



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE SANTO DOMINGO

Escuela de Hábitat, Infraestructura y Creatividad

APLICACIÓN WEB CON MACHINE LEARNING PARA LA GESTIÓN DE ADOPCIÓN DE
MASCOTAS EN EL CENTRO VETERINARIO MI MASCOTA DEL CANTÓN SANTO
DOMINGO

TRABAJO DE TITULACIÓN

Previo a la obtención del título de Ingeniero en Tecnologías de la Información

Línea de investigación: Tecnologías de la información y la comunicación

Autoría:

Ortiz Jaramillo Jaime Arturo

Zamora Menendez Nathaly Jaritza

Dirección:

Ocampo Pazos Willian Javier, Mg.

Santo Domingo – Ecuador
Febrero, 2026



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE SANTO DOMINGO

Escuela de Hábitat, Infraestructura y Creatividad

HOJA DE APROBACIÓN

**APLICACIÓN WEB CON MACHINE LEARNING PARA LA GESTIÓN DE ADOPCIÓN DE
MASCOTAS EN EL CENTRO VETERINARIO MI MASCOTA DEL CANTÓN SANTO
DOMINGO**

Línea de investigación: Tecnologías de la información y la comunicación

Autoría:

Ortiz Jaramillo Jaime Arturo

Zamora Menendez Nathaly Jaritza

Revisado por:

Ocampo Pazos Willian Javier, Mg.
DIRECTOR DEL TRABAJO DE
INTEGRACIÓN CURRICULAR

Ulloa Meneses Luis Javier, Mg.
CALIFICADOR

Córdova Gálvez Rodolfo Sirilo, Mg.
CALIFICADOR

Ulloa Meneses Luis Javier, Mg.
COORDINADOR DE LA CARRERA DE
SISTEMAS DE INFORMACIÓN

Santo Domingo – Ecuador
Febrero, 2026

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Nosotros, Ortiz Jaramillo Jaime Arturo, portador de la cédula de ciudadanía 0705442770, y Zamora Menendez Nathaly Jaritza, portadora de la cédula de ciudadanía 0804261055, declaramos que los resultados obtenidos en la investigación que presentamos como informe final, previo a la obtención del Título de Ingeniero en Tecnologías de la Información son absolutamente originales, auténticos y personales.

En tal virtud, declaramos que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de nuestra sola y exclusiva responsabilidad legal y académica.

Igualmente, declaramos que todo resultado académico que se desprenda de esta investigación y que se difunda tendrá como filiación la Pontificia Universidad Católica del Ecuador Sede Santo Domingo, reconociendo en las autorías al director del Trabajo de Titulación y demás profesores que amerita.

Además, declaro que el presente trabajo, producto de las actividades académicas y de investigación, forma parte del capital intelectual de la Pontificia Universidad Católica del Ecuador, Sede Santo Domingo, de acuerdo con lo establecido en el artículo 16, literal j), de la Ley Orgánica de Educación Superior.

En tal razón, autorizo a la Pontificia Universidad Católica del Ecuador, Sede Santo Domingo, para que pueda hacer uso, con fines netamente académicos, del Trabajo de Titulación, ya sea de forma impresa, digital y/o electrónica o por cualquier medio conocido o por conocerse, siendo el presente documento la constancia del consentimiento autorizado; y, para que sea ingresado al Sistema Nacional de Información de la Educación Superior del Ecuador para su conocimiento público, en cumplimiento del artículo 103 de la Ley Orgánica de Educación Superior.



Ortiz Jaramillo Jaime Arturo
C.C. 0705442770



Zamora Menendez Nathaly Jaritza
C.C. 0804261055

INFORME DE TRABAJO DE INTEGRACIÓN CURRICULAR ESCRITO

Mikel Ugando Peñate, PhD.

Responsable de Investigación Formativa

Pontificia Universidad Católica del Ecuador Sede Santo Domingo

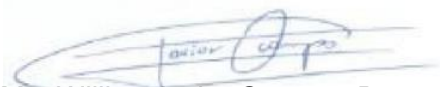
De mi consideración,

Por medio del presente informe en calidad de director del Trabajo de Titulación del Grado de Ingeniería en Tecnologías de la Información titulado: APLICACIÓN WEB CON MACHINE LEARNING PARA LA GESTIÓN DE ADOPCIÓN DE MASCOTAS EN EL CENTRO VETERINARIO MI MASCOTA DEL CANTÓN SANTO DOMINGO, realizado por los estudiantes: Ortiz Jaramillo Jaime Arturo con cédula de ciudadanía 0705442770 y Zamora Menendez Nathaly Jaritza con cédula de ciudadanía 0804261055, previo a la obtención del título de Ingeniero en Tecnologías de la Información, informo que el presente Trabajo de Integración Curricular escrito, se encuentra finalizado conforme a la guía y al formato de la Sede vigente.

Además, certifico haber verificado la originalidad y autenticidad del trabajo de titulación por medio del programa anti plagio Turnitin, en respuesta a la normativa institucional vigente.

Santo Domingo, 19/02/2026.

Atentamente,



Mg. Willian Javier Ocampo Pazos

Profesor Titular Auxiliar II

RESUMEN

La gestión de adopción de mascotas es un proceso de suma importancia para el bienestar animal, pero como cualquier proceso de gestión este enfrenta varios desafíos. En las encuestas aplicadas a los clientes se evidenció escasa información sobre el proceso de adopción, insuficiente visibilidad de animales disponibles para adopción. El enfoque que se adoptó para este proyecto de integración curricular fue cuantitativo con un diseño pre-experimental, se elaboró un muestreo censal que incluyó a los clientes con el fin de reunir datos sobre el proceso. Para enfrentar esta problemática, se propuso la implementación de una aplicación web con *machine learning*, con el objetivo de fortalecer la gestión de adopción de mascotas en el Centro Veterinario Mi Mascota del cantón Santo Domingo. Por consiguiente, en la propuesta de intervención se empleó herramientas como: el *framework frontend Vue.js*, el *frameworks backend Express.js*, el gestor de base de datos *PostgreSQL*, el patrón arquitectónico Modelo Vista Controlador (MVC), y, por último, la técnica de Filtrado Híbrido (*Hybrid Filtering*) por su capacidad para integrar métodos de filtrado, tales como el filtrado colaborativo y filtrado basado en contenido. Por otro lado, para validar la hipótesis, se utilizó regresión logística binaria.

Palabras clave: Aplicación web, Machine Learning, Adopción de Mascotas.

ABSTRACT

Pet adoption management is a crucial process for animal welfare, but like any management process, it faces several challenges. Surveys conducted with clients revealed a lack of information about the adoption process and insufficient visibility of animals available for adoption. The approach adopted for this curriculum integration project was quantitative with a pre-experimental design; a census sampling was developed that included the clients in order to gather data on the process. To address this problem, the implementation of a web application with machine learning was proposed, with the aim of strengthening the management of pet adoption at the Mi Mascota Veterinary Center in the Santo Domingo canton. Therefore, the proposed intervention used tools such as: the Vue.js frontend framework, the Express.js backend framework, the PostgreSQL database manager, the Model View Controller (MVC) architectural pattern, and, finally, the Hybrid Filtering technique for its ability to integrate filtering methods, such as collaborative filtering and content based filtering. On the other hand, binary logistic regression was used to validate the hypothesis.

Keywords: Web application, Machine Learning, Pet Adoption.

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN	10
1.1. Antecedentes	10
1.2. Planteamiento y delimitación del problema.....	12
1.3. Preguntas de investigación.....	14
1.3.1. Pregunta General	14
1.3.2. Preguntas Específicas	14
1.4. Justificación.....	14
1.5. Objetivos de investigación	16
1.5.1. Objetivo general	16
1.5.2. Objetivos específicos.....	16
2. REVISIÓN DE LA LITERATURA.....	17
2.1. Fundamentos teóricos.....	17
2.1.1. Aplicación Web.....	19
2.1.2. Machine Learning.....	23
2.1.3. Gestión de Adopción de Mascotas	29
2.2. Predicción científica.....	33
3. METODOLOGÍA	34
3.1. Enfoque, diseño y tipo de investigación.....	34
3.2. Unidades de análisis	35
3.3. Técnicas e instrumentos de investigación	36
3.4. Técnicas de análisis de datos.....	36
3.5. Operacionalización de las variables	38
4. RESULTADOS	41
4.1. Resultado del primer objetivo específico	41
4.1.1. Aplicación de las encuestas a los clientes	41

4.2.	Resultados del segundo objetivo específico	47
4.3.	Resultado del tercer objetivo específico	51
4.3.1.	Nomenclatura y logotipo de la aplicación web	51
4.3.2.	Marco de trabajo Scrum	52
4.3.3.	Sprint I.....	52
4.3.4.	Sprint 2.....	70
4.3.5.	Sprint 3.....	87
4.4.	Resultado general	51
4.4.1.	Implementación aplicación web con machine learning	113
4.4.2.	Validación de la propuesta de intervención.....	117
4.4.3.	Validación de la hipótesis.....	122
5.	DISCUSIÓN	125
6.	CONCLUSIONES Y RECOMENDACIONES	127
6.1.	Conclusiones.....	127
6.2.	Recomendaciones.....	128
7.	REFERENCIAS	130
8.	ANEXOS.....	140

1. INTRODUCCIÓN

Aunque en Ecuador no se hable mucho sobre el abandono de los animales, puesto que no se considera un tema de importancia, nunca realmente se ha llegado a magnificar el impacto que puede tener la fauna urbana en el país. Por lo tanto, según Ojeda et al. (2024), los principales problemas que vienen de la mano del abandono animal, son los impactos ambientales, sociales y de salud pública (p. 308).

Con lo citado anteriormente, el abandono de mascotas contribuye directamente a que crezca la problemática de la fauna urbana, esto genera que haya conflictos con comunidades humanas y dañando la biodiversidad local. Según Echeverría y Moncada (2024), elementos como falta de educación ambiental y la percepción negativa hacia los animales, es lo que conlleva a que se presente la problemática del abandono de mascotas. (p. 91). Por consiguiente, se sabe que las tecnologías digitales emergentes juegan un papel clave para revolucionar la gestión de centros de adopción.

1.1. Antecedentes

En la actualidad, la promoción de prácticas para la adopción responsable de mascotas ha cobrado un impulso significativo en los últimos años, lo que ha logrado promover la elaboración de mecanismos que permitan gestionar estos procesos. Los avances tecnológicos han sido una parte fundamental en la transformación de cómo los usuarios interactúan con los servicios relacionados con la adopción de mascotas.

En este contexto, de acuerdo con Li et al. (2024) en China, se implementó un algoritmo de *Machine Learning* que permitió crear un sistema de recomendación inteligente para mascotas, con el objetivo de mejorar la interacción que tiene el usuario con las aplicaciones de adopción. Para lograrlo, el sistema fue desarrollado en *Java* con el apoyo de los *frameworks* como *Vue* y *Spring Boot* y la incorporación de un algoritmo de recomendación basado en el aprendizaje automático. La implementación de este sistema

permitió dar sugerencias más acertadas a cada usuario de acuerdo a los intereses que tenían para adoptar una mascota (p. 535).

De igual manera, se abordó otro problema tratado por Alsuwailem et al. (2022) en Arabia Saudita, donde desarrollaron una plataforma *web* para poder gestionar los procesos de adopción de mascotas. En este sentido, se desarrolló una plataforma que ofrece servicios gratuitos de adopción, cuidado de mascotas, donación y búsqueda de clínicas de mascotas. Para la implementación de la plataforma en el lado del cliente se utilizaron herramientas como *HTML*, *CSS*, *JS*, *Bootstrap3* y *Sweetalert2*, mientras que para el lado del servidor se utilizó *Python*, *Flask*, *Twilio*, *phpMyAdmin* y *Xampp*. Como resultado, se promovió la adopción gratuita de mascotas, lo que generó a largo plazo un aporte positivo en la sociedad (p. 775).

Por otra parte, en la investigación de Pérez y Nisperuza (2019) en Colombia, propusieron la elaboración de una aplicación móvil compatible con *Android*, con el propósito de incentivar la adopción de mascotas uniendo a personas que deseen publicar o adoptar. La aplicación tiene como función principal el poder mostrar mediante anuncios virtuales las publicaciones de mascotas en adopción. Se desarrolló con el lenguaje *Java* y utilizó el *framework Spring* para el desarrollo de una *API Rest* y, a su vez, se integró una base de datos relacional en *PostgreSQL*. Además, el estudio analizó proyectos de otros países con especificaciones similares para poder identificar componentes claves como son la funcionalidad, navegabilidad, usabilidad y experiencia al usuario. La implementación de este sistema permitió reducir problemas que surgían durante los procesos de adopción, lo que hizo posible evitar trámites largos al momento de elegir una mascota (pp. 78,112, 114).

Siguiendo la línea de soluciones, López (2022) desarrolló en Quito una aplicación llamada "Patitas Felices", con el fin de fomentar la tenencia responsable de mascotas, buscando abordar el problema del abandono. La aplicación presenta registro de mascotas, filtro para buscar por características deseadas y un apartado dedicado a la comunidad. Aparte, se utilizó SCRUM y la arquitectura MVVM. Los resultados mostraron una gran

acogida por los usuarios, pues incentiva la formación de una comunidad que se dedica a los animales, dando como consecuencia un sistema para fomentar la tenencia responsable de mascotas (pp. 6,11,18).

1.2. Planteamiento y delimitación del problema

A partir de los antecedentes revisados, se evidencia que la implementación de sistemas tecnológicos en la optimización de procesos de adopción de animales ha tenido un impacto positivo. Los cuatro estudios analizados, se centran en la implementación de plataformas digitales orientadas a fomentar una adopción responsable. Cabe destacar que uno de los artículos incorpora técnicas de *Machine Learning* como mecanismo para facilitar la selección de una mascota a partir de ciertos intereses.

De acuerdo con Zhou (2023) en Estados Unidos, en su artículo hace mención que en los refugios la tasa de rescate de animales sin hogar es aproximadamente del 30%, esto implica que el resto de animales son en su gran mayoría sacrificados. Actualmente, el número de mascotas que esperan ser adoptadas ha superado el número de personas que se encuentran dispuestas a realizar el procedimiento de adopción (p. 42). Por otro lado, Convington (2022), que fue citado por Zhou (2023) indica que, en 2021, el 65% de dueños de mascotas optaron por adoptar. Sin embargo, alrededor del 31% de gatos y el 28% de perros han sido rescatados cada año (p. 42).

En esta misma línea, el Congreso de la Ciudad de México (2021) menciona que, al año, son abandonados cerca de 500.000 perros y gatos, lo que se convierte en una modalidad de violencia y se transforma en omisión de cuidados básicos, falta de resguardo y atención veterinaria. Pese a ser una de las formas de maltrato animal más recurrentes, suele pasar desapercibida, debido a un proceso de normalización en la sociedad. Tan solo en Ciudad de México, son condenados a morir sacrificados alrededor de 180.000 animales; se estima que alrededor de 9 de cada 10 perros que son ingresados en refugios de animales terminan siendo sacrificados (pp. 2-3).

Adicionalmente, de acuerdo con la investigación elaborada por Vinueza et al. (2022) en Ecuador, mencionan que los refugios se han convertido en sitios casi permanentes para los animales, lo que conlleva que estos lugares se encuentren colapsados y no puedan dar abasto a más miembros. De acuerdo al estudio realizado en cuatro refugios de animales, de una muestra de 207 perros en espera de ser adoptados, se identificó que los caninos con mayor número de adopciones eran machos (52.2%), en donde la gran mayoría eran cachorros (57.1%) (pp. 43-44).

Además, Vinueza et al. (2022) menciona que los días donde permanecieron los perros en los refugios hasta lograr ser adoptados variaron de acuerdo al rango de edad, en donde la mediana de días de adopción para los perros juveniles era de 36, cachorros de 92 y adultos de 354. Por otro lado, los días de espera fueron mayores en el caso de las hembras (200 días), a diferencia de los machos (188 días). En el caso de perros de raza, el sexo no fue un factor determinante en lo que respecta para ser adoptados, mientras que para perros mestizos sí, en donde la probabilidad de adopción de las hembras era baja en comparación con la de los machos (pp. 47-48).

Por lo tanto, en el Centro Veterinario Mi Mascota, ubicado en la ciudad de Santo Domingo, Ecuador, se llevó a cabo una visita que permitió analizar y constatar los trámites relacionados con los procesos de adopción. Principalmente, se observó que existe una escasa visibilidad de los animales que se encuentran disponibles para adopción, lo que ocasiona que las mascotas pasen desapercibidas para los posibles adoptantes. Además, no disponen de una plataforma *web* que permita la gestión de adopciones, lo que provoca procesos lentos y desorganizados para concretar las adopciones. De igual forma, no cuentan con una infraestructura para mantener mascotas recibidas por clientes, lo que implica insatisfacción por parte de los clientes al no poder dar en adopción.

1.3. Preguntas de investigación

1.3.1. Pregunta General

¿Cómo se puede fortalecer la gestión de adopción de mascotas en el Centro Veterinario “Mi Mascota” en el cantón Santo Domingo?

1.3.2. Preguntas Específicas

- ¿Cuáles son los procesos actuales en el Centro Veterinario en relación con la gestión de adopción de mascotas?
- ¿Qué herramientas y tecnologías son necesarias para el desarrollo e implementación de la propuesta realizada al Centro Veterinario?
- ¿Qué solución se puede implementar para el fortalecimiento en la gestión de adopción en el Centro Veterinario “Mi Mascota” de Santo Domingo?

1.4. Justificación

El presente trabajo de titulación, se justifica según lo establecido por la Asamblea Constituyente (2008) en el numeral 2, del artículo 16 de la Constitución de la República del Ecuador, en donde todas las personas tienen el libre derecho al acceso universal de las TIC (p. 14). Este estatuto constitucional forma las bases para que el desarrollo de este proyecto integre tecnología que permita resolver el problema planteado. En este escenario, la implementación de una aplicación *web* no solo responde a las nuevas tecnologías, si no que se relaciona con el mandato de garantizar un acceso igualitario a las herramientas digitales. Así el proyecto busca aprovechar el potencial de estas plataformas digitales para reinventar los procesos, fortalecer la interacción y sobre todo fomentar la participación ciudadana.

De acuerdo con la Secretaría Nacional de Planificación (2024), en el Objetivo 9 del Plan de Desarrollo para el nuevo Ecuador 2024-2025 se establece la necesidad de preferir la construcción de un estado eficiente, transparente y que tenga como objetivo el bienestar social. En este marco, el uso de tecnologías se vuelve un pilar fundamental para alcanzar

estos objetivos (p. 154). El uso de una aplicación *web* y el uso de *Machine Learning* permiten fortalecer procesos, garantizar transparencias en las gestiones y mejoras de calidad en servicios digitales que son ofrecidos a los ciudadanos. Es decir, busca impulsar la innovación de infraestructuras digitales y se facilita una interacción más eficiente entre sociedad y plataformas tecnológicas.

La Asamblea Nacional de la República del Ecuador (2023), por medio del artículo 1 de la Ley Orgánica para la Transformación Digital y Audiovisual, promueve la adopción de tecnologías digitales como medio para fortalecer el desarrollo económico digital en Ecuador. La presente Ley busca mejorar los servicios ofrecidos a través de entidades públicas y privadas, promoviendo la inversión e innovación con el objetivo de potenciar áreas importantes como la educación, la salud y el empleo a través de la implementación de servicios digitales. Así mismo, menciona el fortalecimiento del ciberespacio, mediante el cual se garantiza la protección de los datos personales de los ciudadanos (p. 4-5). Es crucial considerar la implementación de herramientas tecnológicas como un eje para fortalecer servicios de acuerdo con las necesidades de la sociedad.

Por otro lado, por medio del Reglamento al Código Orgánico del Ambiente emitido por la Presidencia de la República del Ecuador (2019), en el artículo 387, estipuló que los establecimientos dedicados a los animales deberán cumplir con los requisitos pertinentes dispuestos por los Gobiernos Autónomos Descentralizados Municipales y Metropolitanos con el fin de garantizar el bienestar animal. Además, mediante el artículo 388, se estableció que los centros para animales deberán llevar un registro de acuerdo al número de animales, especies e información adicional que será determinada por el GAD municipal o metropolitano en el cantón (pp. 78-79). Esto implica tomar medidas que permitan fortalecer tanto la infraestructura como la gestión de datos y recursos en centros para animales, asegurando condiciones adecuadas y óptimas para cada especie.

Por último, de acuerdo a la ordenanza que regula la protección, tenencia y control de la fauna urbana establecida por el Gobierno Autónomo Descentralizado Municipal de

Guayaquil (2023), se menciona en el Artículo 8 las obligaciones que deben tener los establecimientos para animales. Esto incluye contar con documentos que permitan llevar un registro sobre las características de los animales e implementar mecanismo que permitan promover la adopción de animales de compañía en los centros (pp. 16-18). Estas acciones evidencian la importancia de promover una adopción responsable hacia la sociedad, identificando las mejores formas que permitan visibilizar el propósito planteado.

1.5. Objetivos de investigación

1.5.1. Objetivo general

Implementar una Aplicación *Web* con *Machine Learning* para el fortalecimiento de la gestión de adopción de mascotas en el Centro Veterinario “Mi Mascota” del cantón Santo Domingo durante el año 2025.

1.5.2. Objetivos específicos

- Identificar los procesos en el Centro Veterinario con respecto a la gestión de adopción de mascotas.
- Determinar las herramientas y tecnologías necesarias para el desarrollo de la propuesta.
- Desarrollar una plataforma *web* para la gestión de adopción de mascotas en el Centro Veterinario “Mi Mascota” en el cantón Santo Domingo.

2. REVISIÓN DE LA LITERATURA

2.1. Fundamentos teóricos

Para la revisión de la literatura, se elaboró un diagrama con la estructura de cada variable, según Fink (2019), se debe realizar una extracción de la información relevante del tema de interés. La información recopilada es una guía para desarrollar el fundamento teórico abordando cada uno de los contenidos especificados (p. 6). Por medio de la figura, se encuentra definida la variable *Aplicación Web*, la figura 2 desarrolla la explicación de la variable *Machine Learning* y la figura 3 aborda la Gestión de Adopción de Mascotas.

Figura 1. Índice de la variable independiente *Aplicación Web*

Fases	Apartado	Sub apartado
<i>Aplicación Web</i>	<i>Herramientas de desarrollo</i>	<ul style="list-style-type: none"> <i>Lenguajes de Programación</i> <i>Frameworks</i> <i>Editor de código fuente</i>
	<i>Bases de Datos</i>	<ul style="list-style-type: none"> <i>Modelado de datos</i> <i>Lenguaje de base de datos</i> <i>Tipos de base de datos</i> <i>SGBD</i>
	<i>Seguridad web</i>	<ul style="list-style-type: none"> <i>Factor de autenticación</i> <i>Encriptación de datos</i> <i>Protocolos de seguridad</i>

Figura 2. Índice de la variable independiente Machine Learning

Fases	Apartado	Sub apartado
Machine Learning	Métodos de Machine Learning	<ul style="list-style-type: none"> Aprendizaje Supervisado Aprendizaje No Supervisado Aprendizaje Semisupervisado Aprendizaje de Refuerzo
	Principales Algoritmos de Aprendizaje	<ul style="list-style-type: none"> Árboles de decisión Naive Bayes Regresión logística Regresión lineal K-means Clustering Random Forests
	Canalización de Machine Learning	<ul style="list-style-type: none"> Recopilación de datos Preprocesamiento de datos Ingeniería de características División de datos y selección de modelo Entrenamiento del modelo Evaluación del modelo Despliegue y mantenimiento

Figura 3. Índice de la variable dependiente Gestión de Adopción de Mascotas

Fases	Apartado	Sub apartado
Gestión de Adopción de Mascotas	Procesos de adopción	<ul style="list-style-type: none"> Selección Solicitud Entrevista y aprobación
	Registro médico veterinario	<ul style="list-style-type: none"> Información del paciente y cliente Motivo de consulta Historial médico Examen Físico
	Gestión organizacional	<ul style="list-style-type: none"> Políticas y protocolos internos Capacitación del personal Infraestructura y bienestar animal
	Promoción y sostenibilidad en adopciones	<ul style="list-style-type: none"> Campañas de adopción Donaciones Voluntariados

2.1.1. Aplicación Web

Moscato (2024) menciona que, es una aplicación de *software* que se ejecuta dentro de un navegador *web*, tienen como propósito conceder acceso a las capacidades del *software* y a los datos a través de una red mediante tecnologías *web* estándar, permitiendo a los usuarios acceder a las funcionalidades desde cualquier dispositivo. Además, las aplicaciones *web* son útiles cuando se desea proporcionar acceso a *software* y datos a una gran cantidad de usuarios especialmente dentro de una organización (p. 5).

2.1.1.1. Herramientas de desarrollo

De acuerdo con Cybellium (2024), las herramientas de desarrollo ofrecen una amplia gama de utilidades y plataformas esenciales que asisten a los desarrolladores en la creación, depuración y administración de aplicaciones de *software*. Comprende herramientas como sistemas de control de versiones, entornos de desarrollo integrados (*IDE*), sistemas de control de versiones, analizadores de código y depuradores. La implementación de estas herramientas mejora significativamente la productividad, facilita el trabajo colaborativo, ayuda a mitigar errores y asegura la aplicación de mejores prácticas en todo el ciclo de desarrollo (p. 16).

2.1.1.1.1. Lenguajes de Programación

Manglik (2024c), define un lenguaje de programación como un conjunto de comandos, instrucciones y sintaxis diseñado para crear *software*. Estos se clasifican principalmente en dos tipos: los lenguajes de alto nivel, como *C++*, *Java*, *Perl* y *PHP*, que son fáciles de leer y comprender por los humanos; y los lenguajes de bajo nivel, que incluyen el lenguaje ensamblador con sus instrucciones básicas de difícil lectura, y el lenguaje máquina, compuesto por series de códigos binarios, siendo el más cercano a la comunicación directa con el *hardware* de la computadora (p. 3).

2.1.1.1.2. Frameworks

Byri et al. (2024) indica que, el uso de *frameworks* se ha vuelto indispensable dentro del desarrollo de *software*, lo que, a su vez, enfatiza la necesidad de herramientas robustas que agilicen y permitan construir interfaces modernas y sistemas eficientes. Dominar el uso

de *frameworks* dentro del desarrollo de aplicaciones se ha convertido en una gran ventaja, ya que permite a los desarrolladores centrarse en la implementación lógica empresarial y objetivos del proyecto sin preocuparse de detalles técnicos de bajo nivel (p. 33).

2.1.1.1.3. Editor de Código Fuente

De acuerdo con Vázquez & Lorenzo (2020), un editor de código fuente es un programa el cual permite escribir y editar código de programación. Los editores de código ofrecen funcionalidades adicionales a través de la instalación de paquetes que permiten personalizar el entorno de trabajo. Adicionalmente, una de las características principales de estas herramientas es el resaltado de la sintaxis, ya que muestra diferentes partes del código en colores o tipografía distinta para mejorar la legibilidad (p. 20).

2.1.1.2. Base de Datos

Según Valderrey (2020), una base de datos es un conjunto de datos interrelacionados y estructurados que se encuentran organizados, lo que permite acceder a ellos de forma automática, independientemente de los programas que los utilizan. Esta independencia es un punto clave, ya que permite modificar la estructura interna de los datos sin necesidad de modificar las aplicaciones que los manipulan. Las bases de datos se encuentran conformadas por uno o varios bloques de información denominada tablas, las cuales agrupan datos con características relacionadas (p. 22).

2.1.1.2.1. Modelado de datos

Como explican Sánchez y Mosquera (2020), el modelado de datos está compuesto por un grupo de herramientas conceptuales que facilitan la representación de los objetos de la realidad. Los clasifica en dos tipos principales de modelos: registros y objetos. Los modelos basados en objetos se centran en un diseño conceptual, el cual describe la realidad, que se ajusta de acuerdo a las especificaciones y necesidades que se quieren presentar. Por lo tanto, este modelo permite representar los datos tal cual se captan en el mundo real; entre los más comunes se encuentran el modelo orientado a objetos y modelo entidad-relación. Por otra parte, el modelo basado en registros toma el diseño conceptual de la realidad y lo transforma en un diseño lógico, especificando la estructura lógica general

y el cómo se almacenará y accederá a los datos. Dentro de este modelo encontramos el modelo relacional, modelo jerárquico y modelo reticular (pp. 36-37).

2.1.1.2.2. Lenguaje de base de datos

Manglik (2024a) menciona el lenguaje *SQL* y lo define como un estándar universal diseñado para acceder y manipular bases de datos. *SQL* puede ser utilizado para ejecutar consultas, recuperar, insertar, actualizar y eliminar datos. Por su capacidad para interactuar directamente con la información almacenada. *SQL* es fundamental en desarrollo de aplicaciones *web* que necesitan mostrar y administrar la información dentro del sistema de gestión de datos (p. 80).

Sin embargo, Sarasa (2019) señala que, una de las características claves en las bases de datos *NoSQL* es que el lenguaje de consultas que utilizan para recupera información no es *SQL*. En muchos casos, los sistemas *NoSQL* emplean lenguajes adaptados específicamente a la manera en cómo se organizada la información. No obstante, existen base de datos *NoSQL* que conservan algunas características de *SQL*, como en el caso de *Cassandra* que cuenta con su propio lenguaje de consulta *CQL*, el cual comparte una sintaxis y un enfoque conceptual que resulta muy similar a las bases de datos relacionales (p. 25).

2.1.1.2.3. Tipos de base de datos

Según los expuesto por Ortega (2023), dentro de las clasificaciones de base de datos, se encuentra la de tipo relacional. Estos sistemas se definen por almacenar los datos dentro de tablas, que se distribuyen y organizan a través de la intersección de filas y columnas. Cada tabla está diseñada para representar un tipo específico de entidad y tiene un esquema fijo definido por campos con tipos de datos específicos. Si bien suelen ser muy eficaces pueden tener dificultades al intentar escalar horizontalmente, ya que al incrementar más servidores se requieren configuraciones especiales. Por otro lado, tenemos las bases de datos *NoSQL* las cuales no emplean tablas para el almacenamiento de datos, si no que utilizan modelos como: clave-valor, grafos y documentos. Se caracteriza por su escalado horizontal que permite añadir más servidores de acuerdo a la capacidad que se desea

umentar, lo que permite que puedan evolucionar y adaptarse fácilmente a los cambios y necesidades de las aplicaciones, sin la necesidad de realizar migraciones complejas (pp. 63-64).

2.1.1.2.4. SGBD

Como explican Sánchez y Mosquera (2020), un sistema de gestión de bases de datos es un *software* diseñado para supervisar el ciclo de vida de las bases de datos. Este *software* consta de numerosas rutinas que automatizan las operaciones dentro de las bases de datos, facilitan la creación, acceso y administración. Una de las ventajas de un SGBD es su capacidad de permitirle al usuario poder interactuar con la base de datos sin tener conocimientos avanzados sobre ellas, permite visualizar la estructura de los datos almacenados a través de una interfaz intuitiva de forma rápida y eficiente (pp. 24-25).

2.1.1.3. Seguridad web

Según Karunamurthy (2024), la seguridad *web* comprende una serie de prácticas y tecnologías diseñadas para proteger sitios, aplicaciones y servicios *web* contra amenazas y vulnerabilidades, buscando asegurar la disponibilidad, la integridad y la confidencialidad de los recursos en línea. La autenticación y autorización son características claves dentro de la seguridad *web*, establece una barrera esencial contra accesos no deseados o malintencionados (p. 39).

2.1.1.3.1. Factor de autenticación

Vega (2021) indica que, existen diversos métodos de autenticación, cada uno clasificado como un factor. Estos factores pueden basarse en atributos físicos únicos como huellas dactilares, patrones de iris o características faciales; en conocimiento (algo que el usuario recuerda), como contraseñas o *PIN*; en la posesión física de un objeto, como tarjetas bancarias o *tokens* de seguridad; o en acciones o comportamientos, como patrones de escritura o pulsaciones de teclas (p. 28-30).

2.1.1.3.2. Encriptación de contraseñas

De acuerdo a Barker (2025), las contraseñas pueden ser vulneradas de diferentes maneras ya sea mediante un *malware*, manipulación para compartirla conocida como *phishing* o aplicando intentos de fuerza bruta hasta dar con la contraseña. Para contrarrestar estas amenazas los sitios *web* almacenan las contraseñas de forma segura utilizando técnicas de criptografía, no almacenan las contraseñas tal cual se escriben si no que las convierten en una larga cadena de números mediante una función matemática unidireccional llamado algoritmo *hash*. Cuando se ingresa la contraseña para iniciar sesión dentro de un sitio *web*, el sistema aplica el mismo algoritmo *hash* a la contraseña ingresada y la compara con la cadena almacena al crear la cuenta (p. 52).

2.1.1.3.3. Protocolos de seguridad

Pérez (2024) lo define como un conjunto de reglas y procedimientos que dictan cómo debe llevarse a cabo la comunicación segura, empleando sofisticadas técnicas criptográficas para lograr objetivos críticos como la autenticación de dispositivos y usuarios, la integridad de los datos y la confidencialidad de la información. Protocolos como *SSL/TLS* se encargan de establecer un canal cifrado y seguro entre un cliente y un servidor, garantizando que cualquier dato intercambiado entre ellos permanezca privado y protegido contra escuchas o interceptaciones. *HTTPS* es el protocolo estándar para la navegación *web*, se encarga específicamente de superponer el cifrado *SSL/TLS* sobre la comunicación *HTTP* estándar, cifrando así toda la sesión entre el navegador y el sitio *web* que se visita, garantizando así la confidencialidad de la información (pp. 222-223).

2.1.2. Machine Learning

Joshi (2022) indica que, el aprendizaje automático reúne una amplia gama de algoritmos surgidos inicialmente de campos como la ingeniería eléctrica, estadística, análisis financiero e incluso las ciencias genéticas. Los componentes fundamentales de todos estos algoritmos se basan profundamente en los principios de las matemáticas puras y la estadística. Las computadoras cumplen un papel importante en la automatización de cálculos complejos. Esta automatización permite el procesamiento iterativo, el ajuste de

parámetros y el reconocimiento de patrones, el efecto acumulativo de estos cálculos permite que los sistemas de aprendizaje automático aborden problemas complejos (p. 5).

2.1.2.1. Métodos de Machine Learning

2.1.2.1.1. Aprendizaje Supervisado

Acquaviva (2023) menciona que, en el aprendizaje supervisado se asume la existencia de un conjunto de aprendizaje, conocido como conjunto de entrenamiento. Cada instancia de datos dentro de este conjunto de datos es distinta, se conoce el resultado deseado o propiedad objetivo de cada una. La idea principal del aprendizaje supervisado gira en torno al concepto de aprender a partir de ejemplos. El algoritmo recibe estos ejemplos etiquetados durante su fase de entrenamiento, con el fin de analizar las características de entrada de cada instancia y conocer los patrones, relaciones o reglas que conectan con las propiedades objetivo-conocidas (p. 4).

2.1.2.1.2. Aprendizaje No Supervisado

Acquaviva (2023) señala que, a diferencia del aprendizaje supervisado, las tareas en el aprendizaje no supervisado operan sin la ventaja de ejemplos etiquetados. Esta diferencia indica que no existe una respuesta correcta predefinida ni una propiedad objetivo con la que el algoritmo pueda aprender. Su objetivo principal es descubrir patrones, estructura y relaciones ocultas en los propios datos no etiquetados. El aprendizaje no supervisado es clave para el análisis de datos y detección de anomalías mediante la identificación de patrones inusuales (p. 10).

2.1.2.1.3. Aprendizaje Semisupervisado

Ramana et al. (2024) indica que, el aprendizaje semisupervisado conecta los métodos de aprendizaje supervisado y no supervisado. Utiliza una pequeña cantidad de datos etiquetados difíciles de obtener y un conjunto grande de datos no etiquetados fácilmente disponibles. El aprendizaje semisupervisado aprovecha la poca información etiquetada para guiar el proceso de aprendizaje sobre los datos no etiquetados. Al combinar la guía estructurada de los ejemplos etiquetados con la información oculta en las instancias

no etiquetadas, el aprendizaje semisupervisado puede mejorar en gran medida el rendimiento y la generalización del modelo (p. 3).

2.1.2.1.4. Aprendizaje de Refuerzo

De acuerdo a Dong et al. (2020), posee potentes capacidades de representación de datos, sin embargo, no es suficiente para construir un sistema de inteligencia artificial. El aprendizaje por refuerzo es un subconjunto del aprendizaje automático que permite a las computadoras interactuar con el mundo real, logrando aprender a través de la experiencia. Este modelo de aprendizaje conceptualiza el mundo real en dos componentes que interactúan: un entorno y un agente, en donde el agente interactúa con el entorno ejecutando acciones específicas, en respuesta el entorno proporciona retroalimentación, denominado como recompensa (p. 26).

2.1.2.2. Principales Algoritmos de Aprendizaje

2.1.2.2.1. Árboles de decisión

Chandra y Hareendran (2021) menciona que, ofrece un enfoque de clasificación relativamente simple, pero muy intuitiva y eficaz, este algoritmo imita un diagrama de flujo de preguntas y respuestas. A partir de la raíz el algoritmo recorre a través de cada rama de acuerdo a las respuestas de una serie de preguntas, siendo el final un nodo denominado "hoja". Cada nodo hoja representa la clasificación final para esa instancia de datos en particular (p. 31).

2.1.2.2.2. Naive Bayes

De acuerdo a Chopra (2018), es un algoritmo fundamental de aprendizaje automático supervisado, basado en la probabilidad, que aplica directamente el teorema de Bayes. Determina la probabilidad de que una entrada dada sea parte de una clase específica combinando la probabilidad previa de dicha clase con la probabilidad de observar las características de la entrada dentro de dicha clase, asumiendo independencia de las características (p. 61).

2.1.2.2.3. Regresión logística

Shankar y Spanias (2022) indica que, su propósito principal es predecir resultados discretos a partir de un conjunto de variables de entrada. Modela resultados binomiales, lo que significa que está diseñada para generar un valor de probabilidad de que una instancia pertenezca a una de dos o más clases discretas (p. 13).

2.1.2.2.4. Regresión lineal

Shankar y Spanias (2022) menciona que, la regresión lineal se rige como un modelo fundamental y ampliamente utilizado en el aprendizaje automático, utiliza una técnica puramente estadística con el fin de aprender una función de mapeo que cuantifique la relación entre la variable independientes y una variable dependiente. La regresión lineal proporciona una ecuación clara e interpretable que se utiliza para predecir el valor de una variable dependiente utilizando los valores proporcionados por las variables independientes. (p. 11).

2.1.2.2.5. K-means Clustering

De acuerdo a Manglik (2024b), el método *K-means* destaca como uno de los algoritmos más sencillos y utilizados para la agrupación en clústeres en aprendizaje automático no supervisado. Se trata de un método iterativo diseñado para dividir un conjunto de datos en un número predefinido de clústeres distintos y no superpuestos. El algoritmo comienza realizando una estimación inicial de los centros de los clústeres, luego asigna varias veces cada dato individual a su centro más próximo y, posteriormente, se actualiza la posición de dichos centros, determinando su nueva ubicación a partir de calcular la media de todos los puntos que conforman el nuevo grupo (p. 144).

2.1.2.2.6. Random Forests

Balamanigandan et al. (2024) menciona que, si bien los árboles de decisión individuales son fáciles de comprender, sus limitaciones hacen que un solo árbol sea insuficiente para obtener hallazgos robustos, lo que lleva a los científicos de datos a preferir los bosques aleatorios. Este potente método de conjunto construye un "bosque" de numerosos árboles de decisión, cada uno generado al azar, donde se asigna a cada árbol

un lote de entrenamiento único generado aleatoriamente. Este principio de aleatoriedad se aplica de igual forma para cada nodo basándose únicamente en un subgrupo de características para la división. Al combinar las predicciones de esta variedad de árboles, los bosques aleatorios logran una precisión, estabilidad y generalización significativamente mejores en comparación con cualquier árbol de decisión individual (p. 35).

2.1.2.3. Canalización de Machine Learning

2.1.2.3.1. Recopilación de datos

Bhasin (2023) explica que, la recopilación de datos es la etapa inicial para el desarrollo de modelos de aprendizaje automático. Según los requisitos del modelo, los datos pueden ser estructurados, no estructurados o en tiempo real. Por lo tanto, es fundamental comprender bien las diversas metodologías de recopilación de datos antes de iniciar en cualquier proyecto de aprendizaje automático. (p. 147).

2.1.2.3.2. Preprocesamiento de datos

De acuerdo a Hapke y Nelson (2020), antes de entrenar eficazmente un modelo de aprendizaje automático, es necesario realizar el preprocesamiento de datos para usarlos en sus ejecuciones de entrenamiento, esto implica transformar los datos sin procesar a un formato adecuado para el modelo. Dado que este preprocesamiento es una operación única que se realiza antes de que comience el entrenamiento del modelo, en lugar de en cada periodo de entrenamiento, resulta más práctico ejecutarlo como un paso específico e independiente en el flujo de trabajo de aprendizaje automático (p. 5).

2.1.2.3.3. Ingeniería de características

Bhasin (2023) detalla que, no todas las características de un conjunto de datos contribuyen por igual al rendimiento de un modelo de aprendizaje automático. Algunas características pueden tener un efecto negativo, mientras que otras son significativamente más influyentes. Esto requiere la selección de características, el cual es un proceso centrado en identificar y elegir las más importantes para el modelo (pp. 147-148).

2.1.2.3.4. División de datos y selección de modelo

Bhasin (2023) indica que, una vez preparados los datos, deben dividirse cuidadosamente en conjuntos de entrenamiento, validación y prueba. El primer conjunto es utilizado para entrenar al modelo, la evaluación se realiza utilizando el conjunto de validación y las pruebas con el conjunto de los datos no vistos proporciona una evaluación final sobre el rendimiento del modelo. Tras esta división crucial, se elegirá un modelo según las particularidades de su problema y el rendimiento deseado. Finalmente, comprender y gestionar la varianza del modelo es vital para garantizar que este se generalice correctamente y funcione de forma óptima con nuevos datos (p. 148).

2.1.2.3.5. Entrenamiento del modelo

Para Hapke y Nelson (2020), el entrenamiento de modelos representa la actividad central en el flujo de trabajo del aprendizaje automático. En este paso, se entrena un modelo para tomar entradas y predecir una salida con el menor error posible. Sin embargo, a medida que los modelos aumentan en complejidad y los conjuntos de datos de entrenamiento se vuelven excepcionalmente grandes, este paso crítico puede rápidamente volverse exigente y consumir muchos recursos de un computador. Dado que la memoria es un recurso finito, la asignación y gestión eficientes de recursos durante el entrenamiento de modelos son cruciales para garantizar la viabilidad y el éxito de todo el proceso (p. 5).

2.1.2.3.6. Evaluación del modelo

Bhasin (2023) explica que, tras el entrenamiento, el rendimiento del modelo se evalúa críticamente mediante el conjunto de validación, empleando diversas métricas como exactitud, precisión, recuperación, puntuación $F1$ o error cuadrático medio, según el tipo de problema. Estas métricas ofrecen una comprensión completa de la eficacia del modelo y orientan las mejoras futuras (p. 148).

2.1.2.3.7. Despliegue y mantenimiento

De acuerdo a Hapke y Nelson (2020), una vez que un modelo de aprendizaje automático se ha sometido a un entrenamiento, ajuste y análisis exhaustivos, está listo para su aplicación práctica. La implementación es el proceso de poner el modelo a disposición

para su uso. Los servidores de modelos modernos simplifican enormemente este proceso, eliminando a menudo la necesidad de un desarrollo exhaustivo de aplicaciones *web*. Estos servidores suelen ofrecer diversas *API*, como protocolos de Transferencia de Estado Representacional (*REST*) o Llamadas a Procedimientos Remotos (*RPC*), y permiten alojar varias versiones del mismo modelo simultáneamente. Tras la implementación, la monitorización continua del rendimiento del modelo en el entorno de producción es crucial. Esta observación permite detectar degradaciones del rendimiento o cambios en la distribución de datos (p. 6).

2.1.3. Gestión de Adopción de Mascotas

Según la información de Association of Shelter Veterinarians (2022), las organizaciones dedicadas a la adopción de animales deben implementar prácticas de gestión alineadas a sus directrices. La identificación y mantenimiento de registro de animales son un pilar fundamental dentro de las organizaciones. Estos elementos no solo son esenciales para una operación eficaz, sino que también deben cumplir con todos los requisitos regulatorios establecidos. La integración de sistemas digitales para el mantenimiento de registros ayuda a llevar un control administrativo eficiente y organizado, ya que facilita los procesos de gestión de adopciones dentro de las organizaciones (pp. 5-6).

2.1.3.1. Procesos de adopción

2.1.3.1.1. Selección

De acuerdo a Domínguez y Rivas (2021), es fundamental encontrar la mascota ideal para la persona adecuada. Cada fundación cuenta con personal dedicado para guiar a la persona en el proceso de selección y garantizar que esté listo para comprometerse a brindarle cuidados de por vida a su nuevo compañero (p. 14).

2.1.3.1.2. Solicitud

Secretaría Metropolitana de Salud (2020) indica que, una vez que los posibles adoptantes seleccionan un animal del folleto y manifiestan interés, llenan una solicitud de adopción. Este documento crucial está diseñado para evaluar su capacidad para

proporcionar un hogar adecuado a una nueva mascota, recalcando la importante de tener estabilidad financiera y emocional que se requiere, junto con una verdadera conciencia de la tenencia responsable de mascotas (pp. 7-8).

2.1.3.1.3. Entrevista y Aprobación

Secretaría Metropolitana de Salud (2020) señala que, la entrevista es un paso crucial para corroborar la información proporcionada en la solicitud de adopción inicial, garantizando su precisión. Además, es una oportunidad para educar a los futuros tutores sobre el bienestar responsable. Este abarca toda la dinámica, desde el rescate inicial de animales en estado crítico, pasando por su rehabilitación y, finalmente, hasta su adopción, garantizando que los nuevos tutores comprendan plenamente su compromiso con el bienestar del animal (p. 8).

2.1.3.2. Registro médico veterinario

2.1.3.2.1. Información del paciente y cliente.

De acuerdo a Wingfield y Raffe (2020), la identificación precisa y detallada tanto del paciente como del cliente es crucial. El historial médico de cada animal debe incluir su nombre, especie, raza, fecha de nacimiento, sexo, color y cualquier marca distintiva. Asimismo, se deben registrar con precisión el nombre del propietario, su dirección principal y todos los números de teléfono relevantes. Esto garantiza una comunicación clara y un seguimiento adecuado de todos los servicios veterinarios (p. 1174).

2.1.3.2.2. Motivo de consulta

Wingfield y Raffe (2020) consideran que, es la parte vital dentro del historial médico, ya que refleja la perspectiva del cliente sobre la salud actual del paciente. Debe incluir claramente las observaciones sobre los síntomas, ya que estos suelen proporcionar información crucial y pistas sobre el problema actual y sus posibles causas. Es importante que cualquier problema registrado en el historial médico esté definido y respaldado por datos objetivos siempre que sea posible, garantizando así su claridad y precisión (p. 1174).

2.1.3.2.3. Historial médico

Wingfield y Raffe (2020) mencionan que, un historial médico completo es esencial para un diagnóstico preciso y un tratamiento eficaz. En esta sección se debe documentar toda la información de salud relevante, tanto pasada como presente. Esto incluye, entre otros, todas las enfermedades y lesiones previas, procedimientos quirúrgicos, radiografías, vacunas y cualquier otra afección médica notable que haya experimentado el paciente. Un historial completo proporciona un contexto vital para comprender los problemas de salud actuales (p. 1174).

2.1.3.2.4. Examen Físico

Según Wingfield y Raffe (2020), se debe realizar un estudio físico a todos los pacientes y documentarlo meticulosamente antes de cualquier procedimiento médico o quirúrgico. Se debe seguir siempre un procedimiento sistemático para garantizar su coherencia e integridad. Además, el peso corporal actual del paciente debe registrarse en el historial médico, ya que es un parámetro fundamental para evaluar la salud y calcular las dosis de los medicamentos (p. 1175).

2.1.3.3. Gestión organizacional

2.1.3.3.1. Políticas y protocolos internos

Association of Shelter Veterinarians (2022) explica que, las políticas organizacionales proporcionan un sistema de referencial para garantizar que las necesidades de los animales no sobrepasen el límite de los recursos que tiene la organización para cuidar a los animales. Las áreas claves dentro de las organizaciones abarcan la admisión, tratamiento de enfermedades, adopción, transporte, eutanasia, entre otras (p. 6).

2.1.3.3.2. Capacitación del personal

Association of Shelter Veterinarians (2022) menciona que, debe existir una capacitación eficaz al personal, es crucial realizar este proceso para garantizar tanto el cuidado seguro y humanitario de los animales como de las personas forman para de la organización. La capacitación debe albergar todo tipo de aspectos que sean importantes en

el trabajo dentro de un refugio, como habilidades de comunicación, cría de animales, gestión de la información, entre otros. Un proceso de incorporación es vital para los miembros nuevos del personal, los refugios deben brindar una capacitación específica de acuerdo a cada trabajo que se realice dentro de la organización, requiriendo que el personal se encuentre cualificado para desempeñar el trabajo (p. 6).

2.1.3.3. Infraestructura y bienestar animal

De acuerdo a Association of Shelter Veterinarians (2022), los refugios de animales desempeñan un factor clave en la calidad del cuidado que reciben los animales. A pesar de la creciente visibilidad que están teniendo los programas de acogida y refugio comunitarios, proporcionar un refugio físico sigue siendo un componente vital de sus operaciones. La planificación, uso de edificios e infraestructura exterior del refugio son esenciales para promover la salud emocional y física de los animales. Además, debe ofrecer suficiente espacio para llevar a cabo eficazmente todas las operaciones y programas necesarios que realicen (p. 14).

2.1.3.4. Promoción y sostenibilidad en adopciones

Secretaría Metropolitana de Salud (2020) destaca que, un área central de enfoque es fomentar la adopción. Este proceso se realiza tanto por medio de internet como en actividades presenciales. Los animales que se consideran para adopción se someten a una sesión fotográfica profesional a cargo del equipo de Promoción y Comunicación. Estas imágenes se recopilarán en un folleto que también describe los parámetros esenciales para una tenencia responsable, fomenta la convivencia armoniosa y detalla las normas establecidas para los posibles adoptantes (p. 7).

2.1.3.4.1. Campañas de adopción

De Marco (2025) menciona que, las ferias de adopción y los eventos de adopción de mascotas ofrecen grandes oportunidades para conocer y adoptar animales que necesitan encontrar un hogar. Estos eventos, frecuentemente se encuentran organizados por refugios de animales y organizaciones de rescate. Al adoptar una mascota, no solo le estás dando a ese animal una segunda oportunidad para comenzar una nueva vida, sino que también

estás creando un espacio muy necesario en el refugio para que otro animal pueda ingresar y encontrar un hogar (pp. 126-127).

2.1.3.4.2. Donaciones

De Marco (2025) explica que, las donaciones a refugios permiten a las organizaciones poder solventar los gatos al brindar cuidado, tratamientos médicos y una segunda oportunidad a animales necesitados. Además, las donaciones monetarias recaudadas permiten cubrir gatos operativos, pago de salarios y el mantenimiento de las instalaciones donde albergan a gran número de animales. Cualquier tipo de donación impacta directamente en las necesidades que suelen tener los refugios, permitiendo así poder brindar un servicio adecuado y un lugar cómodo para los animales (p. 127).

2.1.3.4.3. Voluntariados

De acuerdo a De Marco (2025), el voluntariado con organizaciones de animales ofrece una variedad de oportunidades para marcar la diferencia en la vida de los animales. Las organizaciones, suelen tener poco personal y estar saturados, es muy importante contar con personas que estén dispuestas a ser voluntarios y brinden su ayuda en estos lugares. Los voluntarios suelen encargarse de tareas como alimentación, limpieza, suministro de agua, pasar tiempo con los animales, entre otras tareas (pp. 129-130).

2.2. Predicción científica

H0: La aplicación *web* con *Machine Learning* no incide significativamente en el fortalecimiento de la gestión de adopción de mascotas en el Centro Veterinario Mi Mascota del cantón Santo Domingo.

H1: La aplicación *web* con *Machine Learning* incide significativamente en el fortalecimiento de la gestión de adopción de mascotas en el Centro Veterinario Mi Mascota del cantón Santo Domingo.

3. METODOLOGÍA

3.1. Enfoque, diseño y tipo de investigación

La presente investigación se estableció bajo un marco estricto, orientándose hacia un enfoque cuantitativo. Este método, se seleccionó ya que permite la recopilación y análisis sistemático de los datos numéricos, simplificando medir el impacto y que tan eficiente es la Aplicación *Web* con *Machine Learning*.

Además, Hernández y Mendoza (2018) comentan que, la investigación cuantitativa se fundamenta en la recolección y el procesamiento de información con el objetivo de poder responder las preguntas planteadas y comprobar las hipótesis establecidas, haciendo uso de técnicas estadísticas y medición numérica (p. 6).

Por otra parte, el alcance define la profundidad con la que se aborda el objeto de estudio, Hernández y Mendoza (2018) señalan la clasificación tradicional del alcance como: exploratorio, descriptivo, correlacional y explicativo (p. 106). Sin embargo, Avellaneda et al. (2022) proponen una división más completa, añadiendo los niveles predictivo y aplicativo. En este sentido, el nivel aplicativo se caracteriza por su orientación hacia la intervención directa y la resolución práctica de problemas concretos en un entorno determinado. En lugar de limitarse a observar o predecir un fenómeno, este enfoque busca modificar activamente la realidad o el curso natural de una situación específica (p. 42). Por consiguiente, la presente investigación se enmarca dentro de un alcance de nivel aplicativo.

