son ejecutadas sobre MongoDB, que almacena y recupera los registros solicitados, devolviendo los resultados hacia la capa superior.

Finalmente, la respuesta obtenida desde la base de datos se propaga en sentido inverso: el modelo retorna los datos procesados al servicio, el cual los adapta al formato requerido por el controlador. Este último construye la respuesta HTTP final y la envía nuevamente al frontend, donde los resultados son presentados al usuario mediante la interfaz gráfica.

Este flujo de interacción permite mantener una arquitectura modular, escalable y de fácil mantenimiento, en la que cada capa cumple una función específica y se comunica únicamente con las capas adyacentes, garantizando la separación de responsabilidades y la integridad de los datos en todo el proceso.

5.8 Prototipos

Para dar inicio a la creación de los prototipos del aplicativo web, se elaboraron los mockups de las pantallas principales. Estos mockups representan la visión del resultado esperado al final de este proyecto y sirven como guía para el desarrollo de la interfaz de usuario.

A continuación, se muestran los mockups de las pantallas más importantes.

Figura 13

Mockup Pantalla inicial.

The image shows a mockup of a login screen for a web application. The background is a light teal color. In the center, there is a white rounded rectangle containing the following elements:

- A large heading: "Bienvenido a Makteki" in a black serif font.
- A form field for "Correo electronico" with the text "test@test.com" entered.
- A form field for "Contraseña" with 12 black dots representing a password.
- A teal button with the text "INICIAR SESION" in white capital letters.
- A link below the button: "¿Aun no tienes una cuenta?" followed by "Crear cuenta" in teal text.

Figura 14

Mockup Pantalla registro.

The image shows a registration form titled "Regístrate" centered on a white background with rounded corners, set against a light green gradient background. The form contains the following elements:

- Nombre:** A text input field containing "David".
- Apellidos:** A text input field containing "Ortiz Pérez".
- Genero:** A text input field containing "Masculino".
- Contraseña:** A text input field containing "24/07/1999".
- Numero Telefonico:** A text input field containing "3123456789".
- Direccion:** A text input field containing "carrera #calle-1 apto 1".
- Direccion:** A second text input field, also containing "carrera #calle-1 apto 1".
- Contraseña:** A text input field for a password, represented by 12 black dots.
- Buttons:** Two teal buttons at the bottom: "CREAR CUENTA" on the left and "YA TENGO CUENTA" on the right.

Figura 15

Mockup Pantalla inicial cliente.

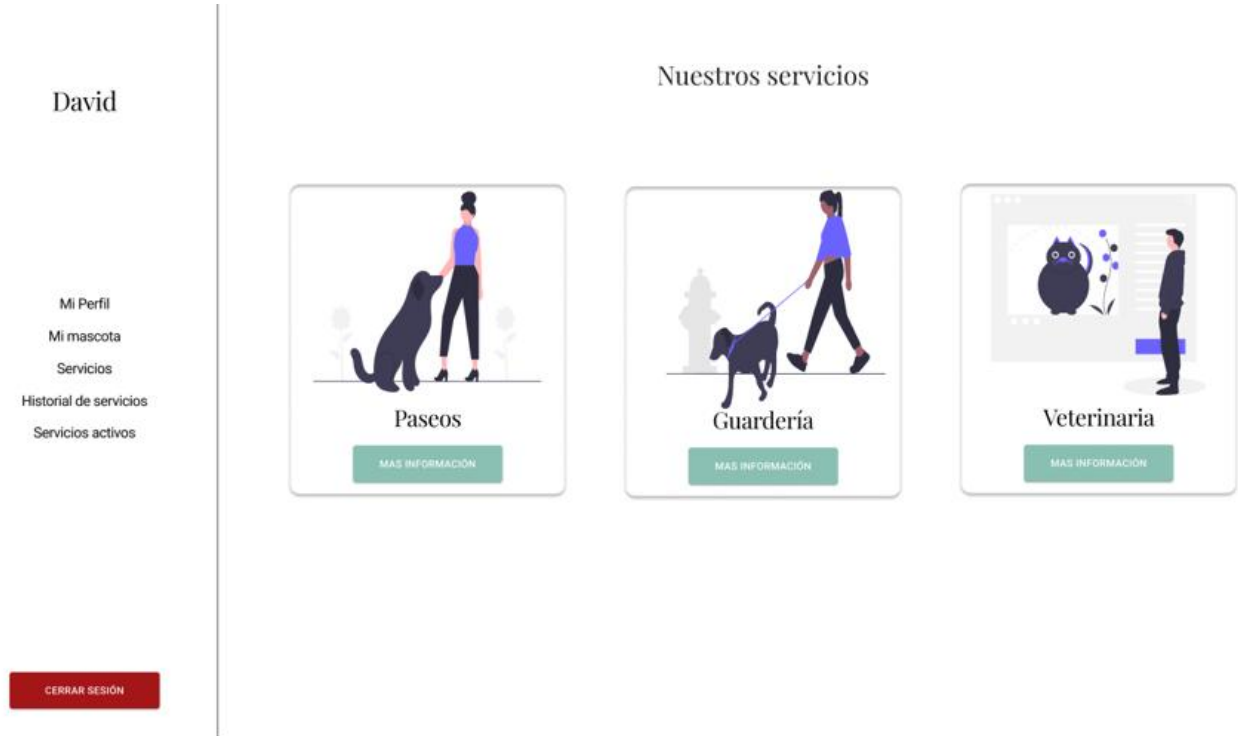


Figura 16

Mockup editar perfil.

David

- Mi Perfil
- Mi mascota
- Servicios
- Historial de servicios
- Servicios activos

CERRAR SESIÓN

Mi perfil

Nombre

Apellidos

Genero

Fecha de nacimiento

Numero Telefonico

Direccion

Correo

Contraseña

EDITAR PERFIL

GUARDAR

Figura 17

Mockup Mi mascota.

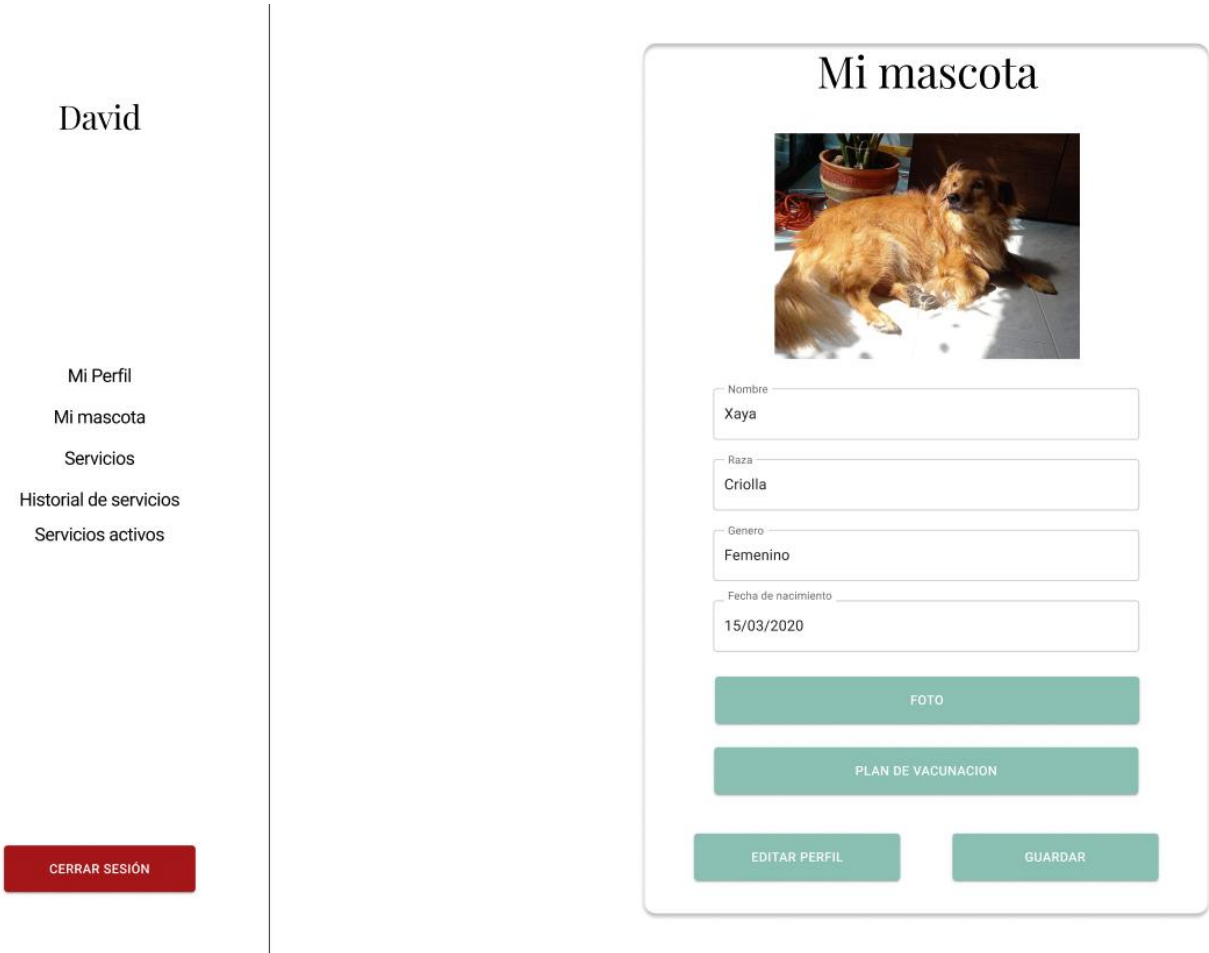


Figura 18

Mockup Plan de vacunación.

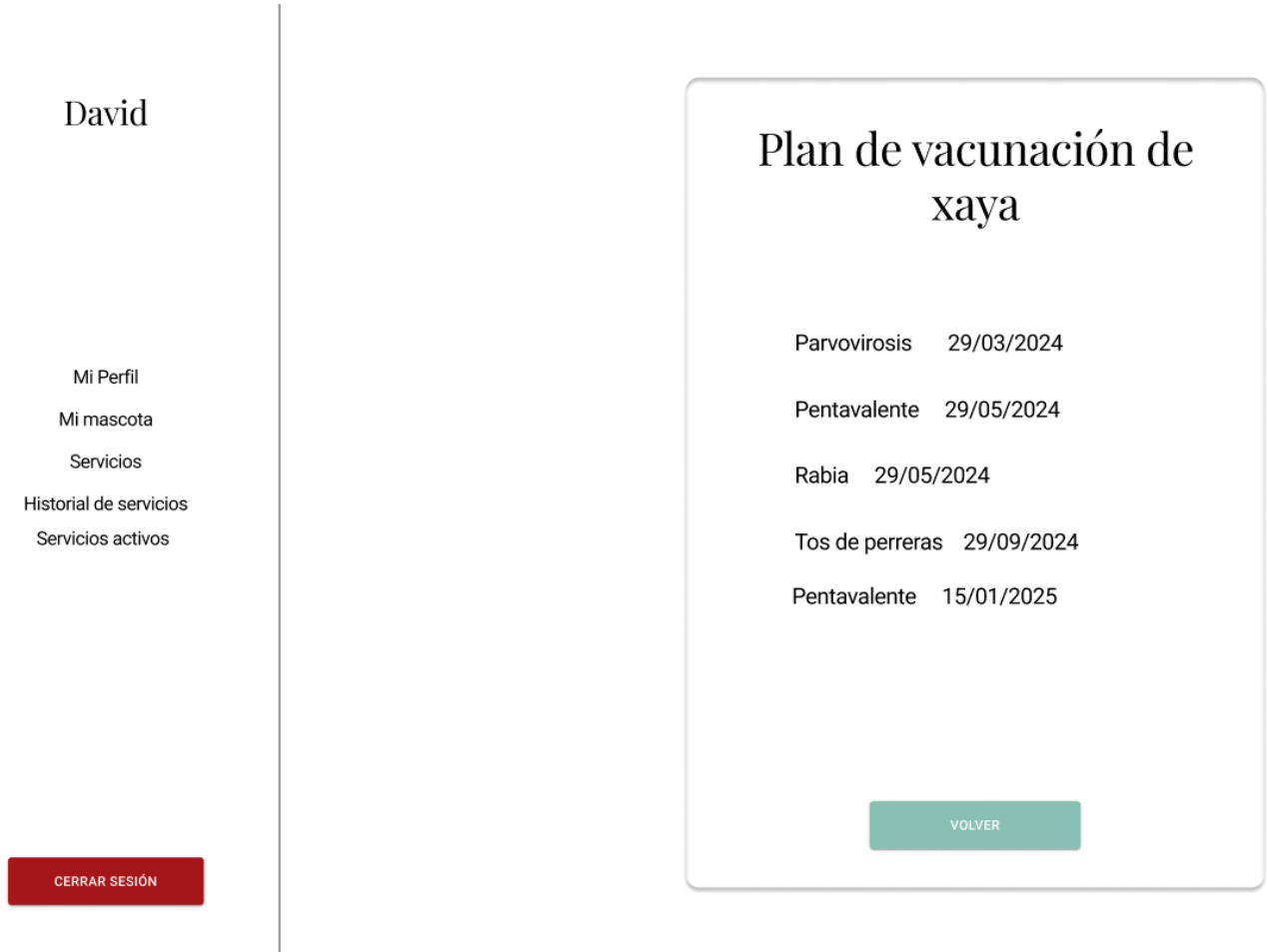
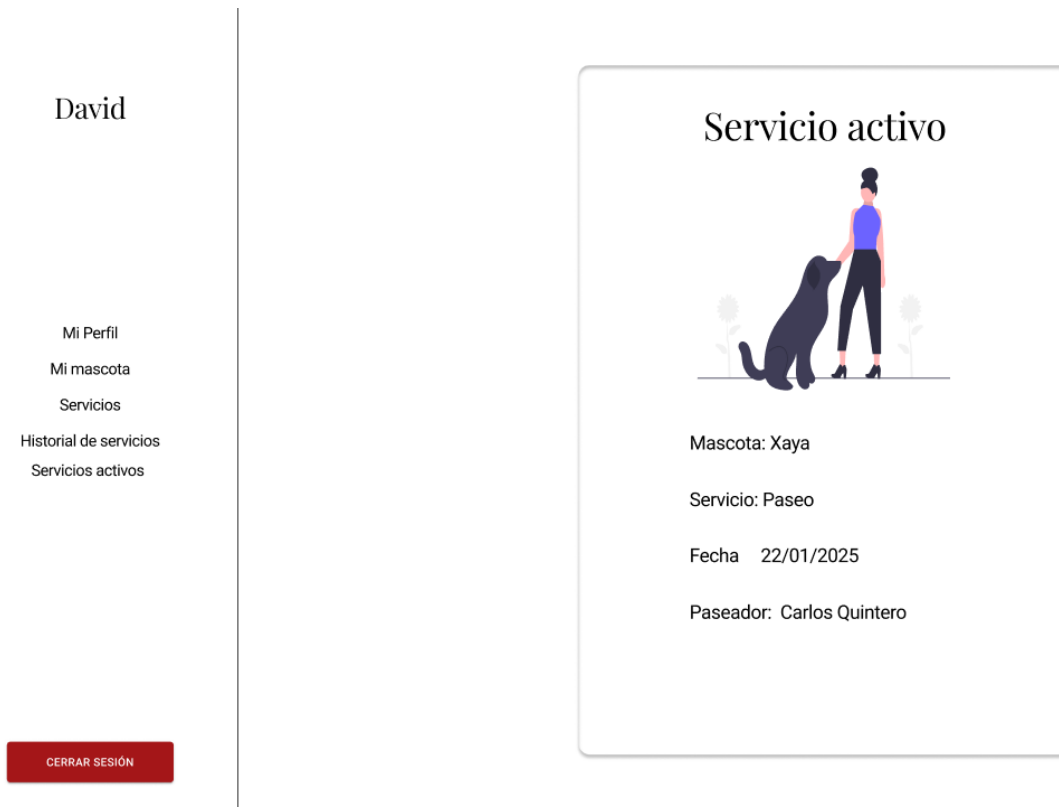


Figura 20

Mockup Servicios activos.



5.8.1 Prototipo uno

Esta fase se centró en la implementación de las funcionalidades iniciales y en la definición de los requerimientos del sistema. Se desarrollaron los módulos de registro y autenticación de usuarios, así como la validación de los campos en los formularios correspondientes.

Adicionalmente, se construyó una vista inicial de los servicios disponibles, con su respectiva validación para permitir su solicitud. Se incluyeron, además, el perfil del usuario con opción de edición, y añadir un perfil de mascota, el cual incorpora un plan de vacunación con la

posibilidad de agregar nuevas vacunas. Finalmente, se habilitó la visualización del historial de servicios y una vista específica para los casos en que exista un servicio activo.

Con el fin de llevar un control del avance y de la implementación de los prototipos, se empleó un sistema de control de versiones (Git) junto con la plataforma GitHub, lo que permitió gestionar los cambios realizados durante el desarrollo. Al finalizar este primer prototipo, se realizaron pruebas internas por parte del desarrollador. Posteriormente, se socializó con un grupo de usuarios, cuyas opiniones, observaciones y sugerencias fueron consideradas para el desarrollo del siguiente prototipo. A continuación, se presentan las primeras interfaces diseñadas en esta etapa.

Figura 21

Pantalla inicial, prototipo uno.



