



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE SOFTWARE

Tema:

**IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN
DE PACIENTES DE LA CLÍNICA VETERINARIA PET'S HOME DE LA
CIUDAD DE AMBATO.**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Software

ÁREA: Gestión de las tecnologías de la información

LÍNEA DE INVESTIGACIÓN: Tecnología de la información y Sistemas de
control

AUTOR: Christian Marcelo Jaramillo Collazo

TUTOR: Ing. Carlos Israel Núñez Miranda, Mg.

Ambato - Ecuador

febrero – 2025

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE PACIENTES DE LA CLÍNICA VETERINARIA PET'S HOME DE LA CIUDAD DE AMBATO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Christian Marcelo Jaramillo Collazo, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2025.

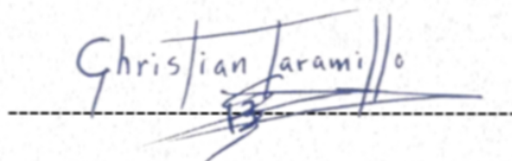
Ing. Carlos Israel Núñez Miranda, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE PACIENTES DE LA CLÍNICA VETERINARIA PET'S HOME DE LA CIUDAD DE AMBATO es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2025.

A handwritten signature in blue ink, reading "Christian Jaramillo", is written over a horizontal dashed line. Below the signature, there is a small, stylized mark that appears to be a combination of the letters "B" and "C".

Christian Marcelo Jaramillo Collazo

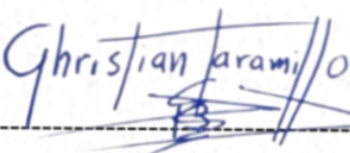
C.C. 1805155569

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2025.



Christian Marcelo Jaramillo Collazo

C.C. 1805155569

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Christian Marcelo Jaramillo Collazo, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE PACIENTES DE LA CLÍNICA VETERINARIA PET'S HOME DE LA CIUDAD DE AMBATO, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, febrero 2025.

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

PRESIDENTE DEL TRIBUNAL

Ing. Edison Álvarez Mayorga, Mg.

PROFESOR CALIFICADOR

Ing. Sandra Lucrecia Carrillo Ríos, Mg.

PROFESOR CALIFICADOR

DEDICATORIA

A mi amada familia, quienes han sido el pilar fundamental en mi vida y el motor de este logro. A mi novia e hija, por su infinita paciencia, apoyo incondicional y comprensión durante este desafiante proceso. En especial, dedico este trabajo a mi hija, quien con su sonrisa y alegría ha sido la mayor inspiración para perseverar y culminar este proyecto.

A mis padres; mi madre por ser un ejemplo constante de superación y esfuerzo, y mi padre, por ser un sabio guía que siempre me ha motivado a buscar el conocimiento y a enfrentar los retos con determinación. Este logro también es suyo.

AGRADECIMIENTO

Con profunda gratitud, expreso mi agradecimiento a todos aquellos que han sido parte de este importante capítulo de mi vida. A mi novia e hija, quienes entendieron mis ausencias y sacrificios para alcanzar este objetivo personal. A mis tíos y mi abuela, por su apoyo incondicional y por ser una fuente constante de fortaleza y motivación. A mi hermano, por su confianza y por creer siempre en mí.

A mis amigos, por confiar en mí y ser un sostén durante los momentos de duda. A mis compañeros de la universidad, quienes compartieron conmigo este viaje académico, brindándome apoyo incondicional.

A los profesores que, con generosidad y dedicación, compartieron su conocimiento, guiando mi aprendizaje y contribuyendo de manera significativa a mi formación profesional. Su enseñanza ha dejado una huella imborrable en mi vida.

A todos ustedes, gracias por ser parte de este sueño hecho realidad.

ÍNDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS	x
ÍNDICE DE FIGURAS	xiii
ÍNDICE DE ANEXOS	xv
RESUMEN EJECUTIVO	xvi
ABSTRACT	xvii
CAPÍTULO I. MARCO TEÓRICO	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema.....	1
1.2 Antecedentes investigativos	2
1.3 Fundamentación teórica	6
1.3.1 Metodología de desarrollo.....	6
1.3.2 Lenguajes de programación	8
1.3.3 Frameworks.....	9
1.3.4 Bases de datos	11
1.4 Objetivos	11
1.4.1 Objetivo general	11
1.4.2 Objetivos específicos	11
CAPÍTULO II. METODOLOGÍA	12
2.1 Materiales.....	12
2.2 Métodos.....	12

2.2.1 Modalidad de la investigación	12
2.2.2 Población y muestra	13
2.2.3 Recolección de información.....	13
2.2.4 Procesamiento y análisis de datos	14
CAPÍTULO III. RESULTADOS Y DISCUSIÓN	28
3.1 Análisis y discusión de los resultados	28
3.1.1 Modelo del proceso manual para la gestión de pacientes en la clínica veterinaria Pet's Home.....	29
3.1.2 Modelo propuesto del proceso con el uso de la aplicación móvil para la gestión de pacientes en la clínica veterinaria Pet's Home.....	30
3.1.3 Análisis comparativo y selección de la metodología de desarrollo	32
3.1.4 Análisis comparativo y selección de la tecnología de desarrollo.....	34
3.1.5 Análisis comparativo y selección de la base de datos.....	35
3.1.6 Diseño de la arquitectura del sistema.....	37
3.2 Desarrollo de la propuesta.....	38
3.2.1 Fase I: Planificación.....	40
3.2.2 Fase II: Diseño	62
3.2.3 Fase III: Codificación.....	87
3.2.4 Fase IV: Pruebas	100
3.2.5 Fase V: Lanzamiento	111
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	113
4.1 Conclusiones	113
4.2 Recomendaciones.....	113
REFERENCIAS BIBLIOGRÁFICAS	115
ANEXOS	117

ÍNDICE DE TABLAS

Tabla 1. Detalle del personal de la veterinaria Pet's Home.....	13
Tabla 2. Entrevista a la propietaria de Pet's Home.	14
Tabla 3. Detalle de resultados: Pregunta 1.....	18
Tabla 4. Detalle de resultados: Pregunta 2.....	19
Tabla 5. Detalle de resultados: Pregunta 3.....	20
Tabla 6. Detalle de resultados: Pregunta 4.....	21
Tabla 7. Detalle de resultados: Pregunta 5.....	22
Tabla 8. Detalle de resultados: Pregunta 6.....	23
Tabla 9. Detalle de resultados: Pregunta 7.....	24
Tabla 10. Detalle de resultados: Pregunta 8.....	25
Tabla 11. Detalle de resultados: Pregunta 9.....	26
Tabla 12. Detalle de resultados: Pregunta 10.....	27
Tabla 13. Alfa de Cronbach: rangos de confiabilidad.....	17
Tabla 14. Descripción del proceso manual.	29
Tabla 15. Descripción del proceso automatizado.	31
Tabla 16. Tabla comparativa de metodologías de desarrollo.....	32
Tabla 17. Tabla comparativa de tecnologías de desarrollo móvil.....	34
Tabla 18. Tabla comparativa de bases de datos.	36
Tabla 19. Roles del proyecto.....	40
Tabla 20. Backlog del producto.	41
Tabla 21. Modelo de una historia de usuario.	43
Tabla 22. Modelo de una tarea.....	44
Tabla 23. Historia de usuario N°1: Implementación de la base de datos.....	45
Tabla 24. Lista de tareas de la historia de usuario N°1.....	45
Tabla 25. Historia de usuario N°2: Implementación de una API.....	46
Tabla 26. Lista de tareas de la historia de usuario N°2.....	46
Tabla 27. Historia de usuario N°3: Gestión de datos de la clínica.....	47
Tabla 28. Lista de tareas de la historia de usuario N°3.....	47
Tabla 29. Historia de usuario N°4: Gestión de empleados.	48
Tabla 30. Lista de tareas de la historia de usuario N°4.....	48
Tabla 31. Historia de usuario N°5: Gestión de usuarios.	49

Tabla 32. Lista de tareas de la historia de usuario N°5.....	49
Tabla 33. Historia de usuario N°6: Autenticación de usuarios.....	50
Tabla 34. Lista de tareas de la historia de usuario N°6.....	50
Tabla 35. Historia de usuario N°7: Gestión de propietarios.	51
Tabla 36. Lista de tareas de la historia de usuario N°7.....	51
Tabla 37. Historia de usuario N°8: Gestión de pacientes.	52
Tabla 38. Lista de tareas de la historia de usuario N°8.....	52
Tabla 39. Historia de usuario N°9: Gestión de vacunas.	53
Tabla 40. Lista de tareas de la historia de usuario N°9.....	53
Tabla 41. Historia de usuario N°10: Gestión de desparasitaciones.	54
Tabla 42. Lista de tareas de la historia de usuario N°10.....	54
Tabla 43. Historia de usuario N°11: Gestión de citas médicas.....	55
Tabla 44. Lista de tareas de la historia de usuario N°11.....	55
Tabla 45. Historia de usuario N°12: Citas médicas programadas.....	56
Tabla 46. Lista de tareas de la historia de usuario N°12.....	56
Tabla 47. Historia de usuario N°13: Gestión de registros médicos.	57
Tabla 48. Lista de tareas de la historia de usuario N°13.....	57
Tabla 49. Historia de usuario N°14: Generación de reportes del historial clínico.....	58
Tabla 50. Lista de tareas de la historia de usuario N°14.....	58
Tabla 51. Estimación de tiempo de desarrollo.	59
Tabla 52. Plan de entregas de la aplicación.	62
Tabla 53. Modelo de una tarjeta CRC.....	63
Tabla 54. Tarjeta CRC de autenticación.....	63
Tabla 55. Tarjeta CRC de clínica.....	64
Tabla 56. Tarjeta CRC de pacientes.....	64
Tabla 57. Tarjeta CRC de empleados.	64
Tabla 58. Tarjeta CRC de cita médica.	65
Tabla 59. Tarjeta CRC de cita médica programada.	65
Tabla 60. Tarjeta CRC de registro clínico.	65
Tabla 61. Tarjeta CRC de reporte.	66
Tabla 62. Tarjeta CRC de vacuna.	66
Tabla 63. Tarjeta CRC de desparasitación.....	66
Tabla 64. Modelo de presentación de la prueba de aceptación.....	100

Tabla 65. Prueba de aceptación N°1.	101
Tabla 66. Prueba de aceptación N°2.	101
Tabla 67. Prueba de aceptación N°3.	102
Tabla 68. Prueba de aceptación N°4.	103
Tabla 69. Prueba de aceptación N°5.	103
Tabla 70. Prueba de aceptación N°6.	104
Tabla 71. Prueba de aceptación N°7.	105
Tabla 72. Prueba de aceptación N°8.	106
Tabla 73. Prueba de aceptación N°9.	106
Tabla 74. Prueba de aceptación N°10.	107
Tabla 75. Prueba de aceptación N°11.	108
Tabla 76. Prueba de aceptación N°12.	109
Tabla 77. Prueba de aceptación N°13.	109
Tabla 78. Prueba de aceptación N°14.	110

ÍNDICE DE FIGURAS

Figura 1. Frecuencias y varianzas de la encuesta aplicada.	16
Figura 2. Gráfico de resultados: Pregunta 1.....	18
Figura 3. Gráfico de resultados: Pregunta 2.....	19
Figura 4. Gráfico de resultados: Pregunta 3.....	20
Figura 5. Gráfico de resultados: Pregunta 4.....	21
Figura 6. Gráfico de resultados: Pregunta 5.....	22
Figura 7. Gráfico de resultados: Pregunta 6.....	23
Figura 8. Gráfico de resultados: Pregunta 7.....	24
Figura 9. Gráfico de resultados: Pregunta 8.....	25
Figura 10. Gráfico de resultados: Pregunta 9.....	26
Figura 11. Gráfico de resultados: Pregunta 10.....	27
Figura 12. Diagrama del proceso manual de gestión de pacientes.	29
Figura 13. Diagrama del proceso automatizado de gestión de pacientes.....	30
Figura 14. Arquitectura del sistema.	37
Figura 15. Modelo relacional de la base de datos.	67
Figura 16. Prototipo de la sección de datos de la clínica.	68
Figura 17. Prototipo de la sección de gestión de empleados.....	69
Figura 18. Prototipo del detalle del empleado.	69
Figura 19. Prototipo del formulario del empleado.	70
Figura 20. Prototipo del formulario de usuario anexo al empleado.	70
Figura 21. Prototipo de la pantalla de autenticación.	71
Figura 22. Prototipo de la pantalla de recuperación de contraseña.....	72
Figura 23. Prototipo de la sección de gestión de propietarios.....	73
Figura 24. Prototipo del detalle del propietario.	73
Figura 25. Prototipo del formulario del propietario.	74
Figura 26. Prototipo de la sección de gestión de pacientes.....	75
Figura 27. Prototipo del detalle del paciente.....	75
Figura 28. Prototipo del formulario del paciente.	76
Figura 29. Prototipo de la sección de gestión de vacunas.....	77
Figura 30. Prototipo del detalle de la vacuna.....	77
Figura 31. Prototipo del formulario de la vacuna.	78

Figura 32. Prototipo de la sección de gestión de desparasitaciones.....	79
Figura 33. Prototipo del detalle de la desparasitación.....	79
Figura 34. Prototipo del formulario de la desparasitación.	80
Figura 35. Prototipo de la sección de gestión de citas médicas.	81
Figura 36. Prototipo del detalle de la cita médica.....	81
Figura 37. Prototipo del formulario de la cita médica.	82
Figura 38. Prototipo de la pantalla principal con las citas programadas.....	83
Figura 39. Prototipo del mensaje predeterminado en Whatsapp.....	83
Figura 40. Prototipo de la sección de gestión de registros médicos.....	84
Figura 41. Prototipo del detalle del registro médico.....	85
Figura 42. Prototipo del formulario del registro médico.....	85
Figura 43. Prototipo de la sección de reportes clínicos.....	86
Figura 44. Prototipo de un reporte clínico.	87
Figura 45. Interfaz de la API en el servidor Somee.	88
Figura 46. Interfaz de gestión de datos de la clínica.....	89
Figura 47. Interfaz de la sección de empleados.	90
Figura 48. Interfaz de los detalles del empleado.....	90
Figura 49. Registro de un usuario acoplado al empleado.	91
Figura 50. Edición de las credenciales asignadas a un empleado.....	91
Figura 51. Autenticación de usuarios.	92
Figura 52. Recuperación de contraseña.	93
Figura 53. Registro de un propietario.	93
Figura 54. Sección de pacientes.....	94
Figura 55. Detalle del paciente.....	95
Figura 56. Sección de gestión de vacunas.....	95
Figura 57. Sección de gestión de desparasitaciones.....	96
Figura 58. Sección de gestión de citas médicas.	97
Figura 59. Sección de citas médicas programadas.....	98
Figura 60. Sección de gestión de registros médicos.	98
Figura 61. Detalle del registro médico.....	99
Figura 62. Ejemplo de reporte gráfico de un signo clínico.....	99

ÍNDICE DE ANEXOS

Anexo A. Formulario de la encuesta al personal de la clínica.	117
Anexo B. Levantamiento de requerimientos acorde al estándar IEEE 830	120
Anexo C. Tareas de la historia de usuario N°1.	142
Anexo D. Tareas de la historia de usuario N°2.	144
Anexo E. Tareas de la historia de usuario N°3.	145
Anexo F. Tareas de la historia de usuario N°4.	148
Anexo G. Tareas de la historia de usuario N°5.	151
Anexo H. Tareas de la historia de usuario N°6.	154
Anexo I. Tareas de la historia de usuario N°7.	156
Anexo J. Tareas de la historia de usuario N°8.	159
Anexo K. Tareas de la historia de usuario N°9.	162
Anexo L. Tareas de la historia de usuario N°10.	165
Anexo M. Tareas de la historia de usuario N°11.	168
Anexo N. Tareas de la historia de usuario N°12.	171
Anexo O. Tareas de la historia de usuario N°13.	173
Anexo P. Tareas de la historia de usuario N°14.	176

RESUMEN EJECUTIVO

La presente investigación aborda la implementación de una aplicación móvil para la gestión de pacientes en la clínica veterinaria Pet's Home, ubicada en la ciudad de Ambato, Ecuador. Ante la problemática de los registros manuales en papel, que generan dificultades de acceso, errores, duplicidad y pérdida de información, el proyecto busca optimizar la administración clínica mediante tecnología.

La aplicación móvil, diseñada en Android Studio con Java y respaldada por una API en .NET, permite registrar, consultar y actualizar información de manera centralizada y segura. Se empleó la metodología ágil Extreme Programming (XP) para asegurar entregas rápidas y adaptativas. Además, se seleccionó SQL Server como sistema gestor de base de datos por su rendimiento y compatibilidad.

Los resultados muestran que la digitalización asegura la integridad de los datos y facilita el acceso remoto, permitiendo a los veterinarios tomar decisiones informadas en tiempo real. Este sistema no solo aborda los desafíos operativos, sino que también mejora la calidad del servicio, promoviendo una atención más precisa y oportuna para las mascotas y sus propietarios. La investigación concluye que la transformación digital de los procesos clínicos en Pet's Home es un paso crucial hacia la modernización del sector veterinario en la región.

Palabras clave: API REST, aplicación móvil, desarrollo de software, gestión de pacientes, optimización de procesos, veterinaria.

ABSTRACT

The present research addresses the implementation of a mobile application for patient management at the Pet's Home veterinary clinic, located in Ambato, Ecuador. Faced with the challenges of manual paper records, which lead to difficulties in access, errors, duplication, and loss of information, the project aims to optimize clinical administration through technology.

The mobile application, designed in Android Studio using Java and supported by a .NET API, enables centralized and secure registration, consultation, and updating of information. The agile methodology Extreme Programming (XP) was employed to ensure rapid and adaptive deliveries. Additionally, SQL Server was selected as the database management system for its performance and compatibility.

The results demonstrate that digitization ensures data integrity and facilitates remote access, allowing veterinarians to make informed decisions in real-time. This system not only addresses operational challenges but also enhances service quality, promoting more precise and timely care for pets and their owners. The research concludes that the digital transformation of clinical processes at Pet's Home is a crucial step toward modernizing the veterinary sector in the region.

Keywords: REST API, mobile application, software development, patient management, process optimization, veterinary medicine.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE PACIENTES DE LA CLÍNICA VETERINARIA PET'S HOME DE LA CIUDAD DE AMBATO.

1.1.1 Planteamiento del problema

La creciente preocupación por el cuidado de las mascotas y animales domésticos en Ecuador se ha vuelto una prioridad cada vez más evidente para numerosas familias conscientes de la importancia del bienestar animal. En consonancia con este aumento de la conciencia social, las clínicas veterinarias desempeñan un papel crucial en el tratamiento y cuidado de las mascotas siendo lugares clave para asegurar su salud y bienestar [1] [2]. No obstante, surge un problema recurrente en este escenario: el manejo deficiente de los registros de los pacientes plantea una serie de desafíos y dificultades tanto para los profesionales veterinarios como para los dueños de mascotas, generando una brecha en la calidad de la atención veterinaria [3].

En la ciudad de Ambato la presencia de sistemas ineficientes y desactualizados para la gestión de registros de pacientes en muchas clínicas veterinarias se convierte en un obstáculo significativo. La prevalencia de registros en papel en los establecimientos añade una capa adicional de complejidad a la situación, dificultando el acceso y la actualización de información crucial [4]. La conformación de una base de datos centralizada y un sistema de seguimiento adecuado busca optimizar la tarea de seguimiento completo de la historia médica de los animales, abarcando tratamientos anteriores, vacunas, alergias y otras condiciones médicas de importancia.

El manejo deficiente de los registros de los pacientes en la clínica veterinaria Pet's Home de la ciudad de Ambato no solo constituye un desafío operativo, sino que también tiene consecuencias negativas palpables tanto para los profesionales veterinarios como para los clientes.

Entre estas consecuencias se incluyen la falta de información completa, la pérdida de tiempo y recursos, así como la dificultad en la coordinación de cuidados, afectando directamente la calidad y continuidad en la atención de los animales. Esta situación puede traducirse en un desgaste en la relación entre los veterinarios y los dueños de las mascotas comprometiendo la confiabilidad del establecimiento médico.

La problemática mencionada representa un desafío sustancial que requiere atención inmediata, por lo que se propone la implementación de un sistema de gestión de registros centralizado como una necesidad imperante para permitir a los profesionales veterinarios acceder fácilmente a información completa y actualizada de los animales. A su vez, se busca mejorar la calidad de la atención ofrecida y contribuir a reducir el tiempo y los recursos desperdiciados, aumentando así la satisfacción de los clientes y solidificando un enfoque más efectivo y completo del cuidado animal en la región.

1.2 Antecedentes investigativos

En [3] se implementa un sistema de gestión veterinaria usando el framework ReactJS mediante el modelo de desarrollo Api Rest. Se usa la metodología XP indicando que es adecuado para un grupo de trabajo pequeño, permite el desarrollo ágil y se optimiza el proceso de generación de documentación. Otras tecnologías que complementaron el proceso son PHP, Laravel, CSS, Javascript y Apache. Como resultado se obtuvo un aplicativo con roles, registro de usuarios, registro de productos médicos, gestión de historias clínicas, tratamientos, diagnósticos y vacunas, y generación de reportes.

En [4] se desarrolló un sistema de gestión para la clínica veterinaria de la Universidad Central del Ecuador, mismo que gestiona las facturas, los clientes, registros médicos de los pacientes y genera reportes de transacciones y estadísticas de facturación. El sistema se desarrolló netamente en Java, con herramientas afines como Enterprise Java Beans para el lado del servidor, Java Transaction API para el manejo de transacciones y Java Server Faces como framework para simplificar la generación de interfaces. Como resultado se obtuvo un sistema informático a partir de software libre para permitir que la clínica veterinaria gestione datos de manera centralizada.

En [5] se desarrolló una aplicación web para la gestión de procesos administrativos y el manejo de la información en la clínica Mundo Animal, reduciendo los tiempos de espera y brindando un mejor trato a los clientes. Se empleó un enfoque iterativo en la ejecución del trabajo, lo cual permitió la validación de cada módulo con el objetivo de alcanzar mayor estabilidad en el software. La aplicación fue desarrollada utilizando tecnologías como HTML, CSS, JavaScript para el frontend, y PHP para el backend. La base de datos se implementó en MySQL, y se utilizó el framework Laravel para el desarrollo, mientras que la aplicación se desplegó en un servidor Apache. Los resultados obtenidos mostraron la reducción de tiempos de espera y un servicio más eficiente en base a calificaciones de los clientes. Además, se identificaron futuras funcionalidades como anexos con el SRI, pagos online, consulta de información para clientes y alertas de caducidad de productos, lo que sugiere un potencial de mejora continua y escalabilidad de la aplicación.

En [6] el objetivo del proyecto es desarrollar una aplicación web para la gestión veterinaria en el Aquarium Finisterrae de A Coruña. La aplicación permitirá a los usuarios añadir notas clínicas, tratamientos y otros informes médicos asociados a las especies del acuario, así como gestionar el inventario de medicamentos disponibles. Para la implementación de la aplicación, se utilizaron diversas tecnologías como PostgreSQL para la persistencia de datos, Java con el framework Spring para el backend, JavaScript con el framework React para el frontend, y Docker para el despliegue de la aplicación. Durante el desarrollo, se siguió la metodología ágil Scrum, estableciendo sprints de 15 días. Como resultado se obtuvo una aplicación web funcional que cumple con los objetivos planteados, permitiendo una gestión eficiente de la información veterinaria en el acuario.

En [7] el objetivo del trabajo es integrar el software de gestión veterinaria ZooVet con la herramienta de gestión de reservas online Bookitit para mejorar la eficiencia de la gestión de agendas en los centros veterinarios. La integración implica analizar la documentación de ambas plataformas e implementar la comunicación bidireccional entre ellas. Se utilizó la metodología en cascada, enfoque secuencial para desarrollar software con fases sucesivas: requisitos, diseño, implementación, pruebas y mantenimiento.

Las tecnologías utilizadas en este proyecto incluyen API REST, biblioteca CURL para conexión HTTP, C++, Visual Studio, JSON y HTTP. La integración de ZooVet y Bookitit permitió optimizar el tiempo en la gestión de agendas de los centros veterinarios y obtuvo una buena aceptación por parte de los usuarios.

En [8] se presenta el desarrollo e implementación de una aplicación móvil llamada Gapp para el seguimiento eficiente del ganado bovino en la Hacienda San Gabriel en Ecuador. La aplicación se llevó a cabo utilizando tecnologías móviles, IoT (Internet de las cosas) y bajo la metodología de la arquitectura limpia. Se utilizaron lenguajes de programación como Java para Android y Swift para iOS, así como bases de datos MySQL para el almacenamiento de datos. Los resultados demostraron que la implementación de la aplicación Gapp cumple con la gestión completo del historial clínico y producción del ganado bovino, permitiendo generar reportes, inventarios y la consulta de las actividades realizadas por los trabajadores.

En [9] el objetivo es mejorar la gestión de ventas en la clínica veterinaria Janavet en Trujillo, Perú, a través de la implementación de una aplicación móvil multiplataforma. Para lograrlo se utilizó la metodología Extreme Programming para el desarrollo del software, y se diseñaron distintas interfaces utilizando Flutter. Además, se implementó el patrón de diseño BLOC para la arquitectura de la aplicación. Los resultados del estudio mostraron mejoras significativas en el nivel de servicio (+3,45%), la ganancia neta (+40,03%), el promedio de venta por cliente (+3,42%) y el número de clientes potenciales (+53,08%) después de la implementación de la aplicación.

En [10] se presenta la implementación de un sistema web para controlar los procesos de la clínica veterinaria Mascotas, permitiendo un acceso eficiente a la información de los pacientes y mejorando la calidad del servicio. Se aplicó una metodología de desarrollo iterativo que permitió obtener versiones del producto software antes de la entrega final, lo que facilitó la corrección de posibles errores y la implementación de cambios. Se utilizaron diversas tecnologías, incluyendo el lenguaje de programación JavaScript, el framework ZK para las interfaces de usuario, el servidor web Apache Tomcat para la conexión entre el usuario y la aplicación, y MySQL Server como sistema gestor de base de datos.

Como resultado se presentó una arquitectura de aplicación web basada en el modelo cliente-servidor, se implementaron mecanismos de seguridad como el control de sesiones y la encriptación de claves, y se demostró la rentabilidad del proyecto a través de un estudio de factibilidad.

En [11] se llevó a cabo una investigación en Ecuador con el objetivo de facilitar el cuidado de mascotas mediante una aplicación web y móvil que permitiera encontrar los mejores centros veterinarios de manera ágil y rápida. Este estudio aporta técnicas adecuadas para definir la población y muestra de investigación, así como para la aplicación de instrumentos de recolección de información como encuestas y entrevistas. Los resultados de cada ítem fueron presentados mediante gráficos estadísticos, acompañados de un análisis individual. Por su parte, el autor destaca la importancia de cumplir con los requisitos mínimos del dispositivo móvil para evitar problemas de funcionamiento del producto final.

En [12] se presenta una solución integrada en una aplicación web de gestión administrativa para clínicas veterinarias, con el fin de mejorar los procesos administrativos y facilitar el trabajo de los empleados. El trabajo se desarrolló bajo un proceso iterativo, lo que permitió una depuración y validación progresiva para obtener un software más estable. Se utilizaron diversas tecnologías de código abierto, incluyendo Python como lenguaje de programación, el framework Django para el desarrollo web, HTML5, CSS3 y JavaScript con la librería jQuery para el FrontEnd, y PostgreSQL como gestor de base de datos. El entorno de desarrollo integrado (IDE) PyCharm se empleó para la programación, y Visual Studio Code para el diseño de las pantallas de la aplicación. Los resultados obtenidos muestran una reducción del tiempo de atención a las mascotas, específicamente en el acceso a la información, la búsqueda de registros y en la gestión de expedientes clínicos.

Según [13] se llevó a cabo un proyecto con el propósito de crear una plataforma modular accesible tanto por web como por una aplicación móvil. Esta plataforma tiene como objetivo gestionar información crítica para clínicas veterinarias. Se usó la metodología RUP para gestionar el proyecto, con lo cual se estudió los sistemas existentes en el mercado para definir un conjunto de funcionalidades para construir una solución viable.

La investigación aporta valiosa información sobre estándares, prácticas y métricas de calidad en el desarrollo de aplicaciones móviles, así como la gestión de recursos, presupuestos y riesgos.

En [14] se llevó a cabo una aplicación móvil para la gestión de servicios de una clínica veterinaria. Se destaca la centralización de los registros clínicos, consiguiendo que se genere un historial médico independiente al centro veterinario visitado. Se usó la metodología Mobile-D que cuenta con cinco fases y prioriza la funcionalidad integral del sistema. La información se recolectó aplicando entrevistas que permitieran conocer los procesos de la atención veterinaria y el registro de los pacientes.

En [15] se realizó una aplicación multiplataforma para la gestión de pacientes y citas médicas en una clínica veterinaria en la ciudad de Durán. Además de una entrevista y una encuesta, se usó una ficha de observación para limitar el alcance del sistema informático. La metodología de trabajo XP o Extreme Programming agilizó el proceso de desarrollo al concentrarse en tareas específicas sin distracciones. El diseño arquitectónico se llevó a cabo con diagramas UML. En cuanto a las herramientas usadas tenemos PHP, JavaScript, Dart y Flutter.

1.3 Fundamentación teórica

La fundamentación teórica constituye el cimiento conceptual sobre el cual se establece el estudio para el "Sistema de Gestión de Recomendaciones de la Contraloría (CGE) para el GAD Municipalidad de Ambato".

1.3.1 Metodología de desarrollo

a. Extreme Programming (XP)

Extreme Programming (XP) es una metodología de desarrollo ágil que se centra en la entrega rápida de software de alta calidad. En [3, 9, 15] se emplea para el desarrollo de aplicaciones veterinarias. Los autores eligieron esta metodología por la iteración rápida, la retroalimentación continua y la atención a la calidad del código, lo que permite a los equipos de desarrollo adaptarse de manera flexible a los cambios en los requisitos del proyecto y garantizar la satisfacción del cliente.

b. Scrum

Scrum es una metodología ágil que se centra en la entrega iterativa e incremental de software. En [6] se sigue la metodología ágil Scrum para el desarrollo de la aplicación de gestión veterinaria en el Aquarium Finisterrae de A Coruña. Se menciona que Scrum se caracteriza por la división del trabajo en sprints cortos, la colaboración estrecha entre los miembros del equipo y la revisión continua del progreso del proyecto. La implementación de Scrum permite una gestión eficiente del desarrollo de software, facilitando la identificación temprana de problemas y la adaptación a los cambios en los requisitos del cliente.

c. Rational Unified Process (RUP)

Rational Unified Process (RUP) es un enfoque de desarrollo de software que se centra en la iteración, la colaboración y la calidad del producto final. En [13] se lleva a cabo un proyecto utilizando la metodología RUP para crear una plataforma modular para clínicas veterinarias. El autor indica que RUP se caracteriza por su enfoque en la gestión de riesgos, la definición clara de roles y responsabilidades, y la orientación hacia la calidad del producto. La implementación de RUP permite una planificación y gestión estructuradas del proyecto, asegurando la entrega de un producto final que cumpla con los requisitos del cliente y las expectativas de calidad.

d. Mobile-D

Mobile-D es una metodología de desarrollo centrada en el diseño de aplicaciones móviles que prioriza la funcionalidad integral del sistema. En [14] se lleva a cabo una aplicación móvil para la gestión de servicios de una clínica veterinaria utilizando la metodología Mobile-D. Según el autor, Mobile-D se caracteriza por su enfoque en la experiencia del usuario, la simplicidad en el diseño y la adaptabilidad a las plataformas móviles. La implementación de Mobile-D permite una rápida iteración en el diseño y desarrollo de la aplicación, asegurando que se satisfagan las necesidades y expectativas de los usuarios finales.