Por otro lado, dentro de la investigación de enfoque cuantitativo, los diseños experimentales se categorizan en tres tipos principales según el nivel de control sobre las variables. Como indican Cruz et al. (2014), el diseño preexperimental utiliza un solo grupo de control; el cuasiexperimental emplea dos o más grupos intactos para comparar efectos; y el experimental puro exige al menos dos grupos: experimental y de control (pp. 130-132). Adicionalmente, Hernández y Mendoza (2018) menciona que, el diseño preexperimental se caracteriza por tener un solo grupo, una medición preprueba y postprueba. Son útiles como

acercamiento a los problemas que hay en las investigaciones (p. 163). Por lo indicado anteriormente, se consideró de acuerdo al contexto de la investigación optar por el diseño preexperimental.

Finalmente, el tipo de investigación se ha clasificado como aplicada, la decisión se fundamenta en el objetivo principal que se enfoca en construir, desplegar y validar un sistema tecnológico. Según Hernández y Mendoza (2018), el objetivo de la investigación aplicada es obtener soluciones y generar información con un uso práctico y resultados palpables para resolver problemas prácticos. Además, se desarrolló una investigación de campo, la cual permitió recolectar los datos en un ambiente real (p. 172), en este caso en el Centro Veterinario Mi Mascota.

3.2. Unidades de análisis

Aunque la clínica ofrece diversos servicios veterinarios, el área de gestión de adopción de animales, se integró en 2025. Para este estudio, se identificó una población de 45 clientes recurrentes (población finita) que muestran un interés activo en el proceso de adopción o que acuden específicamente para informarse sobre los animales disponibles. Dada la naturaleza finita y el tamaño accesible de la población, se aplicó un muestreo censal o población censal.

Esta decisión, se fundamenta en la definición de Hernández y Mendoza (2018) quienes establecen que, aunque lo común es trabajar con muestras para economizar tiempo y recursos, existen casos en los que se estudia a toda la población debido a su tamaño manejable, a este proceso se lo conoce como censo (p. 196), otros autores le denominan muestreo censal.

Por lo tanto, para el estudio de la situación del centro veterinario en el área de gestión de adopción de mascotas, se cuenta con una muestra de 45 personas para realizar

la encuesta, las cuales están conformados por los clientes del centro veterinario, junto con el personal administrativo (3 personas), dan un total de 48 individuos.

Tabla 1. Población

Mi mascota	
Personal	Clientes
3	45
Total=48	

3.3. Técnicas e instrumentos de investigación

La investigación utiliza una encuesta estandarizada como técnica principal para recolectar datos cuantitativos de los 45 participantes. El cuestionario se aplicó dos veces, en una fase previa a la aplicación del producto y una posterior para medir el impacto del fortalecimiento de la gestión de adopción de mascotas a través de la Aplicación *Web* implementada.

De acuerdo a Briones (2002), la encuesta es uno de los enfoques más comunes en la investigación cuantitativa, consiste en un método para recolectar información a través de preguntas formuladas de manera oral o escrita, dirigidas a un grupo o muestra de individuos que poseen las características específicas que fueron necesarias para el problema de investigación (p.51). Además, Hernández y Mendoza (2018) mencionan que, las preguntas se diseñan rigurosamente para abarcar los aspectos relevantes de las variables con revisión de expertos (p. 230).

3.4. Técnicas de análisis de datos

Según Hernández y Mendoza (2018), el análisis de datos en la investigación cuantitativa implica el uso de procedimientos estadísticos para probar hipótesis y hacer estimaciones sobre las características de una población. Por lo tanto, se establece que la estadística descriptiva resumen los datos, y la estadística inferencial se utilizan para tomar decisiones sobre una población basándose en los datos de la muestra (p. 310).

Por consecuente, el análisis de los datos se basó en un inicio en la estadística descriptiva, con el fin de ordenar, sintetizar y mostrar los hallazgos de un modo claro y comprensible, se utilizaron las técnicas de análisis de datos de: distribución de frecuencias relativas (porcentajes); y análisis descriptivo comparativo mediante gráfico de barras agrupadas, con la herramienta de Microsoft Excel.

Posteriormente, para el tratamiento y estudio de la información recabada en los cuestionarios, se utilizó la estadística inferencial, mediante la técnica de regresión logística binaria para el análisis de datos, a través de la herramienta *IBM SPSS*, requerido para la comparación de los hallazgos de la fase inicial (*PreTest*) y final (*PostTest*).

3.5. Operacionalización de las variables

Tabla 2. Operacionalización de variable independiente aplicación web.

Conceptualización	Dimensión	Indicadores	Preguntas	Herramienta
<p>Aplicación Web Moscato (2024) menciona que, es una aplicación de <i>software</i> que se ejecuta dentro de un navegador <i>web</i>, tienen como propósito conceder acceso a las capacidades del <i>software</i> y a los datos a través de una red mediante tecnologías <i>web</i> estándar, permitiendo a los usuarios acceder a las funcionalidades desde cualquier dispositivo. Además, las aplicaciones <i>web</i> son útiles cuando se desea proporcionar acceso a <i>software</i> y datos a una gran cantidad de usuarios especialmente dentro de una organización (p. 5).</p>	Herramientas de desarrollo	Lenguajes de Programación	¿Con qué frecuencia hace uso de aplicaciones <i>web</i> ?	Encuesta al cliente
		Frameworks	¿Cuál es su nivel de dificultad navegar en aplicaciones <i>web</i> ?	Encuesta al cliente
			¿Considera necesario el cómo está diseñada una aplicación <i>web</i> ?	Encuesta al cliente
		Editor de código fuente	¿Qué tan familiarizado se encuentra con aplicaciones web para poder ver mascotas en adopción?	Encuesta al cliente
	Bases de Datos	Modelado de datos	¿Considera útil la implementación de una aplicación <i>web</i> que permita gestionar la adopción de mascotas?	Reunión de planificación del <i>sprint</i> con el gerente
		Lenguaje de base de datos	¿Qué método usan para registrar a las mascotas que están disponibles para adoptar?	Reunión de planificación del <i>sprint</i> con el gerente
		Tipos de base de datos	¿Considera que la información sobre las mascotas y los adoptantes se gestiona de manera organizada?	Reunión de planificación del <i>sprint</i> con el gerente
		SGBD	¿Qué tipo de información considera que debe ser visible en la aplicación <i>web</i> ?	Reunión de planificación del <i>sprint</i> con el gerente
	Seguridad web	Factor de autenticación	¿Con qué frecuencia siente que hay poca información sobre las mascotas en adopción?	Encuesta al cliente
		Encriptación de datos	¿Está de acuerdo en que se implementen sistemas de autenticación de datos?	Encuesta al cliente
Protocolos de seguridad		¿Le parece adecuado que en una aplicación <i>web</i> las publicaciones sean aprobadas antes de publicarlas?	Reunión de planificación del <i>sprint</i> con el gerente	
			¿Qué tan seguro se siente al utilizar aplicaciones <i>web</i> ?	Encuesta al cliente

Tabla 3. Operacionalización de variable independiente *machine learning*.

Conceptualización	Dimensión	Indicadores	Preguntas	Herramienta	
<p>Machine Learning</p> <p>Joshi (2022) indica que, el aprendizaje automático reúne una amplia gama de algoritmos surgidos inicialmente de campos como la ingeniería eléctrica, estadística, análisis financiero e incluso las ciencias genéticas. Los componentes fundamentales de todos estos algoritmos se basan profundamente en los principios de las matemáticas puras y la estadística. Las computadoras cumplen un papel importante en la automatización de cálculos complejos. Esta automatización permite el procesamiento iterativo, el ajuste de parámetros y el reconocimiento de patrones, el efecto acumulativo de estos cálculos permite que los sistemas de aprendizaje automático aborden problemas complejos (p. 5).</p>	<p><i>Métodos de Machine Learning</i></p>	Aprendizaje supervisado	¿Está de acuerdo en que se implemente un sistema que aprenda de las preferencias del usuario en las aplicaciones <i>web</i> ?	Reunión de planificación del <i>sprint</i> con el gerente	
		Aprendizaje no supervisado			
		Aprendizaje semisupervisado	¿Con qué frecuencia utiliza aplicaciones que implementen inteligencia artificial o sistemas que aprendan de las interacciones del usuario?	Encuesta al cliente	
		Aprendizaje de refuerzo			
		Árboles de decisión			
		<i>Naive Bayes</i>			
	<p><i>Principales algoritmos de aprendizaje</i></p>	Regresión logística	¿Con qué frecuencia los animales que ve en adopción son de acuerdo a sus preferencias?	Encuesta al cliente	
		Regresión lineal			
		<i>K-means Clustering</i>	¿Con qué frecuencia recibe recomendaciones sobre que mascota adoptar?	Encuesta al cliente	
		<i>Random forests</i>			
		<p><i>Canalización de Machine Learning</i></p>	Recopilación de datos	¿Considera que es importante la implementación de herramientas que permitan mostrar al usuario recomendaciones personalizadas según su perfil?	Reunión de planificación del <i>sprint</i> con el gerente
			Preprocesamiento de datos		
			Ingeniería de características		
			División y selección datos	¿Cuál es su nivel de dificultad para encontrar una mascota en adopción de acuerdo a sus necesidades?	Encuesta al cliente
Entrenamiento del modelo					
Evaluación del modelo					
Despliegue y mantenimiento					

Tabla 4. Operacionalización de variable dependiente gestión de adopción de mascotas

Conceptualización	Dimensión	Indicadores	Preguntas	Herramienta
Gestión de adopción de mascotas De acuerdo a Association of Shelter Veterinarians (2022), las organizaciones dedicadas a la adopción de animales deben implementar prácticas de gestión alineadas a sus directrices. La identificación y mantenimiento de registro de animales son un pilar fundamental dentro de las organizaciones. Estos elementos no solo son esenciales para una operación eficaz, sino que también deben cumplir con todos los requisitos regulatorios establecidos. La integración de sistemas digitales para el mantenimiento de registros ayuda a llevar un control administrativo eficiente y organizado, ya que facilita los procesos de gestión de adopciones dentro de las organizaciones (pp. 5-6).	<i>Procesos de adopción</i>	Selección	¿Qué tan satisfecho está con las opciones disponibles y la facilidad para encontrar una mascota adecuada a sus necesidades?	Encuesta al cliente
			¿Cuál es su nivel de dificultad al completar la documentación necesaria para la adopción?	Encuesta al cliente
			¿Qué tan difícil le resulta el proceso de registro para los adoptantes interesados?	Reunión de planificación del <i>sprint</i> con el gerente
		Solicitud	¿Con qué frecuencia siente que el proceso de adopción es poco claro o confuso?	Encuesta al cliente
			¿Considera que la evaluación previa de los adoptantes es suficientemente rigurosa para asegurar hogares adecuados?	Reunión de planificación del <i>sprint</i> con el gerente
			¿Qué tan familiarizado se encuentra con entrevistas para la adopción de una mascota?	Encuesta al cliente
			Entrevista y Aprobación	¿Con qué frecuencia ha sido notificado sobre entrevistas o pasos siguientes tras enviar su solicitud?
	¿Está usted de acuerdo en recibir información actual sobre el proceso de adopción?	Encuesta al cliente		
	<i>Registro médico veterinario</i>	Información del paciente y cliente Motivo de consulta Historial médico Examen Físico	¿Qué tan satisfecho se encuentra con la información que se proporciona de las mascotas?	Encuesta al cliente
			¿Qué medidas toman cuando reciben una mascota para dar en adopción?	Reunión de planificación del <i>sprint</i> con el gerente
			¿Utilizan protocolos de seguimiento post-adopción para asegurar el bienestar de los animales?	Reunión de planificación del <i>sprint</i> con el gerente
	<i>Gestión organizacional</i>	Políticas y protocolos internos Capacitación del personal Infraestructura y bienestar animal	¿Con qué regularidad se proporciona capacitación sobre cómo manejar situaciones difíciles o sensibles con posibles adoptantes?	Reunión de planificación del <i>sprint</i> con el gerente
			¿Con qué frecuencia busca mascotas en adopción por internet?	Encuesta al cliente
	<i>Promoción y sostenibilidad en adopciones</i>	Campañas de adopción Donaciones Voluntariados	¿Cree que es necesario mejorar la forma en que se difunden las mascotas disponibles para adopción?	Encuesta al cliente

4. RESULTADOS

4.1. Resultado del primer objetivo específico

La encuesta a clientes y la reunión de planificación con el gerente, seleccionadas como técnica de investigación para la recopilación de datos, detalladas en el anexo III, fueron evaluadas y validadas por expertos en las áreas pertinentes, su información se observa en la tabla 5.

Tabla 5. Expertos en Validación de Instrumentos

Nombres	Área	Título Académico
Luis Javier Ulloa Meneses	Sistemas	Magister en Big Data y Data Science
Jonathan Mario Moreno Rivera	Sistemas	Master en Ingeniería de software y sistemas informáticos
Brithany Naomi Velasco Reyes	Medicina Veterinaria	Médico Veterinaria

4.1.1. Aplicación de las encuestas a los clientes

En la primera etapa, se realizó la encuesta antes de implementar la propuesta de intervención, la cual fue realizada a los clientes del Centro Veterinario Mi Mascota.

4.1.1.1. Presentación de los resultados de la encuesta a los clientes

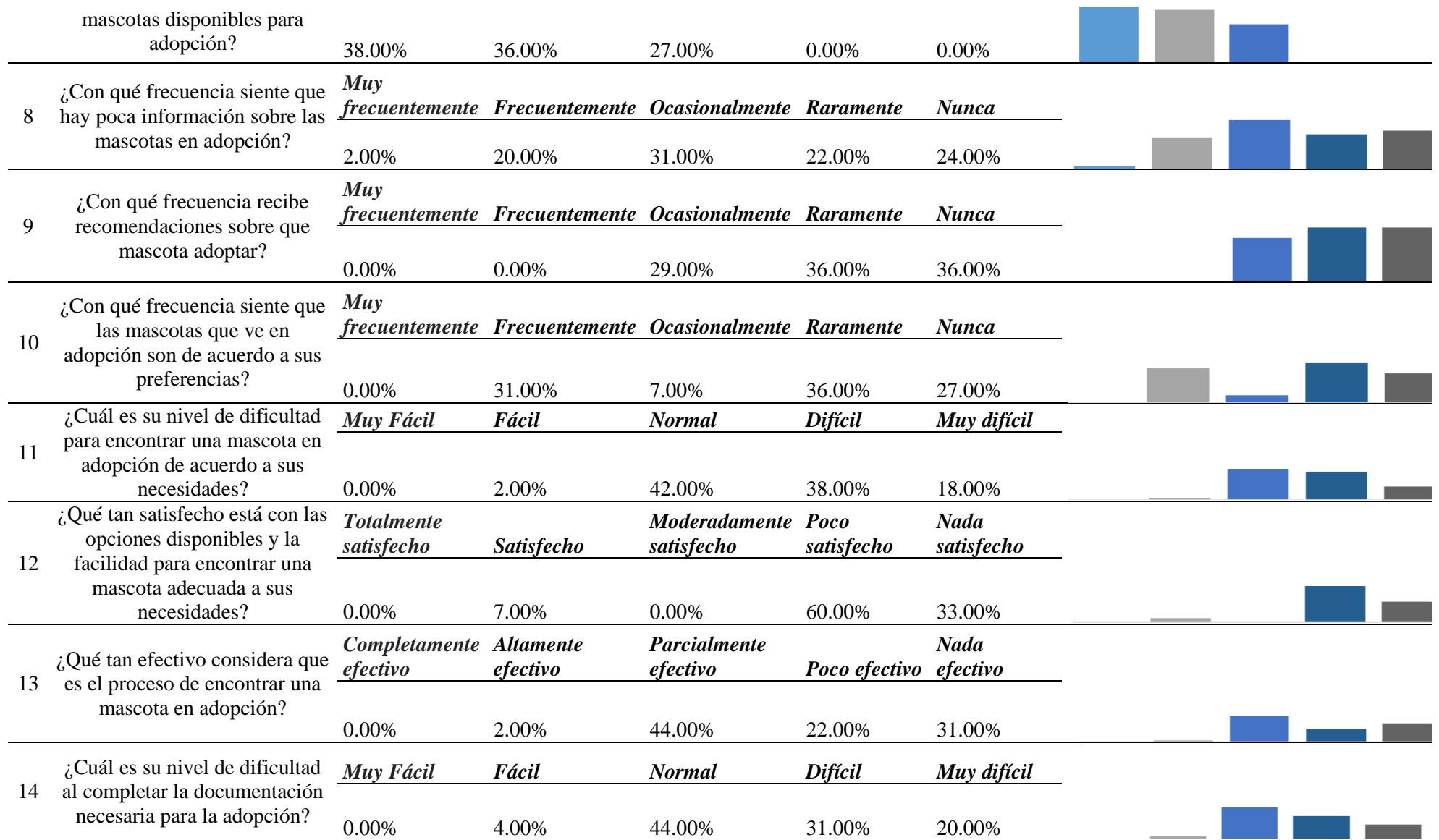
De acuerdo a la recopilación de datos realizada a los 45 clientes del centro veterinario por medio de la aplicación de una encuesta en la fase del *PreTest*, se puede evidenciar que, el 56% de los usuarios considera que encontrar una mascota acorde a sus necesidades es entre “Difícil” y “Muy Difícil”. Por otro lado, un 2.00% percibe el proceso de adopción como “Altamente efectivo”, mientras que el 53.00% lo sitúa en las categorías de “Poco efectivo” y “Nada efectivo”, además el 31.00% y 20.00% considera “Difícil” y “Muy difícil” completar la documentación para adoptar. Por último, el 93.00% de los clientes reporta estar “Poco satisfecho” o “Nada satisfecho” con la información que se proporciona sobre las mascotas, con la información presentada, permitió tener una idea clara sobre las percepciones de los clientes.

4.1.1.2. Resultados de la encuesta a los clientes

En la tabla 6, se visualizan los resultados de la encuesta aplicada a los clientes del Centro Veterinario Mi Mascota.

Tabla 6. Resultados de la encuesta a los clientes del pre test

N° Preguntas	Escala y %					Figuras	
	<i>Muy frecuentemente</i>	<i>Frecuentemente</i>	<i>Ocasionalmente</i>	<i>Raramente</i>	<i>Nunca</i>		
1	¿Con qué frecuencia hace uso de aplicaciones web?	<i>Muy frecuentemente</i>	<i>Frecuentemente</i>	<i>Ocasionalmente</i>	<i>Raramente</i>	<i>Nunca</i>	
		4.00%	18.00%	22.00%	31.00%	24.00%	
2	¿Cuál es su nivel de dificultad navegar en aplicaciones web?	<i>Muy Fácil</i>	<i>Fácil</i>	<i>Normal</i>	<i>Difícil</i>	<i>Muy difícil</i>	
		0.00%	9.00%	40.00%	29.00%	22.00%	
3	¿Qué tan seguro se siente al utilizar aplicaciones web?	<i>Totalmente seguro</i>	<i>Seguro</i>	<i>Moderadamente seguro</i>	<i>Poco seguro</i>	<i>Totalmente inseguro</i>	
		0.00%	2.00%	38.00%	31.00%	29.00%	
4	¿Considera necesario el cómo está diseñada una aplicación web?	<i>Muy necesario</i>	<i>Necesario</i>	<i>Medianamente necesario</i>	<i>Poco necesario</i>	<i>Innecesario</i>	
		29.00%	27.00%	44.00%	0.00%	0.00%	
5	¿Con qué frecuencia busca mascotas en adopción por internet?	<i>Muy frecuentemente</i>	<i>Frecuentemente</i>	<i>Ocasionalmente</i>	<i>Raramente</i>	<i>Nunca</i>	
		4.00%	11.00%	18.00%	33.00%	33.00%	
6	¿Qué tan familiarizado se encuentra con aplicaciones web para poder ver mascotas en adopción?	<i>Completamente familiarizado</i>	<i>Muy familiarizado</i>	<i>Algo familiarizado</i>	<i>Poco familiarizado</i>	<i>Nada familiarizado</i>	
		0.00%	0.00%	27.00%	29.00%	44.00%	
7	¿Cree que es necesario mejorar la forma en que se difunden las	<i>Muy necesario</i>	<i>Necesario</i>	<i>Medianamente necesario</i>	<i>Poco necesario</i>	<i>Innecesario</i>	



15	¿Con qué frecuencia siente que el proceso de adopción es poco claro o confuso?	<i>Muy frecuentemente</i>	<i>Frecuentemente</i>	<i>Ocasionalmente</i>	<i>Raramente</i>	<i>Nunca</i>					
		0.00%	4.00%	40.00%	22.00%	33.00%					
16	¿Qué tan familiarizado se encuentra con entrevistas para la adopción de una mascota?	<i>Completamente familiarizado</i>	<i>Muy familiarizado</i>	<i>Algo familiarizado</i>	<i>Poco familiarizado</i>	<i>Nada familiarizado</i>					
		0.00%	0.00%	47.00%	16.00%	38.00%					
17	¿Con qué frecuencia ha sido notificado sobre entrevistas o pasos siguientes tras enviar su solicitud?	<i>Muy frecuentemente</i>	<i>Frecuentemente</i>	<i>Ocasionalmente</i>	<i>Raramente</i>	<i>Nunca</i>					
		0.00%	0.00%	22.00%	53.00%	24.00%					
18	¿Está usted de acuerdo en recibir información actual sobre el proceso de adopción?	<i>Totalmente de acuerdo</i>	<i>Bastante de acuerdo</i>	<i>Neutral</i>	<i>Poco de acuerdo</i>	<i>Totalmente en desacuerdo</i>					
		33.00%	42.00%	16.00%	9.00%	0.00%					
19	¿Qué tan satisfecho se encuentra con la información que se proporciona de las mascotas?	<i>Totalmente satisfecho</i>	<i>Satisfecho</i>	<i>Moderadamente satisfecho</i>	<i>Poco satisfecho</i>	<i>Nada satisfecho</i>					
		2.00%	0.00%	4.00%	60.00%	33.00%					
20	¿Está de acuerdo en que se implementen sistemas de autenticación de datos?	<i>Totalmente de acuerdo</i>	<i>Bastante de acuerdo</i>	<i>Neutral</i>	<i>Poco de acuerdo</i>	<i>Totalmente en desacuerdo</i>					
		13.00%	27.00%	60.00%	0.00%	0.00%					
21	¿Con qué frecuencia utiliza aplicaciones que implementen inteligencia artificial o sistemas que aprendan de las interacciones del usuario?	<i>Muy frecuentemente</i>	<i>Frecuentemente</i>	<i>Ocasionalmente</i>	<i>Raramente</i>	<i>Nunca</i>					
		0.00%	0.00%	2.00%	33.00%	64.00%					

4.1.2. Identificación de los procesos en la gestión de adopción de mascotas

Por medio de la reunión de planificación del *sprint* realizada al gerente del Centro Veterinario “Mi Mascota” detallada en el anexo X, se reveló información sobre el funcionamiento de la empresa. En concreto, los procesos del Centro Veterinario para la gestión de adopción de mascotas son: Ingreso y recepción, evaluación veterinaria, catalogación y publicidad y, por último, adopción y entrega.

4.1.2.1. Diagrama BPD del proceso de gestión de adopción de mascotas

A través de las visitas realizadas al dueño de la empresa, se obtuvo un panorama claro sobre la actividades y procesos que se realizan con respecto a la gestión de adopción de mascotas. En la figura 4, 5, 6, 7 y 8, se encuentra el *Business Process Diagram (BPD)*, en donde se utilizó *Business Process Model and Notation (BPMN 2.0)*. El diagrama *BPD*, se utilizó exclusivamente como una representación visual para describir la estructura, el flujo y la lógica de los procesos de gestión de adopción de mascotas.

Por lo que, el diagrama *BPD* fue empleado como un modelo de la operatividad para el desarrollo de la propuesta de intervención (aplicación *web* con *machine learning* para la gestión de adopción de mascotas) y sobre todo en la ingeniería de requerimientos con enfoque ágil, en la reunión de planificación del *sprint*, y la generación de las historias de usuarios. Es importante aclarar que, el diagrama *BPD* no se empleó como un artefacto del enfoque de la investigación que consta en la metodología de la investigación.

Figura 4. Diagrama BPD de gestión de adopción de mascotas

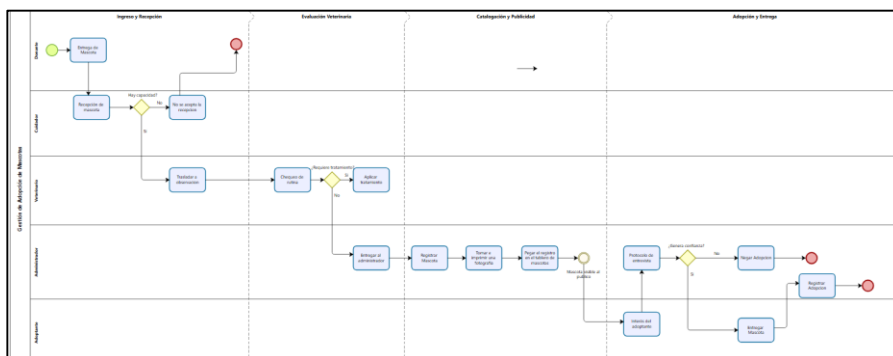


Figura 5. Proceso de ingreso y recepción

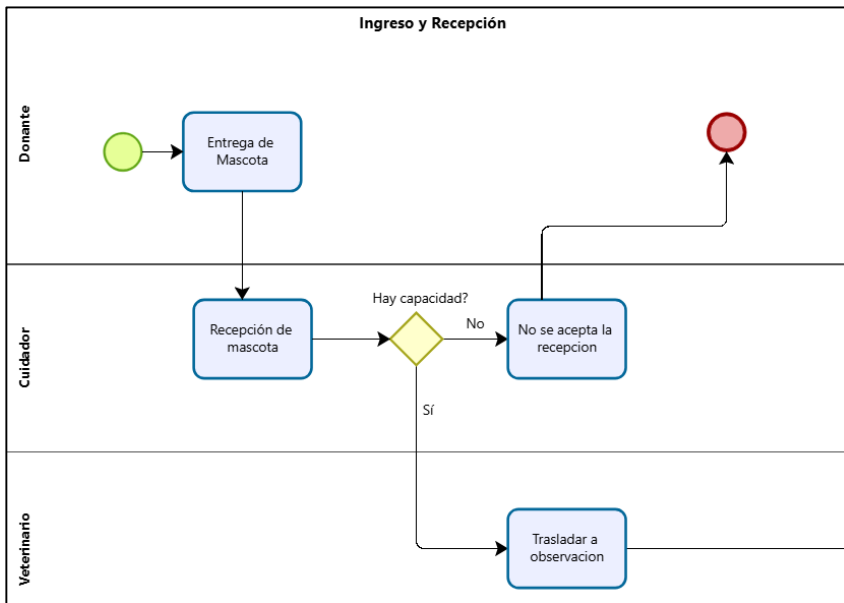


Figura 6. Proceso de evaluación veterinaria

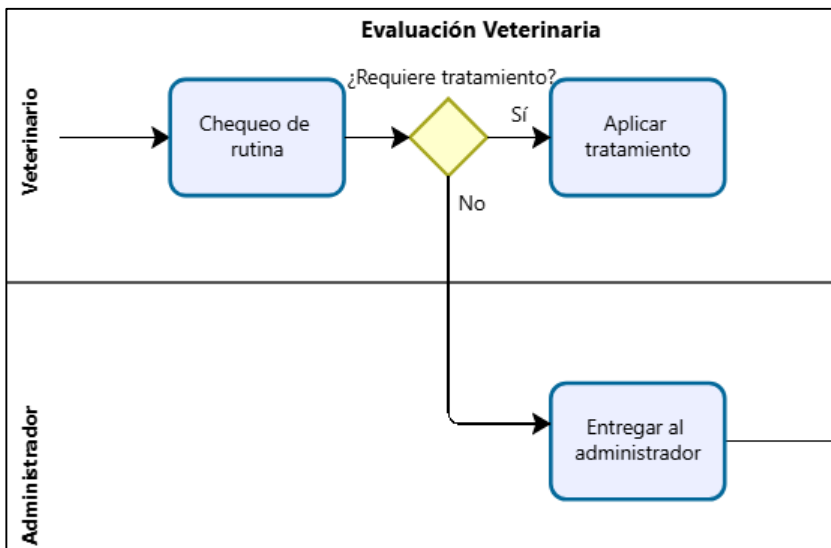


Figura 7. Proceso de catalogación y publicidad

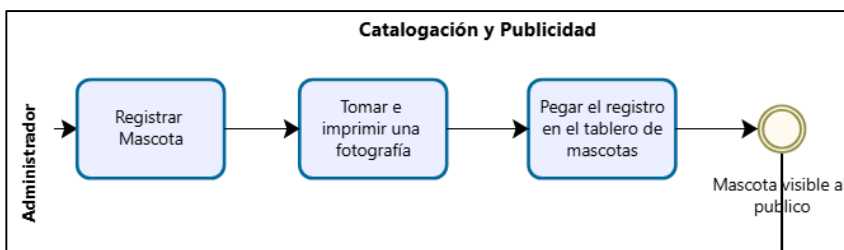
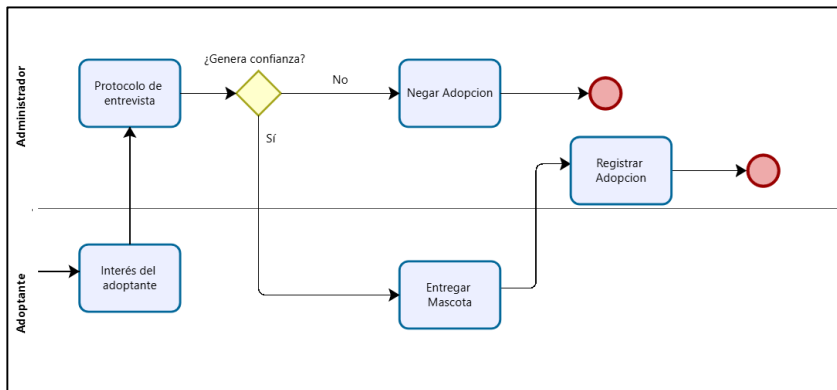


Figura 8. Proceso de adopción y entrega



4.2. Resultados del segundo objetivo específico

El desarrollo de la aplicación, se fundamenta en separar las responsabilidades y se utilizó el marco de trabajo *Scrum*, además, es necesario y fundamental contar con un ecosistema de herramientas. Por lo tanto, se decidió desarrollar la aplicación *web* con *machine learning* para la gestión de adopción de mascotas, como parte inicial, se analizaron los criterios para determinar las herramientas que se emplearán como: *framework* de *frontend*, *framework* de *backend*, gestores de bases de datos, arquitecturas de *software* y algoritmos de *machine learning*.

4.2.1. Framework Frontend

En la actualidad existen un sin número de marcos de trabajo que permiten agilizar el desarrollo de interfaces atractivas y eficientes para el usuario. Por lo tanto, se ha optó por realizar un análisis de las características con las que cuentan diferentes *frameworks* y librerías, tal como se observa en la tabla 7.

Tabla 7. Comparativa de Framework Frontend

Criterios	Frameworks		
	Vue.js^a	React^b	Angular^c
<i>Tipo</i>	<i>Framework</i>	Librería	<i>Framework</i>
<i>Arquitectura</i>	Enfocado en las capas "Vista" y "Modelo"	Se enfoca en la "Vista" del MVC	MVC
<i>Componentes</i>	Single File Component	Basado en funciones	Basados en clases
<i>Lenguaje</i>	<i>JavaScript</i>	<i>JavaScript+JSX</i>	<i>TypeScript</i>
<i>Tamaño</i>	Más ligero	Intermedio	Más grande

<i>Manejo de DOM</i>	Virtual DOM	Virtual DOM	Real DOM
<i>Enlace de datos</i>	Bidireccional	Unidireccional	Bidireccional
<i>Progresividad</i>	Sí	Parcialmente flexible	No
<i>Curva de aprendizaje</i>	Baja	Media	Alta

Nota. Fuente de: ^{a, b, c} Vyas (2022)

En la tabla 7, se encuentran los atributos más importantes de cada uno de los *frameworks* y librerías. De acuerdo al análisis de la comparativa se optó por emplear *Vue.js*, dado su progresividad y su curva de aprendizaje suave que permite una adopción más rápida para el equipo de desarrollo, proporcionando una estructura de código legible y mantenible mediante los *Single File Component*.

4.2.2. Framework Backend

El *Backend* es el responsable de toda la lógica, el procesamiento y persistencia de los datos en el servidor, la elección de un *Framework Backend* radica en que simplifique y acelere el desarrollo detrás de escena, en la tabla 8, se realizó una comparativa entre los *framework* más utilizados en dentro de esta área.

Tabla 8: Comparativa de Framework Backend.

Criterios	Frameworks		
	Express.js^a	Django^b	Sprint^c
<i>Tipo</i>	<i>Minimalist</i>	<i>Marco web</i>	<i>Proyecto de código abierto</i>
<i>Lenguaje Base</i>	<i>Node.js (JavaScript)</i>	<i>Python</i>	<i>Java (Java EE)</i>
<i>Arquitectura</i>	<i>Middleware y Routing</i>	<i>MVC</i>	<i>MVC</i>
<i>Rendimiento</i>	Proporciona una capa ligera de funciones para aplicaciones web	Utiliza gestores caches para liberar recursos	Se apoya en el servidor de aplicaciones para la gestión de carga y calidades de servicio
<i>Escalabilidad</i>	Alto	Alto	Muy Alto
<i>Complejidad</i>	Fácil	Alto	Alto
<i>Velocidad de desarrollo</i>	Rápida	Rápida	Lenta

Nota. Fuente de: ^a Express (2026), ^b IBM (2026), ^c IBM (2025)

Por medio de la tabla 8, se visualizan cada una de las características de los *Frameworks Backend*, permitiendo obtener una vista clara sobre cada una de las herramientas, se optó por elegir *Express.js* debido a su facilidad y velocidad de desarrollo, que evita la sobrecarga y permite crear *APIs* y lógica rápidamente.

4.2.3. Base de Datos

En la base de datos, se almacena y organiza toda la información que la aplicación necesita para funcionar y ofrecer los servicios. En la tabla 9, se detallan las características de los sistemas gestores de bases de datos más utilizados.

Tabla 9. Comparativa entre los SGBD.

Criterios	Base de Datos		
	PostgreSQL^a	MongoDB^b	MySQL^c
<i>Tipo</i>	Relacional	No relacional	Relacional
<i>Modelado de datos</i>	Utiliza filas y columnas para almacenar datos.	Los datos son procesados como tipo <i>JSON</i> en colecciones.	Utiliza filas y columnas para almacenar datos.
<i>Escalabilidad</i>	Escalabilidad Vertical	Escalabilidad Horizontal	Escalabilidad Vertical
<i>Rendimiento</i>	Alto rendimiento en consultas complejas.	Rápido para datos no estructurados/semiestructurados.	Bueno para transacciones simples.
<i>Lenguaje de consulta</i>	Variante de <i>SQL</i> compatible con las consultas estándar <i>SQL</i> .	Lenguaje de consulta de <i>MongoDB</i> .	<i>SQL</i>
<i>Simultaneidad</i>	Alto control mediante <i>MVCC</i> y detección de interbloqueos	Control mediante atomicidad a nivel documento y <i>MVCC</i>	Control mediante bloqueos a nivel de fila o tabla.

Nota. Fuente de: ^{a,b}Amazon Web Services (2025a), ^cAmazon Web Services (2025b)

Basado en la comparativa realizada en la tabla 9, se optó por *PostgreSQL* como elección de base de datos debido a su capacidad para realizar consultas complejas y mantener la consistencia de datos e integridad referencial.

4.2.4. Arquitecturas de software

La arquitectura actúa como el plano del sistema, organiza el código y logra que funcione de manera eficiente. En la tabla 10, se desatacan aspectos importantes de las arquitecturas de *software*.

Tabla 10. Comparativa entre arquitecturas de software

Criterios	Arquitecturas de Software	
	MVC^a	MVVM^b
<i>Patrón</i>	<i>Model, View, Controller.</i>	<i>Model, View, ViewModel.</i>
<i>Intermediario</i>	<i>Controller</i>	<i>ViewModel</i>

<i>Característica</i>	Separación clara de responsabilidades; la vista muestra datos, el modelo gestiona la lógica y el controlador coordina las acciones	Usa <i>data binding</i> que permite sincronización automática entre <i>View</i> y <i>ViewModel</i>
<i>Ventaja</i>	Aísla la lógica de negocio de la capa de presentación, facilitando el mantenimiento y la reutilización de código	Presenta mejores condiciones de rendimiento y productividad
<i>Desventaja</i>	En enfoques tradicionales, puede generar dependencia fuerte entre UI y datos si no se aplica correctamente	Requiere que la vista sea responsabilidad de un diseñador en lugar de un desarrollador especializado en lógica

Nota. Fuente de: ^{a, b} Arcos et al. (2018)

De acuerdo con la comparativa realizada en la tabla 10, se decidió utilizar el Modelo Vista Controlador (MVC) como la arquitectura del producto, ya que permite tener una estructura organizada y dividir las funciones en tres grupos.

4.2.5. Machine Learning

El aprendizaje automático permite analizar tendencias de datos y patrones de comportamiento para mejorar la experiencia de los usuarios. En la tabla 11, se encuentran detalladas diferentes técnicas de aprendizaje automático.

Tabla 11. Comparativa entre técnicas de Machine Learning.

Crterios	Machine Learning		
	Content-Based Filtering^a	Collaborative Filtering^b	Hybrid Filtering^c
<i>Funcionamiento Principal</i>	Analiza las propiedades del ítem para predecir. Además, genera recomendaciones usando perfiles de usuario y características extraídas de registros pasados.	Compila una base de datos de preferencias de usuarios para emparejar usuarios con intereses idénticos.	Combina diferentes métodos para integrar los resultados de múltiples modelos.
<i>Técnicas / Algoritmos</i>	Utiliza modelos de clasificación y espacio vectorial.	Se divide en métodos basados en memoria y basados en modelo.	Incluye estrategias como: cascadas, aumento de características, mixto, etc.
<i>Ventajas</i>	Resuelve el problema de arranque frío porque no depende de datos de otros usuarios. Facilita la escalabilidad para grandes bases de usuarios.	No necesita conocimiento del dominio. Puede ayudar a los usuarios a encontrar nuevos intereses.	Integra las ventajas de múltiples técnicas de filtrado. Puede proporcionar la mayor precisión en las recomendaciones para los usuarios.

<i>Desventajas</i>	Requiere experiencia en el dominio para crear las características manualmente. Tiene capacidad limitada para utilizar los intereses previos de los usuarios.	Sufre del problema de arranque frío. Es difícil aplicar características secundarias debido a los escasos de datos.	El modelo puede ser demasiado complejo para conjuntos de datos simples. Puede requerir hardware costoso y mayor potencia de procesamiento.
--------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------

Nota. Fuente de: ^{a,b,c} Kong et al. (2024)

En la tabla 11, se comparan diferentes técnicas de aprendizaje automático, por lo que se determinó utilizar la técnica de Filtrado Híbrido (*Hybrid Filtering*) por su capacidad para integrar métodos de filtrado, tales como el filtrado colaborativo y filtrado basado en contenido. Esta elección se realizó con el fin de proporcionar precisión en las recomendaciones realizadas para cada uno de los usuarios.

4.3. Resultado del tercer objetivo específico

4.3.1. Nomenclatura y logotipo de la aplicación web

Con relación a la figura 9, se decidió usar la nomenclatura “Adopciones Mi Mascota” para la aplicación *web*, la cual está organizada por la combinación del término “adopciones” con el nombre de la empresa.

Figura 9. Logotipo de la aplicación



Nota. Imagen generada con Canva (2025). Esta imagen se generó en respuesta a la siguiente solicitud: «Generar una imagen de logotipo en donde se encuentren un gato y un perro en blanco y negro».

4.3.2. Marco de trabajo Scrum

Para la creación de la aplicación *web* en el Centro Veterinario “Mi Mascota”, se utilizó el marco *Scrum*, el cual promueve el trabajo en equipo, flexibilidad y la entrega continua, lo que permite llevar la gestión adecuada del producto. El uso del marco de trabajo es una guía que, el resultado del software se encuentre alineado con las expectativas planteadas, gracias a su enfoque en la agilidad y la entrega continua de valor.

4.3.3. Sprint I

4.3.3.1. Planificación del Sprint I – Sprint Planning

Para el desarrollo de un *software*, el primer paso es contar con una planificación, una tarea crucial para poder llevar a cabo la ejecución de la aplicación. Se elaboró el primer *sprint* basado en el enfoque de *Scrum*, el cual refleja las preferencias y requerimientos del usuario.

4.3.3.1.1. Roles

A través de la tabla 12, se describen cada rol con sus respectivas responsabilidades, de acuerdo con el enfoque *Scrum*, siendo estos: *Product Owner* el cual facilita la gestión del *Product Backlog* con el fin de cumplir con los objetivos definidos. *Scrum Master* es el encargado del cumplimiento del marco de trabajo *Scrum*. Por último, el *Development team* se encarga de la elaboración y posterior entrega del producto.

Tabla 12: División de los Roles

Área	Persona	Roles
Propietario	Zoila Caguan	Product Owner
Docente PUCESD	Mg. Willian Ocampo	Scrum Master
Desarrollador, Diseñador y tester	Jaime Ortiz	Development team
Desarrollador, Diseñador y tester	Nathaly Zamora	Development team

4.3.3.1.2. Modelo Vista Controlador

Para la elaboración del producto, se decidió elegir el Modelo Vista Controlador, que tiene como objetivo separar las responsabilidades dentro de una aplicación, esta división es a través de tres componentes: Modelo, Vista y Controlador.

- **Modelo:** Ilustra los datos con los que opera la aplicación y las reglas para manipular esa información.
- **Vista:** Representa el *front-end* de la aplicación, encargado de mostrar información para el usuario de manera legible y atractiva, y de recoger las interacciones del usuario.
- **Controlador:** Funciona como mediador entre los componentes anteriormente mencionados. Procesa las peticiones del usuario, interactúa con el modelo con el fin de obtener los datos, y posteriormente envía la respuesta a la vista apropiada para mostrar los resultados al usuario.

4.3.3.1.3. Parametrización

Durante el desarrollo, con el *Development team* se estableció la parametrización que representa un estándar similar con los elementos del código de la aplicación *web*, con el fin de facilitar al momento de desarrollar. Lo mencionado se ve reflejado en la tabla 13.

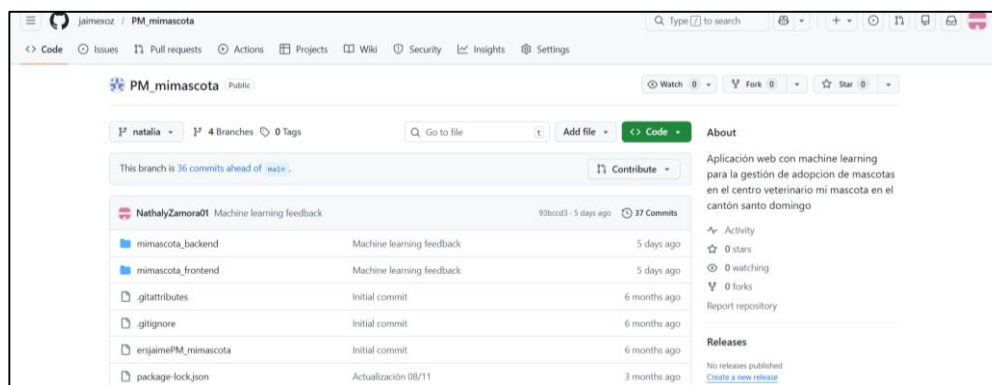
Tabla 13. Parametrización

Modelo	Vista	Controlador
Tablas: Nombre en minúscula, sin tildes, por ejemplo: <i>mascotas</i> <i>caracteristicas</i>	Vistas y componentes: Primera letra en mayúscula de acuerdo a cada palabra, por ejemplo: <i>AuthPage</i>	Contantes: Primera palabra en minúscula, siguientes palabras primera letra en Mayúscula, ejemplo: <i>uploadProfileImage</i>
Tablas intermedias: Nombre en minúscula y unión de los nombres de las dos tablas que referencian, Por ejemplo: <i>mascotas_caracteristicas</i>	Funciones: Primera palabra en minúscula, primera letra de las siguientes palabras en mayúscula, por ejemplo: <i>manejarSubidaArchivo</i>	
Columnas: Nombre en minúscula. Nombre de la columna, guion bajo, nombre la tabla, ambos nombres normalizados en 6 letras, por ejemplo: <i>nombre_mascot</i>		

4.3.3.1.4. Control de versiones

Para llevar un control de las distintas versiones y evitar que no sean compatibles entre los distintos componentes y módulos del proyecto, se optó por utilizar *GitHub* como se visualiza en la figura 10. Esta plataforma permitió llevar un historial detallado de los cambios realizados y versiones anteriores, contribuyendo significativamente al trabajo cooperativo.

Figura 10: Repositorio del sistema de gestión para adopción de mascotas.



4.3.3.1.5. Product Backlog

En este apartado se encuentra establecido las funcionalidades del proyecto por el *Product Owner*, en donde se establece y prioriza todo el trabajo que fue necesario para la elaboración. En la tabla 14 y 15, se encuentran las historias de usuario y la historia técnica que se incluyeron, las cuales están detalladas en el anexo IV.

Tabla 14. Product Backlog

N.	Historias	Estimación	Prioridad negocio	Riesgo de desarrollo
1	HU1-Inicio de sesión	8	100	ALTO
2	HU2-Cierre de sesión	3	100	BAJO
3	HU3-Registro de usuarios	8	100	ALTO
4	HU4-Verificación de cuenta	3	100	MEDIO
5	HU5-Recuperar contraseña	5	100	ALTO
6	HU6- Información de perfil	5	95	MEDIO
7	HU7- Roles de usuario	3	95	MEDIO
8	HU8- Cambio de contraseña	5	90	MEDIO
9	HU9- Edición de perfil	5	90	ALTO
10	HU10-Eliminar cuenta	3	90	ALTO
11	HU11- Agregar mascota	8	90	MEDIO
12	HU12- Eliminar mascota	5	90	ALTO
13	HU13-Editar mascota	8	85	ALTO

14	HU14-Gestión de información de mascotas	5	85	ALTO
15	HU15- Mis publicaciones	5	85	MEDIO
16	HU16- Adopciones realizadas	5	80	BAJO
17	HU17-Visualización de mascotas	8	80	MEDIO
18	HU18-Ficha de información de mascotas	5	80	MEDIO
19	HU19-Favoritos	5	80	ALTO
20	HU20-Solicitud de adopción	5	75	MEDIO
21	HU21-Visualización de solicitudes enviadas	5	75	BAJO
22	HU22-Visualización de solicitudes recibidas	5	75	BAJO
23	HU23-Gestión de solicitud de adopción	5	75	ALTO
24	HU24-Notificación de solicitud	3	70	ALTO
25	HU25-Recomendación de mascotas personalizada	8	70	ALTO
26	HU26-Enviar valoración	3	70	ALTO
27	HU27-Gestión de valoraciones	3	70	MEDIO

Tabla 15. Product Backlog Técnico

N.	Historias	Estimación	Prioridad negocio	Riesgo de desarrollo
1	HT1-Seguridad y privacidad de la aplicación	8	100	ALTO

4.3.3.1.6. Estimación

Para gestionar la estimación, se empleó la métrica Puntos de Historia, que permite estimar cada tarea. Además, se optó por la técnica de la sucesión Fibonacci, ya que permite establecer el nivel de complejidad e incertidumbre de cada una. En este proceso, el *Development team* realiza un análisis de cada historia de usuario y tarea. De acuerdo con el análisis y el dialogo previo, se estimaron los puntos que son necesarios para completar cada una de ellas. En la tabla 16, se define el calendario de trabajo.

Tabla 16. Tiempo de trabajo

Calendario de trabajo			
Mes	Semanas	Días	Horas
1	4	3	4

4.3.3.1.7. Velocidad de desarrollo

El *sprint* inicial abarcó un alcance de 40 puntos de historia. De este total, la primera y tercera historia están ponderadas en 8 puntos cada una; la segunda, cuarta y séptima

estimadas en 3 puntos; y la quinta, sexta y octava historia contaron con 5 puntos, planificadas para un mes.

4.3.3.1.8. Escenarios de prueba

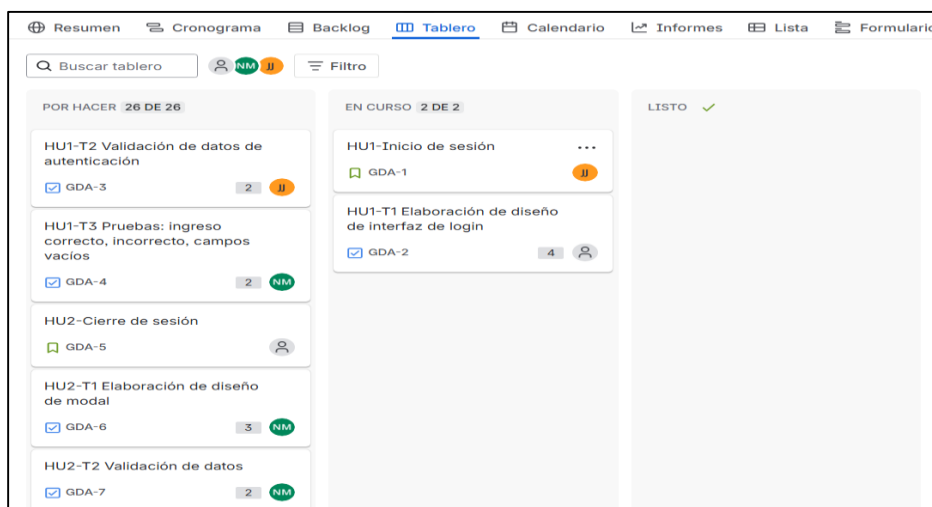
Se desarrollaron las historias de usuario y sus respectivos escenarios de pruebas, las cuales se evidencian en el anexo IV. En la reunión establecida con el *Product Owner*, se examinó cada una de las funcionalidades, de acuerdo a las pruebas de aceptación, detalladas en el anexo V.

4.3.3.1.9. Gestión de tareas de ingeniería

El *software Jira* sirvió como la plataforma elegida para la gestión de incidencias y el seguimiento del progreso, como se ilustra en la figura 11. Esta herramienta permitió llevar una correcta organización y gestión sobre el producto, optimizar el trabajo colaborativo y llevar un control sobre las tareas. La estructura que se empleó en la herramienta es:

- En la primera columna denominada “Por hacer”, se encuentran todas las tareas que aún no han sido puestas en desarrollo.
- En la segunda columna denominada “En curso”, están las tareas que han sido puestas en marcha para realizar de acuerdo al avance del proyecto.
- Por último, la tercera columna “Listo”, están las tareas que ya han sido terminadas.

Figura 11. Tablero del sprint



4.3.3.1.10. *Sprint Backlog*

De acuerdo a la tabla 17, se visualizan las historias de usuario que han sido tomadas en cuenta para la elaboración del primer *sprint backlog*, de las cuales cuentan con una probabilidad alta de desarrollo. Además, se establecieron las tareas vinculadas con cada una de las historias de usuario que se realizaran durante este *sprint*.

