



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
T U N J A

TÍTULO PASANTÍA

SISTEMA INFORMÁTICO PARA EL REGISTRO Y SEGUIMIENTO DE MASCOTAS EN
LA CIUDAD DE TUNJA

PROPONENTE(S)

López Pérez Anthony Jardiel
1007673204
2271557

DIRECTOR

Diego Alejandro Vela Beltrán

Tunja
05 Diciembre 2024

CONTENIDO

<u>1.</u>	<u>FICHA TÉCNICA DE LA PASANTÍA.....</u>	<u>6</u>
<u>2.</u>	<u>PLANTEAMIENTO DEL PROBLEMA.....</u>	<u>7</u>
<u>3.</u>	<u>JUSTIFICACIÓN</u>	<u>11</u>
<u>4.</u>	<u>OBJETIVOS</u>	<u>13</u>
4.1.	OBJETIVO GENERAL	13
4.2.	OBJETIVOS ESPECÍFICOS	13
<u>5.</u>	<u>ESTADO DEL ARTE</u>	<u>14</u>
<u>6.</u>	<u>DESARROLLO METODOLÓGICO</u>	<u>18</u>
6.1.	ROLES Y RESPONSABILIDADES	18
6.2.	ARTEFACTOS DE SCRUM.....	19
6.3.	TABLERO KANBAN EN TRELLO	19
<u>7.</u>	<u>DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS.....</u>	<u>23</u>
7.1.	ACTIVIDADES REALIZADAS EN EL PROYECTO.....	23
7.1.1.	DETALLES DEL DESARROLLO DE ACTIVIDADES.....	24
7.2.	HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL PROYECTO	34
<u>8.</u>	<u>RESULTADOS</u>	<u>35</u>
8.1.	CUMPLIMIENTO DE LOS OBJETIVOS ESPECÍFICOS.....	35
8.1.1.	ESPECIFICACIÓN DE REQUERIMIENTOS	35
8.1.2.	DISEÑO DE LA APLICACIÓN	35
8.1.3.	DESARROLLO DE LA APLICACIÓN.....	36
8.1.4.	PRUEBAS FUNCIONALES.....	36
8.2.	DESPLIEGUE Y IMPLEMENTACIÓN	37
8.2.1.	DESPLIEGUE DEL BACKEND	37
8.2.2.	IMPLEMENTACIÓN DE LA BASE DE DATOS	37
8.2.3.	DESPLIEGUE DEL FRONTEND	37

8.2.4. PUBLICACIÓN EN GOOGLE PLAY STORE Y APP STORE	38
8.3. IMPACTO Y BENEFICIOS DEL SISTEMA.....	39
8.3.1. GESTIÓN DE INFORMACIÓN DE MASCOTAS	39
8.3.2. LOCALIZACIÓN DE MASCOTAS EXTRAVIADAS	39
8.3.3. BIENESTAR Y SALUD DE LAS MASCOTAS.....	39
8.4. APOORTE DE INGENIERÍA	39
8.4.1. ORGANIZACIÓN Y ESTRUCTURA DEL CÓDIGO.....	39
8.4.2. PRINCIPIOS DE PROGRAMACIÓN	40
8.4.3. PRUEBAS Y CALIDAD DE SOFTWARE.....	40
8.4.4. GESTIÓN DE CONTROL DE VERSIONES.....	40
8.4.5. SEGURIDAD	41
<u>9. CONCLUSIONES.....</u>	<u>42</u>
<u>10. REFERENCIAS</u>	<u>44</u>
<u>11. ANEXOS.....</u>	<u>45</u>
11.1. CONTRATO DE CONFIDENCIALIDAD.....	45
11.1.1. ELEMENTOS CUBIERTOS POR EL CONTRATO DE CONFIDENCIALIDAD	48
11.2. ESPECIFICACIÓN ISO/IEC/IEEE 42010	49
11.3. ESPECIFICACIÓN DE REQUERIMIENTOS.....	50
11.4. DIAGRAMAS	56
11.4.1. ANEXO DIAGRAMA DE CLASE.....	56
11.4.2. ANEXO DIAGRAMA ENTIDAD RELACIÓN	57
11.4.3. ANEXO CASOS DE USO	58
11.4.4. ANEXO DIAGRAMA DE DESARROLLO.....	64
11.4.5. ANEXO DIAGRAMA FÍSICO	65
11.4.6. ANEXO DIAGRAMAS DE FLUJO	65
11.4.7. ANEXO LISTAS DE CHEQUEO	67

TABLA DE FIGURAS

Figura 1. Árbol de Problemas	8
------------------------------------	---

Figura 2. Matriz de stakeholders	10
Figura 3. Años de publicación de los artículos en relación con aplicaciones para mascotas	14
Figura 4. Documentos por territorio en Scopus	15
Figura 5. Visualización palabras clave VOSviewer	16
Figura 6. Metodología Scrum.....	18
Figura 7. Tablero Kanban en Trello.....	21
Figura 8. Tablero Kanban físico en la oficina	22
Figura 9. Implementación base de datos local	26
Figura 10. Implementación del backend.....	28
Figura 11. Pruebas unitarias en Postman.	29
Figura 12. Implementación del frontend, estructura de carpetas desarrolladas.....	30
Figura 13. Implementación del frontend y backend.....	31
Figura 14. Versionamiento y despliegue del backend.	32
Figura 15. Versionamiento y despliegue del frontend.....	32
Figura 16. Aplicación en Servidores en Google Cloud Platform.	37
Figura 17. Aplicación en Play Store	38
Figura 18. Aplicación en App Store.....	38
Figura 19: Diagrama de ISO/IEC/IEEE 42010	49
Figura 20: Contenido del documento IEEE 830	51
Figura 21: Inicio del documento IEEE 830	52
Figura 22: Parte final del inicio del documento.....	53
Figura 23: Requisitos Específicos.....	54
Figura 24: Requerimientos no funcionales.	55
Figura 25: Diagrama de clases.	56
Figura 26: Diagrama entidad relación.	57
Figura 27: Caso de uso. Sistema.....	58
Figura 28: Caso de uso. Sistema de enfermedades.....	59
Figura 29: Caso de Uso. Sistema de la mascota.	60
Figura 30: Caso de uso. Sistema de tratamiento.	61
Figura 31: Caso de uso. Sistema de vacunas.	62
Figura 32: Caso de uso. Sistema de veterinario.....	63

Figura 33: Diagrama de desarrollo.....	64
Figura 34: Diagrama de desarrollo.....	65
Figura 35: Diagrama de flujo inicio y registro de sesión.	66
Figura 36: Diagrama de flujo pantalla inicial y registro de información para la mascota. ..	66
Figura 37: Diagrama de flujo para el apartado del action sheet de la aplicación.	66
Figura 38: Diagrama de flujo escaneo de código QR.	67
Figura 39: Lista de chequeo en Excel módulo sistema.	67
Figura 40: Lista de chequeo en Excel módulo sistema 2.	68
Figura 41: Lista de chequeo en Excel Pets.	68
Figura 42: Lista de chequeo en Excel vacunas.	69
Figura 43: Lista de chequeo en Excel Tratamiento.	69
Figura 44: Lista de chequeo en Excel Enfermedades.	70
Figura 45: Lista de chequeo en Excel Veterinario.	70

1. FICHA TÉCNICA DE LA PASANTÍA

Título	Sistema informático para el registro y seguimiento de mascotas en la ciudad de Tunja
Nombre Estudiante	Anthony Jardiel López Pérez
Documento estudiante	1.007.673.204
Correo electrónico estudiante	anthony.lopez@usantoto.edu.co
Director	Diego Alejandro Vela Beltrán
Entidad o sector donde realizará la pasantía	Soluciones de Tecnología eSMART-TEK
Lugar de ejecución de la pasantía	Tunja, Boyacá
Duración	6 meses

Los abajo firmantes confirman que todos los datos incluidos en la presente propuesta son correctos y verídicos, que no incumplen ninguna ley o norma vigente (incluir nombres y firmas de estudiantes y director).



Firma del autor
Anthony Jardiel López Pérez



Firma del director
Diego Alejandro Vela Beltrán

2. PLANTEAMIENTO DEL PROBLEMA

En la ciudad de Tunja, persiste la ausencia de un sistema eficiente que permita el manejo integral de la información relacionada con las mascotas. Aunque existe una plataforma Informática a nivel nacional destinada exclusivamente al registro de mascotas caninas pertenecientes a razas de manejo especial, según lo estipulado en el artículo 126 del Código Nacional de Policía, no se ha desarrollado aún una solución completa para abarcar todas las necesidades de gestión de mascotas en la ciudad (*Ley 1801 de 2016 Congreso de La República de Colombia*, n.d.).

Además, se ha identificado una normativa (Artículo 108-I Capítulo XIII Nuevo de la Ley 746 DE 2002 del Gestor Normativo de la Función Pública) que requiere a los ciudadanos registrar a sus mascotas consideradas peligrosas en las respectivas alcaldías de los diferentes municipios de Boyacá (*Ley 746 de 2002 - Gestor Normativo - Función Pública*, n.d.). A pesar de la existencia de aplicaciones colombianas como Laika (*Laika Mascotas - Universo Peludo*, n.d.), Distrito Appnimal (*Distrito Appnimal, Una App Que Marca Tendencia | Instituto Distrital de Protección y Bienestar Animal*, n.d.) y Movet, (*Servicios, Consultas y Clínicas Veterinarias 24 Horas • Movet*, n.d.) que ofrecen diversos servicios relacionados con el cuidado y la gestión de mascotas, ninguna de ellas satisface plenamente la necesidad de administrar la información básica de las mascotas, su historial médico, identificación y ubicación mediante un código QR (Serrano, 2020).

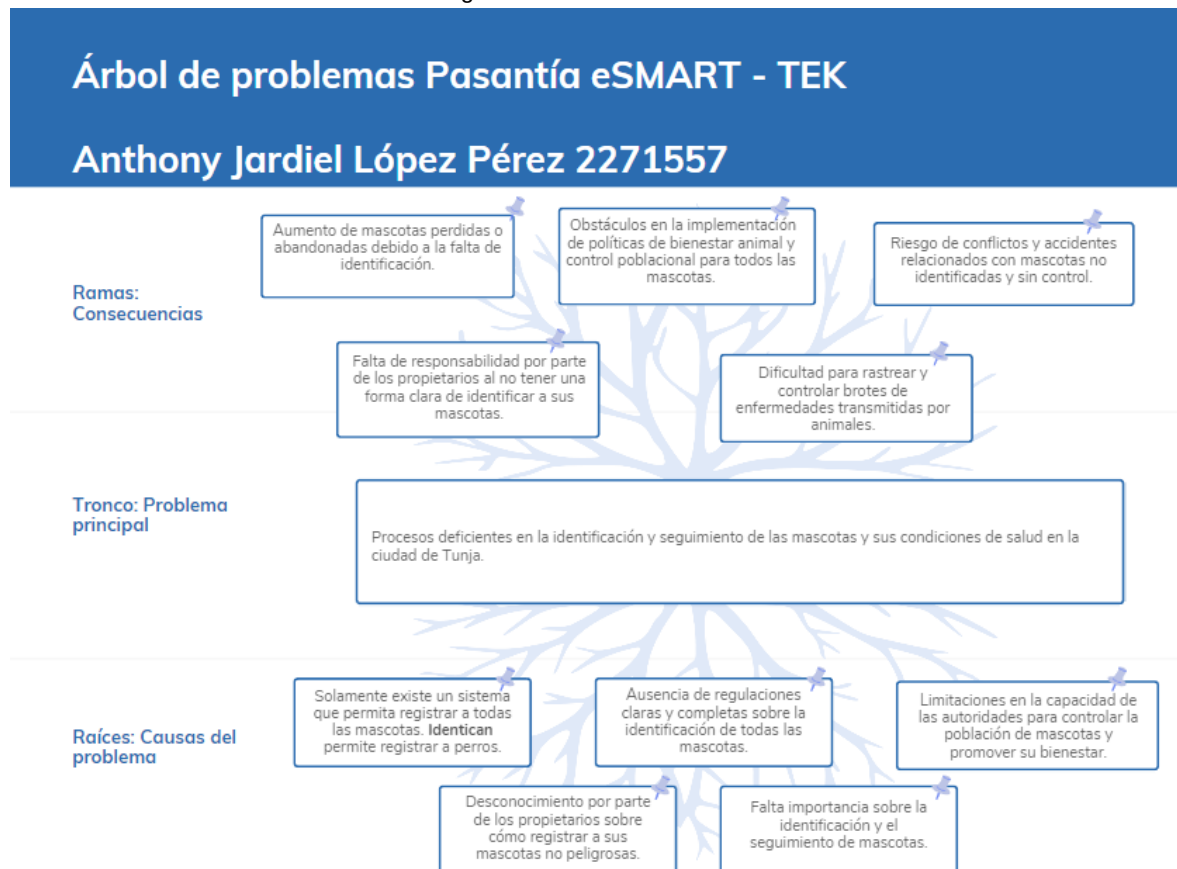
El crecimiento en la inversión en tecnología para mascotas se ve impulsado por la creciente tendencia del comercio electrónico, donde cada vez más personas realizan sus compras online, incluyendo productos y servicios para mascotas. Sin embargo, a pesar de la implementación de la cédula digital y la digitalización de la información en varios sectores desde diciembre de 2020, se ha observado que muchas empresas, especialmente las más pequeñas, pueden estar rezagadas en la adopción de tecnologías debido a limitaciones de recursos, especialmente exacerbadas por la pandemia de Covid-19 (Bloomberg, 2021).

Además, la falta de priorización de la transformación digital por parte de las juntas directivas también puede contribuir al retraso en la implementación de tecnologías en las empresas. Según un estudio citado en Bloomberglinea, solo el 44% de las empresas en Latinoamérica

consideran la transformación digital como una prioridad para sus juntas directivas (Bloomberg, 2021).

También es importante tener en cuenta que ya existen soluciones tecnológicas para el registro de mascotas utilizando códigos QR, como PetHub y Pet Widget (Pet Widget | LinkedIn, n.d.; PetHub | Your Pet Connected, n.d.), lo que podría influir en el diseño y desarrollo del sistema propuesto. Finalmente, la empresa eSmart-Tek, empresa que se dedica a la prestación de servicios y en los últimos años incursionando en el desarrollo de nuevos productos de software, se encuentra desarrollando Cuídame, una aplicación orientada al sector de la salud al priorizar la información de familiares y mascotas como un historial médico en caso de mascotas. La Figura 1 describe el problema principal, las causas y efectos que puede ocasionar el problema identificado.

Figura 1. Árbol de Problemas



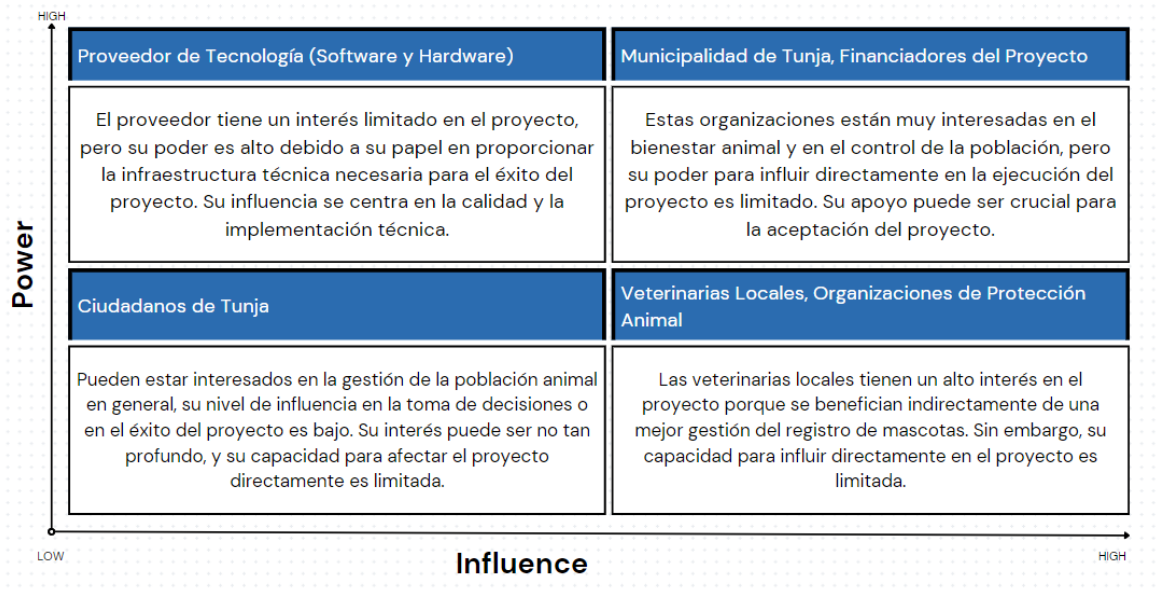
Fuente: Autor

Entre las causas de esta situación se encuentra la falta de regulaciones claras y completas sobre la identificación de todas las mascotas, lo cual dificulta la implementación de un sistema integral. Además, la limitación en la capacidad de las autoridades para controlar la población de mascotas y promover su bienestar contribuye a esta problemática. La falta de conciencia por parte de los propietarios sobre la importancia de registrar a sus mascotas no peligrosas también es un factor relevante.

Los efectos de esta situación son diversos y afectan tanto a los propietarios de mascotas como a la comunidad en general. Aumento de mascotas perdidas o abandonadas, obstáculos en la implementación de políticas de bienestar animal, riesgo de conflictos y accidentes relacionados con mascotas no identificadas, y dificultades para rastrear y controlar brotes de enfermedades transmitidas por animales son algunos de los efectos negativos derivados de la falta de un sistema integral de gestión de mascotas.

La matriz de stakeholders subraya la importancia de la colaboración y el respaldo de los actores clave para el éxito del proyecto. La participación de la municipalidad de Tunja y los financiadores del proyecto es crucial debido a su alto interés y poder, asegurando la implementación efectiva y la viabilidad financiera. El proveedor de tecnología, con su alto poder, pero bajo interés, requiere una gestión técnica rigurosa para asegurar una infraestructura adecuada. Por su parte, las veterinarias locales y las organizaciones de protección animal, aunque con alto interés y bajo poder, ofrecen un apoyo significativo que debe ser fomentado mediante alianzas estratégicas. Los ciudadanos de Tunja, con bajo interés y poder, son abordados a través de comunicación pública para mantenerlos informados y comprometidos.

Figura 2. Matriz de stakeholders



Fuente: Autor

3. JUSTIFICACIÓN

La ciudad de Tunja enfrenta una carencia evidente en su capacidad para gestionar la información relacionada con las mascotas, una problemática que impacta directamente a los propietarios de animales, profesionales veterinarios, las instituciones gubernamentales y, en última instancia, a la comunidad en general.

En un mundo en constante transformación tecnológica, la digitalización de la información se ha convertido en un imperativo. Diversas organizaciones y empresas están avanzando hacia la transformación digital, y la gestión de información de las mascotas no debería ser la excepción. Sin embargo, este proceso de digitalización trae consigo desafíos tecnológicos significativos, como la privacidad de los datos, la experiencia del usuario, y la interoperabilidad entre diferentes sistemas.