1.3.2 Lenguajes de programación

a. Java

Java es un lenguaje de programación ampliamente utilizado en el desarrollo de aplicaciones empresariales y web debido a su portabilidad y robustez. En los proyectos analizados, Java se empleó tanto en el backend de aplicaciones web como en el desarrollo de aplicaciones móviles. En [4] se utilizó Java junto con Java Server Faces (JSF) para simplificar la generación de interfaces en el sistema de gestión veterinaria de la Universidad Central del Ecuador.

En [6] se usó Java con el framework Spring para el backend de la aplicación de gestión veterinaria en el Aquarium Finisterrae de A Coruña. Los autores coinciden en que la utilización de Java garantiza un alto nivel de escalabilidad y rendimiento, lo que lo convierte en una opción sólida para sistemas complejos como los de gestión veterinaria.

b. PHP

PHP es un lenguaje de programación popular en el desarrollo web, especialmente para la creación de aplicaciones dinámicas. En los proyectos analizados PHP se utilizó principalmente en el backend. En [5] se empleó PHP junto con el framework Laravel para desarrollar una aplicación web para la gestión administrativa en la clínica Mundo Animal. En [7] PHP se utilizó para integrar el software de gestión veterinaria ZooVet con la herramienta de gestión de reservas online Bookitit.

c. JavaScript

JavaScript es un lenguaje de programación fundamental para el desarrollo de software. En los proyectos analizados, JavaScript se utilizó para el desarrollo frontend de las aplicaciones web y móviles. En [5] se empleó JavaScript para el frontend de la aplicación web en la clínica Mundo Animal, mientras que en [8] se utilizó para el frontend de la aplicación móvil Gapp para el seguimiento del ganado bovino. La versatilidad y la capacidad de JavaScript para trabajar en múltiples plataformas lo convierten en una opción imprescindible para el desarrollo moderno de aplicaciones.

d. Swift

Swift es un lenguaje de programación desarrollado por Apple para crear aplicaciones para iOS y macOS. En el proyecto [8] Swift se utilizó para desarrollar la aplicación móvil Gapp para el seguimiento del ganado bovino en la Hacienda San Gabriel en Ecuador. Según el autor Swift ofrece un rendimiento superior y una sintaxis más clara en comparación con Objective-C, lo que lo convierte en la opción preferida para el desarrollo de aplicaciones nativas en el ecosistema de Apple.

e. Dart

Dart es un lenguaje de programación desarrollado por Google y utilizado para construir aplicaciones multiplataforma, especialmente en el framework Flutter. En el proyecto [15] Dart se empleó para desarrollar una aplicación multiplataforma para la gestión de pacientes y citas médicas en una clínica veterinaria en Durán. En el proyecto se detalla que Dart proporciona una sintaxis concisa y fácil de entender, junto con un conjunto de herramientas robustas que facilitan el desarrollo de aplicaciones para diferentes plataformas. Su integración con Flutter permite la creación eficiente de interfaces de usuario atractivas y funcionales.

1.3.3 Frameworks

a. ReactJS

ReactJS es un framework de JavaScript utilizado para construir interfaces de usuario interactivas y reutilizables. En [3] se implementó un sistema de gestión veterinaria utilizando ReactJS, aprovechando su capacidad para crear componentes modulares y dinámicos. Además, en [6] React se utilizó para el frontend de la aplicación de gestión veterinaria en el Aquarium Finisterrae de A Coruña. La arquitectura basada en componentes de React permite un desarrollo ágil y escalable, facilitando la creación de interfaces de usuario intuitivas y responsivas.

b. Laravel

Laravel es un framework de PHP que proporciona herramientas y estructuras para el desarrollo rápido de aplicaciones web.

En [3] se complementó el sistema de gestión veterinaria con Laravel, aprovechando su elegante sintaxis y su amplia gama de características integradas. Además, en [5] el framework Laravel se utilizó para el desarrollo de la aplicación web en la clínica Mundo Animal. Laravel facilita tareas comunes como la gestión de bases de datos, la autenticación de usuarios y el enrutamiento, lo que permite a los desarrolladores centrarse en la lógica de la aplicación.

c. Spring

Spring Framework es un framework de Java ampliamente utilizado para el desarrollo de aplicaciones empresariales. En [6] se empleó Spring para el backend de la aplicación de gestión veterinaria en el Aquarium Finisterrae de A Coruña. Según este trabajo Spring ofrece un enfoque integral para el desarrollo de aplicaciones Java, proporcionando soporte para la inyección de dependencias, la gestión transaccional y el desarrollo basado en aspectos. Su modularidad y escalabilidad lo convierten en una opción popular para aplicaciones de gran envergadura.

d. Java Server Faces (JSF)

Java Server Faces (JSF) es un framework de componentes para la construcción de interfaces de usuario en aplicaciones web Java. En [4] JSF se utilizó como framework para simplificar la generación de interfaces en el sistema de gestión veterinaria de la Universidad Central del Ecuador. En el proyecto se menciona que JSF proporciona un conjunto de componentes reutilizables y una arquitectura basada en eventos que facilita el desarrollo de aplicaciones web escalables y mantenibles.

e. Flutter

Flutter es un framework de desarrollo de aplicaciones móviles multiplataforma desarrollado por Google. En [9, 15] Flutter se utilizó para implementar aplicaciones móviles multiplataforma en clínicas veterinarias en Perú y Ecuador, respectivamente. Ambos autores coinciden en que Flutter permite a los desarrolladores crear interfaces de usuario atractivas y de alto rendimiento que se ejecutan de manera nativa en iOS y Android. Su enfoque en el desarrollo rápido y la experiencia de usuario consistente lo convierten en una opción popular para proyectos móviles modernos.

1.3.4 Bases de datos

a. MySQL

MySQL es un sistema de gestión de bases de datos relacional ampliamente utilizado en el desarrollo de aplicaciones. En [5] esta se implementa para la aplicación web en la clínica Mundo Animal, lo que proporciona un almacenamiento eficiente y seguro de la información. En [10] MySQL Server se usa como sistema gestor de base de datos en el sistema web de control de procesos de la clínica veterinaria Mascotas, asegurando la integridad y disponibilidad de datos clínicos y administrativos.

b. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional y de código abierto. En [6] PostgreSQL se utiliza para la persistencia de datos en la aplicación de gestión veterinaria en el Aquarium Finisterrae de A Coruña. Se indica que la robustez y la capacidad de escalabilidad de PostgreSQL lo convierten en una opción sólida para aplicaciones que requieren un manejo eficiente de grandes volúmenes de datos.

1.4 Objetivos

1.4.1 Objetivo general

Implementar una aplicación móvil para la gestión de pacientes en la Veterinaria Pet's Home de Ambato.

1.4.2 Objetivos específicos

- Seleccionar una metodología de desarrollo de software para llevar a cabo el desarrollo de la aplicación.
- Recolectar los requerimientos para asegurar que la aplicación cumpla con las necesidades de la Veterinaria Pet's Home.
- Desarrollar una aplicación móvil que permita la gestión eficiente de historiales clínicos, citas y tratamientos médicos.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Para llevar a cabo el proyecto, se aplica una entrevista y una encuesta como elementos clave en la investigación. La entrevista fue realizada con la propietaria de la veterinaria, proporcionando una visión detallada de los procesos y opiniones internas cruciales para el progreso del proyecto.

Por otro lado, la encuesta se aplica a todo el personal, lo que permite obtener datos cuantitativos sobre sus experiencias, necesidades y puntos de vista. Ambas metodologías, en conjunto, ofrecieron una comprensión integral y equilibrada de los aspectos técnicos y de gestión que influyen en la efectividad y el éxito del proyecto.

2.2 Métodos

2.2.1 Modalidad de la investigación

La presente investigación se fundamenta en dos modalidades de investigación esenciales: de campo y bibliográfica.

a. Modalidad de campo

La modalidad de campo se aplica al recolectar información directamente en el lugar donde se está generando el problema de investigación, permitiendo analizar y obtener la situación actual que encamine el desarrollo de una solución tecnológica para gestionar la información de los pacientes de la clínica veterinaria en cuestión [16].

b. Modalidad bibliográfica

Se considera la modalidad bibliográfica al recurrir a documentos de naturaleza científica como libros, artículos científicos y tesis de universidades para contextualizar el problema de investigación y sus respectivos antecedentes [17].

2.2.2 Población y muestra

La población está conformada por todos los miembros del personal de la clínica veterinaria Pet's Home. El grupo de trabajo está conformado por 7 miembros (Tabla 1). Debido al caso de estudio actual en donde tenemos una población reducida, se utilizará toda la población sin derivar el cálculo de una muestra.

Tabla 1. Detalle del personal de la veterinaria Pet's Home.

Nombre	Cantidad
Veterinario(a) principal	2
Auxiliar veterinario	1
Veterinario(a) residente	1
Pasantes / Estudiantes de Medicina Veterinaria	3
Total	7

2.2.3 Recolección de información

Para la recolección de la información se utilizará una entrevista orientada a conocer la situación actual en cuanto al manejo de información en la clínica veterinaria, a la vez que se orienta el levantamiento de requerimientos. Adicional a esto se realizará una encuesta que estará conformada por preguntas en escala de Likert. Esta encuesta será validada mediante el cálculo del Alfa de Cronbach para definir su índice de confiabilidad.

2.2.4 Procesamiento y análisis de datos

a. Entrevista

Con la finalidad de contextualizar la problemática y encaminar el desarrollo de la solución en la clínica veterinaria Pet's Home se aplicó una entrevista a la propietaria del establecimiento. En la Tabla 2 se detallan las preguntas, una síntesis de las respuestas y observaciones de la entrevista.

Tabla 2. Entrevista a la propietaria de Pet's Home.

Nº	Pregunta	Respuesta	Observación
1	¿Cuál es el proceso actual para registrar a un nuevo paciente en la clínica?	Un miembro del personal se encarga de llenar dos fichas de información: una con los datos informativos de la mascota y el propietario y otra referente a los hábitos y cuidados de la mascota en el hogar. Los documentos son almacenados en archivadores.	Registro manual de información. Almacenamiento sin orden o clasificación específicos.
2	¿Qué datos se registran en las fichas mencionadas?	Datos de la mascota: nombre, especie, sexo, raza, edad, peso, hábitos, cuidados, dieta, enfermedades y cirugías previas, esterilizado, número de partos. Datos del propietario: nombre, cédula, teléfono, dirección, correo.	Integrar los datos en formularios con opciones predefinidas y atajos para un fácil registro.
3	¿Se registran los datos referentes a signos clínicos? ¿Cuáles son estos datos?	No. Los signos clínicos son tomados para un diagnóstico inmediato únicamente. Los datos clínicos son: temperatura corporal, frecuencia cardíaca, frecuencia respiratoria, estado de hidratación, peso corporal, tiempo de llenado capilar, evaluación de la piel, pelaje y movilidad.	Crear una sección de historial de información médica para que el médico pueda definir los antecedentes o la evolución del paciente para la toma de decisiones.

4	¿Qué otros datos se manejan de los pacientes?	Carnets de vacunación, historial de desparasitaciones, resultados de exámenes y citas programadas.	La base de datos debe integrar todos los datos mencionados.
5	¿Cuáles son las dificultades con este proceso de registro?	Llenar las fichas toma un tiempo considerable. Cuando un paciente no es nuevo, es difícil encontrar su registro anterior. También existen registros duplicados e incompletos.	Difícil acceso a los datos de los pacientes.
6	¿Qué mejoras espera con la implementación de un sistema de registro y control de pacientes?	Esperamos tener toda la información centralizada en un mismo lugar: datos, carnets, exámenes. De esta forma, podremos brindar una mejor atención a nuestros clientes.	La centralización de datos es fundamental.
7	¿Qué características serían esenciales en esta aplicación de gestión?	Guardar los datos del propietario, los datos del paciente, el registro de vacunas y desparasitaciones, las fechas de las próximas consultas, medicamentos recetados y exámenes realizados anteriormente.	Presentar la información obtenida como requerimientos del sistema.
8	¿La asignación de funcionalidades en la aplicación de acorde al rango del personal sería beneficioso?	Podría ser beneficioso, ya que se podría limitar el acceso a información restringida al personal secundario o temporal de la clínica.	Implementación de roles en la aplicación.
9	¿Qué herramientas móviles usa actualmente para su labor en la clínica veterinaria?	Únicamente calculadoras para dosificación de medicamentos.	Existe cierta familiaridad con aplicaciones móviles de uso veterinario.
10	¿La integración de las funcionalidades mencionadas en una aplicación móvil mejoraría la atención en la clínica?	Sí. De esta forma tendríamos todo lo necesario para optimizar la atención de los pacientes y brindar un mejor servicio a nuestros clientes.	

Al finalizar la entrevista se concluye que el método actual de gestión de registros de los pacientes de la clínica veterinaria Pet's Home presenta dificultades al personal médico. La implementación de un sistema centralizado de gestión de información deberá atender a los requerimientos mencionados previamente con el fin de cubrir sus necesidades para brindar un mejor servicio.

b. Encuesta

Se aplicó una encuesta al personal de la clínica veterinaria Pet's home conformado por 7 personas. En el Anexo A se muestra el formulario que se aplicó.

- **Cálculo del Alfa de Cronbach**

El cálculo del Alfa de Cronbach permite definir el índice de confiabilidad de una encuesta y si los resultados se encuentran sesgados. En primer lugar, se calcula la varianza de las respuestas de cada pregunta como se muestra en la Figura 11.

Encuestado	ITEMS										Suma
	1	2	3	4	5	6	7	8	9	10	
1	4	5	3	4	3	5	5	4	5	5	43
2	4	3	3	3	4	5	5	5	5	3	40
3	5	5	4	5	4	5	5	5	5	5	48
4	4	4	3	3	5	3	3	4	4	3	36
5	4	4	4	4	3	5	5	4	5	5	43
6	5	5	4	5	5	5	3	5	5	5	47
7	5	5	3	4	5	3	5	5	5	5	45
Varianza	0,24489796	0,53061224	0,24489796	0,57142857	0,69387755	0,81632653	0,81632653	0,24489796	0,12244898	0,81632653	
Sumatoria varianzas	5,10204082		Varianza de la suma de los ítems	14,6938776		Número de ítems	10				

Figura 1. Frecuencias y varianzas de la encuesta aplicada.

Luego, aplicamos la siguiente fórmula:

$$\alpha = \frac{K}{K-1} * \left[1 - \frac{\sum Vi}{Vt} \right]$$

Donde:

K es el número de ítems: 10.

$\sum Vi$ es la sumatoria total de la varianza de cada ítem o pregunta.

Vt es la varianza total.

$$\alpha = \frac{10}{10 - 1} * \left[1 - \frac{5,1020}{14,6938} \right] = 0,7253$$

El valor obtenido del Alfa de Cronbach se traduce como se indica en la Tabla 13.

Tabla 3. Alfa de Cronbach: rangos de confiabilidad.

Rango	Confiabilidad
< 0,54	Confiabilidad nula
0,54 – 0,59	Confiabilidad baja
0,60 – 0,65	Confiable
0,66 – 0,71	Muy confiable
0,72 – 0,99	Excelente confiabilidad
1	Confiabilidad absoluta

Al obtener 0,72 se deduce que las preguntas del cuestionario tienen una correlación, confirmando un instrumento de evaluación con alta confiabilidad.

- **Resultados e interpretación por pregunta**

1. **¿Con qué frecuencia experimenta dificultades al acceder o encontrar información en los registros clínicos de la veterinaria?**

Tabla 4. Detalle de resultados: Pregunta 1.

Indicador	Frecuencia	Porcentaje
Nunca	0	0,00%
Raramente	0	0,00%
Ocasionalmente	0	0,00%
Frecuentemente	4	57,14%
Siempre	3	42,86%
Total	7	100%

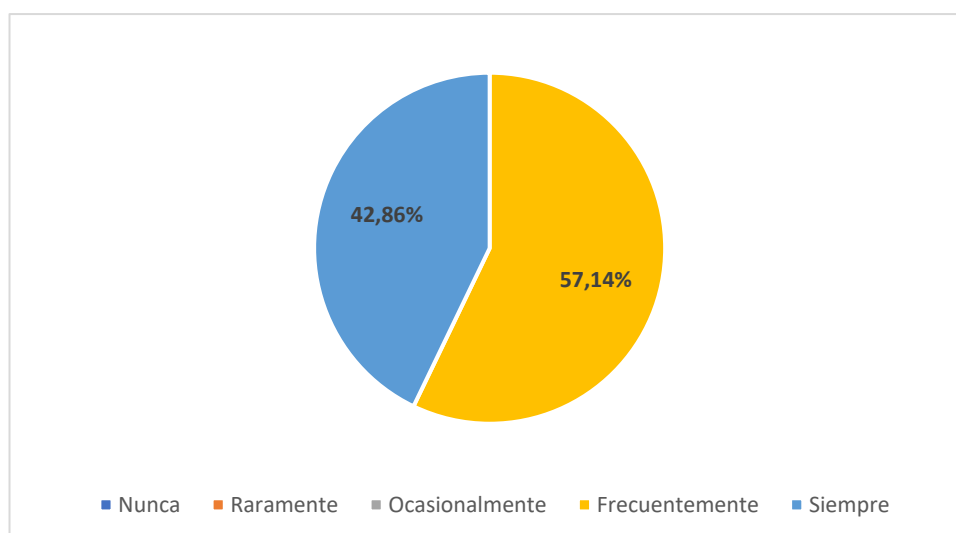


Figura 2. Gráfico de resultados: Pregunta 1.

Los resultados obtenidos en la pregunta 1 indican que el 57,14% de la población experimenta frecuentemente dificultades para acceder a la información de los registros clínicos, mientras que el 42,86% restante ha tenido este problema siempre. Para entender este aspecto, cabe recalcar que los registros actuales se llevan en papel y se redactan a mano, por lo que es complejo acceder a información específica o conlleva mucho tiempo.

2. ¿Cuánto tiempo dedica actualmente a generar el registro clínico de un paciente?

Tabla 5. Detalle de resultados: Pregunta 2.

Indicador	Frecuencia	Porcentaje
1-2 minutos	0	0,00%
3-4 minutos	0	0,00%
5 minutos	1	14,29%
6-10 minutos	2	28,57%
Más de 10 minutos	4	57,14%
Total	7	100%

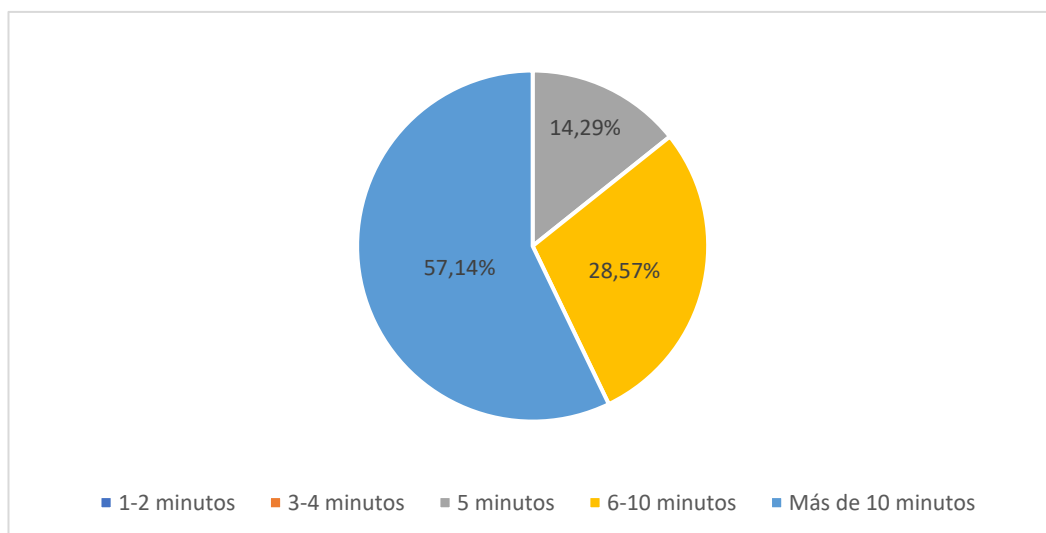


Figura 3. Gráfico de resultados: Pregunta 2.

El 57,14% de la población tarda más de 10 minutos en generar un registro clínico completo de un paciente nuevo en la clínica veterinaria, mientras que el 28,57% tarda de 6 a 10 minutos y el 14,29% aproximadamente 5 minutos.

Generar rápidamente un registro médico es fundamental para garantizar una atención eficiente y precisa. La sistematización de este aspecto optimiza los procesos administrativos, mejora la comunicación con los dueños de las mascotas y asegura un seguimiento adecuado de la salud del paciente.

3. ¿Qué tan satisfecho está con la eficiencia actual de la gestión de registros clínicos en la clínica veterinaria?

Tabla 6. Detalle de resultados: Pregunta 3.

Indicador	Frecuencia	Porcentaje
Muy insatisfecho	0	0,00%
Insatisfecho	0	0,00%
Neutral	4	57,14%
Satisfecho	3	42,86%
Muy satisfecho	0	0,00%
Total	7	100%

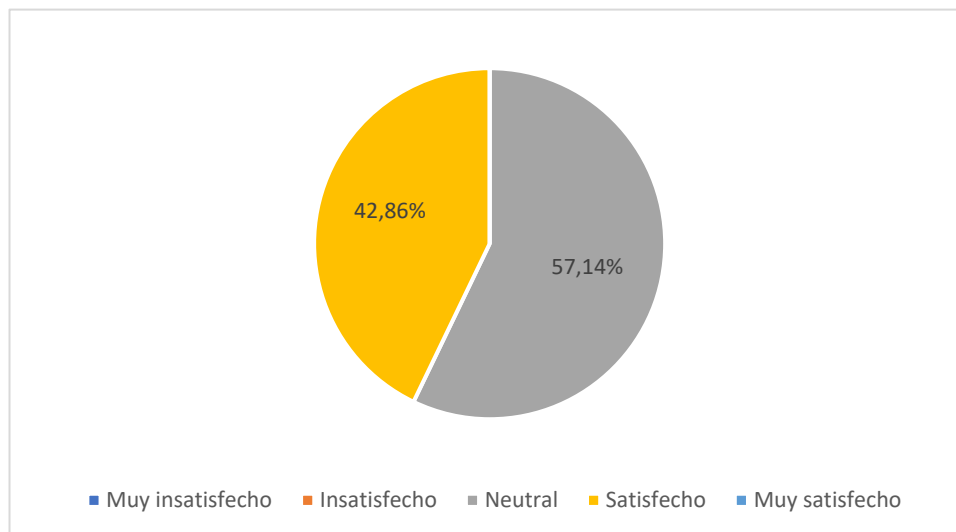


Figura 4. Gráfico de resultados: Pregunta 3.

El 42,86% de los encuestados se muestran insatisfechos con la actual gestión de registros clínicos, mientras que un 57,14% mantiene una opinión neutral. Un sistema de gestión de registros clínicos veterinarios mejoraría la eficiencia al centralizar y organizar de manera digital toda la información relevante de los pacientes.

Este tipo de sistema permite a los profesionales veterinarios acceder de forma rápida y segura a historiales médicos, tratamientos previos, vacunaciones y datos del propietario, optimizando los tiempos de atención y reduciendo errores en el manejo de la información, además, facilita el seguimiento continuo de la salud de las mascotas y agiliza la toma de decisiones clínicas.

4. ¿Encuentra que el manejo manual de registros clínicos afecta la calidad del servicio que se brinda a los clientes y pacientes?

Tabla 7. Detalle de resultados: Pregunta 4.

Indicador	Frecuencia	Porcentaje
Totalmente en desacuerdo	0	0,00%
En desacuerdo	0	0,00%
Neutral	2	28,57%
De acuerdo	3	42,86%
Totalmente de acuerdo	2	28,57%
Total	7	100%

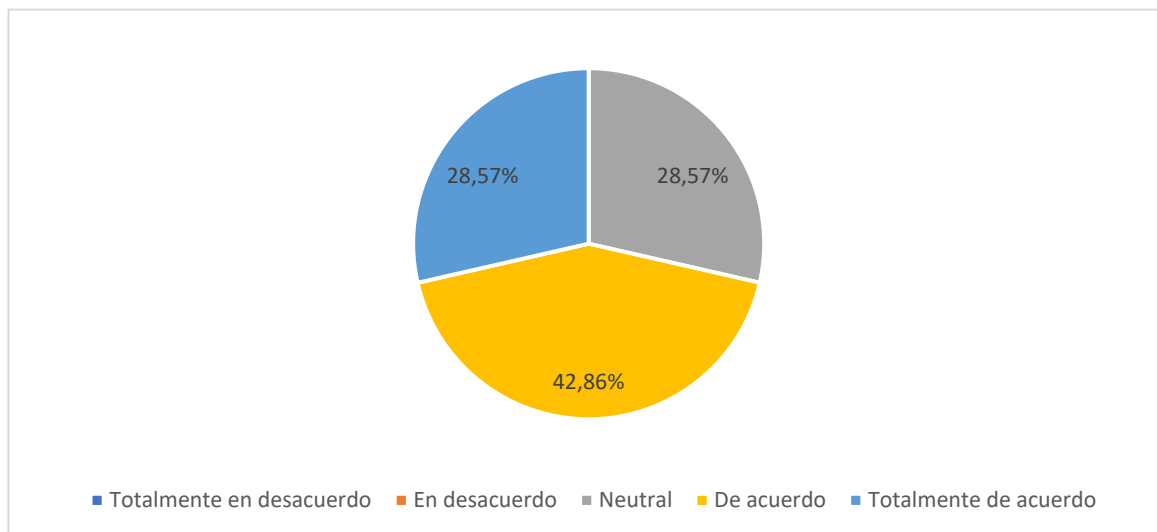


Figura 5. Gráfico de resultados: Pregunta 4.

El 71,43% de la población está de acuerdo, aunque en un nivel ligeramente distinto, en que la calidad de la atención veterinaria se relaciona directamente con el actual manejo manual de los registros. El 28,57% tiene una opinión neutral al respecto.

Los registros clínicos escritos en papel presentan diversas desventajas que afectan la eficiencia y precisión en la gestión de la información. Entre los principales inconvenientes se encuentra la dificultad para acceder rápidamente a los historiales, lo que puede retrasar la atención en situaciones críticas. Además, el riesgo de extravío, daño físico o deterioro de los documentos es elevado, comprometiendo la integridad de la información.

La falta de organización y la posibilidad de errores en la escritura o interpretación también limitan la calidad del servicio, mientras que la necesidad de almacenamiento físico ocupa espacio y dificulta la búsqueda de registros antiguos. En un entorno donde la agilidad y la precisión son esenciales, los sistemas manuales resultan obsoletos e ineficientes.

5. ¿Considera importante que la aplicación de gestión integre toda la información relacionada con los pacientes y sus propietarios?

Tabla 8. Detalle de resultados: Pregunta 5.

Indicador	Frecuencia	Porcentaje
Totalmente en desacuerdo	0	0,00%
En desacuerdo	0	0,00%
Neutral	0	0,00%
De acuerdo	2	28,57%
Totalmente de acuerdo	5	71,43%
Total	7	100%

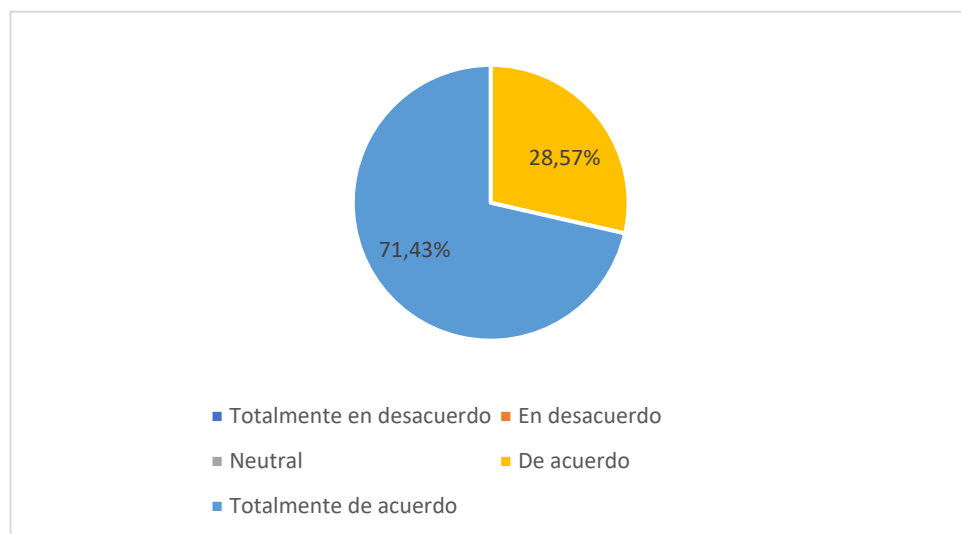


Figura 6. Gráfico de resultados: Pregunta 5.

En cuanto a la integración de toda la información relacionada con los pacientes y sus propietarios, en la aplicación móvil, el 100% de los encuestados están de acuerdo con ello, aunque en un grado ligeramente distinto.

Dispone de los datos de pacientes y propietarios centralizados en una aplicación móvil de gestión facilita un acceso rápido y seguro a la información, permitiendo a los profesionales de la salud tomar decisiones informadas en tiempo real. Además, la centralización de datos mejora la eficiencia operativa al reducir errores y duplicidades, mientras optimiza la coordinación entre diferentes especialistas.

6. ¿Considera importante que la aplicación móvil permita el acceso a los registros clínicos fuera del consultorio veterinario?

Tabla 9. Detalle de resultados: Pregunta 6.