Tabla 17. Sprint backlog 1

Sprint Backlog						
Historia	Est	Categoría	Tarea	Responsable	Est	Estado
HU1-Inicio de sesión	8	Diseño	Elaboración de diseño de interfaz de <i>Login</i>	Jaime Ortiz	4	Completado
		Desarrollo	Validación de datos de autenticación	Jaime Ortiz	2	Completado
		Pruebas	Pruebas: validar emisión de evento <i>handleLogin</i> con credenciales válidas.	Nathaly Zamora	2	Completado
HU2-Cierre de sesión	3	Diseño	Elaboración de diseño de modal	Nathaly Zamora	1	Completado
		Desarrollo	Validación de acción	Nathaly Zamora	1	Completado
		Pruebas	Pruebas: verificar que la función de <i>logout</i> elimine correctamente el token del <i>authStore</i>	Nathaly Zamora	1	Completado
HU3-Registro de usuarios	8	Diseño	Elaboración de diseño del formulario de registro	Jaime Ortiz	3	Completado
		Desarrollo	Validación de campos y datos	Nathaly Zamora	2	Completado
		Desarrollo	Encriptación de contraseñas	Jaime Ortiz	2	Completado
HU4-Verificación de cuenta	3	Pruebas	Pruebas: validar emisión de evento <i>register</i>	Nathaly Zamora	1	Completado
		Diseño	Elaboración de diseño de interfaz de verificación	Jaime Ortiz	1	Completado
		Desarrollo	Generación y validación de token de verificación	Nathaly Zamora	1	Completado
HU5-Recuperar contraseña	5	Pruebas	Pruebas: verificar que capture el token y maneje correctamente los estados de respuesta	Nathaly Zamora	1	Completado
		Diseño	Diseño de flujo de recuperación (<i>email</i> + enlace)	Jaime Ortiz	2	Completado
		Desarrollo	Validación de <i>email</i> existente y enviar enlace	Jaime Ortiz	1	Completado

		Desarrollo	Permiso de ingreso de nueva contraseña tras enlace	Nathaly Zamora	1	Completado
		Pruebas	Pruebas: validación de formato de email en el <i>frontend</i> y el flujo de éxito/error	Nathaly Zamora	1	Completado
HU6- Información de perfil	5	Diseño	Elaboración de diseño de interfaz de perfil	Jaime Ortiz	2	Completado
		Desarrollo	Recopilación de información	Jaime Ortiz	1	Completado
		Desarrollo	Asignación de rutas a funcionalidades	Nathaly Zamora	1	Completado
		Pruebas	Pruebas: Verificar que el componente de perfil renderice correctamente la información recibida desde el <i>store</i>	Nathaly Zamora	1	Completado
		Diseño	Elaboración de diseño de interfaz de rol de usuario	Nathaly Zamora	1	Completado
HU7-Roles de usuario	3	Desarrollo	Actualización de información	Nathaly Zamora	1	Completado
		Pruebas	Pruebas: Validar que la interfaz de administración oculte o muestre elementos basados en los permisos del objeto <i>user</i>	Jaime Ortiz	1	Completado
		Diseño	Diseño de formulario para cambio de contraseña	Jaime Ortiz	2	Completado
HU8- Cambio de contraseña	5	Desarrollo	Validación de contraseña actual antes de permitir cambio	Jaime Ortiz	1	Completado
		Desarrollo	Actualización de contraseña en base de datos	Nathaly Zamora	1	Completado
		Pruebas	Pruebas: comparar la contraseña actual y la nueva antes del envío, verificando que el componente bloquee la acción si no coinciden los criterios de seguridad	Nathaly Zamora	1	Completado

4.3.3.2. Reuniones diarias del Sprint 1 - Daily Scrum

Dada la organización que fue establecida en el *sprint*, se realizaron *Dailys Scrum* de 15 minutos, en donde se comentaron sobre las tareas realizadas, que dificultades u obstáculos se encontraron durante el proceso y cuáles serán los siguientes pasos por

realizar. Para realizar un control y seguimiento de las tareas, se implementó el uso del tablero de la herramienta *Jira Software*.

4.3.3.2.1. Historia de usuario I: Iniciar sesión

En el anexo IV, se divide la primera historia, como primer punto se elaboró el diseño de la interfaz de *Login*, por medio del cual los usuarios ya registrados acceden a la aplicación mediante sus credenciales. La lógica se desarrolla utilizando el *framework Vue.js*, por medio del Componente “*LoginForm.vue*”, tal como se observa en la figura 12 y 13.

Figura 12. Interfaz de Login.

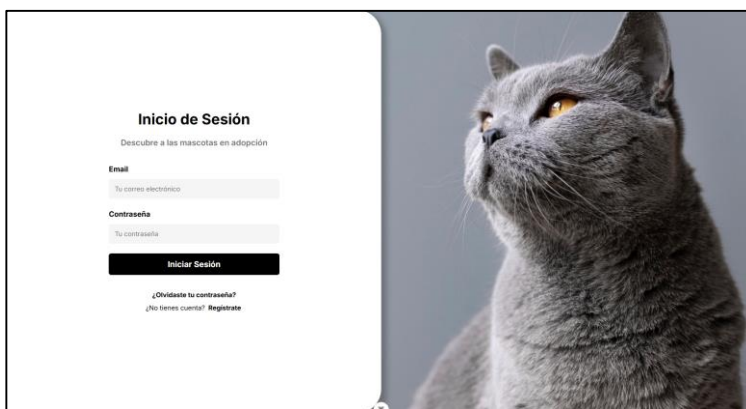


Figura 13. Código de la lógica del Login.

```
const handleLogin = async () => {
  error.value = '';
  loading.value = true;
  try {
    // Axios maneja headers y JSON automáticamente
    const response = await apiClient.post('/auth/login', {
      email_usuario: email.value,
      contra_usuario: password.value
    });
    const data = response.data;
    login(data.token);

    setAuthHeader(data.token);

    // Guardar datos del usuario
    if (data.user) {
      localStorage.setItem('userData', JSON.stringify(data.user));
    }

    router.push('/home');
  } catch (err) {
```

Para la validación de datos, se utilizó un servidor implementado con *Node.js* y *Express.js*, configurado para la autenticación con *JSON Web Tokens (JWT)*. A partir de esta base, la lógica de inicio de sesión se centraliza en el controlador *authController.js*, específicamente en la función “*loginUser*”, que se encarga de manejar la respuesta del servidor a través de “*findUserWithRoleByEmailDB*”. En donde se verifica que el usuario

exista por medio de una consulta realizada a la base de datos. Posteriormente, se genera un *JWT* con la información relevante del usuario. Como se observa en la figura 14 y 15.

Figura 14. Código de la función `loginUser`.

```
const loginUser = async (req, res) => {
  const { email_usuario, contra_usuario } = req.body;

  if (!email_usuario || !contra_usuario) {
    return res.status(400).json({ message: 'Correo y contraseña son requeridos.' });
  }

  try {
    // 1. Obtener usuario y rol (Modelo)
    const result = await authModel.findUserWithRoleByEmailDB(email_usuario);
    const user = result.rows[0];

    if (!user) {
      return res.status(401).json({ message: 'Credenciales inválidas (correo no encontrado).' });
    }

    if (user.eliminado_logico === true) {
      return res.status(401).json({
        message: 'Su cuenta ha sido eliminada. Por favor crea una nueva cuenta.',
        accountDeleted: true
      });
    }

    if (!user.mailve_usuario) {
      return res.status(401).json({ message: 'Por favor, verifica tu correo electrónico para iniciar sesión.' });
    }
  }
}
```

Figura 15. Consulta para recopilar los datos de la cuenta

```
const findUserWithRoleByEmailDB = async (email) => {
  return pool.query(`
    SELECT u.*, r.nombre_rolsx
    FROM usuarios u
    JOIN roles r ON u.idxxxx_rolsx = r.idxxxx_rolsx
    WHERE u.email_usuario = $1
  `, [email]);
};
```

4.3.3.2.2. Historia de usuario II: Cierre de sesión

En el anexo IV, se segmenta la segunda historia, se elaboró el diseño del botón de cierre de sesión. La lógica se realiza por medio del Componente “*NavBar*” y la función “*handleLogout*”. Tal como se observa en la figura 16 y 17.

Figura 16. Botón de cierre de sesión.

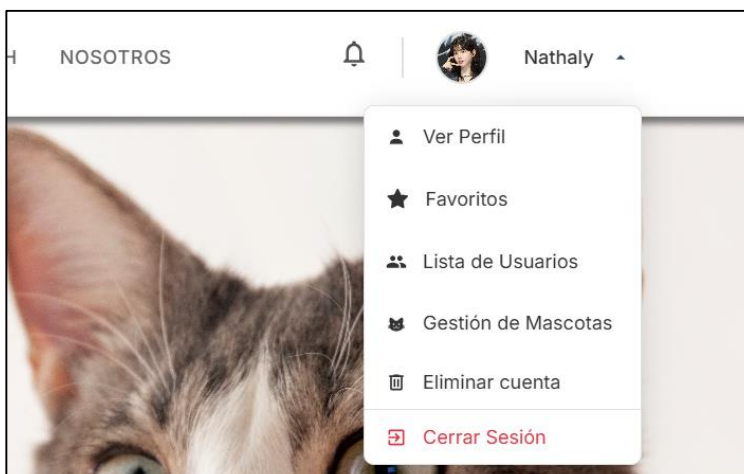


Figura 17. Código de la lógica de cierre de sesión

```

const closeDropdown = () => {
  isDropdownOpen.value = false;
  isNotificationsOpen.value = false;
};

// Manejar Logout
const handleLogout = () => {
  logout();
  closeDropdown();
  router.push({ name: 'auth' });
};

```

4.3.3.2.3. Historia de usuario III: Registro de usuario

La tercera historia se encuentra descrita en el anexo IV, en donde se elaboró el diseño de la interfaz de registro de usuario. Se crearon los campos necesarios para el registro, como nombre, apellido, correo electrónico, contraseña, celular y la opción de subir una imagen de perfil. La lógica del registro de usuario se realiza por medio del Componente “*RegisterForm.vue*”, tal como se observa en la figura 18 y 19.

Figura 18. Interfaz de registro de usuario

The image shows a user registration form on the left and a photograph of a grey cat on the right. The form is titled "Regístrate" and has the subtitle "Crea una cuenta para usar la aplicación". It contains several input fields: "Nombre" (with placeholder "Tu nombre"), "Apellido" (with placeholder "Tu apellido"), "Email" (with placeholder "Tu correo electrónico"), "Contraseña" (with placeholder "Crea tu contraseña"), "Teléfono (opcional)" (with placeholder "Tu número de teléfono"), and "Imagen de perfil (opcional)" (with a "Seleccionar Archivo" button and placeholder "Selecciona archivo seleccionado"). Below the fields is a black "Registrarse" button. At the bottom of the form, there is a link: "¿Ya tienes cuenta? Inicia sesión".

Figura 19. Código de la lógica de registro de usuario.

```

const register = async () => {
  errorMessage.value = '';

  const formData = new FormData();
  formData.append('nombre_usuario', nombre.value);
  formData.append('apellido_usuario', apellido.value);
  formData.append('email_usuario', email.value);
  formData.append('contra_usuario', password.value);
  formData.append('celula_usuario', celular.value);

  if (profileImage.value) {
    formData.append('imagep_usuario', profileImage.value);
  }

  try {
    const response = await axios.post('http://localhost:3000/api/auth/register', formData, {
      headers: {
        'Content-Type': 'multipart/form-data'
      }
    });
  }

  // Emitir el mensaje de éxito al componente padre
  emit('registration-success', response.data.message || 'Registro exitoso. Por favor, revisa tu correo');
};

```

Para la validación de datos, la lógica de registro se centralizó en el controlador *authController.js*, la función encargada de procesar el registro de usuarios es “*RegisterUser*”, la cual envía la información proporcionada por el usuario y encripta la contraseña de usuario a “*createNewUserDB*”. Adicionalmente, el servidor está configurado con *CORS* para permitir peticiones desde el *frontend*, y utiliza la configuración de *multer* junto con *cloudinary* para el procesamiento y almacenamiento de las imágenes de perfil. Como se observa en la figura 20 y 21.

Figura 20. Código de la función registerUser.

```
const registerUser = async (req, res) => {
  const { nombre_usuario, email_usuario, contra_usuario, celula_usuario, apelli_usuario } = req.body;
  let image_usuario = null;

  // 1. Manejo de Archivo
  if (req.file) {
    try {
      const result = await cloudinary.uploader.upload(req.file.path, {
        folder: 'mimascota_perfiles',
        use_filename: true,
        unique_filename: false,
        overwrite: false
      });
      image_usuario = result.secure_url;
      fs.unlinkSync(req.file.path);
    } catch (err) {
      return res.status(500).json({ message: 'Error al subir la imagen a Cloudinary.', error: err.message });
    }
  }

  // 2. Validaciones
  if (!nombre_usuario || !apelli_usuario || !email_usuario || !contra_usuario) {
    return res.status(400).json({ message: 'Por favor, introduce todos los campos requeridos.' });
  }
}
```

Figura 21. Consulta para crear un nuevo usuario

```
const createNewUserDB = async (userData) => {
  const { nombre_usuario, email_usuario, contra_usuario, celula_usuario, image_usuario, verificationToken, verificationExpires, apelli_usuario } = userData;

  const newUser = await pool.query(
    `INSERT INTO usuarios (
      nombre_usuario,
      email_usuario,
      contra_usuario,
      celula_usuario,
      image_usuario,
      idxxxx_rolesx,
      mailve_usuario,
      tokeve_usuario,
      tokeexp_usuario,
      apelli_usuario
    ) VALUES ($1, $2, $3, $4, $5,
      (SELECT idxxxx_rolesx FROM roles WHERE nombre_rolesx = 'Usuario'),
      FALSE, $6, $7, $8)
    RETURNING idxxxx_usuario, nombre_usuario, email_usuario, celula_usuario, image_usuario, mailve_usuario, apelli_usuario,
    [nombre_usuario, email_usuario, contra_usuario, celula_usuario, image_usuario, verificationToken, verificationExpires,
    apelli_usuario]
  `);
  return newUser.rows[0];
};
```

4.3.3.2.4. Historia de usuario IV: Verificación de cuenta

La descripción de la cuarta historia y sus escenarios de prueba están detalladas en el anexo IV, la verificación de cuenta empieza durante el proceso del registro donde el controlador de autenticación, *authController.js*, genera un *token* único y establece una fecha de expiración para este. El *token*, junto con una *URL* de verificación, es enviado al correo

electrónico del usuario recién registrado utilizando *nodemailer*, como se visualiza en la figura 22.

Figura 22. Correo de verificación



El componente que maneja la ruta de verificación procesa la respuesta del *backend* por medio de la función "*verifyEmail*" y muestra un mensaje al usuario indicando el éxito o fracaso de la verificación. Tras una verificación exitosa, el usuario es informado de que ya puede iniciar sesión con su cuenta, como se muestra en la figura 23 y 24.

Figura 23. Interfaz de verificación del correo

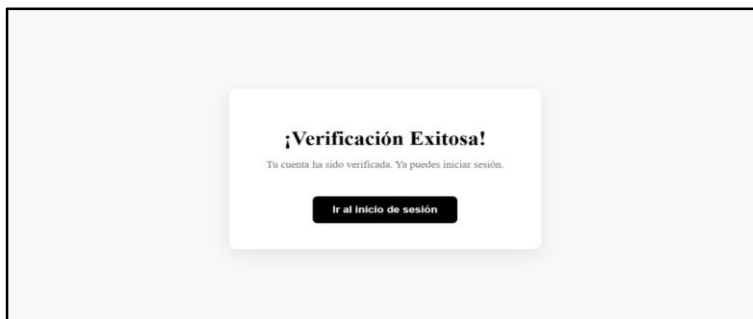


Figura 24. Función de verificación del correo

```
const verifyEmail = async (req, res) => {
  const { token } = req.query;

  if (!token) {
    return res.status(400).json({ message: 'Token de verificación no proporcionado.' });
  }

  try {
    // 1. Buscar usuario (Modelo)
    const result = await authModel.findUserByResetTokenDB(token);
    const userResult = await pool.query(
      'SELECT * FROM usuarios WHERE tokeve_usuario = $1 AND tokexp_usuario > NOW()', [token]
    );

    const user = userResult.rows[0];

    if (!user) {
      return res.status(400).json({ message: 'Token de verificación inválido o expirado.' });
    }

    // 2. Actualizar verificación (Modelo)
    await authModel.verifyEmailDB(user.idxxxx_usuario);

    res.status(200).json({ message: 'Correo electrónico verificado exitosamente. Ya puedes iniciar sesión.' });
  }
};
```

4.3.3.2.5. Historia de usuario V: Recuperar contraseña

En el anexo IV se segmenta la quinta historia, se elaboró el diseño de la interfaz de recuperar contraseña. Se creó el campo para que el usuario pueda ingresar el correo de la cuenta. La lógica de recuperación de contraseña se realiza por medio de la vista “*ResetPassword.vue*” con la función “*handleResetPassword*”, tal como se observa en la figura 25 y 26.

Figura 25. Interfaz de recuperación de contraseña.

Figura 26. Código de la lógica de recuperar contraseña.

```
const handleResetPassword = async () => {
  error.value = '';
  success.value = '';
  loading.value = true;

  // Lógica para solicitar el enlace o para restablecer la contraseña
  if (!token.value) {
    // Lógica para enviar el email de restablecimiento
    try {
      const response = await fetch('http://localhost:3000/api/auth/request-password-reset', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ email_usuario: email.value }),
      });
      const data = await response.json();
      if (response.ok) {
        success.value = data.msg;
      } else {
        error.value = data.msg || 'Error al enviar el enlace.';
      }
    } catch (err) {
      error.value = 'No se pudo conectar con el servidor. Inténtalo de nuevo.';
    } finally {
      loading.value = false;
    }
  } else {
    // Lógica para actualizar la contraseña con el token
    if (newPassword.value !== confirmPassword) {

```

La lógica de recuperación de contraseña se centraliza en el controlador *authController.js*, específicamente en una función “*requestPasswordReset*” por medio de la cual crea un *token* de restablecimiento, que se envía al correo como parte del enlace para restablecer y se observa en la figura 27 y 28.

Figura 27. Correo de restablecimiento de contraseña.



Figura 28. Código de la función requestPasswordReset.

```
const requestPasswordReset = async (req, res) => {
  const { emailx_usuario } = req.body;

  if (!emailx_usuario) {
    return res.status(400).json({ message: 'El correo electrónico es requerido.' });
  }

  try {
    // 1. Buscar usuario (Modelo)
    const result = await authModel.findOne({ email: emailx_usuario });
    const user = result[0];

    if (user) {
      return res.status(200).json({ message: 'Si el correo electrónico existe, se ha enviado un enlace para restablecer la contraseña.' });
    }

    // 2. Generar y guardar token (Modelo)
    const resetToken = uuidv4();
    const resetExpires = new Date(Date.now() + 3600000); // Expira en 1 hora
    await authModel.updateResetTokenDB(resetToken, resetExpires, user.id);

    // 3. Enviar correo
    const resetUrl = `${process.env.FRONTEND_URL}/reset-password?token=${resetToken}`;
    const mailOptions = {
```

4.3.3.2.6. Historia de usuario VI: Información de perfil

La sexta historia se encuentra segmentada en el anexo IV, se elaboró el diseño de la interfaz de información de perfil, se muestra la información del usuario y las funcionalidades que puede gestionar. La lógica del perfil del usuario se realiza por medio de la vista “PerfilUsuario.vue”, en donde se establece el nombre de las rutas para cada una de las funcionalidades. Tal como se observa en la figura 29, 30 y 31.

Figura 29. Interfaz del perfil de usuario.

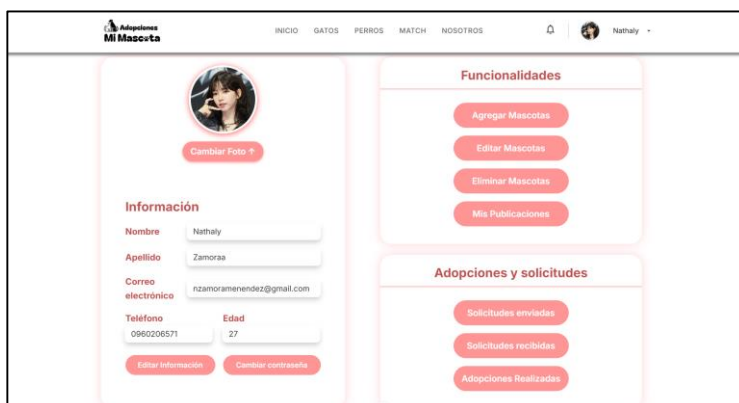


Figura 30. Código de la lógica del perfil de usuario

```
import { getToken, updateUserData, onUserDataChange } from '../utils/auth';

const router = useRouter();
const defaultAvatar = '/default-avatar.png';
const userData = JSON.parse(localStorage.getItem('userData') || '{}');

// Datos de usuario actuales (solo lectura/fuente de verdad)
const userName = ref(userData.nombre || userData.name || '');
const userLastName = ref(userData.apellido || userData.lastname || '');
const userEmail = ref(userData.email || userData.email_usuario || '');
const userPhone = ref(userData.celular || userData.celula_usuario || '');
const userAge = ref(userData.edad || userData.edadxx_usuario || '');
const userImageUrl = ref(userData.imageUrl && userData.imageUrl.startsWith('http') ? userData.imageUrl : defaultAvatar);

// Variables de control
const loading = ref(false);
const isLoading = ref(false); // Para subir foto
const error = ref('');
const success = ref('');
```

Figura 31. Nombre de rutas de las funcionalidades

```
<div class="perfil-card-fun">
  <div class="perfil-col funcionalidades">
    <div class="fun-top">
      <h2>Funcionalidades</h2>
    </div>
    <div class="func-grid">
      <div class="func-column">
        <router-link to="/agregarmascota" class="func-btn">
          Agregar Mascotas
        </router-link>

        <router-link to="/editarmascota" class="func-btn">
          Editar Mascotas
        </router-link>

        <router-link to="/eliminarmascota" class="func-btn">
          Eliminar Mascotas
        </router-link>

        <router-link to="/mypost" class="func-btn">
          Mis Publicaciones
        </router-link>
      </div>
    </div>
  </div>
</div>
```

4.3.3.2.7. Historia de usuario VII: Roles de usuario

En el anexo IV se desglosa la séptima historia, se elaboró el diseño de la interfaz de roles de usuario, se muestra el nombre, apellido, correo, teléfono y el rol que tiene cada usuario. La lógica de la interfaz de roles se realiza por medio del componente “*UserList.vue*”, en donde se encuentra las funciones “*fetchUsuarios*” y “*changeUserRole*”, que permiten obtener y cambiar el rol del usuario. Tal como se observa en la figura 32 y 33.

Figura 32. Interfaz de roles de usuario.

NOMBRE	APELLIDO	CORREO	CELULAR	ROL	ACCIONES
Zola	Caguan	mimascotacta@gmail.com		Admin	
Jaime	Ortiz	dartraiol@gmail.com		Usuario	
Angel	Rodriguez	angelrodi2@gmail.com	0993785179	Usuario	
Nathaly	Zamoraa	nzamoramenendez@gmail.com	0960206571	Admin	
Vanessa	Zambrano	vzambrano1983@outlook.com	0960206571	Empleado	
Nathaly	Zamora	nzamoramenendez@outlook.com		Usuario	

Figura 33. Código de la lógica de cambio de roles.

```
const fetchUsuarios = async () => {
  loading.value = true;
  try {
    const token = getToken();
    if (!token) {
      throw new Error('No hay token de autenticación');
    }

    const response = await fetch('http://localhost:3000/api/auth/usuarios', {
      headers: {
        'Authorization': `Bearer ${token}`,
        'Content-Type': 'application/json'
      }
    });

    if (!response.ok) {
      if (response.status === 403) {
        throw new Error('No tienes permisos para ver la lista de usuarios');
      }
      throw new Error('Error al obtener usuarios');
    }

    usuarios.value = await response.json();
    error.value = null;
  }
}
```

Para la validación de datos, la lógica del cambio de rol se realizó por medio de la función `updateUserInfo` de `authController.js`, la cual se encarga de procesar las respuestas de `updateUserInfo` de `authModel.js` y actualiza la información del usuario en la base de datos. Como se observa en la figura 34 y 35.

Figura 34. Código de la función `updateUserInfo`

```
const updateUserInfo = async (req, res) => {
  try {
    if (!req.user || !req.user.id) {
      return res.status(401).json({ message: 'No autorizado. Se requiere un token de usuario válido' });
    }

    const userId = req.user.id;
    const { nombre, apellido, celular, edad } = req.body;

    console.log('Datos recibidos para actualizar:', { userId, nombre, apellido, celular, edad });

    if (!nombre || nombre.trim() === '' || !apellido || apellido.trim() === '') {
      return res.status(400).json({ message: 'Nombre y Apellido son campos obligatorios.' });
    }

    let edadValue = null;
    if (edad !== undefined && edad !== null && edad !== '') {
      edadValue = parseInt(edad, 10);
      if (isNaN(edadValue) || edadValue < 1) {
        edadValue = null;
      }
    }

    // Normalizar celular (puede ser null o string vacío)
  }
}
```

Figura 35. Consulta para actualizar la información del usuario

```

const updateUserInfo = async (userId, data) => {
  const { nombre, apellido, celular, edadValue } = data;
  return pool.query(
    `UPDATE usuarios SET
      nombre_usuario = $1,
      apelli_usuario = $2,
      celula_usuario = $3,
      edadxx_usuario = $4,
      feactu_usuario = NOW()
    WHERE idxxxx_usuario = $5
    RETURNING *`,
    [nombre, apellido, celular, edadValue, userId]
  );
};

```

4.3.3.2.8. Historia de usuario VIII: Cambio de contraseña

En el anexo IV se segmenta la octava historia, se elaboró el diseño del modal de cambio de contraseña, permite ingresar la contraseña actual, la nueva contraseña y confirmar la contraseña ingresada. La lógica del cambio de contraseña se realiza por medio de la vista “*PerfilUsuario.vue*”, en la función “*changePassword*”, que permite validar la información antes de que se envíen los datos. Tal como se observa en la figura 36 y 37.

Figura 36. Modal de cambio de contraseña.

Figura 37. Código de la lógica de cambio de contraseña.

```

const changePassword = async () => {
  passwordError.value = '';
  passwordSuccess.value = '';

  if (passwordData.value.newPassword !== passwordData.value.confirmPassword) {
    passwordError.value = 'Las contraseñas nuevas no coinciden.';
    return;
  }

  if (passwordData.value.newPassword.length < 6) {
    passwordError.value = 'La nueva contraseña debe tener al menos 6 caracteres.';
    return;
  }

  changingPassword.value = true;

  try {
    const token = getToken();
    if (!token) {
      throw new Error('No hay token de autenticación');
    }
  }

  const response = await fetch('http://localhost:3000/api/auth/change-password', {
    method: 'PUT',
    headers: {
      'Content-Type': 'application/json',
      'Authorization': `Bearer ${token}`
    }
  });
};

```

La función encargada de procesar la lógica del cambio de contraseña es “changePassword”, la cual maneja las respuestas. Por medio de “getCurrentHashDB” se obtiene la contraseña actual encriptada del usuario y se compara con la ingresada por el usuario, una vez la información ingresada sea correcta entonces por medio “changeUserPasswordDB” se actualiza la contraseña en la base de datos. Como se observa en la figura 38 y 39.

Figura 38. Código de la función changePassword

```
const changePassword = async (req, res) => {
  try {
    const { currentPassword, newPassword } = req.body;

    if (!currentPassword || !newPassword) {
      return res.status(400).json({ message: 'Contraseña actual y nueva contraseña son requeridas.' });
    }

    if (newPassword.length < 6) {
      return res.status(400).json({ message: 'La nueva contraseña debe tener al menos 6 caracteres.' });
    }

    // 1. Obtener hash actual (Modelo)
    const userResult = await authModel.getCurrentHashDB(req.user.id);
    const user = userResult.rows[0];

    if (!user) {
      return res.status(404).json({ message: 'Usuario no encontrado.' });
    }
  }
};
```

Figura 39. Código de actualización de contraseña

```
const changeUserPasswordDB = async (userId, hashedPassword) => {
  await pool.query(
    'UPDATE usuarios SET contra_usuario = $1 WHERE idxxxx_usuario = $2',
    [hashedPassword, userId]
  );
};
```

4.3.3.3. Gráfico de trabajo pendiente del Sprint I

En la figura 40, se puede visualizar el trabajo pendiente de las tareas definidas al inicio del *sprint*, mediante *Jira Software* se generó la gráfica del trabajo restante, lo que permitió estar al tanto del progreso de las actividades a realizar.

Figura 40. Gráfica de trabajo restante del sprint 1 en Jira



4.3.3.4. Sprint 1 - Revisión (Sprint Review)

Al finalizar el desarrollo del *sprint* número uno, se procedió a realizar una reunión con todo el equipo de desarrollo y el gerente del centro veterinario, el fin de esta es poder presentar el avance del producto funcional y sobre todo validar el cumplimiento de los objetivos que se han establecido los cuales se centraban en el módulo de autenticación y gestión de usuarios.

Para la supervisión del progreso, se utilizó la herramienta *Jira*, donde se gestionaron las tareas y se generó el Gráfico de Trabajo Pendiente (ver figura 40), el cual evidenció la finalización de las actividades.

4.3.3.5. Sprint 1 - Retrospectiva (Sprint Retrospective)

Se tomaron en cuenta tres preguntas para la retrospectiva de los *sprints*, las cuales permitirán tener un idea clara y concisa de cómo se llevó el primer *sprint*. Como se visualiza en la tabla 18.

Tabla 18. Sprint 1 retrospectiva

Aspectos exitosos del sprint	Mejoras para aplicar	Errores en el sprint
Exitosa asignación de tareas entre los miembros del equipo, Se consiguió una buena sinergia para integrar <i>vue.js</i> con <i>node.js</i> , logrando una comunicación perfecta entre el <i>backend</i> y el <i>frontrend</i> .	Se ha decidido definir un protocolo más estricto para el <i>merging</i> y la revisión de código en futuros <i>sprints</i> .	Se detectaron problemas en la integración con <i>Cloudinary</i> durante el registro de usuarios (HU3). Además, se presentó un error en cada intento de actualización de la foto de perfil (HU6).

4.3.4. Sprint 2

4.3.4.1. Sprint 2 – Planificación (Sprint Planning)

De acuerdo a la tabla 19, se presentan las historias de usuario que han sido tomadas en cuenta para el desarrollo del segundo *sprint backlog*, de las cuales cuentan con una probabilidad alta de desarrollo. Además, se desarrollaron las tareas para cada una de las historias de usuario que se realizan durante este *sprint*.

Tabla 19. Sprint backlog 2

<i>Sprint Backlog</i>						
<i>Historia</i>	<i>Est</i>	<i>Categoría</i>	<i>Tarea</i>	<i>Responsable</i>	<i>Est</i>	<i>Estado</i>
HU9- Edición de perfil	5	Diseño	Diseño en el formulario de perfil para edición	Jaime Ortiz	2	Completado
		Desarrollo	Lógica de código y validación de datos ingresados	Jaime Ortiz	1	Completado
		Desarrollo	Actualización de información	Nathaly Zamora	1	Completado
		Pruebas	Pruebas: verificar que la función de actualización emita los datos correctos y que el estado global se sincronice tras el éxito.	Nathaly Zamora	1	Completado
HU10- Eliminar cuenta	3	Diseño	Elaboración de diseño de botón y cuadro de dialogo	Nathaly Zamora	1	Completado
		Desarrollo	Eliminación de cuenta	Jaime Ortiz	1	Completado
		Pruebas	Pruebas: comprobar la lógica del cuadro de diálogo y asegurar que la función de borrado solo se dispare al confirmar la acción.	Nathaly Zamora	1	Completado
HU11- Agregar mascota	8	Diseño	Creación de formulario para agregar publicación (info + fotos)	Jaime Ortiz	3	Completado
		Desarrollo	Lógica de código y Validación de datos	Jaime Ortiz	2	Completado
		Desarrollo	Guardado de información	Jaime Ortiz	2	Completado
		Pruebas	Pruebas: validar la carga de archivos (fotos) y que el objeto <i>FormData</i> se construya correctamente para la petición.	Jaime Ortiz	1	Completado
HU12- Eliminar mascota	5	Diseño	Elaboración de diseño de interfaz y cuadro de dialogo	Jaime Ortiz	2	Completado
		Desarrollo	Integrar la opción de eliminar en la interfaz de usuario.	Jaime Ortiz	1	Completado
		Desarrollo	Actualización de campos al eliminar	Jaime Ortiz	1	Completado

		Pruebas	Pruebas: verificar que el componente elimine la instancia correcta de la lista reactiva tras recibir la confirmación del servidor.	Jaime Ortiz	1	Completado
HU13- Editar mascota	8	Diseño	Elaboración de diseño de interfaz y formulario de editar	Nathaly Zamora	2	Completado
		Desarrollo	Lógica de código para editar campos específicos	Nathaly Zamora	2	Completado
		Desarrollo	Validación de datos	Nathaly Zamora	2	Completado
		Desarrollo	Actualización de información	Nathaly Zamora	1	Completado
		Pruebas	Pruebas: verificar que el formulario de edición se pre-rellene correctamente con los datos actuales de la mascota	Nathaly Zamora	1	Completado
HU14- Gestión de información de mascotas	5	Diseño	Diseño de interfaz de gestión	Nathaly Zamora	2	Completado
		Desarrollo	Lógica de código y recolección de datos de mascotas	Nathaly Zamora	1	Completado
		Desarrollo	Actualización de cambio de estado	Nathaly Zamora	1	Completado
		Pruebas	Pruebas: validar el cambio de estado y la actualización correspondiente en la interfaz de gestión	Nathaly Zamora	1	Completado
HU15- Mis publicaciones	5	Diseño	Diseño de la interfaz de publicaciones	Nathaly Zamora	2	Completado
		Desarrollo	Implementación de página para mostrar la lista de mascotas del usuario	Jaime Ortiz	1	Completado
		Desarrollo	Recolección de información necesaria de las mascotas	Jaime Ortiz	1	Completado
		Pruebas	Pruebas: verificar que el componente de listado maneje correctamente los estados de carga	Nathaly Zamora	1	Completado
HU16- Adopciones Realizadas	5	Diseño	Diseño de la interfaz de publicaciones (Mis publicaciones)	Nathaly Zamora	2	Completado
		Desarrollo	Implementación de página para mostrar	Jaime Ortiz	1	Completado

		la lista de mascotas adoptadas por el usuario				
	Desarrollo	Recolección de información necesaria de las mascotas	Jaime Ortiz	1	Completado	
	Pruebas	Pruebas: carga correcta, vacía, error al cargar	Nathaly Zamora	1	Completado	
HU17- Visualización de Mascotas	8	Diseño	Elaboración de diseño de listado de mascotas (perros y gatos)	Jaime Ortiz	3	Completado
		Desarrollo	Implementación de página para mostrar la lista de mascotas	Jaime Ortiz	2	Completado
		Desarrollo	Recolección de información necesaria de las mascotas	Jaime Ortiz	2	Completado
		Pruebas	Pruebas: testear el filtrado reactivo de la lista y asegurar que cada tarjeta de mascota reciba los datos correctos del array principal	Jaime Ortiz	1	Completado

4.3.4.1.1. Historia 9: Edición de perfil

En el anexo IV, se presenta la novena historia, permite editar la información personal del usuario, exceptuando el correo electrónico. La función “*saveChanges*”, permite enviar la información que se ha editado al controlador. Tal como se observa en la figura 41 y 42.

Figura 41. Diseño de la edición del perfil.

The screenshot shows a web interface for 'Adopciones MI Mascota'. At the top, there is a navigation bar with links for 'INICIO', 'GATOS', 'PERROS', 'MATCH', and 'NOSOTROS', along with a user profile icon for 'Nathaly'. The main content area is divided into two columns. The left column contains a profile picture with a 'Cambiar Foto' button, followed by a form titled 'Información' with fields for 'Nombre' (Nathaly), 'Apellido' (Zamorichis), 'Correo electrónico' (nzamoramenendez@gmail.com), 'Teléfono' (0960206571), and 'Edad' (54). Below these fields are buttons for 'Guardar', 'Cancelar', and 'Cambiar contraseña'. The right column contains two sections: 'Funcionalidades' with buttons for 'Agregar Mascotas', 'Editar Mascotas', 'Eliminar Mascotas', and 'Mis Publicaciones'; and 'Adopciones y solicitudes' with buttons for 'Solicitudes enviadas', 'Solicitudes recibidas', and 'Adopciones Realizadas'.

Figura 42. Código de la lógica de edición de perfil

```

const saveChanges = async () => {
  // Basic validation
  if (!editableData.value.name || !editableData.value.lastname) {
    error.value = 'El nombre y el apellido son obligatorios.';
    return;
  }
  if (
    editableData.value.name === userName.value &&
    editableData.value.lastname === userLastname.value &&
    editableData.value.phone === userPhone.value &&
    editableData.value.age === userAge.value
  ) {
    error.value = 'No se detectaron cambios. Cancelando edición.'; autoHideSuccess();
    isEditing.value = false; // Salir del modo edición si no hay cambios
    return;
  }
  loading.value = true;
  isloading.value = false;
  error.value = '';
  success.value = '';
  try {
    const token = getToken();
  }
}

```

Para llevar el control de la información, la función encargada de procesar esta solicitud es *“updateUserInfo”* de *authController.js*, la cual envía los datos recibidos a la función de *“updateUserInfo”* de *authModel.js*, que actualiza la información en la base de datos. Como se observa en la figura 43 y 44.

Figura 43. Código de la función updateUserInfo

```

// @desc Actualizar información básica del usuario
// @route PUT /api/auth/update-user-info
// @access Privado
const updateUserInfo = async (req, res) => {
  try {
    if (!req.user || !req.user.id) {
      return res.status(401).json({ message: 'No autorizado. Se requiere un token de usuario válido.' });
    }

    const userId = req.user.id;
    const { nombre, apellido, celular, edad } = req.body;

    console.log('Datos recibidos para actualizar:', { userId, nombre, apellido, celular, edad });

    if (!nombre || nombre.trim() === '' || !apellido || apellido.trim() === '') {
      return res.status(400).json({ message: 'Nombre y Apellido son campos obligatorios.' });
    }

    let edadValue = null;
    if (edad !== undefined && edad !== null && edad !== '') {
      edadValue = parseInt(edad, 10);
      if (isNaN(edadValue) || edadValue < 1) {
        edadValue = null;
      }
    }

    // Normalizar celular (puede ser null o string vacío)
    const celularValue = celular && celular.trim() !== '' ? celular.trim() : null;

    // 1. Actualizar en DB (Modelo)
    const updateResult = await authModel.updateUser(
      nombre: nombre.trim(),
      apellido: apellido.trim(),

```

Figura 44. Consulta para actualizar la información

```

const updateUserInfo = async (userId, data) => {
  const { nombre, apellido, celular, edadValue } = data;
  return pool.query(
    `UPDATE usuarios SET
     nombre_usuario = $1,
     apelli_usuario = $2,
     celula_usuario = $3,
     edadxx_usuario = $4,
     feactu_usuario = NOW()
    WHERE idxxxx_usuario = $5
    RETURNING *`,
    [nombre, apellido, celular, edadValue, userId]
  );
};

```

4.3.4.1.2. Historia 10: Eliminar cuenta

Para la décima historia que se encuentra desglosada en el anexo IV, se elaboró el diseño de la interfaz de eliminar cuenta, que permite deshabilitar la cuenta del usuario. La lógica de eliminación de cuenta se realiza por medio de la función “*handleDeleteAccount*”, que permite mandar una alerta de verificación de la acción. Tal como se observa en la figura 45 y 46.

Figura 45. Diseño de eliminar cuenta.

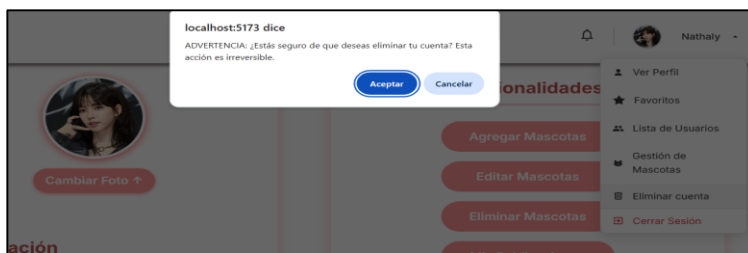


Figura 46. Código de la lógica de eliminar cuenta.

```
const handleDeleteAccount = async () => {
  // 1. Confirmación del usuario
  const confirmation = confirm("ADVERTENCIA: ¿Estás seguro de que deseas eliminar tu cuenta? Esta acción es irreversible.");
  if (!confirmation) {
    return; // Cancelar si el usuario no confirma
  }

  const token = localStorage.getItem('authToken');
  if (!token) {
    alert("Sesión no válida. Por favor, inicia sesión de nuevo.");
    handleLogout(); // Forzar cierre de sesión
    return;
  }

  try {
    // AJUSTA ESTA RUTA a tu backend
    const response = await fetch('http://localhost:3000/api/auth/delete-account', {
      method: 'PUT', // Usamos PATCH para una actualización parcial (cambiar un campo)
      headers: {
        'Authorization': `Bearer ${token}`
      }
    });

    if (response.ok) {
      alert("Tu cuenta ha sido eliminada lógicamente. ¡Gracias por usar nuestros servicios!");
      handleLogout(); // Cerrar sesión después de la eliminación exitosa
    } else {
      // Manejar errores de la API
    }
  }
};
```

La lógica de eliminar cuenta se realiza por medio de la función “*deleteAccountLogic*” que controlara las respuestas. Posteriormente a través de la función “*softDeleteUser*” se aplica un eliminado lógico a la cuenta del usuario en la base de datos. Como se observa en la figura 47 y 48.

Figura 47. Código de la función deleteAccountLogic

```
const deleteAccountLogic = async (req, res) => {
  // EL ID del usuario se obtiene del token inyectada por el middleware 'protect'
  const userId = req.user.id;

  console.log(userId);
  try {
    const success = await authModel.softDeleteUser(userId);

    if (!success) {
      // Esto podría suceder si el ID no existe o ya está eliminado
      return res.status(404).json({ message: "Usuario no encontrado o ya eliminado." });
    }

    // 200 OK - Eliminación Lógica Exitosa
    res.status(200).json({ message: "Cuenta eliminada lógicamente con éxito." });
  } catch (error) {
    console.error('Error al realizar la eliminación lógica del usuario:', error);
    res.status(500).json({ message: "Error interno del servidor al procesar la eliminación." });
  }
};
```

Figura 48. Consulta para aplicar un eliminado lógico de la cuenta

```
const softDeleteUser = async (userId) => {
  return pool.query(
    'UPDATE usuarios SET eliminado_logico = TRUE WHERE idxxxx_usuario = $1',
    [userId]
  );
}
```

4.3.4.1.3. Historia 11: Agregar mascota

En el anexo IV, se presenta la decimoprimer historia, se elaboró el diseño del formulario para agregar la mascota, en donde se puede ingresar la información de la mascota como nombre, especie, sexo, edad, personalidad, imágenes, etc. La lógica de agregar mascota se realiza por medio de la vista “AddMascot.vue”, en la función “publicarMascota”, que permite recolectar los datos del formulario. Tal como se observa en la figura 49 y 50.

Figura 49. Interfaz de agregar mascota

Figura 50. Código de la lógica de agregar mascota

```
async function publicarMascota() {
  const formData = new FormData();

  // 1. Obtener el token de autenticación
  const token = localStorage.getItem('authToken');

  if (!token) {
    alert('Debes iniciar sesión para publicar una mascota.');
```

```
    useRouter().push('/login');
    return;
  }

  // 2. Añadir Los MÚLTIPLES archivos de imagen
  archivosSubidos.forEach((item, index) => {
    formData.append('fotos', item.file);
  });

  // 3. Añadir todos Los campos de texto como un string JSON
  formData.append('datos', JSON.stringify({
    nombre: mascota.nombre,
    // CORRECCIÓN ANTERIOR MANTENIDA: 'mascota.especi' a 'mascota.especie'
    especie: mascota.especie,
    sexo: mascota.sexo,
    edad: mascota.edad,
    raza: mascota.raza,
    peso: mascota.peso,
  }));
}
```

A fin de validar los datos, la lógica de agregar mascota se centralizó en el controlador *mascotaController.js*, la función encargada de procesar esta solicitud es “*createMascota*”. Posteriormente, se envía a “*insertNewMascotaDB*” en donde se ingresa la información de la mascota en la base de datos. Como se observa en la figura 51 y 52.

Figura 51. Código de la función *createMascota*

```

exports.createMascota = async (req, res) => {
  const forane_usuario_id = req.user.id;
  const localImagePaths = req.files ? req.files.map(file => file.path) : [];
  let datosMascota;

  try {
    datosMascota = JSON.parse(req.body.datos);
  } catch (e) {
    cleanupUploadedFiles(localImagePaths);
    return res.status(400).json({ mensaje: 'Formato de datos de la mascota inválido (JSON no válido).' });
  }

  // 1. Subir Imágenes
  let cloudinaryImageUrls = [];
  try {
    const uploadPromises = localImagePaths.map(path => uploadToCloudinary(path, 'mascotas_para_adopcion'));
    cloudinaryImageUrls = await Promise.all(uploadPromises);

    if (cloudinaryImageUrls.includes(null)) {
      throw new Error("Una o más imágenes fallaron al subir a Cloudinary.");
    }
  } catch (uploadError) {
    console.error('Fallo grave durante la subida de imágenes a Cloudinary:', uploadError);
    return res.status(500).json({ mensaje: 'Error al subir una o más imágenes a Cloudinary.' });
  } finally {
    cleanupUploadedFiles(localImagePaths);
  }

  const client = await pool.connect();

```

Figura 52. Consulta para ingresar la información de la mascota

```

const insertNewMascotaDB = async (client, petData, cloudinaryUrls) => {
  const {
    nombre, especie, sexo, edad, raza, peso, tamano, informacionAdicional, forane_usuario_id
  } = petData;
  const [image1_mascot, image2_mascot, image3_mascot] = cloudinaryUrls;
  const infoad_mascot = informacionAdicional || 'N/A';

  const currentTimestamp = new Date().toISOString();

  const insertQuery = `
  INSERT INTO mascotas (
    nombre_mascot, especi_mascot, sexoxm_mascot, edadme_mascot,
    razaxm_mascot, pesokg_mascot, tamano_mascot, infoad_mascot,
    image1_mascot, image2_mascot, image3_mascot, eliminado_logico,
    forane_usuario_id, aprov_mascot, fechap_mascot
  ) VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9, $10, $11, $12, $13, $14, $15)
  RETURNING idxxxx_mascot;
  `;

  const insertValues = [
    nombre, especie, sexo, edad, raza, peso, tamano, infoad_mascot,
    image1_mascot, image2_mascot, image3_mascot, false, forane_usuario_id, 'Pendiente', currentTimestamp
  ];

  const result = await client.query(insertQuery, insertValues);
  const mascotaId = result.rows[0].idxxxx_mascot;
  return { mascotaId, caracteristicasNombres: petData.personalidad };
};

```

4.3.4.1.4. Historia 12: Eliminar mascota

Para la estructura de la decimosegunda historia que se divide en el anexo IV, se elaboró el diseño de la interfaz de eliminar mascota, que permite eliminar la información de la mascota deseada. La lógica del eliminar mascota se realiza por medio de la vista “*DeletePost.vue*”, en la función “*executeDeletePost*”, que envía el código de la mascota. Tal como se observa en la figura 53 y 54.

Figura 53. Interfaz de eliminar mascota.

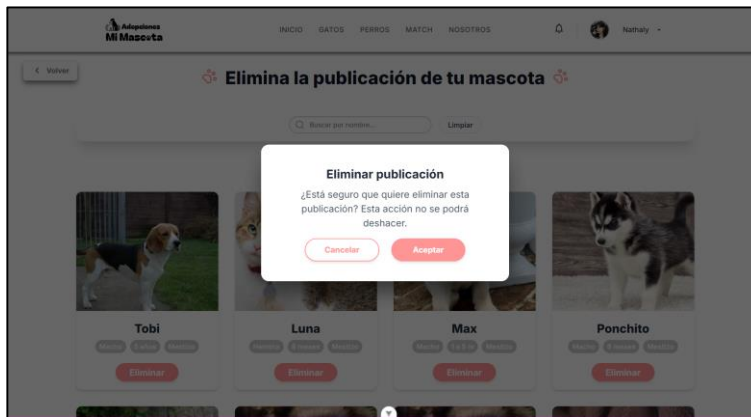


Figura 54. Código de la lógica de eliminar mascota.

```
const executeDeletePost = async () => {
  if (!postIdToDelete.value || !isDeletingPost.value) return;

  isDeletingPost.value = true; // Activa el spinner en el botón
  const token = authStore.token;

  if (!token) {
    alert("Sesión expirada. Por favor, vuelve a iniciar sesión.");
    closeDeleteModal();
    return;
  }

  try {
    // Ejecutar la petición de eliminación usando Axios
    await apiClient.delete(`/mascotas/eliminar/${postIdToDelete.value}`);

    // 6. Manejo de éxito
    alert("¡Publicación eliminada con éxito!");

    // Cierra la modal
    closeDeleteModal();
  }
}
```

Para la lógica de eliminar mascota, la función encargada de procesar la solicitud es “*deleteMascota*”, en donde a través de “*softDeleteMascotaDB*” se aplica un eliminado lógico a la mascota en la base de datos. Como se observa en la figura 55 y 56.

Figura 55. Código de la función deleteMascota

```
exports.deleteMascota = async (req, res) => {
  const petId = req.params.id;
  const userId = req.user.id;

  if (isNaN(petId)) {
    return res.status(400).json({ mensaje: 'ID de mascota inválido.' });
  }

  const client = await pool.connect();
  try {
    // Lógica de DB movida al modelo
    const result = await mascotaModel.softDeleteMascotaDB(client, petId, userId);

    if (result.rowCount === 0) {
      return res.status(404).json({
        mensaje: 'Mascota no encontrada, ya eliminada o no tienes permiso para realizar esta acción.'
      });
    }

    res.status(200).json({
      mensaje: 'Publicación eliminada lógicamente con éxito.',
      id: result.rows[0].idxxx_mascot
    });
  } catch (dbError) {
```

Figura 56. Consulta para aplicar un eliminado lógico a la mascota

```

const softDeleteMascotaDB = async (client, petId, userId) => {
  const updateQuery = `
    UPDATE mascotas
    SET
      eliminado_logico = TRUE,
      fecha_eliminacion = NOW()
    WHERE
      idxxxx_mascot = $1
      AND forane_usuario_id = $2
      AND eliminado_logico = FALSE
    RETURNING idxxxx_mascot;
  `;
  return client.query(updateQuery, [petId, userId]);
};

```

4.3.4.1.5. Historia 13: Editar mascota

En el anexo IV se segmenta la decimotercera historia, se elaboró el diseño de la interfaz de editar mascota. Permite editar toda la información y quitar o agregar nuevas imágenes de la mascota. La lógica del cambio de contraseña se realiza por medio del componente “*EditPostModal.vue*”, en la función “*saveChanges*”, que permite validar la información antes de que se envíen los datos. Tal como se observa en la figura 57 y 58.

Figura 57. Interfaz de editar mascota

Figura 58. Código de la lógica de editar mascota

```

async function saveChanges() {
  const userToken = authStore.token;
  if (!userToken || !isSaving.value) return;
  const MAX_FILE_SIZE_BYTES = 5 * 1024 * 1024; // 5 MB

  isSaving.value = true;

  // 1. Prepara el objeto de datos para enviar
  const dataToSend = {
    ...formData, // Incluye todos los campos de texto: nombre_mascot, especi_mascot, etc.
    // Las URLs de las imágenes existentes que se desean mantener.
    image1_mascot: '',
    image2_mascot: '',
    image3_mascot: ''
  };

  // 2. Prepara FormData para manejar archivos binarios
  const form = new FormData();
  const paddedImages = [...localImages.value, ...Array(3 - localImages.value.length).fill('')];

  // 3. Itera sobre las 3 posibles imágenes

```

Para el control de la lógica de editar las mascotas se utilizó la función “*updateMascota*” que es la encargada de procesar la petición. Posteriormente, llama a la función “*updateMascotaPrincipalDB*” en donde se actualiza la información de la mascota en la base de datos. Como se observa en la figura 59 y 60.

Figura 59. Código de la función *updateMascota*

```

exports.updateMascota = async (req, res) => {
  const petId = req.params.id;
  const localImagePaths = req.files ? req.files.map(file => file.path) : [];
  let datosMascota;

  try {
    datosMascota = JSON.parse(req.body.datos);
  } catch (e) {
    cleanupUploadedFiles(localImagePaths);
    return res.status(400).json({ mensaje: 'Formato de datos de la mascota inválido (JSON no válido)' });
  }

  // 1. Subir nuevas imágenes si existen
  let newCloudinaryUrls = [];
  if (localImagePaths.length > 0) {
    try {
      const uploadPromises = localImagePaths.map(path => uploadToCloudinary(path, 'mascota'));
      newCloudinaryUrls = await Promise.all(uploadPromises);

      if (newCloudinaryUrls.includes(null)) {
        throw new Error("Una o más imágenes fallaron al subir a Cloudinary.");
      }
    } catch (uploadError) {
      cleanupUploadedFiles(localImagePaths);
      console.error('Fallo durante la subida de imágenes para actualización:', uploadError);
      return res.status(500).json({ mensaje: 'Error al subir una o más imágenes nuevas.' });
    }
  }
}

```

Figura 60. Consulta para actualizar la información de la mascota.

```

const updateMascotaPrincipalDB = async (client, petId, petData, finalImageUrls) => {
  const {
    nombre_mascot, especi_mascot, sexo_mascot, edad_mascot,
    raza_mascot, peso_mascot, tamaño_mascot, infoad_mascot,
    image1_mascot: existingUrl1,
    image2_mascot: existingUrl2,
    image3_mascot: existingUrl3,
    status_mascot
  } = petData;

  let urlIndex = 0;
  const isValidCloudinaryUrl = (url) => url && typeof url === 'string' && url.startsWith('http');

  const finalImage1 = isValidCloudinaryUrl(existingUrl1) ? existingUrl1 : (finalImageUrls[urlIndex] || null);
  const finalImage2 = isValidCloudinaryUrl(existingUrl2) ? existingUrl2 : (finalImageUrls[urlIndex + 1] || null);
  const finalImage3 = isValidCloudinaryUrl(existingUrl3) ? existingUrl3 : (finalImageUrls[urlIndex + 2] || null);

  const updateQuery = `
    UPDATE mascotas
    SET
  `;
}

```

4.3.4.1.6. Historia 14: Gestión de información de mascotas

La decimocuarta historia se divide en el anexo IV, en esta parte se elaboró el diseño de la interfaz de gestión de información de mascotas. Permite llevar un control sobre las publicaciones de los usuarios, aceptando o rechazando las mismas. La lógica de gestión se realiza por medio de la vista “*petManagement.vue*”, en la función “*updateApprovalStatus*”, que permite validar la información antes de que se envíen los datos. Tal como se observa en la figura 61 y 62.