Bienvenido a makteki

Correo
mpplas@hotmail.com

Contraseña
••••••••

LOGIN REGISTRO

The image shows a login and registration form titled "Bienvenido a makteki". It features two input fields: "Correo" (Email) with the value "mpplas@hotmail.com" and "Contraseña" (Password) with masked characters. Below the fields are two blue buttons labeled "LOGIN" and "REGISTRO". The form is centered on a light green background.

Figura 22

Registro de usuarios.

The image shows a user registration form titled "Crear tu usuario" centered on a light green background. The form is a white rounded rectangle containing several input fields and two buttons at the bottom. The fields are labeled as follows: "Nombre" (containing "miguel"), "Apellidos" (containing "ortiz"), "Fecha de nacimiento" (containing "05/02/2000" and a calendar icon), "Numero Telefonico" (containing "3323232232"), "Dirección" (containing "carrea 1 #1-11"), "Correo" (containing "mpplas@hotmail.com"), and "Contraseña" (containing "....."). At the bottom, there are two blue buttons: "CREAR CUENTA" on the left and "YA TENGO CUENTA" on the right.

Figura 23

Registro de colaboradores.



Forma parte de nuestro equipo

Nombre

Apellidos

Fecha de nacimiento 

Género 

Numero Telefonico

Dirección

Correo

Contraseña

[CREAR CUENTA](#) [YA TENGO CUENTA](#)

Figura 24

Página inicial cliente.



Figura 25

Vista ver y editar perfil.

The image shows a web application interface for a pet care service. On the left is a vertical sidebar menu with the following items: 'miguel', 'Mi perfil', 'Mi mascota', 'Servicios', 'Servicios activos', and 'Historial de servicios'. At the bottom of the sidebar is a red button labeled 'SALIR' with a right-pointing arrow icon. The main content area features a white box titled 'Mi Perfil'. Inside this box are several input fields: 'Nombre' (containing 'miguel'), 'Apellidos' (containing 'ortiz'), 'Número Telefónico' (containing '3323232232'), 'Dirección' (containing 'carrea 1 #1-11'), 'Correo' (containing 'mpplas@hotmail.com'), and 'Contraseña' (containing a series of dots). Below the fields are two blue buttons: 'EDITAR PERFIL' and 'GUARDAR'.

Figura 26

Vista crear perfil mascota.

miguel

Mi perfil

Mi mascota

Servicios

Servicios activos

Historial de servicios

Mi Mascota

Nombre

Raza

Genero

Fecha de nacimiento

Peso Kg

CREAR PERFIL

PLAN DE VACUNACIÓN

SALIR

Figura 27

Vista plan de vacunación mascota.

miguel

Mi perfil

Mi mascota

Servicios

Servicios activos

Historial de servicios

Plan de vacunación de Xaya

<input type="checkbox"/>	ID	Nombre	fecha
<input type="checkbox"/>	1	Parvovirus	10/10/2021
<input type="checkbox"/>	2	Pentavalente	10/10/2021
<input type="checkbox"/>	3	Rabia	10/10/2021
<input type="checkbox"/>	4	Tos de perreras	10/10/2021
<input type="checkbox"/>	5	Pentavalente	10/10/2021
<input type="checkbox"/>	6	Parvovirus	10/10/2021

1-6 of 6 < >

VOLVER

SALIR

Figura 28

Vista historial de servicios.

miguel

Mi perfil

Mi mascota

Servicios

Servicios activos

Historial de servicios

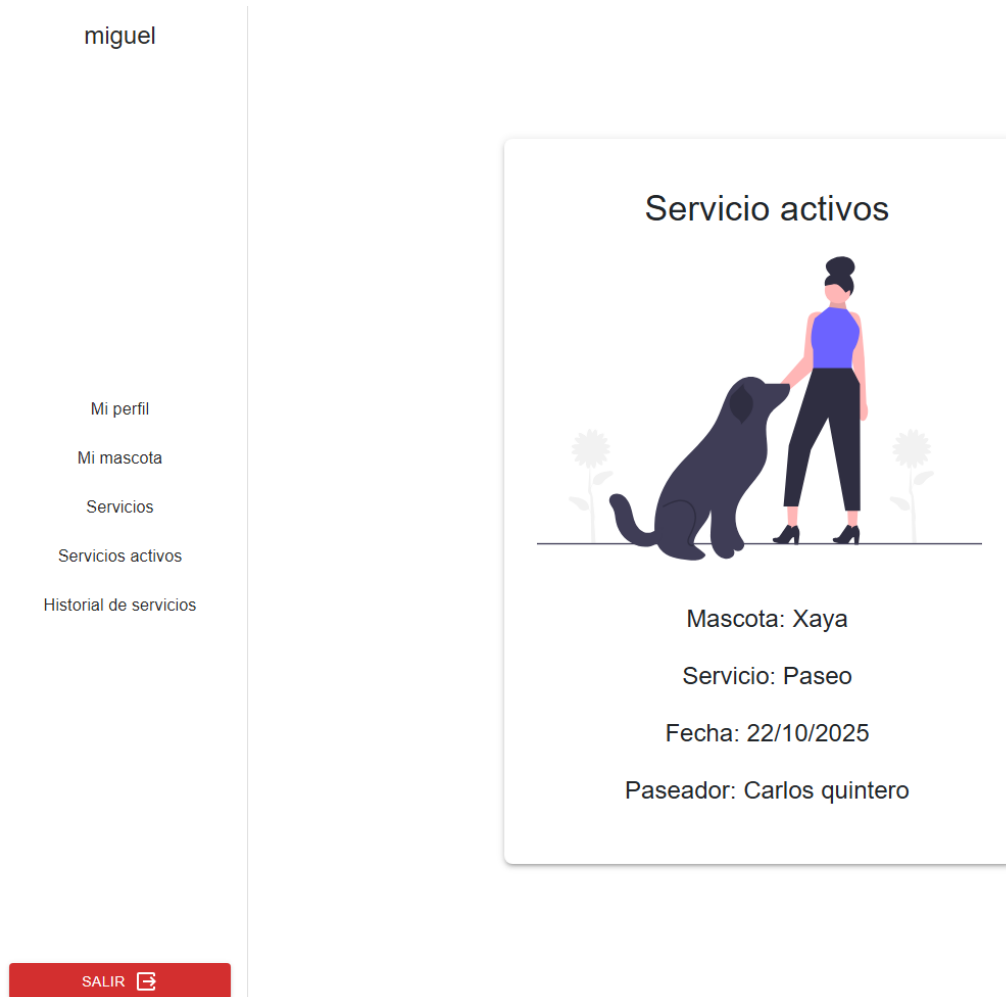
Historial de servicios

Nombre	Categoria	Descripcion	Fecha	Colaborador
Xaya	Paseo	Paseo de 1 hora	22/01/2025	Carlos
Xaya	Paseo	Paseo de 1 hora	22/01/2025	Juan
Xaya	Paseo	Paseo de 1 hora	22/01/2025	Jose
1-3 of 3 < >				

SALIR

Figura 29

Vista de servicio activo.



5.8.2 Prototipo dos.

En el desarrollo del segundo prototipo se incorporaron diversas mejoras relacionadas al diseño de la interfaz de usuario, la usabilidad y la integración de nuevas funcionalidades las cuales son:

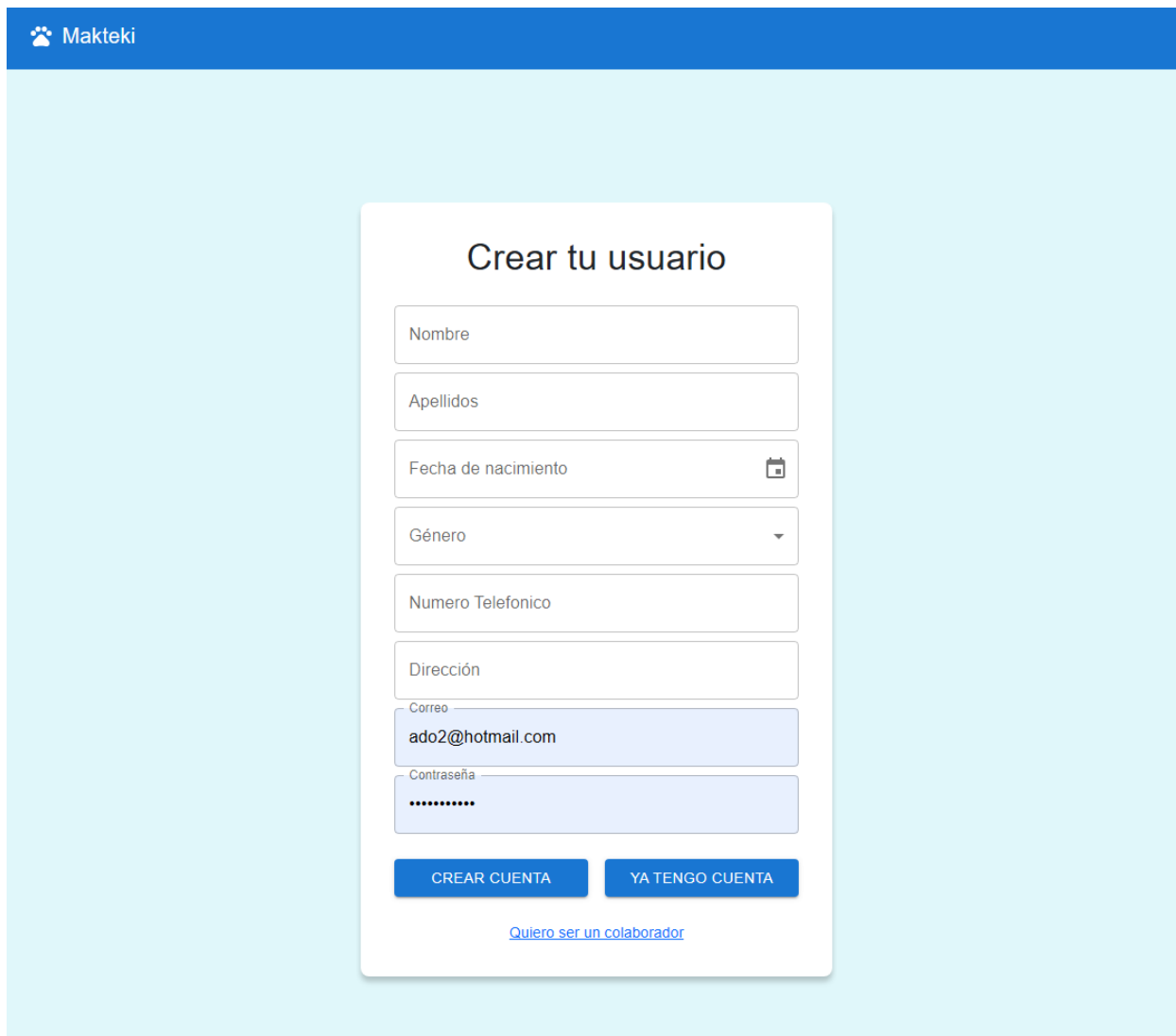
1. Vista de inicio (ver figura 30).
2. Registro usuarios dividido por tipo: colaborador o propietario (ver figura 31 y 32).
3. Definición de los servicios y su respectiva descripción (ver figura 39).
4. Posibilidad de crear más de un perfil de mascota (ver figura 42).
5. Editar perfil de mascota (ver figura 44).
6. Opción de cancelar un servicio (ver figura 49).
7. Visualización de información de contacto del colaborador (ver figura 51).
8. Calificar un servicio una vez finalizado (ver figura 52).
9. Historial de servicios ampliado con información adicional (ver figura 53).
10. Incorporación de la acción “Tomar” un servicio (ver figura 55).
11. Vista de servicios activos tomados por colaboradores (ver figura 56).
12. Vista del historial de servicios y sus respectivas calificaciones (ver figura 58).

Figura 30*Vista de inicio.*

Nota. Esta es la vista principal o página de inicio del prototipo web. En la barra de navegación (navbar) se encuentra el elemento de iniciar sesión. El cuerpo de la página presenta una breve descripción del prototipo, seguida de los elementos para registrarse como dueño de mascota (cliente) o como colaborador (cuidador o veterinario). Es importante señalar que, si el usuario no ha iniciado sesión, no podrá acceder a otras secciones más allá de esta vista.

Figura 31

Vista del formulario de registro para clientes (dueños de mascotas).



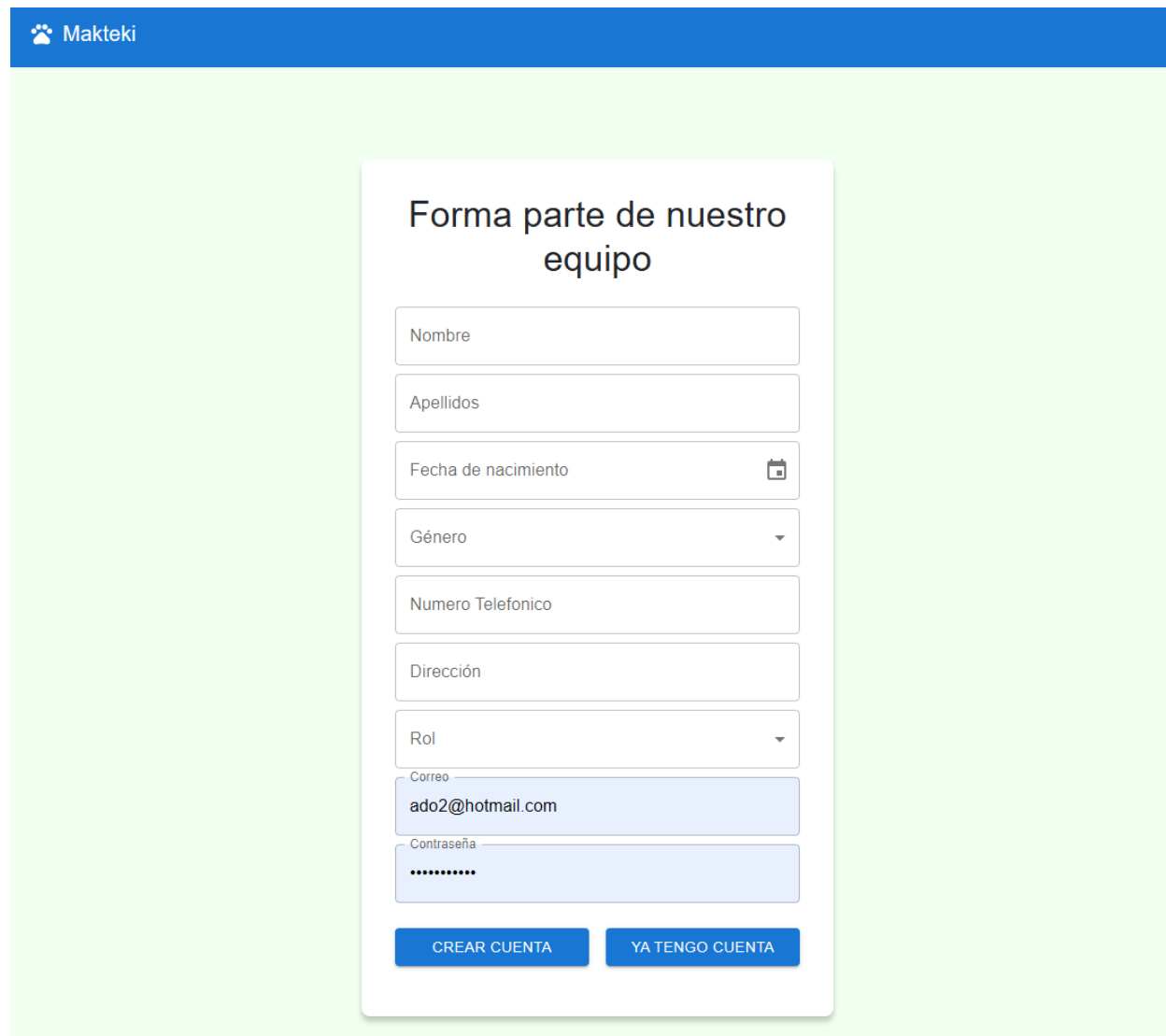
The image shows a registration form titled "Crear tu usuario" on a light blue background. At the top left, there is a blue header with a paw print icon and the text "Makteki". The form itself is a white card with rounded corners. It contains the following fields and elements:

- Nombre**: A text input field.
- Apellidos**: A text input field.
- Fecha de nacimiento**: A date picker field with a calendar icon.
- Género**: A dropdown menu.
- Numero Telefonico**: A text input field.
- Dirección**: A text input field.
- Correo**: A text input field containing "ado2@hotmail.com".
- Contraseña**: A text input field with masked characters (dots).
- CREAR CUENTA**: A blue button.
- YA TENGO CUENTA**: A blue button.
- [Quiero ser un colaborador](#): A blue link at the bottom.

Nota. Para completar el proceso de registro, el usuario debe diligenciar el formulario mostrado en la Figura 31. Este formulario solicita, como mínimo, el ingreso de un nombre y un apellido, fecha de nacimiento, género, número telefónico, dirección de residencia, correo electrónico (puede ser Gmail, Hotmail, Outlook, entre otros) y una contraseña válida.

Figura 32

Vista del formulario de registro para colaboradores (cuidadores o veterinarios).