Indicador	Frecuencia	Porcentaje
Nada importante	0	0,00%
Poco importante	0	0,00%
Neutral	2	28,57%
Importante	0	0,00%
Muy importante	5	71,43%
Total	7	100%

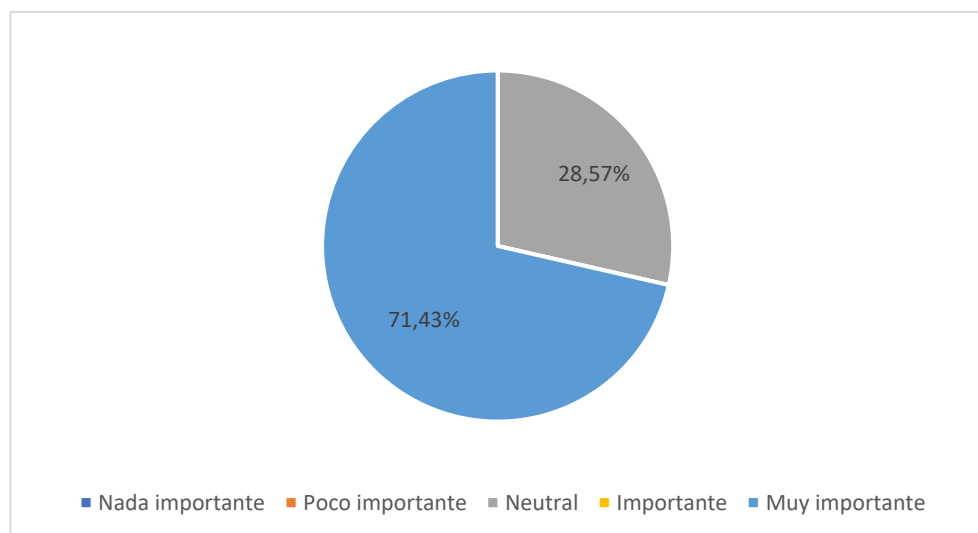


Figura 7. Gráfico de resultados: Pregunta 6.

Respecto al acceso a los registros clínicos fuera de la clínica veterinaria, el 71,43% de la población está de acuerdo con integrar la funcionalidad, mientras que el 28,57% tiene una opinión neutral.

Contar con la disponibilidad de los registros clínicos a través del teléfono móvil ofrece una ventaja para los veterinarios, ya que permite acceder a la información del paciente en tiempo real desde cualquier lugar. Esto facilita la toma de decisiones rápidas y precisas en situaciones de emergencia o cuando se encuentran fuera de la clínica. La movilidad también permite coordinar consultas remotas, revisar historiales médicos y gestionar citas, aumentando la eficiencia y productividad del servicio veterinario.

7. ¿Qué tan crítico considera que es el factor de seguridad al elegir una aplicación móvil de uso veterinario?

Tabla 10. Detalle de resultados: Pregunta 7.

Indicador	Frecuencia	Porcentaje
Nada crítico	0	0,00%
Poco crítico	0	0,00%
Neutral	2	28,57%
Crítico	0	0,00%
Muy crítico	5	71,43%
Total	7	100%

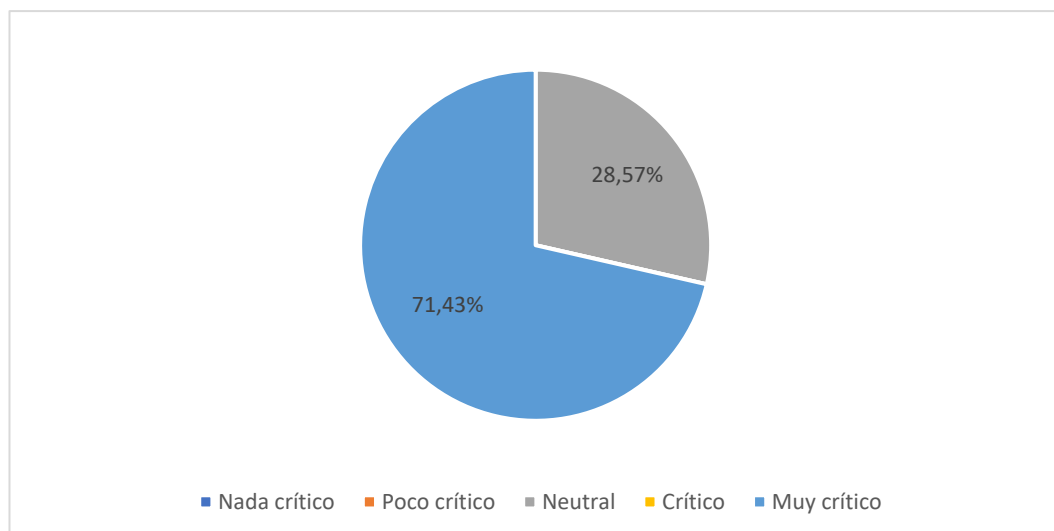


Figura 8. Gráfico de resultados: Pregunta 7.

La seguridad en una aplicación móvil de índole veterinaria es un aspecto crítico para el 71,43% de los encuestados, mientras que el 28,57% se muestra neutral.

La seguridad en las aplicaciones móviles veterinarias es un aspecto fundamental, ya que estas plataformas manejan información sensible tanto de los pacientes como de sus propietarios. Garantizar la protección de los datos clínicos, historiales médicos y datos personales es crucial para cumplir con normativas de privacidad y mantener la confianza de los clientes. Implementar medidas como cifrado de datos, autenticación de usuarios y copias de seguridad previene accesos no autorizados y posibles pérdidas de información.

8. ¿Qué tan importante cree que es la generación automática de informes de signos clínicos del paciente desde la aplicación?

Tabla 11. Detalle de resultados: Pregunta 8.

Indicador	Frecuencia	Porcentaje
Nada importante	0	0,00%
Poco importante	0	0,00%
Neutral	0	0,00%
Importante	3	42,86%
Muy importante	4	57,14%
Total	7	100%

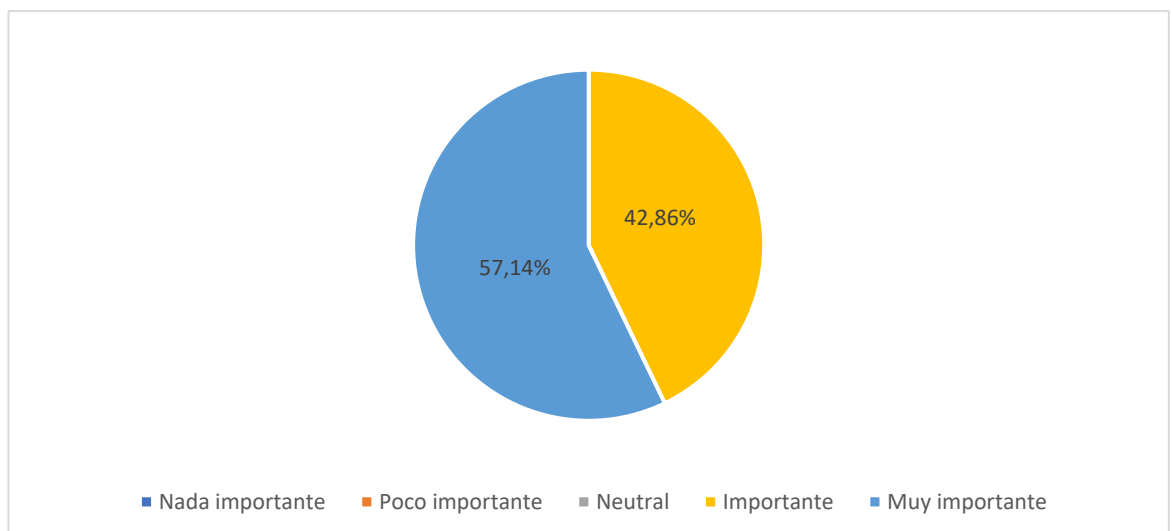


Figura 9. Gráfico de resultados: Pregunta 8.

Respecto a la generación automática de reportes en la aplicación, el 100% está de acuerdo que es un aspecto importante, aunque en un grado ligeramente distinto. La generación de reportes que reflejen la evolución de los signos clínicos del paciente veterinario a lo largo del tiempo es esencial para proporcionar un seguimiento preciso y efectivo de su estado de salud. Estos reportes permiten identificar patrones, mejoras o deterioros en la condición del animal, facilitando la toma de decisiones informadas y oportunas sobre tratamientos o ajustes en el manejo clínico. Además, ofrecen una herramienta visual y organizada que optimiza el trabajo del veterinario al contar con un historial detallado para consultas futuras, lo que contribuye a una atención más proactiva y basada en evidencia, garantizando así un mejor cuidado del paciente.

9. En su experiencia, ¿cómo calificaría la facilidad de uso de una aplicación móvil frente a un registro manual?

Tabla 12. Detalle de resultados: Pregunta 9.

Indicador	Frecuencia	Porcentaje
Mucho más difícil	0	0,00%
Más difícil	0	0,00%
Uso similar	0	0,00%
Más fácil	1	14,29%
Mucho más fácil	6	85,71%
Total	7	100%

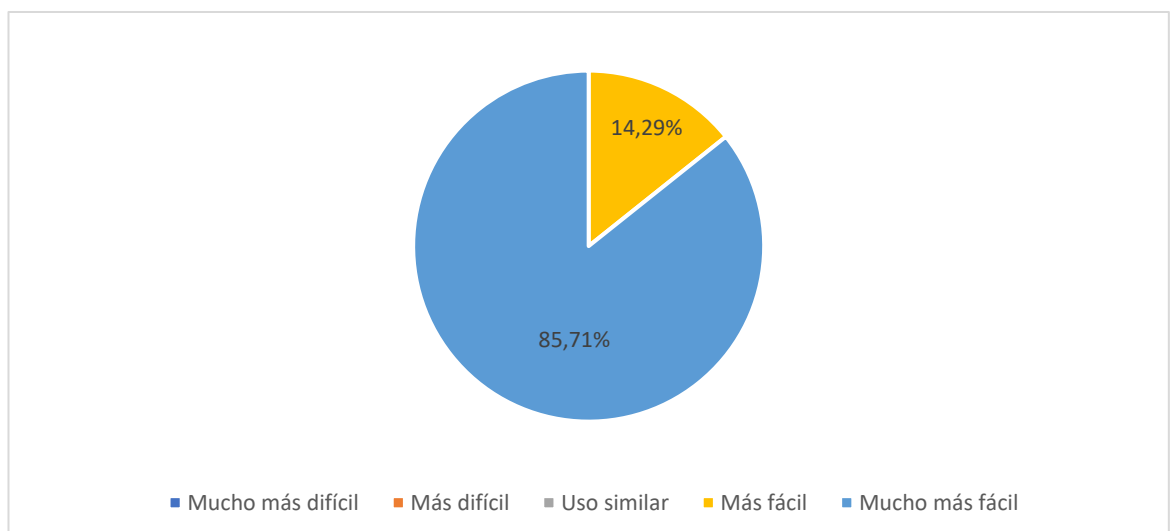


Figura 10. Gráfico de resultados: Pregunta 9.

El 100% de los encuestados coinciden en que sería más sencillo manejar los registros clínicos en una aplicación móvil frente al proceso manual. Esta digitalización ofrece una solución más eficiente y organizada en comparación con el proceso manual actual. Se podrá acceder, actualizar y almacenar la información de forma rápida y segura, eliminando el riesgo de perder datos debido a errores humanos, extravío o deterioro físico de los documentos. Además, facilitan la búsqueda de historiales médicos y la generación de reportes, lo que ahorra tiempo y mejora la precisión en la gestión interna.

10. ¿Estaría dispuesto(a) a invertir tiempo en capacitación para aprender a utilizar una nueva aplicación móvil de gestión de registros clínicos?

Tabla 13. Detalle de resultados: Pregunta 10.

Indicador	Frecuencia	Porcentaje
No	0	0,00%
No estoy seguro (a)	1	14,29%
Sí	6	85,71%
Total	7	100%

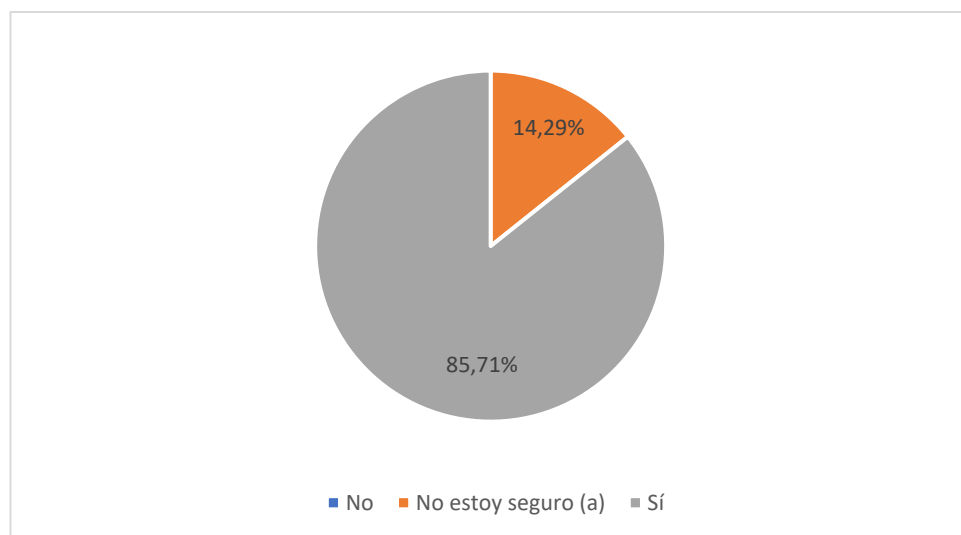


Figura 11. Gráfico de resultados: Pregunta 10.

El 85,71% de las personas que participaron en la encuesta están dispuestas a invertir tiempo de capacitación para el uso de una aplicación móvil de gestión de registros médicos veterinarios, lo que muestra un alto interés por digitalizar este aspecto del manejo clínico.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

El análisis realizado sobre la gestión de los registros clínicos en la veterinaria Pet's Home permitió identificar las diferencias significativas entre el proceso manual y el proceso automatizado mediante una aplicación móvil. En el método manual, se evidenció una serie de problemas recurrentes, como la pérdida de información, el deterioro físico de los registros en papel, y el tiempo considerable invertido en la búsqueda y actualización de historiales clínicos. Además, el almacenamiento de documentos físicos demanda espacio adicional y dificulta el acceso rápido a la información cuando se requiere fuera de la clínica, lo que limita la capacidad de respuesta en situaciones de urgencia.

Por otro lado, al aplicar el proceso automatizado a través de una aplicación móvil se evidenciaron ventajas como la rapidez y eficiencia para acceder a los historiales clínicos, permitiendo búsquedas instantáneas y actualizaciones en tiempo real. La información se almacena de manera segura y centralizada, minimizando los riesgos de pérdida. Además, la posibilidad de generar reportes automáticos sobre la evolución de los signos clínicos facilita un seguimiento detallado del estado de salud de cada paciente, optimizando así la toma de decisiones en cuanto a tratamientos y cuidados específicos.

La comparación entre ambos procesos demuestra que la implementación de un sistema automatizado en la veterinaria Pet's Home contribuirá, no solo a resolver las limitaciones del método manual, sino que también elevará la calidad y eficiencia del servicio. La optimización del tiempo, la seguridad de los datos y la facilidad de acceso a la información permiten a los veterinarios concentrarse en brindar una atención más precisa y oportuna a sus pacientes, beneficiando tanto a la clínica como a los propietarios de las mascotas.

3.1.1 Modelo del proceso manual para la gestión de pacientes en la clínica veterinaria Pet's Home

La Figura 12 muestra el proceso manual para la gestión de pacientes en la clínica veterinaria Pet's Home.

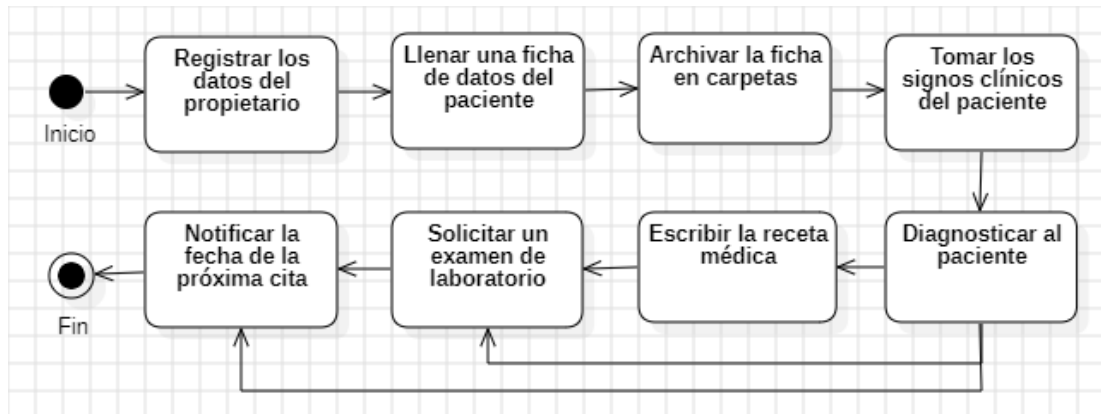


Figura 12. Diagrama del proceso manual de gestión de pacientes.

La Tabla 14 detalla cada una de las actividades llevadas a cabo como parte del proceso manual para la gestión de información de los pacientes de la clínica veterinaria Pet's Home, con la finalidad de comprender de forma precisa el flujo de trabajo del establecimiento. Cabe recalcar que estas tareas son responsabilidad de cualquier miembro del personal a cargo en un determinado momento.

Tabla 14. Descripción del proceso manual.

Nº	Actividad	Descripción
1	Registrar los datos personales del propietario.	Se anotan los datos personales del propietario junto con el nombre del paciente en una libreta.
2	Completar una ficha de información acerca del paciente.	El propietario, junto con un médico o auxiliar, llenan una ficha que contiene los datos de la mascota junto con algunos antecedentes, cuidados habituales y tipo de alimentación.

3	Archivar la ficha en carpetas.	Las fichas se guardan en carpetas, estableciendo un orden por nombre del paciente.
4	Tomar los signos clínicos del paciente.	Se toma los signos clínicos como la temperatura, presión arterial, llenado capilar, entre otros.
5	Diagnosticar y escribir la receta médica.	Una vez realizado el chequeo médico, se diagnostica y entrega una receta médica escrita en papel.
6	Solicitar un examen de laboratorio.	En caso de ser necesario, se adjunta en papel el tipo de examen clínico que requiere el paciente.
7	Notificar la fecha de la siguiente cita médica.	Se notifica verbalmente la fecha de la siguiente cita médica.

3.1.2 Modelo propuesto del proceso con el uso de la aplicación móvil para la gestión de pacientes en la clínica veterinaria Pet's Home

La Figura 13 muestra el proceso automatizado propuesto para la gestión de pacientes en la clínica veterinaria Pet's Home.

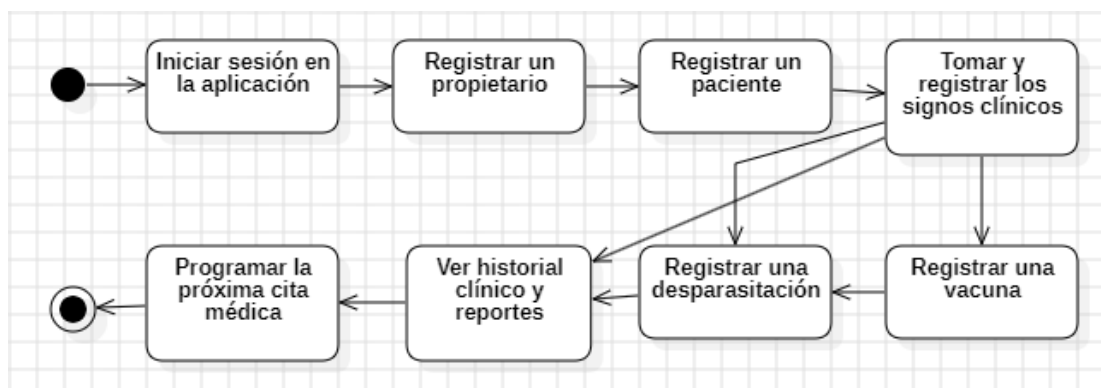


Figura 13. Diagrama del proceso automatizado de gestión de pacientes.

La Tabla 15 detalla cada una de las actividades llevadas a cabo como parte del proceso automatizado propuesto en la aplicación de gestión de información de los pacientes de la clínica veterinaria, con la finalidad de comprender de forma precisa el flujo de acciones que implica una cita médica.

Tabla 15. Descripción del proceso automatizado.

N°	Actividad	Descripción
1	Iniciar sesión en el sistema	Ingresar el nombre de usuario y clave para ingresar al sistema.
2	Registrar un propietario	En la sección de propietarios, se abre el formulario y se llenan los respectivos datos.
3	Registrar un paciente	Desde el registro del propietario o en la sección de pacientes, se abre el formulario y se llenan los datos necesarios.
4	Tomar y registrar los signos clínicos del paciente	Mientras se toman los signos clínicos, se registran en un formulario de una subsección del paciente.
5	Registrar una vacuna	De ser necesario, se registrará la información de una vacuna administrada al paciente en la cita médica actual.
6	Registrar una desparasitación	Si es requerido, se ingresará la información de una desparasitación administrada al paciente en la cita médica actual.
7	Usar el historial clínico o reportes para asistir el diagnóstico	Se puede consultar el historial médico del paciente o los reportes de su evolución clínica para la toma de decisiones.
8	Programar la siguiente cita médica	Ingresar la fecha y hora de la siguiente cita médica, si es que es necesario.

3.1.3 Análisis comparativo y selección de la metodología de desarrollo

Para seleccionar una metodología de trabajo adecuada que se adapte a las necesidades del proyecto, se realizó un análisis comparativo para evaluar las metodologías Kanban, Extreme Programming (XP) y Scrum (Tabla 16), destacando sus características, ventajas y limitaciones en aspectos clave como roles, iteraciones y gestión del equipo. La comparación identificó el enfoque más conveniente para lograr un desarrollo ágil, efectivo y alineado con los objetivos del proyecto.

Tabla 16. Tabla comparativa de metodologías de desarrollo.

	Kanban	XP (Extreme Programming)	Scrum
Filosofía principal	Gestión visual del flujo de trabajo y mejora continua.	Desarrollo ágil centrado en la calidad del código, la colaboración estrecha y la retroalimentación continua.	Entregas iterativas enfocadas en la planificación, ejecución y revisión del producto.
Etapas de desarrollo	No establece etapas definidas, el trabajo fluye continuamente.	Ciclos iterativos breves con énfasis en el diseño, codificación, pruebas y retroalimentación.	Fases bien definidas: planificación, ejecución, revisión y retrospectiva.
Roles	No roles específicos, los equipos comparten responsabilidades.	Roles informales; fomenta la colaboración entre desarrolladores, testers y clientes.	Product Owner, Scrum Master, y Equipo de Desarrollo.
Tamaño del equipo	Equipos de cualquier tamaño, ideal para equipos pequeños o medianos.	Adaptable, puede ser utilizado desde trabajo individual hasta equipos pequeños de 3-10 personas.	Ideal para equipos pequeños de 5-10 personas.
Duración de iteraciones	No aplica, flujo continuo.	Iteraciones cortas de 1 a 2 semanas.	Iteraciones fijas de 2 a 4 semanas llamadas sprints.

Gestión del cambio	Muy flexible, facilita la incorporación de cambios en cualquier momento.	Flexible, acepta cambios en cada iteración si mejoran la calidad o cumplen con requisitos nuevos.	Moderada, los cambios se introducen al inicio de cada sprint.
Visibilidad del progreso	Muy alta gracias al tablero Kanban.	Alta, mediante retroalimentación continua y la entrega de funcionalidades listas al cliente.	Alta, a través de reuniones diarias (Daily Scrum) y revisiones al final de cada sprint.
Priorización de tareas	Basada en columnas del tablero Kanban; las tareas se organizan en función del flujo de trabajo.	Priorización basada en las historias de usuario y las necesidades más urgentes del cliente.	Se priorizan los elementos del backlog al inicio de cada sprint.

La elección de Extreme Programming (XP) como metodología para este proyecto se fundamenta en los aspectos positivos destacados en el análisis comparativo con Scrum y Kanban. XP sobresale por su flexibilidad y adaptabilidad, lo que le permite responder de manera eficiente a cambios en los requisitos, una característica clave para un sistema de gestión clínica veterinaria donde las necesidades pueden evolucionar constantemente.

Según el análisis, XP ofrece iteraciones cortas de una a dos semanas, lo que facilita realizar ajustes continuos y garantiza entregas de alta calidad. Además, su enfoque en la retroalimentación frecuente y la colaboración cercana con los stakeholders asegura que las funcionalidades desarrolladas sean relevantes y aporten valor al cliente. Su capacidad para adaptarse al trabajo individual también la hace idónea para este proyecto, gestionado por una sola persona, permitiendo un flujo de trabajo organizado y eficiente. Estos atributos, destacados en la comparación, hacen de XP la metodología más adecuada para alcanzar los objetivos del sistema.

3.1.4 Análisis comparativo y selección de la tecnología de desarrollo

En el desarrollo de una aplicación móvil para la gestión de registros clínicos en una veterinaria, la elección de la tecnología adecuada es clave para garantizar un rendimiento óptimo, una experiencia de usuario fluida y un desarrollo eficiente.

Este análisis comparativo evalúa tres tecnologías: Android Nativo, Ionic y Xamarin (Tabla 17), contrastándolas en términos de rendimiento, diseño de interfaces, costos, y otros aspectos críticos. El objetivo es seleccionar la tecnología que mejor se alinee con los requisitos del proyecto y las necesidades de los usuarios finales.

Tabla 17. Tabla comparativa de tecnologías de desarrollo móvil.

	Android Nativo	Ionic	Xamarin
Rendimiento	Excelente, aprovecha todas las capacidades del dispositivo.	Moderado, depende del rendimiento del contenedor web y no del hardware directamente.	Bueno, aunque puede haber ligeras caídas en comparación con nativo en aplicaciones complejas.
Experiencia de usuario	Alta personalización con diseño nativo que sigue las pautas de Android Material Design.	Limitada a componentes web, aunque puede simular el diseño nativo.	Buena, utiliza herramientas para emular el diseño nativo, pero con algunas restricciones.
Costo de desarrollo	Mayor inversión inicial debido al desarrollo separado para Android (sin soporte multiplataforma).	Bajo, un solo código base para Android, iOS y web.	Intermedio, permite compartir gran parte del código, pero requiere optimización adicional.
Mantenimiento	Fácil de mantener si se desarrolla exclusivamente para Android.	Menos complejo gracias al código único, pero depende de actualizaciones de terceros.	Requiere actualizaciones del framework y ajustes en las integraciones multiplataforma.

Soporte y comunidad	Excelente, respaldado por Google con una amplia comunidad y documentación oficial.	Amplio soporte, pero depende de la comunidad y las actualizaciones de su entorno web.	Bueno, pero más limitado en comparación con Android Nativo.
Acceso a funcionalidades nativas	Total, sin limitaciones para usar sensores, cámaras y otras características del dispositivo.	Limitado, algunas características avanzadas requieren plugins específicos.	Bueno, con soporte para la mayoría de funcionalidades nativas mediante .NET.
Reusabilidad del código	Ninguna, diseñado exclusivamente para Android.	Alta, un solo código para múltiples plataformas.	Alta, permite compartir lógica entre plataformas, pero las interfaces deben ajustarse.

Tras el análisis comparativo, Android se selecciona como la tecnología más adecuada para el desarrollo de la aplicación móvil destinada a la gestión de registros clínicos de la veterinaria Pet's Home. Esta decisión se sustenta en su rendimiento superior, que permite aprovechar al máximo las capacidades del hardware y garantizar una experiencia fluida para los usuarios. Además, el soporte amplio y la robusta comunidad de desarrollo de Android garantizan una base sólida para mantener y escalar la aplicación en el futuro.

Es importante señalar que todos los usuarios finales utilizan exclusivamente dispositivos Android. Este factor elimina la necesidad de desarrollar para múltiples plataformas y refuerza la conveniencia de emplear un enfoque nativo para maximizar la compatibilidad y el rendimiento.

3.1.5 Análisis comparativo y selección de la base de datos

La elección de una base de datos es un paso fundamental en el desarrollo de la aplicación móvil para la gestión de registros clínicos en la veterinaria. En este análisis se comparan tres opciones ampliamente utilizadas: SQL Server, MySQL y PostgreSQL (Tabla 18), evaluando su rendimiento, escalabilidad, seguridad, facilidad de integración con aplicaciones móviles y otros aspectos relevantes.

Tabla 18. Tabla comparativa de bases de datos.

	SQL Server	MySQL	PostgreSQL
Rendimiento	Excelente, ideal para manejar grandes volúmenes de datos con alta fiabilidad.	Bueno, efectivo para aplicaciones pequeñas y medianas.	Excelente, adecuado para operaciones complejas y cargas intensivas.
Escalabilidad	Muy alta, adecuado para aplicaciones que planean crecer en volumen de datos.	Media, puede enfrentar desafíos en sistemas grandes.	Muy alta, con opciones avanzadas para particionamiento y clustering.
Facilidad de integración con Android	Compatible mediante controladores JDBC y APIs REST para conectar con la aplicación móvil.	Muy fácil de integrar con Android a través de conectores y APIs REST.	Compatible, pero requiere configuraciones adicionales para optimización.
Características avanzadas	Amplias, incluye herramientas de replicación, análisis avanzado y soporte para Big Data.	Básicas, funcionales para aplicaciones estándar pero limitadas para análisis complejos.	Muy avanzadas, incluyendo soporte para tipos de datos complejos y transacciones.
Seguridad	Muy alta, con cifrado, autenticación sólida y auditorías avanzadas.	Buena, con opciones básicas de cifrado y autenticación.	Muy alta, con autenticación avanzada y configuraciones de seguridad robustas.
Costo	Licencia comercial (versiones completas) o gratuita (versión Express).	Gratuito, con licencia de código abierto.	Gratuito, con licencia de código abierto.
Soporte y comunidad	Excelente, respaldado por Microsoft con amplia documentación y recursos.	Muy amplio, con una de las comunidades de código abierto más activas.	Amplio, pero requiere conocimientos técnicos más avanzados para ciertos casos.

Tras analizar algunas opciones disponibles, se selecciona SQL Server como la base de datos más adecuada para este proyecto. Su rendimiento destacado y su capacidad para manejar grandes volúmenes de datos la convierten en una opción confiable para una aplicación que gestionará información crítica de pacientes. Además, su compatibilidad con controladores JDBC y APIs REST garantiza una integración eficiente con la aplicación móvil desarrollada en Android Studio, asegurando una comunicación fluida entre la base de datos y el cliente móvil.

La posibilidad de utilizar la versión gratuita SQL Server Express facilita el desarrollo inicial sin comprometer el rendimiento ni la escalabilidad futura. Este equilibrio entre funcionalidad, seguridad e integración hace de SQL Server la mejor elección para garantizar el éxito del sistema de gestión clínica.

3.1.6 Diseño de la arquitectura del sistema

La arquitectura del sistema se basa en un modelo cliente-servidor distribuido, estructurado en tres capas principales: presentación (cliente), lógica de negocio (servidor) y datos (Figura 14).

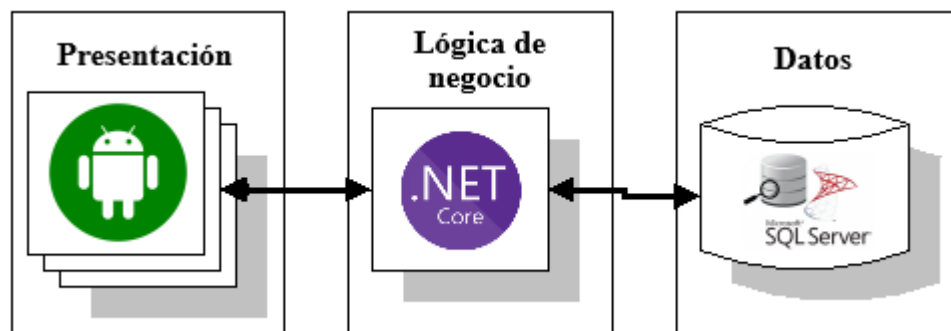


Figura 14. Arquitectura del sistema.