Figura 61. Interfaz de gestión de información de las mascotas



Figura 62. Consulta para actualizar el estado de la publicación de la mascota

```
const updateApprovalStatus = async (mascotId, newStatus) => {
  let userToken = authStore.token; // Intenta obtener el token de la tienda

  if (!userToken) {
    // Si no está en la tienda (aún no cargado), búscalo en LocalStorage
    const storedToken = localStorage.getItem('authToken');

    if (storedToken) {
      userToken = storedToken; // Usa el token almacenado para el fetch
    } else {
      console.error("No hay token para actualizar estado de aprobación.");
      router.push({ name: 'auth' });
      return;
    }
  }

  const postIndex = posts.value.findIndex(p => p.idxxxx_mascot === mascotId);
  if (postIndex === -1) {
    console.warn('Post de mascota con ID ${mascotId} no encontrado localmente.');
    return;
  }

  const oldStatus = posts.value[postIndex].approv_mascot;

  // # OPTIMISTA: Actualiza la interfaz inmediatamente
  posts.value[postIndex].approv_mascot = newStatus;

  try {
    const response = await fetch('http://localhost:3000/api/mascotas/approval/${mascotId}', {
      method: 'PUT', // Usamos PUT como se sugirió en el ejemplo anterior.
      headers: {

```

Para la gestión de información de las mascotas, la función *“updateMascotaApprovalStatus”* es la encargada de manejar esta solicitud por medio de la llamada a la función *“updateMascotaApproval”*, en donde se actualiza el campo de estado de la publicación en la base de datos. Como se observa en la figura 63 y 64.

Figura 63. Código de la función updateMascotaApprovalStatus

```
exports.updateMascotaApprovalStatus = async (req, res) => {
  // La verificación de rol debe ir aquí también
  const mascotaId = req.params.mascotId;
  const { approv_mascot } = req.body;

  // 1. Validar el estado
  const validStatuses = ['Pendiente', 'Aprobada', 'Rechazada'];
  if (!validStatuses.includes(approv_mascot)) {
    return res.status(400).json({ message: "Estado de aprobación inválido." });
  }

  try {
    const updatedMascota = await mascotaModel.updateMascotaApproval(mascotaId, approv_mascot);

    if (!updatedMascota) {
      return res.status(404).json({ message: "Mascota no encontrada para actualizar." });
    }

    res.status(200).json({
      message: 'Estado de aprobación actualizado a ${approv_mascot}',
      mascota: updatedMascota
    });
  } catch (error) {
    console.error('Error al actualizar estado de aprobación:', error);
    res.status(500).json({ message: "Error interno del servidor al actualizar el estado." });
  }
};
```

Figura 64. Consulta para actualizar el estado de la publicación.

```

async function updateMascotaApproval(mascotaId, newStatus) {
  const query = `
    UPDATE mascotas
    SET approv_mascot = $1
    WHERE idxxxx_mascot = $2
    RETURNING *;
  `;

  const result = await pool.query(query, [newStatus, mascotaId]);
  return result.rows[0]; // Devuelve el registro actualizado
}

```

4.3.4.1.7. Historia 15: Mis publicaciones

En el anexo IV se desglosa la decimoquinta historia, se elaboró el diseño de la interfaz de las publicaciones del usuario que permite visualizar todas las mascotas que han sido agregadas por él. La lógica de mis publicaciones se realiza por medio de la vista “MyPosts.vue”, en la función “getMascotas”, que permite obtener todas las mascotas del usuario. Tal como se observa en la figura 65 y 66.

Figura 65. Interfaz de mis publicaciones

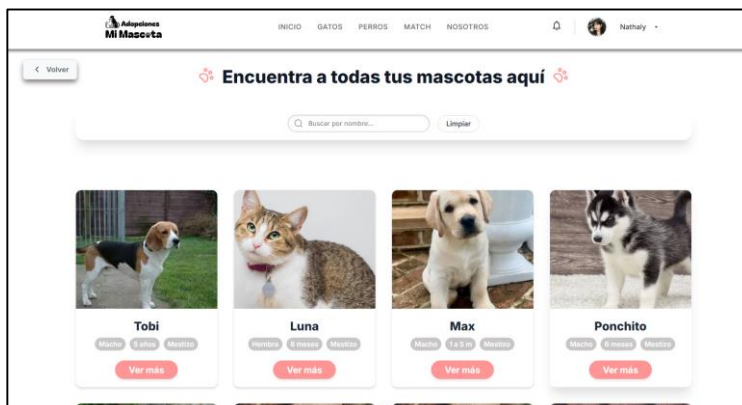


Figura 66. Código de la lógica de mis publicaciones.

```

async function getMascotas() {
  isLoading.value = true;
  const userToken = authStore.token;

  if (!userToken) {
    // Manejar el caso de que no haya token (usuario no autenticado)
    console.error("Usuario no autenticado. No se puede cargar la lista de favoritos.");
    // Opcional: Redirigir al Login
    // router.push('/login');
    isLoading.value = false;
    return;
  }

  try {
    // CORRECCIÓN: Llama a la ruta que devuelve la lista de favoritos del usuario.
    // No necesitas pasar un ID de mascota.
    const response = await fetch('http://localhost:3000/api/mascotas/MyPostUser', {
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${userToken}`,
      },
    });
  }

  if (!response.ok) {
    throw new Error('Error al cargar los favoritos: ' + response.statusText);
  }
}

```

La función encargada de procesar la solicitud es “*getMyPosts*”. Por medio de la llamada a la función “*getMyPostsByUserIdDB*” de *mascotasModel.js* se obtiene la información de todas las mascotas del usuario, de acuerdo con el Id. Como se observa en la figura 67 y 68.

Figura 67. Código de la función *getMyPosts*

```
exports.getMyPosts = async (req, res) => {
  const userId = req.user.id;
  const client = await pool.connect();
  try {
    const result = await mascotasModel.getMyPostsByUserIdDB(client, userId);
    res.status(200).json(result.rows);
  } catch (dbError) {
    console.error('Error al consultar la BD para las publicaciones del usuario:', dbError);
    res.status(500).json({ mensaje: 'Error interno del servidor al obtener las publicaciones.' });
  } finally {
    client.release();
  }
};
```

Figura 68. Consulta para obtener la información de las mascotas

```
const getMyPostsByUserIdDB = async (client, userId) => {
  const query = `
    SELECT
      m.idxxxx_mascot, m.nombre_mascot, m.sex0xx_mascot,
      m.edadme_mascot, m.pesokg_mascot, m.image1_mascot
    FROM
      mascotas m
    WHERE
      m.forane_usuari_id = $1
      AND m.eliminado_logico = FALSE
      AND m.approv_mascot = 'Aprobada'
    ORDER BY m.idxxxx_mascot DESC;
  `;
  return client.query(query, [userId]);
};
```

4.3.4.1.8. Historia 16: Adopciones realizadas

En el anexo IV se segmenta la decimosexta historia, se elaboró el diseño de la interfaz de adopciones realizadas, que permite llevar un control de las adopciones que el usuario ha realizado. La lógica se implementa por medio de la vista “*AdopcionesRealizadas.vue*”, en la función “*fetchUserSolicitudes*”, que permite obtener la información de las solicitudes, la ficha de la mascota y el formulario de adopción. Tal como se observa en la figura 69 y 70.

Figura 69. Interfaz de adopciones realizadas.



Nº	FECHA DE ENVÍO	NOMBRE DE LA MASCOTA	FICHA DE INFORMACIÓN	FORMULARIO DE ADOPCIÓN	ESTADO DE SOLICITUD
1	16/10/2025	Espanada	Ver detalles >	Ver detalles >	Adoptado
2	10/10/2025	Lucky	Ver detalles >	Ver detalles >	Adoptado
3	10/10/2025	Cochi	Ver detalles >	Ver detalles >	Adoptado

Figura 70. Código de la función fetchUserSolicitudes

```

const fetchUserSolicitudes = async () => {
  loading.value = true;
  error.value = null;
  const userToken = authStore.token;

  if (!userToken) {
    error.value = "Debes iniciar sesión para ver tus solicitudes.";
    loading.value = false;
    router.push('/login'); // Redirigir al Login si no hay token
    return;
  }

  try {
    const response = await fetch('http://localhost:3000/api/adoptions/userSuccessAdoption', {
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${userToken}`,
      }
    });

    if (response.status === 401) {
      authStore.logout();
      router.push('/login');
    }
  }
}

```

Para lógica de las adopciones se creó el controlador *“adoptionController.js”*, en donde se encuentra la función *“getUserAdoptionSucess”* encargada del manejo de las solicitudes. Posteriormente, llama a *“getUserAdoptionSucess”* en donde se obtiene la información de las solicitudes. Como se observa en la figura 71 y 72.

Figura 71. Código de la función getUserAdoptionSucess

```

getUserAdoptionSucess: async (req, res) => {
  try {
    const forms = await AdoptionModel.getUserAdoptionSucess(req.user.id);
    res.status(200).json(forms);
  } catch (error) {
    console.error('Error al obtener formularios de adopción exitosos (Controlador):', error);
    res.status(500).json({ message: 'Error interno del servidor al obtener las solicitudes.', error: error.message });
  }
},

```

Figura 72. Consultar para recopilar los datos de la solicitud

```

async function getUserAdoptionSucess(userId) {
  const query = `
    SELECT s.idxxxx_solici, s.fechax_solici, m.nombre_mascot, s.estado_solici, s.adosuc_solici,
    FROM "solicitudes_adopcion" s
    JOIN "mascotas" m ON s.forane_mascot_id = m.idxxxx_mascot
    JOIN "formularioAdopcion" d ON s.forane_forado_id = d.idxxxx_forado
    WHERE s.forane_solici_id = $1 AND s.adosuc_solici=TRUE
    ORDER BY s.fechax_solici DESC;
  `;
  const result = await pool.query(query, [userId]);
  return result.rows;
}

```

4.3.4.1.9. Historia 17: Visualización de mascotas

La decimoséptima historia se encuentra detallada en el anexo IV, se elaboró el diseño de la interfaz de las mascotas, en donde se visualiza respectivamente cada uno de los gatos y perros que han sido subidos por otros usuarios. La lógica de la visualización de mascota se realiza por medio de la vista *“GatosPage.vue”* y *“PerrosPage”*, en la función *“getMascotas”*, que permite obtener la información de las mascotas. Tal como se observa en la figura 73 y 74.

Figura 73. Interfaz de visualización de mascotas

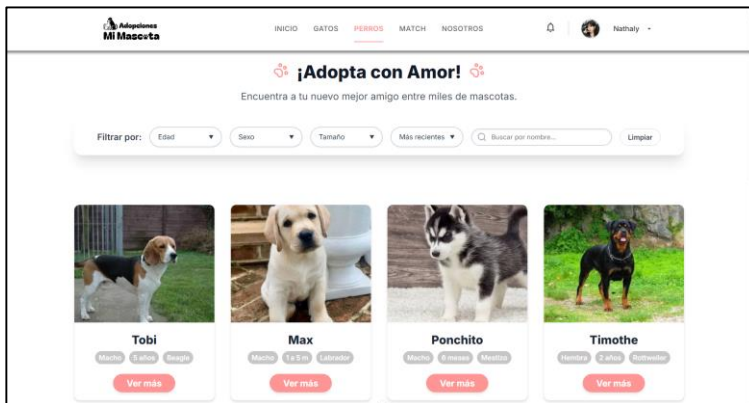


Figura 74. Código de la lógica de visualización de mascotas

```

async function getMascotas() {
  isLoading.value = true;
  try {
    // NOTA: Asegúrate de que tu backend tenga un endpoint que devuelva todas las mascotas
    const response = await fetch('http://localhost:3000/api/mascotas/perros', {
      method: 'GET',
      headers: {
        'Content-Type': 'application/json'
      },
    });
  };
  if (!response.ok) {
    throw new Error('Error al cargar las mascotas: ' + response.statusText);
  }
  const data = await response.json();
  if (data.length === 0) {
    mascotas.value = createMockMascotas(8); // Usar mocks si no hay datos
  } else {
    mascotas.value = data;
  }
}

```

Las funciones encargadas de procesar la solicitud del apartado de gatos y perros son “*getMascotaForGatosFeed*” y “*getMascotaForPerrosFeed*” respectivamente, en donde ambas contienen la función “*getMascotaForFeedDB*”, la cual obtiene la información de las mascotas seccionándola por la especie en la base de datos. Como se observa en la figura 75 y 76.

Figura 75. Función *getMascotasForGatoFeed* y *getMascotasForPerrosFeed*

```

exports.getMascotasForGatoFeed = async (req, res) => {
  const client = await pool.connect();
  try {
    const result = await mascotaModel.getMascotasForFeedDB(client, 'Gato');
    res.status(200).json(result.rows);
  } catch (dbError) {
    console.error('Error al obtener la lista de gatos para el feed:', dbError);
    res.status(500).json({ mensaje: 'Error interno del servidor al obtener gatos para el feed.', error: dbError });
  } finally {
    client.release();
  }
};

// GET /api/mascotas/perros
exports.getMascotasForPerroFeed = async (req, res) => {
  const client = await pool.connect();
  try {
    const result = await mascotaModel.getMascotasForFeedDB(client, 'Perro');
    res.status(200).json(result.rows);
  } catch (dbError) {
    console.error('Error al obtener la lista de perros para el feed:', dbError);
    res.status(500).json({ mensaje: 'Error interno del servidor al obtener perros para el feed.', error: dbError });
  } finally {
    client.release();
  }
};

```

Figura 76. Consulta para recopilar los datos de los perros o gatos

```

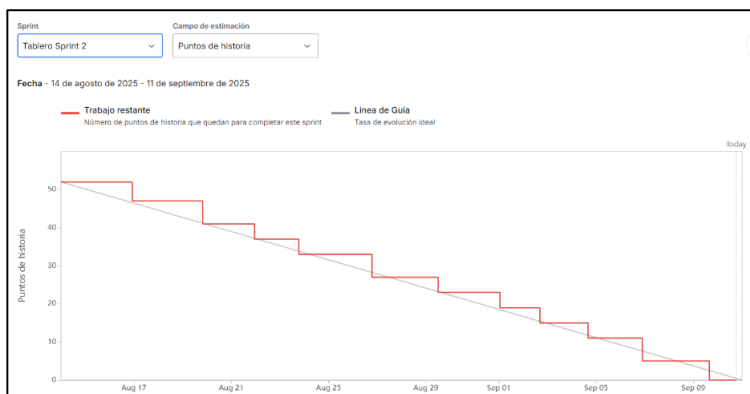
const getMascotasForFeedDB = async (client, especieFilter) => {
  const query = `
    SELECT
      m.idxxxx_mascot, m.nombre_mascot, m.especi_mascot, m.sexxxx_mascot,
      m.edadme_mascot, m.razaxx_mascot, m.tamano_mascot, m.image1_mascot
    FROM
      mascotas m
    WHERE
      m.eliminado_logico = FALSE
    AND
      m.especi_mascot = $1 AND m.approv_mascot = 'Aprobada'
    ORDER BY
      m.idxxxx_mascot DESC;
  `;
  return client.query(query, [especieFilter]);
};

```

4.3.4.2. Sprint 2 – Gráfico de trabajo pendiente

Por medio de *Jira Software* se pudo llevar un control sobre las tareas del segundo grupo de historias. A través del gráfico de trabajo restante se visualizó el progreso de actividades por parte del equipo de trabajo, tomando en cuenta los puntos de historia que fueron completados en determinados días. En la figura 77, se visualiza el gráfico del segundo *sprint*.

Figura 77. Gráfica de trabajo restante del sprint 2 en Jira.



4.3.4.3. Sprint 2 - Revisión (Sprint Review)

Al finalizar el *sprint* número dos, el equipo realizó la reunión de revisión en presencia del gerente del centro veterinario, con el mismo propósito de la revisión del primer *sprint*, esta etapa se centró en la implementación completa del módulo de gestión de mascotas y la consolidación de las opciones del perfil de usuario.

El seguimiento de las tareas se gestionó nuevamente con *Jira*, y el progreso se vigiló a través del gráfico de trabajo restante del *Sprint 2*. Se confirmó que todas las funcionalidades presentadas cumplían con los criterios de aceptación.

4.3.4.4. Sprint 2 - Retrospectiva (Sprint Retrospective)

Para la retrospectiva del segundo sprint se consideraron tres preguntas, por medio de las cuales se tendrá una visión clara sobre cómo se llevó el *sprint*. Como se visualiza en la tabla 20.

Tabla 20. Sprint 2 retrospectiva

Aspectos exitosos del sprint	Mejoras para aplicar	Errores en el sprint
Las tareas se asignaron a los miembros del equipo, logrando completar exitosamente el <i>sprint</i> 2.	Diseño de interfaces y mejoras en la lógica del código.	Problemas en la organización de la información en la base de datos

4.3.5. Sprint 3

4.3.5.1. Sprint Backlog 3 – Planificación (Sprint Planning)

De acuerdo a la tabla 21 y 22, se detallan la historia técnica e historias de usuario que han sido tomadas en cuenta para el desarrollo del tercer *sprint backlog*, de las cuales cuentan con una probabilidad alta de desarrollo. Además, se describieron las tareas para cada una de las historias de usuario que forman parte del *sprint*.

Tabla 21. Sprint 3 retrospectiva

<i>Sprint Backlog</i>						
<i>Historia</i>	<i>Est</i>	<i>Categoría</i>	<i>Tarea</i>	<i>Responsable</i>	<i>Est</i>	<i>Estado</i>
HU18-Ficha de información de mascotas	5	Diseño	Elaboración de diseño de la ficha de mascotas (perros y gatos)	Jaime Ortiz	2	Completado
		Desarrollo	Implementación de página para mostrar la ficha de las mascotas	Jaime Ortiz	1	Completado
		Desarrollo	Recolección de la información de las mascotas	Jaime Ortiz	1	Completado
		Pruebas	Pruebas: verificar que el componente recupere el ID de la mascota desde la ruta y <i>renderice</i> toda la información técnica asociada.	Jaime Ortiz	1	Completado
HU19-Favoritos	5	Diseño	Elaboración de diseño de interfaz de favoritos	Nathaly Zamora	2	Completado

		Desarrollo	Implementación de página para mostrar la lista de favoritos	Jaime Ortiz	1	Completado
		Desarrollo	Recolección de la lista de favoritos	Jaime Ortiz	1	Completado
		Pruebas	Pruebas: verificar los cambios del store al agregar/quitar mascotas de la lista de favoritos en tiempo real	Nathaly Zamora	1	Completado
		Diseño	Elaboración de diseño de formulario para solicitud	Jaime Ortiz	2	Completado
		Desarrollo	Validación de datos	Jaime Ortiz	1	Completado
		Desarrollo	Guardado de información	Jaime Ortiz	1	Completado
		Pruebas	Pruebas: validar que el objeto de solicitud se envíe con la estructura requerida por el <i>backend</i> y manejar errores de validación	Jaime Ortiz	1	Completado
		Diseño	Elaboración de diseño de solicitudes enviadas	Nathaly Zamora	2	Completado
		Desarrollo	Recolección de la información de solicitudes enviadas	Nathaly Zamora	2	Completado
		Pruebas	Pruebas: verificar el <i>renderizado</i> condicional del estado de la solicitud	Nathaly Zamora	1	Completado
		Diseño	Elaboración de diseño de solicitudes recibidas	Jaime Ortiz	2	Completado
		Desarrollo	Recolección de la información de solicitudes recibidas	Jaime Ortiz	2	Completado
		Pruebas	Pruebas: comprobar la emisión de eventos al aprobar/rechazar una solicitud y la actualización inmediata del estado	Nathaly Zamora	1	Completado
HU20- Solicitud de adopción	5					
HU21- Visualización de solicitudes enviadas	5					
HU22- Visualización de solicitudes recibidas	5					

HU23-Gestión de solicitud de adopción	5	Diseño	Elaboración de diseño de gestión de solicitudes	Nathaly Zamora	2	Completado
		Desarrollo	Integrar opciones de gestión	Jaime Ortiz	1	Completado
		Desarrollo	Actualización de estado	Jaime Ortiz	1	Completado
		Pruebas	Pruebas: validar unitariamente los botones, comprobando que emitan la actualización de estado correcta hacia el <i>backend</i> .	Nathaly Zamora	1	Completado
HU24-Notificación de solicitud	3	Diseño	Elaboración de diseño de notificaciones	Nathaly Zamora	1	Completado
		Desarrollo	Recolección de la información	Jaime Ortiz	1	Completado
		Pruebas	Prueba: verificar que el componente o medalla (<i>badge</i>) de notificaciones incremente su contador de forma reactiva al simular la recepción de un dato nuevo.	Nathaly Zamora	1	Completado
HU25-Recomendación de Mascotas Personalizada	8	Diseño	Diseñar interfaz de recomendaciones	Nathaly Zamora	2	Completado
		Desarrollo	Recopilar/limpiar datos, entrenar y desplegar modelo <i>ML</i>	Nathaly Zamora	3	Completado
		Desarrollo	Integrar modelo con <i>backend</i> y <i>frontend</i> , configurar monitoreo	Nathaly Zamora	2	Completado
		Pruebas	Pruebas: comprobar que el componente <i>frontend</i> capture las preferencias del usuario y envíe correctamente el <i>payload</i> al <i>endpoint</i> del modelo de <i>Machine Learning</i>	Jaime Ortiz	1	Completado
HU26-Enviar Valoración	3	Diseño	Elaboración de diseño de envío de valoración	Nathaly Zamora	1	Completado
		Desarrollo	Guardado de información	Jaime Ortiz	1	Completado

		Pruebas	Pruebas: validar la lógica del componente de estrellas (<i>rating</i>), asegurando que el valor numérico seleccionado (1 al 5) se guarde en la variable reactiva y se envíe correctamente.	Nathaly Zamora	1	Completado
HU27-Gestión de Valoraciones	3	Diseño	Elaboración de diseño de valoraciones	Nathaly Zamora	1	Completado
		Desarrollo	Recolección de la información de valoraciones	Jaime Ortiz	1	Completado
		Pruebas	Pruebas: verificar la iteración correcta de la lista de comentarios.	Nathaly Zamora	1	Completado

Tabla 22. Sprint Backlog Técnico

<i>Sprint Backlog Técnico</i>						
<i>Historia</i>	<i>Est</i>	<i>Categoría</i>	<i>Tarea</i>	<i>Responsable</i>	<i>Est</i>	<i>Estado</i>
HT1- Seguridad y privacidad de la aplicación	8	Desarrollo	Autenticación con <i>JWT (JSON Web Tokens)</i>	Jaime Ortiz	1	Completado
		Desarrollo	Encriptación de Contraseñas	Jaime Ortiz	1	Completado
		Desarrollo	Middleware de Autorización por Roles y Permisos	Jaime Ortiz	2	Completado
		Desarrollo	Protección de Rutas por medio de <i>Vue-Router</i>	Jaime Ortiz	2	Completado
		Desarrollo	Conexión Segura a Base de Datos (<i>SSL</i>)	Jaime Ortiz	1	Completado
		Desarrollo	Validación de Dominios de Correo	Jaime Ortiz	1	Completado

4.3.5.1.1. Historia 18: Ficha de información de mascotas

En el anexo IV se segmenta la decimooctava historia, se elaboró el diseño de la ficha de las mascotas que contiene cada una de las características de las mascotas, su nombre, edad, raza, peso, imágenes, etc. La lógica de las fichas de las mascotas se encuentra en la vista “*TarjetaMascota.vue*”, en la función “*fetchPetData*”, que permite obtener los datos de las mascotas. Tal como se observa en la figura 78 y 79.

Figura 78. Ficha de las mascotas

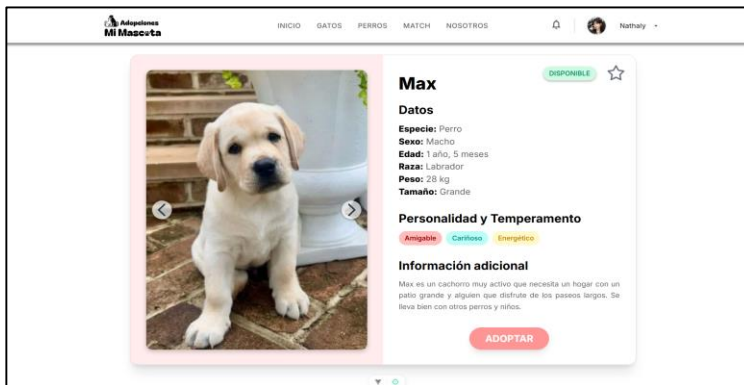


Figura 79. Código de la lógica de la ficha de las mascotas

```
const fetchPetData = async () => {
  const petId = route.params.id;
  if (!petId) {
    error.value = 'No se proporcionó un ID de mascota.';
    loading.value = false;
    return;
  }

  try {
    const response = await fetch(`http://localhost:3000/api/mascotas/card/${petId}`);

    if (!response.ok) {
      const errData = await response.json();
      throw new Error(errData.mensaje || `Error HTTP: ${response.status}`);
    }

    const data = await response.json();

    pet.value = {
      id: data.idxxxx_mascot,
      nombre: data.nombre_mascot,
    };
  }
}
```

En la función `getMascotasCardDetails` del controlador `mascotaController.js` se gestiona la lógica de control, en donde se llama a `getMascotaDetailsByIdDB` que por medio de una consulta obtiene las características de las mascotas de la base de datos. Como se observa en la figura 80 y 81.

Figura 80 Código de la función `getMascotaCardDetails`

```
exports.getMascotaCardDetails = async (req, res) => {
  const petId = req.params.id;
  if (isNaN(petId)) {
    return res.status(400).json({ mensaje: 'ID de mascota inválido.' });
  }

  const client = await pool.connect();
  try {
    const result = await mascotaModel.getMascotaDetailsByIdDB(client, petId);

    if (result.rows.length === 0) {
      return res.status(404).json({ mensaje: 'Mascota no encontrada.' });
    }
    res.json(result.rows[0]);
  } catch (dbError) {
    console.error('Error al consultar la BD para la mascota:', dbError);
    res.status(500).json({ mensaje: 'Error interno del servidor al obtener la mascota.' });
  } finally {
    client.release();
  }
};
```

Figura 81. Consulta para recopilar las características de la mascota.

```

const getMascotaDetailsByIdDB = async (client, petId) => {
  // Primero obtener Los datos de La mascota
  const mascotaQuery = `
  SELECT
    m.idxxxx_mascot,
    m.nombre_mascot,
    m.especi_mascot,
    m.sexoxx_mascot,
    m.edadme_mascot,
    m.razaxx_mascot,
    m.pesokg_mascot,
    m.tamano_mascot,
    m.infoad_mascot,
    m.image1_mascot,
    m.image2_mascot,
    m.image3_mascot,
    m.forane_usuari_id,
    m.status_mascot
  FROM mascotas m
  WHERE m.idxxxx_mascot = $1
  AND m.eliminado_logico = FALSE
  `;
  const mascotaResult = await client.query(mascotaQuery, [petId]);
}

```

4.3.5.1.2. Historia 19: Favoritos

En el anexo IV se detalla la decimonovena historia por segmentos, se elaboró el diseño de la interfaz de favoritos que permite agregar las mascotas al apartado de favoritos. La lógica favoritos se realiza por medio de la vista “*favoritesPage.vue*”, en la función “*getMascotas*”, que permite obtener las mascotas favoritas. Tal como se observa en la figura 82 y 83.

Figura 82. Interfaz de favoritos

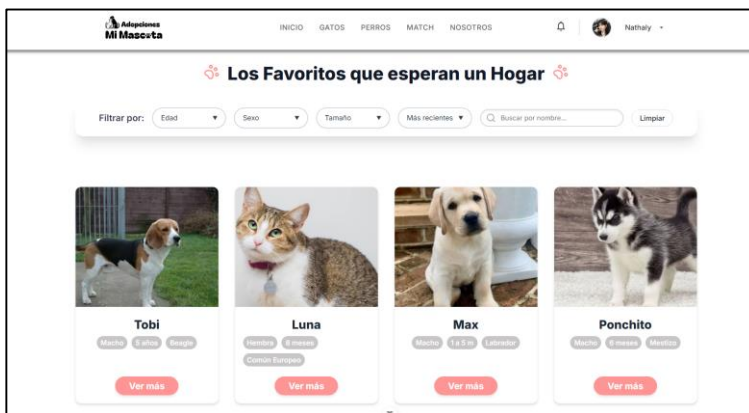


Figura 83. Código de la lógica de favoritos

```

async function getMascotas() {
  isLoading.value = true;
  const userToken = authStore.token;

  if (!userToken) {
    // Manejar el caso de que no haya token (usuario no autenticado)
    console.error("Usuario no autenticado. No se puede cargar la lista de favoritos.");
    // Optional: Redirigir al Login
    // router.push("/Login");
    isLoading.value = false;
    return;
  }

  try {
    // CORRECCIÓN: Llama a la ruta que devuelve La Lista de favoritos del usuario.
    // No necesitas pasar un ID de mascota.
    const response = await fetch('http://localhost:3000/api/favorites/favorites', {
      headers: {
        'Content-Type': 'application/json',
        'Authorization': 'Bearer ${userToken}',
      }
    });

    if (!response.ok) {
      throw new Error('Error al cargar los favoritos: ' + response.statusText);
    }

    const data = await response.json();
    // Asignar Los datos recibidos a 'mascotas.unive'
  }
}

```

La lógica de control de favoritos se realizó por medio de *mascotaController.js*, a través de la función “*checkFavorites*”. Posteriormente, a través de “*getFavoritesByIdDB*” se obtiene las mascotas marcadas como favoritas por el usuario en la base de datos. Como se observa en la figura 84 y 85.

Figura 84. Código de la función *checkFavorites*

```
exports.checkFavorites = async (req, res) => {
  const userId = req.user.id;

  // 1. Validación
  if (!userId) {
    return res.status(400).json({ message: 'ID de usuario faltante.' });
  }

  try {
    // 2. Llamada al Modelo
    const result = await favoritesModel.getFavoritesByIdDB(userId);

    // 3. Respuesta
    res.status(200).json(result.rows);
  } catch (dbError) {
    console.error('Error al obtener la lista de mascotas favoritas:', dbError);
    res.status(500).json({ message: 'Error interno del servidor al obtener mascotas favo
```

Figura 85. Consulta para recopilar las mascotas favoritas

```
const getFavoritesByIdDB = async (userId) => {
  const query = `
  SELECT
    m.*
  FROM
    favoritos AS f
  JOIN
    mascotas AS m ON f.forane_mascot_id = m.idxxxx_mascot
  WHERE
    f.forane_usuari_id = $1
    AND f.status_favori = TRUE
    AND m.eliminado_logico = FALSE
  ORDER BY
    m.nombre_mascot;
  `;
};
```

4.3.5.1.3. Historia 20: Solicitud de adopción

La vigésima historia se encuentra desglosada en el anexo IV, se elaboró el diseño del formulario de adopción, permite ingresar la información personal de usuario, además de preguntas adicionales que determinan si cuentan con aspectos necesarios para adoptar una mascota. La lógica se implementa por medio de la vista “*adoptForm.vue*”, en la función “*submitForm*”, que permite recopilar los datos del formulario enviado. Tal como se observa en la figura 86 y 87.

Figura 86. Interfaz del formulario de adopción

The screenshot shows a web application interface for an adoption form. At the top, there is a navigation bar with links for 'INICIO', 'GATOS', 'PERROS', 'MATCH', and 'NOSOTROS'. The user's name 'Nathaly' is visible in the top right corner. The main content area is titled 'Formulario de Adopción' and features a card for a cat named 'Luna'. The card includes a photo of the cat and text indicating it is a female ('Hembra'), 8 months old, and weighs 3.5kg. Below the card, there is a section for 'Información Personal y de Contacto' with three input fields: 'Cédula', 'Nombre', and 'Fecha de nacimiento'. A 'Volver' button is located in the top left corner of the form area.

Figura 87. Código de la lógica del formulario de adopción

```

async function submitForm() {
  isSubmitting.value = true;
  const userToken = authStore.token;

  const userObject = authStore.user;
  const userId = userObject?.id;

  console.log("DIAGNÓSTICO AUTH STORE (Inicio de Función:");
  console.log("Token (Existe?):", !!userToken);
  console.log("User ID (Valor):", userId);
  console.log("User Object (Estructura):", userObject);

  if (!userToken) {
    console.error('DIAGNÓSTICO FALLA: userToken o userId es nulo/falso.');
```

La lógica de solicitud de adopción se realizó por medio del controlador *adoptionController.js*, la función encargada de procesar la respuesta es “*submitAdoptionForm*” que llama a “*createAdoptionSolicitud*”, en donde se guarda toda la información recopilada del formulario y luego se crea la solicitud de adopción correspondiente. Como se observa en la figura 88 y 89.

Figura 88. Código de la función submitAdoptionForm

```

submitAdoptionForm: async (req, res) => {
  const userId = req.userId;
  const formDetails = req.body;

  if (!userId || !formDetails.forane_mascot_id || !formDetails.nombre_forado || !formDetails.correo_forado) {
    return res.status(400).json({ message: 'Datos incompletos para el formulario de adopción.' });
  }

  try {
    // LLAMADA AL MODELO: Ejecuta La transacción DB y obtiene datos para La notificación.
    const {
      solicitud, nombre_mascot, image1_mascot, forane_public_id, nombre_forado, idxxxx_solici
    } = await AdoptionModel.createAdoptionSolicitud(userId, formDetails);

    // LÓGICA DE COMUNICACIÓN: Notificar al DUEÑO de la mascota
    await NotificationModel.createAndEmitNotification(id, {
      forane_usuario_id: forane_public_id,
      type: 'nueva_solicitud',
      message: `Tienes una nueva solicitud de adopción para <b style="font-weight: 800;">${nombre_mascot}</b>`,
      forane_solicitud_id: idxxxx_solici,
      related_user_id: userId,
      image_url: image1_mascot
    });

    res.status(201).json({
      message: 'Formulario de adopción enviado con éxito. Solicitud creada.',
      solicitud: solicitud
    });
  }
}

```

Figura 89. Consulta para insertar la información del formulario y crear la solicitud

```

async function createAdoptionSolicitud(userId, formDetails) {
  // Desestructuración de Los campos necesarios
  const {
    cedula_forado, nombre_forado, fnacim_forado, correo_forado, telefo_forado,
    tvivie_forado, propie_forado, patjar_forado, tampat_forado, nperca_forado,
    masant_forado, otras_forado, nmasco_forado, motivo_forado, ubimas_forado,
    horas1_forado, encarg_forado, veteri_forado, gastos_forado,
    forane_mascot_id
  } = formDetails;

  // Se mantiene La Lógica transaccional aquí (Modelo/Lógica de Negocio)
  const client = await pool.connect();

  try {
    await client.query('BEGIN');


    // 1. OBTENER ID DEL PUBLICADOR y datos de La mascota
    const publicadorQuery = `
      SELECT forane_usuario_id, nombre_mascot, image1_mascot
      FROM mascotas
      WHERE idxxxx_mascot = $1;
    `;
    const publicadorResult = await client.query(publicadorQuery, [forane_mascot_id]);

    if (publicadorResult.rows.length === 0) {
      throw new Error('Mascota no encontrada o ID de mascota inválido.');
```

4.3.5.1.4. Historia 21: Visualización de solicitudes enviadas

La división de la vigésima primera historia se encuentra en el anexo IV, se elaboró el diseño de la interfaz de solicitudes enviadas que permite visualizar todas las solicitudes que el usuario ha enviado a determinadas mascotas. La lógica se implementa por medio de la vista “*SolicitudesEnviadas.vue*”, dentro de la función “*fetchUserSolicitudes*”, que recopila las solicitudes del usuario. Tal como se observa en la figura 90 y 91.

Figura 90. Interfaz de solicitudes enviadas



N°	FECHA DE ENVÍO	NOMBRE DE LA MASCOTA	FICHA DE INFORMACIÓN	FORMULARIO DE ADOCIÓN	ESTADO DE SOLICITUD
1	10/10/2025	Empanada	Ver Detalles >	Ver Detalles >	Aceptada
2	10/10/2025	Lucky	Ver Detalles >	Ver Detalles >	Rechazada
3	10/10/2025	Lucky	Ver Detalles >	Ver Detalles >	Rechazada
4	10/10/2025	Cochi	Ver Detalles >	Ver Detalles >	Aceptada

Figura 91. Código de la lógica de solicitudes enviadas

```
const fetchUserSolicitudes = async () => {
  loading.value = true;
  error.value = null;
  let userToken = authStore.token;

  if (!userToken) {
    const storedToken = localStorage.getItem('authToken');

    if (storedToken) {
      userToken = storedToken;
    } else {
      error.value = "Debes iniciar sesión para ver tus solicitudes.";
      loading.value = false;
      router.push({ name: 'auth' });
      return;
    }
  }

  try {
    // Importante: Este endpoint debe ser implementado en Express.
    // La ruta asume que el backend usa el token para obtener las solicitudes del usuario.
    const response = await fetch('http://localhost:3000/api/adoptions/user', {
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${userToken}`,
      },
    });
  }
}
```

Las solicitudes enviadas son gestionadas por medio de la función “*getUserAdoptionForms*” que llama a “*getUserAdoptionForms*” y aplica una consulta para recopilar los datos de las solicitudes. Como se observa en la figura 92 y 93.

Figura 92. Código de la función *getUserAdoptionForms*

```
getUserAdoptionForms: async (req, res) => {
  try {
    const forms = await AdoptionModel.getUserAdoptionForms(req.user.id);
    res.status(200).json(forms);
  } catch (error) {
    console.error('Error al obtener formularios de adopción del usuario (Controlador):', error);
    res.status(500).json({ message: 'Error interno del servidor al obtener las solicitudes' });
  }
},
```

Figura 93. Consulta para recopilar las solicitudes enviadas.

```

async function getUserAdoptionForms(userId) {
  const query = `
    SELECT
      s.idxxxx_solici, s.fechax_solici, m.nombre_mascot, s.estado_solici,
      s.forane_mascot_id, d.*
    FROM
      "solicitudes_adopcion" s
    JOIN
      "mascotas" m ON s.forane_mascot_id = m.idxxxx_mascot
    JOIN
      "formularioAdopcion" d ON s.forane_forado_id = d.idxxxx_forado
    WHERE
      s.forane_solici_id = $1
    ORDER BY
      s.fechax_solici DESC;
  `;
  const result = await pool.query(query, [userId]);
  return result.rows;
}

```

4.3.5.1.5. Historia 22: Visualización de solicitudes recibidas

En el anexo IV se segmenta la vigésima segunda historia, se elaboró el diseño de la interfaz de solicitudes recibidas similar a los anteriores. Sin embargo, se añadieron apartados para el cambio de estados, permitiendo visualizar todas las solicitudes de adopción enviadas a las mascotas del usuario. La lógica se implementa por medio de la vista “*SolicitudesRecibidas.vue*”, dentro de la función “*fetchUserSolicitudes*”, que recopila las solicitudes del usuario. Tal como se observa en la figura 94.

Figura 94. Interfaz de solicitudes recibidas

N°	FECHA DE ENVÍO	NOMBRE DE LA MASCOTA	FICHA DE INFORMACIÓN	FORMULARIO DE ADOCIÓN	ESTADO DE SOLICITUD	ADOCIÓN
1	16/10/2025	Empanada	Ver detalles >	Ver detalles >	Aceptar ▾	Adoptado
2	10/10/2025	Lucky	Ver detalles >	Ver detalles >	Rechazar ▾	Adoptado
3	10/10/2025	Lucky	Ver detalles >	Ver detalles >	Rechazar ▾	SI No
4	10/10/2025	Cochi	Ver detalles >	Ver detalles >	Aceptar ▾	Adoptado

La lógica de las adopciones recibidas se centralizó en el controlador *adoptionController.js*, la función encargada de procesar esta solicitud es “*getReceivedAdoptionForms*”. Posteriormente, se llama a “*getReceivedAdoptionForms*” que recopila los datos de las solicitudes, como se observa en la figura 95 y 96.

Figura 95. Código de la función getReceivedAdoptionForms

```

getReceivedAdoptionForms: async (req, res) => {
  try {
    const forms = await AdoptionModel.getReceivedAdoptionForms(req.user.id);
    res.status(200).json(forms);
  } catch (error) {
    console.error('Error al obtener formularios de adopción recibidos (Controlador):', error);
    res.status(500).json({ message: 'Error interno del servidor al obtener las solicitudes' });
  }
}

```

Figura 96. Consulta para recopilar las solicitudes recibidas

```

async function getReceivedAdoptionForms(userId) {
  const query = `
    SELECT s.idxxxx_solici, s.fecha_x_solici, m.nombre_mascot, m.status_mascot, s.estado_solici
    JOIN "mascotas" m ON s.forane_mascot_id = m.idxxxx_mascot
    JOIN "FormularioAdopcion" d ON s.forane_forado_id = d.idxxxx_forado
    WHERE m.forane_usuario_id = $1
    ORDER BY s.fecha_x_solici DESC;
  `;
  const result = await pool.query(query, [userId]);
  return result.rows;
}

```

4.3.5.1.6. Historia 23: Gestión de solicitud de adopción

La vigésima tercera historia se segmenta en el anexo IV, en la interfaz de solicitudes recibidas se gestionan las solicitudes, permitiendo aceptar o rechazar. Además, permite actualizar el estado de la mascota si la adopción fue un éxito. La lógica se implementa por medio de la vista "SolicitudesRecibidas.vue", específicamente, en la función "updateMascotStatus" y "confirmAdoption", que actualiza el estado de la solicitud y adopción. Tal como se observa en la figura 97 y 98.

Figura 97. Diseño de la gestión de solicitudes

N°	FECHA DE ENVÍO	NOMBRE DE LA MASCOTA	FECHA DE INFORMACIÓN	FORMULARIO DE ADOCIÓN	ESTADO DE SOLICITUD	ADOCIÓN
1	16/10/2025	Empanada	Ver detalles >	Ver detalles >	Aceptar ▼	Adoptado
2	10/10/2025	Lucky	Ver detalles >	Ver detalles >	Aceptar Rechazar	Adoptado
3	10/10/2025	Lucky	Ver detalles >	Ver detalles >	Rechazar ▼	No
4	10/10/2025	Cochi	Ver detalles >	Ver detalles >	Aceptar ▼	Adoptado

Figura 98. Código de la lógica de gestión de solicitudes

```

const updateMascotStatus = async (mascotId, newStatus) => {
  const userToken = authStore.token || localStorage.getItem('authToken');

  if (!userToken) {
    console.error("No hay token para actualizar estado de mascota.");
    return;
  }

  // Buscamos todas las solicitudes que usan esta mascota para actualizar todas las filas en la vista
  const solicitudesToUpdate = solicitudes.value.filter(s => s.forane_mascot_id === mascotId);
  if (solicitudesToUpdate.length === 0) return;

  const oldStatus = solicitudesToUpdate[0].status_mascot;

  // OPTIMISTA: Actualiza la interfaz para todas las filas inmediatamente
  solicitudesToUpdate.forEach(s => {
    s.status_mascot = newStatus;
  });

  try {
    // Debes implementar esta nueva ruta en tu backend de Express/Node.js
    const response = await fetch('http://localhost:3000/api/adoptions/${mascotId}/statusM', {
      method: 'PATCH',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${userToken}`,
      },
      body: JSON.stringify({ status: newStatus }),
    });
  }
};

```

La función encargada de manejar las peticiones es “*updateAdoptionStatus*” y “*updateMascotStatus*”, que por medio de “*updateSolicitudStatus*” y “*updateMascotStatus*” actualizan el estado de la solicitud y mascota en la base de datos. Como se observa en la figura 99 y 100.

Figura 99. Código de la función *updateAdoptionStatus*

```
updateAdoptionStatus: async (req, res) => {
  const userId = req.user.id;
  const { formId } = req.params;
  const { status } = req.body;

  if (!status || ![ 'Aceptada', 'Rechazada', 'Pendiente' ].includes(status)) {
    return res.status(400).json({ message: 'Estado inválido proporcionado.' });
  }

  try {
    // LLAMADA AL MODELO: Ejecuta La verificación y La actualización de DB.
    const result = await AdoptionModel.updateSolicitudStatus(formId, userId, status);

    const { updatedForm, solicitanteId, details } = result;

    // LÓGICA DE COMUNICACIÓN: Notificación al SOLICITANTE
    if (updatedForm && (status === 'Aceptada' || status === 'Rechazada')) {
      const notificationType = status === 'Aceptada' ? 'solicitud_aceptada' : 'solicitud_rechazada';
      const notificationMessage = status === 'Aceptada'

```

Figura 100. Consulta para actualizar el estado de la solicitud

```
async function updateSolicitudStatus(formId, userId, status) {
  // 1. Verificar dueño y obtener datos clave
  const verificationQuery = `
    SELECT m.forane_usuari_id, s.forane_solici_id, s.forane_mascot_id
    FROM "solicitudes_adopcion" s
    JOIN "mascotas" m ON s.forane_mascot_id = m.idxxxx_mascot
    WHERE s.idxxxx_solici = $1;
  `;
  const verificationResult = await pool.query(verificationQuery, [formId]);

  if (verificationResult.rows.length === 0) {
    throw new Error('Solicitud no encontrada.');
```

4.3.5.1.7. Historia 24: Notificación de solicitud

En el anexo IV se segmenta la vigésima cuarta historia, se elaboró el diseño del apartado de notificaciones. Permite visualizar las notificaciones enviadas al usuario de acuerdo con ciertas acciones, como solicitud aceptada, rechazada, si la mascota ya fue adoptada por otro usuario, etc. La lógica de notificaciones se realiza por medio del componente “*Navbar.vue*”, en la función “*fectchNotifications*”, que permite obtener las notificaciones recibidas. Tal como se observa en la figura 101 y 102.

Figura 101. Apartado de notificaciones

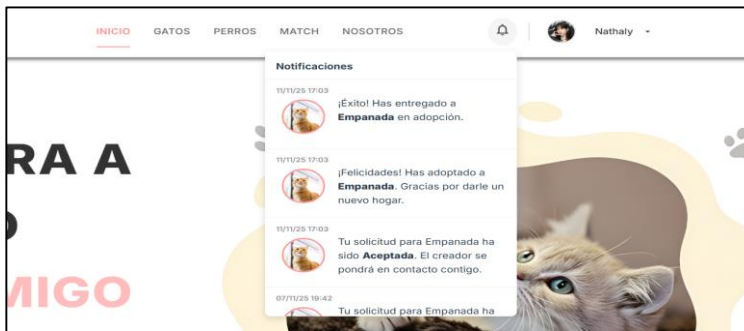


Figura 102. Código de la lógica de notificaciones

```
const fetchNotifications = async () => {
  const token = localStorage.getItem('authToken');
  if (!token) return;

  try {
    // AJUSTA ESTA RUTA a la de tu endpoint de notificaciones
    const response = await fetch('http://localhost:3000/api/adoptions/notificaciones', {
      headers: {
        'Authorization': `Bearer ${token}`
      }
    });

    if (response.ok) {
      const data = await response.json();
      // Asignar Los datos al ref de notificaciones
      notifications.value = data;
    } else {
      console.error('Error al cargar notificaciones:', response.statusText);
    }
  } catch (error) {
    console.error('Error de red al cargar notificaciones:', error);
  }
};
```

La función encargada de procesar la solicitud en la lógica de notificaciones es “*getNotifications*”, que realiza un llamado a “*getNotificationsByUserId*”, la cual obtiene todas las notificaciones del usuario. Como se observa en la figura 103 y 104.

Figura 103. Código de la función getNotifications

```
getNotifications: async (req, res) => {
  const userId = req.user.id;
  if (!userId) {
    return res.status(401).json({ message: 'No autenticado.' });
  }
  try {
    const notifications = await NotificationModel.getNotificationsByUserId(userId);
    res.status(200).json(notifications);
  } catch (error) {
    console.error('Error al obtener notificaciones (Controlador):', error);
    res.status(500).json({ message: 'Error interno del servidor al obtener las notificaciones.' });
  }
};
```

Figura 104. Consulta para recopilar las notificaciones

```
async function getNotificationsByUserId(userId) {
  const query = `
    SELECT
      idxxxx_notifi AS id, type, message, forane_solici_id, related_user_id,
      image_url, is_read, created_at
    FROM notifications
    WHERE forane_usuario_id = $1
    ORDER BY created_at DESC
  `;
  const result = await pool.query(query, [userId]);
  return result.rows;
}
```

4.3.5.1.8. Historia 25: Recomendación de mascotas personalizada

La vigésima quinta historia se encuentra detallada en el anexo IV, en donde se visualizan las mascotas recomendadas para cada usuario de acuerdo a los intereses previos seleccionados en el cuestionario. Tal como se observa en la figura 105 y 106.

Figura 105. Interfaz de match de mascotas

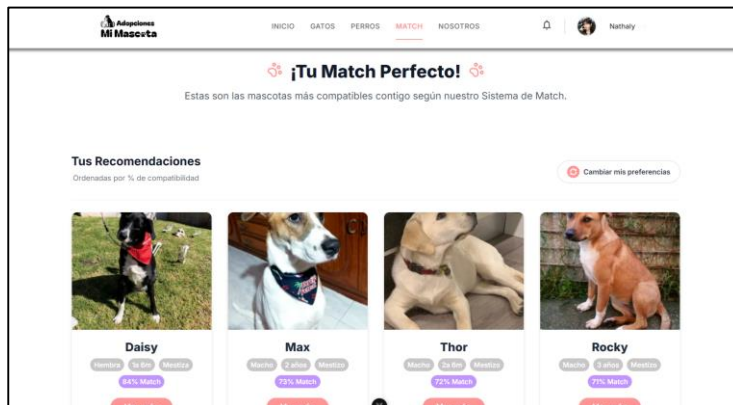
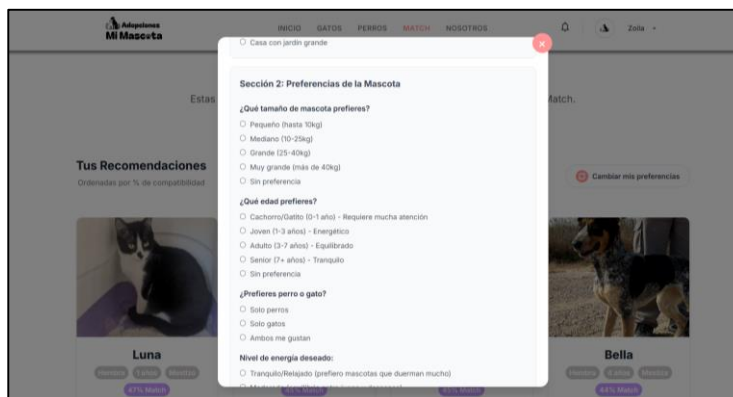


Figura 106. Cuestionario de Match



Para la creación del sistema de recomendación, se seleccionó un enfoque híbrido, en donde se empleó la técnica de aprendizaje no supervisado que utiliza el método de extracción de características *Term Frequency-Inverse Document Frequency (TF-IDF)* con *similitud de coseno* para el Filtrado Basado en Contenido, y la técnica de aprendizaje supervisado, que utiliza *SVD (Singular Value Decomposition)* y factorización de matrices, para el Filtrado Colaborativo.

Los datos sintéticos del proyecto se generaron utilizando *Python* con las librerías *NumPy* y *Pandas*, mediante tres scripts ubicados en *mimascota_ml/src/data/*. El script *generate_synthetic_pets.py* crea 200 mascotas (100 perros y 100 gatos) usando

Finalmente, *generate_synthetic_data.py* realiza todo el proceso y genera 2,700 interacciones entre usuarios y mascotas. Para cada par usuario-mascota se calcula un score de compatibilidad (0 a 1) basado en cuatro factores: coincidencia de especie preferida, adecuación del tamaño al tipo de vivienda, alineación del nivel de energía y preferencia de edad. Según este score, se generan interacciones simuladas: compatibilidad alta (mayor a 0.7) produce varios *clícs*, alta probabilidad de favorito y posibilidad de adopción; compatibilidad media (0.4 a 0.7) genera pocos clics; y compatibilidad baja no genera interacción por lo que las interacciones totales quedan en 2,353. Los datos resultantes se exportan como tres archivos CSV (*synthetic_pets.csv*, *synthetic_users.csv*, *synthetic_interactions.csv*) en la carpeta *data/raw/*, como se observa en la figura 109.

Figura 109. Script de interacciones usuario-mascota

```
ta_ml > src > data > generate_synthetic_data.py > ...
def calculate_compatibility(user, pet):
    score = 0.0

    # Factor 1: Especie (0-0.3)
    user_species_pref = user['perroGato']
    pet_species = pet['especie']

    if user_species_pref == 'ambos':
        score += 0.2
    elif (user_species_pref == 'solo_perros' and pet_species == 'Perro') or \
          (user_species_pref == 'solo_gatos' and pet_species == 'Gato'):
        score += 0.3
    elif (user_species_pref == 'solo_perros' and pet_species == 'Gato') or \
          (user_species_pref == 'solo_gatos' and pet_species == 'Perro'):
        score += 0.0 # Incompatible

    # Factor 2: Tamaño (0-0.25)
    user_size_pref = user['tamañoMascota']
    pet_size = pet['tamaño']
    user_space = user['tipoVivienda']

    # Compatibilidad de tamaño con espacio
    if user_space in ['apto_pequeno'] and pet_size in ['pequeno']:
        score += 0.25
    elif user_space in ['apto_pequeno'] and pet_size in ['mediano']:
        score += 0.15
    elif user_space in ['apto_pequeno'] and pet_size in ['grande', 'muy_grande']:
        score += 0.0
    elif user_space in ['apto_grande'] and pet_size in ['pequeno', 'mediano']:
        score += 0.20
    elif user_space in ['casa_patio', 'casa_jardin'] and pet_size in ['grande', 'muy_grande']:
        score += 0.25
    elif user_space in ['casa_patio', 'casa_jardin']:
        score += 0.20
```

Para entrenar el algoritmo de Filtrado Colaborativo, se utilizaron datos sintéticos *synthetic_interaction.csv*. Se separaron los datos en: 80% para entrenar y 20% para probar, al iniciar el entrenamiento el modelo itera (épocas) tratando de minimizar el error entre su predicción y el rating real. Como se observa en la figura 110 y 111.