La creación de un sistema de registro y seguimiento de mascotas no solo está en sintonía con la tendencia de transformación digital, sino que también aborda estos desafíos mediante la implementación de medidas de seguridad robustas para proteger la información sensible de los usuarios. Además, se enfocó en desarrollar una interfaz amigable y accesible para los propietarios de mascotas y los profesionales veterinarios, asegurando una experiencia de usuario óptima.

La implementación de este sistema traerá consigo una mejora sustancial en la calidad de vida de las mascotas y sus propietarios. Facilitará el registro de información completa y actualizada sobre los animales, lo que a su vez garantizará su bienestar y salud. Además, agilizará el proceso de localización de mascotas extraviadas, lo que resulta de vital importancia en situaciones de emergencia.

No se debe pasar por alto la relación entre la tecnología para mascotas y el comercio electrónico, un sector en constante crecimiento. La adopción de este sistema podría fomentar el empleo y el desarrollo económico local al impulsar la adopción de tecnología en el ámbito de las mascotas.

En cuanto al trabajo, este proyecto de pasantía no solo se limitó a la implementación inicial del sistema, sino que también incluyó un análisis continuo de su impacto y la introducción de mejoras basadas en la retroalimentación de los usuarios. Además, se exploró nuevas funcionalidades y tecnologías emergentes para seguir innovando en la gestión de información de las mascotas.

4. OBJETIVOS

4.1. Objetivo General

Implementar un sistema informático para el registro y seguimiento de mascotas en la ciudad de Tunja, con el fin mantener un repositorio de información de la población animal en la ciudad, utilizando un marco de trabajo de código abierto para desarrollar aplicaciones híbridas.

4.2. Objetivos específicos

1. Construir la especificación de requerimientos de la aplicación, realizando un brainstorm con los stakeholders y leads de la empresa utilizando el estándar ISO/IEC/IEEE 42010.
2. Diseñar la aplicación especificada por medio de la representación de Kruchten 4 + 1, definiendo las interconexiones de las vistas y las conexiones del sistema.
3. Desarrollar una aplicación para que los propietarios puedan registrar información completa y actualizada sobre sus mascotas, incluyendo datos de identificación, salud y bienestar.
4. Realizar pruebas funcionales del sistema informático desarrollado utilizando listas de chequeo que permitan verificar el listado de requisitos funcionales de la aplicación.

5. ESTADO DEL ARTE

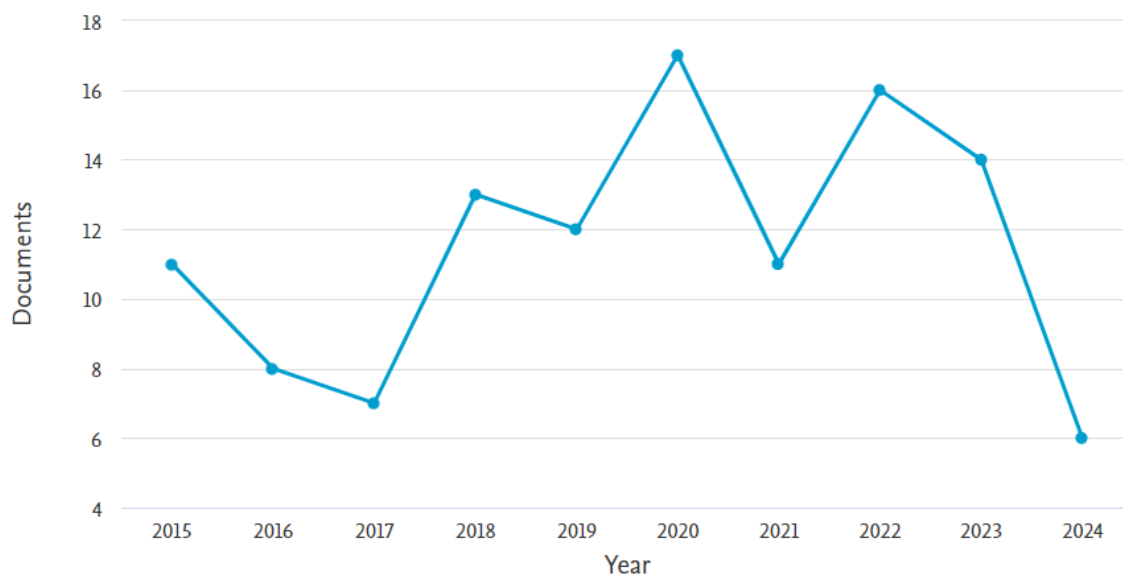
Para entender el contexto y las tendencias actuales en el desarrollo de sistemas de registro y seguimiento de mascotas, se realizó una investigación en diversas bases de datos académicas y científicas de la universidad, con un enfoque principal en Scopus. El objetivo de esta investigación fue identificar los avances recientes y las tecnologías emergentes que podrían aplicarse en el proyecto.

La búsqueda incluyó los términos: "Pet AND Registration AND System", restringiendo los resultados a publicaciones desde 2015 hasta 2024, en inglés, y dentro de las áreas de Ciencias de la Computación e Ingeniería aplicando limitaciones usando palabras clave.

Se identificaron 236 publicaciones relevantes que abordaban diversas facetas del desarrollo de sistemas de registro y seguimiento de mascotas, así como la implementación de aplicaciones híbridas. El análisis temporal de las publicaciones muestra un incremento significativo en el interés sobre este tema desde 2020, alcanzando su punto máximo en los últimos años.

Figura 3. Años de publicación de los artículos en relación con aplicaciones para mascotas

Documents by year



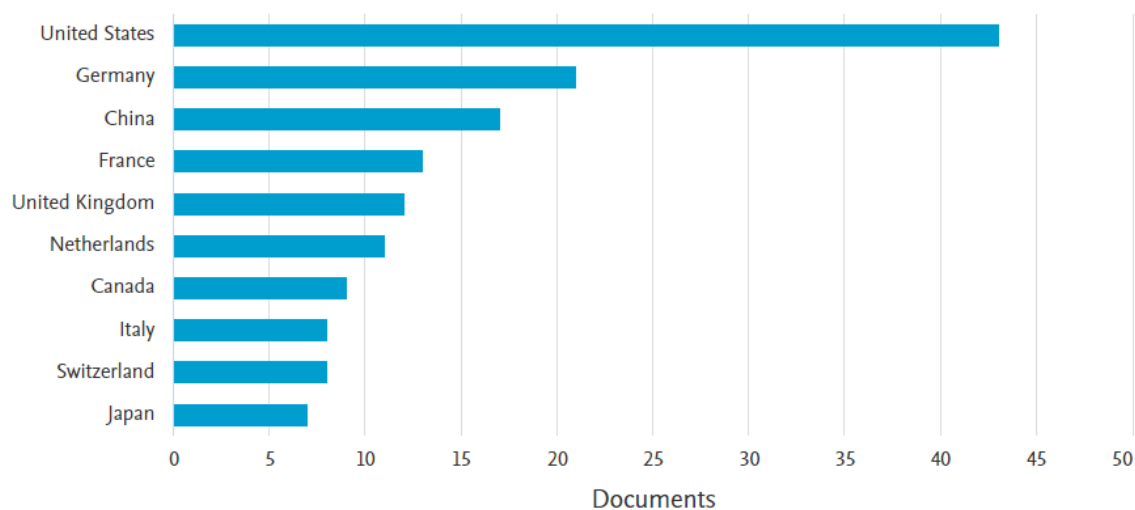
Fuente: Scopus

La distribución geográfica de la investigación refleja una fuerte concentración en países como Estados Unidos y Alemania seguido de China, lo que indica un alto nivel de actividad y desarrollo tecnológico en estas regiones en relación con el seguimiento y registro de animales.

Figura 4. Documentos por territorio en Scopus

Documents by country or territory

Compare the document counts for up to 15 countries/territories.

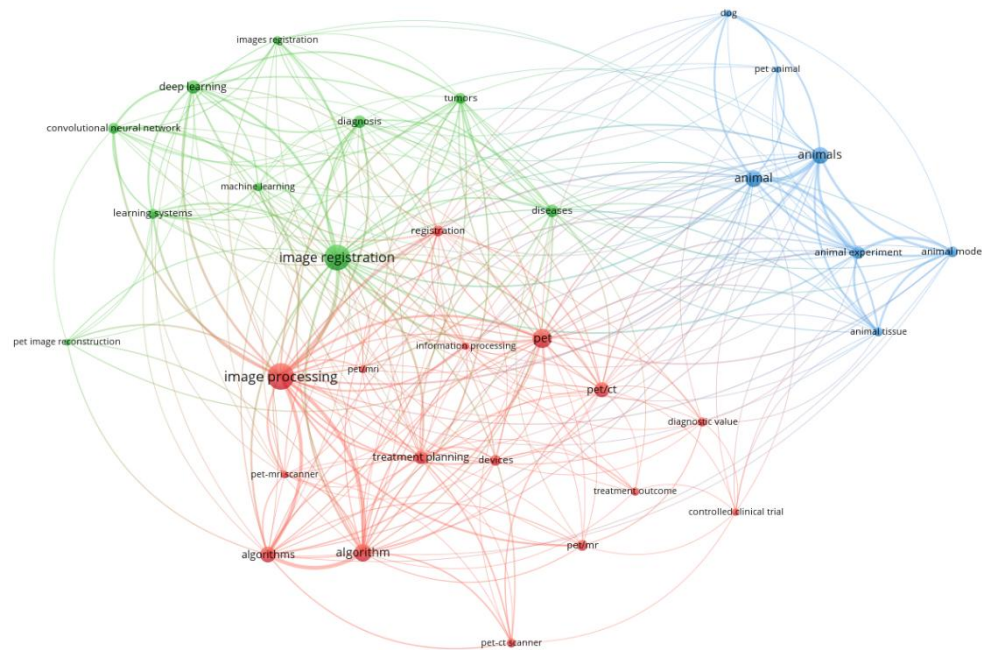


Fuente: Scopus

Para el análisis y visualización de los datos recopilados, se utilizó la herramienta VOSviewer, que permite la creación de mapas de co-ocurrencia de términos, redes de colaboración entre autores, y análisis de co-citación. Este software es particularmente útil para identificar patrones y tendencias en la literatura científica.

La Figura 3 muestra un mapa de co-ocurrencia de términos, destacando los conceptos clave relacionados con el desarrollo de aplicaciones híbridas y tecnologías de seguimiento. Entre los términos más recurrentes se encuentran "images", "pet" y "animals". Estos términos indican las áreas tecnológicas más exploradas en las publicaciones analizadas.

Figura 5. Visualización palabras clave VOSviewer



Fuente: VOSviewer Software

Asimismo, se identificaron varios clústers temáticos que agrupan las publicaciones según su enfoque. Por ejemplo, un clúster significativo se centra en el desarrollo de sistemas de registro y procesamiento de imágenes ("Image Registration" y "Image Processing"), que son cruciales para la identificación y seguimiento de animales. Otro clúster importante se relaciona con la tecnología aplicada específicamente al registro y seguimiento de animales ("Animals"), proporcionando un enfoque especializado en las necesidades y retos únicos de este campo.

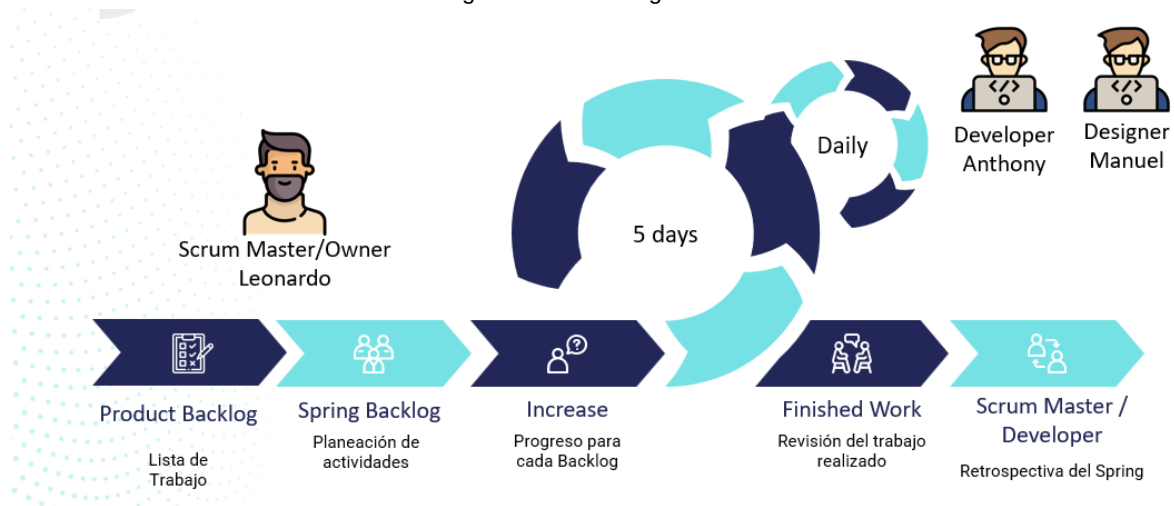
Estos hallazgos proporcionan una base sólida para el desarrollo del sistema propuesto, permitiendo la integración de tecnologías probadas y metodologías innovadoras. Además,

el análisis de la literatura existente asegura que el proyecto esté alineado con las mejores prácticas y las tendencias más recientes en el campo de los sistemas de registro y seguimiento de mascotas.

6. DESARROLLO METODOLÓGICO

Para la gestión del proyecto, se utilizó la metodología ágil SCRUM. Esta metodología es ampliamente utilizada en el desarrollo de software debido a su enfoque iterativo e incremental que permite la entrega continua de productos funcionales. SCRUM se basa en ciclos de trabajo denominados sprints, los cuales tienen una duración fija y se repiten hasta completar el proyecto.

Figura 6. Metodología Scrum



Fuente: Autor

Descripción: La imagen muestra gráficamente la metodología SCRUM.

6.1. Roles y Responsabilidades

En el equipo de trabajo, se definieron los siguientes roles:

Product Owner: El director de la empresa asumió esta responsabilidad.

Scrum Master: El director de la empresa asumió esta responsabilidad.

Equipo de Desarrollo: El equipo de desarrollo está conformado por el director de la empresa, un diseñador UX/UI y el desarrollador como pasante de la empresa.

6.2. Artefactos de SCRUM

Los artefactos principales utilizados fueron:

Product Backlog: Lista priorizada de todas las funcionalidades y requerimientos del producto.

Sprint Backlog: Conjunto de tareas seleccionadas del Product Backlog para ser completadas en un sprint.

Incremento: Resultado del trabajo completado durante el sprint, el cual debe ser un producto potencialmente entregable.

Finished Work: Tareas o incrementos completados asegurando que están listos para ser entregados o integrados en el producto final.

Retrospectiva: Reunión al final de cada sprint donde el equipo reflexiona sobre el proceso, identifica lo que funcionó bien, lo que no funcionó, y propone mejoras para futuros sprints.

6.3. Tablero Kanban en Trello

Para la visualización y gestión de las tareas, se implementó un tablero de Kanban en Trello. Esta herramienta facilita la organización y seguimiento del trabajo en curso mediante el uso de tarjetas y listas.

6.3.1.1 Configuración del Tablero de Kanban

El tablero de Kanban en Trello se configuró con las siguientes listas:

Backlog: Contiene todas las tareas pendientes de ser trabajadas.

To Do: Tareas que han sido priorizadas para el sprint actual.

Doing: Tareas que están actualmente en desarrollo.

Code Review: Tareas que están actualmente en revisión de código.

Testing: Tareas que están actualmente en pruebas.

Done: Tareas que han sido completadas.

Cada lista tiene un propósito específico en el flujo de trabajo del equipo:

Backlog: Aquí se recopilan todas las tareas pendientes que aún no han sido priorizadas para su desarrollo.

To Do: En esta lista se encuentran las tareas que se han seleccionado y priorizado para el sprint actual. Estas tareas están listas para ser trabajadas.

Doing: Esta lista contiene las tareas que el equipo ha comenzado a desarrollar.

Code Review: Las tareas en esta lista están en la fase de revisión de código, asegurando que el código cumpla con los estándares de calidad antes de integrarlo al producto final.

Testing: Las tareas en esta lista están siendo probadas para garantizar su correcto funcionamiento y cumplir con los criterios de aceptación.

Done: Las tareas que han sido completadas y aprobadas se colocan en esta lista.

Cada tarjeta en Trello representaba una tarea específica y contenía detalles como la descripción, criterios de aceptación, responsables, y fechas de entrega. Además, se utilizaron etiquetas para clasificar las tareas según su prioridad y tipo de trabajo.

6.3.1.2 Ventajas del Uso de Kanban

El uso de un tablero de Kanban en Trello proporcionó varios beneficios:

Visualización clara del progreso: Permite ver en qué estado se encuentra cada tarea y el flujo de trabajo del equipo.

Flexibilidad: Facilita la adaptación a cambios y la reorganización de prioridades.

Colaboración: Mejora la comunicación y coordinación entre los miembros del equipo.

Integración con SCRUM

El tablero de Kanban en Trello se integró con la metodología SCRUM de la siguiente manera:

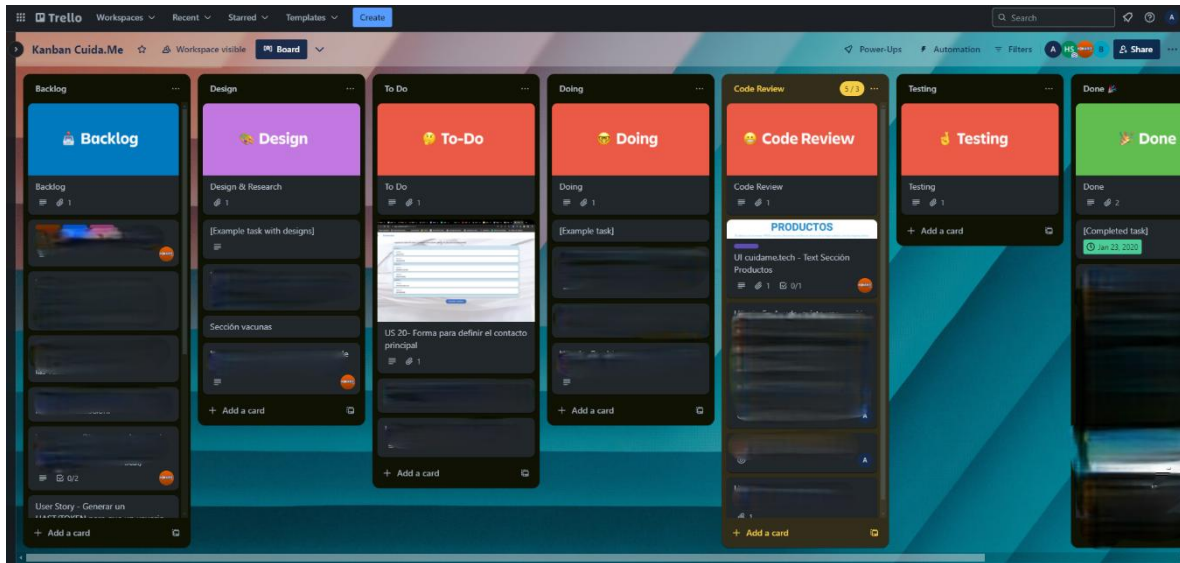
Sprint Planning: Durante la planificación del sprint, se seleccionaban las tareas del Product Backlog y se movían a la lista de To Do.

Daily Standups: En las reuniones diarias, se revisaba el tablero para actualizar el estado de las tareas y discutir bloqueos.