The image shows a registration form titled "Forma parte de nuestro equipo" (Join our team) on the Makteki website. The form is centered on a light green background. It contains the following fields and elements:

- Nombre** (Name): Text input field.
- Apellidos** (Last Name): Text input field.
- Fecha de nacimiento** (Date of Birth): Date picker field with a calendar icon.
- Género** (Gender): Dropdown menu.
- Numero Telefonico** (Phone Number): Text input field.
- Dirección** (Address): Text input field.
- Rol** (Role): Dropdown menu.
- Correo** (Email): Text input field containing "ado2@hotmail.com".
- Contraseña** (Password): Password input field with masked characters.
- CREAR CUENTA** (Create Account): Blue button.
- YA TENGO CUENTA** (I already have an account): Blue button.

Nota. Para completar el proceso de registro, el usuario debe diligenciar el formulario mostrado en la Figura 32. Este formulario solicita, como mínimo, el ingreso de un nombre y un apellido, fecha de nacimiento, género, número telefónico, dirección de residencia, la selección del rol (cuidador o veterinario), correo electrónico (puede ser Gmail, Hotmail, Outlook, entre otros) y una contraseña válida.

Figura 33

Definición del esquema de Usuario en el backend.

```

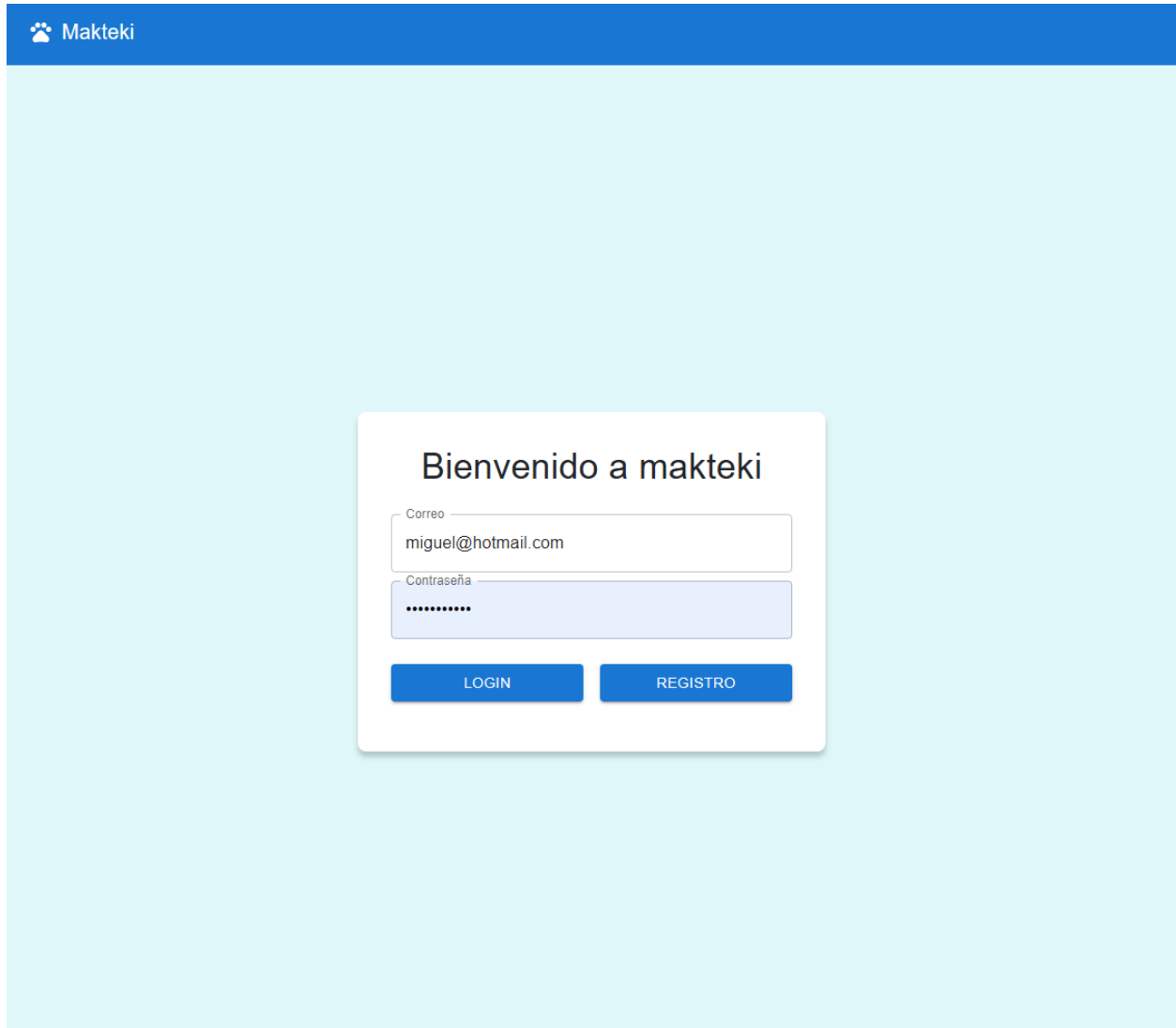
src > auth > entities > TS user.entity.ts > ...
1  import { Prop, Schema, SchemaFactory } from "@nestjs/mongoose";
2  import { Document, SchemaTypes, Types } from "mongoose";
3  import { Roles } from "../dto/create-user.dto";
4
5
6  export type UserDocument = User & Document<Types.ObjectId>;
7
8  @Schema({ timestamps: true })
9  export class User {
10     @Prop({ required: true })
11     name: string;
12     @Prop({ required: true })
13     lastName: string;
14     @Prop({ required: true, unique: true })
15     email: string;
16     @Prop({ required: true })
17     password: string;
18     @Prop({ required: true, unique: true })
19     phone: string;
20     @Prop({ required: true })
21     address: string;
22     @Prop({ required: true })
23     gender: string;
24     @Prop({ required: true, enum: Roles })
25     role: Roles;
26     @Prop({ required: true, type: SchemaTypes.Date })
27     bornDate: Date;
28 }
29
30 export const UserSchema = SchemaFactory.createForClass(User);
31

```

Nota. En este fragmento de código se define la estructura del modelo Usuario, el cual representa a los usuarios registrados dentro del prototipo. Los esquemas definen la estructura que tendrán los documentos dentro de la base de datos definido en un archivo entity.ts. Este esquema de usuario incluye campos esenciales como nombre, apellido, correo electrónico, contraseña, teléfono, dirección, género, rol y fecha de nacimiento. Esta configuración permite gestionar la persistencia de los datos en la base de datos y garantizar la integridad de la información de los usuarios en el prototipo web.

Figura 34

Vista iniciar sesión.



The image shows a web application interface for 'Makteki'. At the top, there is a blue header bar with a white paw print icon and the text 'Makteki'. The main content area has a light blue gradient background. In the center, there is a white rounded rectangle containing the text 'Bienvenido a makteki'. Below this, there are two input fields: 'Correo' with the value 'miguel@hotmail.com' and 'Contraseña' with a masked password of ten dots. At the bottom of the white box, there are two blue buttons: 'LOGIN' and 'REGISTRO'.

Nota. Para iniciar sesión, el usuario debe ingresar su correo electrónico y su contraseña.

Figura 35

Fragmento de código proceso de autenticación.

```

src > auth > TS auth.service.ts > AuthService > findOne
18  export class AuthService {
26  async createUser(userDto: UserDto) {
53      code: "user_creation_failed",
54      message: error.message || "No se pudo crear el usuario",
55  });
56  }
57  }
58
59  async login(
60      loginDto: LoginDto
61  ): Promise<{ access_token: string; role: string }> {
62      // buscar usuario por email
63      const user = await this.userModel.findOne({ email: loginDto.email });
64      if (!user) {
65          throw new UnauthorizedException({
66              code: "user_not_found",
67              message: "El usuario no existe",
68          });
69      }
70
71      // Comparar contraseñas
72      const isValidPassword = await bcrypt.compare(
73          loginDto.password,
74          user.password
75      );
76      if (!isValidPassword) {
77          throw new UnauthorizedException({
78              code: "invalid_credentials",
79              message: "Contraseña incorrecta",
80          });
81      }
82
83      // generar token JWT
84      return {
85          access_token: this.jwtService.sign({
86              username: user.email,
87              sub: user._id.toString(),
88          }),
89          role: user.role,
90      };
91  }
92

```

Nota. En este fragmento se muestra la función login, encargada de autenticar a los usuarios en el sistema. El proceso inicia con la verificación del correo electrónico y la contraseña proporcionados por el usuario, en caso de credenciales incorrectas, se generan excepciones controladas. Si la autenticación es exitosa, el sistema crea un token JWT (JSON Web Token) que contiene la información del usuario y su rol, permitiendo el acceso a las funcionalidades autorizadas dentro del prototipo web.

Figura 36

Archivo de validación de usuarios al iniciar sesión.

```

src > router > AppRouter.tsx > AppRouter
1  import { Route, Routes } from "react-router-dom";
2  import { MaktekiLayout } from "../userui/layout/MaktekiLayout";
3  import { CollabLayout } from "../collaboratorui/layout";
4  import { Profile } from "../userui/pages/Profile";
5  import { LandingPage, LoginPage, RegisterCollabPage, RegisterPage } from "../auth/pages";
6  import { MyPet, ServicesHistory, Vaccination } from "../userui/pages";
7  import { CardsClient } from "../userui/components/CardsClient";
8  import { ProtectedRoute } from "../ProtectedRoute";
9  import { CollabActiveServices, CollabProfile, CollabServices, CollabServicesHistory } from "../collaboratorui/pages";
10
11 export const AppRouter = () => {
12   return (
13     <Routes>
14       { /* ruta publica para el login y registro */ }
15       <Route path="/" element={ <LandingPage /> } />
16       <Route path="/login" element={ <LoginPage /> } />
17       <Route path="/register" element={ <RegisterPage /> } />
18       <Route path="/RegisterCollabPage" element={ <RegisterCollabPage /> } />
19
20       { /* rutas de usuario (clientes) */ }
21       <Route
22         path="/cliente/"
23         element={
24           <ProtectedRoute allowedRoles={["CLIENTE"]} >
25             <MaktekiLayout />
26           </ProtectedRoute>
27         }
28       >
29         <Route path="miperfil" element={ <Profile /> } />
30         <Route path="mimascota" element={ <MyPet /> } />
31         <Route path="servicios" element={ <CardsClient /> } />
32         <Route path="historialdeservicios" element={ <ServicesHistory /> } />
33         <Route path="vacunas" element={ <Vaccination /> } />
34       </Route>
35
36       { /* rutas de colaboradores */ }
37       <Route
38         path="/colaborador/"
39         element={
40           <ProtectedRoute allowedRoles={["CUIDADOR", "VETERINARIO"]} >
41             <CollabLayout />
42           </ProtectedRoute>
43         }
44       >
45         <Route path="miperfil" element={ <CollabProfile /> } />
46         <Route path="serviciodisponibles" element={ <CollabServices /> } />
47         <Route path="serviciosactivos" element={ <CollabActiveServices /> } />
48         <Route path="historialdeservicios" element={ <CollabServicesHistory /> } />
49       </Route>
50     </Routes>
51   );

```

Nota. Una vez que el usuario ha iniciado sesión, se habilitan distintas funcionalidades dentro del prototipo web. Dependiendo del tipo de usuario (cliente, cuidador o veterinario) se presentan vistas y opciones específicas adaptadas a su rol.

Figura 37

Archivo de validación de token al inicio con sesión.

```

src > auth > TS axiosInstance.ts > ...
1  import axios from "axios";
2
3  // Interceptor para adjuntar el token
4  axios.interceptors.request.use(
5    (config) => {
6      config.baseUrl = import.meta.env.VITE_API_URL;
7      const token = localStorage.getItem("token");
8      if (token) {
9        config.headers["Authorization"] = `Bearer ${token}`;
10     }
11     return config;
12   },
13   (error) => {
14     return Promise.reject(error);
15   }
16 );
17 axios.interceptors.response.use(
18   (response) => {
19     return response;
20   },
21   (err) => {
22     console.log("error axios:", err);
23     const error = err;
24     return Promise.reject(error);
25   }
26 );
27
    
```

Nota. Una vez que el usuario ha iniciado sesión, se genera y almacena un token de autenticación, el cual permite validar y autorizar todas las peticiones realizadas al backend. Este token garantiza la seguridad en las comunicaciones entre el cliente y el servidor, asegurando que únicamente los usuarios autenticados puedan acceder a las funcionalidades permitidas según su rol dentro del sistema.

Figura 38

Archivo configuración tiempo de sesión y token.

```

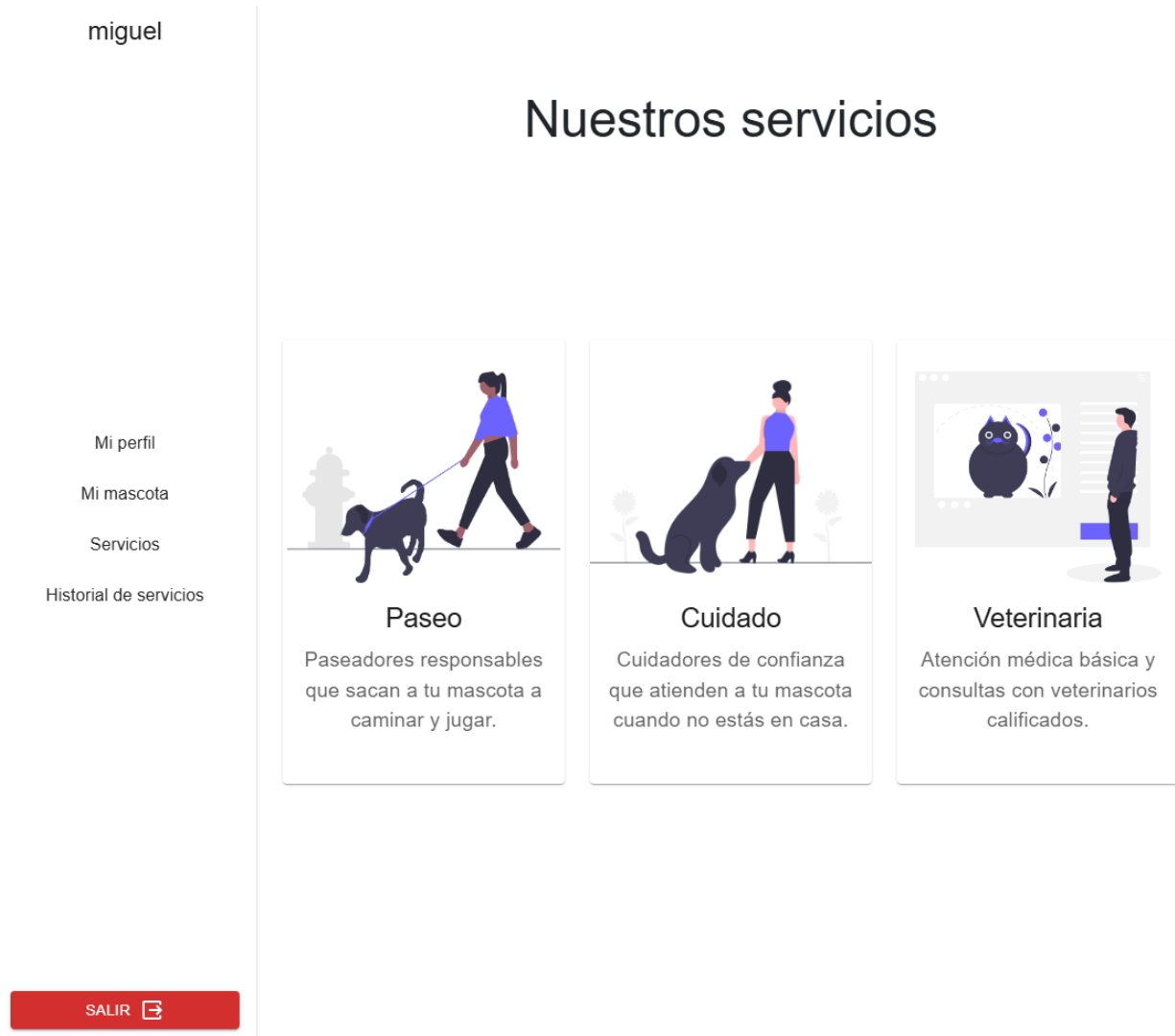
1  import { MiddlewareConsumer, Module, NestModule } from "@nestjs/common";
2  import { AuthService } from "../auth.service";
3  import { AuthController } from "../auth.controller";
4  import { HttpModule } from "@nestjs/axios";
5  import { MongooseModule } from "@nestjs/mongoose";
6  import { User, UserSchema } from "../entities/user.entity";
7  import { PassportModule } from "@nestjs/passport";
8  import { JwtModule } from "@nestjs/jwt";
9  import { ConfigModule, ConfigService } from "@nestjs/config";
10
11 @Module({
12   imports: [
13     MongooseModule.forFeature([
14       { name: User.name, schema: UserSchema }
15     ]),
16     HttpModule,
17     PassportModule,
18     JwtModule.registerAsync({
19       useFactory: async (config: ConfigService) => {
20         const secret = config.get<string>("JWT_SECRET");
21         return {
22           secret,
23           signOptions: { expiresIn: "12h" },
24         };
25       },
26       inject: [ConfigService],
27       imports: [ConfigModule],
28     }),
29   ],
30   controllers: [AuthController],
31   providers: [AuthService],
32   exports: [AuthService, JwtModule],
33 })
34 export class AuthModule implements NestModule {
35   configure(consumer: MiddlewareConsumer) {
36     consumer;
37   }
38 }

```

Nota. Una vez que el usuario ha iniciado sesión, se genera y almacena un token de autenticación JWT (JSON Web Token), el backend guarda el token generado y se asigna una sesión de doce horas lo que proporciona al usuario autenticado un intervalo de tiempo suficiente para el uso continuo del prototipo sin requerir un nuevo inicio de sesión. Transcurrido este periodo, el token expira automáticamente, lo que refuerza las medidas de seguridad y evita el acceso prolongado no autorizado.