La aplicación móvil, desarrollada en Android con Java, representa la capa de presentación. Su principal función es proporcionar una interfaz intuitiva y funcional para los usuarios finales. Para la comunicación con el servidor, se utiliza la biblioteca Volley, que permite realizar solicitudes HTTP y manejar las respuestas en formato JSON.

La lógica de negocio se implementa mediante una API desarrollada en .NET. Esta API expone endpoints para realizar operaciones CRUD (Crear, Leer, Actualizar y Eliminar) sobre los datos de la base de datos. También valida y gestiona las solicitudes provenientes de la aplicación móvil.

Los datos se almacenan en un servidor SQL Server alojado en Somee. La base de datos organiza y guarda la información necesaria para el sistema, como los registros clínicos de los pacientes, el historial médico y otros datos relevantes.

El flujo de interacción comienza con la aplicación móvil enviando solicitudes HTTP a la API mediante URLs específicas. La API procesa estas solicitudes, interactúa con la base de datos y devuelve las respuestas en formato JSON, que luego son interpretadas y presentadas por la aplicación móvil al usuario final.

3.2 Desarrollo de la propuesta

El desarrollo de la aplicación móvil para la gestión de pacientes en la veterinaria se llevó a cabo utilizando un conjunto de tecnologías y una metodología de trabajo que garantizan un producto eficiente, escalable y alineado con las necesidades del proyecto. Se optó por desarrollar la aplicación en Android nativo mediante Android Studio con el lenguaje de programación Java, asegurando una experiencia de usuario fluida y totalmente integrada con los dispositivos Android del personal médico.

Como base de datos, se empleó SQL Server, conocida por su rendimiento robusto y su seguridad, lo que permite manejar los datos clínicos con altos estándares de calidad. Adicionalmente, se desarrolló una API en .NET utilizando C# para la lógica de negocio y la comunicación entre la aplicación móvil y la base de datos. La API utiliza el formato JSON para el intercambio de datos y está protegida con JWT (JSON Web Tokens) para garantizar la autenticidad y seguridad de las comunicaciones.

Para organizar el proceso de desarrollo, se implementó la metodología ágil Extreme Programming (XP), una elección que permite adaptarse a los cambios en los requisitos y entregar un producto funcional en iteraciones rápidas. Las fases de XP que fueron aplicadas se describen a continuación.

- **Planificación**

En esta etapa inicial se identificaron los requisitos funcionales y no funcionales de la aplicación con el estándar IEE 830, y se priorizaron las tareas a realizar. Se trabajó estrechamente con los stakeholders de la veterinaria para entender sus necesidades y traducirlas en historias de usuario claras y detalladas.

- **Diseño**

Durante esta fase se elaboró un diagrama de la base de datos y prototipos de las interfaces de usuario. Se priorizó un diseño simple y eficiente, siguiendo pautas de para garantizar una experiencia intuitiva y agradable.

- **Codificación**

En esta etapa se desarrollaron los componentes principales del sistema: la aplicación móvil en Android, con interfaces diseñadas para facilitar la captura y consulta de registros clínicos; y la API en .NET para gestionar las operaciones CRUD en la base de datos SQL Server. Se aplicaron las prácticas de programación de XP, como la codificación en pequeñas entregas y la integración continua, para garantizar un avance progresivo y controlado.

- **Pruebas**

Durante el desarrollo, se realizaron pruebas de aceptación sobre cada una de las historias de usuario para asegurar la calidad del software. Las pruebas se enfocaron en garantizar la funcionalidad esperada en cada historia, incluyendo los criterios de aceptación que hayan sido establecidos.

- **Lanzamiento**

Finalmente, se preparó la aplicación para su despliegue en los dispositivos Android del personal de la veterinaria. Esta etapa incluyó la configuración del entorno de producción para la API y la base de datos, así como una sesión de capacitación para los usuarios finales sobre el uso de la aplicación.

3.2.1 Fase I: Planificación

Se generó un levantamiento de requerimientos utilizando la norma IEEE 830 (Anexo B) como guía para el registro detallado de los mismos. Así, se lograron establecer de forma sistemática los requisitos funcionales y no funcionales de la aplicación móvil para la gestión de pacientes de la clínica veterinaria Pet's Home.

a. Grupo de interés (Stakeholders)

- **Usuarios del sistema:** Personal de la clínica veterinaria Pet's Home.
- **Desarrollador:** Investigador del proyecto.

b. Roles

La metodología XP define roles con responsabilidades bien definidas para garantizar la fluidez del desarrollo del proyecto. La Tabla 19 detalla cada uno de estos roles y sus respectivas funciones.

Tabla 19. Roles del proyecto.

Rol	Funciones	Miembro del equipo
Cliente	Establecer los requisitos del sistema.	Sandy Senteno
Desarrollador	Crear una aplicación que englobe las funcionalidades establecidas.	Christian Jaramillo
Coach	Orientar el desarrollo del proyecto.	Ing. Carlos Núñez

c. Requerimientos del sistema

La presentación de los requerimientos del sistema se realiza a través de las historias de usuario, asegurando que la aplicación final cumpla con las necesidades del cliente. La interacción entre los stakeholders es continua, por lo que se pueden llevar a cabo cambios en el proyecto sin afectar los tiempos y objetivos de manera significativa. Para facilitar la comprensión y manejo de los requisitos del sistema, esta sección presenta el backlog del producto, las historias de usuario y sus respectivas tareas específicas.

- **Backlog**

El backlog engloba una guía para el desarrollo del aplicativo, presentando cada una de las historias de usuario acompañada con un índice de importancia (establecido en base a su relevancia identificada durante el levantamiento de requerimientos), el esfuerzo estimado (cada punto de esfuerzo se traduce como 2 horas de trabajo), detalles para probarla y notas adicionales. Esto permite al desarrollador priorizar actividades, realizar entregas continuas significativas y adaptarse a los cambios. La Tabla 20 presenta el backlog del producto a desarrollar en el presente proyecto.

Tabla 20. Backlog del producto.

N°	Nombre	Imp. ¹	Esf. ²	Manera de probarlo	Notas
1	Implementación de una base de datos.	100	6	Realizar consultas y verificar las respuestas obtenidas.	Disponer de un backup para la creación.
2	Implementación de una API.	100	10	Probar cada servicio de la API y verificar las respuestas.	Disponibilidad en línea de la API.
3	Gestión de datos de la clínica.	70	5	Registrar y modificar los datos de la clínica.	Función exclusiva del usuario administrador.
4	Gestión de empleados.	100	10	Registrar, modificar y eliminar datos de los empleados.	Función exclusiva del usuario administrador.

5	Gestión de usuarios.	100	10	Acceder al sistema con cada tipo de usuario y verificar sus permisos.	Asignar permisos de visualización por rol de usuario.
6	Autenticación de usuarios.	100	9	Intentar acceder con credenciales correctas e incorrectas.	Mensaje para indicar si las credenciales ingresadas son incorrectas.
7	Gestión de propietarios.	100	10	Registrar, modificar y eliminar datos de los propietarios.	
8	Gestión de pacientes.	100	10	Registrar, modificar y eliminar datos de los pacientes.	
9	Gestión de vacunas.	80	10	Registrar, modificar y eliminar datos de las vacunas de un paciente.	
10	Gestión de desparasitaciones.	80	10	Registrar, modificar y eliminar datos de las desparasitaciones de un paciente.	
11	Gestión de citas médicas.	100	10	Registrar, modificar y eliminar datos de las citas médicas de un paciente.	
12	Citas médicas programadas.	80	5	Visualizar las citas médicas próximas a suceder y poder contactar al propietario.	Añadir un acceso al Whatsapp del propietario.
13	Gestión de registros médicos.	100	10	Registrar, modificar y eliminar datos de un registro médico de un paciente.	
14	Generación de reportes de signos clínicos.	90	5	Generar el reporte de cualquier aspecto de un paciente y compararlo con los datos relacionados.	Reporte gráfico a manera de gráfico.

¹ Imp.: Índice de importancia.

² Esf.: Esfuerzo estimado

- **Historias de usuario**

Las historias de usuario se presentan como tablas que contienen la descripción detallada de una funcionalidad, partiendo de una petición del cliente. Más adelante, de estas se deriva un conjunto de tareas que contribuyen al desarrollo de la funcionalidad. En la Tabla 21 se detalla el modelo de una historia de usuario, estableciendo así un formato claro y estándar para un correcto seguimiento y comprensión. La prioridad y el riesgo de cada historia de usuario se establece con base a lo identificado durante el levantamiento de requerimientos con el cliente.

Tabla 21. Modelo de una historia de usuario.

Historia de usuario	
Número: Identificador numérico.	Usuario: Persona que establece el requerimiento.
Nombre: Título de la historia de usuario.	
Prioridad: Índice de necesidad para el cliente (alta, media o baja).	Riesgo: Impacto en caso de fallo (alto, medio, bajo).
Puntos estimados: Horas estimadas de desarrollo.	Iteración asignada: Número de iteración de la historia.
Programador responsable: Persona encargada del desarrollo de la historia.	
Requerimiento base (IEEE 830): Código del requerimiento presentado en el levantamiento de requisitos con la norma IEEE 830.	
Descripción: Detalles de la historia de usuario.	
Criterios de aceptación: Parámetros necesarios para cumplir con la historia.	

- **Tareas**

Las tareas son acciones específicas que engloban la implementación de una funcionalidad descrita en una historia de usuario. Además de la descripción y detalles de la tarea, se establecen criterios de aceptación, que son parámetros que debe cumplir la tarea para considerarse como culminada. En la Tabla 22 se detalla el modelo de una tarea, que es una tabla con los detalles para llevar a cabo cada una de las acciones que conforman una historia de usuario.

Tabla 22. Modelo de una tarea.

Tarea	
Número: Identificador numérico.	Número de historia: Identificador de la historia pertinente.
Nombre: Título de la tarea.	
Tipo: Desarrollo, corrección, mejora, etc.	Puntos estimados: Horas estimadas de desarrollo de la tarea.
Fecha de inicio: dd/mm/aaaa	Fecha de fin: dd/mm/aaaa
Programador responsable: Persona encargada del desarrollo de la tarea.	
Descripción: Contexto y detalle de la tarea.	

A continuación, se presentan cada una de las historias de usuario, acompañada de la lista de tareas que la conforman. Para obtener más información acerca de cada tarea, se nombrará el anexo en donde constará el detalle de las mismas.

La Tabla 23 presenta la historia de usuario N°1, correspondiente a la implementación de la base de datos.

Tabla 23. Historia de usuario N°1: Implementación de la base de datos.

Historia de usuario	
Número: 1.	Usuario: Cliente.
Nombre: Implementación de la base de datos.	
Prioridad: Alta.	Riesgo: Alto.
Puntos estimados: 12.	Iteración asignada: 1.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF01.	
Descripción: Base de datos para almacenar y organizar toda la información de la clínica veterinaria.	
Criterios de aceptación: - Backup para la creación de la base de datos. - Entidades correctamente relacionadas, de acorde a la lógica de funcionamiento de la clínica veterinaria.	

Las tareas que conforman la historia de usuario N°1 se enlistan en la tabla 24. El detalle de cada una de ellas se encuentra en el Anexo C.

Tabla 24. Lista de tareas de la historia de usuario N°1.

N°	Nombre de la tarea	Descripción
1	Definir entidades y atributos.	Identificar los elementos necesarios para estructurar la base de datos.
2	Establecer relaciones entre entidades.	Identificar claves primarias y foráneas para relacionar entidades.
3	Generar un backup de la base de datos.	Crear un backup que permita restaurar la base de datos.
4	Restauración con el backup de la base de datos.	Ejecutar y verificar el correcto funcionamiento del backup.
5	Realizar pruebas de CRUD.	Ejecutar operaciones CRUD para comprobar los resultados derivados.

La Tabla 25 presenta la historia de usuario N°2, correspondiente a la implementación de una API.

Tabla 25. Historia de usuario N°2: Implementación de una API.

Historia de usuario	
Número: 2.	Usuario: Cliente.
Nombre: Implementación de una API.	
Prioridad: Alta.	Riesgo: Alto.
Puntos estimados: 20.	Iteración asignada: 1.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF02.	
Descripción: Recuperación de los datos de la clínica para poder gestionarlos y mantenerlos siempre actualizados de acorde a las necesidades de la clínica.	
Criterios de aceptación: - API alojada en línea y en funcionamiento para acoplarse a la aplicación móvil. - Servicios probados individualmente para comprobar que los datos se manipulen de acorde a lo establecido en cada uno.	

Las tareas que conforman la historia de usuario N°2 se enlistan en la tabla 26. El detalle de cada una de ellas se encuentra en el Anexo D.

Tabla 26. Lista de tareas de la historia de usuario N°2.

N°	Nombre de la tarea	Descripción
6	Implementar la API.	Implementar una API en .NET en Visual Studio con todos los componentes necesarios.
7	Probar el funcionamiento de la API.	Ejecutar a nivel local la API para verificar su correcto funcionamiento.
8	Publicar la API y la base de datos.	Publicar la API y la base de datos en un servicio en la nube.

La Tabla 27 presenta la historia de usuario N°3, correspondiente a la gestión de datos de la clínica.

Tabla 27. Historia de usuario N°3: Gestión de datos de la clínica.

Historia de usuario	
Número: 3.	Usuario: Cliente.
Nombre: Gestión de datos de la clínica.	
Prioridad: Alta.	Riesgo: Alto.
Puntos estimados: 10.	Iteración asignada: 1.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF03.	
Descripción: Sección para visualizar y modificar los datos de la clínica veterinaria, mismos que podrían ser visibles en otro apartado de la aplicación.	
Criterios de aceptación: - Mensajes de advertencia antes de modificar los datos. - Los datos modificados se reflejan automáticamente en otras secciones, si se está haciendo uso de ellos.	

Las tareas que conforman la historia de usuario N°3 se enlistan en la tabla 28. El detalle de cada una de ellas se encuentra en el Anexo E.

Tabla 28. Lista de tareas de la historia de usuario N°3.

N°	Nombre de la tarea	Descripción
9	Definir los datos esenciales de la clínica veterinaria.	Identificar los datos relevantes de la clínica veterinaria.
10	Diseñar la sección de los datos de clínica.	Crear el diseño de la sección destinada a visualizar y modificar los datos de la clínica.
11	Implementar la sección de datos de la clínica.	Crear la sección de datos de la clínica en Android Studio.
12	Agregar validaciones en los campos de datos de la clínica.	Añadir validaciones para permitir únicamente caracteres válidos y una extensión limitada para cada dato.
13	Implementar la función para modificar los datos de la clínica.	Agregar la función para actualizar los datos modificados a través de un botón.

La Tabla 29 presenta la historia de usuario N°4, correspondiente a la gestión de empleados.

Tabla 29. Historia de usuario N°4: Gestión de empleados.

Historia de usuario	
Número: 4.	Usuario: Cliente.
Nombre: Gestión de empleados.	
Prioridad: Alta.	Riesgo: Medio.
Puntos estimados: 20.	Iteración asignada: 2.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF04.	
Descripción: Agregar, modificar y eliminar empleados de la aplicación para gestionar al personal de la clínica.	
Criterios de aceptación: - Gestionar los datos de cada empleado mediante las funciones de agregar, eliminar y modificar.	

Las tareas que conforman la historia de usuario N°4 se enlistan en la tabla 30. El detalle de cada una de ellas se encuentra en el Anexo F.

Tabla 30. Lista de tareas de la historia de usuario N°4.

N°	Nombre de la tarea	Descripción
14	Diseñar la sección de gestión de empleados.	Crear el diseño de la sección destinada a visualizar y modificar los datos de los empleados.
15	Implementar la sección de gestión de empleados.	Crear la sección de gestión de empleados en Android Studio.
16	Diseñar el formulario para los empleados.	Crear el diseño del formulario relacionado con los empleados.
17	Implementar el formulario para empleados.	Implementar el formulario para el registro y modificación de datos de los empleados en Android Studio.
18	Agregar validaciones en el formulario de empleados.	Añadir validaciones para permitir únicamente caracteres válidos y una extensión limitada para cada dato.
19	Implementar las funciones para agregar, modificar y eliminar empleados.	Agregar las funcionalidades para gestionar los datos de los empleados a partir del formulario.

La Tabla 31 presenta la historia de usuario N°5, correspondiente a la gestión de usuarios.

Tabla 31. Historia de usuario N°5: Gestión de usuarios.

Historia de usuario	
Número: 5.	Usuario: Cliente.
Nombre: Gestión de usuarios.	
Prioridad: Alta.	Riesgo: Medio.
Puntos estimados: 20.	Iteración asignada: 2.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF05.	
Descripción: Asociar un usuario y contraseña al empleado para restringir el acceso al personal autorizado. Existirá un tipo de usuario por cada cargo en la clínica.	
Criterios de aceptación: - Mostrar las secciones y permitir las acciones correspondientes según el rol asignado. Verificar con un usuario de prueba de cada tipo.	

Las tareas que conforman la historia de usuario N°5 se enlistan en la tabla 32. El detalle de cada una de ellas se encuentra en el Anexo G.

Tabla 32. Lista de tareas de la historia de usuario N°5.

N°	Nombre de la tarea	Descripción
20	Definir los tipos de usuario del sistema y sus permisos.	Identificar los tipos de usuario requeridos según los cargos de los empleados de la clínica.
21	Diseñar el formulario para los usuarios.	Crear el diseño del formulario relacionado con los usuarios.
22	Implementar el formulario para usuarios.	Anexar los campos al formulario de empleados ya existente para complementar el registro.
23	Agregar validaciones en el formulario de usuarios.	Añadir validaciones para permitir únicamente caracteres válidos y una extensión limitada para cada dato.
24	Controlar la visualización de cada sección según el tipo de usuario.	Añadir controles que muestren y oculten las secciones según el tipo de usuario que ingrese al sistema.
25	Crear un usuario de cada tipo y verificar los permisos asignados.	Verificar que cada tipo de usuario tenga acceso a lo que le corresponde.

La Tabla 33 presenta la historia de usuario N°6, correspondiente a la autenticación de usuarios.

Tabla 33. Historia de usuario N°6: Autenticación de usuarios.

Historia de usuario	
Número: 6.	Usuario: Cliente.
Nombre: Autenticación de usuarios.	
Prioridad: Alta.	Riesgo: Alto.
Puntos estimados: 18.	Iteración asignada: 3.
Programador responsable: Christian Jaramillo	
Requerimiento base (IEEE 830): RF06.	
Descripción: Acceso a las funcionalidades del sistema a través de credenciales (usuario y contraseña).	
Criterios de aceptación: - Acceso al sistema con las credenciales únicas asignadas al usuario. - Al ingresar con información errónea, el sistema deberá notificarlo.	

Las tareas que conforman la historia de usuario N°6 se enlistan en la tabla 34. El detalle de cada una de ellas se encuentra en el Anexo H.

Tabla 34. Lista de tareas de la historia de usuario N°6.

N°	Nombre de la tarea	Descripción
26	Diseñar el formulario de inicio de sesión.	Crear el diseño de la interfaz del formulario.
27	Implementar el formulario de inicio de sesión.	Crear el formulario en una actividad dentro de Android Studio.
28	Implementar la validación de credenciales.	Configurar el formulario para que verifique las credenciales en la base de datos y maneje errores.
29	Integrar los roles de usuario.	Configurar el acceso al sistema de acorde al rol asignado del usuario.

La Tabla 35 presenta la historia de usuario N°7, correspondiente a la gestión de propietarios.

Tabla 35. Historia de usuario N°7: Gestión de propietarios.

Historia de usuario	
Número: 7.	Usuario: Cliente.
Nombre: Gestión de propietarios.	
Prioridad: Alta.	Riesgo: Medio.
Puntos estimados: 20.	Iteración asignada: 3.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF07.	
Descripción: Agregar, modificar y eliminar propietarios de la aplicación para gestionar a los tutores de los pacientes.	
Criterios de aceptación: - Gestionar los datos de cada propietario mediante las funciones de agregar, eliminar y modificar.	

Las tareas que conforman la historia de usuario N°7 se enlistan en la tabla 36. El detalle de cada una de ellas se encuentra en el Anexo I.

Tabla 36. Lista de tareas de la historia de usuario N°7.

N°	Nombre de la tarea	Descripción
30	Diseñar la sección de gestión de propietarios.	Crear el diseño de la sección destinada a visualizar y modificar los datos de los propietarios.
31	Implementar la sección de gestión de propietarios.	Crear la sección de gestión de propietarios en Android Studio.
32	Diseñar el formulario para los propietarios.	Crear el diseño del formulario relacionado con los propietarios.
33	Implementar el formulario para propietarios.	Implementar el formulario para el registro y modificación de datos de los propietarios en Android Studio.
34	Agregar validaciones en el formulario de propietarios.	Añadir validaciones para permitir únicamente caracteres válidos y una extensión limitada para cada dato.
35	Implementar las funciones para agregar, modificar y eliminar propietarios.	Agregar las funcionalidades para gestionar los datos de los propietarios a partir del formulario.

La Tabla 37 presenta la historia de usuario N°8, correspondiente a la gestión de pacientes.

Tabla 37. Historia de usuario N°8: Gestión de pacientes.

Historia de usuario	
Número: 8.	Usuario: Cliente.
Nombre: Gestión de pacientes.	
Prioridad: Alta.	Riesgo: Medio.
Puntos estimados: 20.	Iteración asignada: 4.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF08.	
Descripción: Agregar, modificar y eliminar pacientes de la aplicación para gestionar su información personal.	
Criterios de aceptación: - Gestionar los datos de cada paciente mediante las funciones de agregar, eliminar y modificar.	

Las tareas que conforman la historia de usuario N°8 se enlistan en la tabla 38. El detalle de cada una de ellas se encuentra en el Anexo J.

Tabla 38. Lista de tareas de la historia de usuario N°8.

N°	Nombre de la tarea	Descripción
36	Diseñar la sección de gestión de pacientes.	Crear el diseño de la sección destinada a visualizar y modificar los datos de los pacientes.
37	Implementar la sección de gestión de pacientes.	Crear la sección de gestión de pacientes en Android Studio.
38	Diseñar el formulario para los pacientes.	Crear el diseño del formulario relacionado con los pacientes.
39	Implementar el formulario para pacientes.	Implementar el formulario para el registro y modificación de datos de los pacientes en Android Studio.
40	Agregar validaciones en el formulario de pacientes.	Añadir validaciones para permitir únicamente caracteres válidos y una extensión limitada para cada dato.
41	Implementar las funciones para agregar, modificar y eliminar pacientes.	Agregar las funcionalidades para gestionar los datos de los pacientes a partir del formulario.

La Tabla 39 presenta la historia de usuario N°9, correspondiente a la gestión de vacunas.

Tabla 39. Historia de usuario N°9: Gestión de vacunas.

Historia de usuario	
Número: 9.	Usuario: Cliente.
Nombre: Gestión de vacunas.	
Prioridad: Alta.	Riesgo: Medio.
Puntos estimados: 20.	Iteración asignada: 4.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF09.	
Descripción: Agregar, modificar y eliminar vacunas para gestionar su historial en cada paciente.	
Criterios de aceptación: - Gestionar los datos de las vacunas administradas en cada paciente mediante las funciones de agregar, eliminar y modificar.	

Las tareas que conforman la historia de usuario N°9 se enlistan en la tabla 40. El detalle de cada una de ellas se encuentra en el Anexo K.

Tabla 40. Lista de tareas de la historia de usuario N°9.

N°	Nombre de la tarea	Descripción
42	Diseñar la sección de gestión de vacunas.	Crear el diseño de la sección destinada a visualizar y modificar los datos de las vacunas.
43	Implementar la sección de gestión de vacunas.	Crear la sección de gestión de vacunas en Android Studio.
44	Diseñar el formulario para las vacunas.	Crear el diseño del formulario relacionado con las vacunas.
45	Implementar el formulario para vacunas.	Implementar el formulario para el registro y modificación de datos de las vacunas en Android Studio.
46	Agregar validaciones en el formulario de vacunas.	Añadir validaciones para permitir únicamente caracteres válidos y una extensión limitada para cada dato.
47	Implementar las funciones para agregar, modificar y eliminar vacunas.	Agregar las funcionalidades para gestionar los datos de las vacunas a partir del formulario.

La Tabla 41 presenta la historia de usuario N°10, correspondiente a la gestión de desparasitaciones.

Tabla 41. Historia de usuario N°10: Gestión de desparasitaciones.

Historia de usuario	
Número: 10.	Usuario: Cliente.
Nombre: Gestión de desparasitaciones.	
Prioridad: Alta.	Riesgo: Medio.
Puntos estimados: 20.	Iteración asignada: 5.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF10.	
Descripción: Agregar, modificar y eliminar desparasitaciones para gestionar su historial en cada paciente.	
Criterios de aceptación: - Gestionar los datos de las desparasitaciones administradas en cada paciente mediante las funciones de agregar, eliminar y modificar.	

Las tareas que conforman la historia de usuario N°10 se enlistan en la tabla 42. El detalle de cada una de ellas se encuentra en el Anexo L.

Tabla 42. Lista de tareas de la historia de usuario N°10.

N°	Nombre de la tarea	Descripción
48	Diseñar la sección de gestión de desparasitaciones.	Crear el diseño de la sección destinada a visualizar y modificar los datos de las desparasitaciones.
49	Implementar la sección de gestión de desparasitaciones.	Crear la sección de gestión de desparasitaciones en Android Studio.
50	Diseñar el formulario para las desparasitaciones.	Crear el diseño del formulario relacionado con las desparasitaciones.
51	Implementar el formulario para desparasitaciones.	Implementar el formulario para el registro y modificación de datos de las desparasitaciones en Android Studio.
52	Agregar validaciones en el formulario de desparasitaciones.	Añadir validaciones para permitir únicamente caracteres válidos y una extensión limitada para cada dato.
53	Implementar las funciones para agregar, modificar y eliminar desparasitaciones.	Agregar las funciones para gestionar los datos de las desparasitaciones a partir del formulario.

La Tabla 43 presenta la historia de usuario N°11, correspondiente a la gestión de citas médicas.

Tabla 43. Historia de usuario N°11: Gestión de citas médicas.

Historia de usuario	
Número: 11.	Usuario: Cliente.
Nombre: Gestión de citas médicas.	
Prioridad: Alta.	Riesgo: Medio.
Puntos estimados: 20.	Iteración asignada: 5.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF11.	
Descripción: Agregar, modificar y eliminar citas médicas para gestionar su asignación en cada paciente.	
Criterios de aceptación: - Gestionar los datos de las citas médicas asignadas a cada paciente mediante las funciones de agregar, eliminar y modificar.	

Las tareas que conforman la historia de usuario N°11 se enlistan en la tabla 44. El detalle de cada una de ellas se encuentra en el Anexo M.

Tabla 44. Lista de tareas de la historia de usuario N°11.

N°	Nombre de la tarea	Descripción
54	Diseñar la sección de gestión de citas médicas.	Crear el diseño de la sección destinada a visualizar y modificar los datos de las citas médicas.
55	Implementar la sección de gestión de citas médicas.	Crear la sección de gestión de citas médicas en Android Studio.
56	Diseñar el formulario para las citas médicas.	Crear el diseño del formulario relacionado con las citas médicas.
57	Implementar el formulario para citas médicas.	Implementar el formulario para el registro y modificación de datos de las citas médicas en Android Studio.
58	Agregar validaciones en el formulario de citas médicas.	Añadir validaciones para permitir únicamente caracteres válidos y una extensión limitada para cada dato.
59	Implementar las funciones para agregar, modificar y eliminar citas médicas.	Agregar las funcionalidades para gestionar los datos de las citas médicas a partir del formulario.

La Tabla 45 presenta la historia de usuario N°12, correspondiente a las notificaciones de citas médicas programadas.

Tabla 45. Historia de usuario N°12: Citas médicas programadas.

Historia de usuario	
Número: 12.	Usuario: Cliente.
Nombre: Citas médicas programadas.	
Prioridad: Alta.	Riesgo: Medio.
Puntos estimados: 10.	Iteración asignada: 6.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF12.	
Descripción: Visualización de las citas médicas que están próximas a suceder y la opción de enviar un mensaje por Whatsapp al propietario.	
Criterios de aceptación: - Sección para ver las citas médicas programadas para el día actual y el día siguiente en la pantalla principal. - Añadir un botón a cada cita programada que permita abrir el chat de Whatsapp del propietario con un mensaje predeterminado.	

Las tareas que conforman la historia de usuario N°12 se enlistan en la tabla 46. El detalle de cada una de ellas se encuentra en el Anexo N.

Tabla 46. Lista de tareas de la historia de usuario N°12.

N°	Nombre de la tarea	Descripción
60	Diseñar la sección de citas médicas programadas.	Crear el diseño de la sección destinada a visualizar las citas médicas programadas.
61	Implementar la sección de citas médicas programadas.	Crear la sección de citas médicas programadas en Android Studio.
62	Implementar una función para contactar al propietario vía Whatsapp.	Añadir un botón que permita abrir el chat de Whatsapp del propietario con un mensaje predeterminado.

La Tabla 47 presenta la historia de usuario N°13, correspondiente a la gestión de registros médicos.

Tabla 47. Historia de usuario N°13: Gestión de registros médicos.

Historia de usuario	
Número: 13.	Usuario: Cliente.
Nombre: Gestión de registros médicos.	
Prioridad: Alta.	Riesgo: Medio.
Puntos estimados: 20.	Iteración asignada: 6.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF13.	
Descripción: Agregar, modificar y eliminar registros médicos para gestionar el historial clínico de cada paciente.	
Criterios de aceptación: - Gestionar los datos de los registros médicos asignados a cada paciente mediante las funciones de agregar, eliminar y modificar.	

Las tareas que conforman la historia de usuario N°13 se enlistan en la tabla 48. El detalle de cada una de ellas se encuentra en el Anexo O.

Tabla 48. Lista de tareas de la historia de usuario N°13.

N°	Nombre de la tarea	Descripción
63	Diseñar la sección de gestión de registros médicos.	Crear el diseño de la sección destinada a visualizar y modificar los datos de los registros médicos.
64	Implementar la sección de gestión de registros médicos.	Crear la sección de gestión de registros médicos en Android Studio.
65	Diseñar el formulario para los registros médicos.	Crear el diseño del formulario relacionado con los registros médicos.
66	Implementar el formulario para registros médicos.	Implementar el formulario para el registro y modificación de datos de los registros médicos en Android Studio.
67	Agregar validaciones en el formulario de registros médicos.	Añadir validaciones para permitir únicamente caracteres válidos y una extensión limitada para cada dato.
68	Implementar las funciones para agregar, modificar y eliminar registros médicos.	Agregar las funcionalidades para gestionar los datos de los registros médicos a partir del formulario.