Sprint Review y Retrospective: Al final de cada sprint, se revisaban las tareas completadas en la lista de Done y se discutían las mejoras para el próximo sprint.

Esta combinación de SCRUM y Kanban en Trello permitió una gestión eficiente del proyecto, asegurando la entrega continua de valor y la mejora constante del proceso.

Figura 7. Tablero Kanban en Trello



Fuente: Autor

Descripción: La imagen muestra el tablero utilizado para el desarrollo de las historias de usuario.

Figura 8. Tablero Kanban físico en la oficina



Fuente: Autor

Descripción: La imagen muestra el tablero utilizado para el desarrollo de las historias de usuario de manera física.

7. DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS

7.1. Actividades realizadas en el proyecto

La siguiente sección tiene como propósito explicar el proceso de desarrollo de la aplicación, especificando las actividades realizadas y las herramientas que se utilizaron a la hora de realizar las actividades y/o desarrollo de la aplicación.

Descripción de la actividad	Fecha inicio	Fecha entrega	Producto entregado
Diseño vista lógica	01-08-2023	08-08-2023	Modelo ER para la base de datos y modelo de clases, arquitectura ISO/IEC/IEEE 42010.
Diseño vista de desarrollo	09-08-2023	16-08-2023	Diagrama de desarrollo.
Diseño de escenarios	17-08-2023	24-08-2023	Diagramas caso de uso.
Diseño de procesos	25-08-2023	01-09-2023	Diagramas de procesos para el flujo de la aplicación.
Vista de Físico	04-09-2023	09-09-2023	Diagrama de la arquitectura física.
Diseño especificación de requerimientos	11-09-2023	22-09-2023	Especificación de Requerimientos de Software IEEE 830, Estructura de Desglose de Trabajo.
Implementación en la base de datos existente	25-09-2023	29-09-2023	Implementación a la base de datos Postgresql.
Creación backend (modelos, controladores, rutas de la API, rutas estáticas)	02-09-2023	03-11-2023	Commits backend en el repositorio de GitHub empresarial.
Creación componentes gráficos en el frontend	06-11-2023	24-11-2023	Commits Frontend en el repositorio de GitHub empresarial.
Implementación del frontend con la API	27-11-2023	08-12-2023	Commits en ambos repositorios e informes al líder del proyecto.
Despliegue del frontend y backend en GCP	11-12-2023	22-12-2023	Aplicativo actualizado para realizar pruebas.
Pruebas funcionales en la aplicación	25-12-2023	05-01-2024	
Despliegue en Google Play Store	08-01-2024	12-01-2024	Aplicativo actualizado en Play Store.
Despliegue en Apple Store	08-01-2024	17-01-2024	Aplicativo actualizado en App Store.

7.1.1. Detalles del desarrollo de actividades

7.1.1.1 Diseño arquitectura de software

El diseño de la arquitectura de software fue un proceso fundamental en el proyecto de desarrollo, ya que sentó las bases para la construcción de un sistema, escalable y eficiente. Antes de proceder al diseño, se llevó a cabo un proceso de especificación de requerimientos.

Este proceso se desarrolló a través de lluvia de ideas, en las que participó el jefe de la empresa y el desarrollador. Estas actividades se realizaron de manera presencial, lo que permitió recopilar información para el desarrollo del proyecto evidenciando el documento de requerimientos con la norma IEEE 830 en el apartado de anexo 11.2 y la implementación del diagrama utilizando la norma ISO/IEC/IEEE 42010. Ver anexo 11.2

Como resultado, se elaboró un documento de especificación de requerimientos que detallaba las funcionalidades del sistema.

El diseño de la arquitectura de software fue un proceso fundamental en el proyecto de desarrollo, ya que sentó las bases para la construcción de un sistema robusto, escalable y eficiente. En el caso de esta empresa, se tomaron decisiones clave en cada etapa del proceso, desde el diseño hasta el despliegue de la aplicación.

En primer lugar, fue crucial mencionar el modelo de negocio que sirvió como guía para el diseño de la arquitectura. Este modelo proporcionó una comprensión de las necesidades del negocio y de los requisitos funcionales y no funcionales del sistema a desarrollar. Esto permitió tomar decisiones informadas en cada paso del proceso evidenciando la arquitectura en el apartado de anexo diagramas 11.4.

En cuanto a las tecnologías empleadas, la elección de Node.js para el desarrollo del backend fue estratégica. Node.js es conocido por su eficiencia en el manejo de tareas concurrentes. Además, al utilizar JavaScript, se facilitó la comunicación entre los diferentes componentes del sistema y se redujo la sobrecarga cognitiva para los desarrolladores.

En lo que respecta a la gestión de datos, PostgreSQL se destacó como una elección sólida. Esta base de datos relacional es conocida por su robustez, rendimiento y capacidad para manejar grandes volúmenes de datos. Además, su amplia documentación y comunidad activa brindaron un sólido soporte para resolver cualquier problema que surgió durante el desarrollo y la implementación. Una parte de la base de datos se encuentra evidenciada en la figura Figura 9. Implementación base de datos local.

En el ámbito del frontend, la combinación de Angular y Ionic ofreció una solución completa para la creación de interfaces de usuario modernas y receptivas. Angular, un framework desarrollado por Google, proporcionó una estructura sólida para construir aplicaciones web dinámicas y escalables. Por otro lado, Ionic es un framework de código abierto que permitió el desarrollo de aplicaciones móviles utilizando tecnologías web estándar como HTML, CSS. Esta combinación permitió crear una experiencia de usuario coherente y de alta calidad tanto en aplicaciones web como móviles.

El desglose del trabajo (Work Breakdown Structure, WBS) proporcionó una visión detallada y organizada de las actividades necesarias para el desarrollo del sistema de gestión de mascotas. Esta estructura jerárquica facilitó la planificación y ejecución del proyecto. Comenzó con el diseño del sistema, seguido por la implementación y el despliegue de la aplicación.

Con el fin de definir la apariencia visual del producto en desarrollo, se crearon las interfaces gráficas para la aplicación web desde cero, utilizando la herramienta Adobe XD.

7.1.1.2 Desarrollo de la aplicación

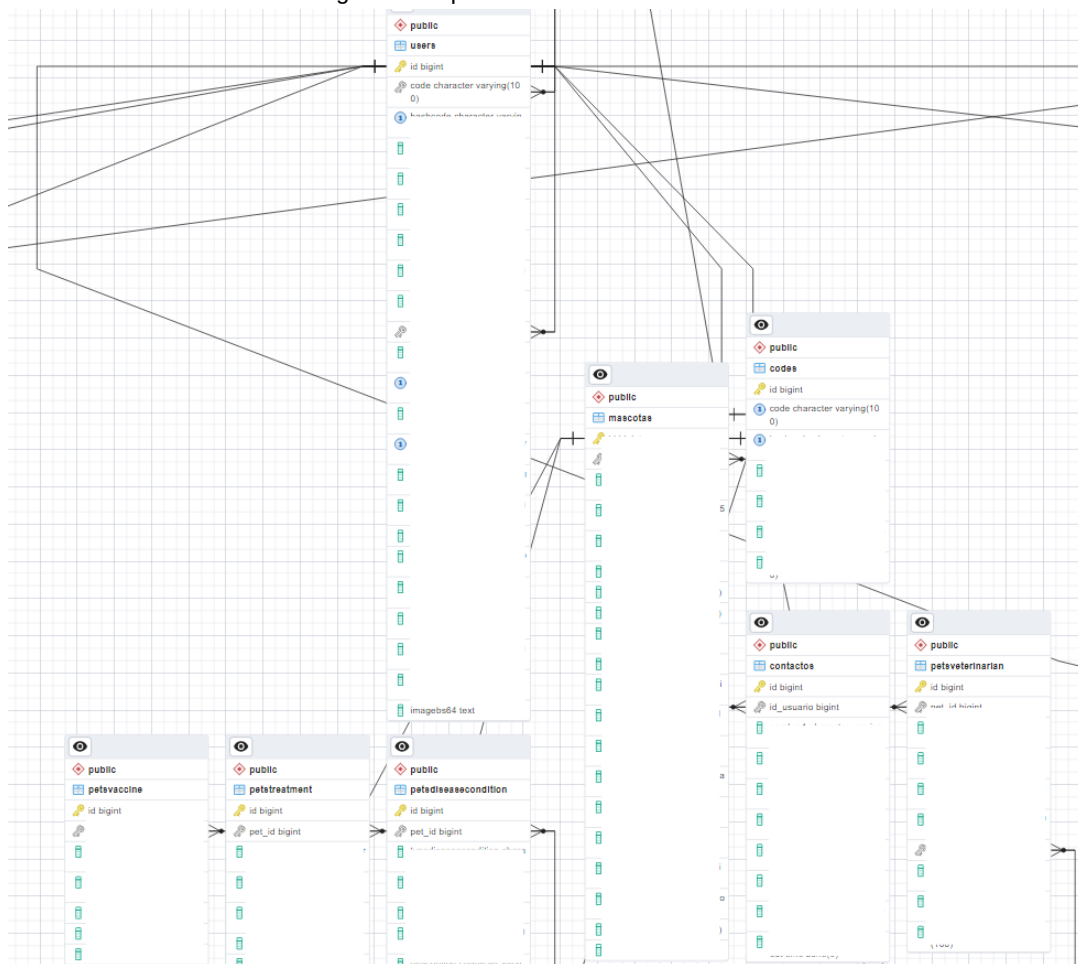
El desarrollo de una aplicación siguió un proceso organizado utilizando la metodología SCRUM para garantizar su eficacia y coherencia. El proyecto se completó en un total de 16 sprints, con revisiones semanales y ocasionales daily standups para revisar el progreso de los sprints.

Sprints 1-2: Implementación en la Base de Datos

El proceso inició con la implementación en la base de datos existente. Durante estos sprints, se evaluó la estructura de datos actual y se determinó cómo integrar la nueva funcionalidad sin afectar la integridad de los datos previos. Esto implicó la creación de nuevas tablas, campos o modificaciones necesarias.

La correcta implementación en la base de datos de desarrollo garantiza el funcionamiento en la base de datos de producción.

Figura 9. Implementación base de datos local



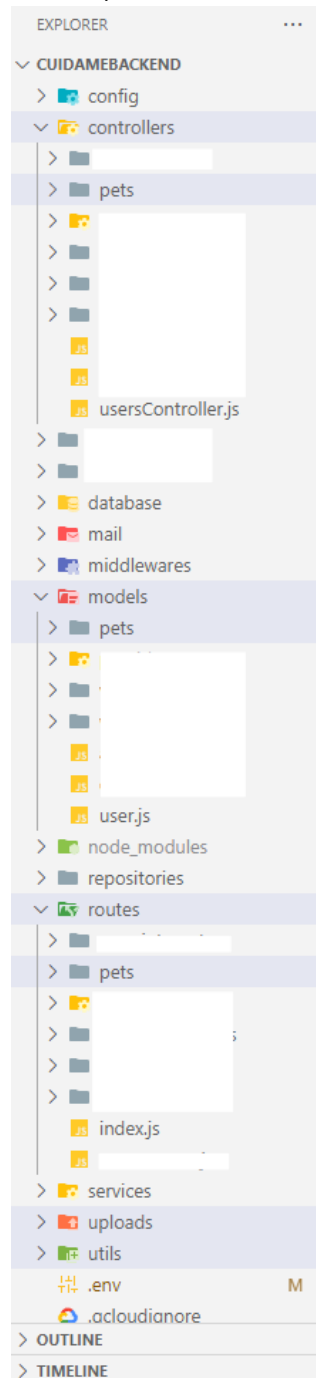
Fuente: Autor

Descripción: La imagen muestra como evidencia la implementación de las nuevas tablas en la base de datos local y en la base de datos en la nube. Debido a restricciones de confidencialidad solamente se muestra parte de la base de datos.

Sprints 3-5: Creación del Backend

Una vez establecida la base de datos, se procedió a la creación del backend en los siguientes sprints. Aquí se definieron los modelos de datos que representaban la lógica de negocio de la aplicación y los controladores que gestionaban las solicitudes del usuario y manipulaban los datos según fuera necesario, autenticación e identificación de usuario JWT. Además, se establecieron las rutas de la API que permitirían la comunicación entre el frontend y el backend, definiendo cómo se accedería a los recursos de la aplicación utilizando la herramienta Postman.

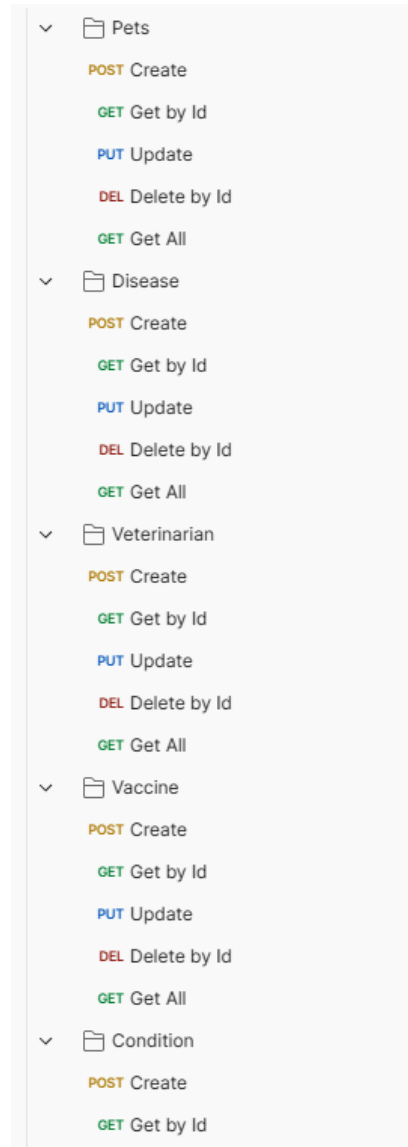
Figura 10. Implementación del backend



Fuente: Autor

Descripción: La imagen muestra la implementación de la estructura modelo, vista, controlador en el backend mostrando los elementos más destacados debido a restricciones de confidencialidad.

Figura 11. Pruebas unitarias en Postman.



Fuente: Autor

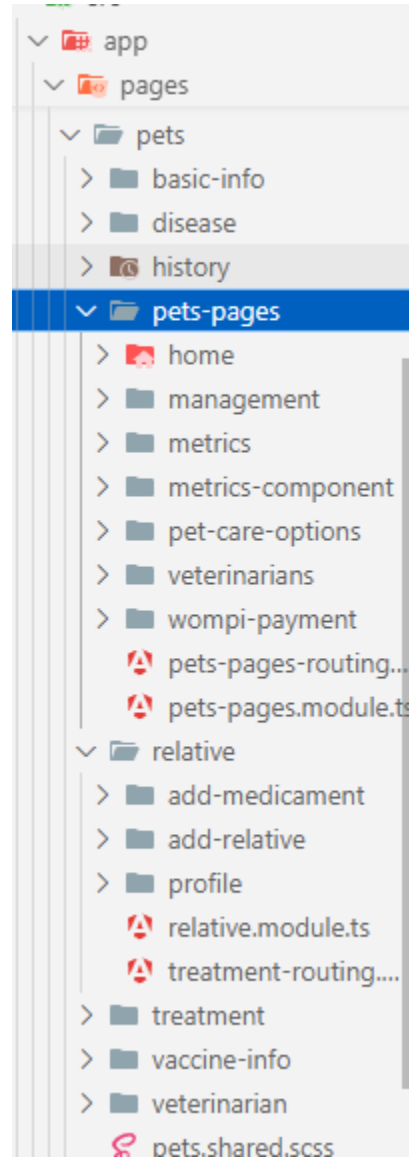
Descripción: La imagen muestra las carpetas para realizar pruebas en la herramienta Postman como evidencia.

Sprints 6-9: Desarrollo del Frontend

Con el backend en su lugar, se pasó a la creación de los componentes gráficos en el frontend durante estos sprints. Esto implicó el diseño y desarrollo de la interfaz de usuario, incluyendo la disposición de elementos, estilos visuales y comportamientos interactivos.

Fue crucial que estos componentes fueran intuitivos y estuvieran diseñados de manera coherente con la experiencia de usuario deseada.

Figura 12. Implementación del frontend, estructura de carpetas desarrolladas.



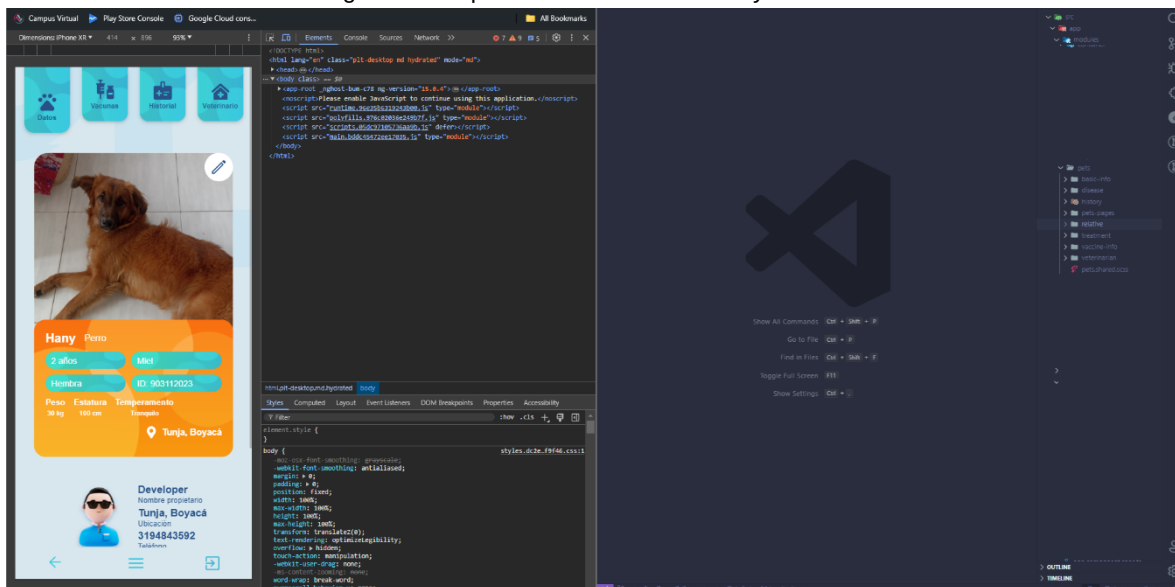
Fuente: Autor

Descripción: La imagen muestra la implementación de la estructura de carpetas implementadas en el frontend del proyecto.

Sprints 10-12: Implementación del Frontend con la API

Una vez que los componentes gráficos estuvieron listos, se procedió a la implementación del frontend con la API en los sprints siguientes. En esta etapa, se integró la lógica del frontend con el backend, haciendo uso de las rutas de la API previamente definidas para acceder y manipular los datos necesarios. Se realizaron pruebas para garantizar que la comunicación entre el frontend y el backend funcionara correctamente y que la aplicación se comportara según lo esperado usando listas de chequeo en Excel.

Figura 13. Implementación del frontend y backend.



