



**UNIVERSIDAD TÉCNICA DE AMBATO**

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

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E  
INFORMÁTICOS**

**Tema:**

---

**APLICACIÓN WEB PROGRESIVA (PWA) CON IOT PARA EL MONITOREO  
DE MASCOTAS EN TIEMPO REAL EN LA CIUDAD DE QUITO**

---

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

**ÁREA:** Software

**LÍNEA DE INVESTIGACIÓN:** Desarrollo de Software

**AUTOR:** Juan Andrés Caisapanta Analuisa

**TUTOR:** Ing. Clay Fernando Aldás Flores, Mg.

**Ambato – Ecuador**

**febrero - 2023**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Titulación con el tema: **APLICACIÓN WEB PROGRESIVA (PWA) CON IOT PARA EL MONITOREO DE MASCOTAS EN TIEMPO REAL EN LA CIUDAD DE QUITO**, desarrollado bajo la modalidad Proyecto de Investigación por el señor Juan Andrés Caisapanta Analuisa, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, 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 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, febrero 2023.

-----  
Ing. Clay Fernando Aldás Flores, Mg.

**TUTOR**

## AUTORÍA

El presente Proyecto de Investigación titulado: APLICACIÓN WEB PROGRESIVA (PWA) CON IOT PARA EL MONITOREO DE MASCOTAS EN TIEMPO REAL EN LA CIUDAD DE QUITO es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2023.



Juan Andrés Caisapanta Analuisa  
C.C. 1726844689  
AUTOR

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, febrero 2023.



Juan Andrés Caisapanta Analuisa

C.C. 1726844689

AUTOR



## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Juan Andrés Caisapanta Analuisa, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **APLICACIÓN WEB PROGRESIVA (PWA) CON IOT PARA EL MONITOREO DE MASCOTAS EN TIEMPO REAL EN LA CIUDAD DE QUITO**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, febrero 2023.

-----  
Ing. Pilar Urrutia, Mg.  
PRESIDENTA DEL TRIBUNAL

-----  
Ing. Hernán Naranjo, Mg.  
PROFESOR CALIFICADOR

-----  
Ing. Edison Álvarez, Mg.  
PROFESOR CALIFICADOR

## DEDICATORIA

Este proyecto de investigación se lo dedico a toda mi familia y amigos que me han apoyado en el transcurso de mi vida como estudiante universitario, motivándome para cumplir un objetivo importante en mi vida.

## AGRADECIMIENTO

Agradezco primeramente a Dios por darme la vida y la sabiduría para poder culminar con éxito mi etapa de estudiante universitario.

A mi abuelito Segundo y mi madre Lourdes que en paz descansen, donde sea que se encuentren sé que velan por mi bienestar.

A mi tía Carmen por ser como una madre para mi guiándome y apoyándome en cada etapa de mi vida.

A mi padre Juan por su apoyo en los momentos que más lo necesite.

A toda mi familia por estar siempre pendientes de mí y ayudarme a cumplir una meta más en vida.

A mi tutor el Ing. Clay Aldas el cual ha sido un excelente profesor en las aulas de clase y me ha dedicado su tiempo, ayuda y guía en el desarrollo del proyecto de investigación.

## ÍNDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR .....	II
AUTORÍA.....	III
DERECHOS DE AUTOR.....	IV
APROBACIÓN TRIBUNAL DE GRADO.....	V
DEDICATORIA .....	VI
AGRADECIMIENTO .....	VII
ÍNDICE GENERAL DE CONTENIDOS .....	VIII
ÍNDICE DE FIGURAS .....	X
ÍNDICE DE TABLAS.....	XVIII
RESUMEN EJECUTIVO .....	XXIII
ABSTRACT.....	XXIV
INTRODUCCIÓN .....	XXV
CAPITULO I MARCO TEÓRICO .....	1
1.1 Tema de Investigación .....	1
1.2 Antecedentes Investigaciones .....	1
1.2.1 Contextualización del problema.....	1
1.2.2 Fundamentación teórica .....	2
1.2.2.1 Monitoreo.....	2
1.2.2.2 Sistemas de monitoreo .....	3
1.2.2.3 Sistemas de monitoreo de mascotas.....	3
1.2.2.4 Internet de las cosas .....	3
1.2.2.5 Redes de área amplia de baja potencia .....	4
1.2.2.6 Arduino .....	6
1.2.2.7 Sistema web .....	7
1.2.2.8 Aplicación web .....	7
1.2.2.9 Aplicación Web Progresiva .....	7
1.2.2.10 Framework de desarrollo web.....	8
1.2.2.11 Tecnologías front-end .....	9
1.2.2.12 Tecnologías back-end .....	11
1.2.2.13 Servicios web tipo REST .....	11
1.2.2.14 Jason Web Token.....	12
1.2.2.15 Spring Boot.....	12
1.2.2.16 Sistemas de gestión de base de datos.....	13
1.2.2.17 Plataformas de almacenamiento en la nube.....	14
1.2.2.18 Metodologías de desarrollo de software .....	16
1.2.2.19 Metodologías ágiles de desarrollo de software.....	16
1.3 Objetivos .....	19
1.3.1 Objetivo general .....	19
1.3.2 Objetivos específicos .....	19
CAPÍTULO II METODOLOGÍA .....	20
2.1 Materiales.....	20

2.2	Métodos.....	20
2.2.1	Modalidad de investigación .....	20
2.2.2	Población y muestra .....	20
2.2.3	Recolección de información.....	21
2.2.4	Procesamiento y análisis de datos .....	26
<b>CAPÍTULO III RESULTADOS Y DISCUSIÓN.....</b>		<b>28</b>
3.1	Análisis y discusión de resultados.....	28
3.1.1	Análisis de la tecnología más accesible para el rastreo de mascotas en la ciudad de Quito .....	28
3.1.2	Determinación de modulo Arduino Sigfox .....	29
3.1.2.1	Ufox .....	30
3.1.3	Estudio de características de una Aplicación Web Progresiva .....	33
3.1.3.1	Estructura de una PWA.....	33
3.1.3.2	Ventajas, desventajas y Características principales de una PWA	33
3.1.3.3	Comparación entre enfoques de desarrollo Nativo, Hibrido y PWA .....	35
3.1.3.4	Tecnologías para el desarrollo de aplicaciones web progresivas .	36
3.1.4	Determinación de la tecnología frond-end.....	38
3.1.5	Determinación de la tecnología back-end .....	39
3.1.6	Determinación de la metodología de desarrollo.....	41
3.2	Desarrollo de la propuesta.....	42
3.2.1	Funcionamiento de la PWA .....	42
3.2.2	Desarrollo de prototipo del dispositivo de monitoreo usando tecnología Sigfox .....	43
3.2.2.1	Esquema de conexiones .....	44
3.2.2.2	Código usado en el prototipo .....	44
3.2.2.3	Diseño 3D usado para la caja del prototipo del dispositivo.....	46
3.2.2.4	Configuración del servidor Sigfox.....	47
3.2.3	Aplicación de la metodología ágil XP al desarrollo de la PWA .....	54
3.2.3.1	Análisis y requerimientos de la PWA.....	54
3.2.3.2	Definición de roles.....	54
3.2.3.3	Fase de planificación .....	55
3.2.3.4	Iteraciones .....	83
3.2.3.5	Fase de diseño .....	98
3.2.3.6	Fase de Codificación.....	138
3.2.3.7	Fase de pruebas .....	150
3.2.3.8	Fase de producción .....	165
<b>CAPÍTULO IV CONCLUSIONES Y RECOMENDACIONES .....</b>		<b>176</b>
4.1	Conclusiones .....	176
4.2	Recomendaciones.....	176
<b>REFERENCIA BIBLIOGRÁFICA .....</b>		<b>178</b>
<b>ANEXOS .....</b>		<b>183</b>

## ÍNDICE DE FIGURAS

Figura 1.1 Esquema red LPWAN .....	4
Figura 1.2 Esquema de funcionamiento de Sigfox .....	5
Figura 1.3 Arquitectura de una Aplicación Web Progresiva .....	8
Figura 2.1 Gráfico Pregunta 1 Encuesta Monitoreo de Mascotas Ciudad de Quito ..	21
Figura 2.2 Gráfico Pregunta 2 Encuesta Monitoreo de Mascotas Ciudad de Quito ..	22
Figura 2.3 Gráfico Pregunta 3 Encuesta Monitoreo de Mascotas Ciudad de Quito ..	22
Figura 2.4 Gráfico Pregunta 4 Encuesta Monitoreo de Mascotas Ciudad de Quito ..	23
Figura 2.5 Gráfico Pregunta 5 Encuesta Monitoreo de Mascotas Ciudad de Quito ..	23
Figura 2.6 Gráfico Pregunta 6 Encuesta Monitoreo de Mascotas Ciudad de Quito ..	24
Figura 2.7 Gráfico Pregunta 7 Encuesta Monitoreo de Mascotas Ciudad de Quito ..	25
Figura 2.8 Gráfico Pregunta 8 Encuesta Monitoreo de Mascotas Ciudad de Quito ..	25
Figura 2.9 Gráfico Pregunta 9 Encuesta Monitoreo de Mascotas Ciudad de Quito ..	26
Figura 3.1 Grafico de las zonas Sigfox .....	30
Figura 3.2 Descripción de pines Ufox.....	31
Figura 3.3 Mapa de cobertura red Sigfox en Ecuador .....	32
Figura 3.4 Mapa de cobertura red Sigfox en la ciudad de Quito .....	32
Figura 3.5 Arquitectura de una PWA.....	33
Figura 3.6 Elementos de Next.js .....	39
Figura 3.7 Arquitectura de flujo de Spring Boot.....	41
Figura 3.8 Funcionamiento de la Aplicación web Progresiva con IoT usando tecnología Sigfox .....	42
Figura 3.9 Esquema de conexiones del prototipo .....	44
Figura 3.10 Diseño digital de la caja del prototipo .....	46
Figura 3.11 Diseño 3D impreso de la caja del prototipo con correa y vistas.....	46
Figura 3.12 Página oficial de la Empresa Teca.....	47
Figura 3.13 Pagina activación de conectividad Sigfox. ....	48
Figura 3.14 Formulario de registro de placa Ufox y creación de cuenta Sigfox .....	48
Figura 3.15 Login back-end Sigfox .....	48
Figura 3.16 Grupo back-end Sigfox.....	49
Figura 3.17 Información Grupo back-end Sigfox .....	49
Figura 3.18 Permisos API back-end Sigfox.....	49
Figura 3.19 Información API back-end Sigfox.....	50
Figura 3.20 Lista de tipos de dispositivos back-end Sigfox .....	50
Figura 3.21 Información del tipo de dispositivo back-end Sigfox.....	51
Figura 3.22 Callbacks back-end Sigfox .....	51
Figura 3.23 Selección del grupo al que pertenecerá el tipo de dispositivo back-end Sigfox.....	52
Figura 3.24 Formulario de creación de nuevo tipo de dispositivo back-end Sigfox. ....	52
Figura 3.25 Lista de dispositivos del usuario back-end Sigfox .....	53
Figura 3.26 Información de dispositivo back-end Sigfox.....	53
Figura 3.27 Lista de mensajes enviados por el dispositivo back-end Sigfox.....	54

Figura 3.28 Boceto preliminar de la historia de usuario H03 Registro de usuario ....	57
Figura 3.29 Boceto preliminar de la historia de usuario H04 Inicio de sesión .....	58
Figura 3.30 Boceto preliminar de la historia de usuario H05 Olvido de contraseña .	58
Figura 3.31 Boceto preliminar de la historia de usuario H06 Panel de administración .....	59
Elaborado por: El Investigador .....	59
Figura 3.32 Boceto preliminar de la sección 1 de la historia de usuario H07 Gestión de usuarios .....	60
Figura 3.33 Boceto preliminar de la sección 2 de la historia de usuario H07 Gestión de usuarios .....	60
Figura 3.34 Boceto preliminar de la sección 3 de la historia de usuario H07 Gestión de usuarios .....	60
Figura 3.35 Boceto preliminar de la sección 1 de la historia de usuario H08 Gestión de mascotas .....	61
Figura 3.36 Boceto preliminar de la sección 2 de la historia de usuario H08 Gestión de mascotas .....	61
Figura 3.37 Boceto preliminar de la sección 3 de la historia de usuario H08 Gestión de mascotas .....	61
Figura 3.38 Boceto preliminar de la sección 1 de la historia de usuario H09 Gestión de dispositivos.....	62
Figura 3.39 Boceto preliminar de la sección 2 de la historia de usuario H09 Gestión de dispositivos.....	62
Figura 3.40 Boceto preliminar de la sección 2 de la historia de usuario H10 Gestión de establecimientos .....	63
Figura 3.41 Boceto preliminar de la sección 2 de la historia de usuario H10 Gestión de establecimientos .....	63
Figura 3.42 Boceto preliminar de la sección 3 de la historia de usuario H10 Gestión de establecimientos .....	63
Figura 3.43 Boceto preliminar de la sección 1 de la historia de usuario H11 Gestión de medicinas.....	64
Figura 3.44 Boceto preliminar de la sección 2 de la historia de usuario H11 Gestión de medicinas.....	64
Figura 3.45 Boceto preliminar de la sección 3 de la historia de usuario H11 Gestión de medicinas.....	64
Figura 3.46 Boceto preliminar de la sección 1 de la historia de usuario H12 Gestión de profesionales .....	65
Figura 3.47 Boceto preliminar de la sección 2 de la historia de usuario H12 Gestión de profesionales .....	65
Figura 3.48 Boceto preliminar de la sección 3 de la historia de usuario H12 Gestión de profesionales .....	66
Figura 3.49 Boceto preliminar de la historia de usuario H13 Gestión de especies....	66
Figura 3.50 Boceto preliminar de la historia de usuario H14 Gestión de razas.....	67

Figura 3.51 Boceto preliminar de la historia de usuario H15 Gestión de tipos de establecimientos .....	67
Figura 3.52 Boceto preliminar de la historia de usuario H16 Gestión de tipos de medicinas.....	68
Figura 3.53 Boceto preliminar de la historia de usuario H17 Gestión de profesiones .....	69
Figura 3.54 Boceto preliminar de la historia de usuario H18 Panel de usuario.....	69
Figura 3.55 Boceto preliminar de la sección 1 de la historia de usuario H19 Configuración cuenta de usuario.....	70
Figura 3.56 Boceto preliminar de la sección 2 de la historia de usuario H19 Configuración cuenta de usuario.....	70
Figura 3.57 Boceto preliminar de la sección 1 de la historia de usuario H20 Gestión de las mascotas del usuario .....	71
Figura 3.58 Boceto preliminar de la sección 2 de la historia de usuario H20 Gestión de las mascotas del usuario .....	71
Figura 3.59 Boceto preliminar de la sección 3 de la historia de usuario H20 Gestión de las mascotas del usuario .....	71
Figura 3.60 Boceto preliminar de la sección 1 de la historia de usuario H21 Gestión de los dispositivos del usuario .....	72
Figura 3.61 Boceto preliminar de la sección 2 de la historia de usuario H21 Gestión de los dispositivos del usuario .....	72
Figura 3.62 Boceto preliminar de la sección 1 de la historia de usuario H22 Secciones de monitorización.....	73
Figura 3.63 Boceto preliminar de la sección 2 de la historia de usuario H22 Secciones de monitorización.....	73
Figura 3.64 Boceto preliminar de la historia de usuario H23 Ubicación de la mascota del usuario en tiempo real .....	73
Figura 3.65 Boceto preliminar de la historia de usuario H24 Historial de ubicaciones de la mascota del usuario en tiempo real .....	74
Figura 3.66 Boceto preliminar de la historia de usuario H25 Temperatura de la mascota del usuario en tiempo real .....	75
Figura 3.67 Boceto preliminar de la historia de usuario H26 Historial de temperaturas de la mascota del usuario en tiempo real .....	75
Figura 3.68 Boceto preliminar de la sección 1 de la historia de usuario H27 Gestión de establecimientos de la mascota .....	76
Figura 3.69 Boceto preliminar de la sección 2 de la historia de usuario H27 Gestión de establecimientos de la mascota .....	76
Figura 3.70 Boceto preliminar de la sección 3 de la historia de usuario H27 Gestión de establecimientos de la mascota .....	77
Figura 3.71 Boceto preliminar de la historia de usuario H28 Contactar establecimiento de la mascota.....	77
Figura 3.72 Boceto preliminar de la sección 1 de la historia de usuario H29 Gestión de medicinas de la mascota.....	78