La Tabla 49 presenta la historia de usuario N°14, correspondiente a la generación de reportes del historial clínico.

Tabla 49. Historia de usuario N°14: Generación de reportes del historial clínico.

Historia de usuario	
Número: 14.	Usuario: Cliente.
Nombre: Generación de reportes del historial clínico.	
Prioridad: Alta.	Riesgo: Medio.
Puntos estimados: 10.	Iteración asignada: 6.
Programador responsable: Christian Jaramillo.	
Requerimiento base (IEEE 830): RF14.	
Descripción: Generar un reporte de un signo clínico seleccionado en un gráfico para visualizar su evolución.	
Criterios de aceptación: - Generar un gráfico que refleje la evolución de los valores de un signo clínico seleccionado dentro del historial médico del paciente.	

Las tareas que conforman la historia de usuario N°14 se enlistan en la tabla 50. El detalle de cada una de ellas se encuentra en el Anexo P.

Tabla 50. Lista de tareas de la historia de usuario N°14.

N°	Nombre de la tarea	Descripción
69	Diseñar la pantalla de visualización del reporte del signo clínico.	Crear el diseño de la pantalla que contendrá el gráfico del reporte del signo clínico seleccionado.
70	Implementar la pantalla de visualización del reporte del signo clínico.	Crear la pantalla de visualización del reporte gráfico del signo clínico correspondiente.

d. Estimación de tiempo de historias de usuario

En la Tabla 51 se detalla el tiempo estimado para llevar a cabo cada una de las tareas de las historias de usuario descritas previamente. Como medida se usará el punto de estimación, mismo que corresponde a 2 horas de trabajo.

Tabla 51. Estimación de tiempo de desarrollo.

Historia	Tarea	Puntos de estimación	Horas	Total de horas
Historia de usuario 1	Tarea 1	1	2	12
	Tarea 2	1	2	
	Tarea 3	1	2	
	Tarea 4	1	2	
	Tarea 5	2	4	
Historia de usuario 2	Tarea 6	7	14	20
	Tarea 7	2	4	
	Tarea 8	1	2	
Historia de usuario 3	Tarea 9	1	2	10
	Tarea 10	1	2	
	Tarea 11	1	2	
	Tarea 12	1	2	
	Tarea 13	1	2	
Historia de usuario 4	Tarea 14	1	2	20
	Tarea 15	2	4	
	Tarea 16	1	2	
	Tarea 17	2	4	
	Tarea 18	1	2	
	Tarea 19	3	6	
Historia de usuario 5	Tarea 20	1	2	20
	Tarea 21	1	2	
	Tarea 22	2	4	

	Tarea 23	1	2	
	Tarea 24	4	8	
	Tarea 25	1	2	
Historia de usuario 6	Tarea 26	1	2	18
	Tarea 27	2	4	
	Tarea 28	2	4	
	Tarea 29	4	8	
Historia de usuario 7	Tarea 30	1	2	20
	Tarea 31	2	4	
	Tarea 32	1	2	
	Tarea 33	2	4	
	Tarea 34	1	2	
	Tarea 35	3	6	
Historia de usuario 8	Tarea 36	1	2	20
	Tarea 37	2	4	
	Tarea 38	1	2	
	Tarea 39	2	4	
	Tarea 40	1	2	
	Tarea 41	3	6	
Historia de usuario 9	Tarea 42	1	2	20
	Tarea 43	2	4	
	Tarea 44	1	2	
	Tarea 45	2	4	
	Tarea 46	1	2	
	Tarea 47	3	6	
Historia de usuario 10	Tarea 48	1	2	20
	Tarea 49	2	4	
	Tarea 50	1	2	
	Tarea 51	2	4	

	Tarea 52	1	2	
	Tarea 53	3	6	
Historia de usuario 11	Tarea 54	1	2	20
	Tarea 55	2	4	
	Tarea 56	1	2	
	Tarea 57	2	4	
	Tarea 58	1	2	
	Tarea 59	3	6	
Historia de usuario 12	Tarea 60	1	2	10
	Tarea 61	2	4	
	Tarea 62	2	4	
Historia de usuario 13	Tarea 63	1	2	20
	Tarea 64	2	4	
	Tarea 65	1	2	
	Tarea 66	2	4	
	Tarea 67	1	2	
	Tarea 68	3	6	
Historia de usuario 14	Tarea 69	1	2	10
	Tarea 70	4	8	
	Total →			240

e. Plan de entregas

La Tabla 52 engloba el tiempo estimado de cada una de las historias de usuario en horas y días, además de la iteración a la que será agregada. Cada iteración es un avance del desarrollo del aplicativo que se presenta al cliente.

Cabe recalcar que el cumplimiento del tiempo estimado se realizará con base a una jornada de 5 días, 8 horas cada día.

Tabla 52. Plan de entregas de la aplicación.

Historia	Tiempo estimado		Iteración asignada					
	Días	Horas	1	2	3	4	5	6
Historia 1	2	12	X					
Historia 2	3	20	X					
Historia 3	2	10	X					
Historia 4	3	20		X				
Historia 5	3	20		X				
Historia 6	2	18			X			
Historia 7	2	20			X			
Historia 8	3	20				X		
Historia 9	3	20				X		
Historia 10	3	20					X	
Historia 11	3	20					X	
Historia 12	2	10						X
Historia 13	3	20						X
Historia 14	2	10						X

3.2.2 Fase II: Diseño

Durante esta etapa se busca establecer las bases estructurales y visuales del sistema, asegurando que tanto la arquitectura de datos como las interfaces de usuario sean funcionales, intuitivas y alineadas con los objetivos del proyecto. Para ello, se diseñará el diagrama de la base de datos, definiendo las tablas y relaciones necesarias para garantizar la correcta organización y almacenamiento de los datos.

Además, se elaborarán prototipos de las interfaces de la aplicación móvil utilizando la herramienta en línea de maquetación Ninjamock, lo que permitirá visualizar el diseño y la funcionalidad de la aplicación antes de iniciar el desarrollo de la misma.

Esta etapa facilita la planificación de la implementación técnica y permite obtener retroalimentación temprana de los stakeholders sobre la estructura y apariencia del sistema, asegurando que el diseño final cumpla con las expectativas y necesidades del cliente.

Tarjetas CRC

En el desarrollo de este proyecto bajo la metodología XP (Extreme Programming), se utilizaron tarjetas CRC (Clase, Responsabilidad, Colaboradores) como herramienta para modelar y estructurar las clases del sistema. Estas tarjetas permiten definir de forma clara las responsabilidades de cada clase y sus relaciones con otras, asegurando un diseño cohesivo y alineado con los principios de simplicidad y comunicación de XP. Su uso facilitó la identificación de las clases principales y la distribución de responsabilidades, optimizando el diseño orientado a objetos y garantizando la modularidad y reutilización del código. La Tabla 53 muestra el modelo de una tarjeta CRC.

Tabla 53. Modelo de una tarjeta CRC.

Clase:	
Responsabilidades:	Colaboradores:

La Tabla 54 muestra la clase de autenticación con el detalle de sus responsabilidades y colaboradores.

Tabla 54. Tarjeta CRC de autenticación.

Clase: Autenticación.	
Responsabilidades:	Colaboradores:
Ingresar al sistema. Recuperar contraseña.	Pantalla de inicio.

La Tabla 55 muestra la clase de clínica con el detalle de sus responsabilidades y colaboradores.

Tabla 55. Tarjeta CRC de clínica.

Clase: Clínica.	
Responsabilidades:	Colaboradores:
Editar la información de la clínica veterinaria.	Menú principal.

La Tabla 56 muestra la clase de pacientes con el detalle de sus responsabilidades y colaboradores.

Tabla 56. Tarjeta CRC de pacientes.

Clase: Pacientes.	
Responsabilidades:	Colaboradores:
Registrar un paciente. Editar o eliminar los datos del paciente. Asociarlo a un propietario. Consultar citas médicas. Consultar registros clínicos. Consultar vacunas. Consultar desparasitaciones.	Menú principal. Citas médicas. Registros clínicos. Vacunas. Desparasitaciones.

La Tabla 57 muestra la clase de empleados con el detalle de sus responsabilidades y colaboradores.

Tabla 57. Tarjeta CRC de empleados.

Clase: Empleados.	
Responsabilidades:	Colaboradores:
Registrar un empleado. Editar o eliminar los datos del empleado. Asociarlo a un usuario.	Menú principal.

La Tabla 58 muestra la clase de cita médica con el detalle de sus responsabilidades y colaboradores.

Tabla 58. Tarjeta CRC de cita médica.

Clase: Cita médica.	
Responsabilidades:	Colaboradores:
Registrar una cita médica. Editar o eliminar los datos de la cita.	Paciente.

La Tabla 59 muestra la clase de cita médica programada con el detalle de sus responsabilidades y colaboradores.

Tabla 59. Tarjeta CRC de cita médica programada.

Clase: Cita médica programada.	
Responsabilidades:	Colaboradores:
Visualizar las citas médicas programadas del día actual y del día siguiente. Contactar al propietario vía Whatsapp.	Pantalla principal.

La Tabla 60 muestra la clase de registro clínico con el detalle de sus responsabilidades y colaboradores.

Tabla 60. Tarjeta CRC de registro clínico.

Clase: Registro clínico.	
Responsabilidades:	Colaboradores:
Añadir un registro clínico. Editar o eliminar los datos del registro. Generar reportes de un signo clínico.	Paciente.

La Tabla 61 muestra la clase de registro clínico con el detalle de sus responsabilidades y colaboradores.

Tabla 61. Tarjeta CRC de reporte.

Clase: Reporte.	
Responsabilidades:	Colaboradores:
Generar un reporte general del paciente. Generar el reporte de evolución de un signo clínico específico seleccionado.	Registro clínico.

La Tabla 62 muestra la clase de vacuna con el detalle de sus responsabilidades y colaboradores.

Tabla 62. Tarjeta CRC de vacuna.

Clase: Vacuna.	
Responsabilidades:	Colaboradores:
Añadir una vacuna. Editar o eliminar los datos de la vacuna.	Paciente.

La Tabla 63 muestra la clase de desparasitación con el detalle de sus responsabilidades y colaboradores.

Tabla 63. Tarjeta CRC de desparasitación.

Clase: Desparasitación.	
Responsabilidades:	Colaboradores:
Añadir una desparasitación. Editar o eliminar los datos de la desparasitación.	Paciente.

a. Iteración I

La primera iteración está conformada por tres historias de usuario: implementación de la base de datos, implementación de una API y gestión de datos de la clínica.

- **Historia de usuario N°1: Implementación de la base de datos**

El modelo relacional de la base de datos (Figura 15) se generó con el asistente integrado de SQL Server, mismo que refleja las entidades, relaciones y las claves.

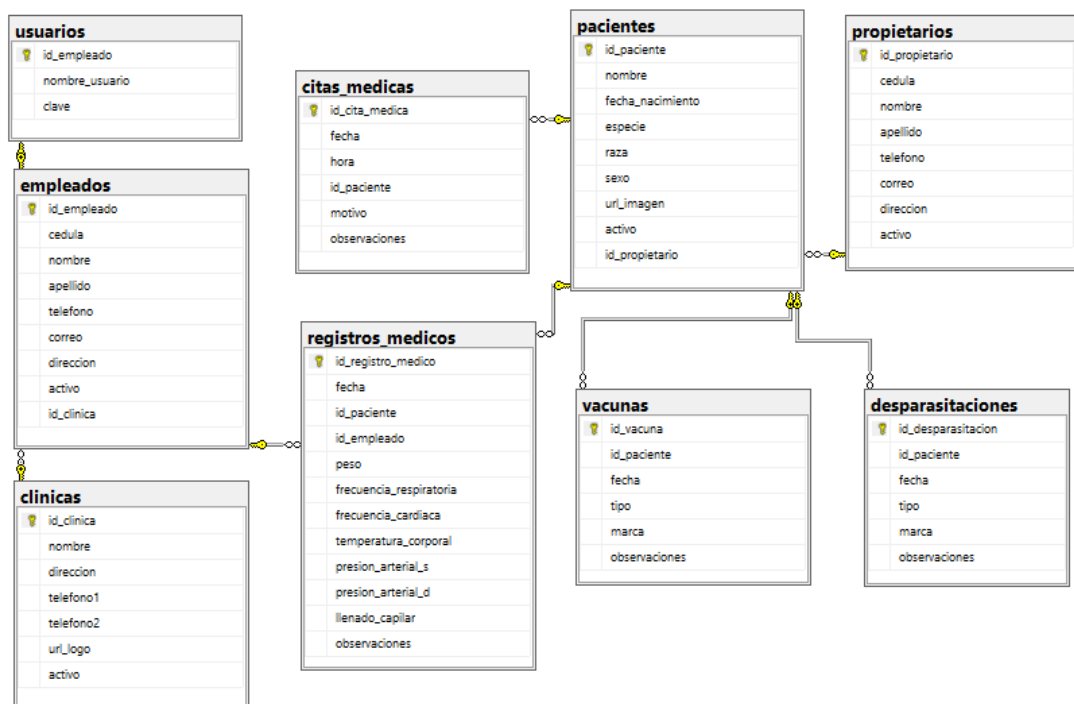


Figura 15. Modelo relacional de la base de datos.

- **Historia de usuario N°2: Implementación de la API**

La API se implementará en .NET en Visual Studio para su posterior publicación en un servidor en línea. El modelo relacional de la base de datos contribuye a la generación de entidades y microservicios que interactúen con la misma a través de operaciones CRUD (crear, leer, actualizar y eliminar).

- **Historia de usuario N°3: Gestión de datos de la clínica**

En la Figura 16 se presenta el prototipo de la sección de datos de la clínica. Aquí se incluirán los datos referentes a la identificación del establecimiento médico como nombre, dirección y teléfonos, mismos que podrán ser editados.

El prototipo muestra una interfaz de usuario en un teléfono móvil. En la parte superior, hay un título "DATOS DE LA CLÍNICA". Debajo del título, hay cinco campos de entrada de texto, cada uno con un label a la izquierda: "Nombre", "Dirección", "Teléfono 1", "Teléfono 2" y "Correo electrónico". Debajo de estos campos, hay un botón rectangular con el texto "Guardar cambios".

Figura 16. Prototipo de la sección de datos de la clínica.

b. Iteración II

La segunda iteración está conformada por dos historias de usuario: gestión de empleados y gestión de usuarios.

- **Historia de usuario N°4: Gestión de empleados**

En la Figura 17 se presenta el prototipo de la sección de empleados, mismo que contiene fichas con los datos esenciales de cada uno de ellos y un botón para agregar uno nuevo. Cada ficha tiene un botón para visualizar los datos completos del empleado; en esa pantalla se podrá editar la información o eliminar el registro (Figura 18). Para agregar o editar la información del empleado se usará un formulario (Figura 19).

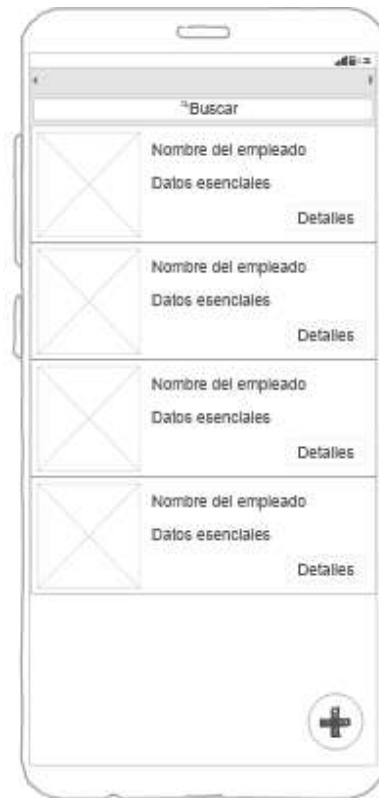


Figura 17. Prototipo de la sección de gestión de empleados.

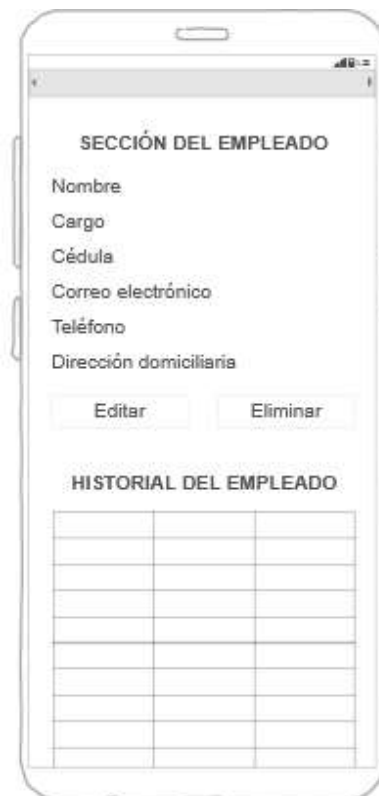


Figura 18. Prototipo del detalle del empleado.

FORMULARIO - EMPLEADO

Nombre

Cédula

Cargo

Teléfono

Correo electrónico

Dirección domiciliaria

Cancelar Aceptar

Figura 19. Prototipo del formulario del empleado.

- **Historia de usuario N°5: Gestión de usuarios**

En la Figura 20 se aprecia el formulario del empleado con los campos para el usuario añadidos. Se podrá ingresar el nombre de usuario, contraseña y el tipo de usuario.

FORMULARIO - EMPLEADO

Nombre

Cédula

Cargo

Teléfono

Correo electrónico

Dirección domiciliaria

Usuario

Contraseña

Cancelar Aceptar

Figura 20. Prototipo del formulario de usuario anexo al empleado.

c. Iteración III

La tercera iteración está conformada por dos historias de usuario: autenticación de usuarios y gestión de propietarios.

- **Historia de usuario N°6: Autenticación de usuarios**

En la Figura 21 se presenta el prototipo de la pantalla inicial para la autenticación de usuarios. Se permitirá el acceso al sistema mediante un usuario y una contraseña. También existirá un enlace para recuperar la contraseña a través de un código enviado al correo electrónico del usuario (Figura 22).



Figura 21. Prototipo de la pantalla de autenticación.



Figura 22. Prototipo de la pantalla de recuperación de contraseña.

- **Historia de usuario N°7: Gestión de propietarios**

En la Figura 23 se presenta el prototipo de la sección de propietarios, mismo que contiene fichas con los datos esenciales de cada uno de ellos y un botón para agregar uno nuevo. Cada ficha tiene un botón para visualizar los datos completos del propietario; en esa pantalla se podrá editar la información o eliminar el registro, además de añadir un paciente a cargo (Figura 24). Para agregar o editar la información del propietario se usará un formulario (Figura 25).



Figura 23. Prototipo de la sección de gestión de propietarios.



Figura 24. Prototipo del detalle del propietario.



Figura 25. Prototipo del formulario del propietario.

d. Iteración IV

La cuarta iteración está conformada por dos historias de usuario: gestión de pacientes y gestión de vacunas.

- **Historia de usuario N°8: Gestión de pacientes**

En la Figura 26 se presenta el prototipo de la sección de pacientes, mismo que contiene fichas con los datos esenciales de cada uno de ellos y un botón para agregar uno nuevo. Cada ficha tiene un botón para visualizar los datos completos del paciente; en esa pantalla se podrá editar la información o eliminar el registro, además de gestionar su información clínica (Figura 27). Para agregar o editar la información del paciente se usará un formulario (Figura 28).



Figura 26. Prototipo de la sección de gestión de pacientes.



Figura 27. Prototipo del detalle del paciente.

El prototipo muestra un formulario titulado "FORMULARIO - PACIENTE" con los siguientes campos de texto: Nombre, Fecha de nacimiento, Especie, Raza, Sexo y Propietario. En la parte inferior del formulario hay dos botones: "Cancelar" y "Aceptar".

Figura 28. Prototipo del formulario del paciente.

- **Historia de usuario N°9: Gestión de vacunas**

En la Figura 29 se presenta el prototipo de la sección de vacunas, dentro del detalle del paciente, mismo que contiene fichas con los datos esenciales de cada una de ellas y un botón para agregar una nueva. Cada ficha tiene un botón para visualizar los datos completos de la vacuna; en esa pantalla se podrá editar la información o eliminar el registro (Figura 30). Para agregar o editar la información de la vacuna se usará un formulario (Figura 31).



Figura 29. Prototipo de la sección de gestión de vacunas.



Figura 30. Prototipo del detalle de la vacuna.



Figura 31. Prototipo del formulario de la vacuna.

e. Iteración V

La quinta iteración está conformada por dos historias de usuario: gestión de desparasitaciones y gestión de citas médicas.

- **Historia de usuario N°10: Gestión de desparasitaciones**

En la Figura 32 se presenta el prototipo de la sección de desparasitaciones, dentro del detalle del paciente, mismo que contiene fichas con los datos esenciales de cada una de ellas y un botón para agregar una nueva. Cada ficha tiene un botón para visualizar los datos completos de la desparasitación; en esa pantalla se podrá editar la información o eliminar el registro (Figura 33). Para agregar o editar la información de la desparasitación se usará un formulario (Figura 34).



Figura 32. Prototipo de la sección de gestión de desparasitaciones.



Figura 33. Prototipo del detalle de la desparasitación.

El prototipo de la interfaz de usuario se muestra dentro de un marco de un teléfono móvil. En la parte superior, hay una barra de estado con un icono de señal y un icono de batería. El formulario principal tiene un título "FORMULARIO DESPARASITACIONES" en un recuadro. Debajo del título, hay cuatro campos de entrada: "Fecha", "Tipo de vacuna", "Marca" y "Observaciones". Cada campo tiene un label a la izquierda y una línea de entrada a la derecha. En la parte inferior del formulario, hay dos botones: "Cancelar" y "Aceptar".

Figura 34. Prototipo del formulario de la desparasitación.

- **Historia de usuario N°11: Gestión de citas médicas**

En la Figura 35 se presenta el prototipo de la sección de citas médicas, dentro del detalle del paciente, mismo que contiene fichas con los datos esenciales de cada una de ellas y un botón para agregar una nueva. Cada ficha tiene un botón para visualizar los datos completos de la cita médica; en esa pantalla se podrá editar la información o eliminar el registro (Figura 36). Para agregar o editar la información de la cita médica se usará un formulario (Figura 37).



Figura 35. Prototipo de la sección de gestión de citas médicas.

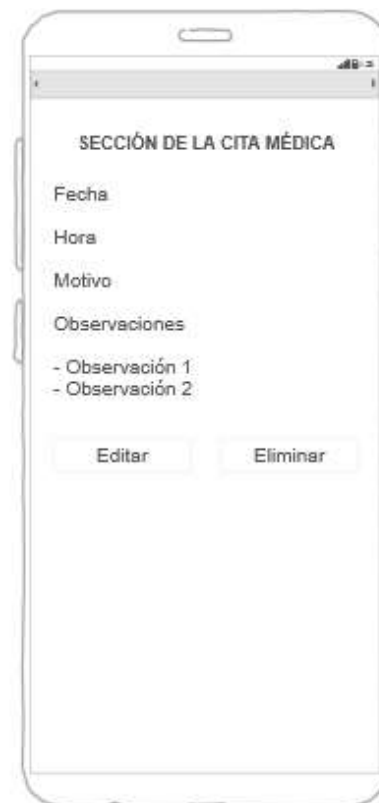


Figura 36. Prototipo del detalle de la cita médica.



Figura 37. Prototipo del formulario de la cita médica.

f. Iteración VI

La sexta iteración está conformada por tres historias de usuario: notificaciones de citas médicas, gestión de registros médicos y generación de reportes del historial clínico.

- **Historia de usuario N°12: Citas médicas programadas**

La pantalla principal de la aplicación muestra una lista de las citas programadas para el día actual y otra lista con las citas más cercanas (Figura 38). Cada ficha tendrá un botón que dirige al usuario al chat de Whatsapp del propietario (Figura 39).



Figura 38. Prototipo de la pantalla principal con las citas programadas.



Figura 39. Prototipo del mensaje predeterminado en Whatsapp.

- **Historia de usuario N°13: Gestión de registros médicos**

En la Figura 40 se presenta el prototipo de la sección de registros médicos, dentro del detalle del paciente, mismo que contiene fichas con los datos esenciales de cada uno de ellos y un botón para agregar uno nuevo. Cada ficha tiene un botón para visualizar los datos completos; en esa pantalla se podrá editar la información o eliminar el registro (Figura 41). Para agregar o editar la información del registro médico se usará un formulario (Figura 42).



Figura 40. Prototipo de la sección de gestión de registros médicos.

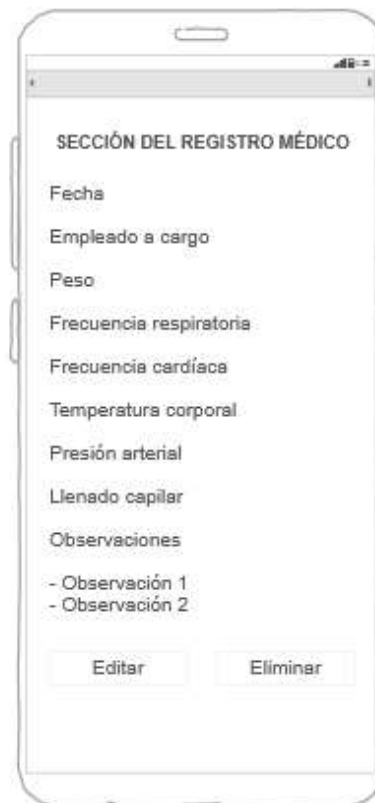


Figura 41. Prototipo del detalle del registro médico.

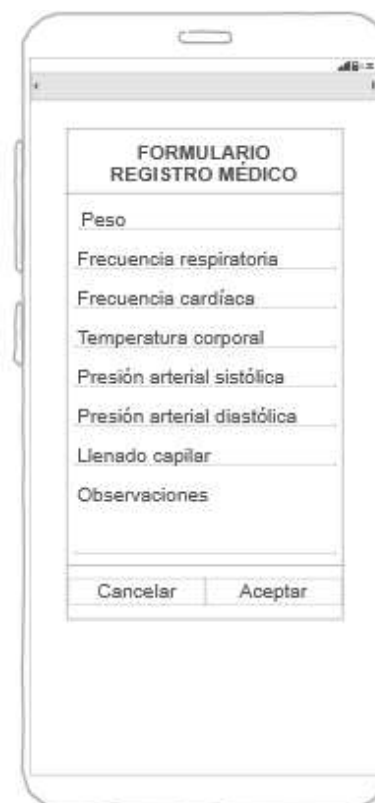


Figura 42. Prototipo del formulario del registro médico.

- **Historia de usuario N°14: Generación de reportes del historial clínico**

En la Figura 43 se presenta el prototipo de la sección de reportes clínicos, donde existirán distintas categorías. Al elegir una categoría, se generará un reporte gráfico a manera de un gráfico de líneas para apreciar la evolución del paciente en un aspecto clínico (Figura 44).

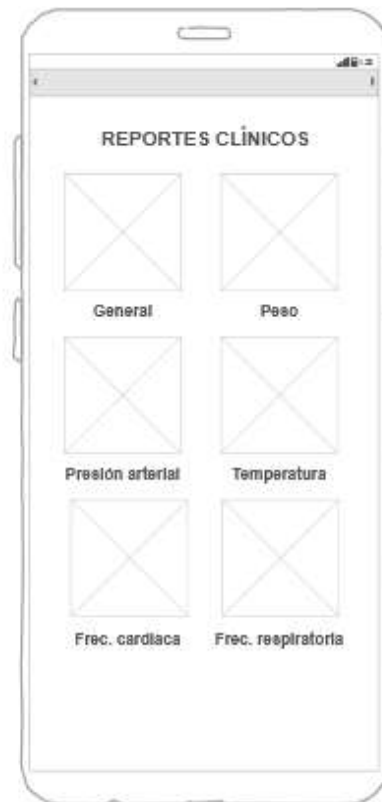


Figura 43. Prototipo de la sección de reportes clínicos.



Figura 44. Prototipo de un reporte clínico.

3.2.3 Fase III: Codificación

La fase de codificación detalla la implementación de las funcionalidades que se plantearon durante la fase previa de diseño. Con base al marco de trabajo planteado en XP, se desarrollaron los componentes del sistema a lo largo de seis iteraciones, generando una integración continua y una retroalimentación rápida.

a. Iteración I

Durante la primera iteración se realizó la implementación de la base de datos, una API y una funcionalidad del sistema referente a la gestión de datos de la clínica.

- **Historia de usuario N°1: Implementación de la base de datos**

La base de datos se realizó en SQL Server, reflejando todas las entidades que participan en la clínica veterinaria y sus respectivas relaciones. En la Figura 15, en la sección de Diseño, se muestra el esquema final de la base de datos.

- **Historia de usuario N°2: Implementación de una API**

Se realizó una API en .NET para disponer de microservicios que permitan realizar operaciones CRUD (crear, leer, actualizar y eliminar) sobre cada entidad de la base de datos. La Figura 45 muestra la interfaz de la API alojada en el servidor Somee.

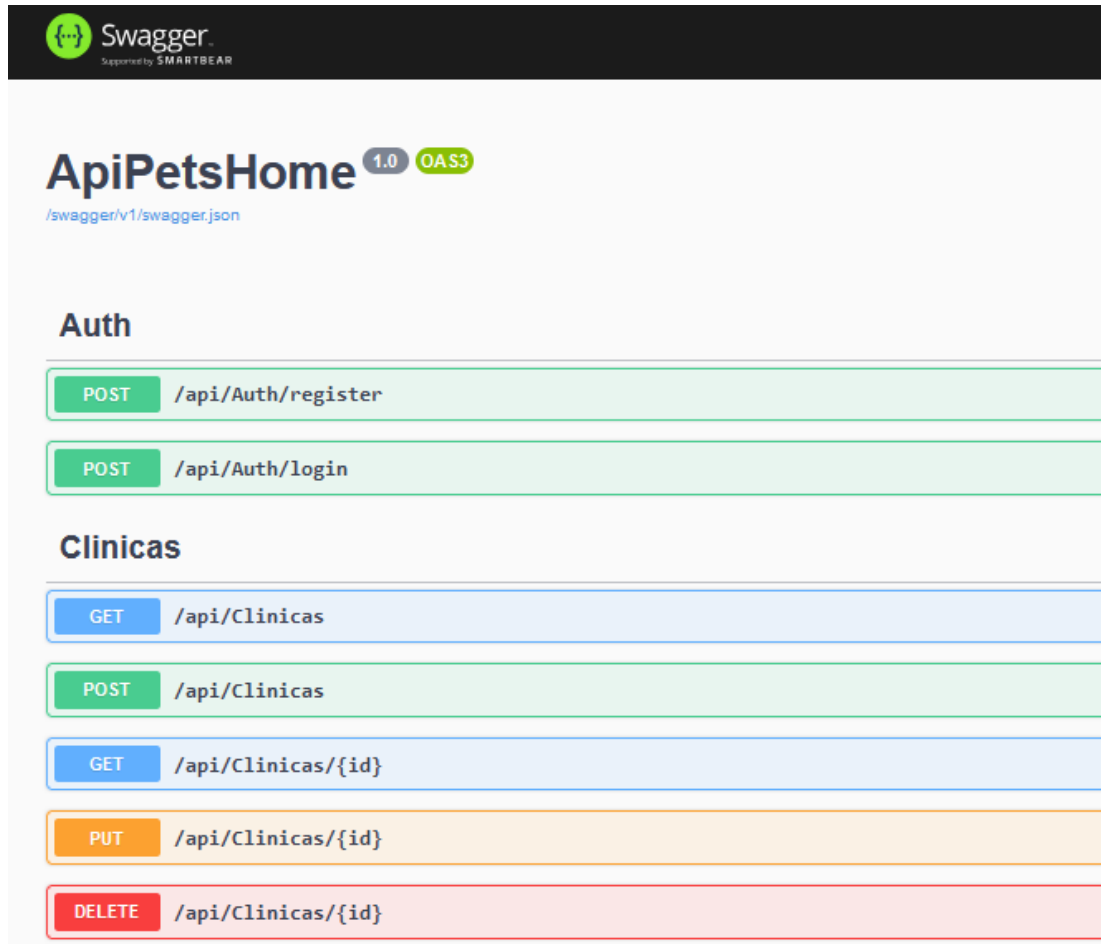


Figura 45. Interfaz de la API en el servidor Somee.