Figura 39

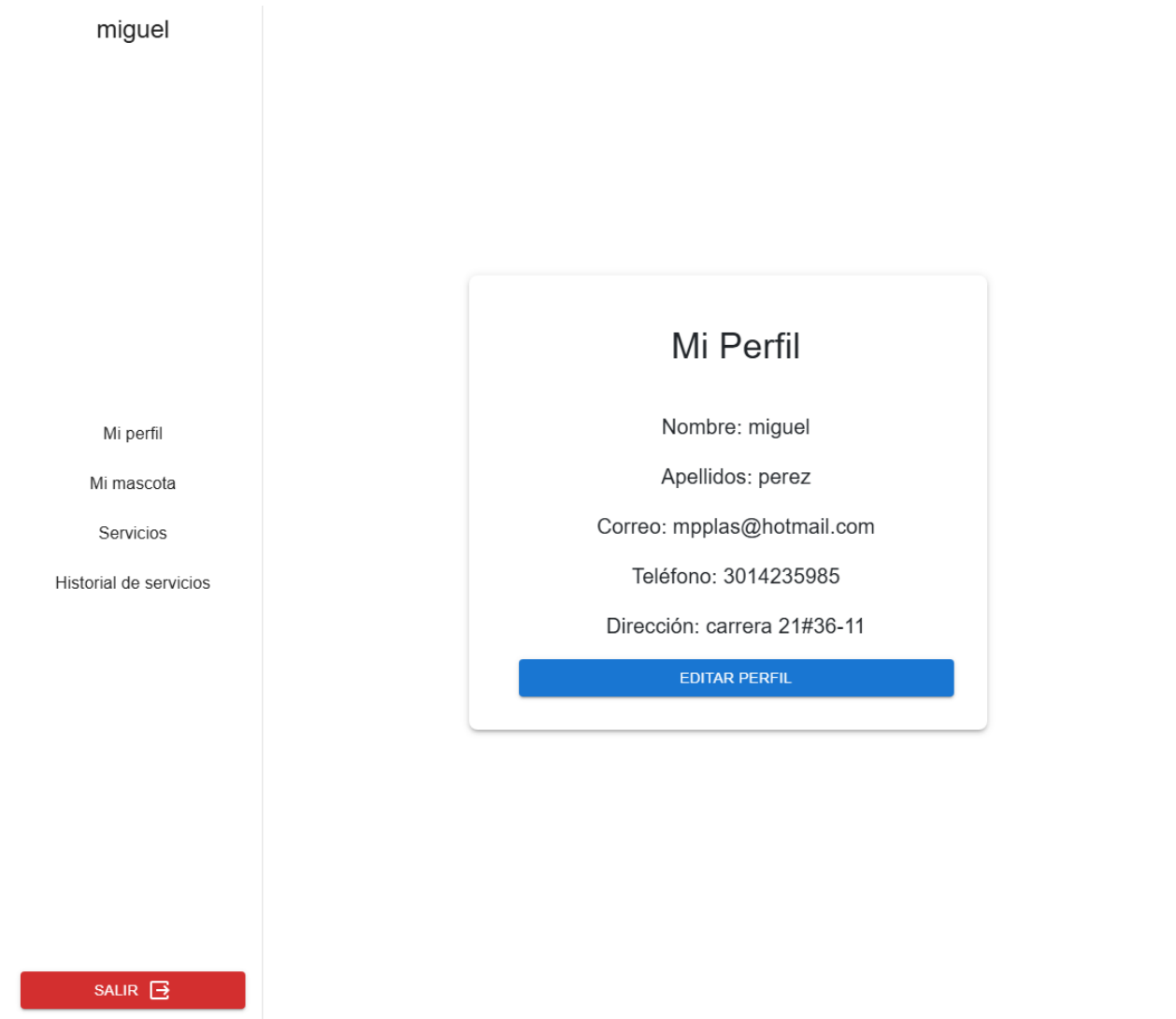
Vista de inicio con sesión iniciada (usuario cliente).



Nota. Una vez que el usuario ha iniciado sesión, se habilitan nuevas funcionalidades dentro del prototipo web. Ahora dispone de un menú lateral desde el cual puede acceder a su perfil, consultar la información de sus mascotas, visualizar los servicios disponibles, gestionar sus solicitudes y cerrar sesión.

Figura 40

Vista ver perfil de usuario.



Nota. El usuario puede visualizar la información asociada a su perfil y acceder a la opción de edición para actualizar sus datos personales.

Figura 41

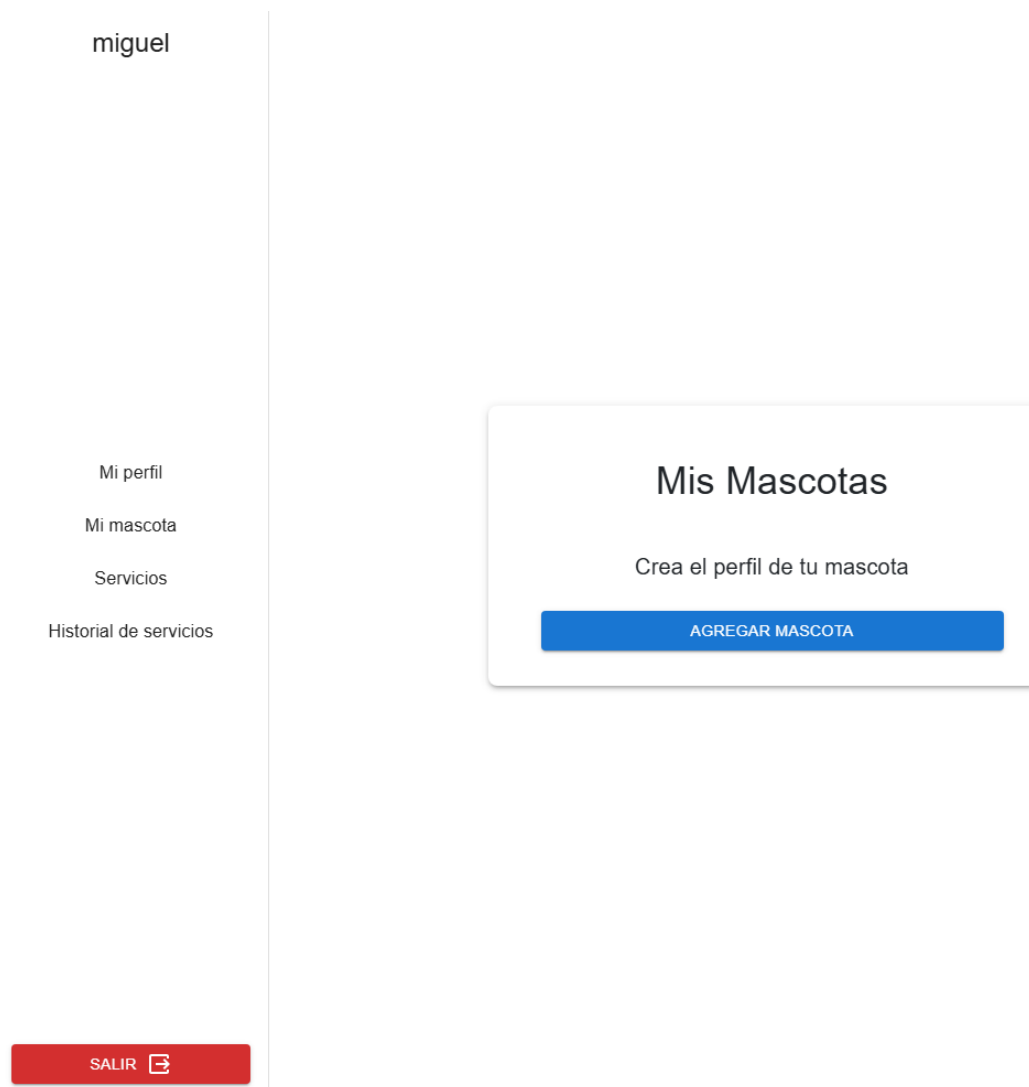
Vista de edición de perfil del usuario.

The image shows a user profile editing interface. A central white modal box titled "Editar Perfil" contains four text input fields. The first field is labeled "Nombre" and contains the text "miguel". The second field is labeled "Apellidos" and contains "perez". The third field is labeled "Teléfono" and contains "3014235985". The fourth field is labeled "Dirección" and contains "carrera 21#36-11". Below these fields is a prominent blue button with the text "GUARDAR CAMBIOS". The background is a dark grey sidebar with a menu: "Mi perfil", "Mi mascota", "Servicios", and "Historial de servicios". At the bottom left of the sidebar is a red button labeled "SALIR" with an exit icon.

Nota. El usuario puede actualizar únicamente los campos que considere necesarios, sin la obligación de modificar toda la información registrada previamente.

Figura 42

Vista de la sección “Mi mascota”.



Nota. En esta sección, el usuario puede crear y administrar los perfiles de sus mascotas. Si es la primera vez que accede, el sistema muestra únicamente la opción para registrar una mascota. Posteriormente, se habilita la posibilidad de añadir y gestionar múltiples perfiles, permitiendo mantener un control individual de la información.

Figura 43

Vista mi mascota.

The image shows a user interface for creating a pet profile. On the left, a sidebar for user 'miguel' contains navigation links: 'Mi perfil', 'Mi mascota', 'Servicios', and 'Historial de servicios'. At the bottom of the sidebar is a 'SALIR' button with an external link icon. The main content area features a 'CREAR MASCOTA' form with the following fields: 'Nombre' (text input), 'Raza' (text input), 'Genero' (dropdown menu), 'Fecha de nacimiento' (calendar icon), 'Peso Kg' (text input), and an image upload area with the instruction 'Arrastra una imagen aquí o haz clic para seleccionar'. A blue 'CREAR MASCOTA' button is positioned at the bottom of the form. A dark red 'SALIR' button is also visible at the bottom left of the page.

Nota. Para registrar una mascota, el usuario debe completar todos los campos del formulario. El sistema no permite guardar la información si alguno de los campos requeridos está vacío.

Figura 44

Vista edición del perfil de mascota.

The image shows a web application interface for editing a pet's profile. A modal form is centered on the screen, containing the following fields:

- Nombre:** Logan
- Raza:** husky siberiano
- Genero:** Macho (with a dropdown arrow)
- Fecha de nacimiento:** (with a calendar icon)
- Peso Kg:** (empty field)

Below the text fields is a dashed-line box containing a photo of a husky dog. At the bottom of the modal is a blue button labeled "ACTUALIZAR DATOS".

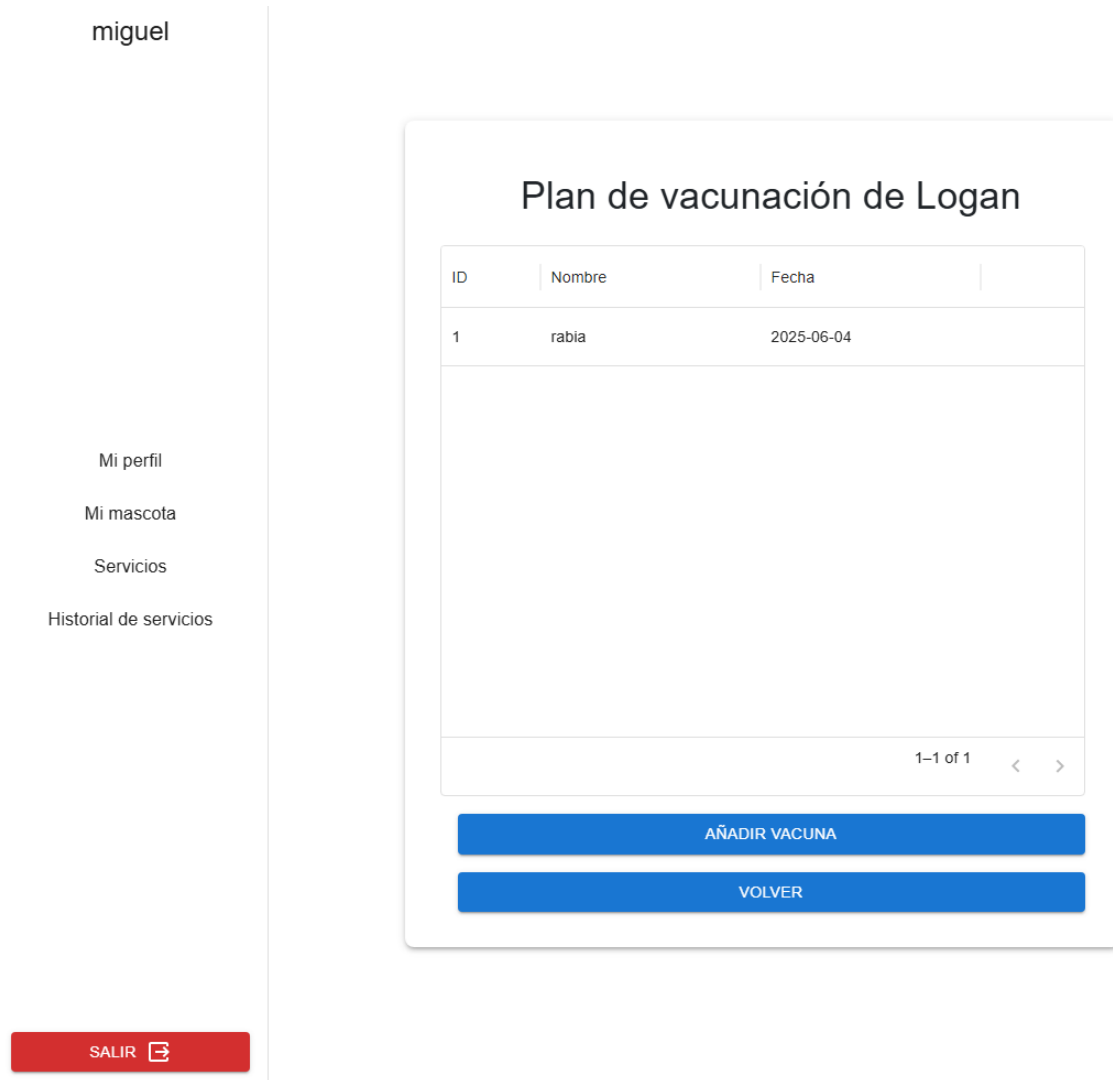
The background is a dark grey sidebar with the name "miguel" at the top. Below it are navigation links: "Mi perfil", "Mi mascota", "Servicios", and "Historial de servicios". At the bottom left of the sidebar is a red button labeled "SALIR" with a door icon. To the right of the modal, a partial view of another pet profile is visible, showing the name "as", a photo of a dog, and an "EDITAR" button.

Nota. En este formulario, el usuario puede actualizar la información registrada sobre su mascota.

El sistema únicamente modificará los campos que hayan sido editados por el usuario.

Figura 45

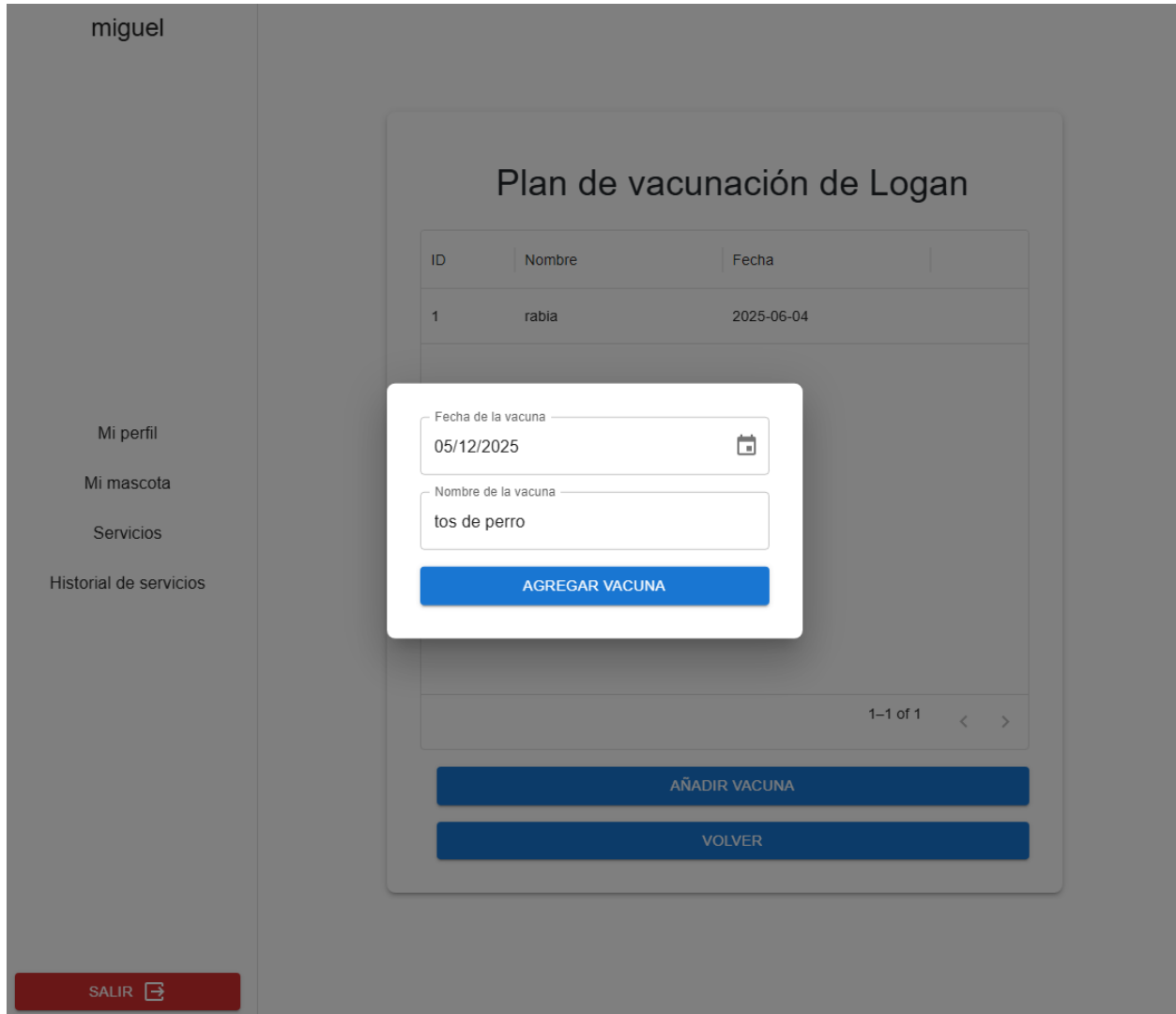
Vista plan de vacunación de mascota.