Figura 3.73 Boceto preliminar de la sección 2 de la historia de usuario H29 Gestión de medicinas de la mascota.....	78
Figura 3.74 Boceto preliminar de la sección 3 de la historia de usuario H29 Gestión de medicinas de la mascota.....	78
Figura 3.75 Boceto preliminar de la sección 1 de la historia de usuario H30 Gestión de profesionales de la mascota.....	79
Figura 3.76 Boceto preliminar de la sección 2 de la historia de usuario H30 Gestión de profesionales de la mascota.....	79
Figura 3.77 Boceto preliminar de la sección 3 de la historia de usuario H30 Gestión de profesionales de la mascota.....	80
Figura 3.78 Boceto preliminar de la historia de usuario H31 Contactar profesional de la mascota.....	80
Figura 3.79 Boceto preliminar de la historia de usuario H32 Consulta de mascotas por secciones .....	81
Figura 3.80 Estructura de carpetas para el desarrollo de servicios web tipo REST en Spring Boot .....	104
Figura 3.81 Estructura de carpetas para el desarrollo de la PWA MonIOpeT en Next.js .....	105
Figura 3.82 Esquema de la base de datos para la PWA MonIOpeT .....	106
Figura 3.83 Vista en computadora del formulario de registro de nuevos usuarios dueños de mascotas .....	107
Figura 3.84 Vista en dispositivo móvil del formulario de registro de nuevos usuarios dueños de mascotas .....	107
Figura 3.85 Vista en computadora del formulario de inicio de sesión de los usuarios .....	108
Figura 3.86 Vista en dispositivo móvil del formulario de inicio de sesión de los usuarios .....	108
Figura 3.87 Vista en computadora del formulario de olvido de contraseña de los usuarios. ....	109
Figura 3.88 Vista en dispositivo móvil del formulario de olvido de contraseña de los usuarios. ....	109
Figura 3.89 Vista en computadora del panel de administración .....	110
Figura 3.90 Vista en computadora de la tabla de información principal de los usuarios con opción de ver la información de cada usuario .....	110
Figura 3.91 Vista en computadora del formulario para guardar un nuevo usuario..	111
Figura 3.92 Vista en computadora del formulario para la actualización y eliminación de información del usuario.....	111
Figura 3.93 Vista en computadora de la tabla de información principal de las mascotas con opción de ver la información de cada mascota.....	112
Figura 3.94 Vista en computadora del formulario para guardar una nueva mascota .....	112
Figura 3.95 Vista en computadora del formulario para la actualización y eliminación de información de la mascota.....	112

Figura 3.96 Vista en computadora de la tabla de información principal de los dispositivos con opción de ver y eliminar la información de cada dispositivo.....	113
Figura 3.97 Vista en computadora del formulario para guardar un nuevo dispositivo .....	113
Figura 3.98 Vista en computadora de la tabla de información de los datos enviados desde el dispositivo .....	113
Figura 3.99 Vista en computadora de la tabla de información principal de los establecimientos con opción de ver la información de cada mascota.....	114
Figura 3.100 Vista en computadora del formulario para guardar un nuevo establecimiento.....	114
Figura 3.101 Vista en computadora del formulario para la actualización y eliminación de información del establecimiento .....	114
Figura 3.102 Vista en computadora de la tabla de información principal de las medicinas con opción de ver la información de cada medicina.....	115
Figura 3.103 Vista en computadora del formulario para guardar una nueva medicina .....	115
Figura 3.104 Vista en computadora del formulario para la actualización y eliminación de información de la medicina .....	115
Figura 3.105 Vista en computadora de la tabla de información principal de los profesionales con opción de ver la información de cada profesional .....	116
Figura 3.106 Vista en computadora del formulario para guardar un nuevo profesional .....	116
Figura 3.107 Vista en computadora del formulario para la actualización y eliminación de información del profesional.....	116
Figura 3.108 Vista en computadora de la tabla y formulario para la gestión de especies .....	117
Figura 3.109 Vista en computadora de la tabla y formulario para la gestión de razas .....	117
Figura 3.110 Vista en computadora de la tabla y formulario para la gestión de los tipos de establecimientos .....	118
Figura 3.111 Vista en computadora de la tabla y formulario para la gestión de los tipos de medicinas.....	118
Figura 3.112 Vista en computadora de la tabla y formulario para la gestión de profesiones .....	119
Figura 3.113 Vista en computadora del panel de usuario .....	119
Figura 3.114 Vista en dispositivo móvil del panel y menú de opciones del usuario	120
Figura 3.115 Vista en computadora del formulario de actualización de perfil de usuario .....	120
Figura 3.116 Vista en computadora del formulario para la actualización de contraseña del usuario .....	121
Figura 3.117 Vista en dispositivo móvil de los formularios de actualización de perfil y contraseña del usuario .....	121

Figura 3.118 Vista en computadora de la información de las mascotas del usuario en cartas con opciones de ingreso a monitoreo, ver y eliminar la información.....	122
Figura 3.119 Vista en computadora del formulario para guardar una nueva mascota del usuario.....	122
Figura 3.120 Vista en computadora del formulario para actualizar la información de una mascota del usuario .....	122
Figura 3.121 Vistas en dispositivo móvil de las opciones de ingreso al monitoreo, ver y eliminar la información de las mascotas del usuario presentada en cartas y los formularios para guardar y actualizar una mascota.....	122
Figura 3.122 Vista en computadora de la información de los dispositivos del usuario en cartas con opción de eliminar la información del dispositivo asignado a una mascota .....	123
Figura 3.123 Vista en computadora del formulario de ingreso de código del dispositivo del usuario y asignación de una mascota. ....	123
Figura 3.124 Vista en dispositivo móvil de la información del dispositivo asignado a una mascota, con opción a eliminar el dispositivo y el formulario de ingreso del código del dispositivo con la asignación de una mascota.....	124
Figura 3.125 Vista en computadora de las secciones de ubicación y temperatura en tiempo real de la mascota del usuario .....	125
Figura 3.126 Vista en computadora de las secciones de gestión de establecimientos, medicinas y profesionales de la mascota del usuario.....	125
Figura 3.127 Vistas en dispositivo móvil de las secciones de ubicación y temperatura en tiempo real, gestión de establecimientos, gestión de medicinas y gestión de profesionales .....	125
Figura 3.128 Vista en computadora de la ubicación en tiempo real de la mascota del usuario.....	126
Figura 3.129 Vista en dispositivo móvil de la ubicación en tiempo real de la mascota del usuario .....	126
Figura 3.130 Vista en computadora del historial de ubicaciones de la mascota del usuario.....	127
Figura 3.131 Vista en dispositivo móvil del historial de ubicaciones de la mascota del usuario.....	127
Figura 3.132 Vista en computadora de la temperatura en tiempo real de la mascota del usuario.....	128
Figura 3.133 Vista en dispositivo móvil de la temperatura en tiempo real de la mascota del usuario .....	128
Figura 3.134 Vista en computadora del historial de temperaturas de la mascota del usuario.....	129
Figura 3.135 Vista en dispositivo móvil del historial de temperaturas de la mascota del usuario.....	129
Figura 3.136 Vista en computadora de la información de los establecimientos de la mascota del usuario en cartas con opciones de ver y eliminar la información del establecimiento.....	130

Figura 3.137 Vista en computadora del formulario para guardar un nuevo establecimiento de la mascota del usuario .....	130
Figura 3.138 Vista en computadora del formulario para actualizar la información del establecimiento de la mascota del usuario .....	130
Figura 3.139 Vistas en dispositivo móvil de las opciones de ver y eliminar la información de los establecimientos de la mascota del usuario presentada en cartas y los formularios para guardar y actualizar un establecimiento.....	131
Figura 3.140 Vista en computadora del formulario para contactar al establecimiento de la mascota del usuario .....	131
Figura 3.141 Vista en dispositivo móvil del formulario para contactar al establecimiento de la mascota del usuario .....	132
Figura 3.142 Vista en computadora de la información de las medicinas de la mascota del usuario en cartas con opciones de ver y eliminar la información de la medicina .....	132
Figura 3.143 Vista en computadora del formulario para guardar una nueva medicina de la mascota del usuario .....	133
Figura 3.144 Vista en computadora del formulario para actualizar la información de la medicina de la mascota del usuario.....	133
Figura 3.145 Vistas en dispositivo móvil de las opciones de ver y eliminar la información de las medicinas de la mascota del usuario presentada en cartas y los formularios para guardar y actualizar una medicina .....	133
Figura 3.146 Vista en computadora de la información de los profesionales de la mascota del usuario en cartas con opciones de ver y eliminar la información del profesional.....	134
Figura 3.147 Vista en computadora del formulario para guardar un nuevo profesional de la mascota del usuario .....	134
Figura 3.148 Vista en computadora del formulario para actualizar la información del profesional de la mascota del usuario .....	135
Figura 3.149 Vistas en dispositivo móvil de las opciones de ver y eliminar la información de los profesionales de la mascota del usuario presentada en cartas y los formularios para guardar y actualizar un profesional .....	135
Figura 3.150 Vista en computadora del formulario para contactar al profesional de la mascota del usuario .....	136
Figura 3.151 Vista en dispositivo móvil del formulario para contactar al profesional de la mascota del usuario .....	136
Figura 3.152 Vista en computadora del formulario para la consulta de las mascotas del usuario por secciones .....	137
Figura 3.153 Vista en computadora de la mascota consultada con la información proporcionada por secciones que gestiona el usuario .....	137
Figura 3.154 Vistas en dispositivo móvil del formulario de consulta de las mascotas del usuario y el resultado de la consulta por secciones que gestiona el usuario presentado en cartas .....	137
Figura 3.155 Resultados de la prueba de carga realizada a la PWA MonIOpeT.....	161

Figura 3.156 Resultados de la prueba de estrés realizada a la PWA MonIOpeT ....	161
Figura 3.157 Resultados de la prueba de estabilidad realizada a la PWA MonIOpeT .....	162
Figura 3.158 Resultados de la prueba de rendimiento realizada con la herramienta de Google Lighthouse a la PWA MonIOpeT. ....	162
Figura 3.159 Registro en Amazon Web Services (AWS).....	166
Figura 3.160 Lista de mensajes enviados por el dispositivo back-end Sigfox.....	166
Figura 3.161 Elección de entorno para una aplicación web.....	166
Figura 3.162 Ingreso de nombre de la aplicación para la creación de un entorno de servidor web .....	167
Figura 3.163 Ingreso de nombre del entorno .....	167
Figura 3.164 Elección de la plataforma en la que esta desarrollada la aplicación web .....	167
Figura 3.165 Elección de modo de carga de la aplicación web y creación de entorno .....	168
Figura 3.166 Opciones de configuración del entorno de la aplicación web .....	168
Figura 3.167 Configuración de la información de la base de datos PostgreSQL en el entorno de la aplicación web.....	169
Figura 3.168 Carga de la aplicación web desarrollada en Spring Boot .....	169
Figura 3.169 Estado de la aplicación web con los servicios web tipo REST en AWS .....	170
Figura 3.170 Repositorio de la PWA desarrollada en Next.js .....	170
Figura 3.171 Registro de usuario de la plataforma Vercel.....	170
Figura 3.172 Importación proyecto PWA de Github .....	171
Figura 3.173 Configuración proyecto PWA con nombre MonIOpeT .....	171
Figura 3.174 Lista de mensajes enviados por el dispositivo back-end Sigfox.....	171
Figura 3.175 Opción de instalación de la PWA MonIOpeT en el navegador Google Chrome.....	172
Figura 3.176 PWA MonIOpeT ejecutada como una aplicación de escritorio en Windows 10 .....	172
Figura 3.177 Opción de instalación de la PWA MonIOpeT en el navegador Google Chrome de sistema operativo móvil Android .....	173
Figura 3.178 Instalación e icono de la PWA instalada MonIOpeT .....	173
Figura 3.179 PWA MonIOpeT ejecutada como una aplicación de nativa en dispositivo móvil Android .....	174
Figura 3.180 Pasos para añadir a la pantalla de inicio la PWA MonIOpeT .....	174
Figura 3.181 Icono de la PWA MonIOpeT en iPhone 13.....	175
Figura 3.182 PWA MonIOpeT ejecutada como una aplicación en dispositivo iPhone 13.....	175

## ÍNDICE DE TABLAS

Tabla 2.1 Proyección del número de habitantes de la ciudad de Quito hasta el 2020	21
Tabla 2.2 Calculo de la muestra de habitantes de la ciudad de Quito.....	21
Tabla 3.1 Comparativa de las 3 tecnologías LPWAN más populares .....	29
Tabla 3.2 Catálogo de zonas Sigfox.....	30
Tabla 3.3 Características de Ufox .....	31
Tabla 3.4 Cuadro de comparación de enfoques de desarrollo Nativo, Hibrido y PWA .....	36
Tabla 3.5 Cuadro comparativo de tecnologías para el desarrollo de PWAs.....	37
Tabla 3.6 Comparación de metodologías ágiles .....	42
Tabla 3.7 Costo total de materiales utilizados en el desarrollo del prototipo .....	43
Tabla 3.8 Definición de roles XP .....	55
Tabla 3.9 Plantilla para Historias de usuario .....	55
Tabla 3.10 Historia de usuario H01 Estructura del proyecto .....	56
Tabla 3.11 Historia de usuario H02 Diseño de la base de datos .....	57
Tabla 3.12 Historia de usuario H03 Registro de usuario .....	57
Tabla 3.13 Historia de usuario H04 Inicio de sesión .....	58
Tabla 3.14 Historia de usuario H05 Olvido de Contraseña .....	58
Tabla 3.15 Historia de usuario H06 Panel de administración .....	59
Tabla 3.16 Historia de usuario H07 Gestión de usuarios.....	59
Tabla 3.17 Historia de usuario H09 Gestión de mascotas .....	60
Tabla 3.18 Historia de usuario H08 Gestión de dispositivos.....	62
Tabla 3.19 Historia de usuario H10 Gestión de establecimientos .....	62
Tabla 3.20 Historia de usuario H11 Gestión de medicinas .....	64
Tabla 3.21 Historia de usuario H12 Gestión de profesionales.....	65
Tabla 3.22 Historia de usuario H13 Gestión de especies.....	66
Tabla 3.23 Historia de usuario H14 Gestión de razas.....	66
Tabla 3.24 Historia de usuario H15 Gestión de tipos de establecimientos.....	67
Tabla 3.25 Historia de usuario H16 Gestión de tipos de medicinas .....	68
Tabla 3.26 Historia de usuario H17 Gestión de profesiones.....	68
Tabla 3.27 Historia de usuario H18 Panel de usuario.....	69
Tabla 3.28 Historia de usuario H19 Configuración cuenta de usuario .....	69
Tabla 3.29 Historia de usuario H20 Gestión de mascotas de usuario.....	70
Tabla 3.30 Historia de usuario H21 Gestión de dispositivos del usuario .....	71
Tabla 3.31 Historia de usuario H22 Secciones de monitorización .....	72
Tabla 3.32 Historia de usuario H23 Ubicación de la mascota del usuario en tiempo real .....	73
Tabla 3.33 Historia de usuario H24 Historial de ubicaciones de la mascota del usuario en tiempo real.....	74
Tabla 3.34 Historia de usuario H25 Temperatura de la mascota del usuario en tiempo real.....	74