- **Historia de usuario N°3: Gestión de datos de la clínica**

Se implementó una sección para visualizar y editar los datos de la clínica veterinaria. En la Figura 46 se muestra la interfaz de la misma. Al actualizar la información directamente en la base de datos, se cumple con el requerimiento de reflejar los cambios si se utilizan los datos de la clínica en otra sección de la aplicación.

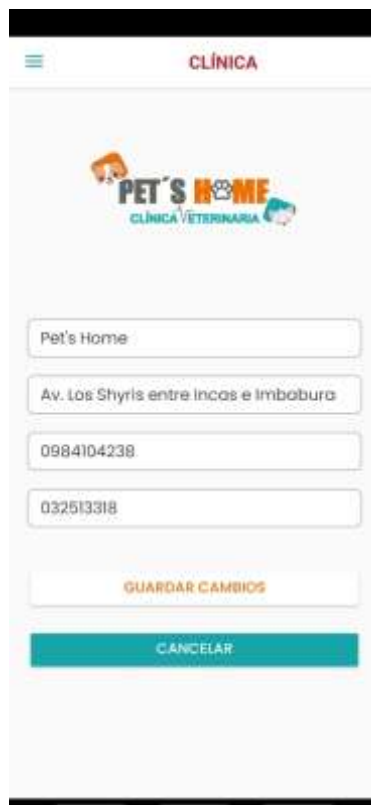


Figura 46. Interfaz de gestión de datos de la clínica.

b. Iteración II

Durante la segunda iteración se realizó la implementación de las funcionalidades de la aplicación relacionadas con la gestión de empleados y usuarios.

- **Historia de usuario N°4: Gestión de empleados**

La sección de empleados de la aplicación cuenta con fichas de información de cada empleado, una barra de búsqueda para filtrar por nombre y un botón para añadir un nuevo empleado (Figura 47). Al ingresar en los detalles del empleado, se podrá editar o eliminar su información (Figura 48).



Figura 47. Interfaz de la sección de empleados.



Figura 48. Interfaz de los detalles del empleado.

- **Historia de usuario N°5: Gestión de usuarios**

La gestión de usuarios se acopla a los formularios del empleado. Al registrar un empleado, se deben ingresar las credenciales para su acceso al sistema (Figura 49). Mientras que, para editar este nombre de usuario y contraseña, se debe acceder al formulario del empleado correspondiente (figura 50).

A screenshot of a mobile application interface showing a registration form. The form has two input fields labeled 'Usuario' and 'Contraseña'. Below these fields are two buttons: 'CANCELAR' in orange and 'ACEPTAR' in teal. The form is overlaid on a dark background with a circular icon in the bottom right corner.

Figura 49. Registro de un usuario acoplado al empleado.

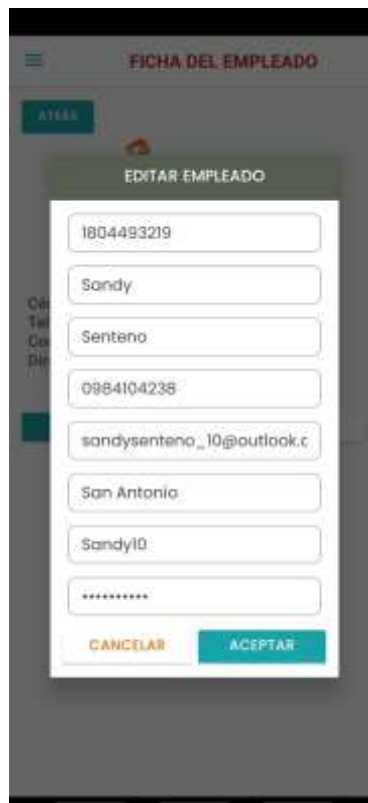
A screenshot of a mobile application interface showing an 'EDITAR EMPLEADO' (Edit Employee) form. The form is titled 'FICHA DEL EMPLEADO' and 'EDITAR EMPLEADO'. It contains several input fields with the following values: '1804493219', 'Sandy', 'Senteno', '0984104238', 'sandyenteno_10@outlook.c', 'San Antonio', 'Sandy10', and a password field with asterisks. At the bottom are 'CANCELAR' and 'ACEPTAR' buttons. The form is overlaid on a dark background with a sidebar menu on the left.

Figura 50. Edición de las credenciales asignadas a un empleado.

c. Iteración III

Durante la tercera iteración se realizó la implementación de las funcionalidades de la aplicación relacionadas con la autenticación de usuarios y la gestión de propietarios.

- **Historia de usuario N°6: Autenticación de usuarios**

La pantalla de inicio de la aplicación presenta un formulario de acceso al sistema que consta de un nombre de usuario y una contraseña (Figura 51). También encontramos un enlace para recuperar la contraseña, lo que nos lleva a un apartado para ingresar nuestro nombre de usuario o correo para recibir un enlace de recuperación (Figura 52).



Figura 51. Autenticación de usuarios.



Figura 52. Recuperación de contraseña.

- **Historia de usuario N°7: Gestión de propietarios**

La gestión de propietarios de acopla a la sección de pacientes para priorizar la información de estos últimos. Al ingresar un paciente, se podrá llenar los datos de su propietario (Figura 53). En la sección del detalle del paciente, habrá un enlace para acceder, editar o eliminar la información del propietario.



Figura 53. Registro de un propietario.

d. Iteración IV

Durante la cuarta iteración se realizó la implementación de las funcionalidades de la aplicación relacionadas con la gestión de pacientes y la gestión de vacunas.

- **Historia de usuario N°8: Gestión de pacientes**

La sección de paciente muestra fichas con los datos esenciales del paciente y un botón para añadir un nuevo registro (Figura 54). Al entrar en el detalle del paciente, podremos editar o eliminar la información del mismo. También tendremos acceso a otros apartados relacionados con el paciente (Figura 55).



Figura 54. Sección de pacientes.



Figura 55. Detalle del paciente.

- **Historia de usuario N°9: Gestión de vacunas**

En la sección del detalle del paciente podremos encontrar un enlace al registro de sus vacunas. Allí, podremos añadir una nueva vacuna y también editar o eliminar la información de las ya existentes (Figura 56).



Figura 56. Sección de gestión de vacunas.

e. Iteración V

Durante la quinta iteración se realizó la implementación de las funcionalidades de la aplicación relacionadas con la gestión de desparasitaciones y la gestión de citas médicas.

- **Historia de usuario N°10: Gestión de desparasitaciones**

En la sección del detalle del paciente podremos encontrar un enlace al registro de sus desparasitaciones. Allí, podremos añadir una nueva desparasitación y también editar o eliminar la información de las ya existentes (Figura 57).



Figura 57. Sección de gestión de desparasitaciones.

- **Historia de usuario N°11: Gestión de citas médicas**

En la sección del detalle del paciente encontramos un enlace al registro de las citas médicas que han sido programadas. Podremos observar cada cita a manera de una ficha con su información fundamental y dos botones para editar o eliminar el registro. En la parte inferior, se podrá agregar una nueva cita médica (Figura 58).



Figura 58. Sección de gestión de citas médicas.

f. Iteración VI

Durante la sexta y última iteración se realizó la implementación de las funcionalidades de la aplicación relacionadas con las citas médicas programadas, la gestión de registros médicos y la generación de reportes del historial clínico.

- **Historia de usuario N°12: Citas médicas programadas**

En la página principal de la aplicación, podremos observar las citas programadas para el día actual y las citas próximas. Cada ficha tendrá la información esencial de la cita médica y un botón para contactar al propietario vía Whatsapp (Figura 59). Se generará un mensaje predeterminado indicando los detalles de la cita médica.



Figura 59. Sección de citas médicas programadas.

- **Historia de usuario N°13: Gestión de registros médicos**

En la sección del detalle del paciente existe un enlace para acceder a los registros médicos del paciente. Cada ficha contiene información primordial del registro clínico. En la parte inferior se encuentra un botón para añadir un nuevo registro (Figura 60). Al ingresar en el detalle del registro clínico podremos visualizar todos los parámetros que han sido medidos (Figura 61).



Figura 60. Sección de gestión de registros médicos.



Figura 61. Detalle del registro médico.

- **Historia de usuario N°14: Generación de reportes del historial clínico**

En la sección del detalle del registro clínico tendremos enlaces para generar reportes de signos clínicos y mostrar su evolución en un gráfico de líneas (Figura 62).

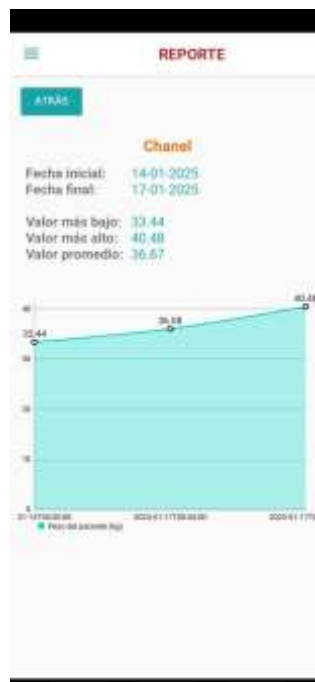


Figura 62. Ejemplo de reporte gráfico de un signo clínico.

3.2.4 Fase IV: Pruebas

La fase de pruebas asegura que cada módulo de la aplicación final funcione de acorde a lo establecido en los requerimientos del sistema. Una prueba considera los criterios de aceptación que se describieron en cada historia de usuario para considerarse o no como aprobada. La Tabla 64 muestra la estructura en donde se muestra la información de cada una de las pruebas de aceptación.

Tabla 64. Modelo de presentación de la prueba de aceptación.

Prueba de aceptación	
Número de prueba: Identificador numérico.	Número de historia de usuario: identificador numérico de la historia.
Nombre de la prueba: Mismo nombre de la historia de usuario.	
Descripción: Detalles del requerimiento que se probará.	
Condiciones de ejecución: Requerimientos para ejecutar la prueba.	
Pasos de ejecución: Proceso para ejecutar la prueba.	
Resultados esperados: Resultados con base a los criterios de aceptación.	
Evaluación: Resultado de la prueba (cumple o no cumple).	

a. Iteración I

Durante la primera iteración se realizaron las pruebas de aceptación sobre la implementación de la base de datos, una API y una funcionalidad del sistema referente a la gestión de datos de la clínica.

- **Historia de usuario N°1: Implementación de la base de datos**

La Tabla 65 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°1.

Tabla 65. Prueba de aceptación N°1.

Prueba de aceptación	
Número de prueba: 1.	Número de historia de usuario: 1.
Nombre de la prueba: Implementación de la base de datos.	
Descripción: Base de datos para almacenar y organizar toda la información de la clínica veterinaria.	
Condiciones de ejecución: Acceso al backup de la base de datos y al servidor en línea Somee.	
Pasos de ejecución: 1. Ingresar al servidor en línea Somee. 2. Restaurar la base de datos usando el backup disponible. 3. Verificar la creación de todas las estructuras definidas en la base de datos.	
Resultados esperados: - Backup para la creación de la base de datos. - Entidades correctamente relacionadas, de acorde a la lógica de funcionamiento de la clínica.	
Evaluación: Cumple.	

- **Historia de usuario N°2: Implementación de una API**

La Tabla 66 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°2.

Tabla 66. Prueba de aceptación N°2.

Prueba de aceptación	
Número de prueba: 2.	Número de historia de usuario: 2.
Nombre de la prueba: Implementación de una API.	
Descripción: Recuperación de los datos de la clínica para poder gestionarlos y mantenerlos siempre actualizados de acorde a las necesidades de la clínica.	
Condiciones de ejecución: Acceso a la API y al servidor en línea Somee.	
Pasos de ejecución: 1. Acceder al servidor Somee. 2. Publicar la API de .NET en Somee desde Visual Studio. 3. Acceder a la url de acceso a la API. 4. Verificar cada uno de los microservicios y sus respuestas.	

Resultados esperados: - API alojada en línea y en funcionamiento para acoplarse a la aplicación móvil. - Servicios probados individualmente para comprobar que los datos se manipulen de acorde a lo establecido en cada uno.
Evaluación: Cumple.

• **Historia de usuario N°3: Gestión de datos de la clínica**

La Tabla 67 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°3.

Tabla 67. Prueba de aceptación N°3.

Prueba de aceptación	
Número de prueba: 3.	Número de historia de usuario: 3.
Nombre de la prueba: Gestión de datos de la clínica.	
Descripción: Sección para visualizar y modificar los datos de la clínica veterinaria, mismos que podrían ser visibles en otro apartado de la aplicación.	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	
Pasos de ejecución: 1. Acceder a la aplicación con el usuario de prueba. 2. Dirigirse a la sección de datos de la clínica veterinaria. 3. Comprobar los datos presentados, modificarlos y guardarlos. 4. Verificar que exista un mensaje de confirmación. 5. Verificar los datos de la clínica modificados si se reflejan en otra parte de la aplicación.	
Resultados esperados: - Mensajes de advertencia antes de modificar los datos. - Los datos modificados se reflejan automáticamente en otras secciones, si se está haciendo uso de ellos.	
Evaluación: Cumple.	

b. Iteración II

Durante la segunda iteración se realizaron las pruebas de aceptación sobre las funcionalidades de gestión de empleados y usuarios.

- **Historia de usuario N°4: Gestión de empleados**

La Tabla 68 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°4.

Tabla 68. Prueba de aceptación N°4.

Prueba de aceptación	
Número de prueba: 4.	Número de historia de usuario: 4.
Nombre de la prueba: Gestión de empleados.	
Descripción: Agregar, modificar y eliminar empleados de la aplicación para gestionar al personal de la clínica.	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	
Pasos de ejecución: 1. Acceder a la aplicación con el usuario de prueba. 2. Dirigirse a la sección de empleados. 3. Verificar las funciones para agregar, editar y eliminar información del empleado.	
Resultados esperados: - Gestionar los datos de cada empleado mediante las funciones de agregar, eliminar y modificar.	
Evaluación: Cumple.	

- **Historia de usuario N°5: Gestión de usuarios**

La Tabla 69 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°5.

Tabla 69. Prueba de aceptación N°5.

Prueba de aceptación	
Número de prueba: 5.	Número de historia de usuario: 5.
Nombre de la prueba: Gestión de usuarios.	
Descripción: Asociar un usuario y contraseña al empleado para restringir el acceso al personal autorizado. Existirá un tipo de usuario por cada cargo en la clínica.	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	

Pasos de ejecución: 1. Acceder a la aplicación con el usuario de prueba. 2. Dirigirse a la sección de empleados. 3. Verificar que el formulario del empleado incluya los campos para el usuario asociado al mismo al registrar y editar la información. 4. Crear un usuario de prueba de cada tipo y verificar las secciones a las que tiene acceso cada uno.
Resultados esperados: - Mostrar las secciones y permitir las acciones correspondientes según el rol asignado. Verificar con un usuario de prueba de cada tipo.
Evaluación: Cumple.

c. Iteración III

Durante la tercera iteración se realizaron las pruebas de aceptación sobre las funcionalidades de autenticación de usuarios y gestión de propietarios.

• Historia de usuario N°6: Gestión de empleados

La Tabla 70 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°6.

Tabla 70. Prueba de aceptación N°6.

Prueba de aceptación	
Número de prueba: 6.	Número de historia de usuario: 6.
Nombre de la prueba: Autenticación de usuarios.	
Descripción: Acceso a las funcionalidades del sistema a través de credenciales (usuario y contraseña).	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	
Pasos de ejecución: 1. Ejecutar la aplicación. 2. Tratar de ingresar al sistema con credenciales incorrectas y verificar si existe un control adecuado del error. 3. Ingresar al sistema con credenciales válidas y verificar si es posible ingresar a la pantalla principal.	
Resultados esperados: - Acceso al sistema con las credenciales únicas asignadas al usuario. - Al ingresar con información errónea, el sistema deberá notificarlo.	
Evaluación: Cumple.	

- **Historia de usuario N°7: Gestión de propietarios**

La Tabla 71 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°7.

Tabla 71. Prueba de aceptación N°7.

Prueba de aceptación	
Número de prueba: 7.	Número de historia de usuario: 7.
Nombre de la prueba: Gestión de propietarios.	
Descripción: Agregar, modificar y eliminar propietarios de la aplicación para gestionar a los tutores de los pacientes.	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	
Pasos de ejecución: 1. Acceder a la aplicación con el usuario de prueba. 2. Dirigirse a la sección de pacientes y abrir el detalle de cualquier paciente. 3. Si no existe un propietario asignado, comprobar que se puede añadir uno. 4. Si existe un propietario asignado, comprobar que se puede abrir sus detalles. 5. Dentro de los detalles del propietario, verificar las funciones para editar y eliminar el propietario.	
Resultados esperados: - Gestionar los datos de cada propietario mediante las funciones de agregar, eliminar y modificar.	
Evaluación: Cumple.	

d. Iteración IV

Durante la cuarta iteración se realizaron las pruebas de aceptación sobre las funcionalidades de gestión de pacientes y gestión de vacunas.

- **Historia de usuario N°8: Gestión de pacientes**

La Tabla 72 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°8.

Tabla 72. Prueba de aceptación N°8.

Prueba de aceptación	
Número de prueba: 8.	Número de historia de usuario: 8.
Nombre de la prueba: Gestión de pacientes.	
Descripción: Agregar, modificar y eliminar pacientes de la aplicación para gestionar su información personal.	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	
Pasos de ejecución: 1. Acceder a la aplicación con el usuario de prueba. 2. Dirigirse a la sección de pacientes y añadir uno nuevo. 3. Verificar que se agrega el nuevo paciente y abrir sus detalles. 4. Verificar las funciones para editar y eliminar el paciente.	
Resultados esperados: - Gestionar los datos de cada paciente mediante las funciones de agregar, eliminar y modificar.	
Evaluación: Cumple.	

- **Historia de usuario N°9: Gestión de vacunas**

La Tabla 73 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°9.

Tabla 73. Prueba de aceptación N°9.

Prueba de aceptación	
Número de prueba: 9.	Número de historia de usuario: 9.
Nombre de la prueba: Gestión de vacunas.	
Descripción: Agregar, modificar y eliminar vacunas para gestionar su historial en cada paciente.	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	
Pasos de ejecución: 1. Acceder a la aplicación con el usuario de prueba. 2. Dirigirse a la sección de pacientes y abrir el detalle de uno de ellos. 3. Abrir la sección de vacunas y verificar los datos mostrados. 4. Agregar una nueva vacuna y comprobar que se añada a la lista del paciente. 5. Verificar las funciones para editar y eliminar la vacuna.	

Resultados esperados: - Gestionar los datos de las vacunas administradas en cada paciente mediante las funciones de agregar, eliminar y modificar.
Evaluación: Cumple.

e. Iteración V

Durante la quinta iteración se realizaron las pruebas de aceptación sobre las funcionalidades de gestión de desparasitaciones y gestión de citas médicas.

• **Historia de usuario N°10: Gestión de desparasitaciones**

La Tabla 74 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°10.

Tabla 74. Prueba de aceptación N°10.

Prueba de aceptación	
Número de prueba: 10.	Número de historia de usuario: 10.
Nombre de la prueba: Gestión de desparasitaciones.	
Descripción: Agregar, modificar y eliminar desparasitaciones para gestionar su historial en cada paciente.	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	
Pasos de ejecución: 1. Acceder a la aplicación con el usuario de prueba. 2. Dirigirse a la sección de pacientes y abrir el detalle de uno de ellos. 3. Abrir la sección de desparasitaciones y verificar que se muestren únicamente las pertenecientes a ese paciente. 4. Agregar una nueva desparasitación y comprobar que se añada a la lista del paciente. 5. Verificar las funciones para editar y eliminar la desparasitación.	
Resultados esperados: - Gestionar los datos de las desparasitaciones administradas en cada paciente mediante las funciones de agregar, eliminar y modificar.	
Evaluación: Cumple.	

- **Historia de usuario N°11: Gestión de citas médicas**

La Tabla 75 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°11.

Tabla 75. Prueba de aceptación N°11.

Prueba de aceptación	
Número de prueba: 11.	Número de historia de usuario: 11.
Nombre de la prueba: Gestión de citas médicas.	
Descripción: Agregar, modificar y eliminar citas médicas para gestionar su asignación en cada paciente.	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	
Pasos de ejecución: 1. Acceder a la aplicación con el usuario de prueba. 2. Dirigirse a la sección de pacientes y abrir el detalle de uno de ellos. 3. Abrir la sección de citas médicas y verificar que se muestren únicamente las pertenecientes a ese paciente. 4. Agregar una nueva cita médica y comprobar que se añada a la lista del paciente. 5. Verificar las funciones para editar y eliminar la cita médica.	
Resultados esperados: - Gestionar los datos de las citas médicas asignadas a cada paciente mediante las funciones de agregar, eliminar y modificar.	
Evaluación: Cumple.	

f. Iteración VI

Durante la sexta y última iteración se realizaron las pruebas de aceptación sobre las funcionalidades de citas médicas programadas, gestión de registros médicos y generación de reportes del historial clínico.

- **Historia de usuario N°12: Citas médicas programadas**

La Tabla 76 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°12.

Tabla 76. Prueba de aceptación N°12.

Prueba de aceptación	
Número de prueba: 12.	Número de historia de usuario: 12.
Nombre de la prueba: Citas médicas programadas.	
Descripción: Visualización de las citas médicas que están próximas a suceder y la opción de enviar un mensaje por Whatsapp al propietario.	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	
Pasos de ejecución: 1. Acceder a la aplicación con el usuario de prueba. 2. En la pantalla principal deberán mostrarse las citas médicas del día actual (si existen) y las próximas a suceder. 3. Seleccionar el botón de contacto presente en la ficha y comprobar que se redirige al chat de Whatsapp del propietario a cargo del paciente que tiene la cita programada. 4. El mensaje debe contener los datos de la cita médica como recordatorio.	
Resultados esperados: - Sección para ver las citas médicas programadas para el día actual y el día siguiente en la pantalla principal. - Añadir un botón a cada cita programada que permita abrir el chat de Whatsapp del propietario con un mensaje predeterminado.	
Evaluación: Cumple.	

• **Historia de usuario N°13: Gestión de registros médicos**

La Tabla 77 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°13.

Tabla 77. Prueba de aceptación N°13.

Prueba de aceptación	
Número de prueba: 13.	Número de historia de usuario: 13.
Nombre de la prueba: Gestión de registros médicos.	
Descripción: Agregar, modificar y eliminar registros médicos para gestionar el historial clínico de cada paciente.	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	

Pasos de ejecución: 1. Acceder a la aplicación con el usuario de prueba. 2. Dirigirse a la sección de pacientes y abrir el detalle de uno de ellos. 3. Abrir la sección de registros médicos y verificar que se muestren únicamente las pertenecientes a ese paciente. 4. Agregar un nuevo registro médico y comprobar que se añada a la lista del paciente. 5. Verificar las funciones para editar y eliminar el registro médico.
Resultados esperados: - Gestionar los datos de los registros médicos asignados a cada paciente mediante las funciones de agregar, eliminar y modificar.
Evaluación: Cumple.

• **Historia de usuario N°14: Generación de reportes del historial clínico**

La Tabla 78 presenta los resultados de la prueba de aceptación realizada sobre la historia de usuario N°14.

Tabla 78. Prueba de aceptación N°14.

Prueba de aceptación	
Número de prueba: 14.	Número de historia de usuario: 14.
Nombre de la prueba: Generación de reportes del historial clínico.	
Descripción: Generar un reporte de un signo clínico seleccionado en un gráfico para visualizar su evolución.	
Condiciones de ejecución: Aplicación instalada en un dispositivo Android y un usuario de prueba para acceder.	
Pasos de ejecución: 1. Acceder a la aplicación con el usuario de prueba. 2. Dirigirse a la sección de pacientes y abrir el detalle de uno de ellos. 3. Abrir la sección de registros médicos y comprobar que existe una sección con los reportes que se pueden generar. 4. Seleccionar cada uno de los reportes y verificar los datos mostrados.	
Resultados esperados: - Generar un gráfico que refleje la evolución de los valores de un signo clínico seleccionado dentro del historial médico del paciente.	
Evaluación: Cumple.	

3.2.5 Fase V: Lanzamiento

Tras obtener resultados positivos en las pruebas de aceptación, inicia la fase de lanzamiento, misma que consta de la preparación del entorno de producción, una revisión final de funcionamiento, capacitación de todos los usuarios y la configuración de usuarios de acceso.

a. Preparación del entorno de producción

La base de datos y la API se encuentran disponibles en el servidor en línea Somee. La base de datos se restaura a partir de un backup completo de la misma, garantizando la integridad de todas las estructuras. Por su parte, desde el asistente integrado de Visual Studio, la API en .NET se publica directamente en el servidor mediante sus credenciales. Las pruebas de aceptación reflejan el correcto funcionamiento de los microservicios y la gestión de la información almacenada.

En Android Studio, se genera un archivo .apk de la aplicación final para instalarla en todos los dispositivos de los usuarios finales. Cada dispositivo Android fue configurado para aceptar la instalación de aplicaciones de fuentes externas.

b. Revisión final de funcionamiento

Con la intervención de un usuario final se verificó el cumplimiento de todos los requerimientos del sistema que se habían planteado. Durante esta revisión, se constató que todos los módulos de la aplicación funcionaran de forma correcta, al igual que durante las pruebas de aceptación.

c. Configuración de usuarios de acceso

A través del dispositivo y credenciales del usuario final que figuraría como administrador, se registró a cada miembro del personal, otorgándole a su vez las credenciales de acceso para cada uno de ellos.

d. Capacitación de usuarios

Mediante una reunión con todo el personal de la clínica, se mostró el funcionamiento general de la aplicación y las restricciones de cada tipo de usuario. Cada usuario tuvo la oportunidad de realizar varias acciones en los diferentes módulos de la aplicación, a la vez que realizaban preguntas acerca del funcionamiento específico de algún aspecto.

e. Prueba de uso en el entorno real

Una vez que todos los usuarios mostraron un dominio sobre las funcionalidades de la aplicación, se la puso a prueba durante las visitas de pacientes de un día convencional. Todos los datos relacionados con el paciente y el propietario a cargo pudieron ser gestionados de manera exitosa. También existió la posibilidad de comprobar la disponibilidad del sistema fuera de clínica, al realizar una consulta en un domicilio externo. La aplicación funcionó correctamente gracias a que todo el personal cuenta con acceso a datos móviles.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- La adopción de Extreme Programming (XP) como metodología de desarrollo permitió un proceso iterativo y adaptable, facilitando la implementación de mejoras continuas durante el desarrollo de la aplicación. Gracias a esta metodología, se logró una entrega progresiva de funcionalidades, lo que permitió realizar pruebas tempranas y asegurar que la aplicación cumpliera con las necesidades operativas de la veterinaria Pet's Home.
- La recopilación detallada de los requerimientos a través de entrevistas y análisis del flujo de trabajo permitió diseñar una aplicación alineada con las necesidades del personal veterinario. Esto permitió abordar problemas previos como la duplicidad de registros, pérdida de información y dificultad de acceso, ofreciendo una solución más estructurada y eficiente para la gestión de datos clínicos.
- La implementación de la aplicación móvil logró optimizar la gestión de historiales clínicos, citas médicas, vacunas y desparasitaciones, centralizando la información en una plataforma digital eficiente. La digitalización de los procesos clínicos permitió agilizar la toma de decisiones mediante un acceso rápido y estructurado a los datos de los pacientes.

4.2 Recomendaciones

- Se recomienda implementar programas de capacitación para el personal de la clínica en el uso de la aplicación móvil y la interpretación de los reportes clínicos generados, asegurando el aprovechamiento pleno del sistema.
- Ampliar las funcionalidades de la aplicación para incluir recordatorios automáticos de citas, integración con sistemas de pago electrónico y notificación de vencimientos de vacunas.
- Establecer un plan de mantenimiento periódico para la aplicación y la base de datos, garantizando la continuidad operativa y la actualización de las tecnologías utilizadas.

- Considerar el uso de dispositivos IoT (Internet de las Cosas) para el monitoreo en tiempo real de signos vitales de los pacientes, lo que podría enriquecer los datos disponibles en el sistema.
- Planificar la adaptación del sistema para futuras expansiones, como la integración con múltiples clínicas o su uso en otras ciudades, promoviendo la replicabilidad del modelo.
- Realizar evaluaciones periódicas del impacto del sistema en la operatividad de la clínica y la satisfacción del cliente, lo que permitirá identificar áreas de mejora y nuevas oportunidades de desarrollo.

REFERENCIAS BIBLIOGRÁFICAS

- [1] M. Solano y H. Rivadeneira, “Evaluación del grado de bienestar animal de los perros en las cuatro áreas de salud del cantón Cuenca, utilizando un test basado en las cinco libertades de los animales”, Tesis de pregrado, Universidad de Cuenca, Cuenca, 2015.
- [2] A. Orozco, “Implementación de aplicación web para el registro y control de perros extraviados utilizando código QR en el centro veterinario Animedica”, Tesis de grado, Universidad Agraria del Ecuador, Milagro, 2022.
- [3] A. Taípe y B. Quishpe, “Desarrollo de un sistema de gestión veterinaria, mediante el modelo Api_Rest y el framework ReactJs como herramientas de software libre para el consultorio Visecpro del cantón Latacunga”, Tesis de pregrado, Universidad Técnica de Cotopaxi, Ecuador, 2022.
- [4] V. Espinosa y A. Gaguancela, “Sistema de gestión para la clínica veterinaria de la Universidad Central del Ecuador”, Tesis de pregrado, Universidad Central del Ecuador, Ecuador, 2012.
- [5] A. Cedeno, A. Catuto y J. Rodas, “El uso de aplicaciones Web para la Gestión de clínicas veterinarias y su incidencia en la mejora de procesos administrativos”, Ecuadorian Science Journal, 5(4), 109-120, 2021.
- [6] R. Martínez, “Desarrollo de una aplicación web para la gestión veterinaria en el Aquarium Finisterrae”, Tesis de pregrado, Universidade Da Coruña, España, 2023.
- [7] H. Campana, “Integración de software de gestión veterinaria ZooVet con un gestor de reservas online Bookitit”, Universitat Autònoma de Barcelona, España, 2020.
- [8] J. Coloma y J. Agualongo, “Innovación móvil para el seguimiento eficiente del ganado bovino en Hacienda San Gabriel”, Pol. Con. (Edición núm. 82) Vol. 8, No 5, 2023.