Nota. En este formulario, el usuario puede visualizar la información registrada sobre el plan de vacunación de su mascota

Figura 46

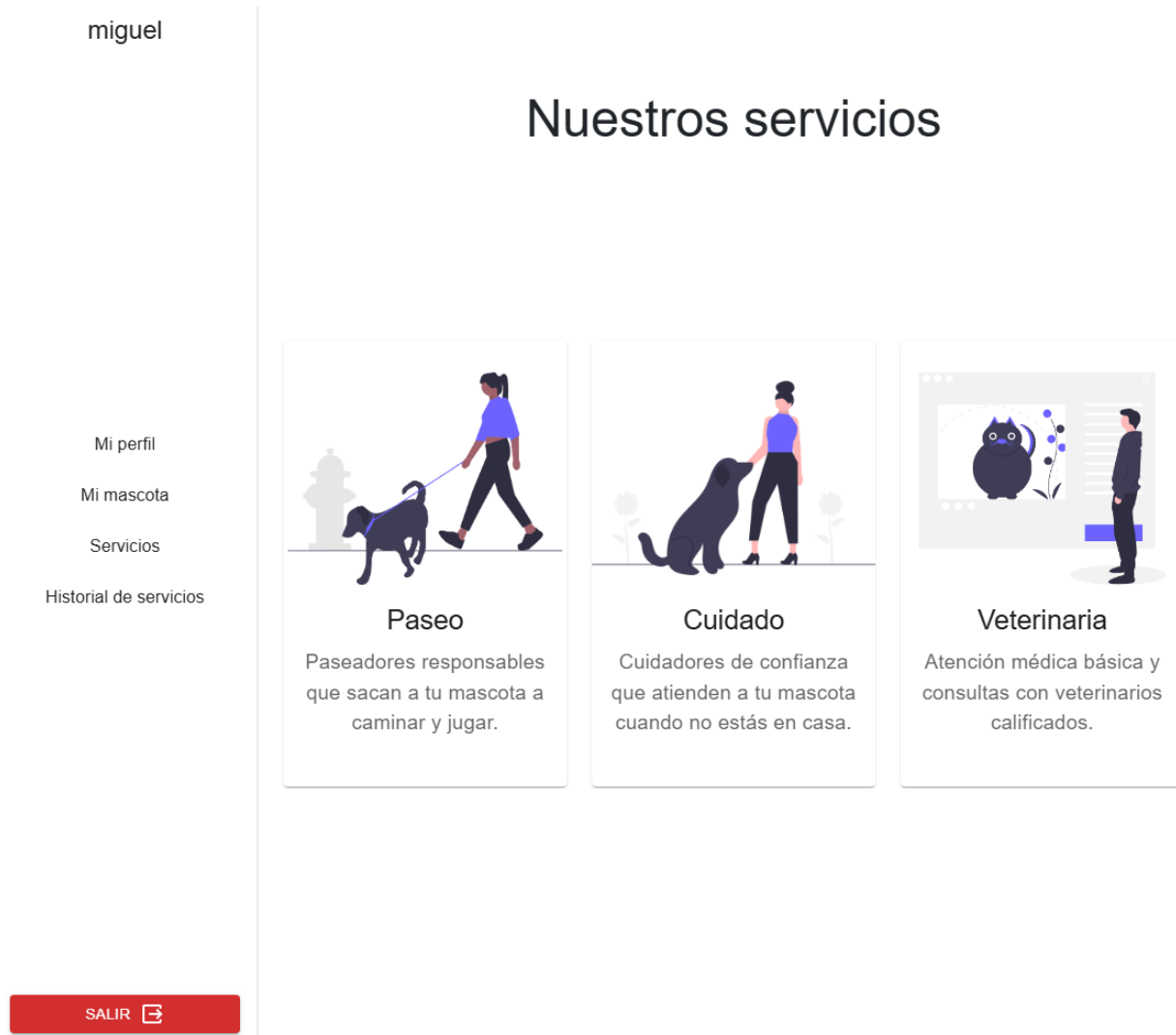
Vista agregar vacuna.



Nota. El usuario puede registrar una nueva vacuna en el plan de vacunación de la mascota, ingresando la información correspondiente como el nombre de la vacuna, la fecha de aplicación.

Figura 47

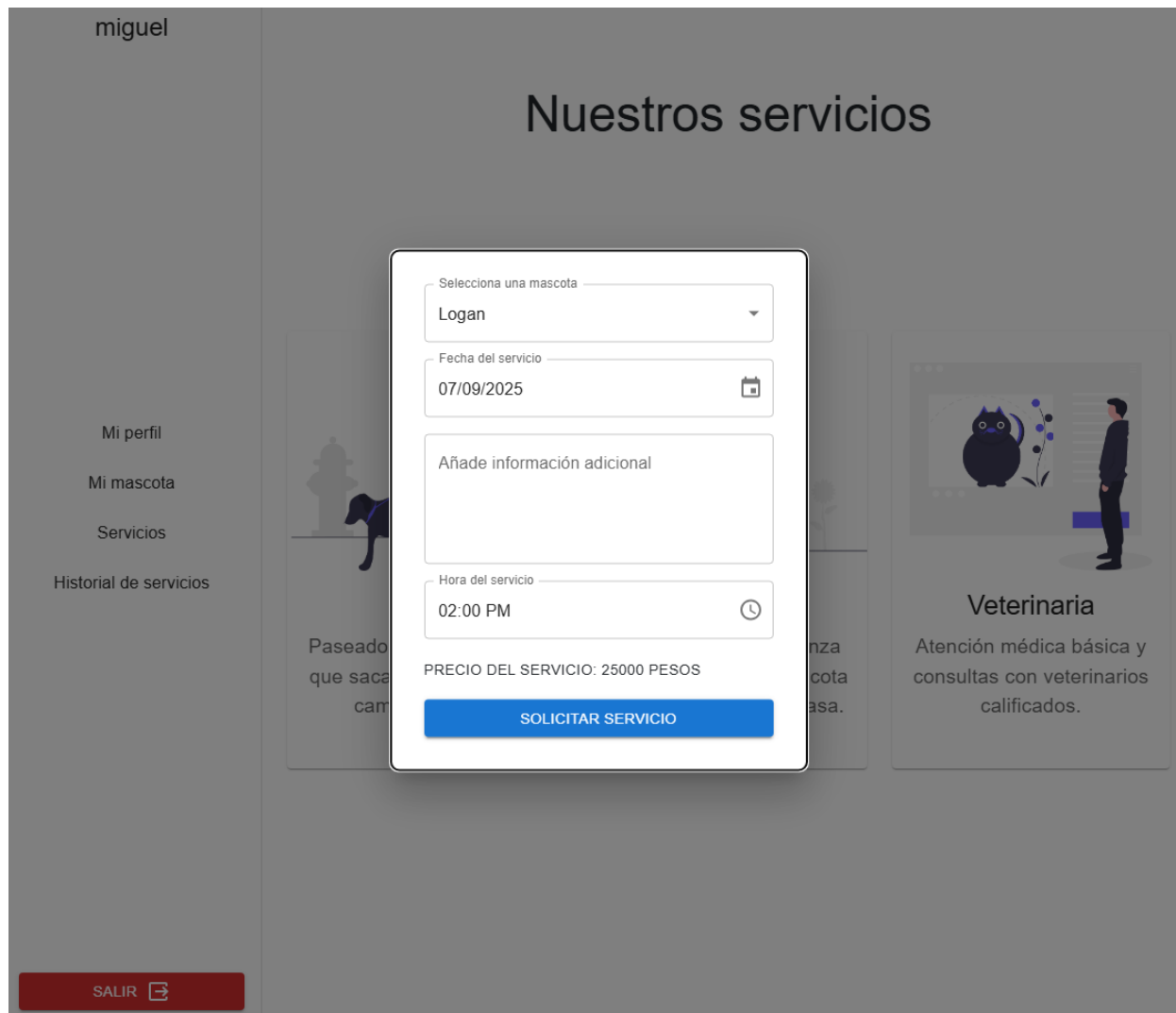
Vista servicios disponibles.



Nota. El usuario puede visualizar los servicios ofrecidos por el prototipo, como cuidado, paseo o atención veterinaria, junto con sus respectivas descripciones.

Figura 48

Vista solicitud de servicio.



Nota. El usuario puede solicitar un servicio específico a través de un formulario. Es obligatorio seleccionar la mascota, indicar la fecha y la hora en que desea que se preste el servicio para enviar la solicitud correctamente. Los precios mostrados en los servicios ofrecidos son ilustrativos y se encuentran basados en un promedio de los valores disponibles en el mercado actualmente.

Figura 49

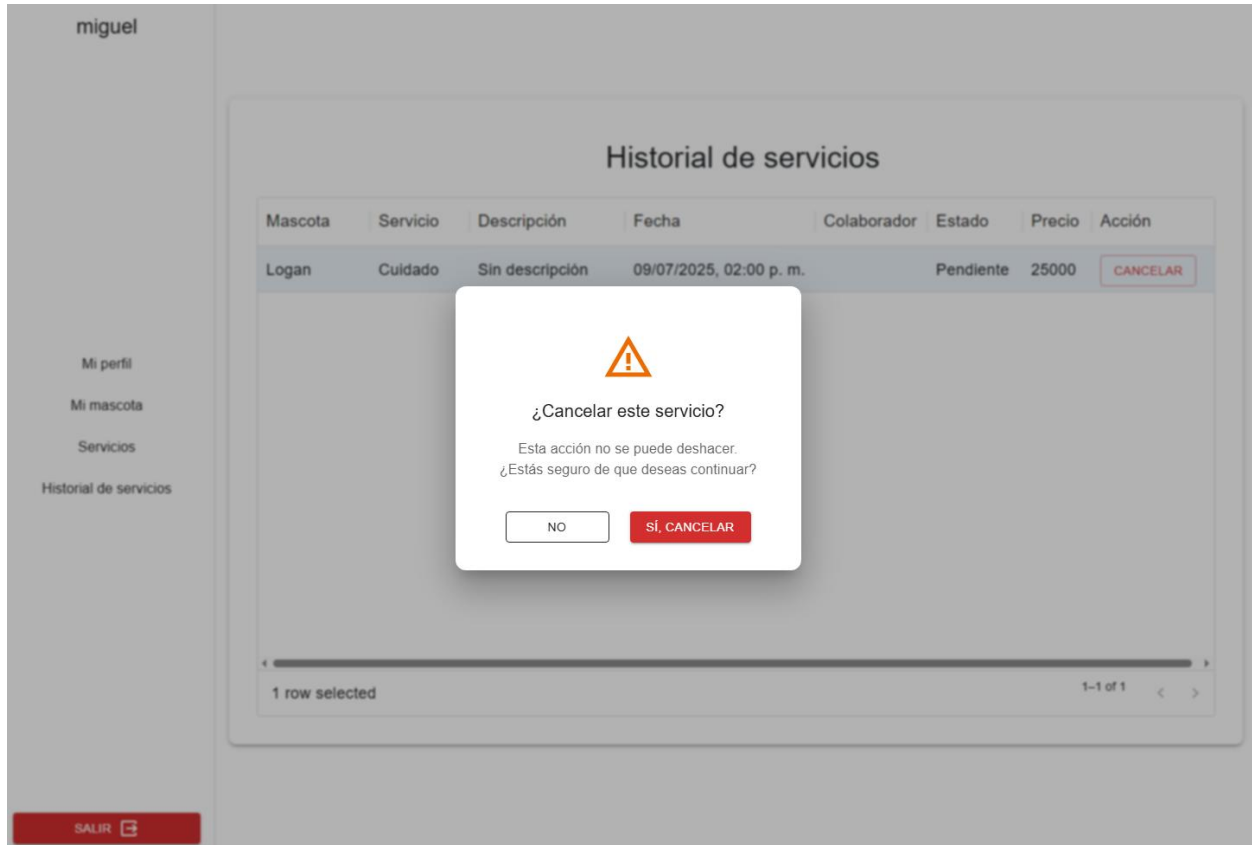
Vista historial de servicios.



Nota. El usuario puede consultar el historial de los servicios que ha solicitado, incluyendo información relevante como la fecha, el tipo de servicio, el estado de la solicitud. Además, desde esta vista tiene la opción de cancelar un servicio en caso de que aún no haya sido iniciado, brindándole un mayor control sobre la gestión de sus solicitudes dentro del prototipo web.

Figura 50

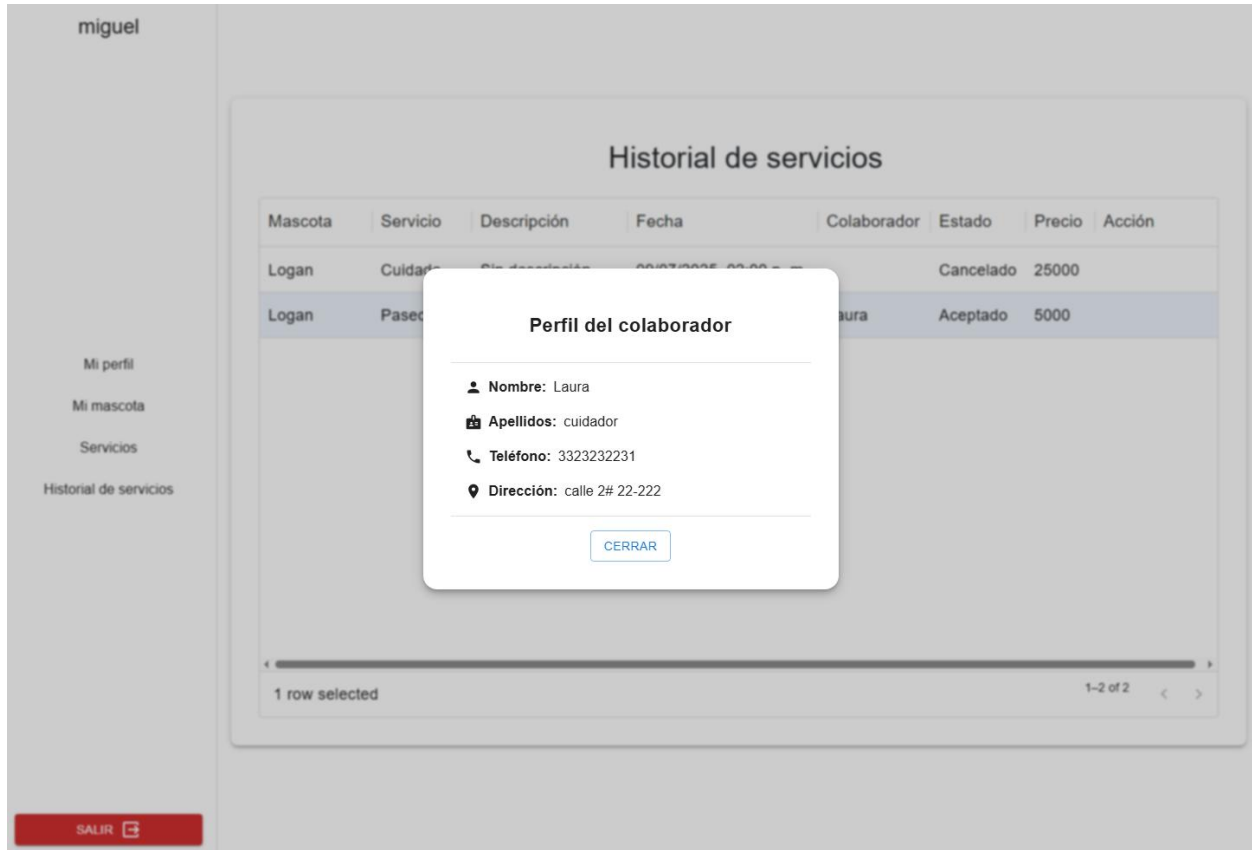
Vista cancelar un servicio.



Nota. El usuario puede tener la opción de cancelar un servicio previamente solicitado si aún no ha sido aceptado. El sistema solicitará confirmación antes de proceder con la cancelación para evitar acciones accidentales.