Tabla 3.35 Historia de usuario H26 Historial de temperaturas de la mascota del usuario en tiempo real.....	75
Tabla 3.36 Historia de usuario H27 Gestión de establecimientos de la mascota.....	76
Tabla 3.37 Historia de usuario H28 Contactar establecimiento de la mascota.....	77
Tabla 3.38 Historia de usuario H29 Gestión de medicinas de la mascota.....	77
Tabla 3.39 Historia de usuario H30 Gestión de profesionales de la mascota.....	79
Tabla 3.40 Historia de usuario H31 Contactar profesional de la mascota.....	80
Tabla 3.41 Historia de usuario H32 Consulta de mascotas por secciones.....	80
Tabla 3.42 Estimación de historias de usuario.....	82
Tabla 3.43 Plan de entregas.....	83
Tabla 3.44 Historias de usuario de la iteración 1.....	83
Tabla 3.45 Tarea T01 Definir estructura de carpetas para el desarrollo de los servicios web tipo REST.....	83
Tabla 3.46 Tarea T02 Definir estructura de carpetas para el desarrollo de la PWA..	84
Tabla 3.47 Tarea T03 Definir las entidades que intervendrán en el diseño de la base de datos.....	84
Tabla 3.48 Tarea T04 Desarrollo de seguridad mediante json web token (JWT).....	84
Tabla 3.49 Tarea T05 Desarrollo del servicio web tipo REST para el registro de nuevos usuarios dueños de mascotas.....	84
Tabla 3.50 Tarea T06 Desarrollo de interfaz gráfica para el registro de nuevos usuarios dueños de mascotas.....	85
Tabla 3.51 Tarea T07 Desarrollo de servicio web tipo REST para el inicio de sesión de los usuarios.....	85
Tabla 3.52 Tarea T08 Desarrollo de interfaz gráfica para el inicio de sesión de los usuarios.....	85
Tabla 3.53 Tarea T09 Desarrollo de servicio web tipo REST en caso de olvido de contraseña de los usuarios dueños de mascotas.....	86
Tabla 3.54 Tarea T10 Desarrollo de interfaz gráfica en caso de olvido de la contraseña de los usuarios dueños de mascotas.....	86
Tabla 3.55 Historias de usuario de la iteración 2.....	86
Tabla 3.56 Tarea T11 Desarrollo de interfaz gráfica para el panel de administración.....	87
Tabla 3.57 Tarea T12 Desarrollo de servicio web tipo REST para la gestión de los usuarios.....	87
Tabla 3.58 Tarea T13 Desarrollo de interfaz gráfica para la gestión de los usuarios	87
Tabla 3.59 Tarea T14 Desarrollo de servicio web tipo REST para la gestión de mascotas.....	87
Tabla 3.60 Tarea T15 Desarrollo de interfaz gráfica para la gestión de mascotas ....	88
Tabla 3.61 Tarea T16 Desarrollo de servicio web tipo REST para la gestión de dispositivos.....	88
Tabla 3.62 Tarea T17 Desarrollo de servicio web tipo REST para obtener datos enviados desde los dispositivos.....	88

Tabla 3.63 Tarea T18 Desarrollo de interfaz gráfica para la gestión de los dispositivos .....	88
Tabla 3.64 Tarea T19 Desarrollo de servicio web tipo REST para la gestión de establecimientos .....	89
Tabla 3.65 Tarea T20 Desarrollo de interfaz gráfica para la gestión de establecimientos .....	89
Tabla 3.66 Tarea T21 Desarrollo de servicio web tipo REST para la gestión de medicinas.....	89
Tabla 3.67 Tarea T22 Desarrollo de interfaz gráfica para la gestión de medicinas...	89
Tabla 3.68 Tarea T23 Desarrollo de servicio web tipo REST para la gestión de profesionales .....	90
Tabla 3.69 Tarea T24 Desarrollo de interfaz gráfica para la gestión de profesionales .....	90
Tabla 3.70 Tarea T25 Desarrollo de servicio web tipo REST para la gestión de especies .....	90
Tabla 3.71 Tarea T26 Desarrollo de interfaz gráfica para la gestión de especies.....	90
Tabla 3.72 Tarea T27 Desarrollo de servicio para la gestión de razas .....	90
Tabla 3.73 Tarea T28 Desarrollo de interfaz gráfica para la gestión de razas.....	91
Tabla 3.74 Tarea T29 Desarrollo de servicio web tipo REST para la gestión de tipos de establecimientos .....	91
Tabla 3.75 Tarea T30 Desarrollo de interfaz gráfica para la gestión de tipos de establecimientos .....	91
Tabla 3.76 Tarea T31 Desarrollo de servicio web tipo REST para la gestión de tipos de medicinas.....	91
Tabla 3.77 Tarea T32 Desarrollo de interfaz gráfica para la gestión de tipos de medicinas.....	92
Tabla 3.78 Tarea T33 Desarrollo de servicio web tipo REST para la gestión de profesiones .....	92
Tabla 3.79 Tarea T34 Desarrollo de interfaz gráfica para la gestión de profesiones.	92
Tabla 3.80 Historias de usuario de la iteración 3 .....	92
Tabla 3.81 Tarea T35 Desarrollo de interfaz gráfica para el panel de usuario .....	93
Tabla 3.82 Tarea T36 Desarrollo de interfaz gráfica para la configuración de la cuenta del usuario .....	93
Tabla 3.83 Tarea T37 Desarrollo de interfaz gráfica para la gestión de las mascotas del usuario .....	93
Tabla 3.84 Tarea T38 Desarrollo de interfaz gráfica para la gestión de los dispositivos del usuario .....	94
Tabla 3.85 Historias de usuario de la iteración 4 .....	94
Tabla 3.86 Tarea T39 Desarrollo de interfaz gráfica para mostrar las secciones de monitorización.....	94
Tabla 3.87 Tarea T40 Desarrollo de interfaz gráfica para mostrar la ubicación de la mascota del usuario en tiempo real .....	95



Tabla 3.88 Tarea T41 Desarrollo de interfaz gráfica para mostrar el historial de ubicaciones de la mascota del usuario en tiempo real.....	95
Tabla 3.89 Tarea T42 Desarrollo de interfaz gráfica para mostrar la temperatura de la mascota del usuario en tiempo real .....	95
Tabla 3.90 Tarea T43 Desarrollo de interfaz gráfica para mostrar el historial de temperaturas de la mascota del usuario en tiempo real.....	96
Tabla 3.91 Tarea T44 Desarrollo de interfaz gráfica para la gestión de establecimientos de la mascota.....	96
Tabla 3.92 Tarea T45 Desarrollo de servicio web tipo REST para contactar el establecimiento de la mascota.....	96
Tabla 3.93 Tarea T46 Desarrollo de interfaz gráfica para contactar el establecimiento de la mascota.....	96
Tabla 3.94 Tarea T47 Desarrollo de interfaz gráfica para la gestión de medicinas de la mascota.....	97
Tabla 3.95 Tarea T48 Desarrollo de interfaz gráfica para la gestión de profesionales de la mascota.....	97
Tabla 3.96 Tarea T49 Desarrollo de servicio web tipo REST para contactar al profesional de la mascota.....	97
Tabla 3.97 Tarea T50 Desarrollo de interfaz gráfica para contactar al profesional de la mascota.....	98
Tabla 3.98 Tarea T51 Desarrollo de servicio web tipo REST para la consulta de mascotas por secciones .....	98
Tabla 3.99 Tarea T52 Desarrollo de interfaz gráfica para la consulta de mascotas por secciones .....	98
Tabla 3.100 Tarjeta CRC Clase Autenticación.....	99
Tabla 3.101 Tarjeta CRC Clase Notificación .....	99
Tabla 3.102 Tarjeta CRC Clase Usuario.....	99
Tabla 3.103 Tarjeta CRC Clase Mascota.....	100
Tabla 3.104 Tarjeta CRC Clase Dispositivo .....	100
Tabla 3.105 Tarjeta CRC Clase Detalle Dispositivo .....	100
Tabla 3.106 Tarjeta CRC Clase Establecimiento.....	101
Tabla 3.107 Tarjeta CRC Clase Medicina .....	101
Tabla 3.108 Tarjeta CRC Clase Profesional .....	101
Tabla 3.109 Tarjeta CRC Clase Especie.....	102
Tabla 3.110 Tarjeta CRC Clase Raza .....	102
Tabla 3.111 Tarjeta CRC Clase Tipo de Establecimiento .....	102
Tabla 3.112 Tarjeta CRC Clase Tipo de Medicina .....	103
Tabla 3.113 Tarjeta CRC Clase Profesión.....	103
Tabla 3.114 Prueba de aceptación P01 Estructura del proyecto .....	150
Tabla 3.115 Prueba de aceptación P02 Diseño de la base de datos .....	150
Tabla 3.116 Prueba de aceptación P03 Registro de usuarios.....	151
Tabla 3.117 Prueba de aceptación P04 Inicio de sesión .....	151
Tabla 3.118 Prueba de aceptación P05 Olvido de contraseña .....	151

Tabla 3.119 Prueba de aceptación P06 Panel de administración .....	152
Tabla 3.120 Prueba de aceptación P07 Gestión de usuarios .....	152
Tabla 3.121 Prueba de aceptación P08 Gestión de mascotas .....	152
Tabla 3.122 Prueba de aceptación P09 Gestión de dispositivos .....	153
Tabla 3.123 Prueba de aceptación P10 Gestión de establecimientos .....	153
Tabla 3.124 Prueba de aceptación P11 Gestión de medicinas .....	153
Tabla 3.125 Prueba de aceptación P12 Gestión de profesionales .....	153
Tabla 3.126 Prueba de aceptación P13 Gestión de especies .....	154
Tabla 3.127 Prueba de aceptación P14 Gestión de razas .....	154
Tabla 3.128 Prueba de aceptación P15 Gestión de tipos de establecimientos .....	154
Tabla 3.129 Prueba de aceptación P16 Gestión de tipos de medicinas .....	155
Tabla 3.130 Prueba de aceptación P17 Gestión de profesiones .....	155
Tabla 3.131 Prueba de aceptación P18 Panel de usuario .....	155
Tabla 3.132 Prueba de aceptación P19 Configuración cuenta de usuario .....	156
Tabla 3.133 Prueba de aceptación P20 Gestión de las mascotas del usuario .....	156
Tabla 3.134 Prueba de aceptación P21 Gestión de dispositivos del usuario .....	156
Tabla 3.135 Prueba de aceptación P22 Secciones de monitorización .....	157
Tabla 3.136 Prueba de aceptación P23 Ubicación de la mascota del usuario en tiempo real .....	157
Tabla 3.137 Prueba de aceptación P24 Historial de ubicaciones de la mascota del usuario en tiempo real .....	157
Tabla 3.138 Prueba de aceptación P25 Temperatura de la mascota del usuario en tiempo real .....	158
Tabla 3.139 Prueba de aceptación P26 Historial de temperaturas de la mascota del usuario en tiempo real .....	158
Tabla 3.140 Prueba de aceptación P27 Gestión de establecimientos de la mascota .....	158
Tabla 3.141 Prueba de aceptación P28 Contactar establecimiento de la mascota .....	159
Tabla 3.142 Prueba de aceptación P29 Gestión de medicinas de la mascota .....	159
Tabla 3.143 Prueba de aceptación P30 Gestión de profesionales de la mascota .....	159
Tabla 3.144 Prueba de aceptación P31 Contactar profesional de la mascota .....	160
Tabla 3.145 Prueba de aceptación P32 Consulta de mascotas por secciones .....	160
Tabla 3.146 Prueba de usabilidad PU01 Administrador .....	163
Tabla 3.147 Prueba de usabilidad PU02 Usuario .....	164
Tabla 3.148 Prueba de usabilidad PU03 Usuario .....	165

## RESUMEN EJECUTIVO

Los animales callejeros en la actualidad se han vuelto un grave problema en la ciudad de Quito, ya que existen varios factores por los que la población de estos animales va en aumento. Uno de los principales factores es la pérdida de mascotas, donde sus dueños no usaban o desconocían herramientas para monitorearlas y mantenerlas seguras.

El presente proyecto tiene como objetivo aprovechar la tecnología del internet de las cosas o IoT mediante la red de área amplia de baja potencia o LPWAN Sigfox, la cual al estar disponible en la ciudad de Quito es posible usarla para el desarrollo de una aplicación web progresiva o PWA de monitoreo en tiempo real de mascotas, donde los usuarios podrán administrar los establecimientos, medicinas, profesionales, ubicación y temperatura en tiempo real de sus mascotas.

Para hacer uso de la LPWAN Sigfox se desarrolló un prototipo de dispositivo para el monitoreo de mascotas usando la placa de desarrollo Ufox fabricado por la empresa Peruana Teca, la cual al estar basada en la placa Arduino Leonardo, dio la posibilidad de incorporar un módulo GPS NEO-6M para obtener la ubicación y sensor infrarrojo MLX90614 con el fin de obtener la temperatura de la mascota, además como medio de protección para estos componentes se diseñó una caja en 3D la cual fue impresa en filamento de ácido poliláctico o PLA.

La PWA fue desarrollada siguiendo las fases de planificación, diseño, codificación, pruebas y despliegue de la metodología Extreme Programming o XP usando dos herramientas de desarrollo web, donde el front-end o la capa de presentación de la PWA se construyó utilizando Next.js, el cual es un framework de la librería React.js que facilita el desarrollo de PWAs mediante instalaciones y configuraciones simples, el back-end o la capa de acceso a datos de la PWA fue construida con Spring Boot, la cual es una tecnología que permite el desarrollo de servicios web tipo REST, los cuales proporcionan rapidez, seguridad y escalabilidad.