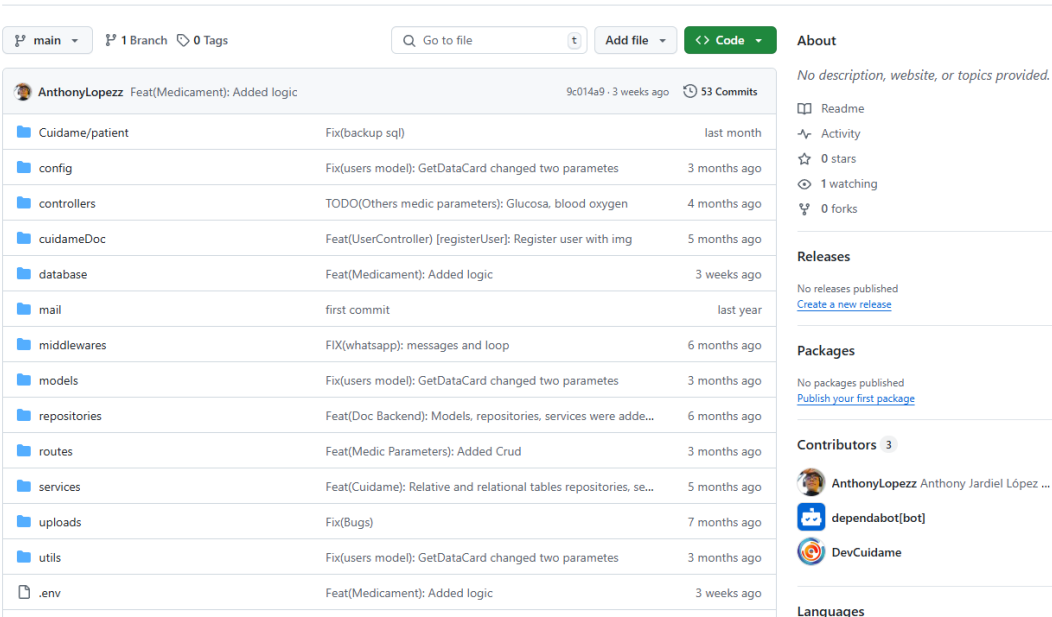
Fuente: Autor

Descripción: La imagen muestra evidencia del desarrollo de la aplicación al consumir los servicios del backend y mostrando información.

Sprints 13-15: Versionamiento y Despliegue

El versionamiento de la aplicación se gestionó subiendo cada desarrollo completo al repositorio de GitHub al final de cada sprint. Una vez aprobado, se cargaba al repositorio. Este proceso se siguió sistemáticamente para asegurar la integridad y rastreabilidad del código.

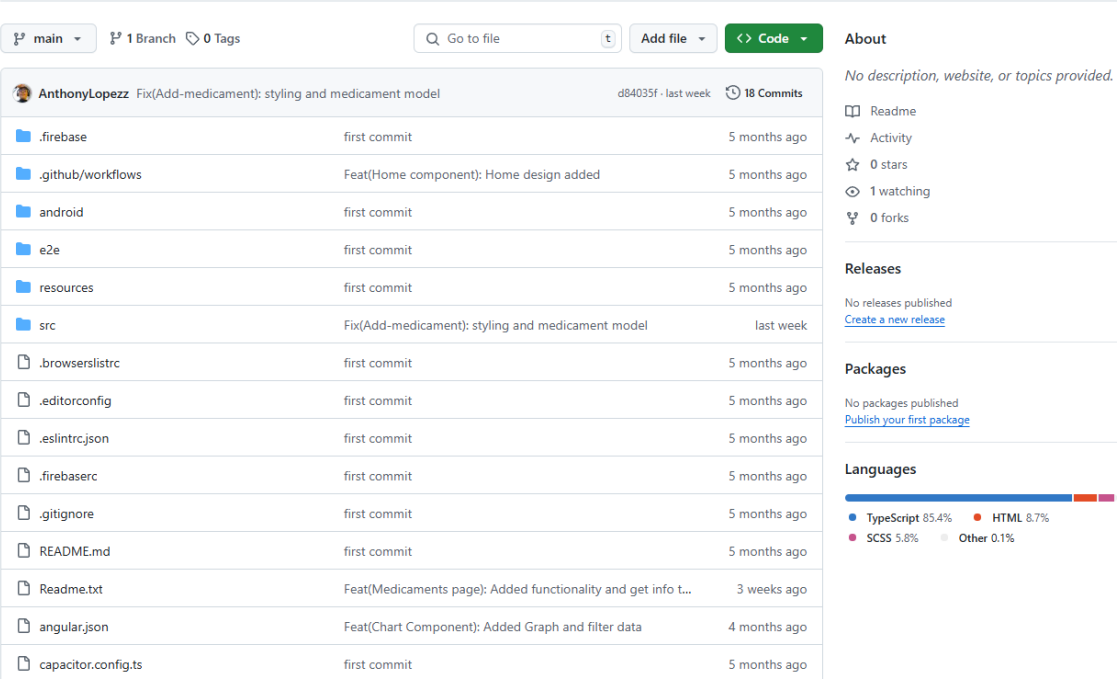
Figura 14. Versionamiento y despliegue del backend.



Fuente: Autor

Descripción: La imagen muestra evidencia del repositorio y manejo de versionamiento en github.

Figura 15. Versionamiento y despliegue del frontend.



Fuente: Autor

Descripción: La imagen muestra evidencia del repositorio y manejo de versionamiento en github.

Sprint 16: Refinamiento y Optimización

Finalmente, se llevó a cabo un proceso de refinamiento y optimización durante el último sprint. Aquí se identificaron y corrigieron posibles errores, se realizaron ajustes de rendimiento y se mejoró la experiencia de usuario en función de los comentarios recibidos. Este proceso implicó iteraciones hasta alcanzar un producto final satisfactorio y listo para su despliegue. Además, se utilizó una lista de chequeo en Excel para documentar las pruebas realizadas en la aplicación en los diferentes casos de uso documentados previamente.

7.1.1.3 Despliegue de la aplicación

El siguiente paso fue desplegar tanto el frontend como el backend en Google Cloud Platform (GCP). Esto implicó configurar los servidores y servicios necesarios en GCP, como Compute Engine para el backend y el frontend, asegurando que la aplicación estuviera accesible y funcionara correctamente en la nube.

Una vez desplegada en GCP, se llevaron a cabo pruebas funcionales en la aplicación para asegurar su calidad y fiabilidad. Estas pruebas unitarias e integrales abarcaron diferentes escenarios de uso, desde la navegación básica hasta la interacción con características específicas, identificando y corrigiendo cualquier error o comportamiento inesperado que se presentara.

Con las pruebas funcionales completadas satisfactoriamente, la siguiente etapa fue desplegar la aplicación en la Google Play Store para dispositivos Android. Esto implicó preparar y enviar la aplicación a través del Google Play Console, siguiendo las pautas y requisitos establecidos por Google para garantizar su aprobación y disponibilidad en la tienda.

Simultáneamente, se procedió al despliegue en la Apple App Store para dispositivos iOS. Esto implicó preparar y enviar la aplicación a través del Apple Developer Console, cumpliendo con los estándares y directrices de Apple para su revisión y aprobación, asegurando así su disponibilidad para los usuarios de dispositivos Apple.

Una vez que la aplicación estuvo disponible tanto en la Google Play Store como en la Apple App Store, se llevó a cabo una última verificación para confirmar que la implementación y la distribución se realizaron correctamente en ambas plataformas. Esto incluyó la descarga y la ejecución de la aplicación desde las respectivas tiendas para asegurarse de que los usuarios pudieran acceder y utilizar la aplicación sin problemas.

7.2. Herramientas utilizadas en el desarrollo del proyecto

Herramienta utilizada	Para qué la utilizó
Angular	Se utilizó junto con TypeScript para desarrollar la interfaz de usuario de la aplicación, aprovechando su capacidad de crear aplicaciones web dinámicas y escalables.
Ionic	Se empleó para crear una aplicación móvil híbrida, compatible con múltiples plataformas, como iOS y Android.
Node.js	Se utilizó como entorno de ejecución para el servidor de la aplicación, aprovechando su capacidad de ejecutar JavaScript en el servidor y gestionar la lógica de la aplicación.
Trello	Se utilizó como una herramienta de gestión de proyectos para organizar tareas, asignar responsabilidades y realizar un seguimiento del progreso.
Draw.io	Se empleó para diseñar y crear diagramas que representan la arquitectura y el flujo de la aplicación.
Google Cloud Platform	Se utilizó como plataforma de computación en la nube para alojar la aplicación y gestionar recursos como bases de datos y servicios de almacenamiento.
Linux Server	Se utilizó como sistema operativo para el servidor de la aplicación, aprovechando su estabilidad y seguridad.
Apache	Se empleó como servidor web para servir páginas web estáticas y dinámicas, así como para gestionar la comunicación entre el cliente y el servidor.
Pm2	Se utilizó como un administrador de procesos para asegurar la disponibilidad y el rendimiento del servidor Node.js.
Adobe xD	Se empleó para diseñar y crear prototipos de la interfaz de usuario de la aplicación.
Android Studio	Se utilizó para compilar la versión de la aplicación destinada a dispositivos Android.
Xcode	Se empleó para compilar la versión de la aplicación destinada a dispositivos iOS.
Google Play Console	Se utilizó para publicar y gestionar la versión de la aplicación para dispositivos Android en Google Play Store.
Apple Developer	Se empleó para publicar y gestionar la versión de la aplicación para dispositivos iOS en App Store.
GitHub	Se empleó para la gestión de versiones y repositorio del código
Google Meet	Se empleó para tener las reuniones con el director empresarial.

8. RESULTADOS

8.1. Cumplimiento de los Objetivos Específicos

8.1.1. Especificación de Requerimientos

La construcción de la especificación de requerimientos de la aplicación se llevó a cabo mediante sesiones de brainstorming con los stakeholders y líderes de la empresa, utilizando el estándar ISO/IEC/IEEE 42010 ver anexo 11.2. Este proceso permitió identificar y documentar de manera detallada las necesidades y expectativas de los usuarios, así como los requisitos técnicos necesarios para el desarrollo del sistema complementándolo con la norma IEEE 830 para la especificación de requisitos funcionales y no funcionales ver anexo 11.3. La especificación de requerimientos incluyó la identificación de funcionalidades clave, restricciones técnicas y operativas, y criterios de aceptación, lo que proporcionó una base sólida para el diseño y desarrollo del sistema.

8.1.2. Diseño de la Aplicación

El diseño de la aplicación se realizó siguiendo la metodología 4+1 de Kruchten (Kruchten, 1995), lo que permitió definir claramente las interconexiones de las vistas y las conexiones del sistema. Esta metodología se basa en cinco vistas principales que abordan diferentes aspectos de la arquitectura del sistema:

Vista Lógica: Se desarrolló utilizando diagramas de clases UML para representar la estructura del sistema, identificando las principales entidades, sus relaciones y las operaciones clave. Este diagrama fue fundamental para garantizar la correcta organización del modelo de datos y la interacción entre los componentes lógicos. Ver anexos 11.4.1 y 11.4.2.

Vista de Proceso: Se utilizó un diagrama de actividades para modelar el flujo de trabajo y las interacciones entre los procesos del sistema. Este diagrama facilitó la identificación de puntos críticos en la ejecución concurrente y en la gestión de recursos. Ver anexo 11.4.6.

Vista de Desarrollo: Representada mediante un diagrama de componentes, esta vista describió la organización del código fuente, los módulos principales y cómo estos se relacionan entre sí. Esto permitió estructurar el sistema de manera escalable y mantenible. Ver anexo 11.4.4.

Vista Física: Se utilizó un diagrama de despliegue para mostrar la distribución del sistema en el entorno físico, incluyendo servidores, bases de datos. Este diagrama ayudó a definir la infraestructura necesaria para el despliegue del sistema. Ver anexo 11.4.5.

Vista de Casos de Uso: Complementando las otras vistas, se emplearon diagramas de casos de uso para identificar los requisitos funcionales clave desde la perspectiva del usuario final, alineándolos con los objetivos del sistema. Ver anexos 11.4.3.

8.1.3. Desarrollo de la Aplicación

Se desarrolló una aplicación híbrida utilizando un marco de trabajo de código abierto, permitiendo a los propietarios de mascotas registrar información completa y actualizada sobre sus animales, incluyendo datos de identificación, salud y bienestar. La aplicación ofrece una interfaz de usuario amigable y accesible, diseñada para facilitar la entrada y gestión de datos por parte de los usuarios. Además, se implementaron funcionalidades clave como la geolocalización de mascotas.

8.1.4. Pruebas Funcionales

Las pruebas funcionales del sistema se realizaron utilizando listas de chequeo que permitieron verificar el cumplimiento de los requisitos funcionales de la aplicación. Este proceso incluyó la ejecución de casos de prueba que cubrieron todas las funcionalidades especificadas, identificando y corrigiendo errores antes del despliegue. Las pruebas funcionales aseguraron que todas las características de la aplicación operaran según lo previsto, proporcionando una experiencia de usuario confiable y sin fallos.

8.2. Despliegue y Implementación

8.2.1. Despliegue del Backend

El backend del sistema se desplegó exitosamente en Google Cloud Platform (GCP), lo que garantiza una infraestructura escalable y segura para el almacenamiento y procesamiento de datos. La elección de GCP permitió aprovechar sus capacidades de alta disponibilidad y recuperación ante desastres, asegurando que el sistema permanezca operativo y accesible en todo momento.

8.2.2. Implementación de la Base de Datos

La implementación de la base de datos fue exitosa, permitiendo un almacenamiento eficiente y seguro de toda la información relacionada con las mascotas registradas en el sistema. Se utilizó un sistema de gestión de bases de datos relacional para garantizar la integridad y consistencia de los datos, además de implementar medidas de seguridad para proteger la información sensible de los usuarios.

8.2.3. Despliegue del Frontend

El frontend de la aplicación se desplegó correctamente, proporcionando una interfaz de usuario accesible y funcional que facilita la interacción de los usuarios con el sistema. Se utilizó una arquitectura de desarrollo web progresiva para asegurar que la aplicación sea rápida, responda adecuadamente y funcione en una amplia variedad de dispositivos y plataformas.

Figura 16. Aplicación en Servidores en Google Cloud Platform.

☐		cu	(nic0)	(nic0)	SSH ▾
☐		in	(nic0)	(nic0)	SSH ▾

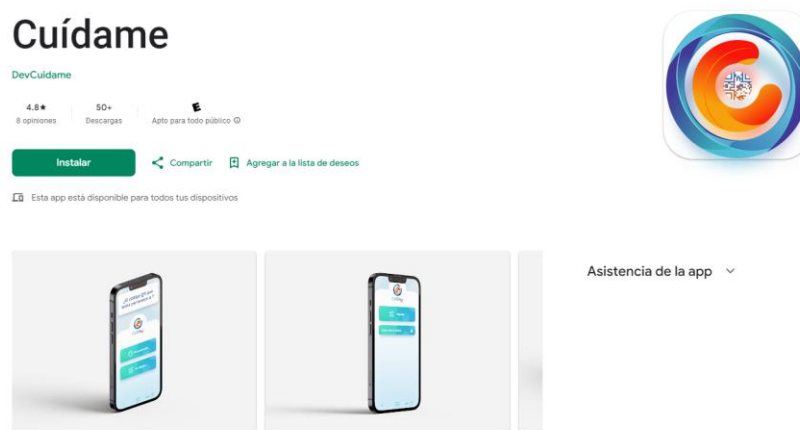
Fuente: Autor

Descripción: La imagen muestra los dos servidores de la aplicación activos y funcionando, ocultando información sensible.

8.2.4. Publicación en Google Play Store y App Store

La aplicación se publicó en Google Play Store y App Store, permitiendo a los usuarios descargarla e instalarla en sus dispositivos móviles. Este paso garantizó que la aplicación estuviera disponible para una amplia audiencia y pudiera ser utilizada de manera efectiva. El proceso de publicación incluyó la realización de pruebas adicionales de compatibilidad y el cumplimiento de las políticas y directrices de ambas tiendas de aplicaciones.

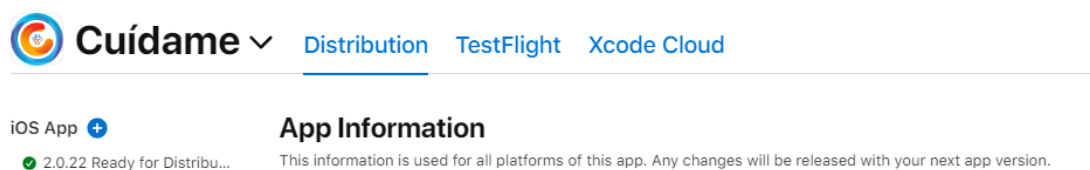
Figura 17. Aplicación en Play Store



Fuente: Autor

Descripción: La imagen muestra la aplicación en la Play Store.

Figura 18. Aplicación en App Store



Fuente: Autor

Descripción: La imagen muestra la aplicación en la App Store desde Apple Developer.

8.3. Impacto y Beneficios del Sistema

8.3.1. Gestión de Información de Mascotas

El sistema permite la gestión de la información de las mascotas en la ciudad de Tunja, proporcionando un repositorio centralizado y accesible de datos. Esto facilita la labor de propietarios, veterinarios y autoridades, permitiendo un acceso rápido y eficiente a información crucial sobre la salud y el bienestar de las mascotas.

8.3.2. Localización de Mascotas Extraviadas

La aplicación permite el proceso de localización de mascotas extraviadas mediante herramientas de geolocalización y notificaciones. Los propietarios pueden recibir notificaciones cuando el código QR de la mascota se escanea. La aplicación alerta a los contactos de emergencia además del dueño de la mascota, aumentando las posibilidades de recuperación rápida y segura de los animales.

8.3.3. Bienestar y Salud de las Mascotas

El sistema contribuye al bienestar y salud de las mascotas al permitir un registro detallado y actualizado de su información de salud. Los propietarios pueden registrar vacunaciones y tratamientos, lo que facilita el seguimiento y la atención médica adecuada. Además, la disponibilidad de esta información para los veterinarios mejora la calidad de la atención proporcionada.

8.4. Aporte de Ingeniería

El aporte ingenieril en el desarrollo del proyecto para la pasantía se organiza en los siguientes apartados:

8.4.1. Organización y estructura del código

- **Modularidad:** el backend está dividido en archivos reutilizables y especializados. Cada archivo debe cumplir una única responsabilidad.

- **Nombres significativos:** el backend utiliza nombres descriptivos para variables, funciones, clases y archivos. Esto mejora la legibilidad y facilita el mantenimiento.
- **Estructura de carpetas:** El backend se encuentra organizado de manera lógica para facilitar la navegación.
- **Comentarios claros:** Solo se comenta el código cuando sea estrictamente necesario; el código bien escrito debe de ser autodescriptivo.

8.4.2. Principios de Programación

- **KISS (Keep It Simple, Stupid):** El código en el backend es fácil de entender ya que en su mayoría comprende peticiones CRUD, carga de imágenes, conversión a base64 para imágenes y archivos PDFs para evitar la sobreingeniería.
- **YAGNI (You Aren't Gonna Neet It) :** En el Código solamente se implementa lo necesario al evitar desarrollar funciones que no sean requeridas aún.

8.4.3. Pruebas y Calidad de Software

- **Pruebas unitarias:** Se aseguró de cubrir funciones críticas con pruebas no automatizadas.
- **Pruebas de integración:** Se validó que los módulos puedan interactuar correctamente al tener un buen funcionamiento entre ellos.
- **Linter y formateadores:** Se usaron herramientas como ESLint, Prettier para mantener un estándar de calidad y organización del código.
- **Code Review:** Se realizó la implementación de revisiones de código no colaborativas para detectar errores y mejorar la calidad.
- **Patrones de diseño:** Se usó el patrón Repository según las necesidades del proyecto.