- [9] J. Reyna, S. Cieza, O. Alcántara y J. Torres, “Aplicación móvil multiplataforma para mejorar la gestión de ventas en la veterinaria Janavet de Trujillo”, Universidad Privada César Vallejo, Perú, 2021.
- [10] J. Molina, “Implementación de un sistema web para el control de procesos de la clínica veterinaria Mascotas”, Tesis de pregrado, Universidad Estatal Península de Santa Elena, Ecuador, 2016.
- [11] B. Vernaza, “Implementación de aplicación móvil para Android o IOS con realidad aumentada y geolocalización para asistencia y generación de citas en veterinarias del sur de Guayaquil sincronizado con gestor de contenido web publicitario”, Tesis de pregrado, Universidad de Guayaquil, 2015.
- [12] A. Catuto, “Desarrollo de una aplicación web de gestión administrativa para clínicas veterinarias”, Universidad Estatal de Milagro, Ecuador, 2021.
- [13] Y. Fuentes, "Healthy Pets, Sistema de Gestión de Información para uso Veterinario", Tesis de máster, Universidad Internacional de la Rioja, Bogotá, Colombia, 2016.
- [14] F. Grados y F. Rodríguez, "Aplicativo móvil para la gestión de servicios clínicos veterinarios en Lima – Metropolitana”, Tesis de pregrado, Universidad Autónoma del Perú, Lima, Perú, 2020.
- [15] A. Naranjo y J. Saltos, “Aplicación web y móvil para la gestión de los procesos de citas médicas, historial clínico y publicidad en la veterinaria Durán: Propuesta tecnológica”, Tesis de pregrado, Universidad Agraria del Ecuador, Guayaquil, Ecuador, 2021.
- [16] J. Guzmán, “Técnicas de la investigación de campo”, Recurso interactivo, Universidad Nacional Autónoma de México, 2020.
- [17] J. Vilanova, “Revisión bibliográfica del tema de estudio de un proyecto de investigación”, Radiología, 54(2), 108-114, 2012.

ANEXOS

Anexo A. Formulario de la encuesta al personal de la clínica.

Encuesta Pet's Home

¿Con qué frecuencia experimenta dificultades al acceder o encontrar información en los registros clínicos de la veterinaria?

- ☐ Nunca
- ☐ Raramente
- ☐ Ocasionalmente
- ☐ Frecuentemente
- ☐ Siempre

¿Cuánto tiempo dedica actualmente a generar el registro clínico de un paciente?

- ☐ 1-2 minutos
- ☐ 3-4 minutos
- ☐ 5 minutos
- ☐ 6-10 minutos
- ☐ Más de 10 minutos

¿Qué tan satisfecho está con la eficiencia actual de la gestión de registros clínicos en la clínica veterinaria?

- ☐ Muy insatisfecho
- ☐ Insatisfecho
- ☐ Neutral
- ☐ Satisfecho
- ☐ Muy satisfecho

¿Encuentra que el manejo manual de registros clínicos afecta la calidad del servicio que se brinda a los clientes y pacientes?

- ☐ Totalmente en desacuerdo
- ☐ En desacuerdo
- ☐ Neutral
- ☐ De acuerdo
- ☐ Totalmente de acuerdo

¿Considera importante que la aplicación de gestión integre calculadoras de uso veterinario?

- ☐ Totalmente en desacuerdo
- ☐ En desacuerdo
- ☐ Neutral
- ☐ De acuerdo
- ☐ Totalmente de acuerdo

¿Considera importante que la aplicación móvil permita el acceso a los registros clínicos fuera del consultorio veterinario?

- ☐ Nada importante
- ☐ Poco importante
- ☐ Neutral
- ☐ Importante
- ☐ Muy importante

¿Qué tan crítico considera que es el factor de seguridad al elegir una aplicación móvil de uso veterinario?

- ☐ Nada crítico
- ☐ Poco Crítico
- ☐ Neutral
- ☐ Crítico
- ☐ Muy crítico

¿Qué tan importante cree que es la generación automática de informes detallados desde la aplicación?

- ☐ Nada importante
- ☐ Poco importante
- ☐ Neutral
- ☐ Importante
- ☐ Muy importante

En su experiencia, ¿cómo calificaría la facilidad de uso de una aplicación móvil frente a un registro manual?

- ☐ Mucho más difícil
- ☐ Más difícil
- ☐ Uso similar
- ☐ Más fácil
- ☐ Mucho más fácil

¿Estaría dispuesto(a) a invertir tiempo en capacitación para aprender a utilizar una nueva aplicación móvil de gestión de registros clínicos?

- ☐ No
- ☐ No estoy seguro(a)
- ☐ Sí

Levantamiento de requerimientos conforme a
la norma IEEE 830

PROYECTO:

IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE
PACIENTES DE LA CLÍNICA VETERINARIA PET'S HOME DE LA CIUDAD
DE AMBATO.

Ficha del documento

Documento validado por las partes en la fecha: 18/10/2024

Tabla B1. Firmas de validación.

Desarrollador	Propietaria de Pet's Home
Christian Jaramillo	Sandy Senteno

ÍNDICE

1. Introducción.....	122
1.1. Propósito.....	123
1.2. Alcance.....	123
1.3. Personal involucrado.....	123
1.4. Definiciones, acrónimos y abreviaturas.....	124
1.5. Referencias.....	125
1.6. Resumen.....	125
2. Descripción general.....	126
2.1. Funcionalidad del producto.....	126
2.2. Características de los usuarios.....	127
2.3. Restricciones.....	128
2.4. Suposiciones y dependencias.....	128
3. Requisitos específicos.....	129
3.1. Requerimientos Funcionales.....	129
3.2. Requerimientos No Funcionales.....	136
3.3. Requisitos comunes de las interfaces.....	138
3.3.1. Interfaces de usuario	138
3.3.2. Interfaces de hardware	139
3.3.3. Interfaces de software	139
3.3.4. Interfaces de comunicación.....	139
3.4. Requisitos de Rendimiento.....	140
3.5. Restricciones de Diseño.....	140
3.6. Atributos del Sistema.....	141

1. Introducción

El presente documento tiene como objetivo principal el proceso de levantamiento de requisitos para el desarrollo de un sistema de gestión de pacientes destinado a la clínica veterinaria Pet's Home de la ciudad de Ambato, Ecuador. Este sistema se presentará como una aplicación móvil para el sistema operativo Android, mismo que se destina a registrar y gestionar datos acerca de los pacientes de la veterinaria y sus propietarios. Para la finalidad en cuestión, nos regiremos a la norma IEEE 830, que proporciona una guía para documentar los requisitos de un software, lo que los presenta de forma clara con base a la identificación de las necesidades específicas de los usuarios finales.

1.1. Propósito

Establecer los requisitos funcionales y no funcionales para el desarrollo de un sistema móvil de gestión de datos de los pacientes de la clínica veterinaria Pet's Home de la ciudad de Ambato, Ecuador. El aplicativo será utilizado por el personal de la clínica para mejorar la gestión de información referente a datos informativos (pacientes y propietarios) y registros clínicos de los pacientes (vacunas, desparasitaciones, perfil clínico, citas médicas, enfermedades previas, exámenes médicos). La definición precisa de los requerimientos del sistema favorece a su correcta implementación, asegurando una aplicación de calidad que cumpla con las necesidades de la clínica veterinaria.

1.2. Alcance

La especificación de requisitos está dirigida a la clínica veterinaria Pet's Home, con el objetivo de gestionar toda la información derivada de los pacientes del establecimiento. La aplicación móvil englobará un sistema integrado que permitirá al personal de la clínica, registrar, modificar y eliminar (cambiar estado de visualización) la información relacionada con los datos del paciente, propietario, administración de vacunas y desparasitantes, historial de citas médicas e historial clínico. A su vez, se dispondrá de una funcionalidad para generar reportes acerca de la evolución de los signos clínicos de un paciente para asistir la toma de decisiones del profesional veterinario.

1.3. Personal involucrado

Tabla B2. Personal involucrado 1.

Nombre	Sandy Senteno.
Rol	Administradora del sistema.
Categoría Profesional	Médica veterinaria y zootecnista.
Responsabilidad	Validar el aplicativo final.
Información de contacto	petshome_ec@outlook.com

Tabla B3. Personal involucrado 2.

Nombre	Christian Jaramillo.
Rol	Desarrollador.
Categoría Profesional	Ingeniería en software.
Responsabilidad	Generar el sistema y la documentación del proyecto.
Información de contacto	christianjaramillo13@hotmail.com

Tabla B4. Personal involucrado 3.

Nombre	Carlos Núñez.
Rol	Director de desarrollo del proyecto.
Categoría Profesional	Ingeniero en computación y sistemas de información.
Responsabilidad	Revisar y validar el sistema y la documentación del proyecto.
Información de contacto	ci.nunez@uta.edu.ec

1.4. Definiciones, acrónimos y abreviaturas

Tabla B5. Definiciones, acrónimos y abreviaturas.

Nombre	Descripción
RF	Requerimiento funcional
RNF	Requerimiento no funcional

1.5. Referencias

Tabla B6. Referencias

Título	Referencia
Norma IEEE 830	IEEE

1.6. Resumen

El contenido de este documento se divide en tres segmentos fundamentales, cada uno enfocada en un aspecto relevante del proyecto. El primer segmento se dedica a definir los recursos necesarios y los objetivos iniciales, proporcionando una visión clara del marco en el que se desarrollará el sistema.

En el segundo segmento se exploran las características generales del sistema, con énfasis en las funciones esenciales que cubrirán las demandas del usuario. Además, se analizan los datos que el sistema manejará, junto con los principales factores externos y condiciones que podrían influir en su desarrollo.

El tercer y último segmento se centra en detallar los requerimientos técnicos y funcionales del sistema. Esta parte actúa como una guía práctica para garantizar que el producto final cumpla con las expectativas y necesidades del usuario, estableciendo los parámetros para una implementación exitosa.

2. Descripción general

El sistema de gestión de pacientes para la clínica veterinaria Pet's Home será una aplicación móvil desarrollada con Java para el sistema operativo Android. Su función primordial es registrar toda la información en relación a los pacientes de la clínica, asegurando su correcta gestión y accesibilidad. La aplicación permitirá registrar, modificar y cambiar el estado de visualización de todos los datos en cuestión según el rol de usuario asignado. El administrador tendrá la capacidad de monitorear cualquier cambio en la información, gestionar al personal de la clínica y generar reportes para la toma de decisiones interna.

2.1. Funcionalidad del producto

En la Figura X se presenta el diagrama de casos de uso del sistema.

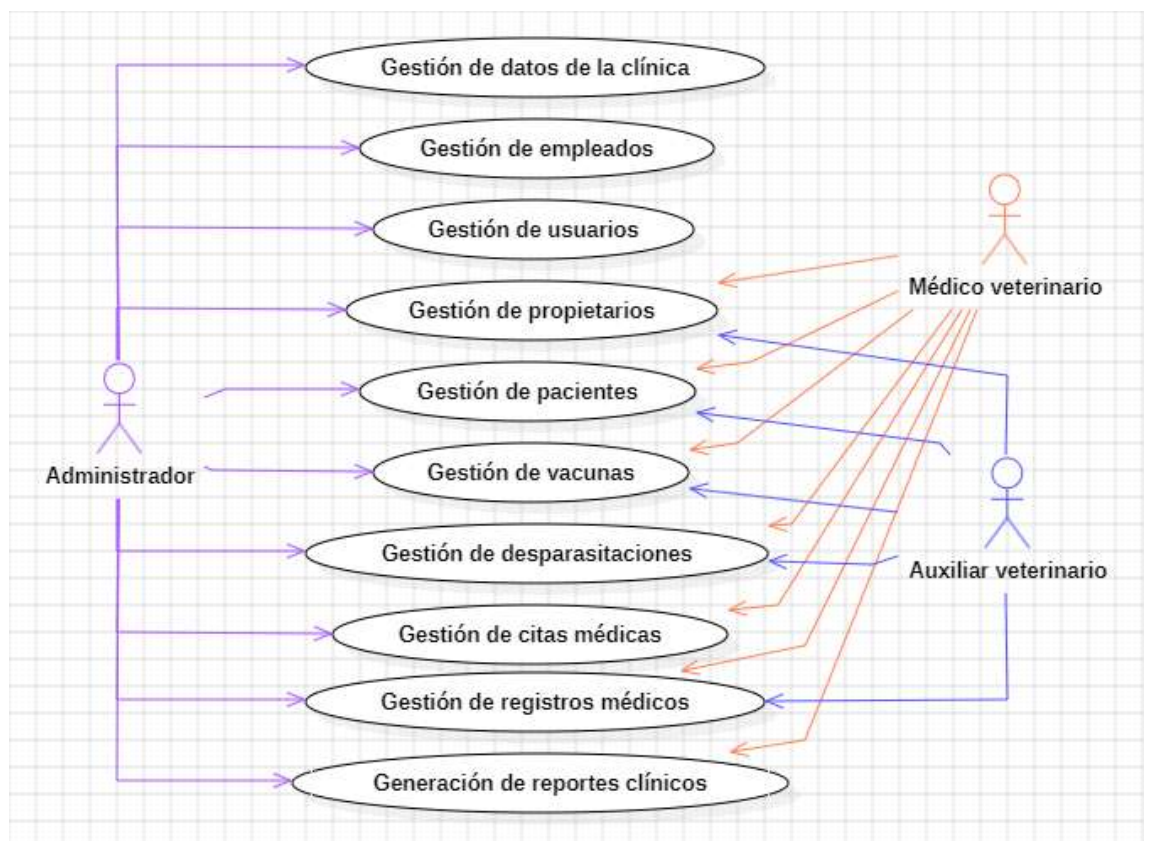


Figura X. Diagrama de casos de uso del sistema.

2.2. Características de los usuarios

Tabla B7. Rol 1

Tipo de usuario	Administrador / Administradora
Formación	Medicina veterinaria
Actividades	• Gestionar la información de la clínica veterinaria. • Crear, modificar y eliminar cuentas de usuario y empleados. • Crear, modificar y eliminar propietarios y pacientes. • Crear, modificar y eliminar registros médicos, vacunas, desparasitaciones y citas de los pacientes. • Visualizar el historial de modificaciones de cualquier registro en la aplicación. • Visualizar reportes de los datos clínicos de los pacientes.

Tabla B8. Rol 2

Tipo de usuario	Médico veterinario / Médica veterinaria
Formación	Medicina veterinaria
Actividades	• Crear, modificar y eliminar propietarios y pacientes. • Crear, modificar y eliminar registros médicos, vacunas, desparasitaciones y citas de los pacientes. • Visualizar reportes de los datos clínicos de los pacientes.

Tabla B9. Rol 3

Tipo de usuario	Auxiliar veterinario / Auxiliar veterinaria
Formación	Medicina veterinaria
Actividades	• Crear y modificar propietarios y pacientes. • Crear y modificar registros médicos, vacunas y desparasitaciones de los pacientes.

2.3. Restricciones

- Sistema operativo Android.
- Integración de roles.
- Seguridad de los datos.
- Escalabilidad.
- Disponibilidad del servicio.

2.4. Suposiciones y dependencias

- El desarrollador tendrá acceso a las herramientas tecnológicas necesarios, incluyendo hardware y software compatibles con el desarrollo para Android.
- Se presupone que habrá recursos humanos disponibles con el conocimiento y las habilidades necesarias para el desarrollo de aplicaciones móviles con Java para en sistemas operativo Android.
- Se espera la participación del personal de la clínica veterinaria Pet's Home para la definición de los requerimientos, pruebas y validación de la aplicación móvil.
- Se considera que se tendrá acceso integral a los registros de datos de la clínica veterinaria que actualmente se presentan en folios físicos.

3. Requisitos específicos

3.1. Requerimientos Funcionales

Tabla B10. RF01.

Identificación del requerimiento:	RF01.
Nombre del requerimiento:	Implementación de una base de datos.
Características:	Organización y gestión de datos de la clínica.
Descripción del requerimiento:	El sistema debe estar enlazado a una base de datos que contenga toda la información que maneja la clínica veterinaria (personal, pacientes, propietarios, registros médicos en general). Los datos deben estar correctamente organizados para permitir el fácil acceso.
Requerimiento NO funcional:	RNF01, RNF02, RNF04.
Prioridad del requerimiento:	Alta.

Tabla B11. RF02.

Identificación del requerimiento:	RF02.
Nombre del requerimiento:	Implementación de una API.
Características:	Comunicación entre la base de datos y la aplicación móvil.
Descripción del requerimiento:	El sistema debe comunicarse con la base de datos a través de una API que contenga los servicios necesarios para realizar operaciones CRUD (crear, leer, actualizar y eliminar), así como manejar la autenticación de usuarios.
Requerimiento NO funcional:	RNF01, RNF02, RNF04.
Prioridad del requerimiento:	Alta.

Tabla B12. RF03.

Identificación del requerimiento:	RF03.
Nombre del requerimiento:	Gestión de datos de la clínica.
Características:	Visualización y modificación de los datos de la clínica.
Descripción del requerimiento:	El sistema dispondrá de un apartado para ver y editar los datos de la clínica: nombre, dirección y teléfonos.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Media.

Tabla B13. RF04.

Identificación del requerimiento:	RF04.
Nombre del requerimiento:	Gestión de empleados.
Características:	Visualización, registro, modificación y eliminación de empleados.
Descripción del requerimiento:	El sistema permitirá al usuario Administrador agregar, editar y eliminar información de los empleados. Estos datos incluirán el nombre, cédula, nombre, apellido, teléfono, correo y dirección domiciliaria. Cada empleado podrá tener asignado un usuario.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Alta.

Tabla B14. RF05.

Identificación del requerimiento:	RF05.
Nombre del requerimiento:	Gestión de usuarios.
Características:	Asociar un usuario y contraseña a un empleado para permitir el acceso únicamente al personal autorizado.
Descripción del requerimiento:	Cada empleado dispondrá de un usuario y contraseña para acceder al sistema. Según su cargo en la veterinaria, podrá tener acceso a las secciones de la aplicación.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Alta.

Tabla B15. RF06.

Identificación del requerimiento:	RF06.
Nombre del requerimiento:	Autenticación de usuarios.
Características:	Inicio de sesión seguro para acceder al sistema.
Descripción del requerimiento:	Función de autenticación que garantice que los usuarios registrados puedan acceder de forma segura al sistema mediante credenciales (nombre de usuario y la contraseña). En caso de que las credenciales no sean correctas, se mostrará un mensaje de error.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Alta.

Tabla B16. RF07.

Identificación del requerimiento:	RF07.
Nombre del requerimiento:	Gestión de propietarios.
Características:	Visualización, registro, modificación y eliminación de propietarios de los pacientes.
Descripción del requerimiento:	Los usuarios autorizados podrán agregar, editar y eliminar datos de los propietarios. Estos datos incluirán el nombre, apellido, cédula, teléfono, correo y dirección. Un propietario puede tener varios pacientes registrados.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Alta.

Tabla B17. RF08.

Identificación del requerimiento:	RF08.
Nombre del requerimiento:	Gestión de pacientes.
Características:	Visualización, registro, modificación y eliminación de los pacientes.
Descripción del requerimiento:	Los usuarios autorizados podrán agregar, editar y eliminar datos de los propietarios. Estos datos incluirán el nombre, fecha de nacimiento, especie, raza y sexo. Cada paciente deberá tener un único tutor o propietario.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Alta.

Tabla B18. RF09.

Identificación del requerimiento:	RF09.
Nombre del requerimiento:	Gestión de vacunas.
Características:	Visualización, registro, modificación y eliminación de las vacunas.
Descripción del requerimiento:	Los usuarios autorizados podrán agregar, editar y eliminar datos de las vacunas que ha recibido cada paciente. Estos datos incluirán la fecha, el tipo de vacuna, la marca y observaciones opcionales.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Alta.

Tabla B19. RF10.

Identificación del requerimiento:	RF10.
Nombre del requerimiento:	Gestión de desparasitaciones.
Características:	Visualización, registro, modificación y eliminación de las desparasitaciones.
Descripción del requerimiento:	Los usuarios autorizados podrán agregar, editar y eliminar datos de los desparasitantes que ha recibido cada paciente. Estos datos incluirán la fecha, el tipo de desparasitante, la marca y observaciones opcionales.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Alta.

Tabla B20. RF11.

Identificación del requerimiento:	RF11.
Nombre del requerimiento:	Gestión de citas médicas.
Características:	Visualización, registro, modificación y eliminación de las citas médicas.
Descripción del requerimiento:	Los usuarios autorizados podrán agregar, editar y eliminar datos de las citas médicas. Estos datos incluirán la fecha, hora, motivo y observaciones opcionales.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Alta.

Tabla B21. RF12.

Identificación del requerimiento:	RF12.
Nombre del requerimiento:	Citas médicas programadas.
Características:	Visualización de las citas médicas programadas del día actual y siguiente. Opción para contactar al propietario vía Whatsapp.
Descripción del requerimiento:	En la pantalla principal de la aplicación se mostrarán las citas médicas planificadas para el día actual y el día siguiente. Cada ficha tendrá un botón de acceso al chat de Whatsapp del propietario con un mensaje predeterminado.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Media.

Tabla B22. RF13.

Identificación del requerimiento:	RF13.
Nombre del requerimiento:	Gestión de registros médicos.
Características:	Visualización, registro, modificación y eliminación de los registros clínicos.
Descripción del requerimiento:	Los usuarios autorizados podrán agregar, editar y eliminar datos de los registros clínicos de un paciente. Estos datos incluirán la fecha, peso, frecuencia respiratoria, frecuencia cardiaca, temperatura corporal, presión arterial, llenado capilar y observaciones opcionales.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Alta.

Tabla B23. RF14.

Identificación del requerimiento:	RF14.
Nombre del requerimiento:	Generación de reportes del historial clínico.
Características:	Visualización de gráficos de evolución de signos clínicos de un paciente.
Descripción del requerimiento:	El sistema deberá mostrar gráficos con los cambios reportados en los signos clínicos (por ejemplo, el peso corporal). Estos favorecerán a la toma de decisiones interna.
Requerimiento NO funcional:	RNF01, RNF02, RNF03, RNF04, RNF05.
Prioridad del requerimiento:	Media.

3.2. Requerimientos No Funcionales

Tabla B24. RNF01.

Identificación del requerimiento:	RNF01.
Nombre del requerimiento:	Seguridad del sistema.
Características:	Autenticación de usuarios y protección de datos.
Descripción del requerimiento:	El acceso debe estar restringido a usuarios autorizados, cuya autenticación deberá estar cifrada. Los datos deberán prevalecer en el tiempo con base en respaldos de información.
Prioridad del requerimiento:	Alta.

Tabla B25. RNF02.

Identificación del requerimiento:	RNF02.
Nombre del requerimiento:	Rendimiento del sistema.
Características:	Tiempo de respuesta, volumen de datos y escalabilidad.
Descripción del requerimiento:	El sistema deberá tener un tiempo aceptable de repuesta a consultas, independientemente de la conexión simultánea de usuarios. También tendrá la capacidad de almacenar todos los datos que se generen a través del tiempo sin experimentar dificultades.
Prioridad del requerimiento:	Alta.

Tabla B26. RNF03.

Identificación del requerimiento:	RNF03.
Nombre del requerimiento:	Usabilidad de la interfaz.
Características:	Facilidad de uso.
Descripción del requerimiento:	La interfaz de usuario de la aplicación deberá ser intuitiva; los usuarios deberán poder llevar a cabo todas las funciones de forma sencilla.
Prioridad del requerimiento:	Alta.

Tabla B27. RNF04.

Identificación del requerimiento:	RNF04.
Nombre del requerimiento:	Disponibilidad del sistema.
Características:	Tiempo de actividad y tolerancia a fallos.
Descripción del requerimiento:	El sistema debe estar disponible todo el tiempo, permitiendo que se desarrollen las actividades internas con normalidad. Se deben capturar los errores para que el sistema continúe en funcionamiento.
Prioridad del requerimiento:	Alta.

Tabla B28. RNF05.

Identificación del requerimiento:	RNF05.
Nombre del requerimiento:	Mantenibilidad del sistema.
Características:	Modularidad y aplicación de estándares.
Descripción del requerimiento:	La aplicación deberá estar constituidas por módulos independientes. Se aplicarán buenas prácticas de desarrollo y estándares que aseguren un aplicativo de calidad.
Prioridad del requerimiento:	Alta.

3.3. Requisitos comunes de las interfaces

3.3.1. Interfaces de usuario

- La interfaz de usuario se presentará como una aplicación móvil accesible desde un dispositivo con el sistema operativo Android.
- Debe estar diseñada con elementos visuales como menús desplegables, campos de texto, botones, barras de búsqueda, tarjetas de datos, entre otros.
- La interfaz debe ser intuitiva y de fácil uso para el personal de la clínica veterinaria. La estructura debe asegurar una navegación fluida entre secciones.
- Debe ser adaptable al tamaño de pantalla de distintos dispositivos móviles y portátiles para mantener el diseño global en celulares, tabletas y notebooks con la capacidad de ejecutar aplicaciones de Android.

3.3.2. Interfaces de hardware

Requisitos de equipo de cómputo

Se requerirá disponer de dispositivos móviles con las siguientes especificaciones mínimas:

- Android 7.0 (Nougat).
- Memoria RAM de 2 GB.
- 100 MB libres de almacenamiento.
- Resolución mínima de 720 x 1080 px.
- Conexión a Wi-Fi.
- Google Play Services.

3.3.3. Interfaces de software

Requisitos del sistema operativo

- El sistema operativo requerido para ejecutar la aplicación es Android.

3.3.4. Interfaces de comunicación

- El sistema debe integrarse con la base de datos de SQL Server almacenada en la nube, pudiendo realizar operaciones de recuperación y actualización de los datos, asegurando la funcionalidad integral de la aplicación.
- Se implementarán medidas de seguridad robustas para proteger la integridad y confidencialidad de los datos durante todas las operaciones de interacción con la base de datos.

3.4. Requisitos de Rendimiento

- El sistema debe ser capaz de manejar simultáneamente un mínimo de 10 usuarios activos sin experimentar una degradación significativa en el tiempo de respuesta.
- El tiempo de carga de la página principal del sistema no debe exceder los 5 segundos en condiciones normales de tráfico.
- El sistema debe ser capaz de procesar al menos 200 transacciones por hora durante las horas pico de uso.
- El tiempo de procesamiento para la generación de informes no debe exceder los 5 segundos, independientemente del tamaño de los datos involucrados.
- El sistema debe garantizar una disponibilidad del 99.9%, lo que permite un tiempo de inactividad máximo de 9 horas, aproximadamente, por año para mantenimiento y actualizaciones programadas.
- El sistema debe ser escalable, permitiendo la adición de recursos de hardware según sea necesario para manejar un aumento en la carga de trabajo sin degradación del rendimiento.
- El sistema debe cumplir con los estándares de seguridad y privacidad establecidos, garantizando que el rendimiento no se vea afectado por medidas de seguridad adicionales.
- El sistema debe tener una capacidad de almacenamiento suficiente para manejar un volumen estimado de datos durante al menos 5 años sin comprometer el rendimiento.

3.5. Restricciones de Diseño

- El sistema debe ser desarrollado utilizando Android como plataforma principal, asegurando la compatibilidad y la integración efectiva con los dispositivos móviles existentes en la clínica.

- El sistema debe ser compatible con Android bajo los requerimientos mínimos que se señalaron anteriormente.
- El sistema debe ser accesible y adaptarse automáticamente a diferentes dispositivos y tamaños de pantalla, garantizando una experiencia óptima para todos los usuarios.

3.6. Atributos del Sistema

- **Flexibilidad:** El sistema debe ser adaptable y flexible frente a los cambios en los requerimientos y necesidades del usuario.
- **Seguridad:** El sistema debe garantizar la protección de los datos confidenciales y sensibles almacenados en la base de datos mediante mecanismos de autenticación, autorización y cifrado.
- **Rendimiento:** El sistema debe ser eficiente y rápido, asegurando tiempos de respuesta ágiles y procesamiento de datos óptimo para ofrecer una experiencia fluida a los usuarios.
- **Disponibilidad:** El sistema debe mantenerse accesible y operativo en todo momento, minimizando los tiempos de inactividad planificados para mantenimiento y actualizaciones.
- **Mantenibilidad:** El sistema debe ser fácil de mantener y actualizar, con un código limpio y bien documentado que facilite la detección y corrección de errores, así como la incorporación de nuevas funcionalidades.
- **Escalabilidad:** El sistema debe ser capaz de manejar un aumento en el volumen de datos y usuarios sin comprometer su rendimiento o funcionalidades actuales.
- **Usabilidad:** El sistema debe ser intuitivo y fácil de utilizar, ofreciendo una interfaz amigable que permita una navegación fluida y la realización efectiva de tareas.

Anexo C. Tareas de la historia de usuario N°1.

Las tareas que conforman la historia de usuario N°1, referente a la implementación de la base de datos, se encuentran desde la Tabla C1 hasta la Tabla C5.

Tabla C1. Tarea N°1: Definir entidades y atributos.

Tarea	
Número: 1.	Número de historia: 1.
Nombre: Definir entidades y atributos.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 01/11/2024.	Fecha de fin: 01/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Establecer cada entidad como una tabla de la base de datos y los atributos como las características que las conforman.	

Tabla C2. Tarea N°2: Establecer relaciones entre entidades.

Tarea	
Número: 2.	Número de historia: 1.
Nombre: Establecer relaciones entre entidades.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 01/11/2024.	Fecha de fin: 01/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Identificar las claves primarias y foráneas en las tablas conformadas para establecer las relaciones adecuadas entre ellas.	

Tabla C3. Tarea N°3: Generar un backup de la base de datos.

Tarea	
Número: 3.	Número de historia: 1.
Nombre: Generar un backup de la base de datos.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 01/11/2024.	Fecha de fin: 01/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear un backup completo de la base de datos (tablas, atributos, claves y relaciones).	

Tabla C4. Tarea N°4: Restauración con el backup de la base de datos.

Tarea	
Número: 4.	Número de historia: 1.
Nombre: Restauración con el backup de la base de datos.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 01/11/2024.	Fecha de fin: 01/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Restaurar la base de datos en el servidor en línea Somee y verificar la creación correcta de todas las estructuras definidas en la base de datos.	

Tabla C5. Tarea N°5: Realizar pruebas de CRUD.

Tarea	
Número: 5.	Número de historia: 1.
Nombre: Realizar pruebas de CRUD.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 04/11/2024.	Fecha de fin: 04/11/2024.
Programador responsable: Christian Jaramillo,	
Descripción: Generar consultas de selección, inserción, eliminación y actualización en todas las tablas de la base de datos para verificar los datos obtenidos.	