**Palabras clave:** sigfox, pwa next.js, spring boot, monitoreo de mascotas

## ABSTRACT

Currently, stray animals have become a serious problem in the city of Quito, as there are several factors why the population of these animals is increasing. One of the main factors is the loss of pets, where their owners didn't use or didn't know the tools to monitor them and keep them safe.

This project aims to take advantage of the technology of the Internet of Things or IoT through the low power wide area network or LPWAN Sigfox, which is available in the city of Quito, it is possible to use it for the development of a progressive web application or PWA for real-time monitoring of pets, where users can manage the facilities, medicines, professionals, location and temperature in real time of their pets.

To make use of the LPWAN Sigfox, a prototype device for monitoring pets was developed using the Ufox development board manufactured by the Peruvian company Teca, which being based on the Arduino Leonardo board, gave the possibility of incorporating a GPS NEO-6M module to obtain the location and infrared sensor MLX90614 in order to obtain the temperature of the pet, also as a means of protection for these components a 3D box was designed which was printed in polylactic acid filament or PLA.

The PWA was developed following the planning, design, coding, testing and deployment phases of the Extreme Programming or XP methodology using two web development tools, where the front-end or presentation layer of the PWA was built using Next.js, which is a framework of the React.js library that facilitates the development of PWAs through simple installations and configurations, the back-end or data access layer of the PWA was built with Spring Boot, which is a technology that allows the development of RESTful web services, which provide speed, security and scalability.

**Keywords:** sigfox, pwa next.js, spring boot, pet monitoring

## INTRODUCCIÓN

El trabajo de investigación titulado “APLICACIÓN WEB PROGRESIVA (PWA) CON IOT PARA EL MONITOREO DE MASCOTAS EN TIEMPO REAL EN LA CIUDAD DE QUITO”, está organizado por los capítulos que se detallan a continuación.

**CAPÍTULO I MARCO TEÓRICO:** Consta del tema del proyecto de investigación y se define la contextualización del problema planteado a nivel maso meso y micro, para así justificar el desarrollo de la investigación mediante los objetivos generales y específicos, como el objetivo general como eje principal.

**CAPÍTULO II METODOLOGÍA:** Se detallan los materiales empleados en la investigación y los métodos usados como la modalidad de investigación, población y muestra, recolección de información, procesamiento y análisis de datos.

**CAPÍTULO III RESULTADOS Y DISCUSIÓN:** Contiene el análisis y discusión de los resultados correspondientes a la metodología usada en la investigación en este caso la metodología Extreme Programming o XP.

**CAPÍTULO IV CONCLUSIONES Y RECOMENDACIONES:** Se establecen las conclusiones y recomendaciones las cuales resultaron del desarrollo del proyecto de investigación.

# CAPITULO I MARCO TEÓRICO

## 1.1 Tema de Investigación

APLICACIÓN WEB PROGRESIVA (PWA) CON IOT PARA EL MONITOREO DE MASCOTAS EN TIEMPO REAL EN LA CIUDAD DE QUITO.

## 1.2 Antecedentes Investigaciones

### 1.2.1 Contextualización del problema

La sobrepoblación de animales callejeros es uno de los problemas globales más serios que causa impactos negativos en la sociedad, en el medio ambiente y en la salud pública. Según la Organización Mundial de la Salud (OMS), el número estimado de perros callejeros fue de aproximadamente de 200 millones en todo el mundo entre 2009 y 2010 [1]. Según la Sociedad Americana para la Prevención de la Crueldad hacia los Animales (ASPCA) en uno de sus boletines publicados en 2017, uno de cada siete propietarios perdió a sus mascotas como perro o gato en los últimos cinco años y con ello ha aumentado la población de animales callejeros[2].

Los sistemas de rastreo GPS es una de las tecnologías, que actualmente se han tomado en cuenta por parte de los amantes de las mascotas alrededor del mundo para evitar su pérdida, ya que para ellos es una parte importante de su familia. [2]

En Latinoamérica existen varios países en vías de desarrollo, donde las leyes y políticas públicas de distintos territorios no ofrecen realizar cambios significativos para mejorar la frágil situación de aquellos animales que viven en las calles y ponen en peligro la salud de los ciudadanos al tener que convivir con toda la insalubridad que los perros de la calle producen, ya que no cuentan con los cuidados apropiados [3].

Ecuador no es la excepción del problema ya que cuenta con aproximadamente 300.000 perros callejeros a nivel nacional. Un estudio realizado para censar a los perros callejeros en el Distrito Metropolitano de Quito en el año 2018 arrojó el resultado final que existe 1 perro callejero por cada 22 habitantes [3].

Lamentablemente en el diario vivir se observa en muchas ciudades del Ecuador, carteles o anuncios digitales en redes sociales dando recompensas por información de mascotas extraviadas, estos antecedentes no quieren decir que las mascotas tengan

malos dueños ya que por naturaleza los animales tienden a ser libres, curiosos, y por ingenuidad suelen perderse, ocasionando así un problema al no poder ser localizadas inmediatamente. [4].

Actualmente en el Ecuador ya existe la implantación de microchips en mascotas, implantados por veterinarios certificados, pero este método solo sirve para saber los datos del dueño de la mascota a través de una lectura del microchip. Lo cual no soluciona el problema de pérdida de mascotas [4]

Según la investigación realizada por la Fundación Rescate Animal Ecuador (FRAE) sobre animales perdidos entre los meses de enero y junio del 2021. La FRAE recibió 63 reportes de mascotas perdidas, dando como resultado el aumento del número de animales callejeros en la ciudad de Quito [5].

La población de perros callejeros en Quito se ha incrementado en los últimos años, por lo que se necesita una solución que otorgue una perspectiva diferente frente al entorno que se vive a diario con animales de la calle y los problemas que traen a la sociedad puesto que no existen políticas estrictas de control en la ciudad, ni tampoco una concientización por parte de las autoridades locales que permita un cambio de mentalidad en la sociedad [4].

La escasez de plataformas tecnológicas que permitan el monitoreo de mascotas en la ciudad de Quito ha ocasionado que los dueños no cuenten con una herramienta la cual ayude con la seguridad de sus mascotas, por lo que es necesario el desarrollo de una aplicación web progresiva con internet de las cosas (IoT) para el monitoreo de mascotas en tiempo real en la ciudad de Quito.

## **1.2.2 Fundamentación teórica**

### **1.2.2.1 Monitoreo**

El monitoreo consiste en hacer observaciones confiables en la naturaleza para detectar, medir, evaluar y sacar conclusiones sobre cambios que ocurren en las especies y ecosistemas en el tiempo y el espacio, de manera natural o como consecuencia de intervenciones humanas deliberadas o involuntarias [6].

Permite identificar de manera organizada la calidad de desempeño de un procedimiento, proyecto, programa o política, generalmente sirve para observar

cambios que se producen en el tiempo y determinar acciones sobre el sistema ya sea de una forma manual o automatizada en el momento oportuno [7].

#### **1.2.2.2 Sistemas de monitoreo**

Un sistema de monitorización consta de un sensor que convierte una entrada física en una salida eléctrica y de un registrador que digitaliza y almacena la señal del sensor analógico en un dispositivo de almacenamiento [8].

Además del hardware utilizado, la extracción de características relacionadas con el comportamiento, el daño o la fatiga de los datos medidos y su posterior análisis y uso, así como la difusión de los resultados asociados, son otros componentes importantes de dichos sistemas [8].

#### **1.2.2.3 Sistemas de monitoreo de mascotas**

Un sistema de monitoreo de mascota es creado para ayudar a los dueños de mascotas, a rastrear la ubicación de su mascota proporcionado por un sensor GPS unido a un objeto que las mascotas pueden llevar. El sistema también ayuda a los usuarios a evitar incidentes como la pérdida de mascotas, al tiempo que reduce la ansiedad y la tristeza de los dueños [9].

#### **1.2.2.4 Internet de las cosas**

El concepto Internet de las Cosas o Internet of Things (IoT) nació de la mano de Kevin Ashton, directivo de Procter & Gamble y también miembro de la comunidad de Desarrollo de Identificación por Radiofrecuencia (RFID) en 1999. Recientemente este concepto se ha vuelto más relevante para el mundo práctico en gran parte debido al crecimiento de los dispositivos móviles, la comunicación integrada, la computación en la nube y el análisis de datos [10].

El IoT se refiere a un tipo de red para conectar cualquier cosa con internet basada en protocolos estipulados a través de equipos de detección de información para realizar intercambios de información y comunicaciones con el fin de lograr reconocimientos inteligentes, posicionamiento, rastreo, monitoreo y administración [10].



### 1.2.2.5 Redes de área amplia de baja potencia

Una de las tecnologías IoT que más despunta son las Redes de Área Amplia de Baja Potencia o Low Power Wide Area Network (LPWAN), que optimiza el alcance, la vida de la batería y el coste de producción a cambio de una menor frecuencia de transmisión de datos, haciéndolo así un estándar idóneo para el IoT [11].

LPWAN es una tecnología de transporte inalámbrico de datos que hoy en día se entiende como uno de los protocolos básicos para la implementación de IoT. Las redes LPWAN son ideales para situaciones donde los dispositivos necesitan enviar pocos datos sobre un área amplia mientras que es necesario que la vida de la batería se prolongue a lo largo de muchos años. Las principales redes LPWAN existentes son: Sigfox, LoRaWAN y NB-IoT [11].

La arquitectura de una red LPWAN se basa en topología tipo estrella, es decir, muchos dispositivos finales, también llamados nodos, se comunican directamente con un dispositivo central, el cual tiene funciones de enrutamiento para direccionar toda la información suministrada por los nodos, a una estación base o servidor, esta estación le brinda al administrador de red la capacidad de instrumentar, monitorear y ejecutar acciones ante eventos y fenómenos en un entorno especificado [12].

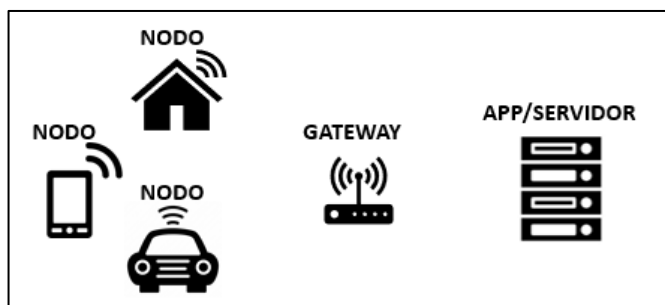


Figura 1.1 Esquema red LPWAN  
Elaborado por: El Investigador

### Sigfox

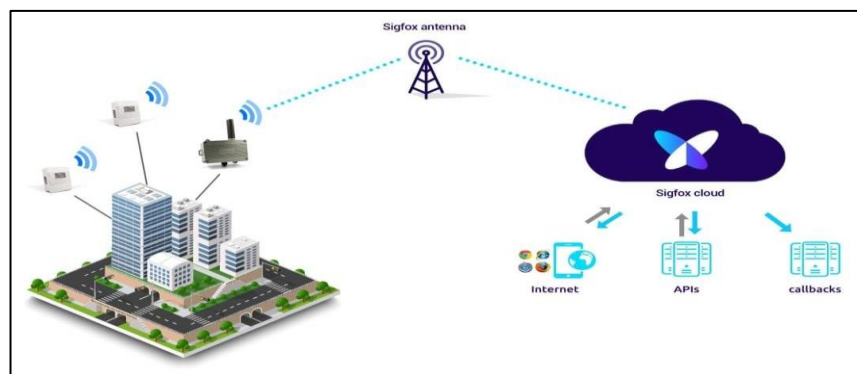
Sigfox fue fundada por Ludovic Le Moan y Christophe Fournet en 2010 con la visión de conectar todos los objetos de nuestro mundo físico con el universo digital. Los fundadores de Sigfox construyeron una red global dedicada al Internet de las cosas basada en datos pequeños, de largo alcance y de bajo consumo que ofrece un servicio de conectividad de extremo a extremo.

Sigfox es una red LPWAN que al implementar tecnología de banda ultra estrecha (UNB) en las bandas ISM, las cuales son usadas para fines industriales, científicos y médicos, utiliza 192 kHz de las bandas disponibles para el envío de mensajes. Las bandas de operación dependen de la región, por ejemplo, en Europa las frecuencias son 868 y 868.2 MHz y en el resto de los continentes son 902-928 MHz [13].

Los datos transmitidos por Sigfox tienen un tamaño de 12 bytes con un límite de 140 mensajes al día en enlace ascendente y 4 mensajes en el enlace descendente con un tamaño de 8 bytes. Cada mensaje tiene un ancho de banda de 100 Hz y la tasa de transmisión de datos varía de 100-600 bps de acuerdo con la región, además el consumo energético en cada transmisión es de 10 mA a 50 mA [13].

Las principales características de Sigfox son:

- Comunicación inalámbrica.
- Bajo consumo de potencia y largo alcance.
- Cobertura amplia [14].



**Figura 1.2 Esquema de funcionamiento de Sigfox**  
Elaborado por: [15]

## **Nb – IoT**

El IoT de Banda Estrecha o Narrowband IoT (NB-IoT) es una nueva tecnología de radio de banda estrecha diseñada específicamente para IoT, que se puede implementar directamente en el sistema global para comunicaciones móviles (GSM) o en redes de evolución a largo plazo (LTE) con el objetivo de reducir el costo de despliegue [16].

NB-IoT ocupa un ancho de banda de frecuencia de 200 KHz, que corresponde a un bloque de recursos en la transmisión GSM y LTE. El protocolo de comunicación NB-

IoT se basa en el protocolo LTE, donde NB-IoT reduce las funcionalidades del protocolo LTE al mínimo y las mejora según sea necesario para las aplicaciones de IoT. Las especificaciones de NB-IoT fija una potencia de emisión de 23 dBm, con los que se consiguen valores elevados de cobertura, de hasta 164 dB de atenuación compensable. Las velocidades de transmisión obtenidas oscilan entre los 160 kbps y los 250 kbps, tanto en subida como en bajada [17].

### **LoraWan**

LoRa es una tecnología de modulación de radio frecuencia para redes LPWAN. El nombre, LoRa, es una referencia a los enlaces de datos de muy largo alcance que permite esta tecnología. Creado por Semtech para estandarizar las LPWAN, LoRa proporciona comunicaciones de largo alcance: hasta tres millas en áreas urbanas y hasta 10 millas o más en áreas rurales [18].

Una característica principal de Lora son los requisitos de energía ultra bajos, lo que permite la creación de dispositivos que funcionan con baterías que pueden durar hasta 10 años. Implementada en una topología en estrella, una red basada en el protocolo abierto LoRaWAN es perfecta para aplicaciones que requieren una comunicación interna profunda o de largo alcance entre una gran cantidad de dispositivos que tienen requisitos de bajo consumo de energía y que recopilan pequeñas cantidades de datos [18].

Lora ofrece un equilibrio entre sensibilidad y velocidad de datos, al tiempo que opera en un canal de ancho de banda fijo de 125 KHz o 500 KHz para canales de subida, y 500 KHz para canales de bajada. Además, utiliza factores de propagación ortogonales. Esto permite a la red preservar la vida de la batería de los nodos finales conectados mediante la optimización adaptativa de los niveles de potencia y las tasas de datos de cada nodo final [18].