8.4.4. Gestión de control de Versiones

- **Git Hub:** Se adoptó una sola rama para el control de versiones al ser un solo desarrollador.
- **Commits claros:** Se escribieron mensajes de commit significativos y sigue un formato estándar, como el de Conventional Commits.

- **Pull Requests:** Se integraron cambios mediante Pull Requests, asegurando las pasar revisiones automáticas y manuales.

8.4.5. Seguridad

- **Validación y sanitización:** Se protegió el sistema validando inputs y evitando vulnerabilidades como la inyección de código.
- **Autenticación y autorización:** Usa estándares como JWT para proteger APIs.
- **Gestión de secretos:** Claves o contraseñas en el código fuente como Dotenv.

9. CONCLUSIONES

Dependencia en Habilidades Full Stack: Aunque el rol del desarrollador full stack fue crucial, se evidenció una dependencia en una sola persona para ambas funciones de frontend y backend. Esto puede representar un riesgo significativo en términos de continuidad y escalabilidad del proyecto, ya que cualquier ausencia o cambio de personal podría afectar el desarrollo y mantenimiento del sistema. Sin embargo, este riesgo puede ser mitigado mediante una correcta y detallada documentación tanto del diseño como del desarrollo del sistema.

Cumplimiento de Estándares y Especificaciones: Se lograron los objetivos específicos relacionados con la especificación de requerimientos y el diseño de la aplicación. La utilización de estándares como ISO/IEC/IEEE 42010, IEEE 830 y el modelo Kruchten 4+1, junto con diagramas detallados, garantizó la calidad y consistencia en la planificación y diseño del sistema.

Uso de la lista de chequeo: La lista permitió verificar los requisitos funcionales y garantizar el correcto funcionamiento de la aplicación. Esto permitió el registro y actualización de los datos, validando la integridad y calidad de los datos de mascotas, vacunas, tratamientos, condiciones y el veterinario.

Etapas de Desarrollo: La etapa de desarrollo se completó con éxito en términos de funcionalidad y eficiencia del sistema. Sin embargo, el proyecto sufrió retrasos significativos debido a una curva de aprendizaje inicial pronunciada. Esto puso en evidencia la necesidad de un mejor plan de capacitación y transferencia de conocimientos para los desarrolladores involucrados.

Limitaciones en la Experiencia: La falta de experiencia previa y conocimientos específicos afectó el ritmo de avance en las primeras fases del proyecto. A pesar de que se logró superar esta limitación y adquirir experiencia valiosa, el proceso reveló la importancia de contar con un equipo con conocimientos robustos y experiencia en proyectos similares desde el inicio.

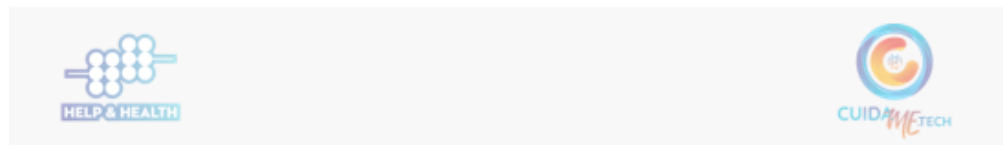
Realizar este proyecto ha sido una oportunidad significativa para desarrollar habilidades técnicas y metodológicas en el campo de la ingeniería de sistemas. Este proceso ha permitido no solo la adquisición de conocimientos prácticos sobre el desarrollo de software full stack, sino también la comprensión integral de cómo implementar soluciones tecnológicas que respondan a necesidades específicas de la comunidad. La experiencia adquirida a través de este proyecto fomenta un enfoque crítico y analítico, incentivando a los desarrolladores a explorar, innovar y aplicar sus conocimientos.

10. REFERENCIAS

- Bloomberg. (2021, October 2). *Transformación digital en Colombia, ¿qué sectores van más avanzados?* <https://www.bloomberglinea.com/2021/10/02/transformacion-digital-en-colombia-que-sectores-van-mas-avanzados/>
- Distrito Appnimal, una App que marca tendencia | Instituto Distrital de Protección y Bienestar Animal.* (n.d.). Retrieved November 27, 2024, from <https://www.animalesbog.gov.co/noticias/distrito-appnimal>
- Kruchten, P. B. (1995). The 4+1 View Model of Architecture. *IEEE Software*, 12(6), 42–50. <https://doi.org/10.1109/52.469759>
- Laika Mascotas - Universo peludo.* (n.d.). Retrieved November 27, 2024, from <https://laika.com.co/>
- Ley 1801 de 2016 Congreso de la República de Colombia.* (n.d.). Retrieved April 30, 2024, from <https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=66661>
- Ley 746 de 2002 - Gestor Normativo - Función Pública.* (n.d.). Retrieved April 30, 2024, from <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=5515>
- Pet Widget | LinkedIn.* (n.d.). Retrieved April 30, 2024, from <https://www.linkedin.com/company/pet-widget>
- PetHub | Your Pet Connected.* (n.d.). Retrieved April 30, 2024, from <https://www.pethub.com/>
- Serrano, R. (2020, August 30). *Colombia: estas aplicaciones te ayudarán a cuidar mejor a tu mascota - FOLOU.* Colombia. <https://folou.co/colombia/colombia-mascota/#:~:text=Aplicaciones para cuidar mejor a tu mascota 1,destaca por ofrecer servicios veterinarios a domicilio.>
- Servicios, Consultas y Clínicas Veterinarias 24 Horas • Movet.* (n.d.). Retrieved November 27, 2024, from <https://www.movet.co/>

11. ANEXOS

11.1. Contrato de Confidencialidad



CONTRATO DE CORRETAJE o ACUERDO COMERCIAL

Entre los suscritos,

LEONARDO ANDRES VINAZCO ULLOA con C.C. 79.957.122 de Bogotá, actuando en su calidad de Representante Legal de **HELP AND HEALTH SAS**, empresa legalmente constituida, con NIT. 901.063.443-8, de una parte, quien en lo sucesivo, y para todos los efectos de este contrato, se denominará **EL EMPRESARIO**, y **ANTHONY JARDIEL LÓPEZ PÉREZ** mayor de edad y vecino de Bogotá, de nacionalidad colombiana, identificada con cédula No. 1.007.673.204 de Sogamoso, obrando en nombre propio, de la otra parte, quien en lo sucesivo, y para todos los efectos de este contrato, se denominará **EL CORREDOR**, hemos convenido en celebrar el contrato de **CORRETAJE COMERCIAL**, de que se da cuenta en las siguientes cláusulas:

PRIMERA. OBJETO: La presente asociación tiene por finalidad y objeto comercializar los servicios que se deriven del software llamado **CUIDAME.tech**.

SEGUNDA. ÁMBITO GEOGRÁFICO: Las operaciones se ejecutarán en Colombia, a través de SOFTWARE **CUIDAME.tech**, sin perjuicio de ser ampliado a otros sitios, por acuerdo entre los contratantes.

TERCERA. BENEFICIOS: En adición a las obligaciones establecidas en el Código de Comercio y en otras cláusulas de este contrato, son obligaciones del **CORREDOR**:

CUARTA. EL CORREDOR debe promover la venta de licencias del SOFTWARE **CUIDAME.TECH**, las cuales son suministradas por **EL EMPRESARIO**.

QUINTA. EL CORREDOR se obliga a suministrar al **EL EMPRESARIO** toda la información que tenga a su **disposicion** y sea de su conocimiento y resulte importante para la **celebracion** del negocio, y especialmente, aquella que sea requerida por **EL EMPRESARIO** con el fin de realizar las respectivas consultas en los sistemas para la administración del Riesgo de Lavados de **Activios**, y Financiación de Terrorismo **SARLAFT**.

SEXTA. EL CORREDOR manifiesta que tiene pleno conocimiento del Sistema de **Administracion** del Riesgo de Lavado de Activos y de la **Financiacion** del Terrorismo - **SARLAF**, según lo establecido en la circular Básica **Juridica** de la Superintendencia de Sociedades, por ende se compromete a recaudar y aportar al **EL EMPRESARIO** toda la **documentación** e información que sea de su conocimiento sobre el posible comprador del servicio, manteniendo y aplicando en sus transacciones y negocios, los procedimientos, herramientas, sistemas y **metodos** suficientes para evitar ser sujeto de lavado de activos.

SÉPTIMA. EL CORREDOR se obliga a notificar por escrito a **EL EMPRESARIO**, de forma **inmediata**, cualquier acto o sospecha de corrupción, soborno, lavado de activos o financiación del



~~terrorismo~~, que se ~~presente~~ o se pudiera presentar en las negociaciones que haga durante la ejecución del presente contrato. EL EMPRESARIO podrá dar por terminado el presente contrato en cualquier tiempo y sin previo aviso a EL CORREDOR si este llegare a incumplir esta obligación.

OCTAVA. EL CORREDOR se obliga a abstenerse de utilizar el nombre EL EMPRESARIO para llevar a cabo en su nombre cualquier tipo de acto o negocio diferente al objeto de este contrato y realizar prácticas indebidas, de competencia desleal o de ~~incumplimiento~~ de la Ley 1480 de 2011.

NOVENA, EL CORREDOR se obliga a suministrar al público, posibles ~~clientes y clientes~~, información veraz, clara, suficiente y oportuna que no induzca a error, en cumplimiento del Estatuto del Consumidor.

DÉCIMA. EL EMPRESARIO aporta la marca con la experiencia de la empresa HELP AND HEALTH SAS.

DÉCIMA PRIMERA. Suministrar a EL CORREDOR toda la información que sea relevante e importante para el desarrollo y ejecución de este contrato.

DÉCIMA SEGUNDA. Pagar a EL CORREDOR la comisión correspondiente.

DÉCIMA TERCERA. EL CORREDOR actuará por su propia cuenta, con independencia, administrativa, financiera, técnica y jurídica, por lo que, EL CORREDOR, no tendrá ningún tipo de vinculación laboral con EL EMPRESARIO, ni podrán exigir de ella pagos por concepto de salarios, prestaciones sociales, y en general ninguna de las sumas que son inherentes al contrato de trabajo. Así mismo, la relación que se genera entre las Partes es estrictamente comercial y se rige por las normas del código de comercio relacionadas con el contrato de corretaje y por las demás que las adicionen o complementen. Por ningún motivo, el presente Contrato se entenderá que muta en uno diferente, razón por la cual las Partes declaran que no ha habido ni habrá lugar al pago de la cesantía comercial ni a remuneración distinta a la que le corresponda a EL CORREDOR producto del presente Contrato.

DÉCIMA CUARTA. EL CORREDOR se compromete a mantener la más estricta confidencialidad y reserva sobre el contenido de este Contrato, así como sobre el negocio objeto ~~del mismo~~ y cualquier información o documento que puedan llegar a conocer sobre la otra Parte como consecuencia del mismo. En cumplimiento de esta obligación, la Parte responsable se compromete a no dar a terceros ningún tipo de información que conozcan sobre la otra Parte. Para efectos de este Contrato se entiende por Información Confidencial la proveniente directamente de la Parte Reveladora, la que llegué a conocer la Parte Receptora por razón del presente Contrato, aún si no le es suministrada expresa y directamente, y la que obtenga durante el desarrollo del objeto de este Contrato. Por lo tanto, su utilización y revelación está restringida de conformidad con las reglas que adelante se mencionan 1) La Parte Receptora entiende y acepta que la Información Confidencial que reciba se debe destinar exclusivamente a cumplir los fines de este Contrato. En consecuencia, no podrá revelar ~~bajo ninguna circunstancia~~ y a persona alguna, diferente del personal autorizado de la Parte Reveladora, ni destinarla a propósitos distintos de aquellos para los cuales fue suministrada o por razón de los cuales fue adquirida. 2) Si por cualquier circunstancia una autoridad competente solicita que sea revelada la Información Confidencial, La Parte Receptora dará aviso de dicha orden a la Parte Reveladora, previamente al cumplimiento de la orden.

3) Cuando cualquier Información Confidencial sea puesta en conocimiento público por medios



lícitos, y sin que haya violación de esta cláusula, dejará de tener tal condición y en consecuencia, no le será aplicable la presente cláusula. 4) Cuando quiera que la Parte Receptora tenga conocimiento de que se ha producido o va a producirse una revelación de la Información Confidencial, se obliga a notificar de este hecho de manera inmediata a la Parte Reveladora.

DÉCIMA QUINTA. DURACIÓN. El término de duración de la asociación que por este contrato se constituye es de SEIS (6) meses, que se contará a partir de la fecha de la firma de este documento. No obstante, las partes harán evaluaciones periódicas del negocio y, si los resultados no corresponden a las expectativas, podrán darlo por terminado en cualquier momento.

DÉCIMA SEXTA. UTILIDADES. EL CORREDOR tiene derecho, al cinco por ciento (5%) del valor de los contratos efectivamente firmados, como consecuencia directa de su labor en la consecución y perfeccionamiento de VENTA DE LICENCIAS DE CUIDAME.TECH y/o negocios en operaciones mercantiles.

DÉCIMA SÉPTIMA. COSTOS OPERATIVOS. Todos los costos relacionados con la operación, ejecución o implementación del servicio son de responsabilidad de EL EMPRESARIO.

DÉCIMA OCTAVA: CLÁUSULA COMPROMISORIA: Las diferencias que ocurran entre las partes con motivo de la interpretación, ejecución y cumplimiento del presente contrato de cuentas en participación, serán sometidas a la jurisdicción de un Tribunal de Arbitramento designado por el Centro de Arbitraje y Conciliación de la Cámara de Comercio de Tunja, a solicitud escrita de cualquiera de las partes contratantes, que se sujetará a las normas vigentes para el Arbitramento, de acuerdo con las siguientes reglas:

- A. El Tribunal estará integrado por un (1) árbitro, que designará la Cámara de Comercio de Tunja.
- B. La organización interna del Tribunal se sujetará a las reglas previstas para el efecto por el Centro de Amigable Composición, Arbitraje, Conciliación de la Cámara de Comercio de Tunja.
- C. El Tribunal decidirá en derecho.
- D. El Tribunal funcionará en el municipio de Tunja en el Centro de Amigable Composición, Arbitraje y Conciliación de la Cámara de Comercio de Tunja. En lo no previsto en esta cláusula, se procederá de acuerdo con lo dispuesto en las normas vigentes en el momento de hacerse necesario el Tribunal de Arbitramento.
- E. Se entiende por parte la persona o grupo de personas que sostengan una misma pretensión.

DÉCIMA NOVENA: La marca CUIDAME.TECH es propiedad exclusiva de EL EMPRESARIO, y por la realización de este acuerdo no hay cesión, licencia o permiso de utilización independiente del negocio de que aquí se trata.

DUODÉCIMA: Los derechos sobre este contrato son intransferibles, toda vez que el mismo se hace en consideración a las partes que en él intervienen. Sólo podrá transferirse con la autorización expresa, y por escrito, de las partes.

11.1.1. Elementos cubiertos por el contrato de confidencialidad

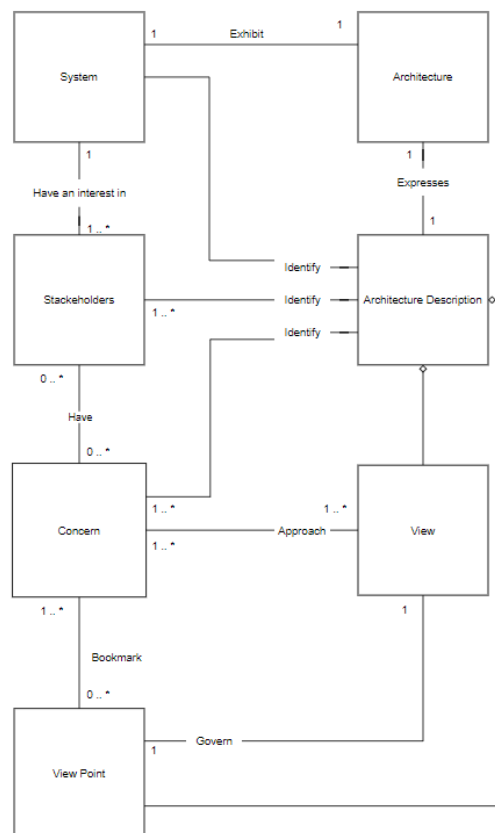
La siguiente información no puede ser divulgada al público debido a su naturaleza confidencial, la cual incluye detalles sobre el modelo de negocio de la empresa y su funcionamiento:

- Estructura de Desglose de Trabajo (WBS)
- Modelo ER
- Diagrama de Clases
- Diagramas de Procesos
- Diagramas de Desarrollo
- Diagrama Físico
- Proyecto de Backend
- Proyecto de Frontend

11.2. Especificación ISO/IEC/IEEE 42010

Este estándar internacional establece un marco para la descripción de arquitecturas de sistemas y software, permitiendo documentar, analizar y comunicar las decisiones arquitectónicas de forma estandarizada. Su alcance abarca la definición, especificación y comunicación de las arquitecturas de sistemas, enfocándose en cómo las vistas arquitectónicas responden a las necesidades y preocupaciones de los interesados. Aplica a sistemas complejos que integren componentes de hardware y software, en cualquier etapa del ciclo de vida del desarrollo.

Figura 19: Diagrama de ISO/IEC/IEEE 42010
ISO/IEC/IEEE 42010 Architecture Design



Fuente: Autor

Descripción: La imagen muestra diagrama utilizado para la especificación e identificación de stakeholders, architecture description, views, entre otros.

11.3. Especificación de requerimientos

Este estándar ofrece una guía estructurada para la creación de especificaciones de requisitos de software (SRS). Su objetivo es garantizar que los requisitos sean claros, completos, consistentes y verificables, sirviendo como base para el diseño, desarrollo y pruebas del sistema. Establece las buenas prácticas para definir las funcionalidades y restricciones de los sistemas de software. Cubre la elaboración de documentos de requisitos de software, orientándose hacia la comunicación efectiva entre los desarrolladores, clientes y otras partes interesadas. Incluye tanto requisitos funcionales como no funcionales.