Figura 110. Código del modelo colaborativo

```
# 1. Cargar interacciones sintéticas
print("Cargando interacciones...")
DATA_DIR = Path(__file__).parent.parent.parent / 'data' / 'raw'

try:
    interactions_df = pd.read_csv(DATA_DIR / 'synthetic_interactions.csv')
    print(f"✓ (len(interactions_df)) interacciones cargadas")

    # Preprocesar: calcular rating desde interacciones
    print("\nPreprocesando datos...")
    interactions_df['rating'] = (
        interactions_df['clicks_inter'] * 1 +
        interactions_df['favori_inter'] * 2 +
        interactions_df['adopti_inter'] * 5
    )

    # Renombrar columnas al formato esperado
    interactions_df = interactions_df.rename(columns={
        'forane_idxxxx_usuario': 'user_id',
        'forane_idxxxx_mascot': 'pet_id'
    })
```

Figura 111. Entrenamiento del modelo colaborativo

```

PS C:\Users\GIGABYTE GAMING A16\Documents\PM_mimascota\mimascota_ml> python src\models\train_collaborative.py
Cargando .env desde: C:\Users\GIGABYTE GAMING A16\Documents\PM_mimascota\mimascota_backend\.env
Cargando .env desde: C:\Users\GIGABYTE GAMING A16\Documents\PM_mimascota\mimascota_backend\.env
=== ENTRENAMIENTO: COLLABORATIVE FILTERING ===
Cargando interacciones...
/ 2353 interacciones cargadas
Preprocesando datos...
/ Datos preprocesados: rating calculado desde interacciones
Entrenando modelo SVD...
Entrenando modelo Collaborative Filtering (Matrix Factorization)...
Interacciones totales: 2353
Train: 1882 interacciones
Test: 471 interacciones
Entrenando por 28 épocas...
Epoca 10/28 - RMSE: 2.3718
Epoca 20/28 - RMSE: 2.2885
Evaluando en test set...
RMSE: 2.3357
MAE: 1.7884
/ Modelo entrenado
Guardando modelo...
Modelo guardado en: C:\Users\GIGABYTE GAMING A16\Documents\PM_mimascota\mimascota_ml\models\collaborative_model.pkl
/ Modelo guardado

```

Por otro lado, para el algoritmo de Filtrado Basado en Contenido, la fase de entrenamiento se basó en aprender el vocabulario y escala, se usaron los datos de *synthetic_pets.csv*. Por medio de NLP se leyó la información de las mascotas, se construyó un diccionario *TF-IDF* con las 50 palabras más importantes y frases de 2 palabras. Además, se ignoraron palabras que aparecen en más del 80% de las descripciones. Como se observa en figura 112 y 113.

Figura 112. Código del modelo basado en contenido

```

print("*"*70)
print("ENTRENANDO MODELO MEJORADO CON NLP Y FEATURE WEIGHTING")
print("*"*70)

# 1. TF-IDF para descripciones
print("\n[1] Entrenando TF-IDF para descripciones...")

# Usar informacion_adicional si existe, sino usar nombre + especie
if 'informacion_adicional' in sample_pets_df.columns:
    descriptions = sample_pets_df['informacion_adicional'].fillna('').astype(str)
else:
    # Fallback: combinar campos disponibles
    descriptions = (
        sample_pets_df['nombre'].fillna('').astype(str) + ' ' +
        sample_pets_df['especie'].fillna('').astype(str) + ' ' +
        sample_pets_df.get('personalidad', pd.Series([''] * len(sample_pets_df))).fillna('').astype(
            str)
    )

self.tfidf = TfidfVectorizer(
    max_features=50, # 50 features más importantes
    stop_words='spanish',
    ngram_range=(1, 2),
    min_df=1,
    max_df=0.8
)
self.tfidf.fit(descriptions)
print(f" OK - TF-IDF entrenado con max_features=50")

```

Figura 113. Entrenamiento del modelo basado en contenido

```

=== ENTRENAMIENTO: CONTENT-BASED ENHANCED ===
Cargando datos de entrenamiento...
/ 200 mascotas cargadas
Entrenando modelo...
ENTRENANDO MODELO MEJORADO CON NLP Y FEATURE WEIGHTING
=====
[1] Entrenando TF-IDF para descripciones...
OK - TF-IDF entrenado con max_features=50
[2] Configurando mapeo para tamaño (peso 1.5x)...
OK - Mapeo: {'pequeno': 0, 'mediano': 1, 'grande': 2}
[3] Entrenando scaler para edad (peso 1.0x)...
OK - Scaler entrenado (rango: 3 - 115 meses)
[4] Configurando mapeo para energía (peso 1.5x)...
OK - Mapeo: {'tranquilo': 0, 'moderado': 1, 'energetico': 2}
=====
PONDERACIÓN DE CARACTERÍSTICAS
=====
Descripción: 2.0x
Tamaño: 1.5x
Energía: 1.5x
Personalidad: 1.5x
Edad: 1.0x
=====
/ Modelo entrenado
Guardando modelo...
Modelo guardado en: C:\Users\GIGABYTE GAMING A16\Documents\PM_mimascota\mimascota_ml\models\content_based_enhanced.pkl
NOTA: Solo se guardaron los procesadores, NO las mascotas
/ Modelo guardado

```

Por otra parte, para el filtrado híbrido se combinaron los puntajes de los dos modelos anteriores, teniendo un porcentaje del 70% para el filtrado basado en contenido y 30% para el colaborativo, dando en total el 100% de score final. Por medio del comando “*Python api/app_enhanced.py*” se levanta el servidor, como se observa en la figura 114 y 115.

Figura 114 Código del modelo híbrido

```
# 1. Conectar a BD
print("\n[1] Conectando a PostgreSQL...")
db = PetDatabase()
success, message = db.test_connection()

if not success:
    print(f" ERROR: {message}")
    return False

print(f" {message}")

# 2. Cargar modelo híbrido mejorado
print("\n[2] Cargando Hybrid Enhanced (NLP + pesos balanceados)...")
hybrid_model = EnhancedHybridRecommender(
    content_weight=0.7, # 70% content-based enhanced
    collaborative_weight=0.3 # 30% collaborative
)

content_path = MODELS_DIR / 'content_based_enhanced.pkl'
collab_path = MODELS_DIR / 'collaborative_model.pkl'

if not content_path.exists():
    print(f" ERROR: {content_path} no encontrado")
    print(" Ejecuta primero: python src/models/train_enhanced.py")
    return False

if not collab_path.exists():
    print(f" ERROR: {collab_path} no encontrado")
    print(" Ejecuta primero: python src/models/train_collaborative.py")
    return False

hybrid_model.load_models(content_path, collab_path)
```

Figura 115 Servicio de app_enhanced.py

```
[3/3] Combinando scores híbridos...
  Usando pesos: Content=0.7, Collab=0.3

[DEBUG] Ejemplo de scores (primeras 3 mascotas):
  Pet 1: Content=0.326, Collab=6.47 (RAM)
  Pet 2: Content=0.469, Collab=7.29 (RAM)
  Pet 3: Content=0.371, Collab=3.89 (RAM)

[DEBUG] Verificando valores en collab_scores dict:
  Primeros 3 valores: [(1, 6.472653791528447), (2, 7.287992196921115), (3, 3.892671588796399)]
10 recomendaciones generadas
[REQUEST] OK - 10 recomendaciones generadas

RESPUESTA EXITOSA (200 OK)
Recomendaciones recibidas:
C:\Users\GIGABYTE GAMING A16\Documents\PM_mimascota\mimascota_ml\api\database.py:71: UserWarning: pandas only supports SQLAlchemy connectable (Engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested.
Please consider using SQLAlchemy.
pets_df = pd.read_sql_query(query, conn)
- Pet 2: Hybrid=0.508 | Content=0.469 | Collab=0.572
- Pet 39: Hybrid=0.488 | Content=0.521 | Collab=0.411
- Pet 46: Hybrid=0.486 | Content=0.471 | Collab=0.522
- Pet 38: Hybrid=0.449 | Content=0.432 | Collab=0.488
- Pet 49: Hybrid=0.449 | Content=0.475 | Collab=0.398
- Pet 54: Hybrid=0.446 | Content=0.462 | Collab=0.489
- Pet 30: Hybrid=0.440 | Content=0.372 | Collab=0.134
- Pet 16: Hybrid=0.435 | Content=0.469 | Collab=0.353
- Pet 41: Hybrid=0.417 | Content=0.455 | Collab=0.329
- Pet 35: Hybrid=0.416 | Content=0.461 | Collab=0.312
[REQUEST] Retornando 10 recomendaciones enriquecidas

127.0.0.1 - - [16/Feb/2026 00:30:04] "POST /api/ml/recommend HTTP/1.1" 200 -
```

Por último, para evaluar la precisión del modelo colaborativo se usaron las métricas de regresión: *RMSE (Root Mean Square Error)*, la cual mide cuánto se desvían las predicciones del valor real y penaliza mucho los errores grandes y *MAE (Mean Absolute Error)*, que hace un promedio directo de la diferencia absoluta entre el rating predicho y el real. Por otro lado, para la evaluación general de todos los modelos, se aplicaron las métricas de clasificación: *Precision@K*: indica qué porcentaje de los primeros K resultados recomendados son realmente relevantes para el usuario, *Recall@K*: indica qué proporción del total de elementos que le gustan al usuario fueron capturados dentro de los primeros K

resultados y $NDCG@K$: mide la calidad del orden de las recomendaciones. Como se observa en la figura 116, 117, 118 y 119.

Figura 116. Resultado de métrica RMSE Y MAE.

```
Evaluando en test set...
RMSE: 2.3357
MAE: 1.7084
✓ Modelo entrenado
```

Figura 117. Gráficas de resultado de Precision@K

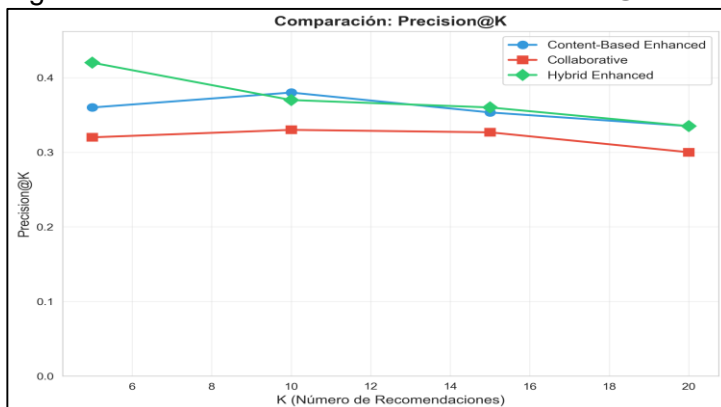


Figura 118. Gráficas de resultado de Recall@K

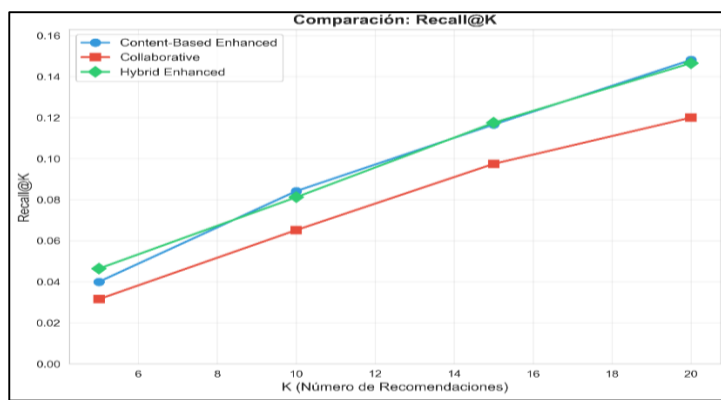
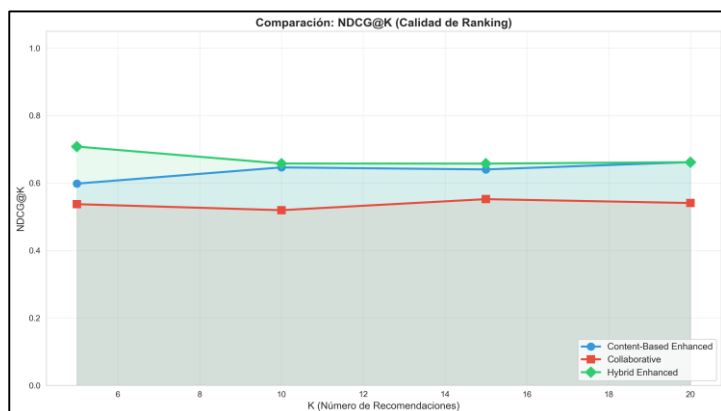


Figura 119. Gráficas de resultado de NDCG@K



4.3.5.1.9. Historia 26: Enviar valoración

En el anexo IV se segmenta la vigésima sexta historia, se realizó el diseño de la interfaz de valoraciones. Cuenta con una sección para que el usuario pueda ingresar su reseña y valorar la página por medio de un rango de estrellas. Además, tiene una sección para ver las reseñas de otros usuarios, la lógica de valoraciones se realiza por medio del componente “Navbar.vue”, dentro de la función “fetchReviews” y “submitReview”, que permiten obtener y enviar la valoración respectivamente. Tal como se observa en la figura 120, 121 y 122.

Figura 120. Interfaz de valoraciones

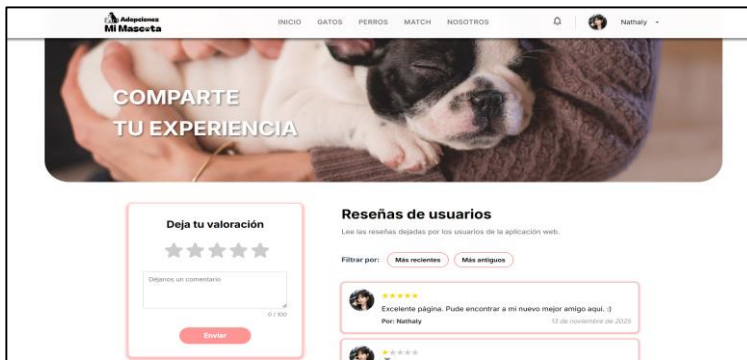


Figura 121. Código de la lógica de obtener valoraciones

```
const fetchReviews = async (page = 1, sortOrder = "DESC") => {
  isLoadingReviews.value = true;
  currentSort.value = sortOrder; // Actualizan el estado del orden
  try {
    const response = await fetch(`${API_BASE_URL}?page=${page}&sortOrder=${sortOrder}`);
    if (!response.ok) {
      throw new Error(`HTTP error! status: ${response.status}`);
    }
    const data = await response.json();
    reviews.value = data.reviews.map(review => ({
      ...review,
      nombre_usuario: review.nombre_usuario,
      avatar_usuario: review.image_usuario,
      nstars_review: review.nstars_review,
      conten_review: review.conten_review,
      fechax_review: review.fechax_review
    }));
    totalPages.value = data.totalPages;
    currentPage.value = data.currentPage;
  } catch (error) {
    console.error('Error al cargar las reseñas:', error);
  } finally {
    isLoadingReviews.value = false;
  }
};
```

Figura 122. Código de la lógica de enviar valoraciones

```
const submitReview = async () => {
  if (newRating.value === 0 || newComment.value.trim() === '') {
    mostrarModal('Por favor, selecciona una valoración y escribe un comentario.', 'error');
    return;
  }
  // 1. Obtener el token de autenticación
  const token = localStorage.getItem('authToken');

  if (!token) {
    mostrarModal('Debes iniciar sesión para dejar una reseña.', 'error');
    router.push('/login'); // Redirigir si no hay token
    return;
  }

  isLoading.value = true;

  const newReviewData = {
    stars: newRating.value,
    comment: newComment.value.trim(),
  };

  try {
    const response = await fetch(API_BASE_URL, {
      method: 'POST',
      // EL Content-Type es necesario para que Express/body-parser entienda el JSON
      headers: {
```

La lógica de control se realizó por medio de *reviewController.js*, la función encargada de procesar las solicitudes respectivas es “*getReviews*” y “*createReview*”. Posteriormente, cada una llama a la función que permite realizar la consulta a la base de datos, de acuerdo con su requerimiento: “*getPaginatedReviews*” o “*createReview*”. Como se observa en la figura 123, 124, 125 y 126.

Figura 123. Código de la función *getReviews*

```

exports.getReviews = async (req, res) => {
  try {
    const page = parseInt(req.query.page, 10) || 1;
    const sortOrder = (req.query.sortOrder || 'DESC').toUpperCase();

    if (sortOrder !== 'ASC' && sortOrder !== 'DESC') {
      return res.status(400).json({ message: 'Parámetro sortOrder inválido.' });
    }

    const offset = (page - 1) * REVIEWS_PER_PAGE;

    const data = await ReviewModel.getPaginatedReviews(REVIEWS_PER_PAGE, offset, sortOrder);

    res.status(200).json({
      status: 'success',
      ...data, // reviews, total, totalPages
      currentPage: page
    });
  } catch (error) {
    res.status(500).json({
      status: 'error',
      message: error.message || 'Error al obtener las reseñas.'
    });
  }
};

```

Figura 124. Código de la función *createReview*

```

exports.createReview = async (req, res) => {
  try {
    const { comment, stars } = req.body;
    // Asumiendo que el ID del usuario se obtiene de la sesión/token de autenticación
    const userId = req.user.id;

    // Validación
    if (!comment || typeof stars !== 'number' || stars < 1 || stars > 5) {
      return res.status(400).json({ message: 'Datos de reseña inválidos. Se requiere comentario y' });
    }

    // Asegurarse de que el comentario no exceda el límite (100)
    if (comment.length > 100) {
      return res.status(400).json({ message: 'El comentario excede el límite de 100 caracteres.' });
    }

    const newReview = await ReviewModel.createReview(userId, comment, stars);

    res.status(201).json({
      status: 'success',
      message: 'Reseña creada exitosamente.',
      review: newReview
    });
  } catch (error) {
    res.status(500).json({
      status: 'error',
    });
  }
};

```

Figura 125. Consulta para crear la reseña

```

static async createReview(userId, content, stars) {
  const query = `
    INSERT INTO reviews (forane_usuario_id, conten_review, nstars_review, eliminado_logico)
    VALUES ($1, $2, $3, FALSE)
    RETURNING idxxxx_review, conten_review, nstars_review, forane_usuario_id;
  `;
  const values = [userId, content, stars];

  try {
    const result = await pool.query(query, values);
    return result.rows[0];
  } catch (error) {
    console.error("Error al crear la reseña:", error);
    throw new Error("Error en la base de datos al crear la reseña.");
  }
}

```

Figura 126. Consulta para obtener las reseñas

```

static async getPaginatedReviews(limit = 6, offset = 0, sortOrder = 'DESC') {
  // Asegurar que la dirección sea una de las dos únicas opciones válidas.
  const orderDirection = (sortOrder === 'ASC') ? 'ASC' : 'DESC';

  // Construir la consulta de forma segura usando concatenación condicional
  // en lugar de interpolación dentro del template literal
  const reviewsQuery = `
  SELECT
    r.idxxxx_review,
    r.conten_review,
    r.nstars_review,
    r.fechaX_review,
    u.nombre_usuario,
    u.imagep_usuario
  FROM
    reviews r
  JOIN
    usuarios u ON r.forane_usuario_id = u.idxxxx_usuario
  WHERE
    r.eliminado_logico = FALSE
  ORDER BY
    r.idxxxx_review ` + orderDirection + `
  LIMIT $1 OFFSET $2;
  `;

  const countQuery = `
  SELECT COUNT(*) as count FROM reviews WHERE eliminado_logico = FALSE;
  `;

```

4.3.5.1.10. Historia 27: Gestión de valoraciones

En el anexo IV se desglosa la vigésima séptima historia, se realizó el diseño del botón para eliminar valoraciones que no sean apropiadas dentro del sistema. Además, se aplicó un modal de confirmación, en donde se muestra un mensaje para reiterar la acción. La lógica se realiza por medio del componente “Navbar.vue”, en la función “deleteReview”, que permite eliminar la reseña seleccionada. Tal como se observa en la figura 127, 128 y 129.

Figura 127. Diseño del botón para gestión de valoraciones

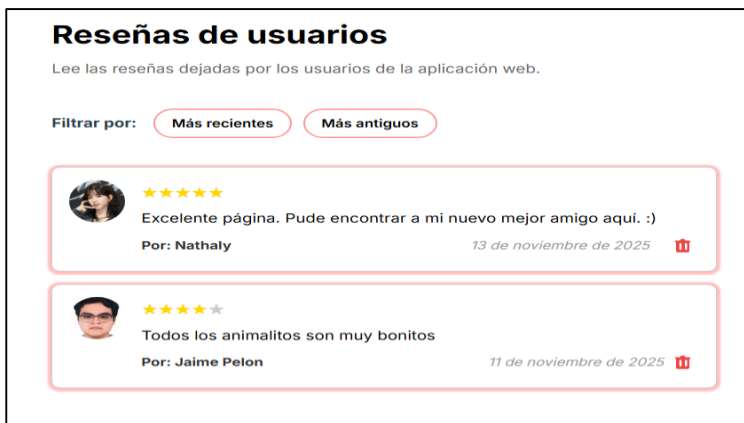


Figura 128. Modal de confirmación

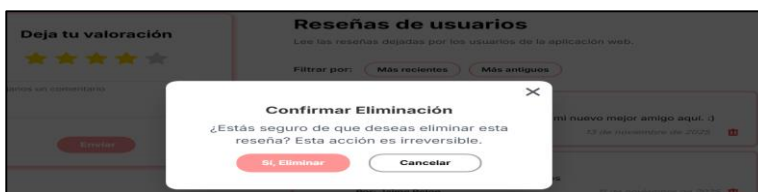


Figura 129. Código de la lógica de gestión de visualizaciones

```

async function deleteReview(reviewId) {
  modal.visible = false
  isDeleting.value = true;
  const token = localStorage.getItem('authToken');

  if (!token) {
    alert('Error: Token de autenticación no encontrado. Inicie sesión. ');
    isDeleting.value = false;
    return;
  }

  try {
    const response = await fetch('http://localhost:3000/api/reviews/${reviewId}', {
      method: 'DELETE',
      headers: {
        'Authorization': 'Bearer ${token}',
      },
    });
  }

  if (response.ok) {
    mostrarModal('Reseña eliminada con éxito.', 'success');
    fetchReviews(currentPage.value, currentSort.value);
  } else if (response.status === 403) {

```

La función encargada de procesar la petición respectiva es “*deleteReview*”, que se encuentra detallada en *reviewController.js*. Posteriormente, llama a la función “*softDeleteByAdmin*” de *reviewModel.js* que permite realizar un eliminado lógico en la base de datos. Como se observa en la figura 130 y 131.

Figura 130. Código de la función deleteReview

```

exports.deleteReview = async (req, res) => {
  const reviewId = req.params.id;
  const userId = req.user.id;
  const userRole = req.user.role;

  if (userRole !== 'Administrador' && userRole !== 'Empleado') {
    // Código 403 Forbidden: Acceso denegado
    return res.status(403).json({ message: 'Acceso denegado. Solo Administradores y Empleados pueden eliminar' });
  }

  if (!reviewId) {
    return res.status(400).json({ message: 'El ID de la reseña es requerido.' });
  }

  try {
    const rowsAffected = await ReviewModel.softDeleteByAdmin(reviewId);
    if (rowsAffected === 0) {
      return res.status(404).json({ message: 'Reseña no encontrada o ya ha sido eliminada.' });
    }
  }

  res.status(200).json({ message: 'Reseña eliminada con éxito.' });
} catch (error) {
  console.error('Error al eliminar reseña:', error);
  res.status(500).json({ message: 'Error interno del servidor al eliminar la reseña.' });
}
};

```

Figura 131. Consulta de eliminación de reseña

```

static async softDeleteByAdmin(reviewId) {
  const query = `
    UPDATE reviews
    SET
      eliminado_logico = TRUE,
      fecha_eliminacion = NOW()
    WHERE idxxxx_review = $1
    AND eliminado_logico = FALSE;
  `;

  try {
    const result = await pool.query(query, [reviewId]);
    return result.rowCount;
  } catch (error) {
    console.error("Error al eliminar lógicamente la reseña (Admin):", error);
    throw new Error("Error en la base de datos al eliminar la reseña.");
  }
}

```

4.3.5.1.11. Historia Técnica 1: Seguridad y privacidad de aplicación

La autenticación de la aplicación se basa en *JSON Web Tokens (JWT)* para verificar la identidad de los usuarios en cada petición protegida. Cuando un usuario inicia sesión correctamente en la función “*loginUser*” del archivo *authController.js*, se genera un *token JWT* firmado con “*jwt.sign*” que contiene el ID del usuario y su rol, con una expiración de 5 horas. Como se observa en la figura 132.

Figura 132. Logica de JSON Web Tokens

```
const loginUser = async (req, res) => {
  const { email_usuario, contra_usuario } = req.body;

  if (!email_usuario || !contra_usuario) {
    return res
      .status(400)
      .json({ message: "Correo y contraseña son requeridos." });
  }

  try {
    // 1. Obtener usuario y rol (Modelo)
    const result = await authModel.findOneWithRoleByEmail(email_usuario);
    const user = result.rows[0];

    if (!user) { ... }
    if (user.eliminado_logico === true) { ... }
    if (!user.mail_usuario) { ... }

    // 2. Comparar contraseña
    const isMatch = await bcrypt.compare(contra_usuario, user.contra_usuario);
    if (!isMatch) { ... }

    // 3. Generar token JWT
    const token = jwt.sign(
      { id: user.id_usuario, role: user.nombre_rol },
      process.env.JWT_SECRET,
      { expiresIn: "5h" },
    );
  }
}
```

Por otro lado, las contraseñas de los usuarios nunca se almacenan en texto plano en la base de datos; en su lugar, se utiliza la librería *bcrypt* para generar un hash seguro antes de guardarlas. En la función “*registerUser*” del archivo *authController.js*, se genera un *salt* aleatorio con *bcrypt.genSalt(10)*, donde 10 representa las rondas de procesamiento y luego se *hashea* la contraseña con *bcrypt.hash(contra_usuario, salt)*, se observa la figura 133.

Figura 133. Lógica de encriptación

```
// 4. Hashear contraseña
const salt = await bcrypt.genSalt(10);
const hashedPassword = await bcrypt.hash(contra_usuario, salt);

// 5. Generar token de verificación
const verificationToken = uuidv4();
const verificationExpires = new Date(Date.now() + 3600000); // Expira en 1 hora

const userData = {
  nombre_usuario,
  email_usuario,
  contra_usuario: hashedPassword,
  celula_usuario,
```

Por medio de *authMiddleware.js* se definen el *middleware* de autorización.

“*authorize(requiredPermissions)*” verifica que el usuario posea todos los permisos requeridos comparando contra *req.user.permissions*, como se observa en la figura 134.

Figura 134. Lógica de autorización de middleware

```
const authorize = (requiredPermissions) => (req, res, next) => {
  // Si no hay un usuario en la solicitud o no tiene permisos, denegar
  if (!req.user || !req.user.permissions) {
    return res.status(403).json({ message: 'Acceso denegado: No se pudieron verificar los permisos.' });
  }

  // Comprobar si el usuario tiene TODOS los permisos requeridos
  const hasAllPermissions = requiredPermissions.every(perm => req.user.permissions.includes(perm));

  if (hasAllPermissions) {
    next();
  } else {
    res.status(403).json({ message: 'Acceso denegado: Permisos insuficientes.' });
  }
};
```

El guardián de navegación global “*router.beforeEach*” en *router/index.js* evalúa cada navegación: si la ruta requiere autenticación (*meta: { requiresAuth: true }*) y el usuario no tiene *token*, lo redirige a */auth*; si ya está autenticado e intenta ir al *login*, lo redirige a */home*. La verificación se realiza con “*isAuthenticated*” en *utils/auth.js*, que comprueba la existencia del *token* en *localStorage*. Todas las páginas funcionales (perfil, mascotas, favoritos, solicitudes, match) están protegidas, tal como se observa en la figura 135.

Figura 135. Lógica Protección de Rutas por medio de Vue-Router

```
{
  path: '/formulario/:formId', // Usamos un path base diferente
  name: 'ViewAdoptionForm', // Nombre único
  component: ViewAdoptionForm,
  meta: { requiresAuth: true }
},
// Ruta comodín para manejar rutas no encontradas
{
  path: '/*',
  name: 'NotFound',
  redirect: '/'
}
]);

// Guardia de navegación global para proteger rutas
router.beforeEach((to, from, next) => {
  const isAuthenticated = isAuthenticated();

  // 1. Si el usuario ya está autenticado e intenta ir a la página de login/registro
  if (to.name === 'auth' && isAuthenticated) {
    // Redirigirlo a la página principal
    next({ name: 'home' });
  }
  // 2. Si la ruta a la que va requiere autenticación y el usuario NO está autenticado
  else if (to.meta.requiresAuth && !isAuthenticated) {
    // Redirige al usuario a la página de autenticación (login)
    next({ name: 'auth' });
  }
  else {
    // En cualquier otro caso, procede
    next();
  }
});
```

La conexión a *PostgreSQL/Supabase* se configura en *config/db.js* con *SSL* habilitado (*ssl: { rejectUnauthorized: false }*) cuando se usa *DATABASE_URL* en producción, encriptando todos los datos en tránsito entre el servidor *Express* y la base de datos. Las

credenciales se manejan mediante variables de entorno con *dotenv* y el archivo “.env”.

Come se observa en la figura 136.

Figura 136. Conexión segura a la base de datos

```

let poolConfig;

if (process.env.DATABASE_URL) {
  // Producción: usar DATABASE_URL (Render + Supabase)
  console.log('Usando DATABASE_URL para la conexión');
  poolConfig = {
    connectionString: process.env.DATABASE_URL,
    ssl: {
      rejectUnauthorized: false
    }
  };
} else if (process.env.DB_USER && process.env.DB_HOST && (process.env.DB_DATABASE || process.env.DB_NAME)) {
  // Local: usar variables individuales
  console.log('Usando variables individuales (DB_USER, DB_HOST, etc.) para la conexión');
  poolConfig = {
    user: String(process.env.DB_USER),
    host: String(process.env.DB_HOST),
    database: String(process.env.DB_NAME || process.env.DB_DATABASE),
    password: String(process.env.DB_PASSWORD || ''),
    port: parseInt(process.env.DB_PORT || '5432', 10),
  };
} else {
  console.error('Error: Faltan variables de entorno para la conexión a la base de datos');
  console.error('Configura DATABASE_URL o las variables: DB_USER, DB_HOST, DB_DATABASE, DB_PASSWORD, DB_

```

La función “*isValidEmailDomain*” en *authController.js* restringe el registro a dominios confiables: *gmail.com*, *outlook.com*, *hotmail.com*, *yahoo.com*, *aol.com* y *protonmail.com*.

Durante el registro, si el dominio no está en la lista blanca, se rechaza con error 400. Esta validación complementa la verificación por correo electrónico (*token UUID* con expiración de 1 hora vía *Brevo API*), creando una doble capa de seguridad en el registro. Se observa en la figura 137.

Figura 137. Lógica de validación de Dominios de Correo

```

// Función de ayuda para validar dominios de correo (VALIDACIÓN ESTRICTA)
const isValidEmailDomain = (email) => {
  const allowedDomains = [
    "gmail.com",
    "outlook.com",
    "hotmail.com",
    "yahoo.com",
    "aol.com",
    "protonmail.com",
  ];

  const parts = email.split("@");
  if (parts.length !== 2) return false;
  const domain = parts[1].toLowerCase();

  return allowedDomains.includes(domain);
};

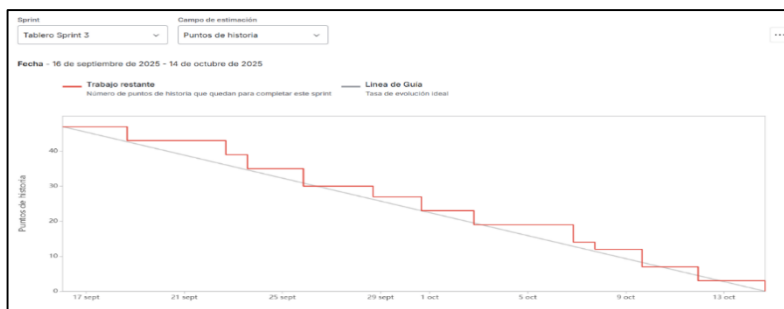
const BREVO_API_URL = "https://api.brevo.com/v3/smtplib/email";

```

4.3.5.2. Sprint 3 – Gráfico de trabajo pendiente

El control de las tareas del tercer grupo de historias se realizó por medio de Jira Software. A través del gráfico de trabajo restante se visualizó el progreso de actividades por parte del equipo de trabajo. En la figura 138, se visualiza el gráfico del segundo sprint.

Figura 138. Gráfica de trabajo restante del sprint 3 en Jira



4.3.5.3. Sprint 3 - Revision (Sprint Review)

Al finalizar el tercer *sprint*, se realizó la reunión respectiva con el gerente del centro veterinario, con el mismo propósito de la revisión del último *sprint*, esta etapa se centró en la implementación completa de los módulos de solicitudes, *machine learning* y valoraciones del usuario.

El seguimiento de las tareas se gestionó nuevamente con *Jira*, y el progreso se vigiló a través del gráfico de trabajo restante del *sprint* 3. Se confirmó que todas las funcionalidades presentadas cumplieran con los criterios de aceptación.

4.3.5.4. Sprint 3 - Retrospectiva (Sprint Retrospective)

Para la retrospectiva del tercer *sprint* se tomaron en cuenta tres preguntas, que permitirán tener una visión clara y precisa sobre cómo se llevó el *sprint*. Como se visualiza en la tabla 23.

Tabla 23. Sprint 3 retrospectiva

Aspectos exitosos del sprint	Mejoras para aplicar	Errores en el sprint
Historias y tareas totales completadas en el tiempo asignado. Se tuvo una asignación eficiente sobre la estimación para cada tarea.	Ampliar el aprendizaje sobre las herramientas utilizadas en el proyecto. Para facilitar el desarrollo de código en futuros proyectos	Problemas para obtener los datos en tiempo real al implementar el apartado de notificaciones

4.4. Resultado general

El resultado general, consistió en la implementación del sistema, validación de la propuesta, y validación de la hipótesis.

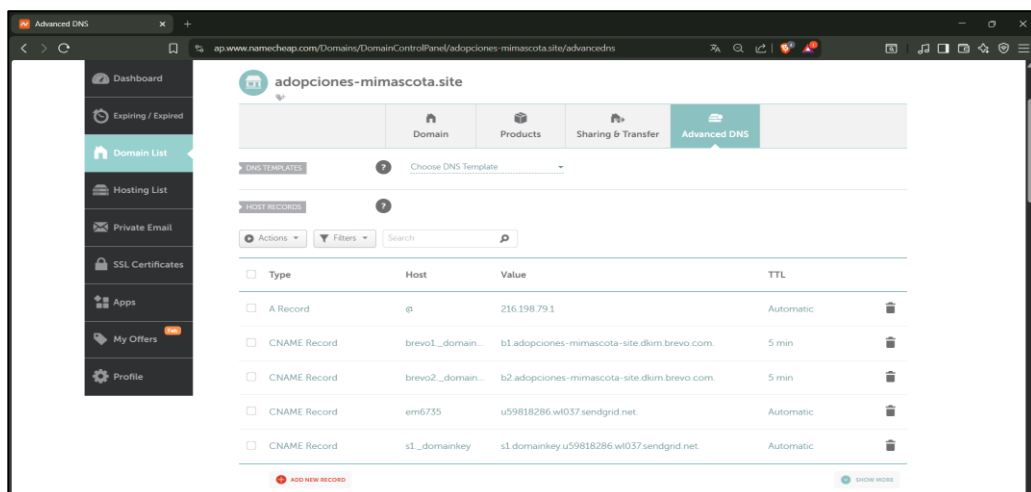
4.4.1. Implementación del sistema

Para la implementación de la aplicación, se realizó mediante una arquitectura distribuida basada en servicios en la nube, priorizando escalabilidad, disponibilidad y separación de responsabilidades entre el *frontend*, *backend* y la base de datos.

4.4.1.1. Configuración y registro de dominio

Como primer paso, se procedió al registro del dominio personalizado “adopciones-mimascota.site” a través de *Namecheap*. El dominio es el punto de acceso público a la aplicación, por ende, fue necesario configurar adecuadamente los *Nameservers* y los registros *DNS* personalizados (*Custom DNS*), permitiendo que el tráfico *web* se dirija correctamente hacia los servicios desplegados. Como se observa en la figura 139.

Figura 139. Configuración del dominio



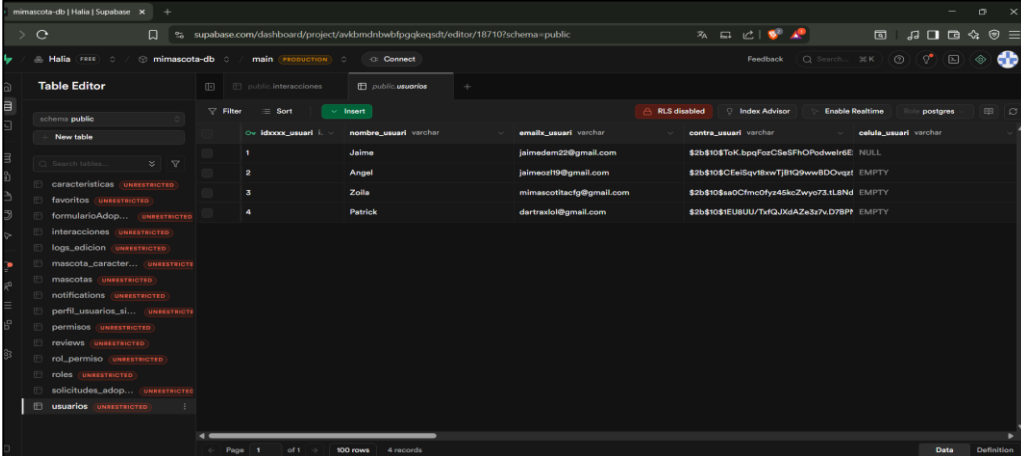
Además, se configuraron los registros correspondientes para que el dominio apunte a la infraestructura del *frontend* desplegada en *Vercel*, asegurando así la correcta resolución del dominio y el acceso seguro mediante *HTTPS*.

4.4.1.2. Configuración de la base de datos en Supabase

La base de datos se implementó mediante *Supabase*, una plataforma que ofrece una instancia de *PostgreSQL* gestionada, también ofrece funcionalidades integradas de autenticación, almacenamiento de archivos y generación automática de *APIs REST/GraphQL*. En esta herramienta se realizaron los siguientes procesos: Creación del

proyecto en *Supabase*, configuración de la instancia de base de datos *PostgreSQL*, diseño y creación de tablas, relaciones entre ellas y restricciones correspondientes, definición y configuración de las variables de entorno, activación y configuración de las políticas de seguridad a nivel de fila (*RLS*). Se observan las tablas en la figura 140.

Figura 140. Configuración base de datos

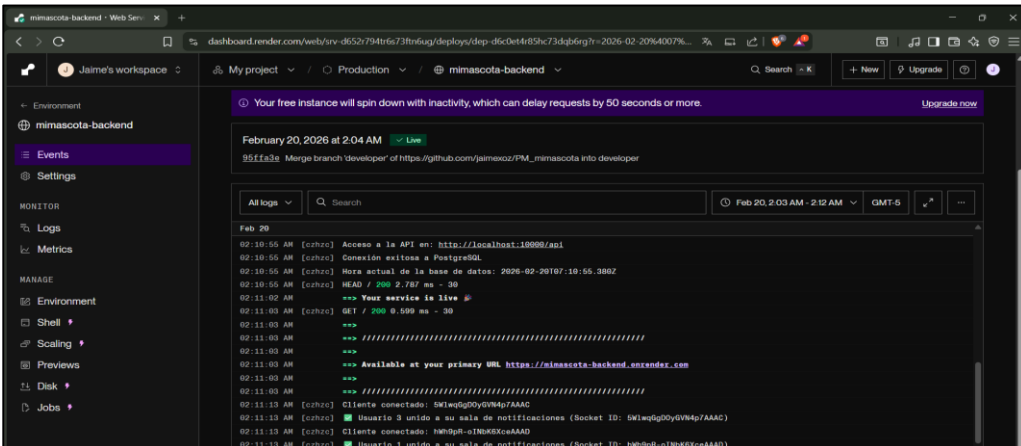


id_usuario	nombre_usuario	email_usuario	contrasena_usuario	celular_usuario
1	Jaime	jaimesm2@gmail.com	\$2b\$10\$ToK.bppForC6e5FhOfodw6r6E	NULL
2	Angel	jaimesm2@gmail.com	\$2b\$10\$C6a5qY8xwTJhQ9wwBD0vqt	EMPTY
3	Zola	mimascotiacf@gmail.com	\$2b\$10\$aaOfmOfy45kcZwyo73.tL6Nd	EMPTY
4	Patrick	detrastol@gmail.com	\$2b\$10\$EUBUUtXqJkAZa3z7n.D7SP	EMPTY

4.4.1.3. Despliegue del Backend en Render

El *backend* fue desplegado en *Render*, donde se implementaron dos servicios independientes. Se desplegó un servicio basado en *Node.js* encargado de: Gestión de usuarios, autenticación, lógica de negocio, conexión con *Supabase* e integración con el servicio de envío de correos. Para ello fue necesario vincular el repositorio de *GitHub*, configurar variables de entorno, definir el puerto de ejecución y establecer el comando de inicio del servidor. Se observa en la figura 141.

Figura 141. Configuración del Backend



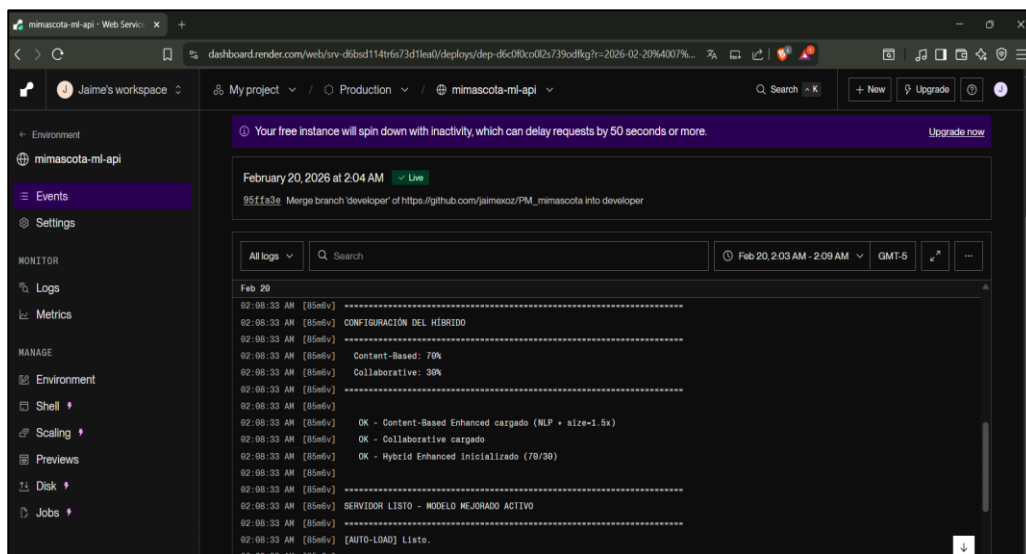
```

February 20, 2026 at 2:04 AM ✔ Live
95ff43a Merge branch 'developer' of https://github.com/jaimexzz/PM_mimascota into developer

All logs
Feb 20
02:10:55 AM [c2hzc] Acceso a la API en: http://localhost:18899/api
02:10:55 AM [c2hzc] Conexión exitosa a PostgreSQL
02:10:55 AM [c2hzc] Hora actual de la base de datos: 2026-02-20T07:10:55.389Z
02:10:55 AM [c2hzc] HEAD / 200 2.787 ms - 36
02:11:02 AM => Your service is live 🎉
02:11:03 AM [c2hzc] GET / 200 0.599 ms - 39
02:11:03 AM =>
02:11:03 AM =>
02:11:03 AM => Available at your primary URL https://mimascota-backend-renderer.com
02:11:03 AM =>
02:11:03 AM =>
02:11:13 AM [c2hzc] Cliente conectado: 5W1wqg0yGVW4p7AAAC
02:11:13 AM [c2hzc] Usuario 3 unido a su sala de notificaciones (Socket ID: 5W1wqg0yGVW4p7AAAC)
02:11:13 AM [c2hzc] Cliente conectado: hNhdpr-c1N6KEXceAAAD
02:11:13 AM [c2hzc] Usuario 1 unido a su sala de notificaciones (Socket ID: hNhdpr-c1N6KEXceAAAD)
  
```

Por otro lado, se desplegó un segundo servicio en *Python* destinado al módulo de *Machine Learning*. Este servicio se encarga de: Procesamiento del modelo de recomendación y ejecución de *endpoints* para comunicación con el *Backend* principal. La separación en dos servicios permite mayor modularidad, mantenimiento independiente y escalabilidad específica del componente de inteligencia artificial. Tal como se observa en la figura 142.

Figura 142. Configuración de modulo machine learning

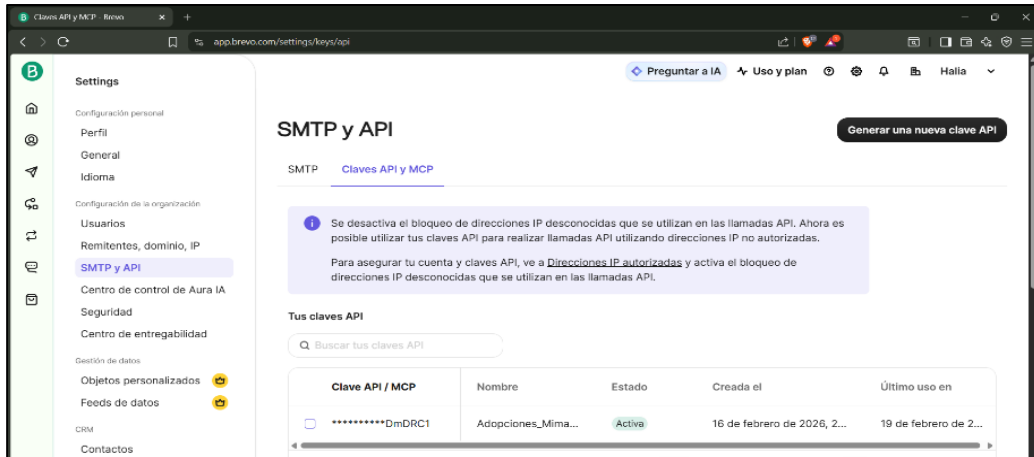


4.4.1.4. Integración del servicio de correos electrónicos

Para gestionar el envío de correos transaccionales, tales como la verificación de cuentas, la recuperación de contraseñas y las notificaciones del sistema, se implementó *Brevo* como servicio de *email*. El proceso de integración comenzó con el registro de una cuenta y la generación de una clave *API*, seguido por la verificación del dominio del remitente mediante la configuración de los registros *DNS* correspondientes en el proveedor (*SPF*, *DKIM* y *DMARC*) para garantizar la autenticidad de los mensajes y minimizar su clasificación como *spam*. Finalmente, para mantener la seguridad, la *API Key* se almacenó como variable de entorno en la plataforma de despliegue *Render*, lo que permitió concretar la conexión del cliente en el *backend* de *Node.js*, ya sea operando a través de un *SMTP*

relay o consumiendo directamente la API REST de Brevo. En la figura 143, se observa la integración de este servicio.

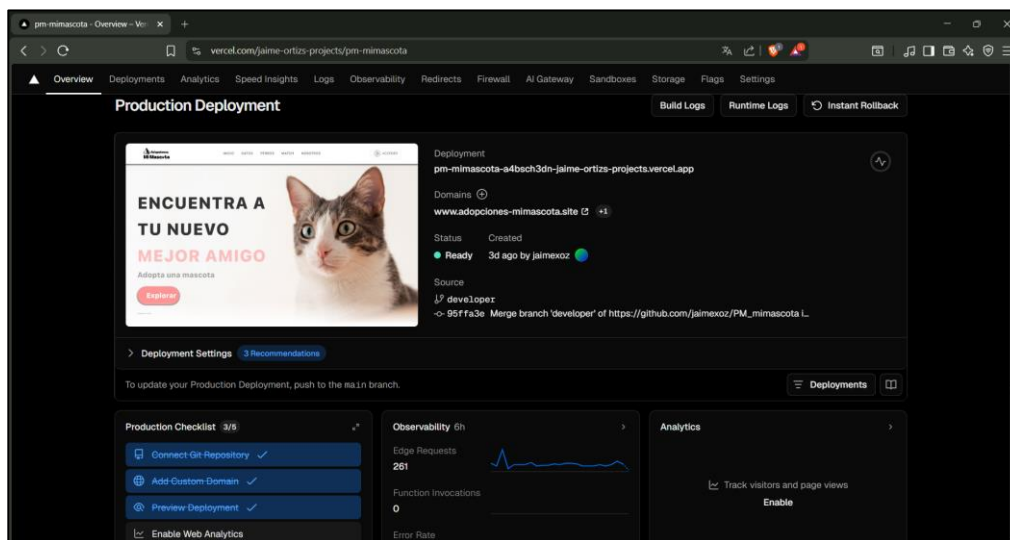
Figura 143. Integración de servicio de correo



4.4.1.5. Despliegue de Frontend en Vercel

Por último, el *frontend* se desplegó en *Vercel* conectándolo directamente al repositorio de *GitHub*. Las etapas realizadas fueron: Importación automática del proyecto desde el repositorio de *GitHub*, configuración de las variables de entorno necesarias (principalmente la *URL* del *backend* y las claves públicas de *Supabase*), asociación y configuración del dominio personalizado, activación automática del certificado *HTTPS* (incluido por defecto en *Vercel*). Como se observa en la figura 144.

Figura 144. Despliegue del Frontend



Como último punto, la aplicación estuvo en producción desde el 12/12/2025. El *PreTest* se realizó el 05/11/2025, después de alrededor de tres meses, se realizó el *PostTest* el 06/02/2026.

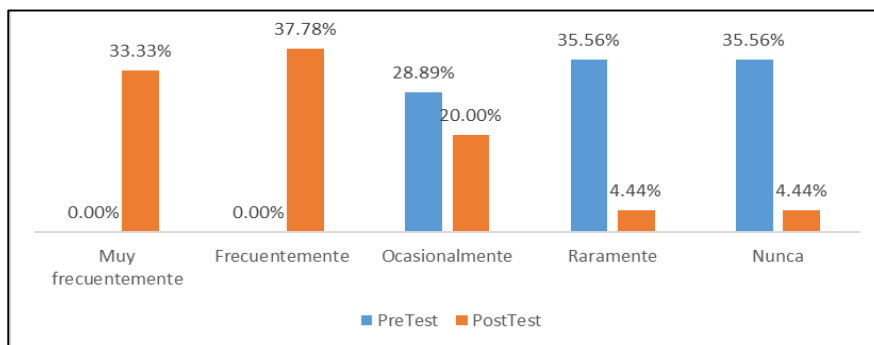
4.4.2. Validación de la propuesta

A través de la aplicación del instrumento de recolección de datos a una muestra de 45 clientes, se dio a conocer la notable satisfacción en cuanto a la gestión de adopción de mascotas. Las preguntas formuladas en la encuesta se encuentran detalladas en el Anexo III.

Se realizó un análisis de los resultados obtenidos mediante la aplicación de la encuesta en *PreTest* y *PostTest*, en donde se compararon 7 preguntas de un total de 26, permitiendo medir el nivel el fortaleciendo de la gestión de adopción de mascotas mediante la aplicación *web*. Los resultados evidenciaron un alto índice de aceptación y un efecto positivo sobre los procesos de adopción de mascotas.

Pregunta 1: ¿Con qué frecuencia recibe recomendaciones sobre que mascota adoptar?

Figura 145. Frecuencia de recomendaciones sobre que mascota adoptar

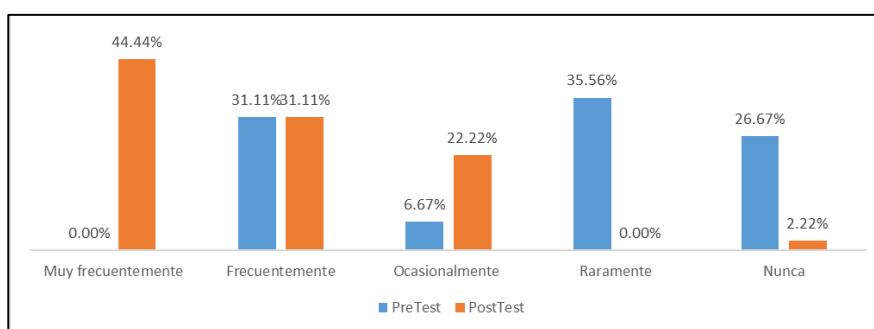


Análisis e interpretación: Los datos obtenidos en el *PreTest* y *PostTest* sobre la frecuencia de recibir recomendaciones sobre que mascota adoptar, se detalla en la figura 145. Se observa que “Nunca” cambia del 35.56% en el *PreTest* frente al 4.44% del *PostTest*. De igual forma, la frecuencia de recibir recomendaciones “Raramente” pasa del 35.56% al 4.44%. En el *PostTest*, alrededor del 37.78% indica recibir recomendaciones “Frecuentemente”, el 33.33% “Muy frecuentemente” y el 20.00% “Ocasionalmente”.