#### **1.2.2.6 Arduino**

Arduino es una plataforma informática física de código abierto para crear objetos interactivos que funcionan solos o colaboran con el software de una computadora. Arduino fue diseñado para artistas, diseñadores y otras personas que desean incorporar computación física en sus diseños. El hardware y software de Arduino es de código

abierto por lo que la filosofía de código abierto fomenta una comunidad que comparte su conocimiento con generosidad [19].

Arduino cuenta con placas electrónicas, las cuales están listas para ser usadas y su trabajo es integrar todos los componentes y conexiones necesarias para que un sensor o un dispositivo funcione y como resultado se presente un producto listo para ser usado. [20].

#### **1.2.2.7 Sistema web**

Los sistemas web son aquellos que se alojan en un servidor en internet o sobre una red local, pueden ser utilizados sobre cualquier navegador web sin importar el sistema operativo, por lo que resulta beneficioso para las empresas que lo utilizan, como también para los usuarios que operan en el sistema ya que se mejoran los costos de producción, la rapidez de obtención de información y la optimización en la realización de tareas. [21].

#### **1.2.2.8 Aplicación web**

Una aplicación Web es una aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor Web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones.

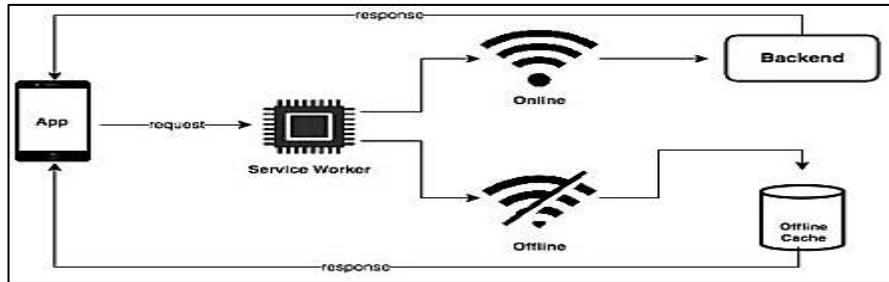
En la actualidad, las aplicaciones Web son cada vez más populares y su uso ha acaparado los ámbitos científico, cultural, académico, empresarial entre otros, y esto es debido a las múltiples ventajas que el usuario tiene respecto a los programas de escritorio. Entre otras, las ventajas que podemos mencionar son: sistema operativo multiplataforma, ejecutadas por cualquier dispositivo informático que tenga conexión a internet, no requiere de la instalación de programas solo un navegador, las copias de seguridad son almacenadas en los servidores, la información que se genera puede ser compartida de forma simultánea por varias personas, el espacio ocupado por los datos está a cargo del servidor y es de fácil uso [22].

#### **1.2.2.9 Aplicación Web Progresiva**

Las Aplicaciones Web Progresivas o Progressive Web Application (PWA) surgen como una opción más simplificada a la hora de desarrollar una aplicación web, son un

nuevo estándar en la era moderna del desarrollo web, son creadas con las mismas tecnologías web, pero se comportan como una aplicación híbrida

Las PWA utilizan service workers en los navegadores modernos por lo que une lo mejor de las tecnologías web y aplicaciones nativas, estas no requieren instalación y están disponibles en un navegador. Algunas tecnologías que intervienen para el desarrollo de las PWA son Manifest File, Responsive Web Design y App Shell [23].



**Figura 1.3 Arquitectura de una Aplicación Web Progresiva**  
Elaborado por: [24]

#### 1.2.2.10 Framework de desarrollo web

Los frameworks de desarrollo web son desarrollados con el propósito de normalizar y estructurar el código del sistema, facilitando un esquema para el desarrollo o la implementación de aplicaciones. Un framework en el desarrollo web reduce el tiempo de elaboración e implementación y ayuda a hacer un trabajo mantenible y escalable.

Un framework aporta funcionalidad extendida a un lenguaje de programación, automatiza muchos de los patrones de programación para orientarlos a un determinado propósito, proporcionando una estructura al código, mejorándolo y haciéndolo más entendible y sostenible.

El usar un framework permite dividir la aplicación en tres capas las cuales son las siguientes.

- La lógica de presentación que administra las interacciones entre el usuario y el software.
- La lógica de datos que permite el acceso a un agente de almacenamiento persistente u otros.
- La lógica de dominio o de negocio, que manipula los modelos de datos de acuerdo con los comandos recibidos desde la presentación [25].

### **1.2.2.11 Tecnologías front-end**

Dentro del desarrollo de aplicaciones web, existe el uso de las tecnologías con las que interactúa directamente el usuario. Normalmente estas tecnologías son desarrolladas en los lenguajes de HTML, CSS y JavaScript; también se usan las herramientas de diseño gráfico como Photoshop o Fireworks.

El objetivo de una tecnología front-end es desarrollar la interfaz gráfica de usuario (GUI), buscando una experiencia de uso bien valorada por el usuario final, siendo en algunos casos necesario hacer investigación, estudios y pruebas para llegar a este fin.

Es importante recalcar que, dentro del desarrollo de las aplicaciones web es posible desarrollar el front-end de la aplicación sin contar con una aplicación back-end que interactúe con la base de datos [26].

#### **Angular**

Angular es un framework de código abierto respaldado por Google. En comparación con React o Vue, que son en su tamaño y características más comparables a las bibliotecas, Angular es un marco de trabajo completo con una arquitectura claramente definida y funciones listas para usarse, incluyendo solicitudes HTTP, utilidades para la construcción de formularios, inyección de dependencias, animaciones, internacionalización y un corredor de pruebas.

Todo el código de Angular también está escrito en TypeScript. Por lo que, se recomienda el uso de TypeScript en las aplicaciones de Angular. Del mismo modo, Angular tiene incorporado RxJs, una popular biblioteca que soporta el paradigma de la programación reactiva en JavaScript.

La programación reactiva es un paradigma de programación asíncrona que se ocupa de los flujos de datos y de la propagación de los cambios. RxJs se basa en el patrón del observador, conocido en la teoría de la ingeniería del software, creando flujos de datos llamados observables. Los observables pueden ser filtrados, mapeados o alterados de alguna manera a través de numerosos operadores encadenables. Uno o varios observadores pueden suscribirse al observable y recibir actualizaciones cada vez que los datos observados cambien. RxJs se utiliza directamente en partes del marco de

trabajo de Angular, principalmente en las partes relacionadas con el enrutamiento y la solicitud de HTTP [27].

## **Vue**

Vue es un framework de código abierto para la creación de interfaces de usuario, desarrollado puramente por la comunidad opensource. Fue lanzado inicialmente en 2014. Vue se ocupa principalmente de la capa de vista de una aplicación, pero hay bibliotecas complementarias opcionales que proporcionan características como la gestión de estados y el enrutamiento, directamente a cargo del equipo de desarrollo de Vue.

Las aplicaciones Vue se escriben generalmente en JavaScript plano, con plantillas en el marcado ampliado de directivas y elementos específicos de Vue. Una ventaja considerable de Vue es que puede ampliarse fácilmente mediante la importación de otras bibliotecas Vue que soportan aplicaciones complejas, pero también puede reducirse. En la versión más básica de la biblioteca, se puede importar como un script externo directamente en HTML, y no necesita un sistema de construcción externo [27].

## **React**

React es una biblioteca de código abierto desarrollada por Facebook, lanzada por primera vez en 2013. El núcleo de esta biblioteca se centra en la construcción de una interfaz de usuario reactiva basada en componentes con vistas declarativas. No incluye características como el enrutamiento, la gestión de estados o las solicitudes HTTP, y los desarrolladores tienen la libertad de elegir cualquier biblioteca desarrollada por terceros. React está escrito en JavaScript, pero es posible escribir aplicaciones React en TypeScript.

En React, las vistas se crean mediante programación utilizando JSX. JSX es una extensión de la sintaxis de JavaScript, que se asemeja al lenguaje de plantillas, pero permite expresiones completas de JavaScript dentro de los bloques declarados [27].

## **Next.js**

Next.js es un framework ligero de React que se utiliza para desarrollar aplicaciones estáticas y renderizadas en el servidor.

Next.js inicia con una configuración mínima y flexible que se puede ampliar para adaptarse al propósito de la aplicación. Next.js utiliza un directorio de páginas donde se almacenan las páginas. Una página es un componente React que se asocia con una ruta basada en su nombre de archivo. Al utilizar el directorio de páginas, Next.js proporciona a la página un enrutamiento automáticamente.

Next.js tiene diferentes tipos de renderización del lado del servidor, los cuales son:

- Generación estática que obtiene los datos en tiempo de construcción
- Renderizado del lado del servidor que obtiene y renderiza los datos para cada solicitud [28].

#### **1.2.2.12 Tecnologías back-end**

Al desarrollar aplicaciones web se implican las actividades realizadas del lado del servidor; las cuales son tareas de base de datos y los servidores de aplicaciones que el usuario no puede visualizar en el explorador de Internet.

Los lenguajes usados comúnmente son PHP, Java, Ruby, .NET, Python, entre otros, los cuales son los encargados de interactuar con la base de datos [26].

#### **1.2.2.13 Servicios web tipo REST**

El termino Transferencia de Estado Representacional o Representational State Transfer (REST) hace referencia a una arquitectura de software que se encarga de representar la transferencia de datos apoyándose en el protocolo HTTP para la transferencia de información entre máquinas, esta arquitectura de software es principalmente usada para la construcción de servicios web, los cuales, son llamados servicios RESTful

Un servicio RESTful es básicamente un servicio web que está basado en la arquitectura REST, está centrado en recursos y el acceso a los mismos, además su prioridad es que sea sencillo y rápido, por ello tienen una identificación de los mencionados recursos mediante URLs para un direccionamiento global donde es posible usar los principales métodos REST los cuales son: put, delete, get y post [29].



#### **1.2.2.14 Jason Web Token**

JSON Web Token (JWT) es un estándar abierto (RFC 7519) que define una forma compacta y autónoma de transmitir información de forma segura entre las partes como un objeto JSON.

La información puede ser verificada y confiable porque está firmada digitalmente. En JWT los tokens firmados pueden verificar la integridad de las afirmaciones contenidas en ellos, mientras que los tokens cifrados ocultan esas afirmaciones a otras partes. Cuando los tokens se firman utilizando pares de claves públicas/privadas, la firma también certifica que sólo la parte que posee la clave privada es la que lo ha firmado [30].

#### **1.2.2.15 Spring Boot**

Spring es un Framework de código abierto para el desarrollo de aplicaciones empresariales en Java que posee una gran cantidad de herramientas y utilidades para agilizar el proceso de desarrollo y despliegue de aplicaciones web.

En la actualidad Spring está compuesto por un conjunto de proyectos de gran importancia para el desarrollo de aplicaciones entre ellos esta Spring Boot.

Spring Boot es la extensión más usada de Spring Framework que sigue el enfoque de "Convención sobre configuración", que ayuda a construir aplicaciones basadas en Spring de manera rápida y fácil.

El objetivo principal de Spring Boot es simplificar el desarrollo de aplicaciones por medio de la autogestión de un gran número de configuraciones, tareas y componentes que son necesarios para la ejecución de una aplicación. Con ello se logra que los desarrolladores se enfoquen en el desarrollo de la lógica de negocio del sistema.

Para cumplir con el principal objetivo de Spring Boot posee un conjunto de bibliotecas llamadas starters que contienen una colección de todas las dependencias relevantes preconfiguradas que se necesitan para iniciar una funcionalidad particular. Estas pueden ser usadas solo con incluirla como dependencia de un proyecto.

Las principales características de Spring Boot son las siguientes:

- Posee todas las características de Spring Framework.

- Implementa el principio de diseño de Abierto-Cerrado, donde esta es cerrada a modificación, pero abierto a extensión cuando se desee tener un mayor control de sus componentes.
- Posee servidores de aplicaciones y contenedores de Servlet embebido.
- Eliminar la necesidad de configurar la aplicación por medio de código o XML. Para ello utiliza ficheros: .properties o .yaml.
- Es una gran opción para crear aplicaciones basadas en microservicios.
- Posee configuraciones automáticas para diferentes Frameworks como son Spring Security, Spring Batch y otros.
- Ofrece un amplio soporte a diferentes tecnologías como son bases de datos relaciones y no relacionales, almacenamiento en caché, mensajería, procesamiento por lotes y más.
- Ofrece métricas de la aplicación por medio de una librería llamada Actuator, utilizada para exponer información importante sobre la aplicación que se encuentra en ejecución a través del monitoreo [31].

#### **1.2.2.16 Sistemas de gestión de base de datos**

Un Sistema de Gestión de Bases de Datos (SGBD) es un sistema general de gestión de la información que puede gestionar muchos tipos diferentes de datos, almacenados en la base de datos. En los SGBD relacionales los datos se definen con relaciones entre los datos que se almacenan en tablas y a las que se puede acceder mediante consultas lógicas, o vistas relacionales.

El SGBD ofrece soporte para las vistas lógicas de los datos que están separadas de las vistas físicas, es decir, de cómo se almacenan realmente los datos en la base de datos. Esta separación se consigue permitiendo que las aplicaciones definan, accedan y actualicen los datos a través de un Lenguaje de Definición de Datos (DDL) y un Lenguaje de Manipulación de Datos (DML) combinados en un lenguaje de consulta declarativo como el lenguaje de consulta relacional SQL.

Un SGBD garantiza la persistencia de los datos, asegurando que no se pierdan en caso de fallo del sistema. La persistencia puede lograrse de muchas maneras, por ejemplo,

almacenando los datos y un registro de los cambios no comprometidos en los datos en el disco.

Tradicionalmente, los SGBD son sistemas autónomos y los usuarios o aplicaciones se ejecutan en procesos separados (a menudo en máquinas distintas) y se les proporciona acceso al SGBD a través de una interfaz cliente/servidor. Para muchas aplicaciones técnicas, los requisitos de rendimiento obligan a una mayor integración entre la aplicación y el SGBD. El SGBD se integra con las aplicaciones y se ejecuta en el mismo proceso. Las aplicaciones con SGBD integrados tienen un acceso rápido a los datos a través de una interfaz de ruta rápida [32].

### **Postgresql**

PostgreSQL es uno de los sistemas de gestión de bases de datos más avanzados y libres. No está controlado por una sola empresa, sino que pertenece a una comunidad global de desarrolladores y empresas que desarrollan el sistema. Es un sistema de gestión de bases de datos objeto-relacional. Es de código abierto y gratuito, PostgreSQL soporta la mayoría de los estándares SQL y ofrece muchas otras características: como: consultas complejas, claves foráneas, triggers, vistas, integridad transaccional y control de concurrencia. Además, se puede ampliar mediante la adición de nuevos tipos de datos, funciones, operadores, funciones de agregación, métodos de índice, lenguajes de procedimiento y debido a la flexibilidad del permiso, cada usuario puede utilizar cualquier propósito libre de carga, modificar y distribuir PostgreSQL, no importa para uso privado, comercial o académico.

PostgreSQL soporta casi todos los tipos de interfaces de cliente de base de datos, y es compatible con diferentes sistemas operativos, como Linux MacOS, Solaris y Windows [33].