Figura 51

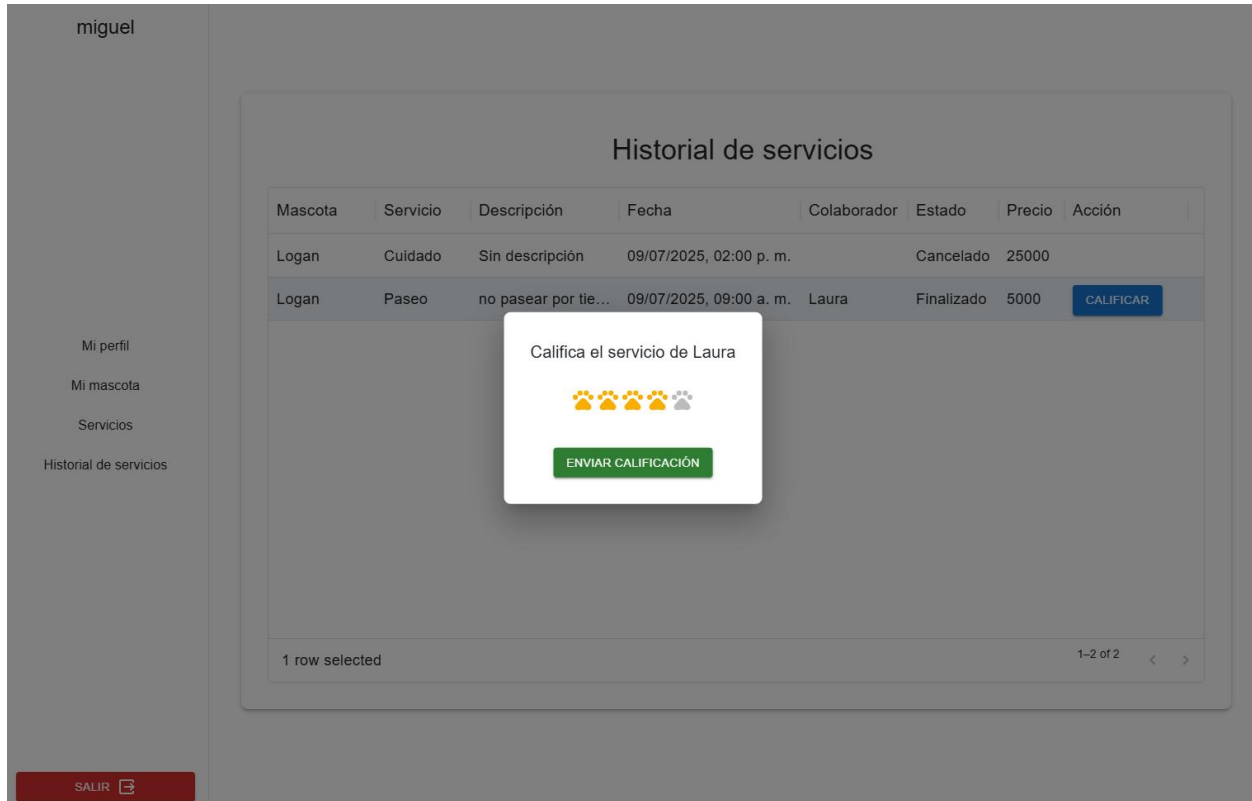
Vista información del colaborador.



Nota. Esta sección muestra los datos relevantes del colaborador asignado a un servicio, incluyendo su nombre, tipo de perfil (cuidador o veterinario) e información de contacto.

Figura 52

Vista calificar servicio.



Nota. El usuario puede evaluar el servicio recibido mediante un sistema de calificación, lo cual contribuye a mantener la calidad del servicio ofrecido por los colaboradores.

Figura 53

Vista historial de servicios.

miguel

Mi perfil

Mi mascota

Servicios

Historial de servicios

Historial de servicios

Mascota	Servicio	Descripción	Fecha	Colaborador	Estado	Precio	Acción
Logan	Cuidado	Sin descripción	09/07/2025, 02:00 p. m.		Cancelado	25000	
Logan	Paseo	no pasear por tie...	09/07/2025, 09:00 a. m.	Laura	Finalizado	5000	👍👍👍👍🐾

1 row selected 1-2 of 2 < >

SALIR 🏠

Nota. El usuario puede consultar un registro detallado de todos los servicios que ha solicitado. La información incluye la fecha del servicio, el tipo de atención recibida, el estado de la solicitud y los datos del colaborador que atendió.

Figura 54

Vista inicial del panel de colaboradores.



Nota. El colaborador accede a una vista principal donde puede visualizar las solicitudes de servicios que aún no han sido tomadas por otros colaboradores. Desde esta sección, tiene la posibilidad de revisar los detalles de cada solicitud y aceptar aquellas que desee atender, gestionando así su participación dentro del prototipo web.

Figura 55

Vista inicial del panel de colaboradores.



Nota. El colaborador puede visualizar las solicitudes de servicios recibidas y acceder a los detalles de cada una. Además, cuenta con la opción de aceptar un servicio mediante el botón “Tomar”, lo que asigna automáticamente la solicitud a su perfil y actualiza el estado del servicio. Si el colaborador hace clic en el nombre de la mascota, puede acceder directamente al perfil de dicha mascota, donde se muestran sus datos básicos (ver figura 58).

Figura 56

Fragmento de código del método FindCollabOrders implementado en el backend.

```

16  export class ordersService {
110  async findCollaborders(userId: string) {
111    try {
112      // 1. Buscar servicios disponibles según el rol del colaborador
113      const user = await this.authService.findOne(userId);
114      const role = user?.role;
115      if (!role) {
116        throw new BadRequestException("Rol de usuario no encontrado");
117      }
118      const services = await this.servicesService.findByRole(role);
119      const serviceIds = services.map((s) => s.id);
120      // 2. Buscar ordens sin colaborador asignado
121      const orders = await this.orderModel.find({
122        serviceId: { $in: serviceIds },
123        colabId: { $in: [null, "" ]},
124      });
125      if (!orders.length) return [];
126      // 3. Buscar mascotas asociadas
127      const petIds = orders.map((inv) => inv.petId);
128      const pets = await this.petService.findByIds(petIds);
129      // 4. Buscar dueños
130      const ownerIds = pets.map((p) => p.ownerId);
131      const uniqueOwnerIds = [...new Set(ownerIds)];
132      const owners = await this.authService.findByIds(uniqueOwnerIds);
133      // 5. Armar respuesta enriquecida
134      return orders.map((inv) => {
135        const pet = pets.find((p) => p._id.toString() === inv.petId.toString());
136        const owner = pet
137          ? owners.find((o) => o._id.toString() === pet.ownerId.toString())
138          : null;
139        const service = services.find(
140          (s) => s._id.toString() === inv.serviceId.toString()
141        );
142        return {
143          id: inv._id,
144          clientName: owner?.name || "N/A",
145          address: owner?.address || "Dirección no disponible",
146          petName: pet?.name || "Mascota desconocida",
147          pet: pet || null,
148          serviceType: service?.serviceType || "Servicio desconocido",
149          description: inv.description || "Sin descripción",
150          date: inv.serviceDate,
151          status: inv.status,
152        };
153      });
154    } catch (error) {
155      console.error("Error in findCollaborders:", error);
156      throw new BadRequestException(
157        "Error retrieving collaborator orders: " + error.message
158      );
159    }

```

Nota. En este fragmento de código se muestra el proceso mediante el cual el sistema obtiene las solicitudes de servicios disponibles para los colaboradores. La función recupera la información asociada al rol del usuario autenticado y filtra las órdenes que no han sido asignadas a ningún colaborador. Posteriormente, asocia cada solicitud con los datos de la mascota, su propietario y el tipo de servicio correspondiente. Este procedimiento permite que los cuidadores y médicos veterinarios visualicen únicamente los servicios disponibles según su perfil.

Figura 57

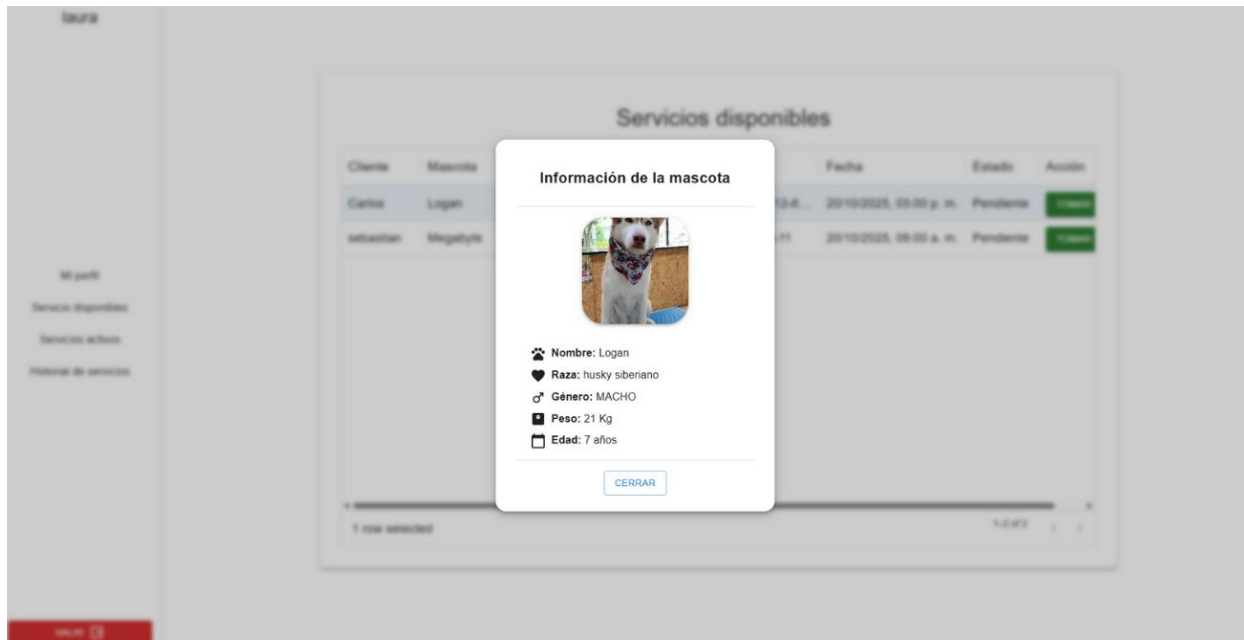
Fragmento de código del método assignorderToColab implementado en el backend

```
165
166   async update(id: string, updateorderDto: UpdateorderDto) {
167     |   return this.orderModel.findByIdAndUpdate(id, updateorderDto).exec();
168   }
169
170   // Asigna una orden a un colaborador
171   async assignorderToColab(orderId: string, colabId: string) {
172     |   try {
173     |     const order = await this.orderModel.findById(orderId).exec();
174     |     if (!order) {
175     |       |   throw new NotFoundException("orden no encontrada");
176     |     }
177     |     if (order.colabId) {
178     |       |   throw new BadRequestException(
179     |       |     |   "Esta orden ya fue tomada por otro colaborador"
180     |       |   );
181     |     }
182     |     order.colabId = colabId;
183     |     order.status = orderStatus.ACEPTADO; // Cambia el estado a ACEPTADA
184
185     |     await order.save();
186     |     return { message: "orden asignada correctamente" };
187     |   } catch (error) {
188     |     |   throw new BadRequestException(
189     |     |     |   "Error al asignar la orden: " + error.message
190     |     |   );
191     |   }
192   }
```

Nota. Este método cumple la función de gestionar la asignación controlada de órdenes a colaboradores, evitando duplicidades y garantizando que cada solicitud sea atendida por un único colaborador disponible. Además, contribuye a la trazabilidad de los servicios, permitiendo identificar qué colaborador aceptó cada orden.

Figura 58

Vista perfil de mascota.



Nota. El colaborador puede ver el perfil de la mascota al hacer clic en su nombre.

Figura 59

Vista servicios tomados por el colaborador.



Nota. El colaborador puede visualizar la información detallada de los servicios que ha aceptado. Además, cuenta con la opción de finalizar el servicio una vez este haya sido prestado. El número de contacto del propietario de la mascota solo se encuentra disponible en esta vista, con el fin de garantizar la privacidad de los usuarios y facilitar la comunicación únicamente durante la prestación del servicio.

Figura 60

Vista historial de servicios ofrecidos.

Laura

Mi perfil

Servicio disponibles

Servicios activos

Historial de servicios

Historial de servicios

Ciente	Mascota	Servicio	Descripción	Fecha ↑	Dirección	Estado	Calificación
miguel	Eva	Paseo	paseo por tierra	01/06/2025, 09:00 a. m.	carrera 21#36-11	Finalizado	🐾🐾🐾🐾🐾
miguel	Xaya	Cuidado	Sin descripción	02/07/2025, 03:10 p. m.	carrera 21#36-11	Finalizado	🐾🐾🐾🐾🐾
miguel	logan	Paseo	Sin descripción	08/07/2025, 08:00 a. m.	carrera 21#36-11	Finalizado	🐾🐾🐾🐾🐾
miguel	Logan	Paseo	no pasear por tie...	09/07/2025, 09:00 a. m.	calle 36 # 30-57	Finalizado	Sin calificación
miguel	Xaya	Cuidado	Sin descripción	11/07/2025, 03:15 p. m.	carrera 21#36-11	Finalizado	🐾🐾🐾🐾🐾
miguel	logan	Cuidado	Sin descripción	16/07/2025, 03:15 p. m.	carrera 21#36-11	Finalizado	🐾🐾🐾🐾🐾
miguel	Eva	Cuidado	Sin descripción	16/07/2025, 06:00 p. m.	carrera 21#36-11	Finalizado	🐾🐾🐾🐾🐾
miguel	Xaya	Paseo	Sin descripción	20/06/2025, 04:00 p. m.	carrera 21#36-11	Finalizado	🐾🐾🐾🐾🐾

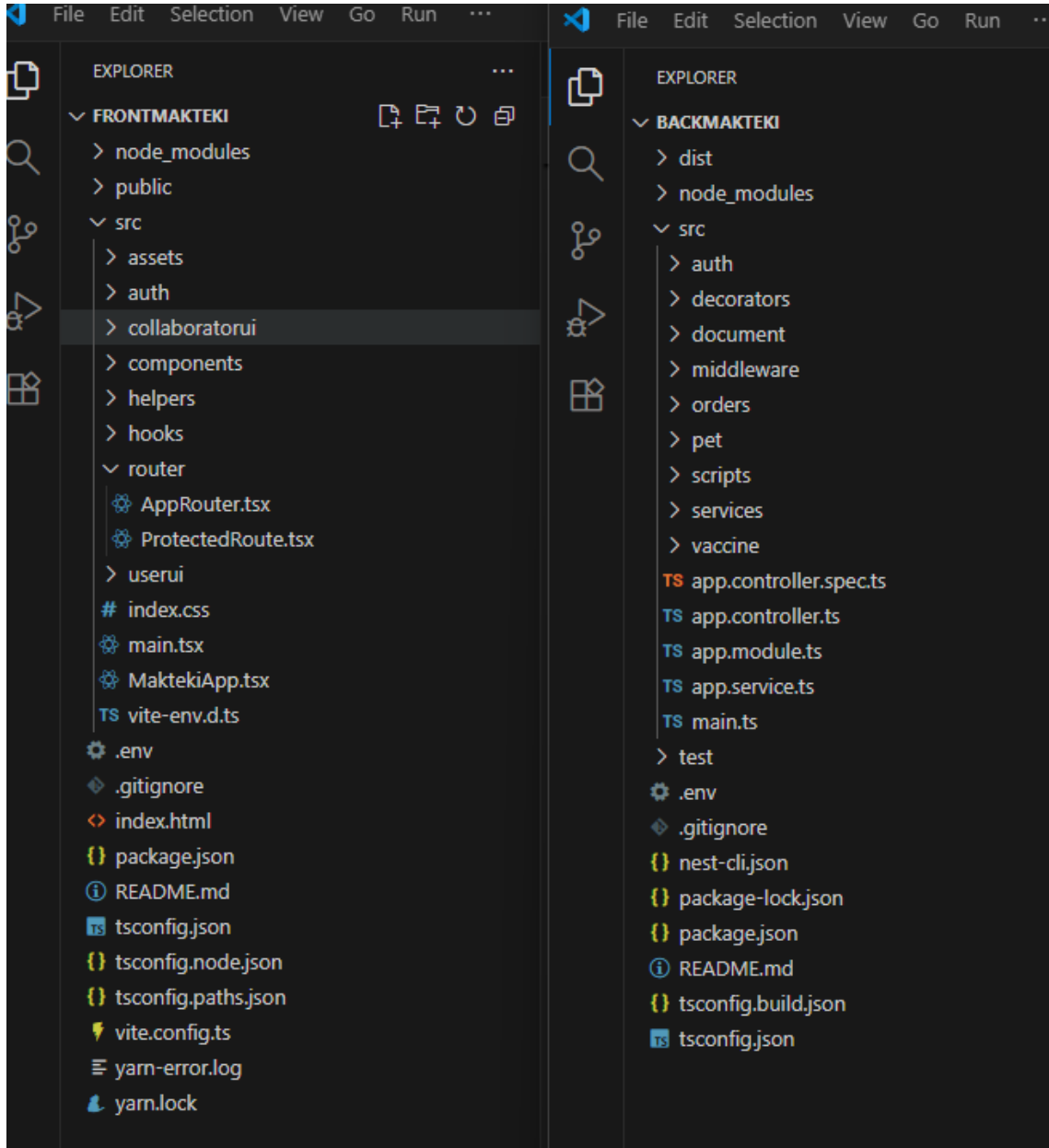
1-8 of 8 < >

SALIR

Nota. El colaborador puede consultar un registro detallado de los servicios que ha prestado, incluyendo información como la fecha, el tipo de servicio, el estado de la solicitud y los datos de la mascota atendida.