Los resultados obtenidos indican que una gran parte de los clientes reciben recomendaciones sobre que mascotas adoptar tras la integración de aplicativo *web*. El cambio más notable se dio en las categorías “Nunca” y “Raramente”, que experimentaron una disminución del 31.12% cada una; además se percibió un incremento del 37.78% en la categoría “Frecuentemente” y del 33.33% en “Muy frecuentemente”.

Pregunta 2: ¿Con qué frecuencia siente que las mascotas que ve en adopción son de acuerdo a sus preferencias?

Figura 146. Frecuencia de ver mascotas de su preferencia.

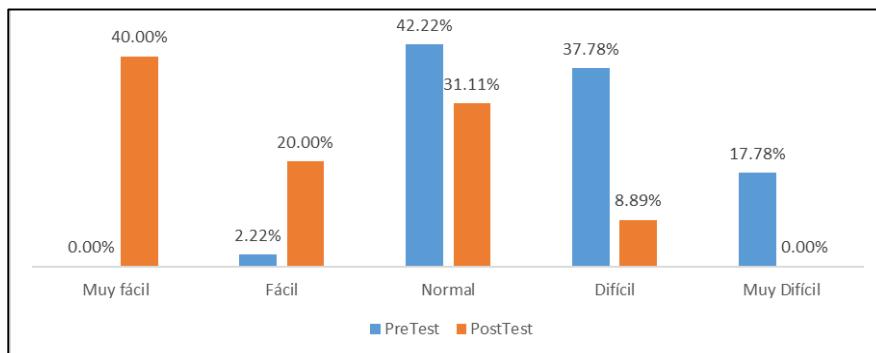


Análisis e interpretación: En la figura 146, se tienen los datos obtenidos en el *PreTest* y *PostTest* con respecto a la frecuencia de ver mascotas de acuerdo a la preferencia del cliente. Durante el *PreTest*, “Raramente” cuenta con el 35.56% y “Nunca” con 26.67% en correspondencia al 0.00% y 2.22% de la fase final *PostTest*. Por otro lado, “Muy frecuentemente” y “Frecuentemente” se tiene el valor de 0.00% y 31.11% del *PreTest* en relación al 44.44% y 31.11% del *PostTest*.

Los datos indican un aumento positivo en mascotas de acuerdo a las preferencias de los usuarios tras la implementación de la aplicación *web*. El impacto más contundente se visualiza en la disminución del 35.56% de “Raramente” y del 24.45% en “Nunca”. Por otro lado, se generó un aumento del 44.44% en “Muy Frecuentemente” y un alza del 15.55% en “Ocasionalmente”.

Pregunta 3: ¿Cuál es su nivel de dificultad para encontrar una mascota en adopción de acuerdo a sus necesidades?

Figura 147. Dificultad de encontrar una mascota de acuerdo a sus necesidades.

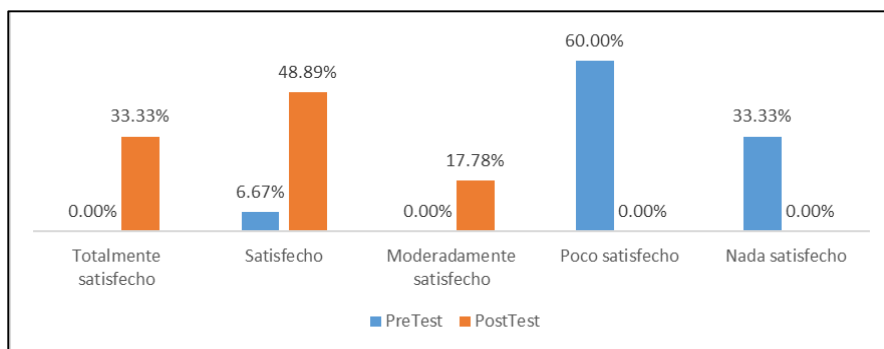


Análisis e interpretación: Los datos obtenidos en el *PreTest* y *PostTest* con respecto al nivel de dificultad para encontrar una mascota de acuerdo a las necesidades, se encuentran en la figura 147. Durante el *PreTest*, “Difícil” cuenta con el 37.78% y “Muy Difícil” con 17.78% en relación al 8.89% y 0.00% de la fase *PostTest*. Por otro lado, “Muy fácil” y “Fácil” se tiene el valor de 0.00% y 2.22% del *PreTest* frente al 40.00% y 20.00% del *PostTest*.

Los resultados demuestran una mejora positiva en cuanto a las mascotas que los usuarios pueden encontrar, alineándose estas a sus necesidades y requerimientos. Se percibe mediante la disminución del 28,89% en “Difícil” y 17,78% en “Muy Difícil” y el aumento del 40.00% en “Muy fácil” y 17.78% en “Fácil”.

Pregunta 4: ¿Qué tan satisfecho está con las opciones disponibles y la facilidad para encontrar una mascota adecuada a sus necesidades?

Figura 148. Disponibilidad de mascotas de acuerdo a sus necesidades

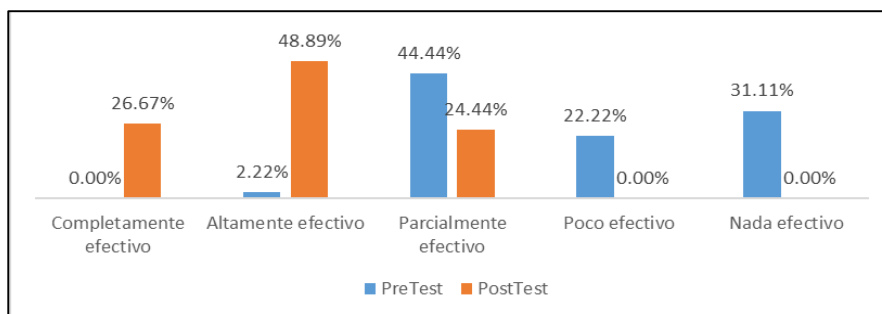


Análisis e interpretación: Las cifras observadas en la figura 148, al nivel de satisfacción con las opciones disponibles y la facilidad para encontrar una mascota adecuada. Durante el *PreTest*, la categoría “Poco satisfecho” cuenta con el 60.00% y “Nada satisfecho” con 33.33% con respecto al 0.00% de la fase *PostTest* en ambos casos. Por otro lado, “Totalmente satisfecho” y “Satisfecho” se tiene el valor de 0.00% y 6.67% del *PreTest* frente al 33.33% y 48.89% del *PostTest*.

Los resultados evidencian el grado de satisfacción de los usuarios con respecto a encontrar mascotas disponibles que se ajusten a sus necesidades y gustos. Se evidencia con la disminución del 60.00% en la categoría “Poco satisfecho” y 33.33% en “Nada satisfecho” y el aumento del 33.33% en “Totalmente satisfecho” y 42.22% en “Satisfecho”.

Pregunta 5: ¿Qué tan efectivo considera que es el proceso de encontrar una mascota en adopción?

Figura 149. Efectividad del proceso de adopción

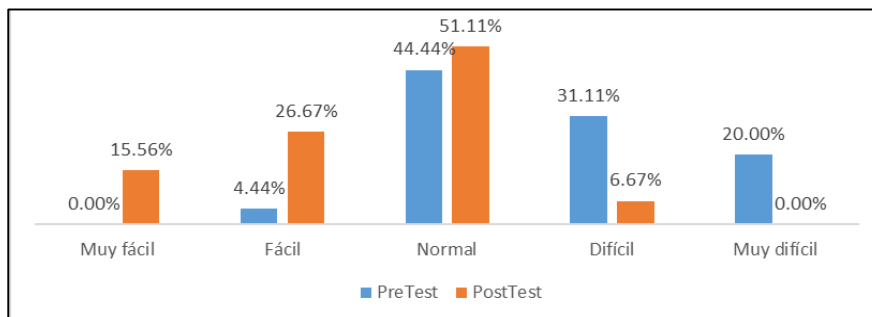


Análisis e interpretación: La información recopilada en el *PreTest* y *PostTest* con respecto a qué tan efectivo se considera el proceso de encontrar una mascota en adopción, se encuentran en la figura 149. Durante el *PreTest*, “Nada efectivo” cuenta con el 31.11% y “Poco efectivo” con 22.22% en relación al 0.00% de la fase del *PostTest* en ambos casos. Por otro lado, “Completamente efectivo” y “Altamente efectivo” se tiene el valor de 0.00% y 2.22% del *PreTest* frente al 26.67% y 48.89% del *PostTest*.

Estos datos indica que la percepción sobre la manera en que se manejan los procesos de adopción ha presentado una notable mejora. Lo cual se refleja en la disminución del 31.11% en “Nada efectivo” y 22.22% en “Poco efectivo” y el aumento del 26.67% en “Completamente efectivo” y 46.67% en “Altamente efectivo”.

Pregunta 6: ¿Cuál es su nivel de dificultad al completar la documentación necesaria para la adopción?

Figura 150. Dificultad para completar la documentación

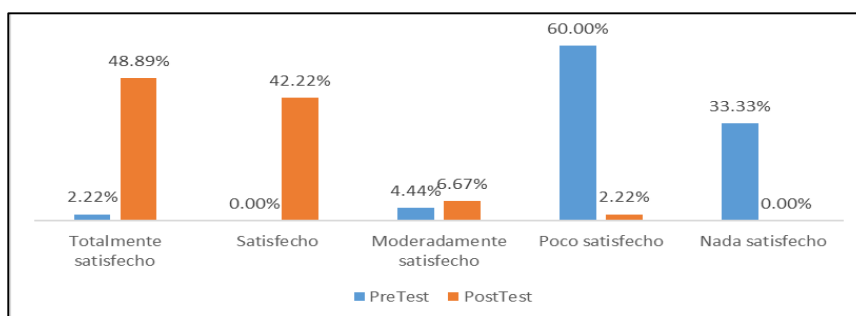


Análisis e interpretación: Los datos obtenidos en el *PreTest* y *PostTest* con respecto al nivel de dificultad al completar la documentación necesaria para la adopción, se encuentran en la figura 150. Durante el *PreTest*, “Difícil” cuenta con el 31.11% y “Muy difícil” con 20.00% en relación al 6.67% y 0.00% de la fase final *PostTest*. Por otro lado, “Muy fácil” y “Fácil” se tiene el valor de 0.00% y 4.44% del *PreTest* frente al 15.56% y 26.67% del *PostTest*.

Los resultados muestran que a los usuarios les resulta mucho más manejable los trámites administrativos para proceder con una adopción. Se percibe mediante la disminución del 24.44% en “Difícil” y 20.00% en “Muy difícil” y el aumento del 15.56% en “Muy fácil” y 22.23% en “Fácil”.

Pregunta 7: ¿Qué tan satisfecho se encuentra con la información que se proporciona de las mascotas?

Figura 151. Información de las mascotas



Análisis e interpretación: Los datos obtenidos en el *PreTest* y *PostTest* con respecto al nivel de satisfacción con la información que se proporciona de las mascotas, se encuentran en la figura 151. Durante el *PreTest*, “Poco satisfecho” cuenta con el 60.00% y “Nada satisfecho” con 33.33% en relación al 2.22% y 0.00% de la fase final *PostTest*. Por otro lado, “Totalmente satisfecho” y “Satisfecho” se tiene el valor de 2.22% y 0.00% del *PreTest* frente al 48.89% y 42.22% del *PostTest*.

Los resultados determinan la gran satisfacción de los usuarios al encontrar información importante de las mascotas en la aplicación *web*. Se percibe mediante la disminución del 57.78% “Poco satisfecho” y 33.33% en “Nada satisfecho” y el aumento del 46.67% en “Totalmente satisfecho” y 42.22% en “Satisfecho”.

4.4.3. Validación de la hipótesis

Tras definir los códigos para los escenarios del *PreTest* y *PostTest*, se realizó el ajuste en la asignación de valores a las respuestas obtenidas de las encuestas. Los detalles de la codificación se presentan en la tabla 24.

Tabla 24. Recodificación de Escenarios

Recodificación	Escenario
0	Sin Aplicación <i>Web</i> y sin <i>Machine Learning</i>
1	Con Aplicación <i>Web</i> y <i>Machine Learning</i>

Para el tratamiento de los datos, se ajustaron las puntuaciones de la escala *Likert* vinculadas a la medición de impacto, siguiendo el esquema presentado en Anexo IX. Esta fase de análisis fue desarrollada con el apoyo de *IBM SPSS*, cuyo procedimiento se ilustra en la figura 152 y 153.

Figura 152. Análisis en el SPSS (IBM Corporation, 2011)

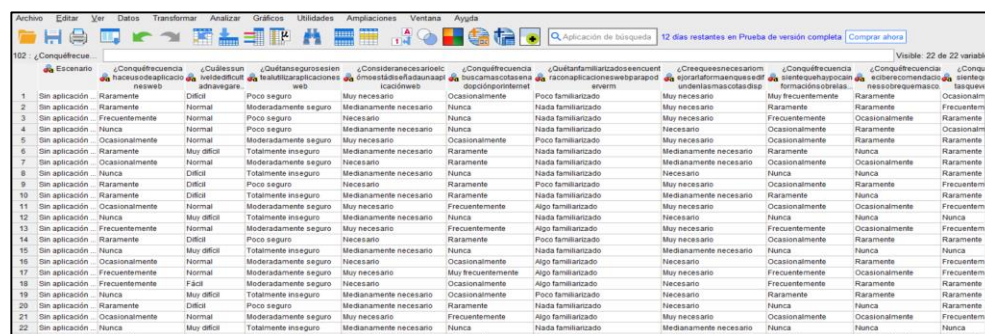
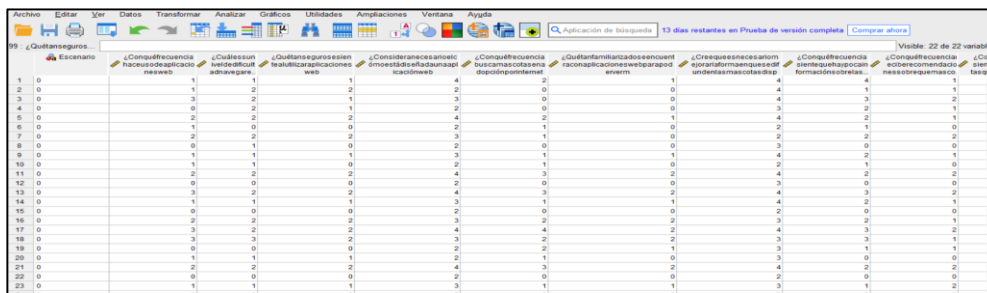


Figura 153. Análisis en el SPSS (IBM Corporation, 2011)



Mediante el uso de regresión logística binaria, con grado de libertad en 1 ($gl=1$), se procesaron los datos del instrumento para identificar puntos determinados en la gestión de adopción de mascotas. Según se ilustra en la tabla 25, se analizaron aspectos como: la frecuencia que recibe recomendaciones, preferencias de adopción, opciones de disponibilidad y la facilidad para encontrar una mascota de acuerdo a sus necesidades, y la satisfacción de la información que se proporciona de las mascotas. Todas las preguntas evaluadas presentan las respuestas con una significación de probabilidad menor a 0.05 ($p < 0.05$).

Tabla 25. Estudio cruzado en base a la aplicación web

Recodificación	Puntuación	gl	Sig.
¿Con qué frecuencia recibe recomendaciones sobre que mascota adoptar?	47.586	1	.001
¿Con qué frecuencia siente que las mascotas que ve en adopción son de acuerdo a sus preferencias?	36.109	1	.001
¿Cuál es su nivel de dificultad para encontrar una mascota en adopción de acuerdo a sus necesidades?	39.729	1	.001
¿Qué tan satisfecho está con las opciones disponibles y la facilidad para encontrar una mascota adecuada a sus necesidades?	65.342	1	.001
¿Qué tan efectivo considera que es el proceso de encontrar una mascota en adopción?	50.617	1	.001
¿Qué tan satisfecho se encuentra con la información que se proporciona de las mascotas?	68.978	1	.001

A partir de los resultados presentados en la tabla 25, se acepta la hipótesis alternativa (H_1). Este hallazgo confirma que la implementación de la Aplicación *Web* con *Machine Learning* incide significativamente en la gestión de adopción de mascotas en el Centro Veterinario Mi Mascota del cantón Santo Domingo.

5. DISCUSIÓN

El desarrollo de este trabajo de investigación se apoyó en el uso del instrumento metodológico encuesta, que permitieron determinar los procesos mencionados en el primer objetivo específico. Por medio del análisis de la reunión de planificación del *sprint* se identificó que la gestión dentro de la veterinaria en cuanto al proceso de adopción de mascotas consistía en registros con poca información sobre las mascotas y poco control sobre la entrega al adoptante. Por otra parte, por medio de la encuesta se pudo evidenciar la información que buscaban los clientes al querer adoptar una mascota, lo que permitió determinar las funcionalidades claves e indispensables en el desarrollo de la aplicación. Esto concuerda con lo que mencionan Celi-Parraga et al. (2023), donde se indica la necesidad de recolectar información con la finalidad de entender los requerimientos que se necesitan integrar dentro del sistema. Este proceso se lo realiza por medio entrevistas, en donde a los participantes se le formulan determinadas preguntas, con el fin de conocer las necesidades que existen (p. 45).

Por otra parte, en el segundo objetivo por medio de tablas comparativas, se seleccionaron las herramientas principales que regirían el proyecto, las cuales son *Vue.js* como *framework* de desarrollo para el *frontend*, *Express.js* para el *backend*, *PostgreSQL* como sistema de gestión de base de datos. Además, se seleccionó un enfoque híbrido para el sistema de recomendaciones de mascotas, en donde se empleó la técnica de aprendizaje no supervisado (*TF-IDF* + similitud de coseno) para el Filtrado Basado en Contenido y la técnica de aprendizaje supervisado (*SVD*) para el Filtrado Colaborativo. Por último, para estructurar el proyecto se utilizó la arquitectura Modelo Vista Controlador (*MVC*). Esta estructura se alinea con el análisis de Richards y Ford (2020), donde mencionan que dividir la funcionalidad del sistema en capas técnicas resulta en una organización lógica y eficiente del código base. Esta estructura garantiza un sistema ordenado donde cada tecnología

cumple su función, facilitando de esta forma a los desarrolladores la localización y mantenimiento del código a largo plazo (p. 103).

Por último, para el tercer objetivo, se desarrolló la aplicación *web* con *machine learning* para la gestión de adopción de mascotas, que cuenta con módulos para llevar un control sobre la información de las mascotas, solicitudes y formularios de adopción. Para la gestión del proyecto, se implementó la metodología *Scrum* que permitió de dividir la complejidad del desarrollo en entregas parciales las cuales se denominan *sprints*. Como referencia Singh y Diko (2023), la metodología *Scrum* permite gestionar proyectos complejos de desarrollo de *software*, ofreciendo directrices para que los equipos de desarrollo trabajen de forma colaborativa e iterativa, permitiendo entregar avances de manera continua (p. 49).

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Se concluye que el desarrollo e implementación de la aplicación *web* dentro de la entidad generó un impacto positivo. Adicionalmente, los resultados obtenidos por medio del análisis de la reunión de planificación del *sprint* con el gerente y encuestas realizadas a los clientes permitieron identificar los puntos claves a implementar por medio de las funcionalidades como control en los procesos de adopción, gestión de la información de mascotas y administración de usuarios, que permitieron fortalecer la gestión de adopción de mascotas en el centro veterinario y lograr que los procesos se desarrollaran de manera eficiente.

La evaluación exhaustiva en cuanto a herramientas tecnológicas permitió definir las que serían utilizadas en el desarrollo del aplicativo *web*. Se utilizó *Vue.js* y *Express.js*, dado a que su sinergia permite lograr una comunicación eficiente entre el cliente y servidor, gracias a esta integración mediante una *API* asíncrona, la aplicación responde de forma rápida y ágil a las acciones del usuario. Adicionalmente, se hizo uso de *PostgreSQL* debido a su capacidad de ofrecer integridad y fiabilidad de los datos, que resulta un punto clave al manejar múltiples transacciones. A su vez, la implementación de *Machine Learning* por medio de la combinación de filtros permitió optimizar sustancialmente la precisión del sistema, ya que, al combinar las fortalezas del filtro colaborativo con el basado en contenido, se lograron superar limitaciones críticas como la escasez de datos y el problema del arranque en frío.

Además, el uso de la metodología *Scrum* facilitó la gestión del proyecto, demostró ser una herramienta esencial gracias a su versatilidad y su capacidad de administrar el proyecto con fluidez, adecuándose a los cambios continuos en el desarrollo del *software*. Esta adaptabilidad fue un punto clave en la correcta organización de las tareas, ya que permitió planificar y asignar prioridades conforme surgían nuevos requerimientos,

garantizando un flujo de desarrollo ordenado y controlado frente a las necesidades cambiantes del proyecto.

Por último, se realizó el análisis estadístico por medio de la comparativa entre los datos obtenidos del *PreTest* y *PostTest*, que permitieron confirmar que la Aplicación *Web* con *Machine Learning* logro un cambio sustancial en cuanto a la gestión de adopción de mascotas, permitiendo no solo fortalecerla, sino también llevar a cabo un control sobre cada uno de los procesos que forman parte de la adopción.

6.2. Recomendaciones

Dado que las necesidades de los usuarios cambian constantemente, se recomienda evaluar la aplicación periódicamente para implementar nuevas funcionalidades que cumplan con las expectativas de los usuarios en cuanto al aplicativo *web*. En este sentido, es importante tomar en cuenta las reseñas dejadas por parte de los usuarios, las cuales serán una guía para identificar puntos críticos de mejora y garantizar una experiencia de usuario satisfactoria.

Para asegurar que el sistema continúe siendo funcional a largo plazo frente a los cambios tecnológicos, es necesario mantenerse actualizado con las nuevas versiones de las herramientas para prevenir problemas de compatibilidad en los servicios. Además, se debe considerar mantener un aprendizaje continuo que permita ampliar y retroalimentar los conocimientos adquiridos, logrando así asegurar que la plataforma permanezca estable y segura.

En cuanto al sistema de *Machine Learning*, se recomienda realizar actualizaciones periódicas o reentrenamiento del modelo para poder integrar los nuevos datos generados en el aplicativo. Es fundamental que el algoritmo procese las interacciones más recientes y los nuevos registros de mascotas ingresadas para ajustar sus parámetros y patrones de decisión.

Para finalizar, la elección correcta del análisis estadístico es fundamental, pues este debe estar perfectamente alineado con la naturaleza y el alcance del estudio. Es clave que los instrumentos de medición sean evaluados por profesionales que tengan un amplio

conocimiento en lo que respecta al tema evaluado, garantizando la validez del contenido del instrumento y asegurando su pertinencia antes de su aplicación definitiva.

7. REFERENCIAS

- Association of Shelter Veterinarians. (29 de Diciembre de 2022). The Guidelines for Standards of Care in Animal Shelters. *Journal of Shelter Medicine and Community Animal Health*. doi:<https://doi.org/10.56771/ASVguidelines.2022>
- Acquaviva, V. (2023). *Machine Learning for Physics and Astronomy*. Princeton University Press. Obtenido de https://books.google.com.ec/books?id=SE2zEAAAQBAJ&pg=PA248&dq=Machine+Learning+for+Physics+and+Astronomy&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwi-ua6j_vqNAxUCr4QIHVEDC6cQ6AF6BAgIEAM
- Alsuwailam, R., Almobarak, R., Aboali, R., Alrubaiea, S., & Aldossary, J. (22 de Mayo de 2022). *Leen: Plataforma web para la adopción de mascotas*. Obtenido de <https://doi.org/10.1109/GlobConET53749.2022.9872481>
- Amazon Web Services. (2025a). *¿Cuál es la diferencia entre MongoDB y PostgreSQL?* Obtenido de AWS: <https://aws.amazon.com/es/compare/the-difference-between-mongodb-and-postgresql/>
- Amazon Web Services. (2025b). *¿Cuál es la diferencia entre SQL y MySQL?* Obtenido de AWS: <https://aws.amazon.com/es/compare/the-difference-between-sql-and-mysql/>
- Arcos, G., Menéndez, J., & Vallejo, J. (20 de Enero de 2018). Comparative Study of Performance and Productivity of MVC and MVVM design patterns. *KnE Engineering*. doi:<https://doi.org/10.18502/keg.v1i2.1498>
- Asamblea Constitucional. (2008). Constitución de la República del Ecuador. Montecristi, Ecuador. Obtenido de https://www.defensa.gob.ec/wp-content/uploads/downloads/2021/02/Constitucion-de-la-Republica-del-Ecuador_act_ene-2021.pdf
- Asamblea Nacional de la República del Ecuador. (2023). LEY ORGÁNICA PARA LA TRANSFORMACIÓN DIGITAL Y AUDIOVISUAL. Quito, Ecuador. Obtenido de

<https://www.asambleanacional.gob.ec/es/system/files/ro-245-3er-supl-07-02-2023.pdf>

Avellaneda, L., Morante, P., Mundaca, J., & Cherre, C. (2022). *Investigación científica:*

Diseños y niveles de investigación. Instituto Latinoamericano de Altos Estudios.

Obtenido de <https://libroselectronicos.ilae.edu.co/index.php/ilae/catalog/book/337>

Balamanigandan, R., Sivakumar, T. B., Thumilvannan, S., & Sagaya, A. (2024). *Machine*

Learning and Neural Networks Essentials. SK Research Group of Companies.

Obtenido de

[https://books.google.com.ec/books?id=7DBDEQAAQBAJ&pg=PA35&dq=random+forest+machine+learning&hl=es-](https://books.google.com.ec/books?id=7DBDEQAAQBAJ&pg=PA35&dq=random+forest+machine+learning&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwiN8aXBo_uNAxWTRTABHb4BNslQ6AF6BAgJEAM)

[419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwiN8aXBo_uNAxWTRTABHb4BNslQ6AF6BAgJEAM](https://books.google.com.ec/books?id=7DBDEQAAQBAJ&pg=PA35&dq=random+forest+machine+learning&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwiN8aXBo_uNAxWTRTABHb4BNslQ6AF6BAgJEAM)

Barker, J. (2025). *Hackeados Los secretos detrás de los ciberataques*. Ediciones Granica.

Obtenido de

https://elibro.puce.elogim.com/es/ereader/puce/280510?as_all=Hackeados__Los__secretos__detr%C3%A1s__de__los__ciberataques&as_all_op=unaccent__icontains&prev=as

Bhasin, H. (2023). *Machine Learning for Beginners Build and Deploy Machine Learning*

Systems Using Python. BPB Publications. Obtenido de

[https://books.google.com.ec/books?id=0xPdEAAAQBAJ&pg=SA1-](https://books.google.com.ec/books?id=0xPdEAAAQBAJ&pg=SA1-PA13&dq=Machine+Learning+for+Beginners+Build+and+Deploy+Machine+Learning+Systems+Using+Python&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjs-MfutZeOAxWSRTABHeCdOZkQ6AF6BAgMEAM)

[PA13&dq=Machine+Learning+for+Beginners+Build+and+Deploy+Machine+Learning+Systems+Using+Python&hl=es-](https://books.google.com.ec/books?id=0xPdEAAAQBAJ&pg=SA1-PA13&dq=Machine+Learning+for+Beginners+Build+and+Deploy+Machine+Learning+Systems+Using+Python&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjs-MfutZeOAxWSRTABHeCdOZkQ6AF6BAgMEAM)

[419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjs-](https://books.google.com.ec/books?id=0xPdEAAAQBAJ&pg=SA1-PA13&dq=Machine+Learning+for+Beginners+Build+and+Deploy+Machine+Learning+Systems+Using+Python&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjs-MfutZeOAxWSRTABHeCdOZkQ6AF6BAgMEAM)

[MfutZeOAxWSRTABHeCdOZkQ6AF6BAgMEAM](https://books.google.com.ec/books?id=0xPdEAAAQBAJ&pg=SA1-PA13&dq=Machine+Learning+for+Beginners+Build+and+Deploy+Machine+Learning+Systems+Using+Python&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjs-MfutZeOAxWSRTABHeCdOZkQ6AF6BAgMEAM)

Briones, G. (2002). *Metodología de la investigación cuantitativa en las ciencias sociales*.

Bogotá: ARFO Editores e Impresores Ltda.

Byri, A., Govindarajan, B., Vadlamani, S., & Goel, O. (2024). *ADVANCED JAVA*

PROGRAMMING: PATTERNS, FRAMEWORKS, AND BEST PRACTICES.

DeepMisti Publication. Obtenido de

https://books.google.com.ec/books?id=zgssEQAAQBAJ&pg=PA33&dq=frameworks&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwid5_-Lz8OQAxXnfDABHSm1E8kQ6AF6BAgNEAM

Canva. (2025). *Canva*. Obtenido de <https://www.canva.com/>

Celi-Párraga, R. J., Boné-Andrade, M. F., Mora-Olivero, A. P., & Sarmiento-Saavedra, J. C. (2023). *Ingeniería del Software I: Requerimientos y Modelado del Software*. Santo Domingo, Ecuador: Editorial Grupo AEA.
doi:<https://doi.org/10.55813/EGAEA.L.2022.21>

Chandra, V., & Hareendran, A. (2021). *Machine Learning a practitioner's approach*. PHI

Learnig Private Limited. Obtenido de

https://books.google.com.ec/books?id=b_wdEAAAQBAJ&pg=PA31&dq=machine+learning+decision+tree&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjomeScnvuNAXARDABHcDYHtYQ6AF6BAgNEAM

Chopra, R. (2018). *Deep Learning*. KHANNA PUBLISHING HOUSE. Obtenido de

https://books.google.com.ec/books?id=VjsEEAAAQBAJ&pg=PA62&dq=rajiv+chopra+naive+bayes&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwj31p6KksOQAxXcRTABHXoIASEQ6AF6BAgIEAM

Congreso de la Ciudad de México. (2021). INICIATIVA CON PROYECTO DE DECRETO POR EL QUE SE ADICIONA UN ARTÍCULO AL CÓDIGO PENAL PARA EL DISTRITO FEDERAL, EN MATERIA DE ABANDONO ANIMAL. Ciudad de México.

Cruz, C., Olivares, S., & González, M. (2014). *Metodología de la investigación*. Grupo Editorial Patria.

Cybellium. (2024). *Mobile App Development Exam Study Essentials A Comprehensive Guide to Mobile App Development Concepts for Exams*. Cybellium Ltd. Obtenido de <https://books.google.com.ec/books?id=5mAsEQAAQBAJ&pg=PA149&dq=Mobile+A>

pp+Development+Exam+Study+Essentials&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwiB_s7w0PaNAXWIRDABHQ6UBmgQ6AF6BAgHEAM

De Marco, P. (2025). *Amazing Pets*. Australia. Obtenido de https://books.google.com.ec/books?id=UfdLEQAAQBAJ&pg=PA126&dq=Pet+Adoption&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjy-fmwZeOAxWNTTABHQzYFsUQ6AF6BAgJEAM

Domínguez, M., & Rivas, F. (2021). *Yo cuido a mi mascota Yo cuido a mi mascota*. Zig-Zag. Obtenido de https://books.google.com.ec/books?id=yAdDEAAAQBAJ&pg=PA14&dq=elegir+una+mascota&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwj63-_K1PuNAXUNTTABHV7uNgYQ6AF6BAgFEAM

Dong, H., Zhang, S., & Ding, Z. (2020). *Deep Reinforcement Learning Fundamentals, Research and Applications*. Springer Nature Singapore. Obtenido de https://books.google.com/books?id=31juDwAAQBAJ&printsec=frontcover&dq=Fundamentals+of+Reinforcement+Learning+pdf&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjImsTijfuNAXXLRzABHbenBBEQ6AF6BAgJEAM

Echeverría-Galindo, J. C., & Moncada-Rangel, J. A. (01 de Mayo de 2024). Percepción del conflicto gente – animales de compañía en el cantón Antonio Ante, Ecuador: Propuestas educativas ambientales. Perception of the people - companion animal conflict in the Antonio Ante canton, Ecuador: environmental educational proposals. *Aibi revista de investigación, administración e ingeniería*, 91. doi:<https://doi.org/10.15649/2346030X.3592>

Express. (2026). *Express Fast, unopinionated, minimalist web framework for Node.js*. Obtenido de <https://expressjs.com/>

Fink, A. (2019). *Conducting Research Literature Reviews From the Internet to Paper*. California: SAGE Publications. Obtenido de

https://www.google.com.ec/books/edition/Conducting_Research_Literature_Reviews/IVh_DwAAQBAJ?hl=es-419&gbpv=0

Gobierno Autónomo Descentralizado Municipal de Guayaquil. (16 de Febrero de 2023).

Ordenanza que regula la protección, tenencia y control de la fauna urbana en el cantón Guayaquil. Obtenido de <https://www.guayaquil.gob.ec/wp-content/uploads/Documentos/Gacetas/Periodo%202019-2023/Gaceta-58.pdf>

Hapke, H., & Nelson, C. (2020). *Building Machine Learning Pipelines*. O'Reilly Media.

Obtenido de

https://books.google.com/books?id=Ha_wDwAAQBAJ&printsec=frontcover&dq=pipeline+machine+learning&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwj_x97ktpeOAxUzSzABHUyUDmEQ6AF6BAgGEAM

Hernández, R., & Mendoza, C. (2018). *Metodología de la investigación*. McGRAW-HILL

INTERAMERICANA EDITORES, S.A.

IBM. (01 de Julio de 2025). *Spring Framework*. Obtenido de IBM:

<https://www.ibm.com/docs/es/was/9.0.5?topic=applications-spring-framework>

IBM. (2026). *Django*. Obtenido de ¿Qué es Django?: [https://www.ibm.com/es-](https://www.ibm.com/es-es/think/topics/django)

[es/think/topics/django](https://www.ibm.com/es-es/think/topics/django)

IBM Corporation. (2011). *SPSS Statistics Versión 20*. IBM.

Joshi, A. (2022). *Machine Learning and Artificial Intelligence*. Springer International

Publishing. Obtenido de

https://www.google.com.ec/books/edition/_/S9ekEAAAQBAJ?hl=es-419&sa=X&ved=2ahUKEwjk68CXqPqNAXUCSDABHVm9FFYQre8FegQIBBAI

Karunamurthy , A. (2024). *Cybersecurity and IoT*. Puducherry: Quing Publications. Obtenido

de

<https://books.google.com.ec/books?id=mHhVEQAAQBAJ&pg=PA39&dq=security+web&hl=es->

419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwj2mYyJIPqNAXWdRTABHXJ
GJCAQ6AF6BAgIEAM

- Kong, W. E., Haw, S. C., Tai, T. E., Naveen, P., & Ng, K. W. (2024). E-learning Recommender System based on k-Nearest Neighbor (KNN), Singular Value Decomposition (SVD), and CoClustering Approaches. *International Journal on Advanced Science Engineering Information Technology*.
- Li, H., Zhou, Y., & Li, Z. (2024). *Design and Implementation of Pet Recommendation System based on Machine Learning*. Obtenido de <https://doi.org/10.1109/ITNEC60942.2024.10732980>
- López Puetate, J. A. (05 de Abril de 2022). Aplicación móvil para tenencia responsable y adopción de mascotas "Patitas Felices" "Patitas Felices" mobile application for responsible pet ownership and pet adoption. 6,9-10,18.
doi:<https://doi.org/10.62465/riif.v1n1.2022.1>
- Manglik, R. (2024a). *Advanced Database Architecture and Management*. EduGorilla Publication. Obtenido de https://books.google.com.ec/books?id=oXFdEQAAQBAJ&pg=PR138&dq=sql+datab+ase&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwif-M_0g_qNAXVeSzABHVn3MzQQ6AF6BAgEEAM
- Manglik, R. (2024b). *Data Science and Machine Learning*. EduGorilla Publication. Obtenido de <https://books.google.com.ec/books?id=-gU8EQAAQBAJ&printsec=frontcover>
- Manglik, R. (2024c). *Principles of Programming Languages*. EduGorilla Publication. Obtenido de https://books.google.com.ec/books?id=UYhIEQAAQBAJ&newbks=0&dq=Principles+of+Programming+Languages+Rohit+Manglik&source=gbs_navlinks_s
- Moscato, R. (2024). *Web App Development Made Simple with Streamlit A Web Developer's Guide to Effortless Web App Development, Deployment, and Scalability*. Packt Publishing. Obtenido de <https://books.google.com.ec/books?id=pLfuEAAAQBAJ&pg=PA5&dq=web+app&hl=>

es-

419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjmx6HPnZOOAxWAtQIHR
AwKG0Q6AF6BAgKEAM

Ojeda Colcha, G. S., Pachay Guaño, C. A., Peñafiel Oleas, J. H., & Guambo Llerena, M. Á. (20 de Diciembre de 2024). El abandono indiscriminado de animales de compañía a la proliferación de fauna urbana Indiscriminate abandonment of companion animals to the proliferation of urban fauna. *Revista Arbitrada de Ciencias Jurídicas y Sociales*, 308. doi:<https://doi.org/10.62574/mtsar765>

Ortega, J. (2023). *Big data, machine learning y data science en Python*. RA-MA. Obtenido de <https://bibliotecadigitalpuce.puce.elogim.com/library/publication/big-data-machine-learning-y-data-science-en-python-1744232291>

Pérez Ruíz, J. S., & Nisperuza, J. (Abril de 2019). Adoptapp. Aplicación móvil de adopción de animales domésticos que habitan las calles de Bogotá* Adoptapp. Mobile application for the domestic animal adoption that inhabit in Bogotá's streets. *Designio. Investigación en diseño gráfico y estudios de la imagen*, 75-118. doi:10.52948/ds.v1i1.57

Pérez, M. (2024). *Desconexión Digital y Seguridad Informática en el Trabajo*. ICB Editores. Obtenido de <https://digitalia.puce.elogim.com/a/175396/desconexion-digital-y-seguridad-informatica-en-el-trabajo>

Presidencia de la República del Ecuador . (2019). Reglamento al Código Orgánico del Ambiente. Obtenido de <https://site.inpc.gob.ec/pdfs/lotaip2020/REGLAMENTO%20AL%20CODIGO%20ORGANICO%20DEL%20AMBIENTE.pdf>

Ramana, T. V., Pradeep, G. S., Sathiyaraj, R., & Khan, M. (2024). *Artificial Intelligence and Machine Learning for Smart Community Concepts and Applications*. CRC Press. Obtenido de <https://books.google.com.ec/books?id=SyHnEAAAQBAJ&pg=PP34&dq=Machine+le>

arning&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwixvY-
pi_uNAXURRjABHX2BMS0Q6AF6BAgFEAM

Richards, M., & Ford, N. (2020). *Fundamentals of Software Architecture An Engineering Approach*. O'Reilly Media. Obtenido de
https://books.google.com/books?id=xa7MDwAAQBAJ&printsec=frontcover&dq=richards+%26+ford+fundamentals+of+software+architecture&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwifq67FpuuRAxWwTDABHT5XFOEQ6AF6BAgNEAM

Sánchez, C., & Mosquera, F. (2020). *Modelamiento de base de datos Metodología práctica y aplicada*. Siglo del Hombre Editores. Obtenido de
https://books.google.com/books?id=j_xTEAAQBAJ&printsec=frontcover&dq=modelado+de+datos+base+de+datos&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwi21qHRopOOAxW2QzABHZrFK_sQ6AF6BAgEEAM

Sarasa, A. (2019). *Introducción a las bases de datos NSQL clave-valor usando Redis*. Editorial UOC, S.L. Obtenido de
https://books.google.com.ec/books?id=Q6i8DwAAQBAJ&pg=PT14&dq=lenguajes+base+de+datos+nosql&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjhc6fjZWOAxXvRTABHXB BMFYQ6AF6BAgMEAM

Secretaría Metropolitana de Salud. (2020). *Informe adopciones campaña "adopta a tu mejor amigo"*. Quito. Obtenido de
https://www7.quito.gob.ec/mdmq_ordenanzas/Administraci%C3%B3n%202019-2023/Sesiones%20de%20Concejo/2020/Sesi%C3%B3n%20077%20Extraordinaria%202020-07-09/II.%20Fauna%20Urbana/2.2.%20Informe%20Campa%C3%B1a%20Adopci%C3%B3n.pdf

- Secretaría Nacional de Planificación. (2024). *Plan de Desarrollo para el Nuevo Ecuador 2024 - 2025*. Obtenido de <https://www.planificacion.gob.ec/wp-content/uploads/2024/08/RESUMEN-PND-ES.pdf>
- Shankar, U., & Spanias, A. (2022). *Machine and Deep Learning Algorithms and Applications*. Morgan & Claypool Publishers. Obtenido de https://books.google.com/books?id=C_9VEAAQBAJ&printsec=frontcover&dq=Machine+Learning+Algorithms&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjvtNmykfuNAxX9STABHQm2CEQQ6AF6BAgMEAM
- Singh, H., & Diko, D. (2023). *Agile Fundamentals: Scrum With Jira Introduction*. Marc & Zed | Publishing. Obtenido de https://books.google.com.ec/books?id=wym3EAAQBAJ&pg=PA97&dq=Agile+Fundamentals:+Scrum+With+Jira+Introduction+pdf&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwje3O-Nn-uRAXUPQTABHbvRMCMQ6AF6BAgMEAM
- Supo, J. (2012). *Seminarios de Investigación Científica: Metodología de la investigación para las ciencias de la salud*. CreateSpace Independent Publishing Platform. doi:1477449043
- Valderrey Sanz, P. (2020). *Administración de sistemas gestores de base de datos*. Madrid: Ra-Ma Editorial.
- Vázquez, J., & Lorenzo, D. (2020). *Ruby on rails Aprende a crear aplicaciones web desde cero*. Bogotá: Alpha Editorial. Obtenido de https://books.google.com.ec/books?id=xXJ6EAAQBAJ&pg=PA20&dq=editores+de+c%C3%B3digo&hl=es-419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjwodDq-vaNAxUWTDABHcPWEUQ6AF6BAgJEAM
- Vega Briceño, E. (2021). *Seguridad de la información*. 3Ciencias. Obtenido de <https://books.google.com.ec/books?id=nx4uEAAQBAJ&pg=PA30&dq=Factor+de+a>

utenticaci%C3%B3n&hl=es-

419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjQq63KIPqNAXVaQzABHUP

JM-8Q6AF6BAgFEAM

Vinueza Veloz, P., Vinueza, A. F., Aldaz Pachacama, N. R., & Álvarez Almeida, K. M.

(2022). Factores asociados al tiempo de adopción de perros procedentes de refugios en Ecuador. Un estudio observacional longitudinal. *AlfaPublicaciones*, 4(2).

doi:<https://doi.org/10.33262/ap.v4i2.198>

Vyas, R. (2022). Comparative Analysis on Front-End Frameworks for Web Applications.

International Journal for Research in Applied Science & Engineering Technology

(IJRASET), 298-307. doi:<https://doi.org/10.22214/ijraset.2022.45260>

Wingfield, W. E., & Raffe, M. R. (2020). *The Veterinary ICU Book*. CRC Press. Obtenido de

<https://books.google.com.ec/books?id=iUDpDwAAQBAJ&pg=PT987&dq=veterinary+medical+history&hl=es->

419&newbks=1&newbks_redir=1&sa=X&ved=2ahUKEwjB75WW_5qOAxU-

STABHYFUB9AQ6AF6BAgMEAM

Zhou, Y. (2023). Applications of Blockchain Techniques in Pet Adoption. *BCP Business &*

Management, 44, 42-47. doi:<https://doi.org/10.54691/bcpbm.v44i.4791>

Anexo III: Validación de instrumentos de recolección de datos

Santo Domingo, 06 de julio del 2025

Estimada Médico Veterinaria Brithany Velasco,
De mi consideración:


El motivo del presente es que le hemos elegido a usted para redactar la solicitud de revisión y validación de los instrumentos de recolección de datos.

A continuación, encontrará la entrevista y encuesta que contienen las preguntas que permitirá recolectar información de acuerdo al trabajo de titulación de grado "APLICACION WEB CON MACHINE LEARNING PARA LA GESTION DE ADOPCIONES DE MASCOTAS EN EL CENTRO VETERINARIO MI MASCOTA EN EL CANTON SANTO DOMINGO", dirigida a los clientes y gerente del centro veterinario Mi Mascota.


Para la validación de los instrumentos se adjunta la operacionalización de variables, con la finalidad de que se visualice la relación de las preguntas con las categorías e indicadores. Además, se encuentran divididos los instrumentos en dos partes, la primera corresponde a la entrevista (preguntas de **fondo teórico** para las dos variables) y la segunda a la encuesta (preguntas de la variable independiente de **fondo teórico** y las preguntas de la variable dependiente Fondo gráfico).

Gracias por su valiosa colaboración en este trabajo de titulación de grado.

Ateentamente,



Jaime Arturo Ortiz Jaramillo
CI: 0700442170
jaortiz@pucc.edu.ec



Nathaly Jarriza Zamora Mendez
CI: 0910421055
njzamoram@pucc.edu.ec

Operacionalización de la variable independiente aplicación web

Conceptualización	Categorías	Indicadores	Preguntas	Herramienta
Aplicación Web (OW), menciona que es una aplicación de software que se ejecuta dentro de un navegador web, tiene como propósito conceder acceso a las capacidades del software y a los datos a través de una red mediante tecnologías estándar, permitiendo a los usuarios acceder a las funcionalidades desde cualquier dispositivo. Además, las aplicaciones web son útiles cuando se desea proporcionar acceso a software y datos a una gran cantidad de usuarios especialmente dentro de una organización (p. 5).	Desarrollo de software	Lenguajes de Programación	¿Con qué frecuencia hace uso de aplicaciones web? ¿Qué es su nivel de dificultad navegar en aplicaciones web?	Encuesta al cliente
		Frameworks	¿Considera necesario el conocimiento de una aplicación web? ¿Que tan familiarizado se encuentra con aplicaciones web para poder evaluar procesos de adopción?	Encuesta al cliente
Base de Datos	Base de Datos	Editor de código fuente	¿Considera vital la implementación en sus aplicaciones web que permita gestionar la adopción de mascotas?	Revisión de planificación del sprint con el gerente
		Modelado de datos	¿Que atención usan para mejorar y los recursos que están disponibles para mejorar?	Revisión de planificación del sprint con el gerente
Seguridad web	Seguridad web	Lenguaje de base de datos	¿Considera que la información sobre las mascotas y los adoptantes es confiable?	Revisión de planificación del sprint con el gerente
		Tipos de base de datos	¿Que tipo de información consideran que debe ser visible en la aplicación web?	Revisión de planificación del sprint con el gerente
Seguridad web	Seguridad web	SGBD	¿Con qué frecuencia asiste que hay poca información sobre las mascotas en adopción?	Encuesta al cliente
		Factor de autentificación	¿Qué de acuerdo es que se implementen sistemas de verificación de datos?	Encuesta al cliente
Seguridad web	Seguridad web	Encriptación de datos	¿La página web que se va implementar web, los procedimientos de seguridad para la publicación?	Revisión de planificación del sprint con el gerente
		Protocolos de seguridad	¿Que tan seguro se siente al utilizar aplicaciones web?	Encuesta al cliente

Tabla 1. Variable independiente

Operacionalización de la variable independiente machine learning

Conceptualización	Categorías	Indicadores	Preguntas	Herramienta
Machine Learning (ML) (2021), indica que el aprendizaje automático tiene un amplio rango de algoritmos surgidos principalmente de campos como la ingeniería eléctrica, estadística, análisis financiero y muchos más. Los componentes fundamentales de todos estos algoritmos se basan profundamente en los principios de las matemáticas pura y la estadística. Los computadores cumplen un papel importante en la automatización de cálculos complejos. Esta automatización permite el procesamiento iterativo, el ajuste de parámetros y el reconocimiento de patrones, el efecto acumulativo de estos cálculos permite que los sistemas de aprendizaje automático aborden problemas complejos (p. 5).	Métodos de Machine Learning	Aprendizaje supervisado	¿Qué tan cómodo se siente al implementar un algoritmo de aprendizaje supervisado en la clasificación de mascotas?	Revisión de planificación del sprint con el gerente
		Aprendizaje no supervisado	¿Que que frecuencia utiliza aplicaciones que implementen algoritmos estadísticos y matemáticos para aprender de las interacciones del usuario?	Encuesta al cliente
Métodos de Machine Learning	Métodos de Machine Learning	Aprendizaje de refuerzo	¿Que que frecuencia utiliza los algoritmos que se adaptan a los cambios en los datos de entrenamiento?	Encuesta al cliente
		Árboles de decisión	¿Que que frecuencia utiliza los algoritmos que se adaptan a los cambios en los datos de entrenamiento?	Encuesta al cliente
Métodos de Machine Learning	Métodos de Machine Learning	Naive Bayes	¿Que que frecuencia recibe recomendaciones sobre qué mascotas adoptar?	Encuesta al cliente
		Regresión logística	¿Que tan cómodo se siente que es el proceso de encontrar las mascotas en adopción?	Encuesta al cliente
Métodos de Machine Learning	Métodos de Machine Learning	Random forests	¿Que tan cómodo se siente que es el proceso de encontrar las mascotas en adopción?	Encuesta al cliente
		Innovación K-means Clustering	¿Que tan cómodo se siente que es el proceso de encontrar las mascotas en adopción?	Encuesta al cliente
Métodos de Machine Learning	Métodos de Machine Learning	Recopilación de datos	¿Considera que es importante la implementación de algoritmos que permitan mejorar el tiempo de procesamiento de datos?	Revisión de planificación del sprint con el gerente
		Ingeniería de características	¿Considera que es importante la implementación de algoritmos que permitan mejorar el tiempo de procesamiento de datos?	Revisión de planificación del sprint con el gerente
Métodos de Machine Learning	Métodos de Machine Learning	División y selección de datos	¿Que tan cómodo se siente que es el proceso de encontrar las mascotas en adopción?	Encuesta al cliente
		Entrenamiento del modelo	¿Que tan cómodo se siente que es el proceso de encontrar las mascotas en adopción?	Encuesta al cliente
Métodos de Machine Learning	Métodos de Machine Learning	Evaluación del modelo	¿Que tan cómodo se siente que es el proceso de encontrar las mascotas en adopción?	Encuesta al cliente
		Despliegue y mantenimiento	¿Que tan cómodo se siente que es el proceso de encontrar las mascotas en adopción?	Encuesta al cliente

Tabla 2. Variable independiente

Operacionalización de la variable dependiente gestión de adopciones de mascotas

Conceptualización	Categorías	Indicadores	Preguntas	Herramienta
Gestión de adopciones de mascotas (2021), indica que el aprendizaje automático tiene un amplio rango de algoritmos surgidos principalmente de campos como la ingeniería eléctrica, estadística, análisis financiero y muchos más. Los componentes fundamentales de todos estos algoritmos se basan profundamente en los principios de las matemáticas pura y la estadística. Los computadores cumplen un papel importante en la automatización de cálculos complejos. Esta automatización permite el procesamiento iterativo, el ajuste de parámetros y el reconocimiento de patrones, el efecto acumulativo de estos cálculos permite que los sistemas de aprendizaje automático aborden problemas complejos (p. 5).	Selección	Selección	¿Que tan satisfecho está con las opciones disponibles y la facilidad para encontrar una mascota adecuada a sus necesidades? ¿Qué es su nivel de dificultad al completar la documentación necesaria para la adopción? ¿Que tan cómodo se siente el proceso de encontrar mascotas en adopción? ¿Que tan frecuente sienten que el proceso de adopción es poco claro o confuso?	Encuesta al cliente
		Proceso de adopción	¿Considera que la evaluación previa de los adoptantes es importante para evaluar su idoneidad? ¿Que tan familiarizado se encuentra con entrevistas para la adopción de una mascota?	Revisión de planificación del sprint con el gerente
Gestión de adopciones de mascotas	Gestión de adopciones de mascotas	Entrevista y Aprobación	¿Que tan frecuente ha sido notificado sobre entrevistas o pasos siguientes sin estar en solución? ¿Qué tan cómodo se siente al recibir información actual sobre el proceso de adopción?	Encuesta al cliente
		Registro médico	¿Que tan satisfecho se encuentra con la información que se proporciona de las mascotas? ¿Que tan cómodo se siente el proceso de encontrar mascotas en adopción?	Encuesta al cliente
Gestión de adopciones de mascotas	Gestión de adopciones de mascotas	Historial médico	¿Que tan cómodo se siente el proceso de encontrar mascotas en adopción?	Revisión de planificación del sprint con el gerente
		Examen físico	¿Que tan cómodo se siente el proceso de encontrar mascotas en adopción?	Revisión de planificación del sprint con el gerente
Gestión de adopciones de mascotas	Gestión de adopciones de mascotas	Políticas y procedimientos internos	¿Que tan cómodo se siente el proceso de encontrar mascotas en adopción?	Revisión de planificación del sprint con el gerente
		Organización del personal	¿Que tan cómodo se siente el proceso de encontrar mascotas en adopción?	Revisión de planificación del sprint con el gerente
Gestión de adopciones de mascotas	Gestión de adopciones de mascotas	Industria y bienestar animal	¿Que tan cómodo se siente el proceso de encontrar mascotas en adopción?	Revisión de planificación del sprint con el gerente
		Promoción y visibilidad	¿Que tan cómodo se siente el proceso de encontrar mascotas en adopción?	Revisión de planificación del sprint con el gerente
Gestión de adopciones de mascotas	Gestión de adopciones de mascotas	Dedicaciones	¿Que tan cómodo se siente el proceso de encontrar mascotas en adopción?	Encuesta al cliente
		Voluntariado	¿Que tan cómodo se siente el proceso de encontrar mascotas en adopción?	Encuesta al cliente

Tabla 3. Variable dependiente

Preguntas

Baremo:
Claridad: Se refiere si la pregunta está comprendida por los destinatarios.
Pertinencia: Se refiere si la pregunta corresponde con lo que se quiere indagar.
 Las preguntas en cuanto a su claridad y pertinencia tienen una escala valorativa Likert del 1 al 5 (donde 1 es el menor valor y 5 el mayor). Podrá añadir una formulación alternativa y observación, en caso que considere necesario.
 Marque con una cruz (X) el tramo del baremo que exprese mejor su juicio "Claridad" y "Pertinencia" sobre los ítems propuestos:

Reunión de planificación del sprint con el gerente del centro veterinario Mi Mascota

Mascota

Variable independiente: Aplicación web (Categoría: Bases de Datos)

1. ¿Qué método usas para registrar a las mascotas que están disponibles para adoptar?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

Variable independiente: Aplicación web (Categoría: Bases de Datos)

2. ¿Consideras que la información sobre las mascotas y los adoptantes se gestiona de manera organizada?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

Variable independiente: Aplicación web (Categoría: Bases de Datos)

3. ¿Qué tan difícil le resulta el proceso de registro para los adoptantes interesados?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Procesos de adopción)

4. ¿Consideras que la evaluación previa de los adoptantes es suficientemente rigurosa para registrar hogares adecuados?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

5

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Gestión organizacional)

5. ¿Con qué regularidad se proporciona capacitación sobre cómo manejar situaciones difíciles o sensibles con posibles adoptantes?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Gestión organizacional)

6. ¿Utilizan protocolos de seguimiento post-adopción para asegurar el bienestar de la mascota?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Registro médico veterinario)

7. ¿Qué métodos tienen cuando reciben una mascota para dar en adopción?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

Variable independiente: Aplicación Web (Categoría: Herramientas de desarrollo)

8. ¿Consideras útil la implementación de una aplicación web que permita gestionar adopciones de mascotas?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

Variable independiente: Aplicación web (Categoría: Bases de Datos)

9. ¿Qué tipo de información consideras que debe ser visible en la aplicación web?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

6

Variable independiente: Aplicación web (Categoría: Seguridad web)

10. ¿Le parece adecuado que en una aplicación web las aplicaciones sean aprobadas antes de publicarse?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

Variable independiente: Machine Learning (Categoría: Canalización de Machine Learning)

11. ¿Consideras que es importante la implementación de herramientas que permitan generar recomendaciones personalizadas según su perfil?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

Variable independiente: Machine Learning (Categoría: Métodos de Machine Learning)

12. ¿Está de acuerdo en que se implemente un sistema que aprenda de las preferencias de usuario en las aplicaciones web?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
		X							X		

7

Encuesta a clientes

Tema del Trabajo de Titulación de grado: Aplicación web con machine learning para la gestión de adopción de mascotas en el centro veterinario Mi Mascota en el cantón santo domingo.