#### **1.2.2.17 Plataformas de almacenamiento en la nube**

Este tipo de plataforma permite a los desarrolladores escribir aplicaciones que se ejecutan en la nube, o utilizar servicios proporcionados desde la nube, o ambas cosas. Hoy en día se utilizan diferentes nombres para este tipo de plataformas, como plataforma bajo demanda y plataforma como servicio (PaaS).

Una plataforma de almacenamiento en la nube consta de tres partes que son:

- Una base: Casi todas las aplicaciones utilizan algún software de plataforma en la máquina en la que se ejecutando, donde se incluyen bibliotecas estándar, almacenamiento, y un sistema operativo de base.
- Un grupo de servicios de infraestructura: En un entorno distribuido moderno, es común proporcionar almacenamiento remoto, por ejemplo, servicios de integración, un servicio de identidad, y más.
- Un conjunto de servicios de aplicación: A medida que más y más aplicaciones se orientan a los servicios, las funciones que ofrecen se vuelven accesibles a nuevas aplicaciones [34].

### **Amazon Web Services**

Amazon lanzó Amazon Web Services (AWS) para que otras organizaciones pudieran beneficiarse de la experiencia e inversión de Amazon en el funcionamiento de una infraestructura de TI transaccional y distribuida a gran escala. AWS lleva funcionando desde 2006, y en la actualidad presta servicio a cientos de miles de clientes en todo el mundo. En la actualidad, Amazon.com gestiona una plataforma web global que atiende a millones de clientes y gestiona miles de millones de dólares de comercio cada año.

AWS proporciona potencia informática, almacenamiento y otros servicios en cuestión de minutos y tener la flexibilidad de elegir la plataforma de desarrollo o el modelo de programación que más se adecue a los problemas que intenta resolver. El pago mensual es correspondiente a lo que se utiliza, sin gastos iniciales ni compromisos a largo plazo, lo que convierte a AWS en una forma rentable de ofrecer aplicaciones [35].

### **Vercel**

Vercel es una plataforma en la nube que permite a los desarrolladores alojar sitios web y servicios web que se implementan instantáneamente, escalan automáticamente y no requieren supervisión. Fundada en 2015 por Guillermo Rauch,

Vercel ofrece una interfaz de usuario intuitiva con una configuración mínima para alojar generadores de sitios estáticos como Gatsby o Hugo y varios CMS como Contentful, Prismic o WordPress. Vercel también es una empresa matriz del framework Next.js.

Estas son algunas características importantes a tener en cuenta de Vercel:

- Permite a los desarrolladores configurar sus proyectos en su dominio personalizado un SSL automático gratuito, protegiendo los datos cifrados compartidos entre el servidor y el navegador.
- Se integra directamente con GitHub o GitLab, por lo que, permite a los desarrolladores alojar sitios web estáticos y aplicaciones web que se implementan en cada inserción en ramas o solicitudes de extracción para obtener una vista previa de los cambios.
- Escala automáticamente; no requiere orquestación, y opera y monitorea sin configuración de servidor [36].

#### **1.2.2.18 Metodologías de desarrollo de software**

Una metodología de desarrollo de software se ha definido como un conjunto de actividades, prácticas o procesos que una organización utiliza para desarrollar y mantener el software y los productos asociados, también se define la metodología de desarrollo de software como "la colección documentada de políticas, procesos y procedimientos utilizados por un equipo de desarrollo u organización para practicar la ingeniería de software".

Las prácticas de desarrollo de software son actividades y acciones concretas que una metodología define para ser utilizadas en el proceso, es importante señalar que la metodología de desarrollo de software no implica simplemente el desarrollo de código de software, sino que abarca todo el proceso de desarrollo y mantenimiento de un producto de software [37].

#### **1.2.2.19 Metodologías ágiles de desarrollo de software**

Las metodologías ágiles de desarrollo de software tienen que ver con la entrega rápida de pequeñas piezas de software en funcionamiento para mejorar la satisfacción del cliente utilizando enfoques adaptativos y trabajo en equipo para centrarse en la mejora continua.

Las metodologías ágiles son mayormente preferidas por los startups, por la razón de que el tiempo de desarrollo es más corto, son ideales para empresas de tamaño

pequeño, dan facilidad para gestionar los equipos y además favorece un enfoque liviano para la documentación del software y acepta cambios en cualquier etapa del ciclo de vida del desarrollo de software.

Algunas ventajas que proporcionan las metodologías ágiles son:

- La metodología ágil proporciona satisfacción al cliente con la entrega temprana y continua de software y mejora la gestión del proceso de desarrollo con una mejor relación con el cliente.
- Los principios flexibles de la metodología ágil permiten que los requisitos y las soluciones evolucionen a través de la colaboración autoorganizada entre equipos multifuncionales.
- La metodología ágil promueve la planificación adaptativa, que permite la mejora continua o la respuesta al cambio, y por estar centrada en la solución [37].

## **Kanban**

El método Kanban ofrece un marco de gestión visual del flujo de trabajo del ciclo de vida del desarrollo de software

Kanban se basa en tres prácticas principales los cuales son:

- La visualización del progreso, que incluye el trabajo por hacer, el trabajo en curso y el trabajo terminado, se representa en un tablero físico o electrónico.
- La práctica de limitar el trabajo en curso significa esencialmente que antes de que comience una nueva etapa del proceso, todas las tareas pendientes se han realizado.
- La práctica de la mejora del flujo de trabajo tiene por objeto maximizar el valor de la entrega al cliente, garantizar la fluidez del proceso y minimizar el coste y el tiempo del proceso.

El método Kanban no especifica los roles distintos del director de proyecto convencional y del equipo de desarrollo [38].

## **Scrum**

Scrum es el marco ágil más popular utilizado en la industria. En particular, es el único marco ágil incluido en la categoría de habilidades del sitio de búsqueda de empleo más popular de alcance mundial. Esta popularidad es una función de la capacidad de este marco para impulsar la eficiencia del proceso de desarrollo de software, incluyendo la velocidad de entrega y la calidad del propio software. Otro aspecto que ha hecho que este marco sea muy popular es que se presta a ser adoptado como forma de gestión del flujo de trabajo en otros ámbitos empresariales, es decir, su uso no se limita a la gestión de proyectos de software.

Scrum es un marco de trabajo sencillo y ligero que se utiliza para desarrollar productos complejos, Scrum también es un proceso empírico que se basa en tres pilares: transparencia, inspección y adaptación. Como otras metodologías ágiles, Scrum entrega el proyecto en iteraciones de intervalos fijos llamados sprints. Así, para maximizar la oportunidad, durante cada sprint, el equipo de desarrollo necesita visitar empíricamente los tres pilares. Esta metodología consta de tres roles de Scrum, cinco eventos de Scrum, tres artefactos de Scrum y las reglas que los unen [38].

## **Extreme Programming**

La Programación Extrema o Extreme Programming (XP) se introdujo por primera vez en 1996. En este enfoque se hace hincapié en la satisfacción del cliente, que se garantiza mediante la retroalimentación continua. Este método permite al equipo que desarrolla el software responder a los cambios en los requisitos del software, definidos por el cliente.

Aplicar XP implica la colaboración entre un pequeño equipo de desarrollo y el cliente. El equipo de desarrollo está formado por entre dos y diez miembros que trabajan en subcomponentes del software, como un problema a resolver o un nuevo requisito funcional. Los miembros del equipo se organizan de la forma más eficiente posible en torno a un problema, o a un requisito, que hay que abordar.

XP consiste en reglas simples que guían las etapas de implementación. Éstas consisten en planificar, gestionar, diseñar, codificar y probar. El proceso de diseño de XP está simplificado y utiliza una metáfora del sistema. La codificación se produce de forma

programada por pares, siguiendo estándares acordados, se prueba por unidades, se controla la versión y se integra continuamente.

Las pruebas se llevan a cabo para cada unidad de código, componente separado y componentes integrados, para garantizar la corrección de todos los errores. El XP es adecuado para proyectos de alto riesgo, de corta duración, con requisitos dinámicos y ejecutados por un equipo pequeño [38].

### **1.3 Objetivos**

#### **1.3.1 Objetivo general**

Desarrollar una Aplicación Web Progresiva con IoT para el monitoreo de mascotas en tiempo real en la ciudad de Quito.

#### **1.3.2 Objetivos específicos**

- Analizar la tecnología más accesible para el rastreo de mascotas en la ciudad de Quito (redes LPWAN).
- Desarrollar un prototipo del dispositivo que permita la conexión entre la mascota y la tecnología seleccionada.
- Estudiar las características del desarrollo de Aplicaciones Web Progresivas (PWA).
- Desarrollar una Aplicación Web Progresiva que permita obtener datos del dueño y de su mascota junto con la ubicación y temperatura para realizar el monitoreo en tiempo real de las mascotas en la ciudad de Quito.



## **CAPÍTULO II METODOLOGÍA**

### **2.1 Materiales**

En el presente proyecto de investigación se usaron fuentes confiables de información como lo son: sitios web, revistas científicas y repositorios de diferentes instituciones universitarias, además se aplicó el conocimiento académico adquirido en la institución de enseñanza superior, junto a cursos que ayudaron a reforzar estos conocimientos.

Para la recolección de información se aplicó la técnica de encuesta y como instrumento diseñado como material se usó el formulario de preguntas, dirigida a los dueños de mascotas en la ciudad de Quito los cuales son los principales beneficiarios de este proyecto de investigación

### **2.2 Métodos**

#### **2.2.1 Modalidad de investigación**

##### **Modalidad Aplicada**

La modalidad de investigación es aplicada porque se ocuparán los conocimientos adquiridos durante la formación académica y con ellos desarrollar un proyecto de investigación eficiente.

##### **Modalidad bibliográfica**

La modalidad de investigación es bibliográfica porque el desarrollo del proyecto de investigación estará apoyado por documentos con información técnica verificada.

##### **Modalidad de campo**

La modalidad de investigación es de campo porque gracias a técnicas de recolección de información se obtendrán datos de gran ayuda para el desarrollo de del proyecto de investigación.

#### **2.2.2 Población y muestra**

La ciudad de Quito actualmente está conformada por 2'781.641 habitantes según las proyecciones de población ecuatoriana por cantones que determinó el INEC para el año 2020, por lo que para este proyecto de investigación se tomará ese número de habitantes como referencia, ya que el tema tiene como alcance la ciudad de Quito.

Nombre Cantón	2018	2019	2020
Quito	2'690.150	2'735.987	2'781.641

**Tabla 2.1 Proyección del número de habitantes de la ciudad de Quito hasta el 2020**  
Elaborado por: El investigador

Para el cálculo de la muestra se usará la siguiente formula:

$n = \frac{N\sigma^2 Z^2}{(N-1)e^2 + \sigma^2 Z^2}$	
n	Tamaño de muestra
N	Población
Z	Nivel de confianza 95% equivalente a 1,96
$\sigma$	Desviación estándar constante 0,5
e	margen de error: 5%
Muestra	385

**Tabla 2.2 Calculo de la muestra de habitantes de la ciudad de Quito**  
Elaborado por: El investigador

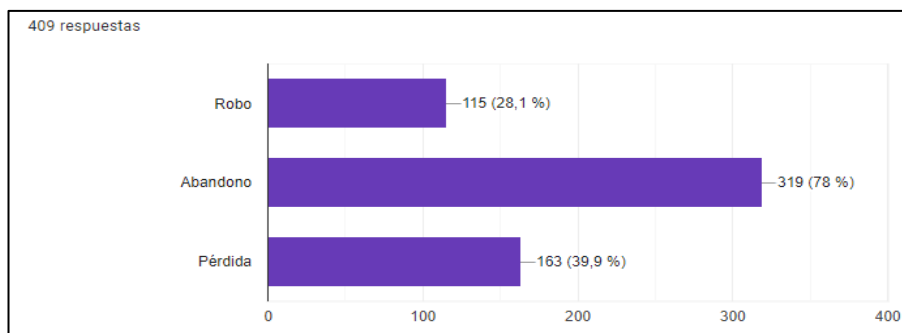
Para la selección de la muestra de habitantes se utilizó el método probabilístico aleatorio simple, es decir escogiendo de forma aleatoria tantos usuarios como requiera completar la muestra.

### 2.2.3 Recolección de información

Se aplicó una encuesta a los dueños de mascotas en la ciudad de Quito enfocada en el monitoreo en tiempo real de sus mascotas. Obteniendo 409 respuestas, donde la información proporcionada a través de la encuesta es de gran ayuda para dar validez al desarrollo del software y hardware del proyecto de investigación.

Los resultados obtenidos son los siguientes:

**Pregunta 1 ¿Cuáles considera usted que son las razones por las que existen animales callejeros?**

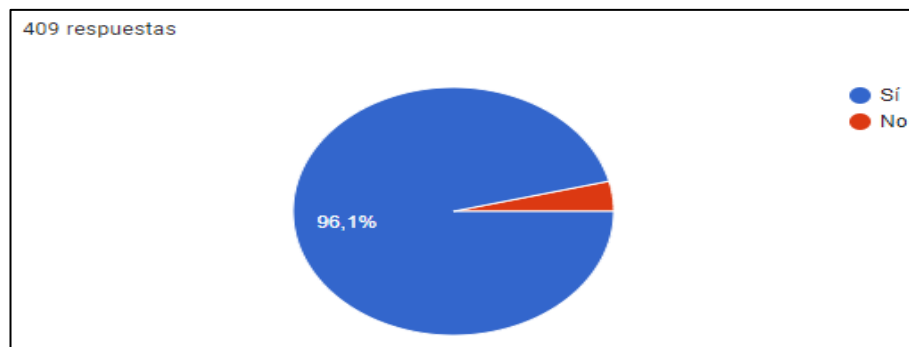


**Figura 2.1 Gráfico Pregunta 1 Encuesta Monitoreo de Mascotas Ciudad de Quito**  
Elaborado por: El investigador

### **Análisis e interpretación de resultados:**

De acuerdo con los resultados de la figura 2.1 se observa que el 78% de los dueños de mascotas encuestados consideran que una razón por la cual existen animales callejeros es por el abandono, otra razón que considera el 39,9% de encuestados es la pérdida de los animales y el 28,1% de los encuestados consideran que una razón es el robo de los animales. Se puede identificar que los encuestados consideran como la segunda razón más probable de la existencia de animales callejeros su pérdida convirtiendo esta razón en un problema a solucionar.

### **Pregunta 2 ¿Considera usted que la seguridad de una mascota es importante?**

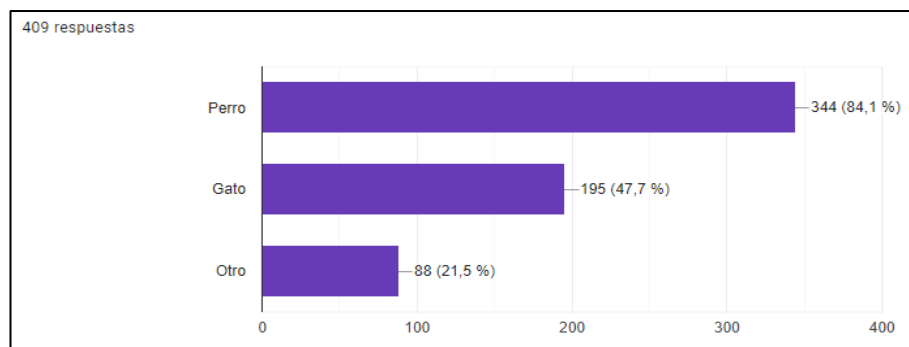


**Figura 2.2 Gráfico Pregunta 2 Encuesta Monitoreo de Mascotas Ciudad de Quito**  
Elaborado por: El investigador

### **Análisis e interpretación de resultados:**

De acuerdo con los resultados de la figura 2.2 se observa que el 96,1% de los dueños de mascotas encuestados consideran que la seguridad de sus mascotas es importante y el 3,9% consideran que no lo es. Se puede identificar que para la mayoría encuestados la seguridad de sus mascotas es una prioridad.