Figura 61

Archivo estructura general del proyecto.



Nota. En la figura 61 se observa la estructura final de carpetas del prototipo web, la cual se organiza los diferentes componentes, módulos y recursos del proyecto. Lo que favorece la mantenibilidad

y escalabilidad del proyecto manteniendo una buena legibilidad permitiendo una distribución ordenada de funcionalidades.

5.9 Pruebas de funcionalidad

Se realizaron pruebas de funcionalidad con el propósito de determinar el correcto desempeño de cada prototipo. En cada caso, se planteó un plan de pruebas orientado a verificar los casos de uso definidos como los requerimientos funcionales.

5.9.1 Pruebas primer prototipo

Tabla 18

Pruebas funcionales realizadas al primer prototipo.

Casos de prueba	Resultado
Registro de usuarios en la aplicación	Exitoso
Registro de usuario como colaborador	Exitoso
Inicio de sesión mediante correo electrónico y contraseña registradas	Exitoso
Visualización de los servicios ofrecidos	Exitoso
Solicitud de un servicio	Exitoso
Visualización del perfil de usuario	Exitoso
Creación del perfil de mascota	Exitoso
Visualización del perfil de mascota	Exitoso
Consultar plan de vacunación de mascota	Exitoso
Consultar historial de servicios	Exitoso

5.9.2 Pruebas prototipo final

Tabla 19

Pruebas funcionales realizadas al prototipo final.

Casos de prueba	Resultado
Implementación de pantalla inicial	Exitoso
Registro de usuarios	Exitoso
Registro de usuarios como colaboradores	Exitoso
Inicio de sesión mediante correo electrónico y contraseña registradas	Exitoso
Mejora visual en los formularios de registro	Exitoso
Navegación entre formularios de registro y pantalla inicial	Exitoso
Solicitud de un servicio	Exitoso
Confirmación de información de la solicitud de servicio	Exitoso
Mejora visual en la solicitud de servicio	Exitoso
Visualización de perfil de usuario	Exitoso
Edición de perfil de usuario	Exitoso
Visualización de la sección “Mis mascotas”	Exitoso
Agregar una mascota	Exitoso
Edición de información de mascota	Exitoso
Agregar una mascota nueva	Exitoso
Consulta del plan de vacunación de la mascota	Exitoso
Registro de vacunas	Exitoso
Consulta del historial de servicios	Exitoso
Visualización los servicios activos	Exitoso
Permitir al usuario cancelar un servicio	Exitoso

Visualización de la información del colaborador asociado a un servicio	Exitoso
Calificar un servicio finalizado	Exitoso
Visualización de servicios disponibles para colaboradores	Exitoso
Visualización de servicios disponibles para veterinarios	Exitoso
Consulta de servicios tomados por colaboradores/veterinarios	Exitoso
Visualización de información de la mascota a la cual se prestará un servicio	Exitoso
Finalización de un servicio por parte del colaborador	Exitoso
Consulta del historial de servicios con calificaciones	Exitoso

7. Conclusiones

El desarrollo del prototipo web para servicios de cuidado de mascotas permitió comprobar la viabilidad de implementar un prototipo orientada a conectar de manera confiable a dueños de mascotas con cuidadores y veterinarios. A través de la metodología incremental, se desarrolló un proceso de construcción progresivo que facilitó la integración de mejoras continuas en cada fase, consolidando un sistema funcional y adaptable a las necesidades de los usuarios.

Las pruebas de funcionalidad demostraron que tanto el primer como el segundo prototipo respondieron satisfactoriamente a los casos de uso definidos, cumpliendo con los requerimientos funcionales planteados. Esto valida la correcta implementación de funcionalidades clave como el registro y autenticación de usuarios, la gestión de perfiles de usuario y de mascotas, el plan de vacunación y el historial de servicios, consolidando un prototipo intuitivo y confiable. Así mismo, la realimentación obtenida de los usuarios potenciales permitió realizar ajustes de usabilidad y diseño que fortalecieron la experiencia de navegación. En este sentido, el proyecto no solo representa una contribución tecnológica, sino también una propuesta innovadora que impacta positivamente en el bienestar animal y la confianza de los propietarios simultáneamente.

Finalmente, se resalta que esta iniciativa aporta al contexto local al ofrecer una alternativa accesible y confiable, con potencial de escalabilidad hacia escenarios regionales y nacionales.

8. Recomendaciones

Se propone integrar un sistema de pagos en línea a través de plataformas como PayU, PSE, ePayCo o Mercado Pago, que permita a los clientes realizar transacciones seguras directamente desde el prototipo, facilitando la gestión de pagos por los servicios solicitados.

Incorporación de un sistema de notificaciones por mensajes SMS, con el fin de mantener informados a los usuarios sobre actualizaciones relacionadas a sus servicios, solicitudes o recordatorios de vacunación.

Integración de mapas interactivos para la localización de cuidadores y veterinarios cercanos, optimizando la conexión entre usuarios y colaboradores según su ubicación.

Incorporación de un módulo de comercio electrónico que permita a los usuarios adquirir suplementos, alimentos especializados y otros productos para el cuidado de mascotas directamente desde el prototipo.

Ampliación del plan de vacunación hacia un historial completo con la información de un historial médico e información de salud de cada mascota.

Referencias Bibliográficas

- Amazon (s.f.). ¿Qué es el e-commerce?.
https://sell.amazon.com/es/learn/what-is-ecommerce/?mons_sel_locale=es_US.
- Azevedo, M. A., Mascarenhas, N. (24 de junio de 2019). Colombian On-Demand delivery unicorn Rappi raises \$1B from SoftBank. Crunchbase News.
<https://news.crunchbase.com/venture/colombian-unicorn-rappi-reportedly-raising-1b-from-softbank/>.
- Cifras y conceptos. Presentación Resultados Polimétrica febrero 2024.pdf. Google Docs. (4 de septiembre de 2024).
<https://drive.google.com/file/d/1jQDJAMFgBwFkM67IUFToq5UZiAThQ2qY/view>.
- Cifras y conceptos. Presentación Resultados Polimétrica marzo 2023. (4 de septiembre de 2024).
<https://www.cifrasyconceptos.com/wp-content/uploads/2023/04/Resultados-Polimetrica-marzo-2023-V3-publicacion.pdf>.
- Cuidado de mascotas Medellín a Domicilio. Ohana. (5 de diciembre de 2023).
<https://ohana.com.co/>.
- Documentation for Visual Studio Code. (7 de diciembre de 2023).
<https://code.visualstudio.com/docs>.
- Documentation | NestJS A Progressive Node.js Framework. (2 de abril de 2024).
<https://docs.nestjs.com/>.
- Laika. (s. f.). Acerca de nosotros | Laika Mascotas. Laika Mascotas.
https://laika.com.co/about_us/bog
- Mateos, S. M. U. (26 de octubre de 2018). ¿Qué es Amazon? Actualidad eCommerce.
<https://www.actualidadecommerce.com/que-es-amazon/>.
- MDN Web Docs. Fundamentos de JavaScript. (7 de diciembre de 2023).
https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics.
- MDN Web Docs. ¿Cómo se utiliza Github pages? (7 de diciembre de 2023).
https://developer.mozilla.org/es/docs/Learn/Common_questions/Tools_and_setup/Using_GitHub_pages.

- MDN Web Docs. Trabajando con JSON. (7 de diciembre de 2023).
<https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>.
- Mercado Libre. (21 de junio de 2023). Time.
<https://time.com/collection/time100-companies-2023/6285124/mercado-libre-leaders/>.
- MongoDB. ¿Qué Es MongoDB? (7 de diciembre de 2023).
<https://www.mongodb.com/es/what-is-mongodb>.
- NextU LATAM. ¿Qué es React JS y cómo funciona? (15 de septiembre de 2022)
<https://www.nextu.com/blog/que-es-react-js-como-funciona-rc22/>.
- Petcrew.co. (4 de diciembre de 2023).
<https://petcrew.co/>.
- React.dev. Inicio rápido. (7 de diciembre de 2023).
<https://es.react.dev/learn>.
- Rover. dog sitters & walkers. App Store. (7 de diciembre de 2023).
<https://apps.apple.com/es/app/rover-dog-sitters-walkers/id547320928?ign-mpt=uo%3D4>.
- Stanford Graduate School of Business. The rise of Mercado libre. (5 de diciembre de 2023)
<https://www.gsb.stanford.edu/faculty-research/case-studies/rise-mercado-libre>.
- Typescriptlang.org. JavaScript with syntax for types. (2 de abril de 2024).
<https://www.typescriptlang.org/>.
- Universidad EAN. Mascotas, los otros héroes de la pandemia. (1 de diciembre de 2023).
<https://universidadean.edu.co/blog/mascotas-los-otros-heroes-de-la-pandemia>.
- Vázquez, S. (23 de julio de 2020). Cómo funciona AliExpress: cómo comprar en AliExpress, usar cupones y más. Sabes Aprender.
<http://sabesaprender.com/como-comprar-en-aliexpress/>.
- Wag.Co. Wag! – any day, any time pet care. (5 de diciembre de 2023).
<https://wag.co/>.

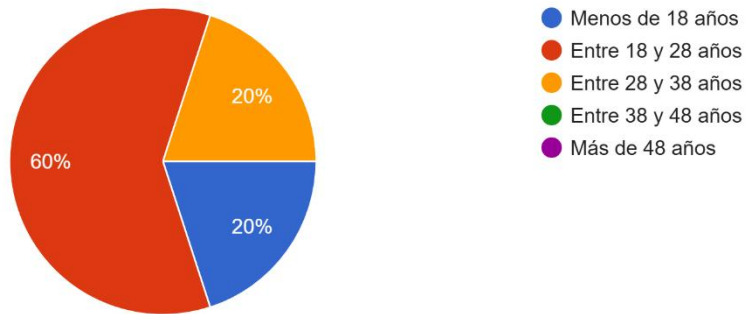
Apéndices

Apéndice A. análisis de acercamiento inicial

Al inicio de este proyecto se formularon una serie de preguntas de acercamiento inicial con el propósito de identificar el alcance y recopilar información relevante para el desarrollo del prototipo. Dichas preguntas abordaron aspectos relacionados con el uso de la tecnología como herramienta para la solución de problemáticas, la búsqueda de empleo y las percepciones de los participantes, cuyo aporte fue fundamental para orientar el crecimiento del prototipo y las iteraciones realizadas durante su desarrollo web.

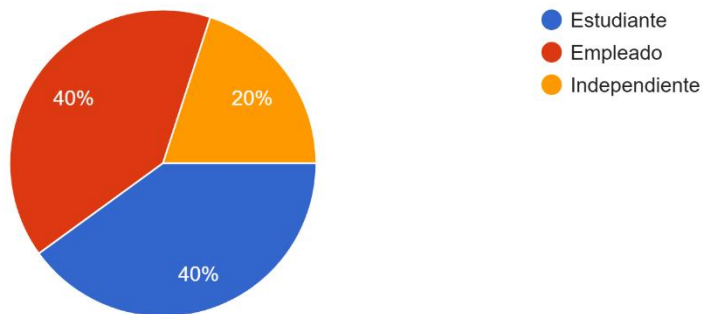
Edad del encuestado

5 respuestas



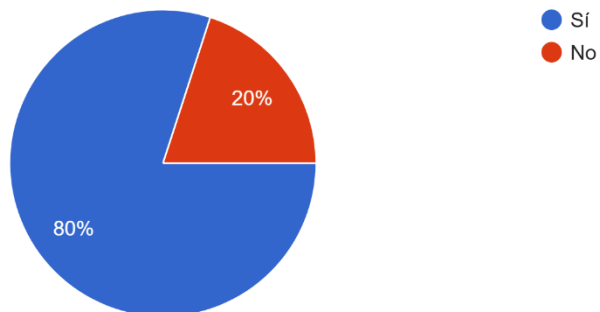
Su ocupación actual es

5 respuestas



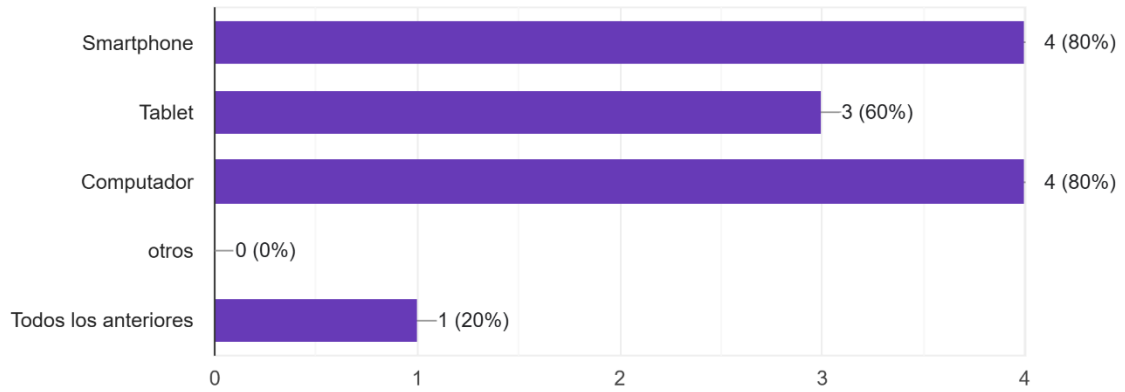
Es usted dueño de una mascota

5 respuestas



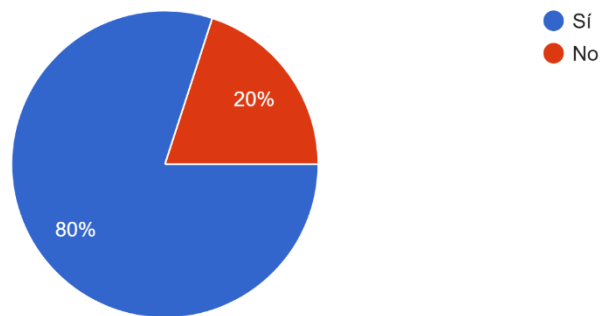
¿cuenta con usted con alguno de los siguientes dispositivos tecnológicos?

5 respuestas



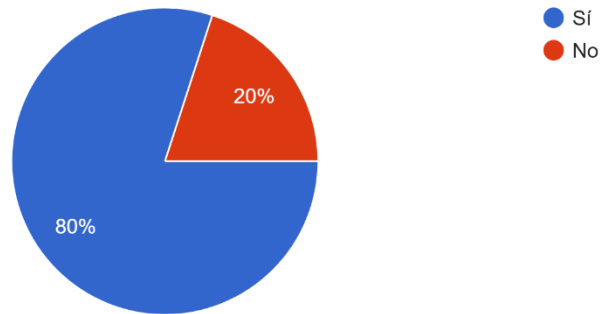
¿ha adquirido usted alguna vez productos o servicios en línea?

5 respuestas



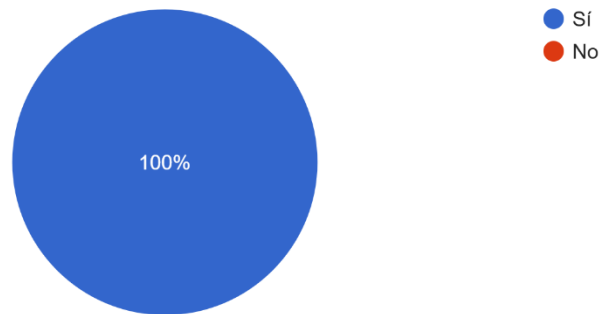
¿Considera necesarias sitios web enfocados a la búsqueda de trabajo hoy en la actualidad?

5 respuestas



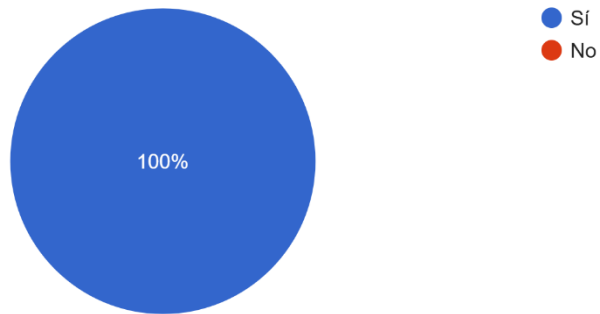
¿Confiaría su información personal con el fin de prestar o adquirir un servicio a este tipo de sitios web?

5 respuestas



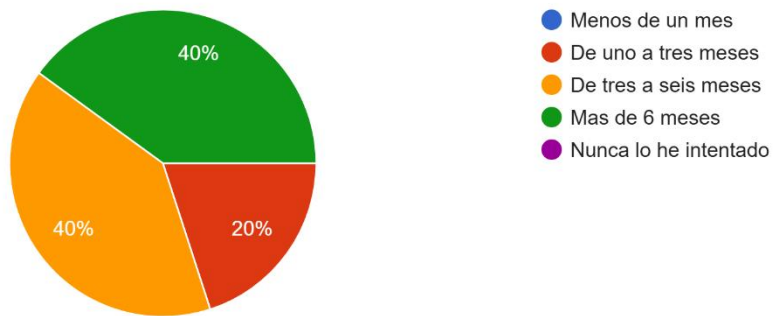
¿ha intentado encontrar trabajo en estos sitios web?