Objetivo: Recolectar información para validar propuesta de intervención enfocada a la gestión de adopción de mascotas en el centro veterinario.

Instrucciones al público objetivo: La encuesta está dirigida a los clientes del centro veterinario Mi Mascota en el cantón Santo Domingo, la información recolectada permitirá conocer el manejo del proceso de gestión de adopción de mascotas desde el enfoque del cliente.

PREGUNTAS

Variable independiente: Aplicación web (Categoría: Herramienta de desarrollo)

1. ¿Con qué frecuencia hace uso de aplicaciones web? (PRE)

- Muy frecuentemente
- Frecuentemente
- Ocasionalmente
- Raramente
- Nunca

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5		
	X		

Variable independiente: Aplicación web (Categoría: Herramienta de desarrollo)

2. ¿Cuál es su nivel de dificultad navegar en aplicaciones web? (PRE)

- Muy fácil
- Fácil
- Normal
- Difícil
- Muy difícil

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5		
	X		

Variable independiente: Aplicación web (Categoría: Seguridad web)

3. ¿Qué tan seguro se siente al utilizar aplicaciones web? (PRE)

- Totalmente seguro
- Seguro
- Moderadamente seguro

8

PUCE

c) Moderadamente seguro
d) Poco seguro
e) Totalmente inseguro

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Variable independiente: Aplicación web (Categoría: Herramientas de desarrollo)

4. ¿Considera necesario el cómo está diseñada una aplicación web? (PRE)

a) Muy necesario
b) Necesario
c) Medianamente necesario
d) Poco necesario
e) Innecesario

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Variable dependiente: Gestión de adopción de mascotas (Categoría: Promoción y sostenibilidad en adopciones)

5. ¿Con qué frecuencia busca mascotas en adopción por internet? (PRE)

a) Muy frecuentemente
b) Frecuentemente
c) Ocasionalmente
d) Raramente
e) Nunca

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Variable independiente: Aplicación web (Categoría: Herramientas de desarrollo)

6. ¿Que tan familiarizado se encuentra con aplicaciones web para poder ver mascotas en adopción? (PRE Y POST)

a) Completamente familiarizado
b) Muy familiarizado
c) Algo familiarizado
d) Poco familiarizado
e) Nada familiarizado

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

9

PUCE

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Promoción y sostenibilidad en adopciones)

7. ¿Cree que es necesario mejorar la forma en que se difunden las mascotas disponibles para adopción? (PRE)

a) Muy necesario
b) Necesario
c) Medianamente necesario
d) Poco necesario
e) Innecesario

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Variable independiente: Aplicación web (Categoría: Base de datos)

8. ¿Con qué frecuencia siente que hay poca información sobre las mascotas en adopción? (PRE)

a) Muy frecuentemente
b) Frecuentemente
c) Ocasionalmente
d) Raramente
e) Nunca

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Variable independiente: Machine Learning (Categoría: Principales algoritmos de aprendizaje)

9. ¿Con qué frecuencia recibe recomendaciones sobre que mascota adoptar? (PRE Y POST)

a) Muy frecuentemente
b) Frecuentemente
c) Ocasionalmente
d) Raramente
e) Nunca

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

10

PUCE

Variable independiente: Machine Learning (Categoría: Principales algoritmos de aprendizaje)

10. ¿Con qué frecuencia siente que las mascotas que ve en adopción son de acuerdo a sus preferencias? (PRE Y POST)

a) Muy frecuentemente
b) Frecuentemente
c) Ocasionalmente
d) Raramente
e) Nunca

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>

Variable independiente: Machine Learning (Categoría: Canalización de Machine Learning)

11. ¿Cuál es su nivel de dificultad para encontrar una mascota en adopción de acuerdo a sus necesidades? (PRE Y POST)

a) Muy fácil
b) Fácil
c) Normal
d) Difícil
e) Muy difícil

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Procesos de adopción)

12. ¿Qué tan satisfecho está con las opciones disponibles y la facilidad para encontrar una mascota adecuada a sus necesidades? (PRE Y POST)

a) Totalmente satisfecho
b) Satisfecho
c) Moderadamente satisfecho
d) Poco satisfecho
e) Nada satisfecho

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

11

PUCE

Variable independiente: Machine learning (Categoría: Principales algoritmos de aprendizaje)

13. ¿Qué tan efectivo considera que es el proceso de encontrar una mascota en adopción? (PRE Y POST)

a) Completamente efectivo
b) Altsamente efectivo
c) Parcialmente efectivo
d) Poco efectivo
e) Nada efectivo

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Procesos de adopción)

14. ¿Cuál es su nivel de dificultad al completar la documentación necesaria para la adopción? (PRE Y POST)

a) Muy fácil
b) Fácil
c) Normal
d) Difícil
e) Muy difícil

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Procesos de adopción)

15. ¿Con qué frecuencia siente que el proceso de adopción es poco claro o confuso? (PRE Y POST)

a) Muy frecuentemente
b) Frecuentemente
c) Ocasionalmente
d) Raramente
e) Nunca

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5	-	-
<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

12

PUCE

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Procesos de adopción)

16. ¿Qué tan familiarizado se encuentra con entrevistas para la adopción de una mascota? (PRE)

- a) Completamente familiarizado
- b) Muy familiarizado
- c) Algo familiarizado
- d) Poco familiarizado
- e) Nada familiarizado

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Procesos de adopción)

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5		
	X		X

17. ¿Con qué frecuencia ha sido notificado sobre entrevistas o pasos siguientes tras enviar su solicitud? (PRE Y POST)

- a) Muy frecuentemente
- b) Frecuentemente
- c) Ocasionalmente
- d) Raramente
- e) Nunca

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5		
	X		X

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Procesos de adopción)

18. ¿Está usted de acuerdo en recibir información actual sobre el proceso de adopción? (PRE Y POST)

- a) Totalmente de acuerdo
- b) Bastante de acuerdo
- c) Neutral
- d) Poco de acuerdo
- e) Totalmente en desacuerdo

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5		
	X		X

13

PUCE

Variable dependiente: Gestión de adopciones de mascotas (Categoría: Registro médico veterinario)

19. ¿Qué tan satisfecho se encuentra con la información que se proporciona de las mascotas? (PRE Y POST)

- a) Totalmente satisfecho
- b) Satisfecho
- c) Moderadamente satisfecho
- d) Poco satisfecho
- e) Nada satisfecho

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5		
	X		X

Variable independiente: Aplicación web (Categoría: Seguridad web)

20. ¿Está de acuerdo en que se implementen sistemas de autenticación de datos? (PRE)

- a) Totalmente de acuerdo
- b) Bastante de acuerdo
- c) Neutral
- d) Poco de acuerdo
- e) Totalmente en desacuerdo

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5		
	X		X

Variable independiente: Machine learning (Categoría: Métodos de Machine learning)

21. ¿Con qué frecuencia utiliza aplicaciones que implementen inteligencia artificial o sistemas que aprendan de las interacciones del usuario? (PRE)

- a) Muy frecuentemente
- b) Frecuentemente
- c) Ocasionalmente
- d) Raramente
- e) Nunca

Relevancia	Claridad	Formulación alternativa:	Observación:
1 2 3 4 5	1 2 3 4 5		
	X		X

14

PUCE

Datos generales

Edad:

- a) 18-28
- b) 29-38
- c) 39-48
- d) 49-58
- e) 59 o más

Género:

- a) Femenino
- b) Masculino
- c) Prefiero no decirlo

¿Posee alguna discapacidad?

- a) Sí
- b) No

En caso de ser afirmativa la pregunta anterior, ¿Qué tipo de discapacidad posee?

- a) Visual
- b) Auditiva
- c) Física
- d) Intelectual
- e) Otra (indique): _____

¡GRACIAS POR SU COLABORACIÓN!

Una vez finalizada su validación, puede realizar comentarios, sugerencias o la aprobación, además, es pertinente que agregue sus datos personales.

Comentarios de validación:

Las preguntas realizadas en su mayoría cumplen con el propósito esperado, sin embargo hay breves detalles en un número mínimo de preguntas que bajo mi sugerencia requieren ser modificadas: dado que la terminología o en su defecto, la formulación de la pregunta no resulta ser un 100% concisa.


Datos informativos del experto

Nombres y Apellidos: BRITHANY NAOMI VELASCO REYES
Profesión y cargo: MEDICO VETERINARIA - MEDICO VETERINARIA DEL MUNICIPIO DEL CANTON PASAJE
Título universitario: MEDICO VETERINARIA
Email: brithany.veterinaria@gmail.com
Fecha y hora de validación: 07/07/2025 12:28pm

15

Datos informativos del experto

Nombres y Apellidos: Luis Javier Ulloa Meneses
Profesión y cargo: Profesor Tiempo Completo (Titular Agregado I)
Título universitario: Magíster en Big Data y Data Science
Email: ljuiloo@pucesd.edu.ec
Fecha y hora de validación: 15/07/2025 - 12H00


 Firma

¡GRACIAS POR SU COLABORACIÓN!

Una vez finalizada su validación, puede realizar comentarios, sugerencias o la aprobación, además, es pertinente que agregue sus datos personales.

Comentarios de validación:

Datos informativos del experto

Nombres y Apellidos: Jonathan Mario Moreno Rivera
Profesión y cargo: Desarrollador de Software
Título universitario: Mg. en Ingeniería de Software y Sistemas Informáticos
Email: jmarcomorenor@gmail.com
Fecha y hora de validación:

Anexo IV: Historias de Usuario e Historia técnica

Historia de Usuario	
Número: 1	Usuario: Administrador/Empleado/Cliente
Nombre historia: Inicio de sesión	
Prioridad en negocio: 100	Riesgo en desarrollo: ALTO
Puntos estimados: 8	Iteración asignada: 1
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción: Como Administrador/Empleado/Cliente Quiero ingresar mi usuario y contraseña Para acceder a las funcionalidades del sistema.	
Escenario de prueba: Dado el ingreso de usuario y contraseña Cuando presione el botón "Iniciar sesión" Entonces se muestra el menú principal de la aplicación. Dado el incorrecto o incompleto ingreso de usuario y contraseña Cuando presione el botón "Iniciar sesión" Entonces se muestra un mensaje de error.	

Historia de Usuario	
Número: 2	Usuario: Administrador/Empleado/Cliente
Nombre historia: Cierre de sesión	
Prioridad en negocio: 100	Riesgo en desarrollo: BAJO
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción: Como Administrador/Empleado/Cliente Quiero visualizar una opción que me permita cerrar sesión Para proteger la información de mi cuenta.	
Escenario de prueba: Dado la visualización de la opción para cerrar sesión Cuando presione "Cerrar sesión" Entonces se muestra el apartado de inicio de sesión de la aplicación.	

Historia de Usuario	
Número: 3	Usuario: Empleado/Cliente
Nombre historia: Registro de usuarios	
Prioridad en negocio: 100	Riesgo en desarrollo: ALTO
Puntos estimados: 8	Iteración asignada: 1
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción: Como Empleado/Cliente Quiero registrar mi información Para tener acceso a la interfaz de usuario.	
Escenario de prueba: Dado el registro de mi información Cuando presione el botón "Registrarse" Entonces se muestra un mensaje de "Usuario Registrado". Dado el registro de mi información Cuando el usuario o correo ya se encuentre en uso Entonces se muestra un mensaje de error. Dado el registro de mi información Cuando uno de los campos este vacío Entonces se muestra un mensaje de error.	

Historia de Usuario	
Número: 4	Usuario: Empleado/Cliente
Nombre historia: Verificación de cuenta	
Prioridad en negocio: 100	Riesgo en desarrollo: MEDIO
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción: Como Empleado/Cliente Quiero verificar mi cuenta Para tener acceso a las funcionalidades del sistema.	
Escenario de prueba: Dado la verificación de mi cuenta Cuando presione el botón para verificar cuenta Entonces se envía un enlace de verificación al correo. Dado la verificación de mi cuenta Cuando ingrese al enlace de verificación Entonces se mostrará un mensaje de confirmación. Dado la verificación de mi cuenta Cuando el enlace de verificación haya expirado Entonces se mostrará un mensaje de error.	

Historia de Usuario	
Número: 5	Usuario: Empleado/Cliente
Nombre historia: Recuperar contraseña	
Prioridad en negocio: 100	Riesgo en desarrollo: ALTO
Puntos estimados: 5	Iteración asignada: 1
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción: Como Empleado/Cliente Quiero ingresar mi correo electrónico Para recuperar mi contraseña	
Escenario de prueba: Dado el ingreso correcto de mi correo electrónico Cuando presione el botón "Enviar" Entonces se envía un correo con el enlace de recuperación. Dado el ingreso inválido de mi correo electrónico Cuando presione el botón "Enviar" Entonces se mostrará un mensaje de error. Dado el ingreso de mi correo electrónico Cuando ingrese al enlace de recuperación Entonces se redirigirá a la página para ingresar una nueva contraseña. Dado el ingreso de mi correo electrónico Cuando ingrese una nueva contraseña Entonces se muestra un mensaje de confirmación y redirige la página inicio de sesión. Dado el ingreso de mi correo electrónico Cuando el enlace de recuperación haya expirado Entonces se mostrará un mensaje de error.	

Historia de Usuario	
Número: 6	Usuario: Administrador/Empleado/Cliente
Nombre historia: Información de perfil	
Prioridad en negocio: 95	Riesgo en desarrollo: MEDIO
Puntos estimados: 5	Iteración asignada: 1
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción: Como Administrador/Empleado/Cliente Quiero visualizar la información de mi perfil Para verificar que la información sea correcta.	
Escenario de prueba: Dado la visualización de la información de mi perfil Cuando presione en "ver perfil" Entonces se redireccionará al perfil de usuario.	

Historia de Usuario	
Número: 7	Usuario: Administrador
Nombre historia: Roles de usuario	
Prioridad en negocio: 95	Riesgo en desarrollo: MEDIO
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador Quiero asignar roles de usuario Para habilitar funciones adicionales.	
Escenario de prueba:	
Dado la asignación de roles de usuario Cuando modifique el rol de un usuario Entonces el rol del usuario cambiará.	

Historia de Usuario	
Número: 8	Usuario: Empleado/Cliente
Nombre historia: Cambio de contraseña	
Prioridad en negocio: 90	Riesgo en desarrollo: MEDIO
Puntos estimados: 5	Iteración asignada: 1
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Empleado/Cliente Quiero ingresar una nueva contraseña Para cambiar la anterior y mantener la seguridad en mi cuenta.	
Escenario de prueba:	
Dado el ingreso de una nueva contraseña Cuando ingrese de forma correcta mi contraseña actual Entonces se realizará el cambio de contraseña.	
Dado el ingreso de una nueva contraseña Cuando ingrese de forma incorrecta mi contraseña actual Entonces se mostrará un mensaje de error.	

Historia de Usuario	
Número: 9	Usuario: Administrador/Empleado/Cliente
Nombre historia: Edición de perfil	
Prioridad en negocio: 90	Riesgo en desarrollo: ALTO
Puntos estimados: 5	Iteración asignada: 2
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Cliente Quiero editar la información del perfil Para actualizar mis datos personales.	
Escenario de prueba:	
Dado la edición de la información del perfil Cuando presione el botón "Editar información" Entonces se activarán los campos para editar.	
Dado la edición de la información del perfil Cuando edite la información y presione el botón "Guardar" Entonces se actualizará la información.	

Historia de Usuario	
Número: 10	Usuario: Empleado/Cliente
Nombre historia: Eliminar cuenta	
Prioridad en negocio: 90	Riesgo en desarrollo: ALTO
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Empleado/Cliente Quiero eliminar mi cuenta Para no seguir utilizando los servicios.	
Escenario de prueba:	
Dado la eliminación de mi cuenta Cuando presione "Eliminar cuenta" Entonces se mostrará un mensaje de advertencia.	
Dado la eliminación de mi cuenta Cuando presione el botón "Aceptar" Entonces se mostrará un mensaje de confirmación.	

Historia de Usuario	
Número: 11	Usuario: Administrador/Empleado/Cliente
Nombre historia: Agregar mascota	
Prioridad en negocio: 90	Riesgo en desarrollo: MEDIO
Puntos estimados: 8	Iteración asignada: 2
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Cliente Quiero ingresar la información de una mascota Para poder darla en adopción.	
Escenario de prueba:	
Dado el ingreso de la información de la mascota Cuando presione el botón "Agregar" Entonces se mostrará un mensaje de confirmación.	

Historia de Usuario	
Número: 12	Usuario: Administrador/Empleado/Cliente
Nombre historia: Eliminar mascota	
Prioridad en negocio: 90	Riesgo en desarrollo: ALTO
Puntos estimados: 5	Iteración asignada: 2
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Cliente Quiero eliminar la información de una mascota Para poder borrar un ingreso incorrecto de una mascota.	
Escenario de prueba:	
Dado la eliminación de la información de una mascota Cuando se presione el botón "Eliminar" Entonces se muestra una alerta de confirmación para eliminar la mascota.	

Historia de Usuario	
Número: 13	Usuario: Administrador/Empleado/Cliente
Nombre historia: Editar mascota	
Prioridad en negocio: 85	Riesgo en desarrollo: ALTO
Puntos estimados: 8	Iteración asignada: 2
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Cliente Quiero editar la información de una mascota Para corregir la información que ha sido ingresada incorrectamente.	
Escenario de prueba:	
Dado la edición de la información de una mascota Cuando se presione el botón "Guardar" Entonces se mostrará un mensaje de confirmación.	

Historia de Usuario	
Número: 14	Usuario: Administrador/Empleado
Nombre historia: Gestión de información de mascotas	
Prioridad en negocio: 85	Riesgo en desarrollo: ALTO
Puntos estimados: 5	Iteración asignada: 2
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado Quiero seleccionar la información de la mascota Para Aceptar/Rechazar la información.	
Escenario de prueba:	
Dado la selección de la información de la mascota Cuando presione el botón "Aceptar" Entonces se mostrará un mensaje de confirmación.	
Dado la selección de la información de la mascota Cuando presione el botón "Rechazar" Entonces se mostrará un mensaje de confirmación.	

Historia de Usuario	
Número: 15	Usuario: Administrador/Empleado/Ciente
Nombre historia: Mis publicaciones	
Prioridad en negocio: 85	Riesgo en desarrollo: MEDIO
Puntos estimados: 8	Iteración asignada: 2
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Ciente Quiero visualizar mis publicaciones Para llevar un control de las mascotas que he agregado.	
Escenario de prueba:	
Dado la visualización de mis publicaciones Cuando presione "Mis publicaciones" Entonces se mostrarán todas la mascotas que he agregado.	

Historia de Usuario	
Número: 16	Usuario: Administrador/Empleado/Ciente
Nombre historia: Adopciones realizadas	
Prioridad en negocio: 80	Riesgo en desarrollo: BAJO
Puntos estimados: 5	Iteración asignada: 2
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Ciente Quiero visualizar las adopciones realizadas Para llevar un control de las mascotas que he adoptado.	
Escenario de prueba:	
Dado la visualización de las adopciones realizadas Cuando presione "Adopciones realizadas" Entonces se mostrarán todas la mascotas que he adoptado.	

Historia de Usuario	
Número: 17	Usuario: Administrador/Empleado/Ciente
Nombre historia: Visualización de Mascotas	
Prioridad en negocio: 80	Riesgo en desarrollo: MEDIO
Puntos estimados: 8	Iteración asignada: 2
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Ciente Quiero visualizar las mascotas Para conocer cuáles están disponibles para adopción.	
Escenario de prueba:	
Dado la visualización de las mascotas Cuando presione el botón "Gatos" Entonces se mostrará el apartado de las publicaciones de los gatos disponibles para adoptar.	
Dado la visualización de las mascotas Cuando presione el botón "Perros" Entonces se mostrará el apartado de las publicaciones de los perros disponibles para adoptar.	

Historia de Usuario	
Número: 18	Usuario: Administrador/Empleado/Ciente
Nombre historia: Ficha de información de mascotas	
Prioridad en negocio: 80	Riesgo en desarrollo: MEDIO
Puntos estimados: 5	Iteración asignada: 3
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Ciente Quiero visualizar las fichas informativas de las mascotas Para poder ver las características respectivas.	
Escenario de prueba:	
Dado la visualización de las fichas informativas de las mascotas Cuando presione el botón "Ver más" Entonces se mostrará todas las características disponibles de la mascota.	

Historia de Usuario	
Número: 19	Usuario: Administrador/Empleado/Ciente
Nombre historia: Favoritos	
Prioridad en negocio: 80	Riesgo en desarrollo: ALTO
Puntos estimados: 5	Iteración asignada: 3
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Ciente Quiero marcar una mascota como favorito Para poder guardarla como preferida.	
Escenario de prueba:	
Dado la marcación de una mascota como favorito Cuando marque el icono de estrella Entonces se guardará como favoritos.	
Dado la marcación de una mascota como favorito Cuando desmarque el icono de estrella Entonces eliminara de favoritos.	

Historia de Usuario	
Número: 20	Usuario: Administrador/Empleado/Ciente
Nombre historia: Solicitud de adopción	
Prioridad en negocio: 75	Riesgo en desarrollo: MEDIO
Puntos estimados: 5	Iteración asignada: 3
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Ciente Quiero solicitar una adopción Para adoptar una mascota.	
Escenario de prueba:	
Dado la solicitud de adopción Cuando presione el botón "Adoptar" Entonces mostrara un formulario de solicitud.	
Dado la solicitud de adopción Cuando presione el botón "Enviar" Entonces mostrara un mensaje de confirmación.	
Dado la solicitud de adopción Cuando uno de los campos este vacío y presione "Enviar" Entonces se mostrará un mensaje de error.	

Historia de Usuario	
Número: 21	Usuario: Administrador/Empleado/Ciente
Nombre historia: Visualización de solicitudes enviadas	
Prioridad en negocio: 75	Riesgo en desarrollo: BAJO
Puntos estimados: 5	Iteración asignada: 3
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Ciente Quiero visualizar las solicitudes de adopción enviadas Para conocer el estado actual de la solicitud.	
Escenario de prueba:	
Dado la visualización de las solicitudes de adopción enviadas Cuando presione "Solicitudes enviadas" Entonces se desplegará una lista de las solicitudes que han sido enviadas.	

Historia de Usuario	
Número: 22	Usuario: Administrador/Empleado/Ciente
Nombre historia: Visualización de solicitudes recibidas	
Prioridad en negocio: 75	Riesgo en desarrollo: BAJO
Puntos estimados: 5	Iteración asignada: 3
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Ciente Quiero visualizar las solicitudes de adopción recibidas Para llevar un control de las solicitudes receptadas.	
Escenario de prueba:	
Dado la visualización de las solicitudes de adopción recibidas Cuando presione "Solicitudes recibidas" Entonces se desplegará una lista de las solicitudes que han sido enviadas por cada mascota.	

Historia de Usuario	
Número: 23	Usuario: Administrador/Empleado/Cliente
Nombre historia: Gestión de solicitudes de adopción	
Prioridad en negocio: 75	Riesgo en desarrollo: ALTO
Puntos estimados: 5	Iteración asignada: 3
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Empleado/Cliente Quiero gestionar las solicitudes de adopción Para poder Aceptar/Rechazar la solicitud.	
Escenario de prueba:	
Dado la gestión de las solicitudes de adopción Cuando seleccione la opción "Aceptar" Entonces cambiara el estado de la solicitud a Aceptada.	
Dado la gestión de las solicitudes de adopción Cuando seleccione la opción "Rechazar" Entonces cambiara el estado de la solicitud a Rechazada.	

Historia de Usuario	
Número: 24	Usuario: Administrador/Empleado/Cliente
Nombre historia: Notificación de solicitud	
Prioridad en negocio: 70	Riesgo en desarrollo: ALTO
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Cliente Quiero visualizar notificaciones sobre mis solicitudes de adopción Para estar informado del cambio realizado en su estado.	
Escenario de prueba:	
Dado la visualización de las notificaciones de mis solicitudes de adopción Cuando presione el icono de notificación Entonces se mostrarán las notificaciones recibidas.	

Historia de Usuario	
Número: 25	Usuario: Administrador/Empleado/Cliente
Nombre historia: Recomendación de mascotas personalizada	
Prioridad en negocio: 70	Riesgo en desarrollo: ALTO
Puntos estimados: 8	Iteración asignada: 3
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado/Cliente Quiero obtener una recomendación de mascotas personalizada Para encontrar mascotas de acuerdo a mis intereses.	
Escenario de prueba:	
Dado la obtención de la recomendación de mascotas personalizada Cuando acceda al apartado de "Match" Entonces se mostrarán las recomendaciones de mascotas.	

Historia de Usuario	
Número: 26	Usuario: Cliente
Nombre historia: Enviar valoración	
Prioridad en negocio: 70	Riesgo en desarrollo: ALTO
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Cliente Quiero enviar una valoración de la página Para expresar mi experiencia en la aplicación.	
Escenario de prueba:	
Dado el envío de la valoración de la página Cuando seleccione la cantidad de estrellas que desea marcar, escriba un comentario y presione "Enviar" Entonces mostrara un mensaje de confirmación.	

Historia de Usuario	
Número: 27	Usuario: Administrador/Empleado
Nombre historia: Gestión de valoraciones	
Prioridad en negocio: 70	Riesgo en desarrollo: MEDIO
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Jaime Ortiz y Nathaly Zamora	
Descripción:	
Como Administrador/Empleado Quiero gestionar las valoraciones de la página Para tener un control de la información publicada.	
Escenario de prueba:	
Dado a la gestión de valoraciones de la página Cuando seleccione la valoración y de click en el icono de eliminar Entonces mostrara un mensaje de confirmación.	

Historia de técnica	
Título: Seguridad y Privacidad de Aplicación	Estimación: 8
Historia relacionada: Seguridad y privacidad de aplicación	
Descripción: Se enfoca en implementar las medidas de seguridad en las tres capas de la aplicación MiMascota (frontend, backend y base de datos). Se utiliza autenticación con JWT para validar la identidad del usuario en cada petición protegida, encriptación de contraseñas con bcrypt para proteger las credenciales almacenadas, y un sistema de autorización por roles y permisos para controlar el acceso a operaciones sensibles. En el frontend se protegen las rutas con guardias de navegación de Vue Router, la conexión a la base de datos se asegura con SSL en producción, y se restringe el registro a dominios de correo confiables mediante una lista blanca con verificación por correo electrónico.	
Criterio de aceptación: <ul style="list-style-type: none"> • Generar y verificar tokens JWT en cada petición protegida del backend • Hashear todas las contraseñas con bcrypt antes de almacenarlas en la base de datos • Controlar el acceso a operaciones sensibles según el rol y permisos del usuario • Redirigir a usuarios no autenticados al login desde cualquier ruta protegida del frontend • Establecer conexión SSL entre el servidor y la base de datos en producción • Rechazar registros con dominios de correo no autorizados 	
Beneficio de justificación: Garantizar que la aplicación proteja la información personal de los usuarios, sus credenciales y los datos de las mascotas, cumpliendo con buenas prácticas de seguridad en cada capa del sistema para prevenir accesos no autorizados, interceptación de datos y registros fraudulentos.	
Impacto en el producto o proyecto: Sin estas medidas, cualquier persona podría acceder a datos privados de usuarios y mascotas sin autenticación, las contraseñas quedarían expuestas ante una brecha de seguridad, usuarios sin privilegios podrían ejecutar operaciones administrativas, y la plataforma sería vulnerable a cuentas falsas y ataques de intermediario.	
Dependencias: <ul style="list-style-type: none"> • Librerías jsonwebtoken, bcrypt, uuid y dotenv (npm) • Variable de entorno JWT_SECRET y DATABASE_URL • Base de datos PostgreSQL con tablas roles, permisos y rol_permiso • Vue Router y Pinia en el frontend • API de Brevo para envío de correos de verificación 	
Tareas o subtareas: <ul style="list-style-type: none"> • Autenticación con JWT (JSON Web Tokens) • Encriptación de Contraseñas • Middleware de Autorización por Roles y Permisos • Protección de Rutas por medio de Vue-Router • Conexión Segura a Base de Datos (SSL) • Validación de Dominios de Correo 	

Anexo V: Pruebas de Aceptación

Pontificia Universidad Católica del Ecuador

Sede Santo Domingo

Escuela de Sistemas

CASO DE PRUEBA 01
Fecha: 16/11/2025

Nombre caso de prueba: Inicio de sesión **Sprint:** 1

Módulo/sección a evaluar: Inicio de sesión **Historia de usuario asociada:** 1

Técnica de prueba: Caja Negra ☒ Caja Blanca ☐ **Tipo:** Prueba de Aceptación

Descripción:

COMO Administrador/Empleado/Cliente QUIERO ingresar mi usuario y contraseña PARA acceder a las funcionalidades del sistema.

Escenario de prueba:

DADO el ingreso de usuario y contraseña CUANDO oprima el botón "Iniciar sesión" ENTONCES se muestra el menú principal de la aplicación.

DADO el incorrecto o incompleto ingreso de usuario y contraseña CUANDO oprima el botón "Iniciar sesión" ENTONCES se muestra un mensaje de error.

Pre-condiciones

- Tener acceso a internet.
- Contar con un usuario registrado.

Pasos y condiciones de ejecución

- Navegar a la pantalla de inicio de sesión de la aplicación.
- Ingresar un correo y contraseña.
- Pulsar el botón "Iniciar sesión".

Resultado esperado

- El usuario es redirigido al menú principal.

Estado de prueba	Éxito	Falló
Errores asociados:	Si	No

Lic. Zolte Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador

Sede Santo Domingo

Escuela de Sistemas

CASO DE PRUEBA 02
Fecha: 16/11/2025

Nombre caso de prueba: Cierre de sesión **Sprint:** 1

Módulo/sección a evaluar: Cierre de sesión **Historia de usuario asociada:** 2

Técnica de prueba: Caja Negra ☒ Caja Blanca ☐ **Tipo:** Prueba de Aceptación

Descripción:

Como Administrador/Empleado/Cliente QUIERO visualizar una opción que me permita cerrar sesión Para proteger la información de mi cuenta.

Escenario de prueba:

Dado la visualización de la opción para cerrar sesión Cuando presione "Cerrar sesión" ENTONCES se muestra el apartado de inicio de sesión de la aplicación.

Pre-condiciones

- Tener acceso a internet.
- Contar con un usuario registrado.
- Haber iniciado sesión.

Pasos y condiciones de ejecución

- Dar click en el menú desplegable en la barra de navegación.
- Presionar "Cerrar sesión"

Resultado esperado

- Sesión cerrada en la aplicación.

Estado de prueba	Éxito	Falló
Errores asociados:	Si	No

Sra. Zolte Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santa Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 03 Fecha: 16/11/2025

Nombre caso de prueba: Registro de usuarios Sprint: 1
 Módulo/sección a evaluar: Registro de usuarios Historia de usuario asociada: 3
 Técnica de prueba: Caja Negra & Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Empleado/Cliente Quiero registrar mi información Para tener acceso a la interfaz de usuario.

Escenario de prueba:
 Dado el registro de mi información Cuando presione el botón "Registrarse" Entonces se muestra un mensaje de confirmación.

Dado el registro de mi información Cuando el usuario o correo ya se encuentre en uso Entonces se muestra un mensaje de error.

Dado el registro de mi información Cuando uno de los campos este vacío Entonces se muestra un mensaje de error.

Pre-condiciones

- Tener acceso a internet.

Pasos y condiciones de ejecución

- Ir al apartado de inicio de sesión.
- Presionar sobre "Registrarse"
- Ingresar los datos requeridos.
- Presionar el botón "Registrarse"

Resultado esperado

- Visualización de un mensaje de confirmación y verificación de cuenta.

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santa Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 04 Fecha: 16/11/2025

Nombre caso de prueba: Verificación de cuenta Sprint: 1
 Módulo/sección a evaluar: Verificación de cuenta Historia de usuario asociada: 4
 Técnica de prueba: Caja Negra & Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Empleado/Cliente Quiero verificar mi cuenta Para tener acceso a las funcionalidades del sistema.

Escenario de prueba:
 Dado la verificación de mi cuenta Cuando haya completado mi registro Entonces se envía un enlace de verificación al correo.

Dado la verificación de mi cuenta Cuando ingrese al enlace de verificación Entonces se mostrará un mensaje de confirmación.

Dado la verificación de mi cuenta Cuando el enlace de verificación haya expirado Entonces se mostrará un mensaje de error.

Pre-condiciones

- Tener acceso a internet.
- Contar con un usuario registrado.

Pasos y condiciones de ejecución


- Acceder a la bandeja de entrada del correo asociado a la cuenta.
- Hacer clic en el enlace de verificación.

Resultado esperado

- Verificación de la cuenta.
- Visualización de un mensaje de confirmación.

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santa Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 05 Fecha: 16/11/2025

Nombre caso de prueba: Recuperar contraseña Sprint: 1
 Módulo/sección a evaluar: Recuperar contraseña Historia de usuario asociada: 5
 Técnica de prueba: Caja Negra & Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Empleado/Cliente Quiero ingresar mi correo electrónico Para recuperar mi contraseña.

Escenario de prueba:
 Dado el ingreso correcto de mi correo electrónico Cuando presione el botón "Enviar" Entonces se envía un correo con el enlace de recuperación.

Dado el ingreso de mi correo electrónico Cuando ingrese el enlace de recuperación Entonces se redirigirá a la página para ingresar una nueva contraseña.

Dado el ingreso de mi correo electrónico Cuando ingrese una nueva contraseña y confirme la contraseña Entonces se muestra un mensaje de confirmación y redirige la página inicio de sesión.

Dado el ingreso de mi correo electrónico Cuando la nueva contraseña y la confirmación de contraseña no sean iguales Entonces se muestra un mensaje de error.

Dado el ingreso de mi correo electrónico Cuando el enlace de recuperación haya expirado Entonces se mostrará un mensaje de error.

Pre-condiciones

- Tener acceso a internet.
- Contar con un usuario registrado.

Pasos y condiciones de ejecución


- Acceder a la bandeja de entrada del correo ingresado.
- Hacer clic en el enlace de verificación.

Resultado esperado

- Actualización de la contraseña.

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santa Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 06 Fecha: 16/11/2025

Nombre caso de prueba: Información de perfil Sprint: 1
 Módulo/sección a evaluar: Información de perfil Historia de usuario asociada: 6
 Técnica de prueba: Caja Negra & Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Administrador/Empleado/Cliente Quiero visualizar la información de mi perfil Para verificar que la información sea correcta.

Escenario de prueba:
 Dado la visualización de la información de mi perfil Cuando presione en "ver perfil" Entonces se redireccionará al perfil de usuario.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.

Pasos y condiciones de ejecución


- Dar clic en el menú desplegable en la barra de navegación.
- Presionar "Ver perfil"

Resultado esperado

- Visualización del perfil del usuario.

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santa Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 07 Fecha: 16/11/2025

Nombre caso de prueba: Roles de usuario Sprint: 1
 Módulo/sección a evaluar: Roles de usuario Historia de usuario asociada: 7
 Técnica de prueba: Caja Negra & Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Administrador Quiero asignar roles de usuario Para habilitar funciones adicionales.

Escenario de prueba:
 Dado la asignación de roles de usuario Cuando modifique el rol de un usuario Entonces el rol del usuario cambiará.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Tener rol de administrador.

Pasos y condiciones de ejecución

- Dar clic en el menú desplegable en la barra de navegación.
- Presionar "Lista de usuarios"
- Hacer clic en el ícono de lápiz y cambiar el rol de un usuario.

Resultado esperado

- Modificación del rol de un usuario.

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santa Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 08 Fecha: 16/11/2025

Nombre caso de prueba: Cambio de contraseña Sprint: 1
 Módulo/sección a evaluar: Cambio de contraseña Historia de usuario asociada: 8
 Técnica de prueba: Caja Negra & Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Empleado/Cliente Quiero ingresar una nueva contraseña Para cambiar la anterior y mantener la seguridad en mi cuenta.

Escenario de prueba:
 Dado el ingreso de una nueva contraseña Cuando ingrese de forma correcta mi contraseña actual Entonces se realizará el cambio de contraseña.

Dado el ingreso de una nueva contraseña Cuando ingrese de forma incorrecta mi contraseña actual Entonces se mostrará un mensaje de error.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estor en el perfil de usuario.

Pasos y condiciones de ejecución

- Hacer clic en el botón "Cambiar contraseña"
- Ingresar contraseña actual, nueva contraseña y confirmar contraseña
- Hacer clic en el botón "Guardar"

Resultado esperado

- Modificación de la contraseña.

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 09 Fecha: 16/11/2025

Nombre caso de prueba: Edición de perfil Sprint: 2
 Módulo/sección a evaluar: Edición de perfil Historia de usuario asociada: 9
 Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Administrador/Empleado/Cliente Quiero editar la información del perfil Para actualizar mis datos personales.

Escenario de prueba:
 Dado la edición de la información del perfil Cuando presione el botón "Editar información" Entonces se activarán los campos para editar.
 Dado la edición de la información del perfil Cuando edite la información y presione el botón "Guardar" Entonces se actualizará la información.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en el perfil de usuario.

Pasos y condiciones de ejecución


- Hacer click en el botón "Editar información".
- Editar los campos deseados.
- Hacer click en el botón "Guardar"

Resultado esperado

- Actualización de la información del usuario.

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Sra. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 10 Fecha: 16/11/2025

Nombre caso de prueba: Eliminar cuenta Sprint: 2
 Módulo/sección a evaluar: Eliminar cuenta Historia de usuario asociada: 10
 Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Empleado/Cliente Quiero eliminar mi cuenta Para no seguir utilizando los servicios.

Escenario de prueba:
 Dado la eliminación de mi cuenta Cuando presione "Eliminar cuenta" Entonces se mostrará un mensaje de advertencia.
 Dado la eliminación de mi cuenta Cuando presione el botón "Aceptar" Entonces se mostrará un mensaje de confirmación.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.

Pasos y condiciones de ejecución


- Dar click en el menú desplegable en la barra de navegación.
- Presionar "Eliminar cuenta"
- Hacer click en el botón "SI"

Resultado esperado

- Eliminación de la cuenta.

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Sra. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 11 Fecha: 16/11/2025

Nombre caso de prueba: Agregar mascota Sprint: 2
 Módulo/sección a evaluar: Agregar mascota Historia de usuario asociada: 11
 Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Administrador/Empleado/Cliente Quiero ingresar la información de una mascota Para poder cuida en adopción.

Escenario de prueba:
 Dado el ingreso de la información de la mascota Cuando presione el botón "Agregar" Entonces se mostrará un mensaje de confirmación.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en el perfil de usuario.

Pasos y condiciones de ejecución

- Hacer click en el botón "Agregar Mascotas".
- Llenar los campos necesarios.
- Hacer click en el botón "Publicar"

Resultado esperado

- Creación de la información de la mascota.

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Sra. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 12 Fecha: 16/11/2025

Nombre caso de prueba: Eliminar mascota Sprint: 2
 Módulo/sección a evaluar: Eliminar mascota Historia de usuario asociada: 12
 Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Administrador/Empleado/Cliente Quiero eliminar la información de una mascota Para poder borrar un ingreso incorrecto de una mascota.

Escenario de prueba:
 Dado la eliminación de la información de una mascota Cuando se presione el botón "Eliminar" Entonces se muestra una alerta de confirmación para eliminar la mascota.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en el perfil de usuario.

Pasos y condiciones de ejecución

- Presionar "Eliminar Mascotas".
- Flagar la mascota.
- Hacer click en el botón "Eliminar".
- Hacer click en el botón "SI".

Resultado esperado

- Eliminación de la información de la mascota.

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Sra. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 13 Fecha: 16/11/2025

Nombre caso de prueba: Editar mascota Sprint: 2
 Módulo/sección a evaluar: Editar mascota Historia de usuario asociada: 13
 Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Administrador/Empleado/Cliente Quiero editar la información de una mascota Para corregir la información que ha sido ingresada incorrectamente.

Escenario de prueba:
 Dado la edición de la información de una mascota Cuando se presione el botón "Guardar" Entonces se mostrará un mensaje de confirmación.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en el perfil de usuario.

Pasos y condiciones de ejecución


- Presionar "Editar Mascotas".
- Elegir la mascota.
- Hacer click en el botón "Editar".
- Modificar los datos deseados.
- Hacer click en el botón "Guardar cambios"

Resultado esperado

- Edición de la información de la mascota.

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Sra. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 14 Fecha: 16/11/2025

Nombre caso de prueba: Gestión de información de mascotas Sprint: 2
 Módulo/sección a evaluar: Gestión de información de mascotas Historia de usuario asociada: 14
 Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
 Como Administrador/Empleado Quiero seleccionar la información de la mascota Para Aceptar/Rechazar la información.

Escenario de prueba:
 Dado la selección de la información de la mascota Cuando presione el botón "Aceptar" Entonces se mostrará un mensaje de confirmación.
 Dado la selección de la información de la mascota Cuando presione el botón "Rechazar" Entonces se mostrará un mensaje de confirmación.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en el perfil de usuario.

Pasos y condiciones de ejecución


- Dar click en el menú desplegable en la barra de navegación.
- Presionar "Gestión de mascotas".
- Seleccionar la información de la mascota a gestionar.
- Dar click en el botón "Aceptar/Rechazar"

Resultado esperado

- Aceptación/Rechazo de la información de la mascota

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:


Sra. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 15 Fecha: 16/11/2025

Nombre caso de prueba: Mis publicaciones Sprint: 2
Módulo/sección a evaluar: Mis publicaciones Historia de usuario asociada: 15
Técnica de prueba: Caja Negra Caja Blanca Tipo: Prueba de Aceptación

Descripción:
Como Administrador/Empleado/Cliente Quiero visualizar mis publicaciones Para llevar un control de las mascotas que he agregado.

Escenario de prueba:
Dado la visualización de mis publicaciones Cuando presione "Mis publicaciones" Entonces se mostrarán todas las mascotas que he agregado.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en el perfil de usuario.

Pasos y condiciones de ejecución


- Presionar "Mis Publicaciones".

Resultado esperado

- Visualización de las publicaciones del usuario.

Estado de prueba	Éxito	Falló
	SI	No

Errores asociados:


Sra. Zolia Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 16 Fecha: 16/11/2025

Nombre caso de prueba: Adopciones realizadas Sprint: 2
Módulo/sección a evaluar: Adopciones realizadas Historia de usuario asociada: 16
Técnica de prueba: Caja Negra Caja Blanca Tipo: Prueba de Aceptación

Descripción:
Como Administrador/Empleado/Cliente Quiero visualizar las adopciones realizadas Para llevar un control de las mascotas que he adoptado.

Escenario de prueba:
Dado la visualización de las adopciones realizadas Cuando presione "Adopciones realizadas" Entonces se mostrarán todas las mascotas que he adoptado.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en el perfil de usuario.

Pasos y condiciones de ejecución


- Presionar "Adopciones Realizadas".

Resultado esperado

- Visualización de las adopciones realizadas del usuario.

Estado de prueba	Éxito	Falló
	SI	No

Errores asociados:


Sra. Zolia Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 17 Fecha: 16/11/2025

Nombre caso de prueba: Adopciones realizadas Sprint: 2
Módulo/sección a evaluar: Adopciones realizadas Historia de usuario asociada: 17
Técnica de prueba: Caja Negra Caja Blanca Tipo: Prueba de Aceptación

Descripción:
Como Administrador/Empleado/Cliente Quiero visualizar las mascotas Para conocer cuáles están disponibles para adopción.

Escenario de prueba:
Dado la visualización de las mascotas Cuando presione el botón "Gatos" Entonces se mostrará el apartado de las publicaciones de los gatos disponibles para adoptar.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.

Pasos y condiciones de ejecución


- Presionar "Gatos" en la barra de navegación.
- Presionar "Perros" en la barra de navegación.

Resultado esperado

- Visualización del apartado "Gatos" o "Perros".

Estado de prueba	Éxito	Falló
	SI	No

Errores asociados:


Sra. Zolia Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 18 Fecha: 16/11/2025

Nombre caso de prueba: Ficha de información de mascotas Sprint: 3
Módulo/sección a evaluar: Ficha de información de mascotas Historia de usuario asociada: 18
Técnica de prueba: Caja Negra Caja Blanca Tipo: Prueba de Aceptación

Descripción:
Como Administrador/Empleado/Cliente Quiero visualizar las fichas informativas de las mascotas Para poder ver las características respectivas.

Escenario de prueba:
Dado la visualización de las fichas informativas de las mascotas Cuando presione el botón "Ver más" Entonces se mostrará todas las características disponibles de la mascota.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.

Pasos y condiciones de ejecución


- Elegir una mascota.
- Dar click en el botón "Ver más"

Resultado esperado

- Visualización de las fichas informativas de la mascota.

Estado de prueba	Éxito	Falló
	SI	No

Errores asociados:


Sra. Zolia Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 19 Fecha: 16/11/2025

Nombre caso de prueba: Favoritos Sprint: 3
Módulo/sección a evaluar: Favoritos Historia de usuario asociada: 19
Técnica de prueba: Caja Negra Caja Blanca Tipo: Prueba de Aceptación

Descripción:
Como Administrador/Empleado/Cliente Quiero marcar una mascota como favorito Para poder guardarla como preferida.

Escenario de prueba:
Dado la marcación de una mascota como favorito Cuando marque el icono de estrella Entonces se guardará como favoritos.
Dado la marcación de una mascota como favorito Cuando desmarque el icono de estrella Entonces eliminará de favoritos.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en la ficha de una mascota.

Pasos y condiciones de ejecución


- Marcar/Desmarcar el icono de estrella.

Resultado esperado

- La mascota se agregará/elimina de favoritos.

Estado de prueba	Éxito	Falló
	SI	No

Errores asociados:


Sra. Zolia Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 20 Fecha: 16/11/2025

Nombre caso de prueba: Solicitud de adopción Sprint: 3
Módulo/sección a evaluar: Solicitud de adopción Historia de usuario asociada: 20
Técnica de prueba: Caja Negra Caja Blanca Tipo: Prueba de Aceptación

Descripción:
Como Administrador/Empleado/Cliente Quiero solicitar una adopción Para adoptar una mascota.

Escenario de prueba:
Dado la solicitud de adopción Cuando presione el botón "Adoptar" Entonces mostrará un formulario de solicitud.
Dado la solicitud de adopción Cuando presione el botón "Enviar" Entonces mostrará un mensaje de confirmación.
Dado la solicitud de adopción Cuando uno de los campos este vacío y presione "Enviar" Entonces se mostrará un mensaje de error.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en la ficha de una mascota.

Pasos y condiciones de ejecución

- Dar click en el botón "Adoptar"
- Llenar la solicitud de adopción.
- Dar click en el botón "Enviar".

Resultado esperado

- Envío de la solicitud de adopción.

Estado de prueba	Éxito	Falló
	SI	No

Errores asociados:


Sra. Zolia Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 21 Fecha: 19/11/2025

Nombre caso de prueba: Visualización de solicitudes enviadas Sprint: 3

Módulo/sección a evaluar: Visualización de solicitudes enviadas Historia de usuario asociada: 21

Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
Como Administrador/Empleado/Cliente Quiero visualizar las solicitudes de adopción enviadas Para conocer el estado actual de la solicitud.

Escenario de prueba:
Dado la visualización de las solicitudes de adopción enviadas Cuando presione "Solicitudes enviadas" Entonces se desplegará una lista de las solicitudes que han sido enviadas.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en el perfil de usuario.

Pasos y condiciones de ejecución

- Presionar "Solicitudes Enviadas"

Resultado esperado

- Visualización de las solicitudes enviadas.

Estado de prueba	Éxito	Falló
Errores asociados:	Si	No


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 22 Fecha: 19/11/2025

Nombre caso de prueba: Visualización de solicitudes recibidas Sprint: 3

Módulo/sección a evaluar: Visualización de solicitudes recibidas Historia de usuario asociada: 22

Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
Como Administrador/Empleado/Cliente Quiero visualizar las solicitudes de adopción recibidas Para llevar un control de las solicitudes recibidas.

Escenario de prueba:
Dado la visualización de las solicitudes de adopción recibidas Cuando presione "Solicitudes recibidas" Entonces se desplegará una lista de las solicitudes que han sido enviadas por cada mascota.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en el perfil de usuario.


Pasos y condiciones de ejecución

- Presionar "Solicitudes Recibidas"

Resultado esperado

- Visualización de las solicitudes recibidas.

Estado de prueba	Éxito	Falló
Errores asociados:	Si	No


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 23 Fecha: 19/11/2025

Nombre caso de prueba: Gestión de solicitudes de adopción Sprint: 3

Módulo/sección a evaluar: Gestión de solicitudes de adopción Historia de usuario asociada: 23

Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
Como Empleado/Cliente Quiero gestionar las solicitudes de adopción Para poder Aceptar/Rechazar la solicitud.

Escenario de prueba:
Dado la gestión de las solicitudes de adopción Cuando seleccione la opción "Aceptar" Entonces cambiará el estado de la solicitud a Aceptada.
Dado la gestión de las solicitudes de adopción Cuando seleccione la opción "Rechazar" Entonces cambiará el estado de la solicitud a Rechazada.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en solicitudes recibidas.


Pasos y condiciones de ejecución

- Elegir la solicitud a gestionar.
- Dar click en el botón "Aceptar/Rechazar"

Resultado esperado

- Aceptación/Rechazo de la solicitud.

Estado de prueba	Éxito	Falló
Errores asociados:	Si	No


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 24 Fecha: 19/11/2025

Nombre caso de prueba: Notificación de solicitud Sprint: 3

Módulo/sección a evaluar: Notificación de solicitud Historia de usuario asociada: 24

Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
Como Administrador/Empleado/Cliente Quiero visualizar notificaciones sobre mis solicitudes de adopción Para estar informado del cambio realizado en su estado.

Escenario de prueba:
Dado la visualización de las notificaciones de mis solicitudes de adopción Cuando presione el icono de notificación Entonces se mostrarán las notificaciones recibidas.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en solicitudes de ejecución.
- Dar click en icono de notificación.

Resultado esperado

- Visualización de las notificaciones recibidas.