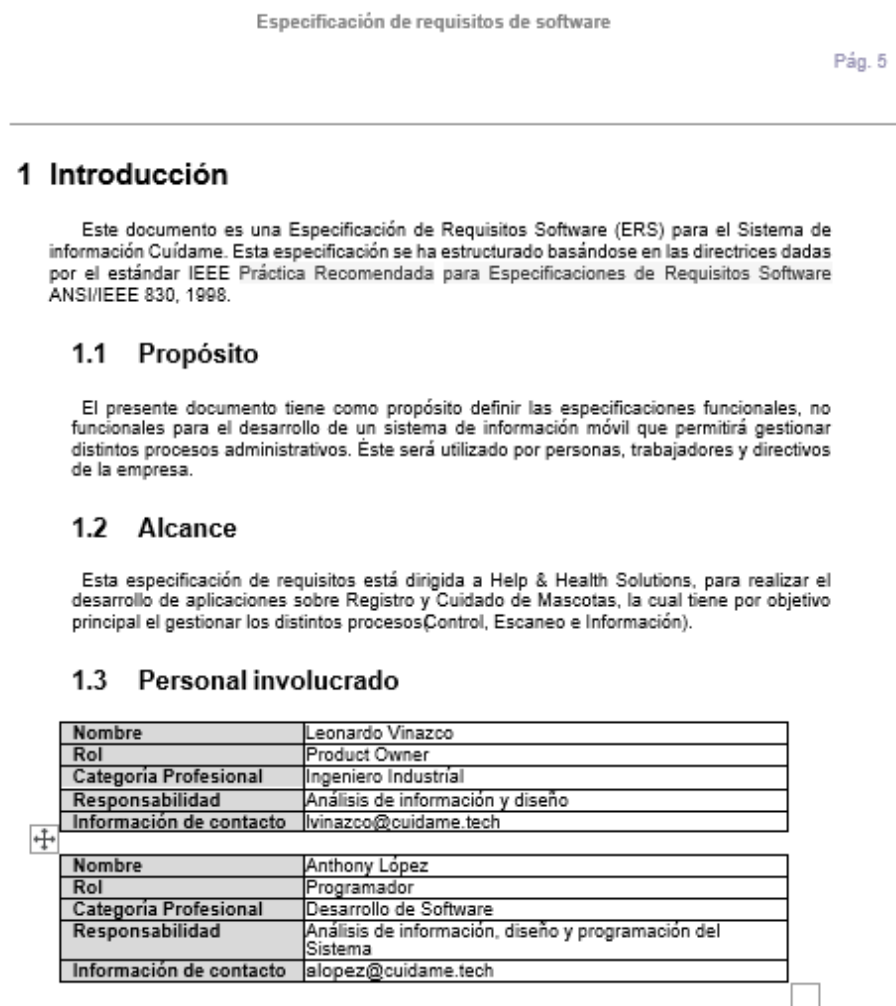
Figura 20: Contenido del documento IEEE 830

Especificación de requisitos de software		Pág. 4
Contenido		
Ficha del documento		2
Contenido		4
1	Introducción.....	5
1.1	Propósito.....	5
1.2	Alcance	5
1.3	Personal involucrado.....	5
1.4	Definiciones, acrónimos y abreviaturas.....	6
1.6	Resumen.....	6
2	Descripción general	6
2.1	Perspectiva del producto	6
2.2	Funcionalidad del producto	7
2.4	Restricciones	7
2.5	Suposiciones y dependencias	7
3	Requisitos específicos.....	8
3.2	Requisitos funcionales	14
	3.2.1 Requisito funcional 1	14
	3.2.2 Requisito funcional 2	15
	3.2.3 Requisito funcional 3	15
	3.2.4 Requisito funcional 4	15
	3.2.5 Requisito funcional 5	15
	3.2.6 Requisito funcional 7	15
	3.2.7 Requisito funcional 8	15
	3.2.8 Requisito funcional 9	16
3.3	Requisitos no funcionales	16
	3.3.1 Requisitos de rendimiento	16
	3.3.2 Seguridad	16
	3.3.3 Fiabilidad	16
	3.3.4 Disponibilidad	16
	3.3.5 Mantenibilidad	16
	3.3.6 Portabilidad	16

Fuente: Autor

Descripción: La imagen muestra a cada uno de los ítems que se desarrollaron en el documento de requisitos del software.

Figura 21: Inicio del documento IEEE 830



Fuente: Autor

Descripción: La imagen es un ejemplo de cómo se abarcó el documento de especificación de requerimientos.

Figura 22: Parte final del inicio del documento

Especificación de requisitos de software

Pág. 7

2.2 Características de los usuarios

Tipo de usuario	Usuario
Formación	Básica
Actividades	Registrar información personal e información básica de mascotas e información de salud.

2.3 Restricciones

- Interfaz para ser usada con internet.
- Uso de Dominio (app.cuidame.tech)
- Lenguajes y tecnologías en uso: HTML, TYPESCRIPT, ANGULAR, JAVASCRIPT, CSS, IONIC, NODE.JS, GOOGLE CLOUD PLATFORM, POSTGRESQL
- Los servidores deben ser capaces de atender consultas concurrentemente.
- El sistema se diseñará según un modelo cliente/servidor.
- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

2.4 Suposiciones y dependencias

- Se asume que los requisitos aquí descritos son estables
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta de la misma

Fuente: Autor

Descripción: La imagen es un ejemplo de cómo se abarcó el documento de especificación de requerimientos.

Figura 23: Requisitos Específicos

Especificación de requisitos de software

Pág. 8

3 Requisitos específicos

Requerimientos Funcionales

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Autenticación de Usuario.
Características:	Los usuarios deberán identificarse para acceder a cualquier parte del sistema.
Descripción del requerimiento:	El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad.
Requerimiento NO funcional:	• RNF01 • RNF02 • RNF05 • RNF08

Prioridad del requerimiento:
Alta

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Registrar Usuarios.
Características:	Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema.
Descripción del requerimiento:	El sistema permitirá al usuario registrarse. El usuario debe suministrar datos como: Nombre, Apellido, Tipo de Documento, Número de Documento, Email, Departamento y Ciudad, Celular y Password.
Requerimiento NO funcional:	• RNF01 • RNF02 • RNF05 • RNF08

Prioridad del requerimiento:
Alta

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Consultar Información.
Características:	El sistema ofrecerá al usuario información general acerca de la información básica, lista de mascotas, información básica de la mascota, información de vacunas, medicamentos, tratamientos y veterinario personal.
Descripción del requerimiento:	<u>Consultar Información:</u> Muestra información general sobre la mascota, vacunas, medicamentos, tratamientos y veterinario personal.
Requerimiento NO funcional:	• RNF01 • RNF02
Prioridad del requerimiento:	
Alta	

Fuente: Autor

Descripción: La imagen es un ejemplo de cómo se abarcó el documento de especificación de requisitos funcionales.

Figura 24: Requerimientos no funcionales.

Especificación de requisitos de software

Pág. 12

Requerimientos No Funcionales.

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Interfaz del sistema.
Características:	El sistema presentara una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
Descripción del requerimiento:	El sistema debe tener una interfaz de uso intuitiva y sencilla.
Prioridad del requerimiento:	Alta

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Ayuda en el uso del sistema.
Características:	La interfaz del usuario deberá de presentar un sistema de ayuda para que los mismos usuarios del sistema se les faciliten el trabajo en cuanto al manejo del sistema.
Descripción del requerimiento:	La interfaz debe estar complementada con un buen sistema de ayuda (la administración puede recaer en personal con poca experiencia en el uso de aplicaciones informáticas).
Prioridad del requerimiento:	Alta

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Diseño de la interfaz.
Características:	El sistema deberá de tener una interfaz de usuario, teniendo en cuenta las características móviles.
Descripción del requerimiento:	La interfaz de usuario debe ajustarse a las características móviles, dentro de la cual estará incorporado el sistema de gestión de procesos, información y escaneo.
Prioridad del requerimiento:	Alta

Fuente: Autor

Descripción: La imagen es un ejemplo de cómo se abarcó el documento de especificación de requerimientos no funcionales.

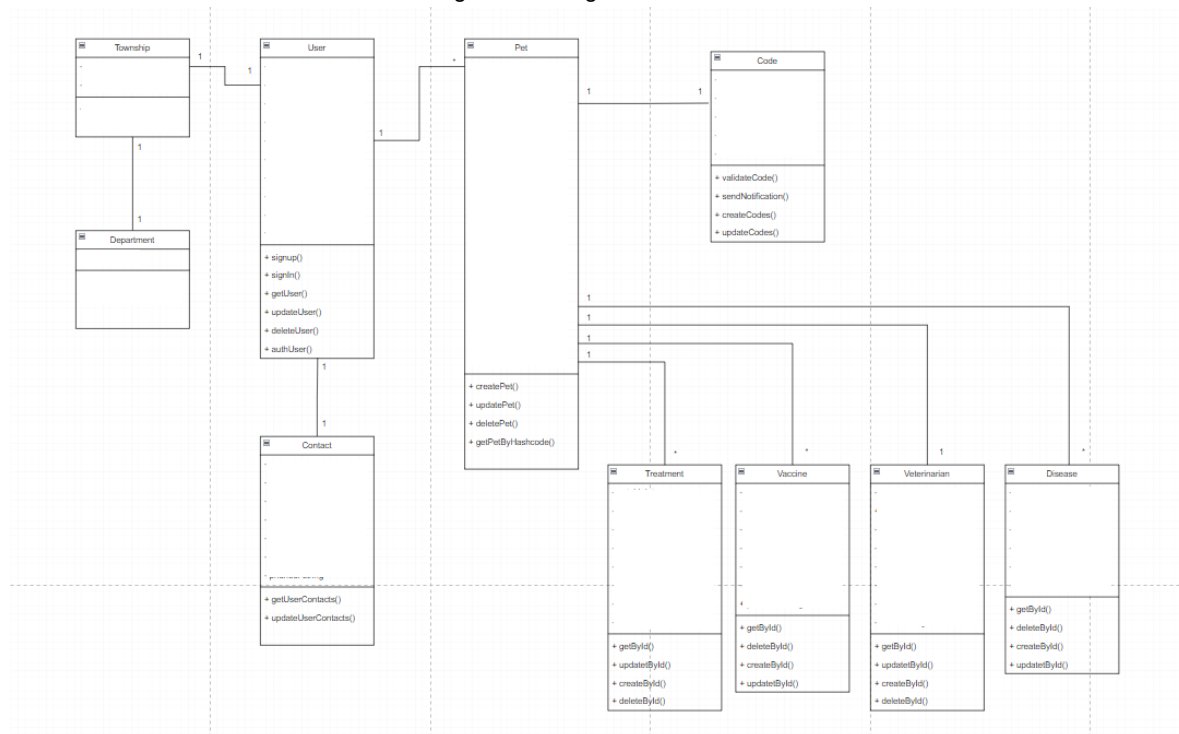
11.4. Diagramas

Los siguientes anexos se adjuntarán mediante imágenes evidenciando el desarrollo de los diseños previos al desarrollo de la pasantía ocultando información sensible.

11.4.1. Anexo diagrama de clase

Representación gráfica en UML que muestra la estructura estática del sistema, destacando sus clases, atributos, métodos y relaciones. Este diagrama permite modelar la organización y las interacciones entre las entidades principales del sistema, sirviendo como una herramienta clave en el diseño y documentación.

Figura 25: Diagrama de clases.



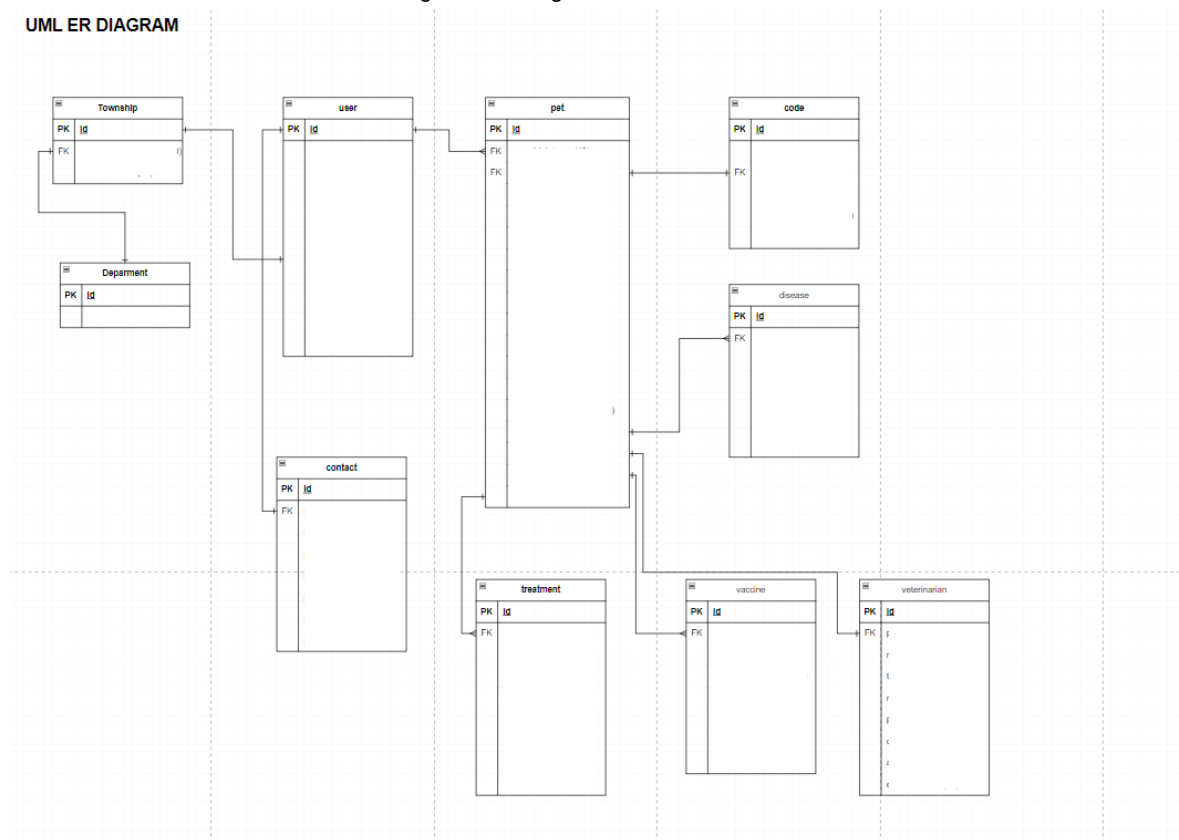
Fuente: Autor

Descripción: La imagen es un ejemplo de cómo se abarcó el diagrama de clases para el desarrollo.

11.4.2. Anexo diagrama entidad relación

Representación gráfica que modela los datos de un sistema, destacando las entidades principales, sus atributos y las relaciones entre ellas. Cada entidad representa un conjunto de objetos del mundo real con características comunes, mientras que las relaciones indican cómo interactúan o se asocian dichas entidades. Este diagrama es fundamental para el diseño de bases de datos, proporcionando una estructura clara y organizada que facilita la comprensión y posterior implementación del modelo de datos.

Figura 26: Diagrama entidad relación.



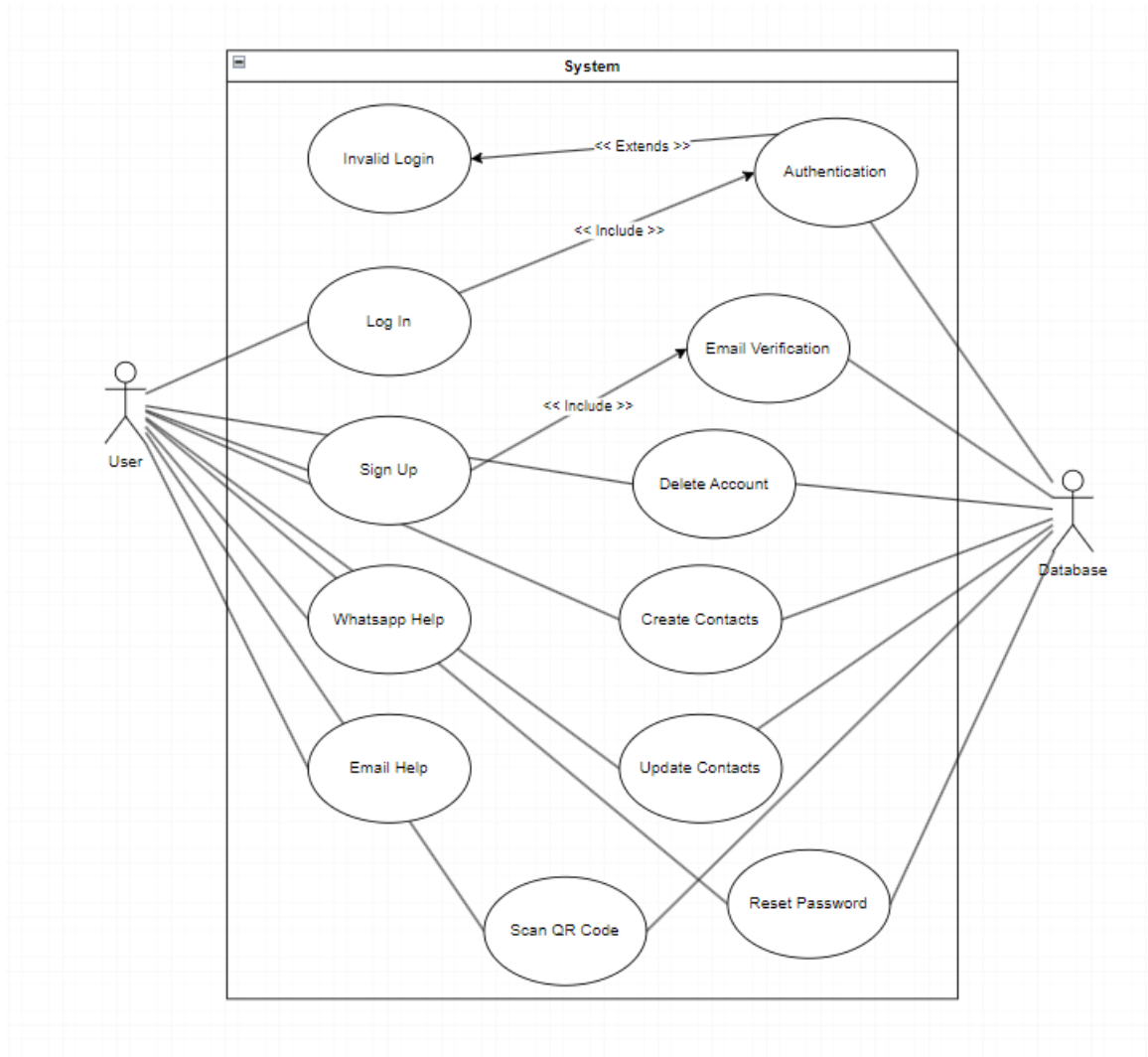
Fuente: Autor

Descripción: La imagen es un ejemplo de cómo se abarcó el diagrama entidad relación para el desarrollo tecnológico.