5 respuestas



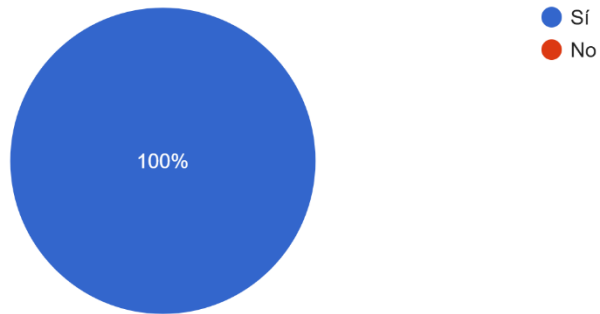
¿Cuánto tiempo tarda en buscar empleo mediante esto sitios web?

5 respuestas



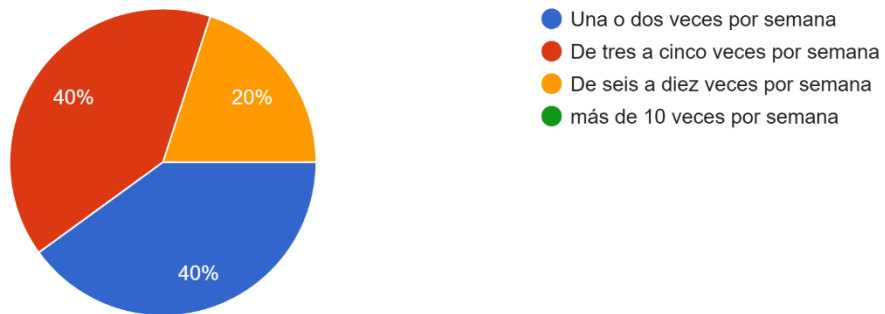
¿Usaría un aplicativo web enfocado en buscar cuidadores y veterinarios para macotas que brinde confiabilidad?

5 respuestas



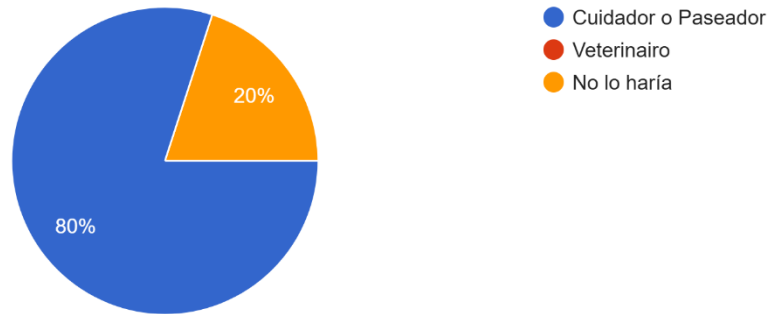
¿Qué tan frecuentemente la usaría?

5 respuestas



¿Si saliera al mercado un aplicativo como este estaría dispuesto a unirse al equipo como colaborador?

5 respuestas



¿Qué funcionalidades cree que debería tener un aplicativo de este tipo?

5 respuestas

poder solicitar un cuidador o paseador a cualquier hora del dia

me gustaría el servicio de veterinaria

Poder encontrar paseadores confiables

Alguna forma de tener un registro de vacunas

poder calificar a los cuidadores y veterinarios

¿cuál es su opinión sobre esta propuesta de aplicativo web?

5 respuestas

es una idea bastante llamativa y ayudaría al empleo

buena

innovadora

novedosa

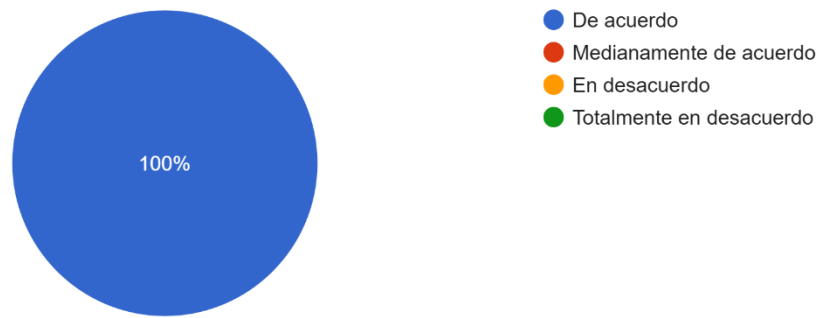
llamativa

Apéndice B. Análisis preguntas aplicadas al primer prototipo

Al entregar el prototipo final se formularon preguntas con el propósito de conocer las percepciones de los usuarios sobre el funcionamiento del prototipo, así como sus sugerencias para futuras versiones. En esta etapa, se preguntó nuevamente al mismo grupo de cinco personas que participó en las preguntas de acercamiento inicial del apéndice A.

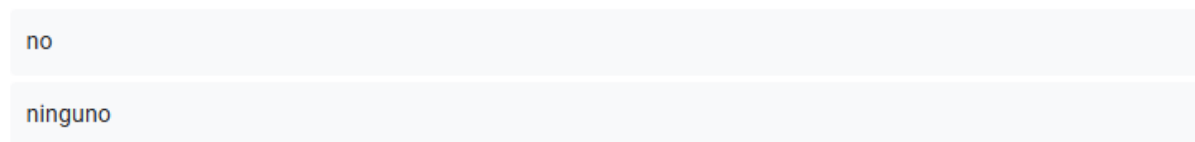
¿le parece útil el aplicativo para encontrar un cuidado o veterinario?

5 respuestas



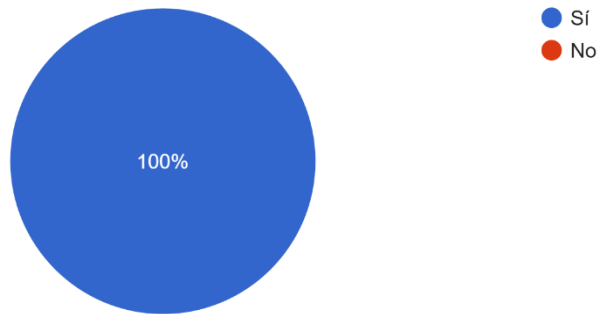
¿Tuvo algún problema al momento de solicitar un servicio?

5 respuestas



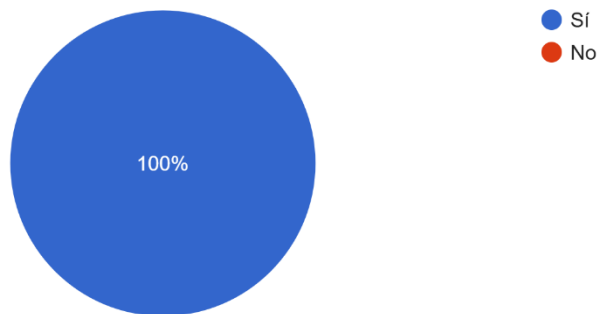
¿son los servicios ofrecidos lo que esperaba encontrar?

5 respuestas



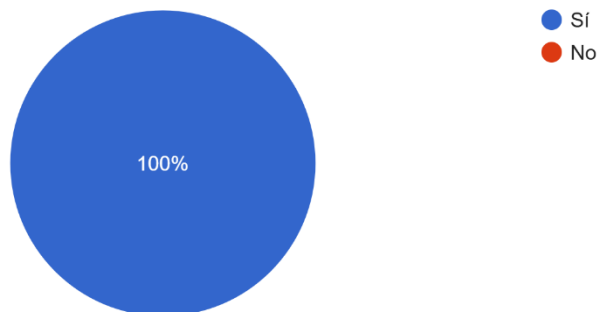
¿Al registrarse y crear su perfil, le parece suficiente la información solicitada?

5 respuestas



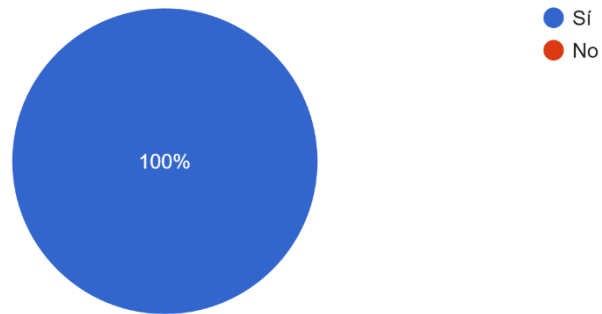
¿le resulta fácil el uso del aplicativo?

5 respuestas



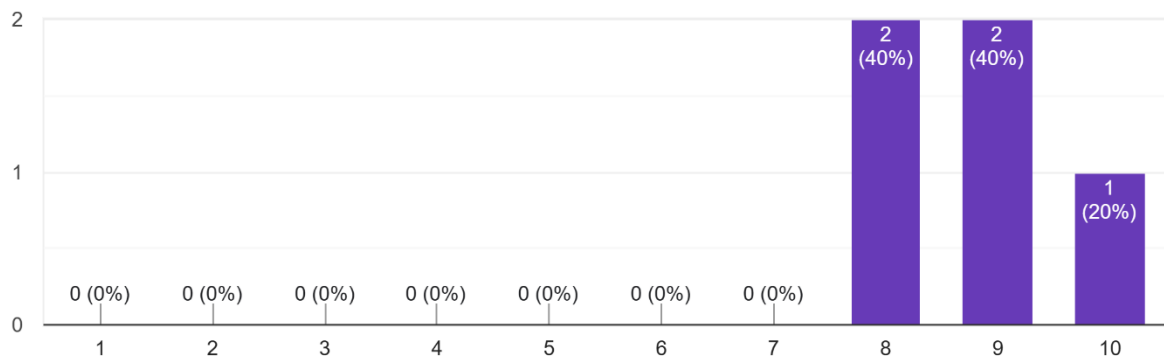
¿el aplicativo le parece una oferta novedosa?

5 respuestas



en escala del 1 al 10 ¿qué tan útil le resulta el aplicativo? siendo 1 para nada útil y 10 totalmente útil

5 respuestas



¿cuál es su opinión respecto al aplicativo?

5 respuestas

me parece que se prestan servicios básicos para empezar

llamativa, es fácil de usar

Bastante útil, pero hace falta la opción de calificar los cuidadores

deberían agregar más servicios

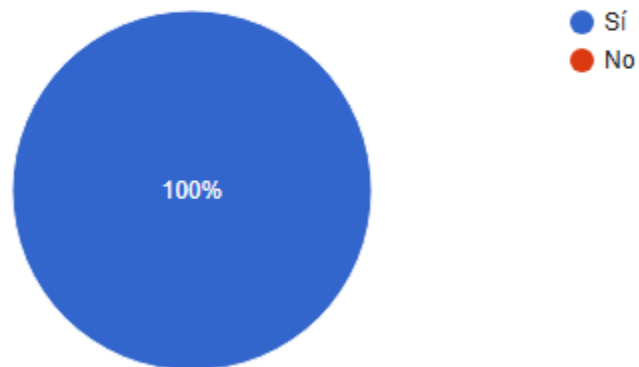
buena, deben agregar la opción de calificar un servicio

Apéndice C. Análisis preguntas aplicadas al segundo prototipo

Al entregar el segundo prototipo se formularon preguntas con el propósito de conocer las percepciones de los usuarios sobre el funcionamiento del prototipo, así como sus sugerencias para futuras versiones. En esta etapa, se aplicaron nuevas preguntas a una muestra más amplia conformada por treinta personas pertenecientes a los barrios de San Alonso, Álvarez, Antonia Santos y Cabecera, entre las cuales se encontraban un médico veterinario, doce cuidadores y diecisiete propietarios de mascotas.

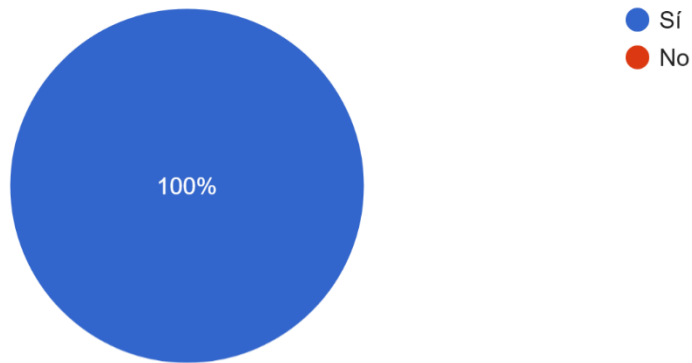
¿El prototipo funciona correctamente?

30 respuestas



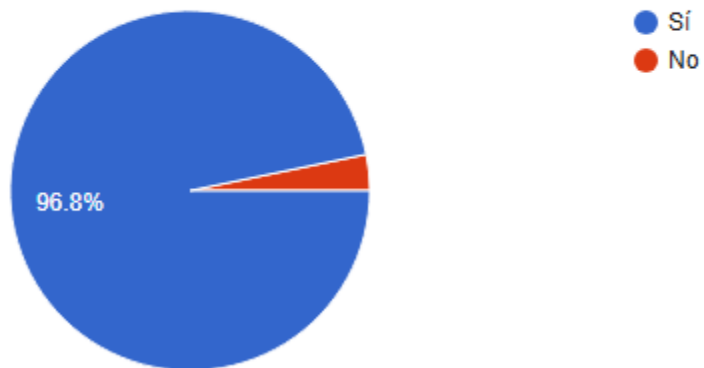
¿Contemplantaría el prototipo para solicitar servicios de cuidado o veterinaria a sus mascotas?

30 respuestas



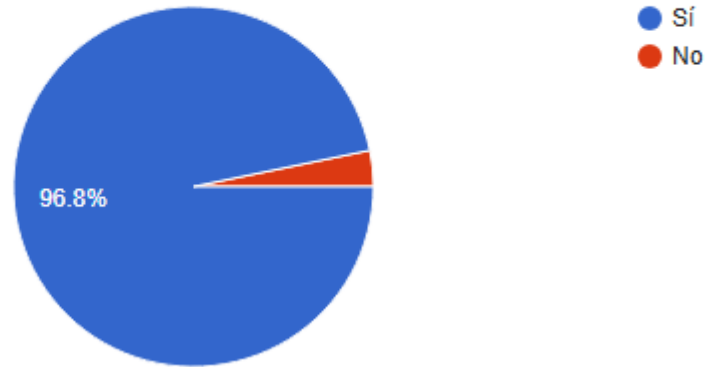
¿Contemplantaría el prototipo como oferta laboral y prestar servicios de cuidado de mascotas y veterinaria?

30 respuestas



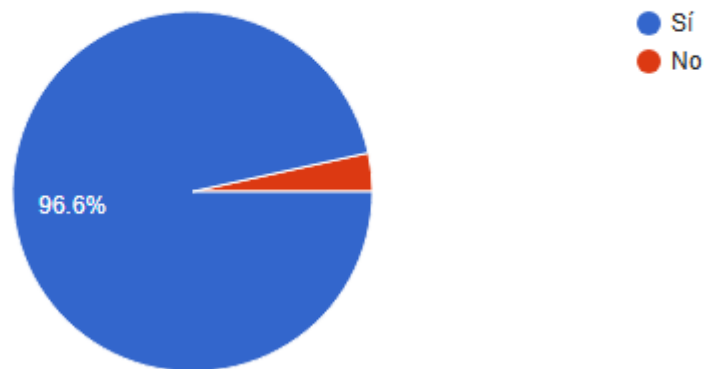
¿Usaría el prototipo con frecuencia?

30 respuestas



¿Recomendaría el prototipo a amigos y familiares?

30 respuestas



¿qué funcionalidades considera que deberían agregarse en un futuro?

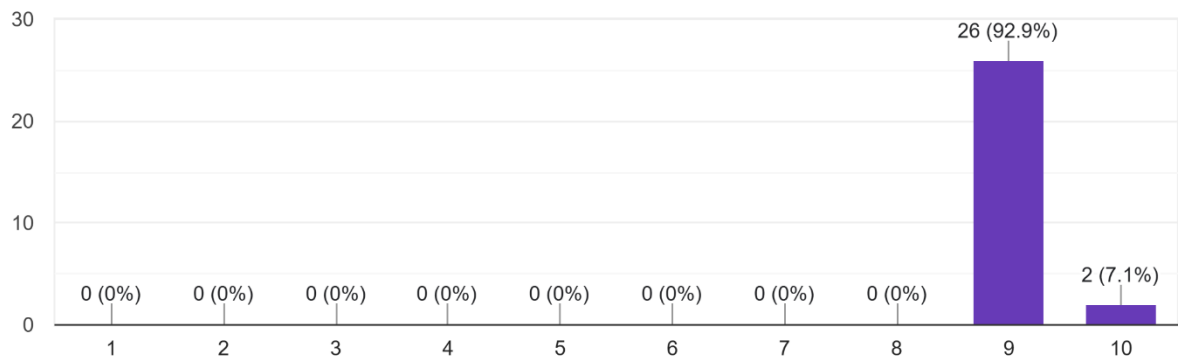
22 respuestas

- Agregar un servicio de compra de suplementos
- agregar funcionalidades de pagos directos
- Alguna formad de comprar alimentos
- Algunos servicios como bañar a mi mascota
- Agregar más Servicios

Nota. En esta sección se presentan únicamente cinco de las respuestas más frecuentes obtenidas en el formulario aplicado.

Califique el prototipo en una escala del uno al diez, uno siendo la calificación más baja y diez la más alta

28 respuestas



Califique su nivel de satisfacción dentro del prototipo

30 respuestas