Estado de prueba	Éxito	Falló
Errores asociados:	Si	No


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 25 Fecha: 19/11/2025

Nombre caso de prueba: Recomendación de mascotas personalizadas Sprint: 3

Módulo/sección a evaluar: Recomendación de mascotas personalizadas Historia de usuario asociada: 25

Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
Como Administrador/Empleado/Cliente Quiero obtener una recomendación de mascotas personalizadas Para encontrar mascotas de acuerdo a mis intereses.

Escenario de prueba:
Dado la obtención de la recomendación de mascotas personalizadas Cuando acceda al apartado de "Match" Entonces se mostrarán las recomendaciones de mascotas.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.

Pasos y condiciones de ejecución

- Presionar "Match" en la barra de navegación.

Resultado esperado

- Visualización de las recomendaciones de mascotas.

Estado de prueba	Éxito	Falló
Errores asociados:	Si	No


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
Sede Santo Domingo
Escuela de Sistemas

PUCE

CASO DE PRUEBA 26 Fecha: 19/11/2025

Nombre caso de prueba: Enviar valoración Sprint: 3

Módulo/sección a evaluar: Enviar valoración Historia de usuario asociada: 26

Técnica de prueba: Caja Negra @ Caja Blanca □ Tipo: Prueba de Aceptación

Descripción:
Como Cliente Quiero enviar una valoración de la página Para expresar mi experiencia en la aplicación.

Escenario de prueba:
Dado el envío de la valoración de la página Cuando seleccione la cantidad de estrellas que desea mostrar, escriba un comentario y presione "Enviar" Entonces mostrara un mensaje de confirmación.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en el inicio.


Pasos y condiciones de ejecución

- Desplazarse al final de la página.
- Escribir la valoración.
- Dar click en el botón "Enviar".

Resultado esperado

- Se eliminará la valoración seleccionada.

Estado de prueba	Éxito	Falló
Errores asociados:	Si	No


Srta. Zola Caguan
PRODUCT OWNER

Pontificia Universidad Católica del Ecuador
 Santo Domingo
 Ecuador

PUCE

CASO DE PRUEBA 27 Fecha: 16/11/2025

Nombre caso de prueba: Gestión de valoración Sprint: 3
 Modificación a realizar: Gestión de valoración Historia de usuario asociada: 27
 Técnica de prueba: Caja Negra B Caja Blanca C Tipo: Prueba de Aceptación

Descripción:
 Como Administrador/Empleado Quiero gestionar las valoraciones de la página Para tener un control de la información publicada.

Escenario de prueba:
 Dado a la gestión de valoraciones de la página Cuando seleccione la valoración y de click en el icono de eliminar Entonces mostrará un mensaje de confirmación.

Pre-condiciones

- Tener acceso a internet.
- Haber iniciado sesión.
- Estar en el apartado de valoraciones.

Pasos y condiciones de ejecución

- Encontrar la valoración que desee.
- Dar click en el icono de eliminar

Resultado esperado

- Mensaje de confirmación.

Estado de prueba	Éxito	Fallo
Errores asociados:	Si	No

Sra. Zule Caguán
 PRODUCT OWNER

Anexo VI: Manual de Usuario

31 DE DICIEMBRE DE 2025

APLICACIÓN WEB CON MACHINE LEARNING PARA LA GESTIÓN DE ADOPCIONES DE MASCOTAS EN EL CENTRO VETERINARIO MI MASCOTA EN EL CANTÓN SANTO DOMINGO.

MANUAL DE USUARIO
 IEEE 1063-2001

JAIME ORTIZ – NATHALY ZAMORA
 PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
 Santo Domingo

MI MASCOTA V 1.0

REGISTRO DE CAMBIOS

FECHA	USUARIO	VERSIÓN	ACCIONES
31-DEC-2025	Jaime	1.0	CREACIÓN

Manual de usuario | IEEE 1063 – 2001 1

MI MASCOTA V 1.0

Tabla de Contenidos

1	INTRODUCCIÓN.....	3
2	CONCEPTO DE LAS OPERACIONES.....	3
3	PROCEDIMIENTOS.....	4
3.1	Aplicación web.....	4
3.1.1	Página Principal.....	4
3.1.2	Inicio de sesión.....	6
3.1.3	Registro de usuario.....	7
3.1.4	Apartado de Perros.....	9
3.1.5	Detalle y Ficha Técnica de la Mascota.....	11
3.1.6	Formulario de Adopción.....	12
3.1.7	Sobre la Empresa (Nosotros).....	13
3.1.8	Perfil de Usuario.....	14
3.1.9	Agregar Mascota.....	15
3.1.10	Editar Mascota.....	16
3.1.11	Formulario de Editar Mascotas.....	17
3.1.12	Eliminar Mascota.....	18
3.1.13	Mis Publicaciones.....	19
3.1.14	Solicitudes Enviadas.....	20
3.1.15	Solicitudes recibidas.....	21
3.1.16	Adopciones Realizadas.....	22
3.1.17	Favoritos.....	23
3.1.18	Experiencias.....	24
3.1.19	Recomendaciones.....	25
3.1.20	Administración de Usuarios (SOLO ADMINISTRADORES).....	27
3.1.21	Gestión de Aprobación de Publicaciones.....	28
4	MENSAJES DE ERROR Y RESOLUCIÓN DE PROBLEMAS.....	30
5	GLOSARIO.....	32
6	REFERENCIAS.....	32

Manual de usuario | IEEE 1063 – 2001 2

MI MASCOTA V 1.0

1 INTRODUCCIÓN

En la actualidad, el abandono de animales domésticos se ha consolidado como una problemática social y de salud pública en el Ecuador, especialmente en cantones con un crecimiento urbano acelerado como Santo Domingo. Miles de perros y gatos enfrentan situaciones de vulnerabilidad, lo que exige una respuesta coordinada entre la comunidad y las organizaciones de rescate. Ante este escenario, la adopción responsable emerge no solo como un acto de humanidad, sino como la estrategia más efectiva para garantizar el bienestar animal y fomentar una cultura de respeto hacia la vida silvestre y doméstica.

El Manual de Usuario tiene como finalidad proporcionar una guía detallada sobre el funcionamiento y uso del sistema web "Adopciones MI Mascota". Este documento está dirigido tanto al personal administrativo (Gerente y trabajadores) de la empresa, como a los usuarios finales que interactúan con la plataforma, sirviendo como el instrumento técnico que facilita la navegación y operación de todas las funcionalidades del software.

El propósito de este manual es describir de manera secuencial y lógica las funcionalidades del sistema, permitiendo a los usuarios comprender el flujo de navegación, desde el registro y autenticación, hasta la gestión integral de mascotas y la formalización de procesos de adopción.

2 CONCEPTO DE LAS OPERACIONES

Dado que es una aplicación web, no requiere instalación local, pero se sugiere:

- Dispositivo: Computadora de escritorio, Laptop, Tablet o Teléfono Inteligente.
- Conexión a Internet: Estable, para la carga de imágenes y comunicación con el servidor.
- Pantalla: Resolución mínima de 1280x720 para una visualización óptima en escritorio.

Manual de usuario | IEEE 1063 – 2001 3

MI MASCOTA V 1.0

3 PROCEDIMIENTOS

3.1 Aplicación web

3.1.1 Página Principal

La página de inicio es el primer punto de contacto del usuario con la plataforma "MI Mascota". Su objetivo es presentar la misión de la veterinaria, guiar al usuario en el proceso de adopción y mostrar las mascotas destacadas.



Barra de Navegación: Este componente se mantiene visible en la parte superior para facilitar el acceso a todo el sistema:

- Menú de Navegación:** Enlaces directos a las secciones de Inicio, Gatos, Perros, Match, Nosotros, Notificaciones.
- Acceso de Usuario:** Iniciar Sesión. Al hacer clic, el usuario es redirigido a la página de Login/registro.

Botón Explorar: Al hacer clic, el sistema realiza un desplazamiento automático hacia el catálogo de mascotas disponibles.

Manual de usuario | IEEE 1063 – 2001 4

MI MASCOTA V 1.0

Cómo Funciona el Proceso de Adopción



En el cuerpo central de la página se detalla brevemente el proceso de adopción.

- Guía de Pasos:** Línea numerada que detalla el flujo del proceso.
- Botón Comienza ahora:** Redirige al usuario a la página de inicio de sesión para iniciar su trámite formal.



Al final de la página se dispone de un espacio dedicado a las valoraciones. En este apartado, los clientes pueden visualizar las experiencias de otros adoptantes, lo que contribuye a la transparencia y confianza en la plataforma.

3.1.2 Inicio de sesión

Manual de usuario | IEEE 1063 – 2001 5



Esta sección permite a los usuarios registrados acceder a su cuenta personalizada para interactuar con todas las funcionalidades de la plataforma.

Formulario de Acceso: Permite validar la identidad del usuario mediante credenciales previamente registradas:


- **Email:** Campo de texto para ingresar el correo electrónico asociado a la cuenta.
- **Contraseña:** Campo de seguridad para ingresar la clave de acceso.
- **Botón "Iniciar Sesión":** Ejecuta la validación de los datos y permite el ingreso al sistema.

Opciones de Recuperación: Funciones adicionales para asistir al usuario en caso de inconvenientes o falta de cuenta:

- **¿Olvidaste tu contraseña?:** Enlace que redirige a la vista de "Restablecer Contraseña", donde el usuario puede solicitar un enlace de recuperación vía correo electrónico.
- **¿No tienes cuenta? Regístrate:** Enlace directo para acceder al formulario de creación de una nueva cuenta de usuario.

3.1.3 Registro de usuario

Manual de usuario | IEEE 3063 – 2001 6




Los nuevos usuarios pueden crear una cuenta proporcionando su información personal y de contacto para participar en los procesos de adopción.

Formulario de Datos Personales: Recopila la información necesaria para establecer el perfil del usuario en el sistema:

- **Nombre y Apellido:** Campos de texto para la identificación oficial del usuario.
- **Email:** Registro del correo electrónico que servirá como identificador de cuenta.
- **Contraseña:** Creación de una clave segura para futuros accesos.
- **Teléfono (opcional):** Campo para ingresar un número de contacto adicional.

Manual de usuario | IEEE 3063 – 2001 7



- **Imagen de Perfil (opcional):** Botón "Seleccionar Archivo" que permite cargar una fotografía en formatos JPEG, JPG o PNG. Al activarse, se despliega el explorador de archivos del sistema operativo para elegir la imagen.

Tras completar la información y aceptar los términos, se debe presionar el botón de Registrarse para procesar la solicitud en el servidor.

Manual de usuario | IEEE 3063 – 2001 8



Mensaje de Verificación: Al finalizar el registro, el sistema despliega un aviso destacado en color verde indicando que se ha enviado un enlace de verificación al correo electrónico proporcionado.

Nota Importante: Es un requisito obligatorio que el usuario acceda a su bandeja de entrada y verifique su cuenta antes de que el sistema le permita realizar el primer inicio de sesión.

3.1.4 Apartado de Perros



En esta sección, el sistema permite la exploración y filtrado preciso de los ejemplares caninos disponibles para adopción.

Manual de usuario | IEEE 3063 – 2001 9

MI MASCOTA V 1.0

Herramientas de búsqueda y filtrado: Este bloque permite refinar los resultados mediante criterios específicos:

- **Edad:** Menú desplegable con rangos predefinidos: Cachorro (0-12m), Joven (1-3a) y Adulto (>3a).
- **Sexo:** Selección exclusiva entre Macho o Hembra.
- **Tamaño:** Opciones de clasificación según dimensiones: Pequeño, Mediano o Grande.
- **Orden:** Permite clasificar por "Más recientes", "Más antiguos" o por orden alfabético "A-Z".
- **Buscar por nombre:** Campo de texto libre para localizar una mascota específica.
- **Botón Limpiar:** Restablece todos los parámetros para mostrar el catálogo completo.



Visualización de tarjetas de mascotas: Presenta los resultados en una cuadrícula con información resumida de cada perro:

- **Imagen:** Fotografía principal de la mascota.
- **Nombre:** Identificador del ejemplar.
- **Chips de información:** Etiquetas visuales que resumen el sexo, la edad y la raza de forma rápida.
- **Botón Ver más:** Al hacer clic, el usuario accede a la ficha técnica completa.

Manual de usuario | IEEE 2063 – 2001 10

MI MASCOTA V 1.0

3.1.5 Detalle y Ficha Técnica de la Mascota



Al seleccionar una mascota, el usuario accede a este panel donde se detalla toda la información relevante para el proceso de adopción.

Área Multimedia e Interacción:

- **Carousel de Imágenes:** Permite visualizar diferentes fotografías de la mascota mediante controles de navegación lateral.
- **Etiqueta de Estado:** Indica la disponibilidad actual (ej. ADOPTADO).
- **Favoritos:** Icono de estrella que permite al usuario guardar el perfil en su lista personal.

Información Detallada:

- **Datos:** Lista de atributos físicos: Especie, Sexo, Edad, Raza, Peso y Tamaño.
- **Personalidad y Temperamento:** Etiquetas dinámicas que describen el comportamiento.
- **Información adicional:** Bloque de texto descriptivo con detalles sobre la historia o necesidades de la mascota.
- **Botón ADOPTAR:** Inicia el flujo formal de solicitud de adopción.

Manual de usuario | IEEE 2063 – 2001 11

MI MASCOTA V 1.0

3.1.6 Formulario de Adopción



Este módulo recolecta la información del solicitante para asegurar el cumplimiento de los estándares de bienestar animal.

Resumen de la Solicitud: Muestra una ficha técnica reducida de la mascota seleccionada para confirmación del usuario.


Secciones del Formulario:

- **Información Personal y de Contacto:** Campos para Cédula, Nombres, Fecha de nacimiento, Correo electrónico y Teléfono.
- **Información sobre la Vivienda y Entorno:** Recopila datos sobre el tipo de hogar, régimen de propiedad (dueño o alquila), existencia y tamaño de patio o jardín, y número de convivientes.
- **Experiencia con Mascotas:** Consulta sobre la tenencia previa de animales y los motivos para la adopción.
- **Cuidado y Compromiso:** Preguntas sobre la ubicación de la mascota (dentro o fuera), horas de soledad, responsables directos del cuidado, existencia de un veterinario de confianza y disposición económica.
- **Botón Enviar:** Remite la solicitud al personal administrativo para su evaluación.

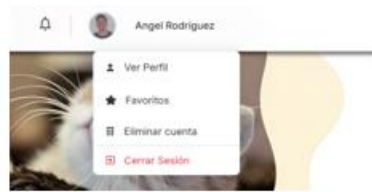
Manual de usuario | IEEE 2063 – 2001 12

MI MASCOTA V 1.0

3.1.7 Sobre la Empresa (Nosotros)



Este apartado tiene como función principal establecer un vínculo de confianza con el usuario, comunicando los valores y la capacidad operativa de la organización.



Menú de Usuario: Accesible desde la barra de navegación, ofrece opciones rápidas de "Ver Perfil", "Favoritos", "Eliminar cuenta" y "Cerrar Sesión".

Manual de usuario | IEEE 2063 – 2001 13

MINASCOTA V 1.0

3.1.8 Perfil de Usuario



Panel centralizado donde el usuario administra sus datos y monitorea sus actividades en el sistema.

Gestión de Identidad y Seguridad:

- **Foto de Perfil:** Visualización de la imagen actual con opción "Cambiar Foto" para actualización.
- **Información Personal:** Campos editables de Nombre, Apellido, Correo electrónico, Teléfono y Edad.
- **Acciones de Cuenta:** Botones directos para "Editar Información" y "Cambiar contraseña".

Herramientas de Control:

- **Funcionalidades:** Botones para la gestión de publicaciones (Agregar, Editar o Eliminar mascotas) y acceso a "Mis Publicaciones".
- **Adopciones y solicitudes:** Acceso rápido al seguimiento de solicitudes enviadas, solicitudes recibidas por terceros y el registro de adopciones que ya se han completado.

Manual de usuario | IEEE 2003 – 2001 14

MINASCOTA V 1.0

3.1.9 Agregar Mascota



Este apartado permite a los usuarios registrar una nueva mascota en la plataforma para que esté disponible en el sistema de adopciones.

Formulario de Datos Básicos y Descriptivos:

Este bloque centraliza la captura de información estructurada necesaria para los filtros de búsqueda del sistema:

- **Identificación Básica:** Campos para el Nombre, Especie y Sexo
- **Atributos Físicos:** Campos numéricos y de texto para definir la Edad (Meses), la Raza y el Peso (Kg).
- **Categorización de Tamaño:** Menú desplegable para clasificar al ejemplar según dimensiones.
- **Personalidad y temperamento:** Nutrio de etiquetas que permiten definir el perfil psicológico de la mascota. El usuario puede seleccionar múltiples atributos.
- **Información adicional:** Área de texto extendido para detallar antecedentes médicos, historia de rescate o recomendaciones específicas de convivencia.

Manual de usuario | IEEE 2003 – 2001 15

MINASCOTA V 1.0

Gestión Multimedia y Carga de Archivos:

Panel dedicado a la identidad visual del anuncio, optimizado para asegurar la calidad de las publicaciones:

- **Zona de Carga:** Área interactiva con soporte para la función "Drag & Drop" o selección manual mediante el botón "Subir Imagen".
- **Visualización Previa:** Permite al usuario verificar la imagen antes de proceder al guardado final en el servidor.

3.1.10 Editar Mascota



Este módulo permite actualizar o corregir la información de una mascota ya existente en el sistema. Se accederá al formulario de actualización de datos, por medio del botón "Editar"

Manual de usuario | IEEE 2003 – 2001 16

MINASCOTA V 1.0

3.1.11 Formulario de Editar Mascotas



Este módulo permite actualizar o corregir la información de una mascota ya existente en el sistema. Al acceder, el formulario se presenta con la información cargada previamente para su modificación.

Formulario de Actualización de Datos:

- **Campos Modificables:** Permite editar el Nombre, Especie, Sexo, Edad, Raza, Peso y Tamaño.
- **Ajuste de Temperamento:** Posibilidad de añadir o remover etiquetas de personalidad según la evolución del comportamiento del animal.
- **Información Adicional:** Área de texto para actualizar detalles sobre el estado de salud o nuevas observaciones.

Gestión Multimedia Avanzada:

A diferencia del registro inicial, aquí el usuario puede gestionar las fotos actuales:

- **Visualización de Galería:** Muestra las imágenes que ya están en el sistema (Máximo 3).
- **Eliminar Fotos:** Cada imagen cuenta con un icono de cierre (X) para eliminar fotografías antiguas o de baja calidad.

Manual de usuario | IEEE 2003 – 2001 17

MIMASCOTA V 1.0

Una vez verificada la nueva información, con el botón de "Guardar Cambios" se realizarán los nuevos cambios.

3.1.12 Eliminar Mascota



Para mantener la integridad del catálogo, el sistema permite retirar anuncios de mascotas que ya han sido adoptadas o que, por motivos externos, ya no se encuentran disponibles. Desde el panel de "Mis Publicaciones", el usuario selecciona el ejemplar y hace clic en el botón Eliminar.

3.1.13 Ventana de Eliminar Publicación



Manual de usuario | IEEE 2063 – 2001 18

MIMASCOTA V 1.0

Protocolo de Confirmación:

Para evitar eliminaciones accidentales, el sistema despliega un cuadro de diálogo de seguridad:

- **Mensaje de Advertencia:** Notifica al usuario que esta acción no se puede deshacer.
- **Botón Cancelar:** Cierra la ventana sin realizar cambios, manteniendo la publicación intacta.
- **Botón Aceptar:** Confirma la voluntad del usuario, retirando permanentemente la mascota del sistema y de la base de datos pública.

3.1.14 Mis Publicaciones



Este panel centralizado permite al usuario visualizar y administrar de forma exclusiva los ejemplares que ha registrado previamente en la plataforma.

Herramientas de Localización:

Ubicadas en la parte superior para facilitar la gestión cuando el usuario posee múltiples registros:

- **Buscador por nombre:** Campo de texto dinámico para filtrar rápidamente entre las mascotas publicadas.
- **Botón Limpiar:** Restablece la vista para mostrar la totalidad de las publicaciones propias.

Manual de usuario | IEEE 2063 – 2001 19

MIMASCOTA V 1.0

3.1.15 Solicitudes Enviadas



Este módulo permite al usuario realizar un control detallado y en tiempo real del estado de sus peticiones de adopción.

Panel de Navegación y Cabecera:

- **Botón Volver:** Ubicado en la parte superior izquierda, permite al usuario regresar al panel de su perfil de forma inmediata.

Tabla de Gestión de Solicitudes:

La información se presenta en una tabla estructurada que facilita la lectura de múltiples registros simultáneos, organizada en seis columnas técnicas:

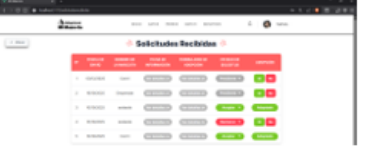
- **Fecha de Envío:** Registra el día exacto en que se remitió el formulario a la plataforma.
- **Nombre de la Mascota:** Identifica al ejemplar al que se aspira adoptar.
- **Ficha de Información:** Contiene el botón "Ver Detalles", el cual redirige al usuario al perfil público de la mascota para revisar sus características nuevamente.
- **Formulario de Adopción:** Incluye un botón "Ver Detalles" que permite al usuario visualizar el formulario que llenó originalmente.
- **Estado de Solicitud:** Columna de indicadores visuales que muestran la fase actual del proceso mediante etiquetas de color:

Manual de usuario | IEEE 2063 – 2001 20

MIMASCOTA V 1.0

- **Grís (Pendiente):** La solicitud ha sido recibida y está a la espera de revisión por el personal responsable.
- **Verde (Aceptada):** El perfil del solicitante ha sido aprobado para continuar con el proceso de adopción.
- **Rojo (Rechazada):** La solicitud no ha cumplido con los requisitos necesarios para la adopción del ejemplar.

3.1.16 Solicitudes recibidas



Este panel controla todas las peticiones de adopción que otros usuarios han realizado sobre las mascotas publicadas por el propietario actual.

Matriz de Evaluación de Candidatos:

La información se organiza en una tabla detallada que permite comparar y gestionar múltiples solicitudes:

- **Fecha de Envío:** Muestra el día en que el interesado remitió su postulación.
- **Nombre de la Mascota:** Identifica específicamente a qué ejemplar se refiere la solicitud.
- **Ficha de Información:** Incluye el botón "Ver detalles", el cual redirige al usuario al perfil público de su mascota para corroborar datos si fuera necesario.
- **Formulario de Adopción:** Contiene el botón "Ver detalles", que permite abrir y revisar toda la información personal, de vivienda y compromisos que el solicitante completó.

Manual de usuario | IEEE 2063 – 2001 21

MI MASCOTA V 1.0

- **Estado de Solicitud:** Menú desplegable de selección que permite al propietario marcar la fase de evaluación actual:
 - **Pendiente:** Estado inicial de toda solicitud recibida.
 - **Aceptar:** Indica que el perfil del solicitante ha superado la revisión inicial y es apto para continuar.
 - **Rechazar:** Indica que el solicitante no cumple con los requisitos necesarios para este ejemplar específico.
- **Adopción:** Esta columna representa la conclusión del proceso físico de entrega.
 - **Botones Sí / No:** Permiten confirmar si el proceso de adopción se completó satisfactoriamente.
 - **Etiqueta Adoptado:** Una vez que el usuario presiona "Sí", el sistema registra al ejemplar como exitosamente adoptado.

3.1.17 Adopciones Realizadas



Historial de Adopciones Exitosas:

La información se presenta en una tabla de solo lectura, diseñada para la consulta rápida de los procesos finalizados, organizada en seis columnas informativas:

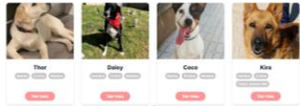
- **Fecha de Inicio:** Muestra la fecha original en la que se inició el contacto o se recibió la solicitud para dicho ejemplar.
- **Nombre de la Mascota:** Identifica al animal que fue adoptado satisfactoriamente.
- **Ficha de Información:** Incluye el botón "Ver detalles", que permite revisar el perfil histórico de la mascota.

Manual de usuario | IEEE 1063 – 2001 22

MI MASCOTA V 1.0

- **Formulario de Adopción:** Contiene el botón "Ver detalles", el cual permite consultar el documento del adoptante.
- **Estado de Solicitud:** Muestra una etiqueta visual de color verde con el texto "Adoptado".

3.1.17 Favoritos



Esta interfaz permite al usuario gestionar y visualizar de forma centralizada todas las mascotas que han sido marcadas como de su interés. Es una herramienta de seguimiento personalizada para facilitar la toma de decisiones antes del proceso de adopción.

- **Imagen:** Fotografía principal de la mascota.
- **Nombre:** Identificador del ejemplar.
- **Chips de información:** Etiquetas visuales que resumen el sexo, la edad y la raza de forma rápida.
- **Botón Ver más:** Al hacer clic, el usuario accede a la ficha técnica completa.

Manual de usuario | IEEE 1063 – 2001 23

MI MASCOTA V 1.0

3.1.18 Experiencias



Este apartado está diseñado para que los adoptantes compartan sus testimonios y califiquen su experiencia con la plataforma. Fomenta la transparencia y permite que otros usuarios conozcan el impacto de las adopciones realizadas a través de la aplicación web.

Gestión de Valoraciones:

La sección izquierda presenta un formulario interactivo para que el usuario pueda registrar su opinión:

- **Sistema de Calificación:** Permite otorgar una puntuación mediante una escala de cinco estrellas.
- **Caja de Comentarios:** Un campo de texto limitado a 100 caracteres para redactar una reseña breve sobre la experiencia de adopción.

Visualización y Filtros de Reseñas:

En la parte derecha se despliega el historial de testimonios de la comunidad, el cual cuenta con las siguientes herramientas de navegación:


- **Filtros Cronológicos:** Botones para organizar las opiniones entre "Más recientes" o "Más antiguas", facilitando la lectura de la actividad actual.

Manual de usuario | IEEE 1063 – 2001 24

MI MASCOTA V 1.0

- **Tarjetas de Reseña:** Cada entrada muestra la identidad del usuario, su calificación por estrellas, el comentario redactado y la fecha exacta de publicación.
- **Gestión de Propias Reseñas:** Los usuarios tienen la posibilidad de identificar sus comentarios previos.

3.1.19 Recomendaciones



Esta interfaz modal constituye el núcleo del sistema de recomendación inteligente. A través de una serie de preguntas estratégicas, la plataforma recopila datos sobre el perfil del adoptante para alimentar el algoritmo que genera el Match Perfecto.

Estructura del Formulario:


El cuestionario está organizado por secciones temáticas para evaluar la compatibilidad del entorno del usuario con las necesidades de las mascotas, por ejemplo:

- **Sección 1: Estilo de Vida:** Enfocada en determinar la disponibilidad y dinámica diaria del usuario.
 - **Disponibilidad horaria:** Consulta cuántas horas permanece el usuario en casa (menos de 4 horas hasta más de 8 horas).
 - **Nivel de actividad física:** Define el perfil del adoptante desde sedentario hasta muy activo.

Manual de usuario | IEEE 1063 – 2001 25

MI MASCOTA V 1.0

- o **Experiencia previa:** Evalúa el historial del usuario con otras mascotas (ninguna, poca, moderada o experta).
- **Control de Navegación:**
 - o **Botón Cerrar (X):** Permite salir del cuestionario y regresar a la vista de recomendaciones sin guardar cambios parciales.
 - o **Opciones de Selección Única:** Utiliza botones de radio para asegurar respuestas precisas en cada categoría.



Una vez completado el cuestionario, los datos son procesados para actualizar dinámicamente el listado de "Match", ajustando los porcentajes de compatibilidad mostrados en las tarjetas de las mascotas.

Panel de Recomendaciones:

- **Porcentaje de Compatibilidad:** Cada ejemplar muestra una etiqueta con el "% Match", indicando el nivel de afinidad calculado por el sistema.
- **Criterio de Ordenamiento:** Las mascotas aparecen listadas automáticamente de mayor a menor grado de compatibilidad para facilitar la elección.
- **Botón Cambiar mis preferencias:** Permite al usuario regresar al cuestionario de perfilado para ajustar sus intereses y actualizar las recomendaciones generadas.

Manual de usuario | IEEE 1063 – 2001 26

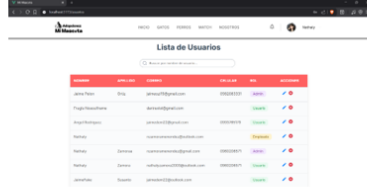
MI MASCOTA V 1.0

Visualización de Resultados:

Las sugerencias se presentan mediante tarjetas informativas que contienen:

- **Fotografía de Identificación:** Imagen principal de la mascota recomendada.
- **Atributos Destacados:** Nombre, sexo, edad aproximada y raza/tipo.
- **Enlace de Detalle:** el botón "Ver más" dirige al usuario a la ficha completa de la mascota para profundizar en su historia y necesidades médicas.

3.1.20 Administración de Usuarios (SOLO ADMINISTRADORES)



Este módulo está reservado exclusivamente para usuarios con privilegios de Administrador o Empleado. Permite la supervisión, control y moderación de todas las cuentas registradas en la plataforma.

Herramientas de Localización:

- **Buscador por nombre de usuario:** Campo de texto dinámico que permite filtrar y encontrar rápidamente a cualquier integrante de la plataforma mediante su nombre.

Matriz de Información de Usuarios:

La información se presenta en una tabla técnica diseñada para la auditoría rápida de perfiles, organizada en las siguientes columnas:

Manual de usuario | IEEE 1063 – 2001 27

MI MASCOTA V 1.0

- **Nombre y Apellido:** Identificación oficial del usuario registrado.
- **Correo:** Dirección electrónica vinculada a la cuenta para comunicaciones oficiales.
- **Celular:** Número de contacto telefónico del usuario.
- **Rol:** Indicador visual que define el nivel de acceso y permisos dentro del sistema mediante etiquetas de colores distintivos:
 - o **Admin (Púrpura):** Usuarios con control total sobre las funciones administrativas y de configuración.
 - o **Empleado (Amarillo):** Personal con permisos para gestionar mascotas y supervisar procesos de adopción.
 - o **Usuario (Verde):** Perfil estándar destinado a la adopción y publicación de mascotas.

Control de Acciones Administrativas:

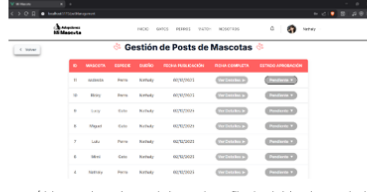
Permite ejecutar cambios directos sobre las cuentas de terceros mediante dos funciones:

- **Gestión de Roles (Icono Lápiz Azul):** Permite al administrador modificar el rango de un usuario. A través de esta herramienta se puede ascender a un Usuario a Empleado o Administrador.
- **Bloqueo de Usuario (Icono Rojo):** Función de moderación que permite restringir el acceso a la plataforma a usuarios que no cumplan con las normas de convivencia o bienestar animal, inhabilitando su cuenta de forma inmediata.

Manual de usuario | IEEE 1063 – 2001 28

MI MASCOTA V 1.0

3.1.21 Gestión de Aprobación de Publicaciones



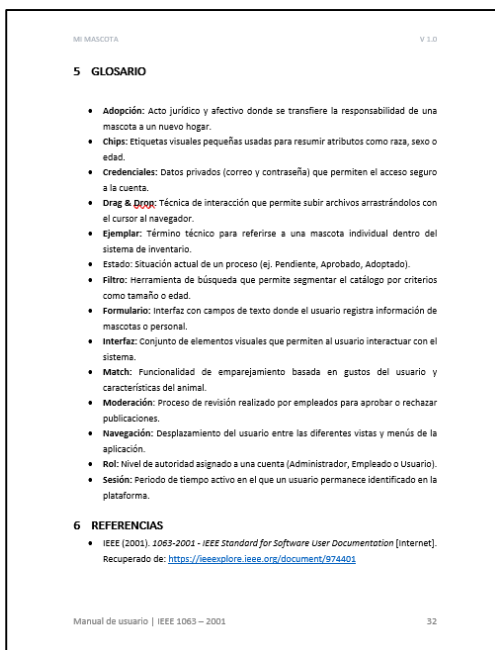
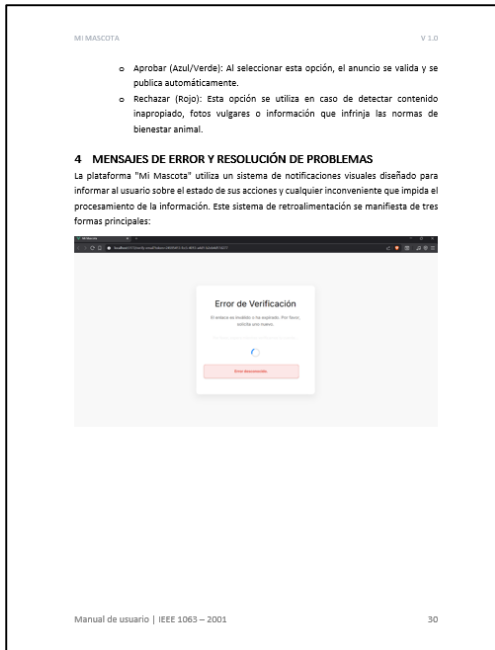
Este módulo es una herramienta exclusiva para los perfiles de Administrador y Empleado. Su objetivo principal es actuar como un filtro de seguridad y calidad, permitiendo supervisar cada anuncio antes de que sea visible para el público general.

Matriz de Control de Publicaciones:

La información de los anuncios entrantes se presenta en una tabla técnica organizada en siete columnas:

- **ID:** Código numérico único asignado por el sistema para el seguimiento interno de cada publicación.
- **Mascota:** Nombre del ejemplar registrado por el usuario.
- **Especie:** Categorización del animal (Perro o Gato).
- **Dueño:** Identifica al usuario responsable de la publicación, facilitando el contacto o seguimiento en caso de irregularidades.
- **Fecha de Publicación:** Registra el momento exacto en que se remitió la información al sistema.
- **Ficha Completa:** Contiene el botón "Ver Detalles", el cual es obligatorio consultar para revisar las fotografías y la descripción del anuncio antes de emitir un juicio.
- **Protocolo de Aprobación y Seguridad:** Permite gestionar la visibilidad del anuncio mediante un menú desplegable de tres estados:
 - o **Pendiente (Gris):** Estado inicial de toda publicación nueva. El anuncio permanece oculto al catálogo público hasta que un administrador lo revise.

Manual de usuario | IEEE 1063 – 2001 29



Anexo VII: Manual técnico

28 DE DICIEMBRE DE 2025

MANUAL TÉCNICO
IEEE 1063-2001

JAIME ORTIZ – NATHALY ZAMORA
INSTR. EN INGENIERÍA EN SISTEMAS DE COMPUTACIÓN
Barru, Durango

V 1.0

REGISTRO DE CAMBIOS

FECHA	USUARIO	VERSIÓN	ACCIONES
28 DIC-25	NATHALY	1.0	CREACIÓN

Manual técnico | IEEE 1063 – 2001 1

V 1.0

Tabla de Contenidos

1	OBJETIVOS	3
1.1	General	3
1.2	Específico	3
2	INTRODUCCIÓN	3
3	PROCEDIMIENTOS	3
3.1	Clonar repositorio de GitHub	3
3.2	Archivos del Backend	4
3.2.1	Archivo Principal	4
3.2.2	Configuración	5
3.2.3	Modelos de Datos	5
3.2.4	Controladores	6
3.2.5	Rutas del Sistema	6
3.2.6	Middleware	7
3.3	Archivos del Frontend	7
3.3.1	Carpeta principal	7
3.3.2	Archivos assets	8
3.3.3	Configuración e Inicialización	9
3.3.4	Comunicación HTTP	9
3.3.5	Rutas	10
3.3.6	Vistas	10
3.3.7	Componentes	11
4	ACCESO A LOS DATOS	12
5	GLOSARIO	12
6	REFERENCIAS	12

Manual técnico | IEEE 1063 – 2001 2

V 1.0

1 OBJETIVOS

1.1 General

Brindar una guía exhaustiva sobre los procesos y requisitos de la aplicación, permitiendo a los usuarios un manejo óptimo y eficiente de la aplicación web de gestión adopción de mascotas.

1.2 Específico

- Detallar las funcionalidades que integran la aplicación.
- Redactar las instrucciones para la correcta implementación y ejecución de la aplicación.
- Definir recomendaciones que aseguren la máxima eficiencia en el uso de la aplicación.

2 INTRODUCCIÓN

En el presente documento, se detallarán las funcionalidades que integran la aplicación, las instrucciones para su correcta implementación y ejecución, así como las recomendaciones para asegurar su óptimo funcionamiento en la gestión de la adopción de mascotas.

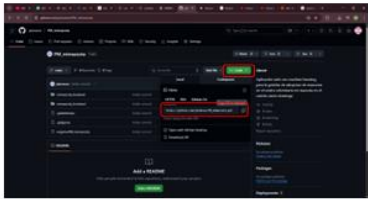
3 PROCEDIMIENTOS

3.1 Clonar repositorio de GitHub

Ingresar al repositorio de GitHub del proyecto y copiar la dirección web que se encuentra al desplegar el botón verde "Code". Posteriormente, se debe acceder a la consola o terminal del sistema, se ingresa el comando de clonación seguido de la URL obtenida: `git clone https://github.com/jaimexoz/PM_mimascota`


Manual técnico | IEEE 1063 – 2001 3

V 1.0



3.2 Archivos del Backend

3.2.1 Archivo Principal
 El archivo "server.js", ubicado en la raíz del backend, funciona como el punto de entrada y configuración central del servidor Express, encargándose de inicializar la aplicación y coordinar sus servicios. Este módulo implementa los middlewares esenciales y centraliza las rutas de la API.



Manual técnico | IEEE 1063 – 2001 4

V 1.0

3.2.4 Controladores
 La carpeta "controllers" actúa como intermediario entre las rutas y los modelos de datos. Sus archivos procesan las peticiones del cliente ejecutando validaciones, coordinando servicios externos como Cloudinary y manejando transacciones, para finalmente devolver las respuestas y códigos HTTP adecuados.



3.2.5 Rutas del Sistema
 La carpeta "routes" estructura la API REST organizando los endpoints por módulos funcionales mediante Express Router. Se encarga de mapear las peticiones HTTP hacia los controladores correspondientes e integra middlewares esenciales, como la autenticación y la carga de archivos.



Manual técnico | IEEE 1063 – 2001 6

V 1.0

3.2.6 Middleware
 El archivo middleware/authMiddleware.js constituye la capa de seguridad de la API, encargándose de validar la identidad y los permisos de los usuarios. Mediante la función protect, verifica los tokens JWT entrantes, mientras que authorize y authorizeOwnerOrAdmin gestionan el control de acceso estricto basado en roles y la propiedad de los recursos.



3.3 Archivos del Frontend

3.3.1 Carpeta principal
 La carpeta "mimascota_frontend" contiene la aplicación cliente. Esta estructura centraliza la configuración del entorno mediante vite.config.js y gestiona dependencias clave como Pinia, Vue Router y Axios. Organiza el código fuente en src y los recursos estáticos en public.

Manual técnico | IEEE 1063 – 2001 7

V 1.0



3.3.2 Archivos assets
 La carpeta "src/assets" centraliza la gestión de los recursos estáticos de la interfaz, almacenando imágenes y tipografías que definen la identidad visual y los temas globales de la aplicación.




Manual técnico | IEEE 1063 – 2001 8

V 1.0


3.3.3 Configuración e Inicialización

El archivo "main.js" actúa como el punto de entrada de la aplicación, encargado de inicializar la instancia de Vue e integrar servicios críticos como Pinia, Vue Router y la restauración de la sesión de usuario.



3.3.4 Comunicación HTTP

El archivo "http.js" centraliza y gestiona la comunicación entre el cliente y el servidor mediante la configuración de Axios. Se encarga de automatizar la seguridad de las comunicaciones, adjuntando las credenciales del usuario a todas las solicitudes y centralizando el control de los errores que pueda devolver el sistema.




Manual técnico | IEEE 1063 – 2001 9

V 1.0

3.3.5 Rutas

El archivo "router/index.js" actúa como el mapa de navegación de la aplicación. Su función es conectar cada dirección web (URL) con la vista correspondiente sin necesidad de recargar la página completa. Además, funciona como un filtro de seguridad, ya que antes de permitir la entrada a secciones privadas, verifica automáticamente si el usuario ha iniciado sesión, redirigiéndolo al login si no tiene permiso.




3.3.6 Vistas

En la carpeta "views" se encuentran los archivos que representan las páginas principales de la aplicación. Cada vista corresponde a una dirección web específica. Estos archivos gestionan la lógica de esa página, se conectan con la base de datos para traer información y organizan los elementos visuales para que el usuario pueda interactuar con ellos fácilmente.


Manual técnico | IEEE 1063 – 2001 10

V 1.0



3.3.7 Componentes

En la carpeta "components" se encuentran los módulos de la interfaz que están diseñados para ser reutilizados en múltiples vistas.



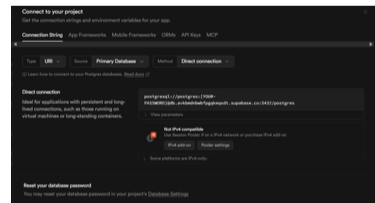
Manual técnico | IEEE 1063 – 2001 11

V 1.0

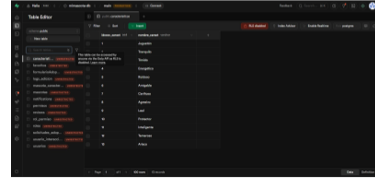
4 ACCESO A LOS DATOS

4.1 Supabase Database (PostgreSQL)

La base de datos fue desplegada en la nube utilizando el servicio BaaS (Backend as a Service) de Supabase. Para acceder a la base de datos se utilizan las credenciales y el string de conexión provistos por la plataforma en el apartado de configuración.



Se utiliza el Table Editor integrado en la interfaz de Supabase para acceder a la base de datos remota, permitiendo una gestión gráfica que facilita la administración de las tablas y registros de las adopciones.



Manual técnico | IEEE 1063 – 2001 12

BONES 3D V 1.0

5 GLOSARIO

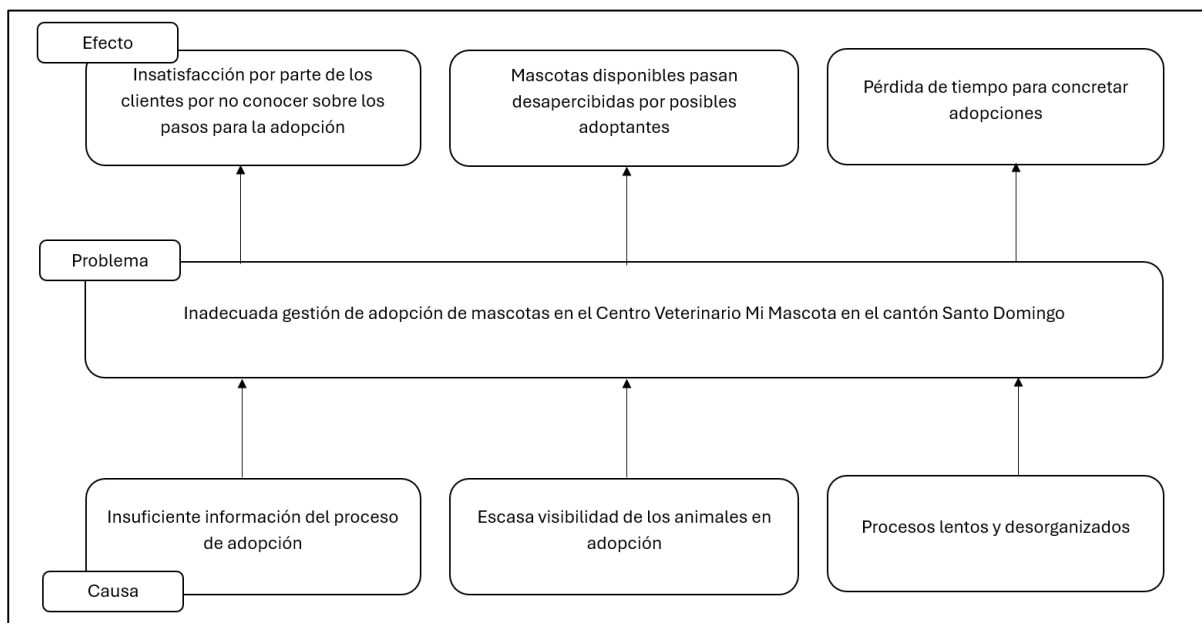
- **Axios:** Librería utilizada en el Frontend para gestionar la comunicación HTTP con el servidor, automatizando el envío de credenciales y el manejo de errores.
- **Backend:** Parte del sistema que se ejecuta en el servidor, abarcando la configuración central, acceso a datos, lógica de negocio y seguridad.
- **Cloudinary:** Servicio externo configurado en el Backend para gestionar el almacenamiento y optimización de las imágenes de las mascotas en la nube.
- **Componentes:** Módulos de la interfaz de usuario diseñados bajo el principio de reutilización para ser implementados en múltiples vistas de la aplicación.
- **Controladores:** Archivos ubicados que actúan como intermediarios, procesan las peticiones del cliente, validan datos y generan las respuestas HTTP adecuadas.
- **CRUD:** Se refiere a las operaciones básicas de base de datos encapsuladas dentro de los Modelos de Datos.
- **JWT (JSON Web Token):** Mecanismo de seguridad verificado por el middleware para validar la identidad de los usuarios en las peticiones entrantes.
- **Middleware:** Funciones intermedias que ejecutan lógica de seguridad, validación de permisos o procesamiento antes de que una petición llegue al controlador.
- **Modelos:** Capa del Backend que abstrae la complejidad de las consultas SQL y gestiona las transacciones con la base de datos PostgreSQL.
- **Nodemailer:** Librería para gestionar el envío de correos electrónicos desde el sistema.
- **Pinia:** Librería de gestión de estado utilizada en el Frontend para manejar datos compartidos entre componentes.
- **Vistas:** Archivos que representan las páginas principales y completas de la aplicación, encargadas de la lógica y presentación de cada sección.

6 REFERENCIAS

- IEEE (2001). 1063-2001 - *IEEE Standard for Software User Documentation* [Internet]. Recuperado de: <https://ieeexplore.ieee.org/document/974401>

Manual técnico | IEEE 1063 – 2001 13

Anexo VIII: Árbol del problema



Anexo IX: Recodificación de instrumentos.

Recodificación		Respuestas
		Edad
1		18-28
2		29-38
3		39-48
4		49-58
5		59 o más
		Género
1		Femenino

2	Masculino
Discapacidad	
1	Sí
0	No
Tipo de Discapacidad	
0	Sin selección
1	Visual
2	Auditiva
3	Física
4	Intelectual
5	Otra (Indique)
Recodificación	
Frecuencia	
4	Muy frecuentemente
3	Frecuentemente
2	Ocasionalmente
1	Raramente
0	Nunca
Dificultad	
4	Muy fácil
3	Fácil
2	Normal
1	Difícil
0	Muy difícil
Familiaridad	
4	Completamente familiarizado
3	Muy familiarizado
2	Algo familiarizado
1	Poco familiarizado
0	Nada familiarizado
De acuerdo	
4	Totalmente de acuerdo
3	Bastante de acuerdo
2	Neutral
1	Poco de acuerdo
0	Totalmente de desacuerdo
Efectividad	
4	Completamente efectivo
3	Altamente efectivo
2	Parcialmente efectivo
1	Poco efectivo
0	Nada efectivo
Satisfacción	
4	Totalmente satisfecho
3	Satisfecho
2	Moderadamente satisfecho
1	Poco satisfecho
0	Nada satisfecho
Necesidad	
4	Muy necesario

3	Necesario
2	Medianamente necesario
1	Poco necesario
0	Innecesario
Seguridad	
4	Totalmente seguro
3	Seguro
2	Moderadamente seguro
1	Poco seguro
0	Totalmente inseguro

Anexo X: Evidencia de la reunión de planificación del sprint

<p style="text-align: center;">Entrevista dirigida al gerente del Centro Veterinario Mi Mascota</p> <p>Pregunta 1: ¿Qué método usan para registrar a las mascotas que están disponibles para adoptar?</p> <p>Respuesta: Por lo general tenemos una Agenda. El día de la semana, agendamos a las personas cuantos gatitos van a traer. Y en eso nos basamos. Llenamos el nombre de la persona y la cantidad de animales.</p> <p>Pregunta 2: ¿Considera que la información sobre las mascotas y los adoptantes se gestiona de manera organizada?</p> <p>Respuesta: No, porque no llevamos un registro de quienes los llevan o como es la persona.</p> <p>Pregunta 3: ¿Qué tan difícil le resulta el proceso de registro para los adoptantes interesados?</p> <p>Respuesta: No llevamos un registro, no llevan ningún requisito, así que no es difícil.</p> <p>Pregunta 4: ¿Considera que la evaluación previa de los adoptantes es suficientemente rigurosa para asegurar hogares adecuados?</p> <p>Respuesta: Por lo general, siempre nos fijamos bien quien se los lleva. No le damos a cualquier persona y por lo general son clientes que vienen, los llevan y vuelven.</p> <p>Pregunta 5: ¿Con qué regularidad se proporciona capacitación sobre cómo manejar situaciones difíciles o sensibles con posibles adoptantes?</p> <p>Respuesta: No lo hacemos.</p> <p>Pregunta 6: ¿Utilizan protocolos de seguimiento post-adopción para asegurar el bienestar de los animales?</p> <p>Respuesta: No lo hacemos.</p> <p>Pregunta 7: ¿Qué medidas toman cuando reciben una mascota para dar en adopción?</p>	<p>Respuesta: Que vengan sanos, limpios y desparasitados. Hay muchas veces que los dejan aquí, los dejan botados en un saco. A veces están enfermos y tenemos que curarlos. Se los damos en adopción cuando ya son gatos grandes. También dejan botando perritos grandes, también los esterilizamos.</p> <p>Pregunta 8: ¿Considera útil la implementación de una aplicación web que permita gestionar la adopción de mascotas?</p> <p>Respuesta: Sí, sería bastante útil.</p> <p>Pregunta 9: ¿Qué tipo de información considera que debe ser visible en la aplicación web?</p> <p>Respuesta: Los datos de las personas adoptante, los datos de la persona y también las imágenes de las mascotas.</p> <p>Pregunta 10: ¿Le parece adecuado que en una aplicación web las publicaciones sean aprobadas antes de publicarlas?</p> <p>Respuesta: Sí, claro.</p> <p>Pregunta 11: ¿Considera qué es importante la implementación de herramientas que permitan mostrar al usuario recomendaciones personalizadas según su perfil?</p> <p>Respuesta: Sí, sería bueno.</p> <p>Pregunta 12: ¿Está de acuerdo en que se implemente un sistema que aprenda de las preferencias del usuario en las aplicaciones web?</p> <p>Respuesta: Sí, sería bueno.</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

[Evidencia de reunión de planificación de sprint](#)

Anexo XI: Evidencia de encuesta

[Evidencia encuesta](#)

Anexo XII: Informe del turnitin

TTG - V8

por Jaime Arturo Ortiz Jaramillo

Fecha de entrega: 24-feb-2026 09:10a. m. (UTC-0500)
Identificador de la entrega: 2887281762
Nombre del archivo: TIC-2025-02_Ortiz_y_Zamora_V11.docx (32.87M)
Total de palabras: 27070
Total de caracteres: 156783

TTG - V8

INFORME DE ORIGINALIDAD

7%	7%	3%	2%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	issuu.com Fuente de Internet	1%
----------	----------------------------------------	-----------

Jaime Arturo Ortiz Jaramillo

TTG - V8

TTG - V8

Trabajo Titulación Grado 202501

PUCE SANTO DOMINGO MOODLE

Detalles del documento

Identificador de la entrega

trn:oid:::1:3490018229

Fecha de entrega

24 feb 2026, 9:09 a.m. GMT-5

Fecha de descarga

24 feb 2026, 6:05 p.m. GMT-5

Nombre del archivo

TIC-2025-02_Ortiz_y_Zamora_V11.docx

Tamaño del archivo

32.9 MB

179 páginas

27.070 palabras

156.783 caracteres



Página 1 de 181 - Portada

Identificador de la entrega trn:oid:::1:3490018229

*% detectado como IA

La detección de IA incluye la posibilidad de que haya falsos positivos. Aunque cierto texto en esta entrega se generó probablemente con IA, los puntajes inferiores al umbral del 20 % no aparecen porque tienen una mayor probabilidad de falsos positivos.

Precaución: Se necesita revisión.

Es esencial comprender los límites de la detección de IA antes de tomar decisiones acerca del trabajo del estudiante. Te alentamos a obtener más información acerca de las funciones de detección de IA de Turnitin antes de usar la herramienta.

Aviso legal

Nuestra evaluación de escritura con IA está diseñada para ayudar a los académicos a identificar texto que podrían haberse preparado mediante una herramienta de IA generativa. Es posible que nuestra evaluación de escritura con IA no siempre sea precisa (existe la posibilidad de que identifique erróneamente redacciones probablemente generadas por humanos como generadas por IA, y redacciones probablemente generadas por IA como generadas por humanos), por lo que no debe usarse como único fundamento para aplicar sanciones a un estudiante. Para determinar si es un caso de deshonestidad académica, se necesita de un escrutinio mayor y el juicio humano, junto con la aplicación de las políticas académicas específicas de la organización.

Anexo XIII: Certificados de participación