### **Pregunta 3 ¿De qué especie de mascota usted ha sido dueño?**

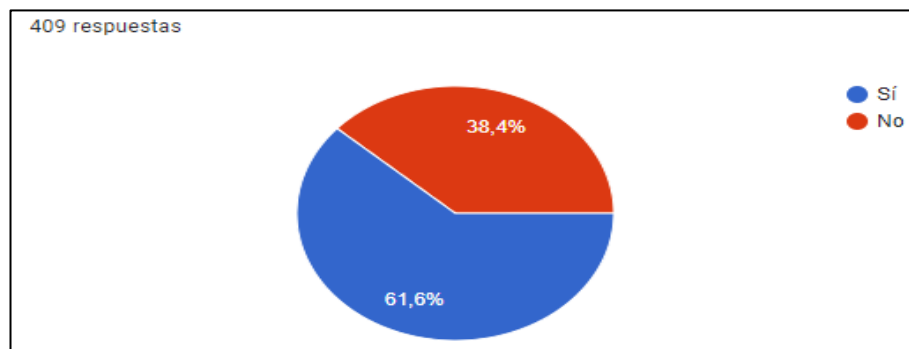


**Figura 2.3 Gráfico Pregunta 3 Encuesta Monitoreo de Mascotas Ciudad de Quito**  
Elaborado por: El investigador

### **Análisis e interpretación de resultados:**

De acuerdo con los resultados de la figura 2.3 se observa que el 84% de los encuestados han sido dueños de perros mientras que el 47,7% de los encuestados han sido dueños de gatos y el 21,5% han sido dueños de otra especie animal. Se puede identificar que las personas eligen ser dueños de perros en primer lugar y en segundo lugar ser dueños de gatos.

### **Pregunta 4 ¿Ha perdido alguna vez una mascota?**

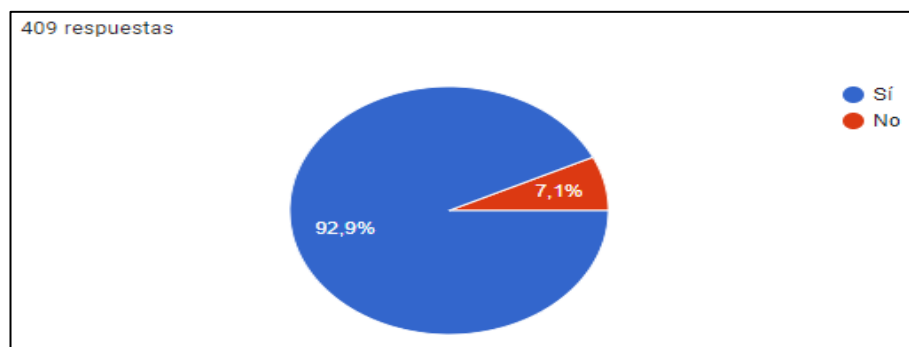


**Figura 2.4 Gráfico Pregunta 4 Encuesta Monitoreo de Mascotas Ciudad de Quito**  
Elaborado por: El investigador

### **Análisis e interpretación de resultados:**

De acuerdo con los resultados de la figura 2.4 se observa que el 61,6% de las personas encuestadas alguna vez a perdido una mascota y el 38,4% no ha perdido una mascota. Se puede identificar que la mayoría de las personas encuestadas han perdido una mascota a lo largo de su vida.

### **Pregunta 5 ¿Le gustaría saber la ubicación y temperatura de su mascota en tiempo real?**

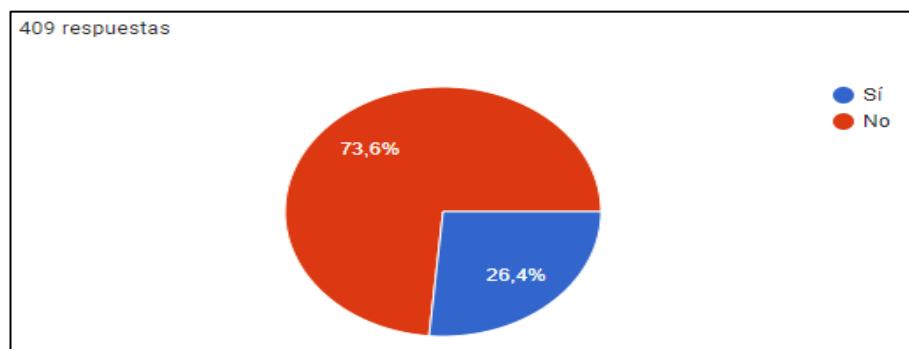


**Figura 2.5 Gráfico Pregunta 5 Encuesta Monitoreo de Mascotas Ciudad de Quito**  
Elaborado por: El investigador

### **Análisis e interpretación de resultados:**

De acuerdo con los resultados de la figura 2.5 se observa que al 92,9% de los dueños de mascotas les gustaría saber la ubicación y la temperatura que sus mascotas tienen, mientras que el 7,1% no está interesado. Se puede identificar que a la mayoría de encuestados les interesa la ubicación de sus mascotas junto a la temperatura que estas tienen en tiempo real.

### **Pregunta 6 ¿Lleva usted el registro de los datos de su mascota en alguna plataforma tecnológica?**

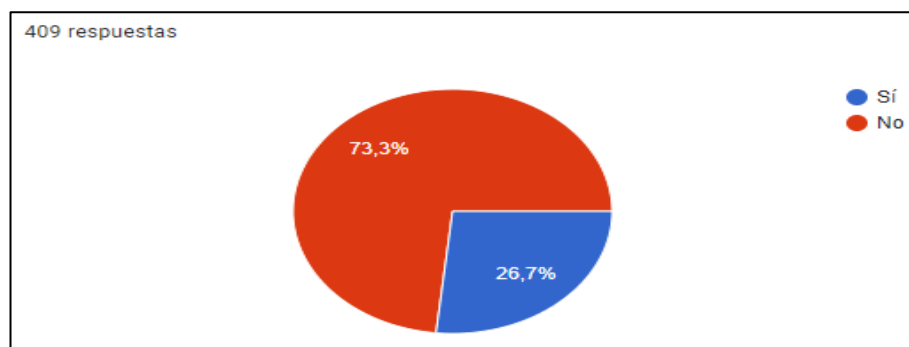


**Figura 2.6 Gráfico Pregunta 6 Encuesta Monitoreo de Mascotas Ciudad de Quito**  
Elaborado por: El investigador

### **Análisis e interpretación de resultados:**

De acuerdo con los resultados de la figura 2.6 se observa que el 73,6% de dueños de mascotas no usan una plataforma tecnológica para llevar el registro de los datos de su mascota, mientras que el 26,4% de los encuestados si llevan un registro de los datos de sus mascotas. Se puede identificar que la mayoría de encuestados no usan plataformas tecnológicas para llevar el registro de los datos de sus mascotas.

### **Pregunta 7 ¿Tendría algún problema usted con que su mascota use un pequeño dispositivo en su cuello para su monitoreo en tiempo real?**

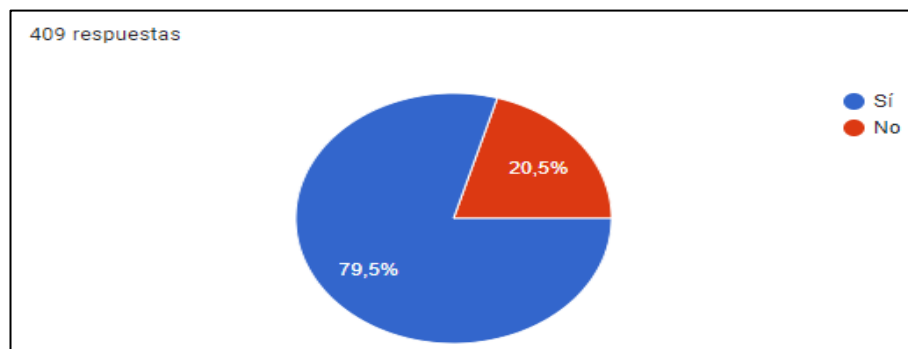


**Figura 2.7 Gráfico Pregunta 7 Encuesta Monitoreo de Mascotas Ciudad de Quito**  
Elaborado por: El investigador

**Análisis e interpretación de resultados:**

De acuerdo con los resultados de la figura 2.7 se observa que el 73,3% de los dueños de mascotas encuestados no tienen problema con que su mascota use un dispositivo en su cuello para su monitoreo en tiempo real y el 26,7% si tiene un inconveniente con que este dispositivo este en su cuello. Se puede identificar que la mayoría de encuestados no tienen inconveniente con que su mascota pueda usar un pequeño dispositivo en el cuello.

**Pregunta 8 ¿Estaría usted interesado en una plataforma de monitoreo para su mascota en tiempo real de pago anual?**

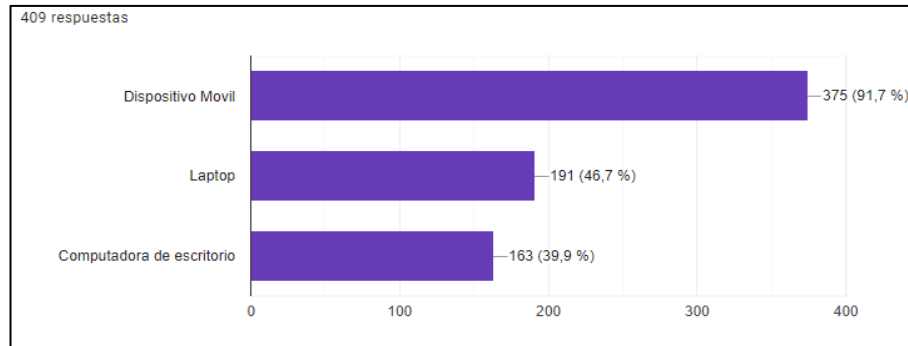


**Figura 2.8 Gráfico Pregunta 8 Encuesta Monitoreo de Mascotas Ciudad de Quito**  
Elaborado por: El investigador

**Análisis e interpretación de resultados:**

De acuerdo con los resultados de la figura 2.8 se observa que el 79,5% de los dueños de mascotas encuestados están interesados en una plataforma de monitoreo de sus mascotas de pago anual, mientras que el 20,5% de los encuestados no están interesados. Se puede identificar que la mayoría de los dueños de mascotas están interesados en una plataforma para el monitoreo de sus mascotas de pago anual.

**Pregunta 9 ¿Cuál es el equipo tecnológico con el que usted accede a internet frecuentemente?**



**Figura 2.9 Gráfico Pregunta 9 Encuesta Monitoreo de Mascotas Ciudad de Quito**  
Elaborado por: El investigador

### **Análisis e interpretación de resultados:**

De acuerdo con los resultados de la figura 2.9 se observa que el 91,7% de los dueños de mascotas encuestados usan un dispositivo móvil para acceder a internet, mientras que el 46,7% de los encuestados acceden al internet a través de una laptop y el 39,9% de los encuestados usan computadora de escritorio para acceder al internet. Se puede identificar que la mayoría de encuestados usan un dispositivo móvil para acceder al internet mientras que una parte de los encuestados también usan computadoras para acceder al internet.

#### **2.2.4 Procesamiento y análisis de datos**

De acuerdo con la encuesta aplicada a los dueños de mascotas en la ciudad de Quito se puede concluir que:

- Una de las razones por las que consideran los dueños de mascotas que existen animales callejeros, además de su abandono es por su pérdida, por lo que es un factor importante del aumento de animales callejeros en la ciudad de Quito.
- La mayoría de los dueños están interesados en la seguridad de sus mascotas, aunque hayan perdido una de ellas a lo largo de su vida.
- Para los dueños de mascotas una forma de poder cuidarlas es conociendo la ubicación y temperatura de sus mascotas en tiempo real.
- Dado que la mayoría de los dueños de mascotas no lleva el registro de los datos de sus mascotas en una plataforma tecnológica, es importante el desarrollo de una plataforma tecnológica la cual sirva para que los dueños de mascotas tengan una opción para realizar esta tarea.

- La mayoría de los dueños encuestados usan un dispositivo móvil y otra gran parte usan computadora ya sea laptop o de escritorio la plataforma tecnológica debería ser multiplataforma, por lo que el usar la tecnología PWA sería la mejor opción.
- Un gran número de dueños está interesado en una plataforma de pago anual, por lo que es posible incorporar nuevas tecnologías de costo a la plataforma tecnológica a desarrollarse.
- Los dueños no tienen problema con que sus mascotas usen un pequeño dispositivo en sus cuellos.
- La mayoría de encuestados son dueños de perros y gatos por lo que es factible desarrollar un pequeño dispositivo el cual pueda ser usado por estas dos especies.



## CAPÍTULO III RESULTADOS Y DISCUSIÓN

### 3.1 Análisis y discusión de resultados

#### 3.1.1 Análisis de la tecnología más accesible para el rastreo de mascotas en la ciudad de Quito

Una ciudad inteligente es una ciudad innovadora que utiliza las tecnologías de la información y otros medios para mejorar la calidad de vida, mediante la eficiencia de las operaciones, los servicios urbanos y la competitividad, a la vez que satisface las necesidades de las generaciones presentes y futuras con respecto a los aspectos económicos, sociales y ambientales [39].

El Municipio del Distrito Metropolitano de Quito, con la iniciativa de convertir a la ciudad de Quito en una ciudad inteligente, inicio el proyecto “Orquestador de la Movilidad”, el cual, con la ayuda de la Inteligencia Artificial, el Big Data y el Machine Learning, integra por medio de una plataforma digital todos los elementos claves de la movilidad, enfocándose en la seguridad vial, la movilidad inteligente y la sostenibilidad [40].

Dentro del marco de una ciudad inteligente las tecnologías IoT han aumentado en popularidad gracias a sus características de bajo consumo energético y una comunicación de larga duración [40].

Para aprovechar las características de la tecnología IoT se han desarrollado Redes de Área Amplia de Baja Potencia o LPWANs, las cuales su uso más frecuente es en el contexto de la agricultura, ganadería, medicina, industria, dispositivos personales, dispositivos para ciudades inteligentes, rastreo o seguimiento, y telemetría [41].

Las 3 tecnologías más populares de LPWAN actualmente son NB-IoT, Sigfox y LoRaWan, por lo que para el desarrollo del proyecto de investigación se realizará un cuadro comparativo para determinar la más accesible de ellas.