11.4.3. Anexo casos de uso

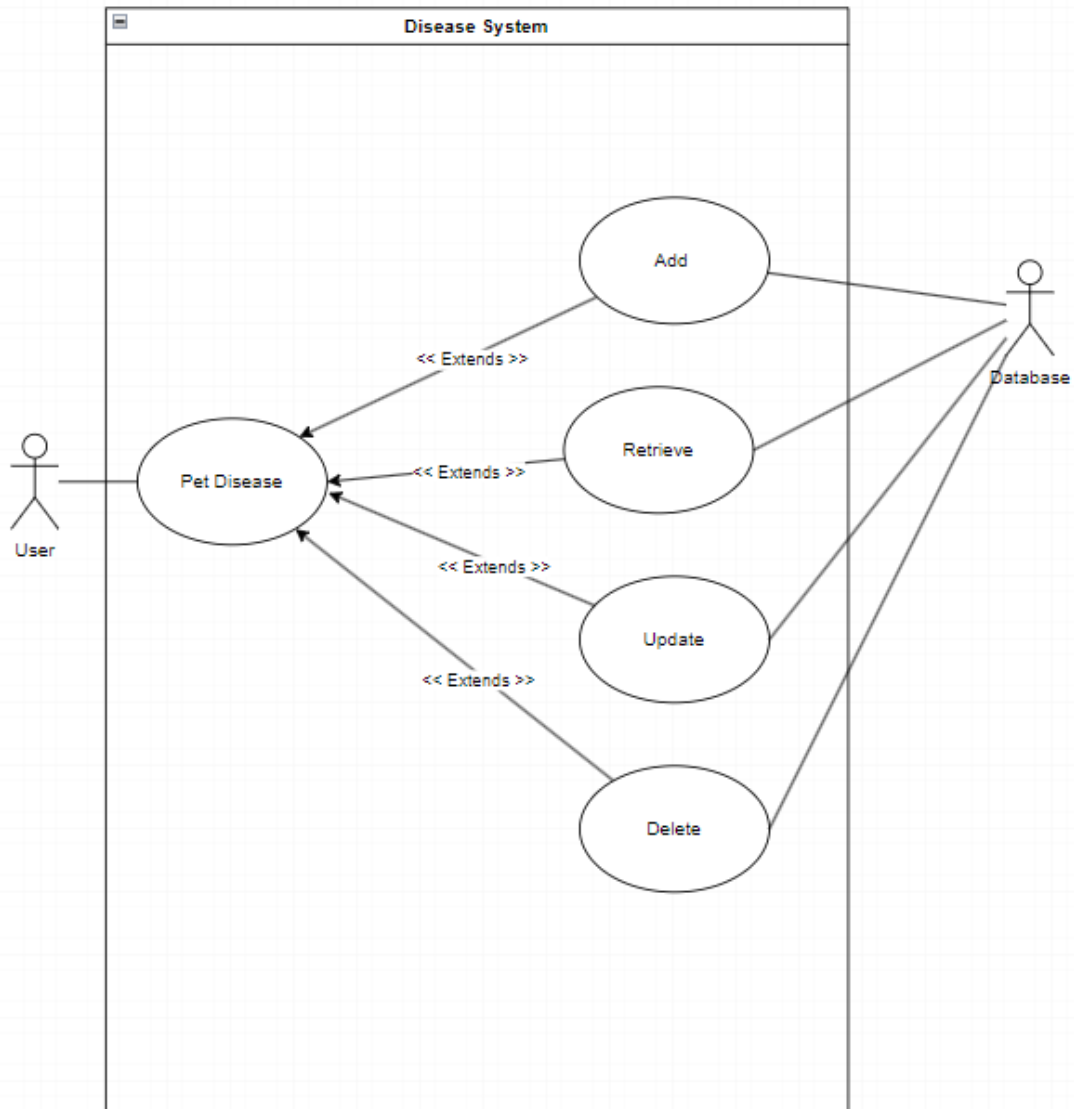
El diagrama de caso de uso describe las interacciones entre los usuarios (actores) y el sistema en relación con funcionalidades específicas. Este diagrama representa visualmente cómo los usuarios realizan acciones clave en el sistema.

Figura 27: Caso de uso. Sistema



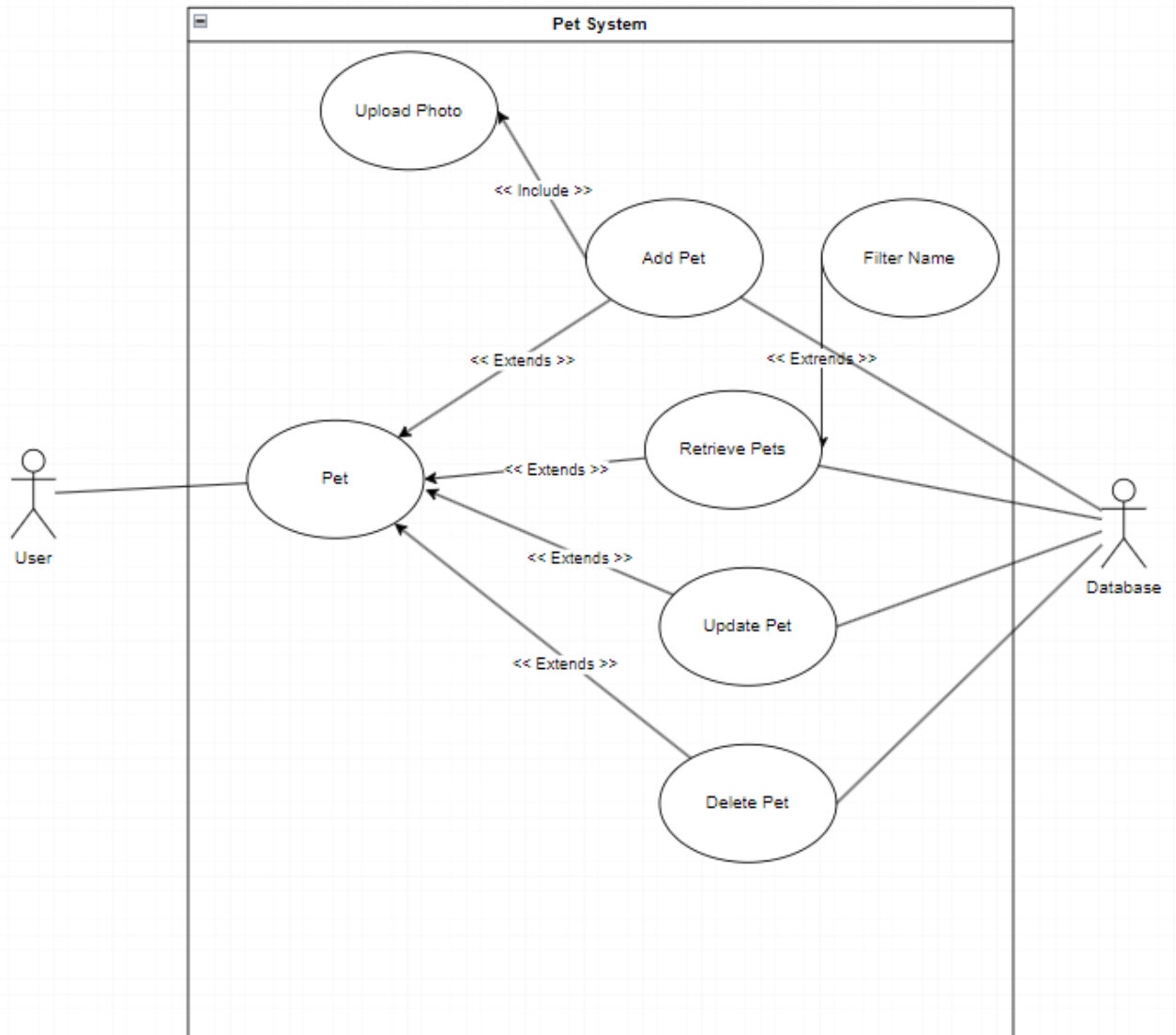
Fuente: Autor

Figura 28: Caso de uso. Sistema de enfermedades.



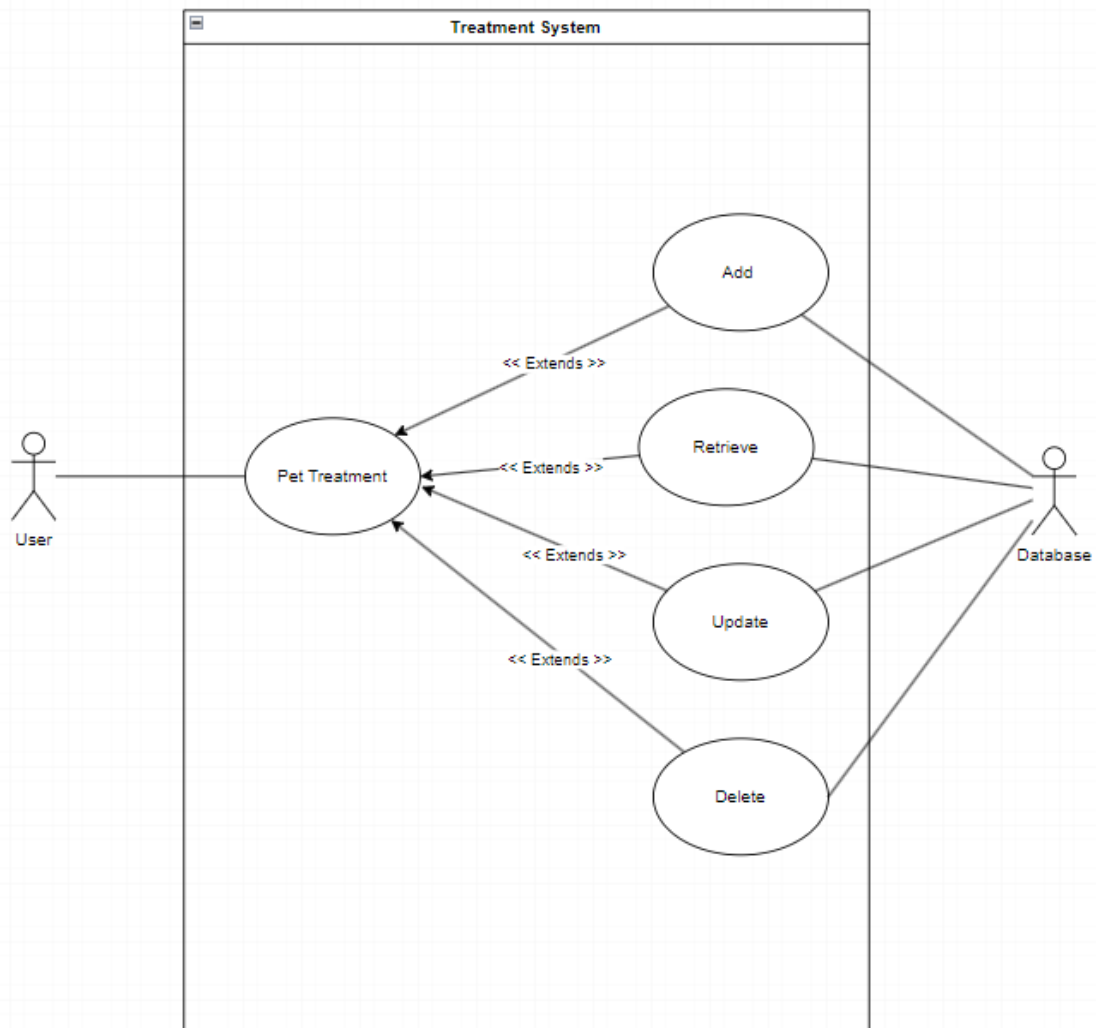
Fuente: Autor

Figura 29: Caso de Uso. Sistema de la mascota.



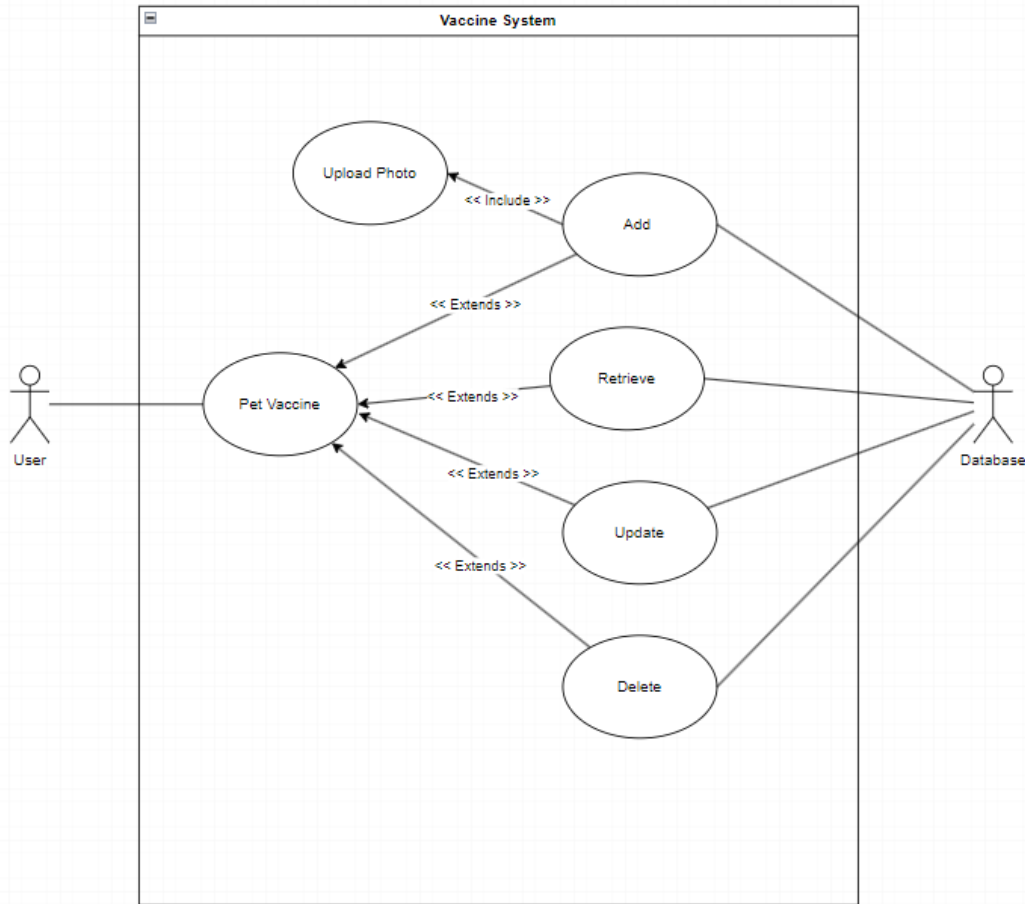
Fuente: Autor

Figura 30: Caso de uso. Sistema de tratamiento.



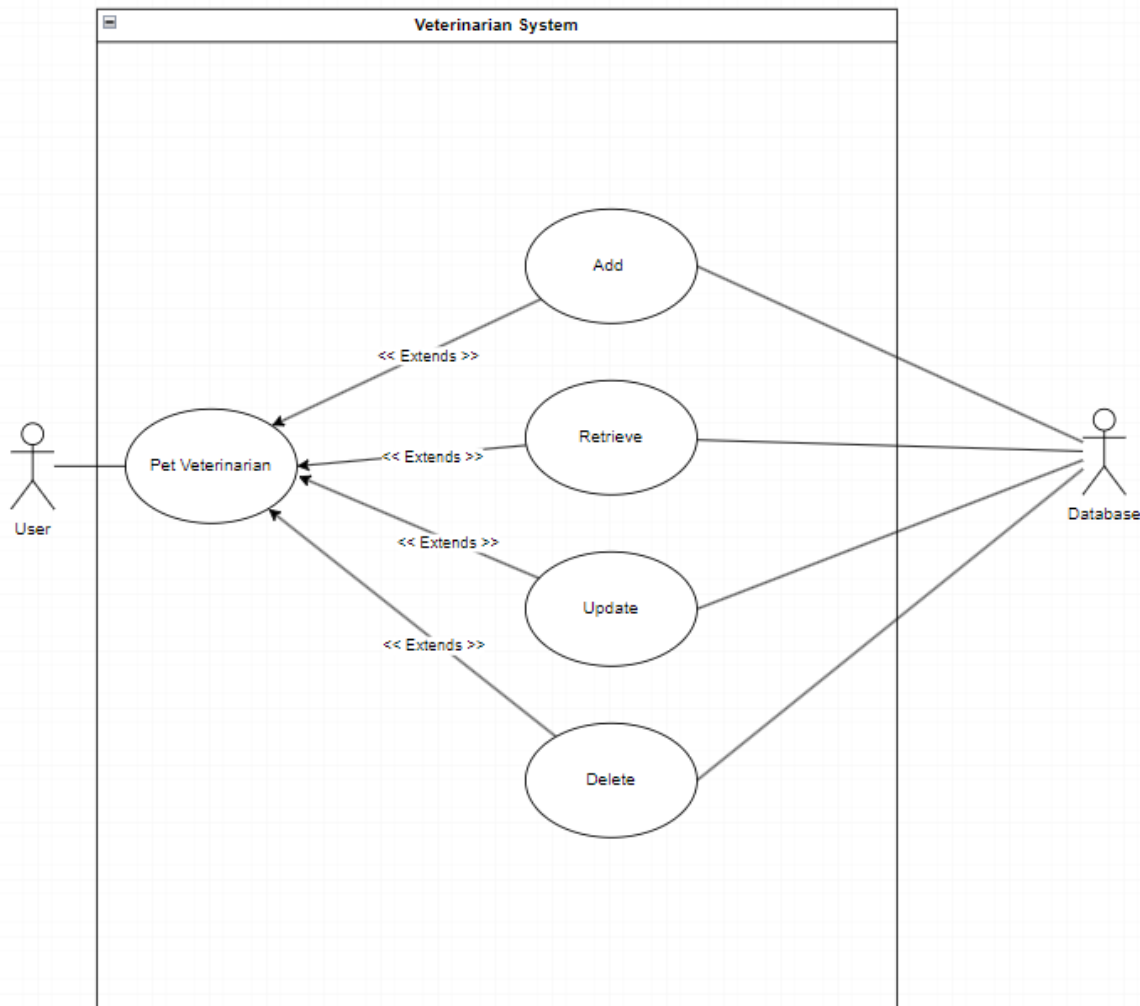
Fuente: Autor

Figura 31: Caso de uso. Sistema de vacunas.



Fuente: Autor

Figura 32: Caso de uso. Sistema de veterinario.

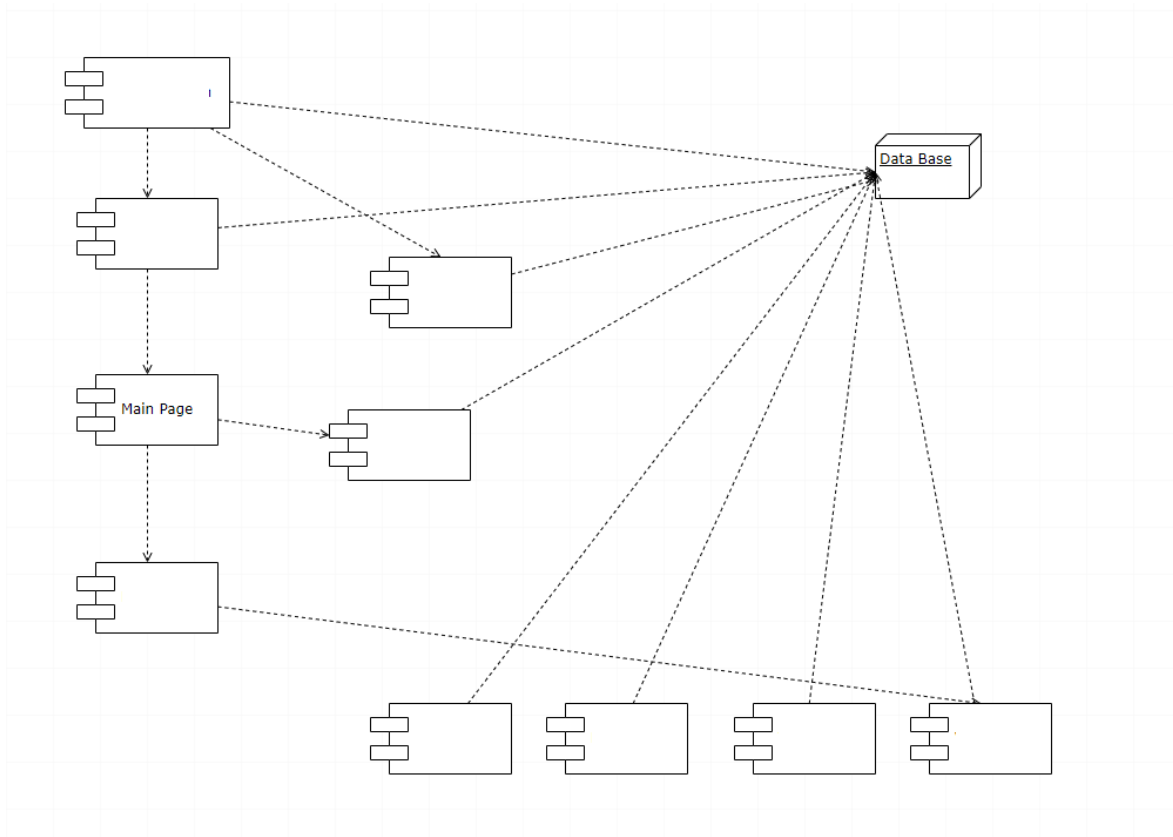


Fuente: Autor

11.4.4. Anexo diagrama de desarrollo

El diagrama de desarrollo bajo la vista de Kruchten 4+1 ocultando información que podría ser sensible, es parte del modelo arquitectónico que organiza el diseño del sistema en cinco vistas, integrando distintos aspectos del desarrollo. Este enfoque asegura que todas las perspectivas relevantes para desarrolladores, arquitectos y usuarios se contemplen. La vista de desarrollo se enfoca específicamente en la organización estática del software desde el punto de vista del programador.