Anexo D. Tareas de la historia de usuario N°2.

Las tareas que conforman la historia de usuario N°2, referente a la implementación de la API, se encuentran desde la Tabla D1 hasta la Tabla D3.

Tabla D1. Tarea N°6: Implementar la API.

Tarea	
Número: 6.	Número de historia: 2.
Nombre: Implementar la API.	
Tipo: Desarrollo.	Puntos estimados: 7.
Fecha de inicio: 04/11/2024.	Fecha de fin: 06/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Implementar una API en .NET para la base de datos establecida. Se deberá evidenciar la estructura de carpetas con los archivos correspondientes a controladores, datos, DTO'S, mapeados, modelos, repositorios y servicios.	

Tabla D2. Tarea N°7: Probar el funcionamiento de la API.

Tarea	
Número: 7.	Número de historia: 2.
Nombre: Probar el funcionamiento de la API.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 06/11/2024.	Fecha de fin: 06/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Probar la API a nivel local para comprobar su funcionamiento al ejecutar cada servicio y verificar la respuesta.	

Tabla D3. Tarea N°8: Publicar la API y la base de datos.

Tarea	
Número: 8.	Número de historia: 2.
Nombre: Publicar la API.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 06/11/2024.	Fecha de fin: 06/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Publicar la API de .NET y la base de datos en el servicio en línea Somee para garantizar su disponibilidad durante el desarrollo del sistema.	

Anexo E. Tareas de la historia de usuario N°3.

Las tareas que conforman la historia de usuario N°3, referente a la gestión de datos de la clínica, se encuentran desde la Tabla E1 hasta la Tabla E5.

Tabla E1. Tarea N°9: Definir los datos esenciales de la clínica veterinaria.

Tarea	
Número: 9.	Número de historia: 3.
Nombre: Definir los datos esenciales de la clínica veterinaria.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 07/11/2024.	Fecha de fin: 07/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Identificar los datos importantes de la clínica veterinaria en referencia a su identificación y contacto.	

Tabla E2. Tarea N°10: Diseñar la sección de los datos de la clínica.

Tarea	
Número: 10.	Número de historia: 3.
Nombre: Diseñar la sección de los datos de la clínica.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 07/11/2024.	Fecha de fin: 07/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la sección de datos de la clínica usando una herramienta para generar mockups. Se debe incluir los campos adecuados para cada uno de los datos establecidos previamente.	

Tabla E3. Tarea N°11: Implementar la sección de los datos de la clínica.

Tarea	
Número: 11.	Número de historia: 3.
Nombre: Implementar la sección de los datos de la clínica.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 07/11/2024.	Fecha de fin: 07/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear la sección destinada a los datos de la clínica en Android Studio de acorde al diseño establecido.	

Tabla E4. Tarea N°12: Agregar validaciones en los campos de datos de la clínica.

Tarea	
Número: 12.	Número de historia: 3.
Nombre: Agregar validaciones en los campos de datos de la clínica.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 07/11/2024.	Fecha de fin: 07/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Implementar validaciones de caracteres y extensión permitidos de acorde a cada dato de la sección.	

Tabla E5. Tarea N°13: Implementar la función para modificar los datos de la clínica.

Tarea	
Número: 13.	Número de historia: 3.
Nombre: Implementar la función para modificar los datos de la clínica.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 08/11/2024.	Fecha de fin: 08/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Agregar la función para actualizar los datos que han sido modificados a través de un botón. Se deberá agregar un mensaje de confirmación de los cambios.	

Anexo F. Tareas de la historia de usuario N°4.

Las tareas que conforman la historia de usuario N°4 se encuentran desde la Tabla F1 hasta la Tabla F6.

Tabla F1. Tarea N°14: Diseñar la sección de gestión de empleados.

Tarea	
Número: 14.	Número de historia: 4.
Nombre: Diseñar la sección de gestión de empleados.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 08/11/2024.	Fecha de fin: 08/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la sección de gestión de empleados usando una herramienta para generar mockups. Se debe incluir fichas con los datos esenciales de cada empleado y un botón para agregar uno nuevo.	

Tabla F2. Tarea N°15: Implementar la sección de gestión de empleados.

Tarea	
Número: 15.	Número de historia: 4.
Nombre: Implementar la sección de gestión de empleados.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 08/11/2024.	Fecha de fin: 08/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear la sección destinada a la gestión de empleados en Android Studio de acorde al diseño establecido.	

Tabla F3. Tarea N°16: Diseñar el formulario para los empleados.

Tarea	
Número: 16.	Número de historia: 4.
Nombre: Diseñar el formulario para los empleados.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 11/11/2024.	Fecha de fin: 11/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la interfaz del formulario para los empleados usando una herramienta para generar mockups. Se debe incluir un campo de texto para cada dato requerido (nombre, apellido, cédula, teléfono, correo y dirección domiciliaria).	

Tabla F4. Tarea N°17: Implementar el formulario para empleados.

Tarea	
Número: 17.	Número de historia: 4.
Nombre: Implementar el formulario para empleados.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 11/11/2024.	Fecha de fin: 11/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el formulario para los empleados en Android Studio de acorde al diseño establecido.	

Tabla F5. Tarea N°18: Agregar validaciones en el formulario de empleados.

Tarea	
Número: 18.	Número de historia: 4.
Nombre: Agregar validaciones en el formulario de empleados.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 11/11/2024.	Fecha de fin: 11/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Añadir la validación en cada campo de texto para permitir únicamente información válida de los empleados: solo letras (nombre, apellido, correo, dirección), solo números (cédula, teléfono), así como una extensión máxima o fija de caracteres (cédula [10], correo [320], etc.).	

Tabla F6. Tarea N°19: Implementar las funciones para agregar, modificar y eliminar empleados.

Tarea	
Número: 19.	Número de historia: 4.
Nombre: Implementar las funciones para agregar, modificar y eliminar empleados.	
Tipo: Desarrollo.	Puntos estimados: 3.
Fecha de inicio: 12/11/2024	Fecha de fin: 12/11/2024
Programador responsable: Christian Jaramillo.	
Descripción: Agregar una función para registrar los datos del formulario del empleado en la base de datos a través de un botón. El botón de modificar abrirá el formulario, pero con los datos del empleado en los campos, mismos que podrán editarse y guardarse. Finalmente, el botón de eliminar desplegará un mensaje de confirmación para llevar a cabo dicha acción.	

Anexo G. Tareas de la historia de usuario N°5.

Las tareas que conforman la historia de usuario N°5 se encuentran desde la Tabla G1 hasta la Tabla G6.

Tabla G1. Tarea N°20: Definir los tipos de usuario del sistema y sus permisos.

Tarea	
Número: 20.	Número de historia: 5.
Nombre: Definir los tipos de usuario del sistema y sus permisos.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 12/11/2024.	Fecha de fin: 12/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Identificar los roles de usuario que tendrá el sistema con base a los cargos que existen en la clínica veterinaria.	

Tabla G2. Tarea N°21: Diseñar el formulario para los usuarios.

Tarea	
Número: 21.	Número de historia: 5.
Nombre: Diseñar el formulario para los usuarios.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 13/11/2024.	Fecha de fin: 13/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la interfaz del formulario para los usuarios usando una herramienta para generar mockups. Se debe incluir un campo para el nombre de usuario, la contraseña y el tipo de usuario.	

Tabla G3. Tarea N°22: Implementar el formulario para usuarios.

Tarea	
Número: 22.	Número de historia: 5.
Nombre: Implementar el formulario para usuarios.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 13/11/2024.	Fecha de fin: 13/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el formulario para los usuarios en Android Studio de acorde al diseño establecido. Añadir estos campos al formulario de empleados para complementar el registro previo.	

Tabla G4. Tarea N°23: Agregar validaciones en el formulario de usuarios.

Tarea	
Número: 23.	Número de historia: 5.
Nombre: Agregar validaciones en el formulario de usuarios.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 13/11/2024.	Fecha de fin: 13/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Añadir las validaciones correspondientes para el nombre de usuario, contraseña y un campo de selección con únicamente los roles de usuario que permite el sistema.	

Tabla G5. Tarea N°24: Controlar la visualización de cada sección según el tipo de usuario.

Tarea	
Número: 24.	Número de historia: 5.
Nombre: Controlar la visualización de cada sección según el tipo de usuario.	
Tipo: Desarrollo.	Puntos estimados: 4.
Fecha de inicio: 14/11/2024.	Fecha de fin: 14/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Añadir controles en el sistema que muestren u oculten cada sección o pestaña de acceso según cada tipo de usuario que ingrese y los permisos que le fueron asignados.	

Tabla G6. Tarea N°25: Crear un usuario de cada tipo y verificar los permisos asignados.

Tarea	
Número: 25.	Número de historia: 5.
Nombre: Crear un usuario de cada tipo y verificar los permisos asignados.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 15/11/2024	Fecha de fin: 15/11/2024
Programador responsable: Christian Jaramillo.	
Descripción: Crear un usuario de prueba de cada tipo en la base de datos, acceder con cada uno de ellos y verificar que las secciones que pueden visualizar, estén acorde a los permisos predefinidos.	

Anexo H. Tareas de la historia de usuario N°6.

Las tareas que conforman la historia de usuario N°6 se encuentran desde la Tabla H1 hasta la Tabla H4.

Tabla H1. Tarea N°26: Diseñar el formulario de inicio de sesión.

Tarea	
Número: 26.	Número de historia: 6.
Nombre: Diseñar el formulario de inicio de sesión.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 15/11/2024.	Fecha de fin: 15/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la interfaz del formulario de inicio de sesión usando una herramienta para generar mockups. Se debe incluir dos campos de texto para el usuario y la clave, así como un botón para ingresar al inicio de la aplicación.	

Tabla H2. Tarea N°27: Implementar el formulario de inicio de sesión.

Tarea	
Número: 27.	Número de historia: 6.
Nombre: Implementar el formulario de inicio de sesión.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 15/11/2024.	Fecha de fin: 15/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el formulario de inicio de sesión en Android Studio de acorde al diseño establecido.	

Tabla H3. Tarea N°28: Implementar la validación de credenciales.

Tarea	
Número: 28.	Número de historia: 6.
Nombre: Implementar la validación de credenciales.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 18/11/2024.	Fecha de fin: 18/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear un usuario de prueba en la base de datos. Configurar el formulario de inicio de sesión de Android para que se conecte a la base de datos y verifique las credenciales que ingrese el usuario; la aplicación deberá mostrar un mensaje de error en caso de que las credenciales no correspondan a ningún usuario registrado.	

Tabla H4. Tarea N°29: Integrar los roles de usuarios.

Tarea	
Número: 29.	Número de historia: 6.
Nombre: Integrar los roles de usuarios.	
Tipo: Desarrollo.	Puntos estimados: 4.
Fecha de inicio: 18/11/2024.	Fecha de fin: 19/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Configurar la aplicación para que muestre secciones o permita el acceso a funcionalidades según un rol asignado que se encuentra registrado en la base de datos.	

Anexo I. Tareas de la historia de usuario N°7.

Las tareas que conforman la historia de usuario N°7 se encuentran desde la Tabla I1 hasta la Tabla I6.

Tabla I1. Tarea N°30: Diseñar la sección de gestión de propietarios.

Tarea	
Número: 30.	Número de historia: 7.
Nombre: Diseñar la sección de gestión de propietarios.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 19/11/2024.	Fecha de fin: 19/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la sección de gestión de propietarios usando una herramienta para generar mockups. Se debe incluir fichas con los datos esenciales de cada propietario y un botón para agregar uno nuevo.	

Tabla I2. Tarea N°31: Implementar la sección de gestión de propietarios.

Tarea	
Número: 31.	Número de historia: 7.
Nombre: Implementar la sección de gestión de propietarios.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 19/11/2024.	Fecha de fin: 20/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear la sección destinada a la gestión de propietarios en Android Studio de acorde al diseño establecido.	

Tabla I3. Tarea N°32: Diseñar el formulario para los propietarios.

Tarea	
Número: 32.	Número de historia: 7.
Nombre: Diseñar el formulario para los propietarios.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 20/11/2024.	Fecha de fin: 20/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la interfaz del formulario para los propietarios usando una herramienta para generar mockups. Se debe incluir un campo de texto para cada dato requerido (nombre, apellido, cédula, teléfono, correo y dirección domiciliaria).	

Tabla I4. Tarea N°33: Implementar el formulario para propietarios.

Tarea	
Número: 33.	Número de historia: 7.
Nombre: Implementar el formulario para propietarios.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 20/11/2024.	Fecha de fin: 20/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el formulario para los propietarios en Android Studio de acorde al diseño establecido.	

Tabla I5. Tarea N°34: Agregar validaciones en el formulario de propietarios.

Tarea	
Número: 34.	Número de historia: 7.
Nombre: Agregar validaciones en el formulario de propietarios.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 21/11/2024.	Fecha de fin: 21/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Añadir la validación en cada campo de texto para permitir únicamente información válida de los propietarios: solo letras (nombre, apellido, correo, dirección), solo números (cédula, teléfono), así como una extensión máxima o fija de caracteres (cédula [10], correo [320], etc.).	

Tabla I6. Tarea N°35: Implementar las funciones para agregar, modificar y eliminar propietarios.

Tarea	
Número: 35.	Número de historia: 7.
Nombre: Implementar las funciones para agregar, modificar y eliminar propietarios.	
Tipo: Desarrollo.	Puntos estimados: 3.
Fecha de inicio: 21/11/2024	Fecha de fin: 21/11/2024
Programador responsable: Christian Jaramillo.	
Descripción: Agregar una función para registrar los datos del formulario del propietario en la base de datos a través de un botón. El botón de modificar abrirá el formulario, pero con los datos del propietario en los campos, mismos que podrán editarse y guardarse. Finalmente, el botón de eliminar desplegará un mensaje de confirmación para llevar a cabo dicha acción.	

Anexo J. Tareas de la historia de usuario N°8.

Las tareas que conforman la historia de usuario N°8 se encuentran desde la Tabla J1 hasta la Tabla J6.

Tabla J1. Tarea N°36: Diseñar la sección de gestión de pacientes.

Tarea	
Número: 36.	Número de historia: 8.
Nombre: Diseñar la sección de gestión de pacientes.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 22/11/2024.	Fecha de fin: 22/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la sección de gestión de pacientes usando una herramienta para generar mockups. Se debe incluir fichas con los datos esenciales de cada paciente y un botón para agregar uno nuevo.	

Tabla J2. Tarea N°37: Implementar la sección de gestión de pacientes.

Tarea	
Número: 37.	Número de historia: 8.
Nombre: Implementar la sección de gestión de pacientes.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 22/11/2024.	Fecha de fin: 22/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear la sección destinada a la gestión de pacientes en Android Studio de acorde al diseño establecido.	

Tabla J3. Tarea N°38: Diseñar el formulario para los pacientes.

Tarea	
Número: 38.	Número de historia: 8.
Nombre: Diseñar el formulario para los pacientes.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 22/11/2024.	Fecha de fin: 22/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la interfaz del formulario para los pacientes usando una herramienta para generar mockups. Se debe incluir un campo de texto para cada dato requerido (nombre, fecha de nacimiento, especie, raza, sexo).	

Tabla J4. Tarea N°39: Implementar el formulario para pacientes.

Tarea	
Número: 39.	Número de historia: 8.
Nombre: Implementar el formulario para pacientes.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 25/11/2024.	Fecha de fin: 25/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el formulario para los pacientes en Android Studio de acorde al diseño establecido.	

Tabla J5. Tarea N°40: Agregar validaciones en el formulario de pacientes.

Tarea	
Número: 40.	Número de historia: 8.
Nombre: Agregar validaciones en el formulario de pacientes.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 25/11/2024.	Fecha de fin: 25/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Añadir la validación en cada campo de texto para permitir únicamente información válida de los pacientes: solo letras (nombre), selección de un calendario (fecha de nacimiento) y listas desplegables con opciones predefinidas (especie, raza, sexo).	

Tabla J6. Tarea N°41: Implementar las funciones para agregar, modificar y eliminar pacientes.

Tarea	
Número: 41.	Número de historia: 8.
Nombre: Implementar las funciones para agregar, modificar y eliminar pacientes.	
Tipo: Desarrollo.	Puntos estimados: 3.
Fecha de inicio: 25/11/2024	Fecha de fin: 26/11/2024
Programador responsable: Christian Jaramillo.	
Descripción: Agregar una función para registrar los datos del formulario del paciente en la base de datos a través de un botón. El botón de modificar abrirá el formulario, pero con los datos del paciente en los campos, mismos que podrán editarse y guardarse. Finalmente, el botón de eliminar desplegará un mensaje de confirmación para llevar a cabo dicha acción.	

Anexo K. Tareas de la historia de usuario N°9.

Las tareas que conforman la historia de usuario N°9 se encuentran desde la Tabla K1 hasta la Tabla K6.

Tabla K1. Tarea N°42: Diseñar la sección de gestión de vacunas.

Tarea	
Número: 42.	Número de historia: 9.
Nombre: Diseñar la sección de gestión de vacunas.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 26/11/2024.	Fecha de fin: 26/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la sección de gestión de vacunas usando una herramienta para generar mockups. Se debe incluir fichas con los datos esenciales de cada vacuna y un botón para agregar una nueva.	

Tabla K2. Tarea N°43: Implementar la sección de gestión de vacunas.

Tarea	
Número: 43.	Número de historia: 9.
Nombre: Implementar la sección de gestión de vacunas.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 26/11/2024.	Fecha de fin: 27/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear la sección destinada a la gestión de vacunas en Android Studio de acorde al diseño establecido.	

Tabla K3. Tarea N°44: Diseñar el formulario para las vacunas.

Tarea	
Número: 44.	Número de historia: 9.
Nombre: Diseñar el formulario para las vacunas.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 27/11/2024.	Fecha de fin: 27/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la interfaz del formulario para las vacunas usando una herramienta para generar mockups. Se debe incluir un campo de texto para cada dato requerido (tipo de vacuna, marca de la vacuna, fecha de aplicación y observaciones).	

Tabla K4. Tarea N°45: Implementar el formulario para vacunas.

Tarea	
Número: 45.	Número de historia: 9.
Nombre: Implementar el formulario para vacunas.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 27/11/2024.	Fecha de fin: 27/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el formulario para las vacunas en Android Studio de acorde al diseño establecido.	

Tabla K5. Tarea N°46: Agregar validaciones en el formulario de vacunas.

Tarea	
Número: 46.	Número de historia: 9.
Nombre: Agregar validaciones en el formulario de vacunas.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 28/11/2024.	Fecha de fin: 28/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Añadir la validación en cada campo de texto para permitir únicamente información válida de las vacunas: selección de un calendario (fecha de aplicación) y listas desplegables con opciones predefinidas (tipo y marca de vacuna).	

Tabla K6. Tarea N°47: Implementar las funciones para agregar, modificar y eliminar vacunas.

Tarea	
Número: 47.	Número de historia: 9.
Nombre: Implementar las funciones para agregar, modificar y eliminar vacunas.	
Tipo: Desarrollo.	Puntos estimados: 3.
Fecha de inicio: 28/11/2024	Fecha de fin: 28/11/2024
Programador responsable: Christian Jaramillo.	
Descripción: Agregar una función para registrar los datos del formulario de la vacuna en la base de datos a través de un botón. El botón de modificar abrirá el formulario, pero con los datos de la vacuna en los campos, mismos que podrán editarse y guardarse. Finalmente, el botón de eliminar desplegará un mensaje de confirmación para llevar a cabo dicha acción.	

Anexo L. Tareas de la historia de usuario N°10.

Las tareas que conforman la historia de usuario N°10 se encuentran desde la Tabla L1 hasta la Tabla L6.

Tabla L1. Tarea N°48: Diseñar la sección de gestión de desparasitaciones.

Tarea	
Número: 48.	Número de historia: 10.
Nombre: Diseñar la sección de gestión de desparasitaciones.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 29/11/2024.	Fecha de fin: 29/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la sección de gestión de desparasitaciones usando una herramienta para generar mockups. Se debe incluir fichas con los datos esenciales de cada desparasitación y un botón para agregar una nueva.	

Tabla L2. Tarea N°49: Implementar la sección de gestión de desparasitaciones.

Tarea	
Número: 49.	Número de historia: 10.
Nombre: Implementar la sección de gestión de desparasitaciones.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 29/11/2024.	Fecha de fin: 29/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear la sección destinada a la gestión de desparasitaciones en Android Studio de acorde al diseño establecido.	

Tabla L3. Tarea N°50: Diseñar el formulario para las desparasitaciones.

Tarea	
Número: 50.	Número de historia: 10.
Nombre: Diseñar el formulario para las desparasitaciones.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 29/11/2024.	Fecha de fin: 29/11/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la interfaz del formulario para las desparasitaciones usando una herramienta para generar mockups. Se debe incluir un campo de texto para cada dato requerido (tipo de desparasitación, marca de la desparasitación, fecha de aplicación y observaciones).	

Tabla L4. Tarea N°51: Implementar el formulario para desparasitaciones.

Tarea	
Número: 51.	Número de historia: 10.
Nombre: Implementar el formulario para desparasitaciones.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 02/12/2024.	Fecha de fin: 02/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el formulario para las desparasitaciones en Android Studio de acorde al diseño establecido.	

Tabla L5. Tarea N°52: Agregar validaciones en el formulario de desparasitaciones.

Tarea	
Número: 52.	Número de historia: 10.
Nombre: Agregar validaciones en el formulario de desparasitaciones.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 02/12/2024.	Fecha de fin: 02/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Añadir la validación en cada campo de texto para permitir únicamente información válida de las desparasitaciones: selección de un calendario (fecha de aplicación) y listas desplegables con opciones predefinidas (tipo y marca de vacuna).	

Tabla L6. Tarea N°53: Implementar las funciones para agregar, modificar y eliminar desparasitaciones.

Tarea	
Número: 53.	Número de historia: 10.
Nombre: Implementar las funciones para agregar, modificar y eliminar desparasitaciones.	
Tipo: Desarrollo.	Puntos estimados: 3.
Fecha de inicio: 02/12/2024	Fecha de fin: 03/12/2024
Programador responsable: Christian Jaramillo.	
Descripción: Agregar una función para registrar los datos del formulario de la desparasitación en la base de datos a través de un botón. El botón de modificar abrirá el formulario, pero con los datos de la desparasitación en los campos, mismos que podrán editarse y guardarse. Finalmente, el botón de eliminar desplegará un mensaje de confirmación para llevar a cabo dicha acción.	

Anexo M. Tareas de la historia de usuario N°11.

Las tareas que conforman la historia de usuario N°11 se encuentran desde la Tabla M1 hasta la Tabla M6.

Tabla M1. Tarea N°54: Diseñar la sección de gestión de citas médicas.

Tarea	
Número: 54.	Número de historia: 11.
Nombre: Diseñar la sección de gestión de citas médicas.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 03/12/2024.	Fecha de fin: 03/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la sección de gestión de citas médicas usando una herramienta para generar mockups. Se debe incluir fichas con los datos esenciales	

de cada cita médica y un botón para agregar una nueva.

Tabla M2. Tarea N°55: Implementar la sección de gestión de citas médicas.

Tarea	
Número: 55.	Número de historia: 11.
Nombre: Implementar la sección de gestión de citas médicas.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 03/12/2024.	Fecha de fin: 04/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear la sección destinada a la gestión de citas médicas en Android Studio de acorde al diseño establecido.	

Tabla M3. Tarea N°56: Diseñar el formulario para las citas médicas.

Tarea	
Número: 56.	Número de historia: 11.
Nombre: Diseñar el formulario para las citas médicas.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 04/12/2024.	Fecha de fin: 04/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la interfaz del formulario para las citas médicas usando una herramienta para generar mockups. Se debe incluir un campo de texto para cada dato requerido (fecha, frecuencia respiratoria, frecuencia cardiaca, temperatura corporal, presión arterial, llenado capilar y observaciones).	

Tabla M4. Tarea N°57: Implementar el formulario para citas médicas.

Tarea	
Número: 57.	Número de historia: 11.
Nombre: Implementar el formulario para citas médicas.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 04/12/2024.	Fecha de fin: 04/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el formulario para las citas médicas en Android Studio de acorde al diseño establecido.	

Tabla M5. Tarea N°58: Agregar validaciones en el formulario de citas médicas.

Tarea	
Número: 58.	Número de historia: 11.
Nombre: Agregar validaciones en el formulario de citas médicas.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 05/12/2024.	Fecha de fin: 05/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Añadir la validación en cada campo de texto para permitir únicamente información válida de las desparasitaciones: selección de un calendario (fecha), números enteros (frecuencia respiratoria, frecuencia cardiaca, presión arterial, llenado capilar) y números decimales (temperatura corporal).	

Tabla M6. Tarea N°59: Implementar las funciones para agregar, modificar y eliminar citas médicas.

Tarea	
Número: 59.	Número de historia: 11.
Nombre: Implementar las funciones para agregar, modificar y eliminar citas médicas.	
Tipo: Desarrollo.	Puntos estimados: 3.
Fecha de inicio: 05/12/2024	Fecha de fin: 05/12/2024
Programador responsable: Christian Jaramillo.	
Descripción: Agregar una función para registrar los datos del formulario de la cita médica en la base de datos a través de un botón. El botón de modificar abrirá el formulario, pero con los datos de la cita médica en los campos, mismos que podrán editarse y guardarse. Finalmente, el botón de eliminar desplegará un mensaje de confirmación para llevar a cabo dicha acción.	

Anexo N. Tareas de la historia de usuario N°12.

Las tareas que conforman la historia de usuario N°12 se encuentran desde la Tabla N1 hasta la Tabla N3.

Tabla N1. Tarea N°60: Diseñar la sección de citas médicas programadas.

Tarea	
Número: 60.	Número de historia: 12.
Nombre: Diseñar la sección de citas médicas programadas.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 06/12/2024.	Fecha de fin: 06/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la sección de citas médicas programadas usando una herramienta para generar mockups. Se debe incluir fichas con la información básica de la cita médica (paciente, fecha y hora).	

Tabla N2. Tarea N°61: Implementar la sección de citas médicas programadas.

Tarea	
Número: 61.	Número de historia: 12.
Nombre: Implementar la sección de citas médicas programadas.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 06/12/2024.	Fecha de fin: 06/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear la sección destinada a la sección de citas médicas programadas en Android Studio de acorde al diseño establecido.	

Tabla N3. Tarea N°62: Implementar una función para contactar al propietario vía Whatsapp.

Tarea	
Número: 62.	Número de historia: 12.
Nombre: Implementar una función para contactar al propietario vía Whatsapp.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 06/12/2024.	Fecha de fin: 09/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Añadir una función en Android Studio que permita contactar al propietario vía Whatsapp con un mensaje predeterminado.	

Anexo O. Tareas de la historia de usuario N°13.

Las tareas que conforman la historia de usuario N°13 se encuentran desde la Tabla O1 hasta la Tabla O6.

Tabla O1. Tarea N°63: Diseñar la sección de gestión de registros médicos.

Tarea	
Número: 63.	Número de historia: 13.
Nombre: Diseñar la sección de gestión de registros médicos.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 09/12/2024.	Fecha de fin: 09/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la sección de gestión de registros médicos usando una herramienta para generar mockups. Se debe incluir fichas con los datos esenciales de cada registro médico y un botón para agregar uno nuevo.	

Tabla O2. Tarea N°64: Implementar la sección de gestión de registros médicos.

Tarea	
Número: 64.	Número de historia: 13.
Nombre: Implementar la sección de gestión de registros médicos.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 09/12/2024.	Fecha de fin: 09/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear la sección destinada a la gestión de registros médicos en Android Studio de acorde al diseño establecido.	

Tabla O3. Tarea N°65: Diseñar el formulario para los registros médicos.

Tarea	
Número: 65.	Número de historia: 13.
Nombre: Diseñar el formulario para los registros médicos.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 10/12/2024.	Fecha de fin: 10/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la interfaz del formulario para los registros médicos usando una herramienta para generar mockups. Se debe incluir un campo de texto para cada dato requerido (fecha, empleado a cargo, peso, frecuencia respiratoria, frecuencia cardiaca, temperatura corporal, presión arterial, llenado capilar y observaciones).	

Tabla O4. Tarea N°66: Implementar el formulario para registros médicos.

Tarea	
Número: 66.	Número de historia: 13.
Nombre: Implementar el formulario para registros médicos.	
Tipo: Desarrollo.	Puntos estimados: 2.
Fecha de inicio: 10/12/2024.	Fecha de fin: 10/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el formulario para los registros médicos en Android Studio de acorde al diseño establecido.	

Tabla O5. Tarea N°67: Agregar validaciones en el formulario de registros médicos.

Tarea	
Número: 67.	Número de historia: 13.
Nombre: Agregar validaciones en el formulario de registros médicos.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 10/12/2024.	Fecha de fin: 10/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Añadir la validación en cada campo de texto para permitir únicamente información válida de los registros médicos: selección de un calendario (fecha), lista desplegable con opciones predefinidas (empleado a cargo), números enteros (frecuencia respiratoria, frecuencia cardiaca, presión arterial, llenado capilar) y números decimales (peso, temperatura corporal).	

Tabla O6. Tarea N°68: Implementar las funciones para agregar, modificar y eliminar registros médicos.

Tarea	
Número: 68.	Número de historia: 13.
Nombre: Implementar las funciones para agregar, modificar y eliminar registros médicos.	
Tipo: Desarrollo.	Puntos estimados: 3.
Fecha de inicio: 11/12/2024	Fecha de fin: 11/12/2024
Programador responsable: Christian Jaramillo.	
Descripción: Agregar una función para registrar los datos del formulario del registro en la base de datos a través de un botón. El botón de modificar abrirá el formulario, pero con los datos del registro médico en los campos, mismos que podrán editarse y guardarse. Finalmente, el botón de eliminar desplegará un mensaje de confirmación para llevar a cabo dicha acción.	

Anexo P. Tareas de la historia de usuario N°14.

Las tareas que conforman la historia de usuario N°14 se encuentran desde la Tabla P1 hasta la Tabla P2.

Tabla P1. Tarea N°69: Diseñar la pantalla de visualización del reporte del signo clínico.

Tarea	
Número: 69.	Número de historia: 14.
Nombre: Diseñar la pantalla de visualización del reporte del signo clínico.	
Tipo: Desarrollo.	Puntos estimados: 1.
Fecha de inicio: 11/12/2024.	Fecha de fin: 11/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear el diseño de la pantalla en donde se podrá visualizar el reporte gráfico (gráfico de líneas). Se debe incluir las etiquetas, valores y unidades necesarias para identificar correctamente los datos que se estén mostrando.	

Tabla P2. Tarea N°70: Implementar la pantalla de visualización del reporte del signo clínico.

Tarea	
Número: 70.	Número de historia: 14.
Nombre: Implementar la pantalla de visualización del reporte del signo clínico.	
Tipo: Desarrollo.	Puntos estimados: 4.
Fecha de inicio: 12/12/2024.	Fecha de fin: 12/12/2024.
Programador responsable: Christian Jaramillo.	
Descripción: Crear la sección destinada a la visualización del reporte gráfico del signo clínico seleccionado en Android Studio de acorde al diseño establecido.	