Figura 33: Diagrama de desarrollo.

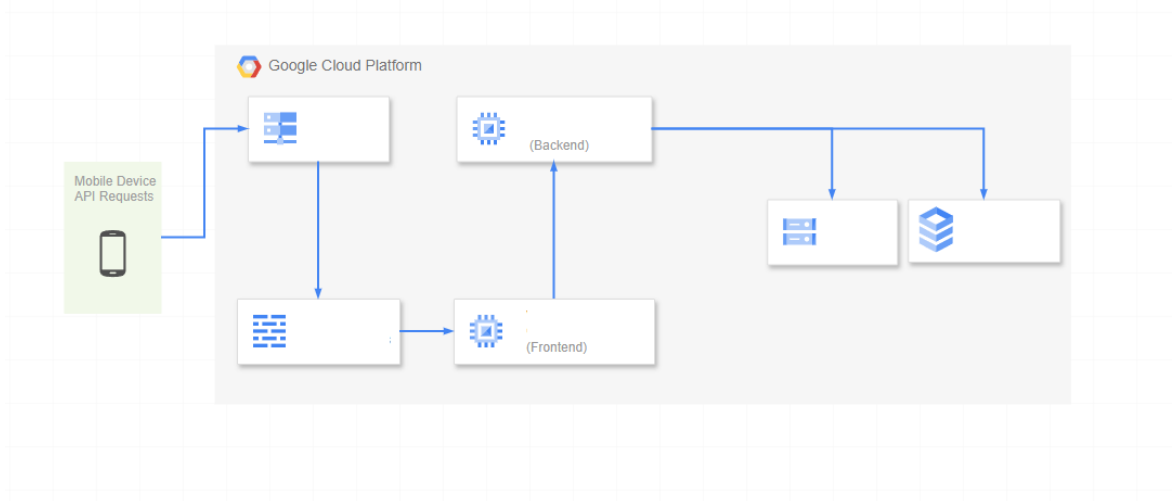


Fuente: Autor

11.4.5. Anexo diagrama físico

El diagrama físico, dentro del modelo arquitectónico 4+1, representa la disposición y organización del sistema en términos de su infraestructura física y los componentes de hardware necesarios para su ejecución. Este diagrama muestra cómo se despliegan los elementos del sistema en servidores, redes y dispositivos, proporcionando una vista clara del entorno operativo ocultando información sensible.

Figura 34: Diagrama de desarrollo.



Fuente: Autor

11.4.6. Anexo diagramas de flujo

Los diagramas de flujo son una representación gráfica que detalla el flujo de procesos o actividades dentro del sistema. Se utilizaron símbolos estandarizados como rectángulos, flechas, diamantes y óvalos para describir los pasos, decisiones y conexiones en un procedimiento lógico o secuencia de operaciones. Este tipo de diagrama es ampliamente utilizado para visualizar y analizar procesos, identificar cuellos de botella y optimizar la ejecución de tareas.

```
graph LR
    User((User)) --> Start((start))
    Start --> OpenApp[Open App]
    OpenApp --> HaveAccount{Have an Account?}
    HaveAccount -- Yes --> Login[Log in]
    HaveAccount -- No --> SignUp[Sign up]
    SignUp --> FillInfo[Fill out the information]
    FillInfo --> Submit[Submit]
    Submit --> CheckEmail[check email address]
    CheckEmail --> Login
    Login --> FillCreds[Fill credentials]
    FillCreds --> ForgotPass{Forgot Password?}
    ForgotPass -- Yes --> ResetPass[Reset Password]
    ResetPass --> FillCreds
    ForgotPass -- No --> ValidCreds{Valid credentials?}
    ValidCreds -- Yes --> End((End))
    ValidCreds -- No --> FillCreds
```

Figura 36: Diagrama de flujo pantalla inicial y registro de información para la mascota.

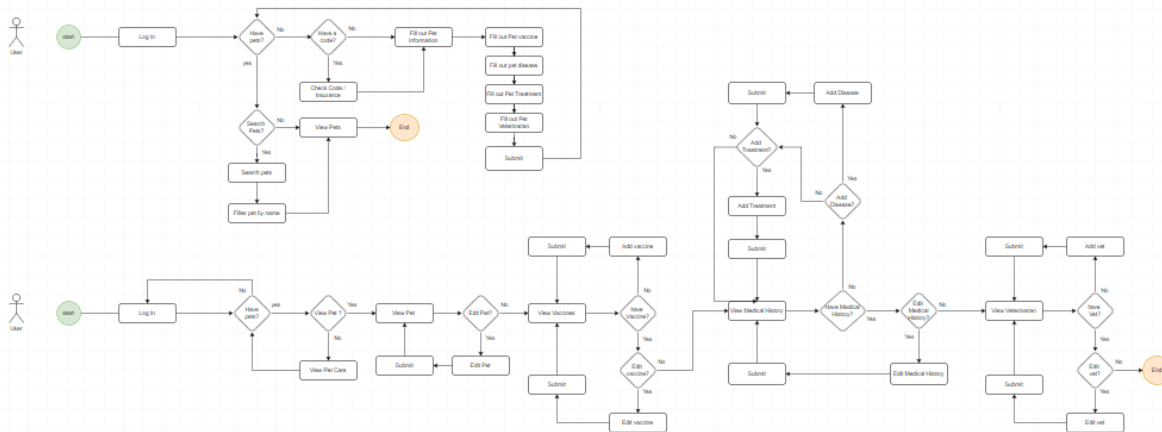
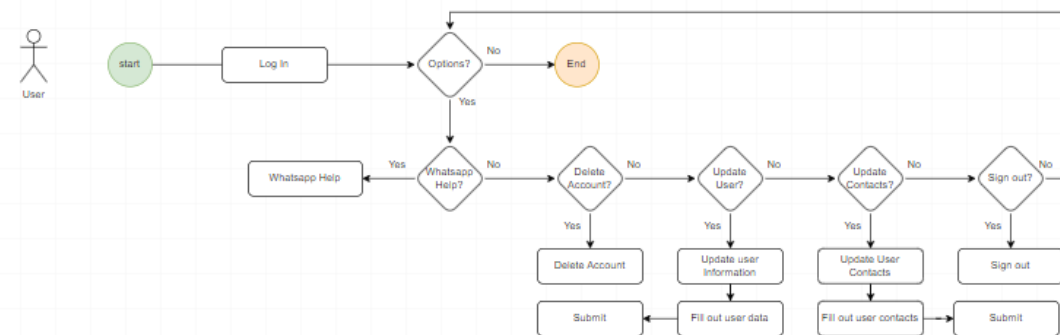
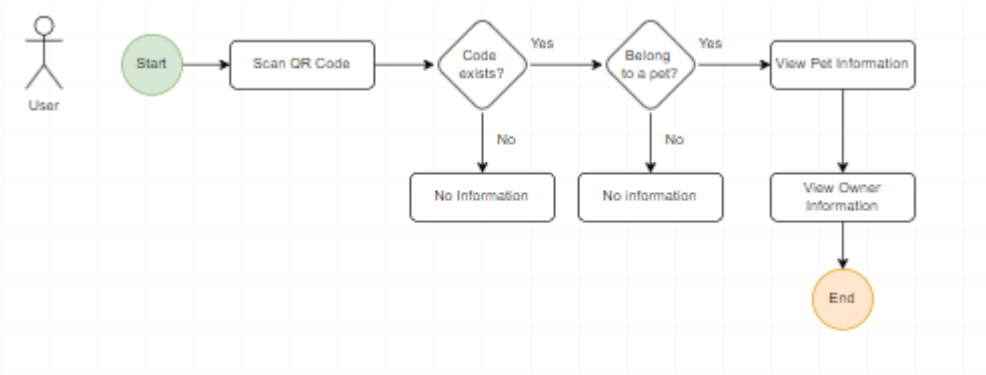


Figura 37: Diagrama de flujo para el apartado del action sheet de la aplicación.



Fuente: Autor

Figura 38: Diagrama de flujo escaneo de código QR.



Fuente: Autor

11.4.7. Anexo listas de chequeo

La lista de chequeo de requerimientos y funcionalidades en Excel es una herramienta organizada que permite planificar, documentar y verificar el cumplimiento de las necesidades del sistema durante su desarrollo. Este formato tabular facilita la trazabilidad de los requisitos, asegurando que cada uno sea evaluado y cumplido en el proceso adjuntando evidencias.

Figura 39: Lista de chequeo en Excel módulo sistema.

CUIDAME APP									
System Module									
ID #	Título	Precondiciones	Pasos	Resultado Esperado	Status	Prioridad	Evidencia	Evidencia	Evidencia
	Verificar que se pueda logear al sistema 1. utilizando credenciales validas	Estar situado en el login de la plataforma Tener credenciales validas	1- Ingresar el usuario en el campo user 2- Ingresar la contraseña en el campo password 3- Clickear en el botón "Entrar"	El sistema debería redireccionar al dashboard principal	PASS	Crítica			
	Cerrar sesión existente	Estar logueado en el sistema	1- Ingresar al sistema. 2- Desplegar menú. 3- Presionar botón cerrar sesión	El sistema deberá cerrar la sesión del usuario.	PASS	Crítica			
	Verificar que NO se pueda logear al sistema utilizando credenciales invalidas 2.	Estar situado en el login de la plataforma	1- Ingresar el usuario en el campo user 2- Ingresar la contraseña en el campo password 3- Clickear en el botón "Entrar"	El sistema debería mostrar el siguiente mensaje de error: "El usuario o contraseña son incorrectos"	PASS	Crítica			
	Trasr todas los familiares que el usuario 2 tenga guardados	Estar logueado en la plataforma	1- Ingresar al sistema	El sistema debe traer el id, el nombre, la foto de cada familiar registrado.	PASS	Crítica			
	Verificar el redireccionamiento al chat de 3 ayuda de Whatsapp	Estar logueado en el sistema	1- Ingresar al sistema. 2- Desplegar menú. 3- Presionar botón ayuda en whatsapp.	El sistema redirige al usuario al navegador para posteriormente abrir el chat de whatsapp	PASS	Crítica			
	Verificar el redireccionamiento al correo de 4 ayuda de Cuidame	Estar logueado en el sistema	1- Ingresar al sistema. 2- Desplegar menú. 3- Presionar botón ayuda en Email	El sistema redirige al usuario al navegador para redactar un email al correo electrónico.	PASS	Crítica			
	Scanear código y habilitar permisos de		1. Scanear código QR usando la cámara del						

Fuente: Autor

Figura 40: Lista de chequeo en Excel módulo sistema 2.

5	Scanear código y habilitar permisos de ubicación	Dispositivo con cámara.	1. Scanear código QR usando la cámara del teléfono.	Pedir permisos de ubicación.	PASS	Crítica			
6	Scanear código QR con una mascota asociada	Dispositivo con cámara. Mascota registrada.	1. Scanear código QR usando la cámara del teléfono.	Mostrar la información básica de la mascota, información del propietario, enviar notificación por whatsapp.	PASS	Crítica			
7	Scanear código QR sin una mascota asociada	Dispositivo con cámara. Código no vinculado.	1. Scanear código QR usando la cámara del teléfono.	Mostrar la información básica de la mascota, información del propietario, NO enviar notificación por whatsapp.	PASS	Crítica			
6	Recuperar la contraseña	Cuenta registrada en la app.	1. Click olvidar contraseña. 2. El sistema verifica si el correo está registrado. 3. Recibir enlace de recuperación de contraseña. 4. Cambiar contraseña.	Enviar correo de recuperación de contraseña. Cambiar contraseña.	PASS	Crítica			
7	Eliminar la cuenta	Ingresar al sistema	1- Ingresar al sistema. 2. Desplegar menú. 3. Presionar botón delete account. 4. Confirmar la eliminación de la cuenta.	Eliminar toda la información relacionada con la cuenta.	PASS	Crítica			
8	Guardar contactos de emergencia.	Ingresar al sistema	1- Ingresar al sistema. 2. Desplegar menú. 3. Presionar botón Contactos. 4. Guardar los contactos existentes.	Guardar los contactos de emergencia.	PASS	Crítica			
9	Actualizar contactos de emergencia.	Ingresar al sistema	1- Ingresar al sistema. 2. Desplegar menú. 3. Presionar botón Contactos. 4. Actualizar los contactos existentes.	Actualizar los contactos de emergencia.	PASS	Crítica			
10	Búsqueda de los familiares registrados	Ingresar al sistema	1- Ingresar al sistema. 2. Utilizar la barra de búsqueda.	Filtrar la información de acuerdo a la cadena de búsqueda.	PASS	Crítica			

Fuente: Autor

Figura 41: Lista de chequeo en Excel Pets.

CUIDAME APP Pet Module									
ID #	Título	Precondiciones	Pases	Resultado Esperado	Status	Prioridad	Evidencia	Evidencia	Evidencia
	Registrar la información básica de la mascota	Estar logueado en el sistema.	1. Presionar el botón de agregar. 2. Seleccionar mascota. 3. Continuar sin seguro. 3. Llenar el formulario.	Guardar la información de la mascota. Asignar un código QR existente para la mascota, sin licencia y de tipo Pets. En caso de que no sepa la raza el sistema asignará por defecto "Criolla". El sistema asignará un identificador para la mascota.	PASS	Crítica			
9	Obtener la información de la mascota	Estar logueado en el sistema.	1. Ver todos los familiares. 2. Seleccionar la mascota. 3. Ver información de la mascota.	Obtener la información básica de la mascota y el propietario.	PASS	Crítica			
9	Actualizar la información de la mascota	Estar logueado en el sistema.	1. Ver todos los familiares. 2. Seleccionar la mascota. 3. Ver información de la mascota. 4. click en editar. 5. Editar información.	Actualizar la información de la mascota.	PASS	Crítica			
	Registrar la información básica de la mascota con un código de seguro.	Estar logueado en el sistema.	1. Presionar el botón de agregar. 2. Seleccionar mascota. 3. Validar el código del seguro. 3. Llenar el formulario.	Guardar la información de la mascota. Asignar un código QR que el usuario ingresó para la mascota, sin licencia y de tipo Pets. En caso de que no sepa la raza el sistema asignará por defecto "Criolla". El sistema asignará un identificador para la mascota.	PASS	Crítica			
	Eliminar la información de la mascota.	Sin precondiciones de momento	Desarrollo Backend	Eliminar toda la información relacionada a la mascota.	PASS	Crítica			

Fuente: Autor

Figura 42: Lista de chequeo en Excel vacunas.

CUIDAME APP Vaccine Module									
ID #	Título	Precondiciones	Pasos	Resultado Esperado	Status	Prioridad	Evidencia	Evidencia	Evidencia
9	Registrar la información de vacunas de la mascota.	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar vacunas. 3. Agregar Vacuna y llenar el formulario.	Guardar la información de la vacuna.	PASS	Crítica			
9	Obtener la información de las vacunas	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar vacunas. 3. ver información	Obtener la información de las vacunas.	PASS	Crítica			
9	Actualizar la información de la vacuna.	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar vacunas. 3. Actualizar Vacuna.	Actualizar la información de la vacuna.	PASS	Crítica			
	Eliminar la información de la vacuna.	Sin precondiciones de momento	Desarrollo Backend	Eliminar la información relacionada a la vacuna.	PASS	Crítica			

Fuente: Autor

Figura 43: Lista de chequeo en Excel Tratamiento.

CUIDAME APP Treatment Module									
ID #	Título	Precondiciones	Pasos	Resultado Esperado	Status	Prioridad	Evidencia	Evidencia	Evidencia
9	Registrar la información de los tratamientos de la mascota.	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar Historial. 3. Agregar Vacuna y llenar el formulario.	Guardar la información del tratamiento.	PASS	Crítica			
9	Obtener la información de los tratamientos	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar Historial. 3. ver información	Obtener la información de los tratamientos.	PASS	Crítica			
9	Actualizar la información del tratamiento.	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar historial. 3. Actualizar tratamiento.	Actualizar la información del tratamiento.	PASS	Crítica			
	Eliminar la información del tratamiento.	Sin precondiciones de momento	Desarrollo Backend	Eliminar la información relacionada al tratamiento.	PASS	Crítica			

Fuente: Autor

Figura 44: Lista de chequeo en Excel Enfermedades.

CUIDAME APP Disease Module								
ID #	Título	Precondiciones	Pasos	Resultado Esperado	Status	Prioridad	Evidencia	Evidencia
	Registrar la información de enfermedad o condición de la mascota.	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar historial. 3. Agregar enfermedad o condición y llenar el formulario.	Guardar la información de la enfermedad o condición.	PASS	Crítica		
	Obtener la información de la enfermedad o condición	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar historial. 3. ver información	Obtener la información de la enfermedad o condición.	PASS	Crítica		
	Actualizar la información de la enfermedad o condición	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar historial. 3. Actualizar enfermedad o condición.	Actualizar la información de la enfermedad o condición.	PASS	Crítica		
	Eliminar la información de la enfermedad o condición	Sin precondiciones de momento	Desarrollo Backend	Eliminar la información relacionada a la enfermedad o condición.	PASS	Crítica		

Fuente: Autor

Figura 45: Lista de chequeo en Excel Veterinario.

CUIDAME APP Veterinaries Module								
ID #	Título	Precondiciones	Pasos	Resultado Esperado	Status	Prioridad	Evidencia	Evidencia
	Registrar la información de vacunas del veterinario.	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar veterinario. 3. Agregar del veterinario	Guardar la información del veterinario.	PASS	Crítica		
9	Obtener la información del veterinario.	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar veterinario. 3. ver información	Obtener la información del veterinario.	PASS	Crítica		
9	Actualizar la información del veterinario.	Estar logueado en el sistema.	1. Dentro de la información básica de la mascota. 2. Seleccionar veterinario. 3. Actualizar veterinario	Actualizar la información del veterinario.	PASS	Crítica		
	Eliminar la información del veterinario.	Sin precondiciones de momento	Desarrollo Backend	Eliminar la información relacionada al veterinario.	PASS	Crítica		

Fuente: Autor