<b>Características</b>	<b>LoraWan</b>	<b>Nb-IoT</b>	<b>Sigfox</b>
<b>Frecuencias</b>	No licenciadas	Licenciadas	No licenciadas
<b>Distancia</b>	20 km (Rural) 5 km (Urbano)	10 km (Rural) 1 km (Urbano)	50 km (Rural) 10 km (Urbano)
<b>Implementación</b>	Compleja	Compleja	Sencilla
<b>Mensajes al día</b>	Ilimitado	Ilimitado	140
<b>Carga útil</b>	243 bytes	62,5 bytes	12 bytes

<b>Proveedores</b>	Varios	Varios	Único
<b>Costo modulo</b>	3-5 \$	> 20 \$	< 2\$
<b>Costo despliegue</b>	> 100 \$ Gateway / >10000 \$ Estación base	> 15000 \$ Estación base	> 40000 \$ Estación base

**Tabla 3.1 Comparativa de las 3 tecnologías LPWAN más populares**  
Elaborado por: El investigador

De acuerdo con la tabla 3.1 para el proyecto de investigación se ha determinado que la tecnología más accesible para implementar un dispositivo de monitoreo de mascotas es la tecnología Sigfox, por las siguientes razones:

- La distancia urbana y rural de cobertura Sigfox: Actualmente en la ciudad de Quito existe ya el despliegue de antenas que proporcionan una señal que abarca gran parte de su territorio de la ciudad.
- La implementación: La tecnología Sigfox es sencilla de usar, ya que funciona sin necesidad de elementos extra, lo único que se necesita es un nodo con posibilidad de conectarse a la red de Sigfox.
- El costo: Su costo es relativamente bajo, ya que cuenta con un método de pago anual por el uso de la red que proporciona Sigfox y además los dispositivos con tecnología Sigfox son baratos y accesibles.
- La carga útil: Los 12 bytes posibles para usarse es ideal para los datos que se usaran para el desarrollo de la PWA de monitoreo de mascotas.

### 3.1.2 Determinación de modulo Arduino Sigfox

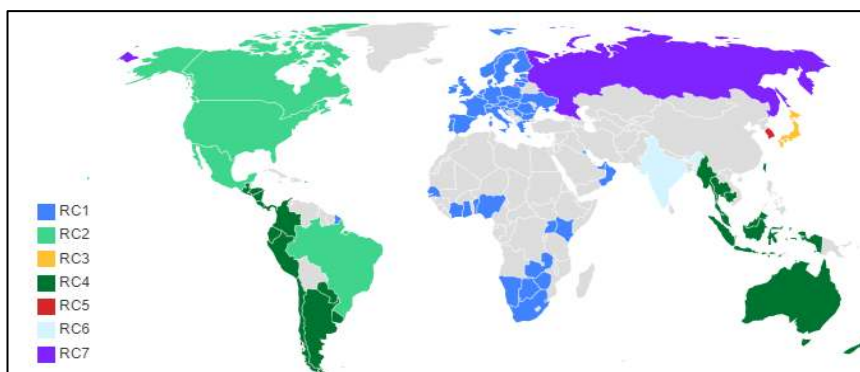
Varias compañías han desarrollado módulos con tecnología Sigfox, los cuales dan la posibilidad a que el desarrollador pueda configurarlos para los proyectos en los que esté trabajando.

Dado que Sigfox opera en bandas sin licencia en todo el mundo, con frecuencias de radio que van desde 862 a 928 MHz, algunas de estas no están disponibles en algunas partes del mundo, por lo que Sigfox ha catalogado las frecuencias en diferentes zonas, las cuales son:

Zonas	Países
<b>RC1</b>	<ul style="list-style-type: none"> <li>• Europa: Unión Europea, Andorra, Liechtenstein, Noruega, Serbia, Suiza, Ucrania, Reino Unido.</li> </ul>

	<ul style="list-style-type: none"> <li>• Francia de ultramar: Guayana Francesa, Polinesia Francesa, Guadalupe, Martinica, Mayotte, Nueva Caledonia, Reunión.</li> <li>• Medio Oriente y África: Botswana, Costa de Marfil, Kenia, Mauricio, Namibia, Nigeria, Omán, Senegal, Sudáfrica, Suazilandia, Emiratos Árabes Unidos.</li> </ul>
<b>RC2</b>	<ul style="list-style-type: none"> <li>• Brasil, Canadá, México, Puerto Rico, Estados Unidos.</li> </ul>
<b>RC3</b>	<ul style="list-style-type: none"> <li>• Japón.</li> </ul>
<b>RC4</b>	<ul style="list-style-type: none"> <li>• Latinoamérica: Argentina, Chile, Colombia, Costa Rica, Ecuador, El Salvador, Guatemala, Honduras, Nicaragua, Panamá, Paraguay, Perú, Trinidad &amp; Tobago, Uruguay.</li> <li>• Asia Pacífico: Australia, Hong Kong, Indonesia, Malasia, Nueva Zelanda, Singapur, Taiwán, Tailandia.</li> </ul>
<b>RC5</b>	<ul style="list-style-type: none"> <li>• Corea del Sur.</li> </ul>
<b>RC6</b>	<ul style="list-style-type: none"> <li>• India.</li> </ul>
<b>RC7</b>	<ul style="list-style-type: none"> <li>• Rusia.</li> </ul>

**Tabla 3.2 Catálogo de zonas Sigfox**  
Elaborado por: El investigador



**Figura 3.1 Grafico de las zonas Sigfox**  
Elaborado por: [42]

Al analizar la tabla 3.2 y la figura 3.1 se concluyó que la zona a usarse es la RC4, ya que en esta zona está incluido el país de Ecuador.

### 3.1.2.1 Ufox

Ufox es un kit de desarrollo para IoT fabricado por la empresa peruana Teca que permite a los estudiantes y profesionales crear soluciones IoT en base a hardware y conectividad inalámbrica.

Ufox posee conexión USB, esto facilita el proceso de prototipado y desarrollo de una solución IoT. Ufox se conecta a internet mediante la red de Sigfox lo que habilita el funcionamiento de dispositivos, los cuales pueden transmitir datos por muchos años [43].

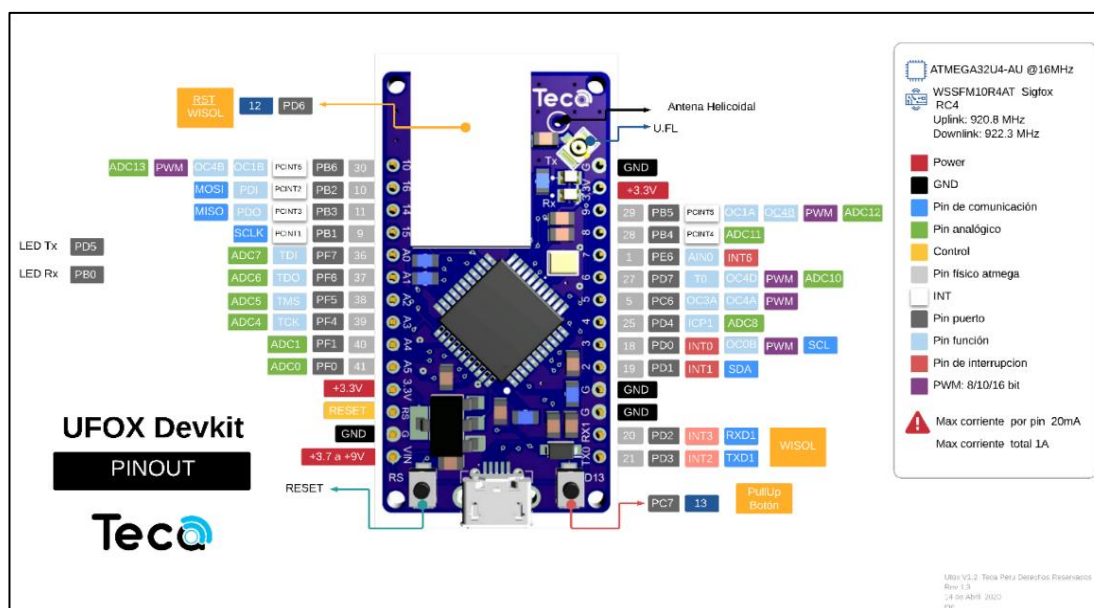


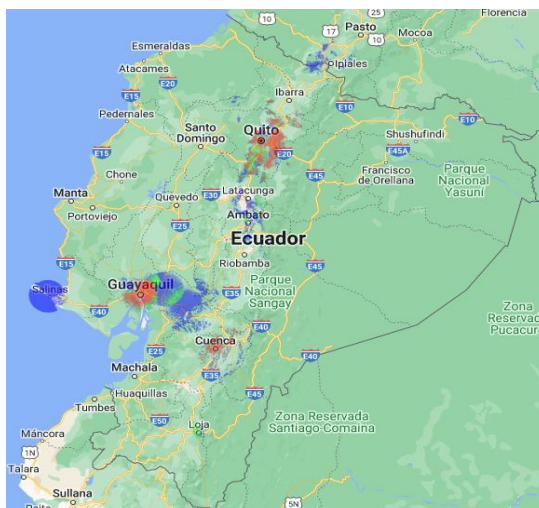
Figura 3.2 Descripción de pines Ufox  
Elaborado por: [44]

En la figura 3.2 se muestra que la placa Ufox es compatible con los pines de un Arduino Leonardo, por lo que es posible incorporar módulos adicionales para ser usados y mediante la programación de la placa se puede aprovechar los datos que estos módulos entregan.

Descripción	Unidad
<b>Región de funcionamiento</b>	RC4
<b>Frecuencia enlace ascendente</b>	920,8 MHz
<b>Frecuencia enlace descendente</b>	922,3 MHz
<b>Voltaje de entrada USB (Conector micro USB)</b>	5 voltios
<b>Voltaje mínimo entrada (Pin VIN)</b>	3.7 voltios
<b>Voltaje máximo de entrada (Pin VIN)</b>	9 voltios
<b>Voltaje máximo absoluto entrada (Pin 3.3V)</b>	4 voltios
<b>Voltaje mínimo absoluto entrada (Pin 3.3V)</b>	2.7 voltios
<b>Corriente máxima</b>	1 A
<b>Voltaje de funcionamiento interno típico</b>	2.7 voltios

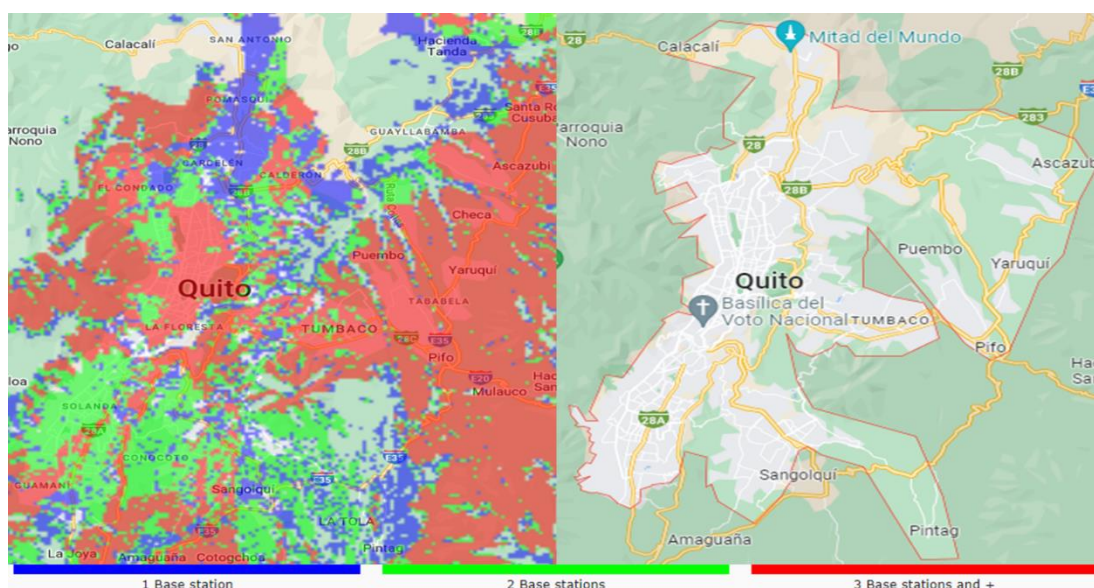
Tabla 3.3 Características de Ufox  
Elaborado por: El investigador

En la tabla 3.3 se especifica que la zona de funcionamiento de la placa Ufox es en la RC4, es decir, funciona en el territorio del país de Ecuador.



**Figura 3.3 Mapa de cobertura red Sigfox en Ecuador**  
Elaborado por: [45]

En la figura 3.3 se muestran las ciudades donde existe cobertura de la red Sigfox y una de las ciudades con mayor cobertura es la ciudad de Quito.



**Figura 3.4 Mapa de cobertura red Sigfox en la ciudad de Quito**  
Elaborado por: El investigador

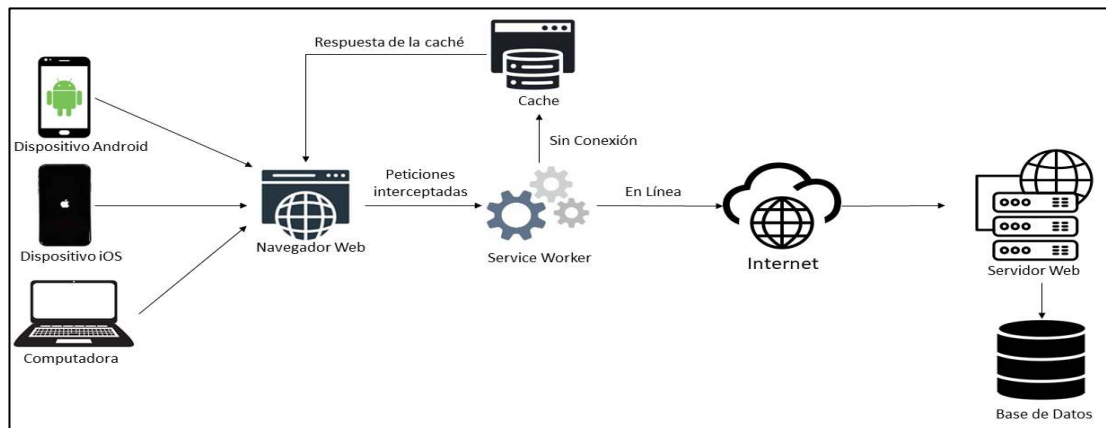
En la figura 3.4 se puede observar que la cobertura de la red Sigfox cubre la mayoría de la ciudad de Quito, gracias a que existen varias estaciones base que entregan señal Sigfox a los sectores de la ciudad, por lo que el uso de la placa Ufox es ideal para el desarrollo del dispositivo de monitoreo en tiempo real de mascotas.

### 3.1.3 Estudio de características de una Aplicación Web Progresiva

#### 3.1.3.1 Estructura de una PWA

Las PWA son desarrolladas utilizando tecnologías web modernas, proporcionando acceso a las Interfaces de Programación de Aplicaciones (API) de la plataforma desde un enfoque nativo combinado con habilidades específicas de la web, como permitir al usuario probar la aplicación sin requisitos de instalación [46].

Las PWA se ejecutan como las aplicaciones web en un navegador, pero con la capacidad de ocultar las herramientas típicas de este, como la barra de direcciones y los menús, haciendo que la PWA tenga un aspecto más parecido a una aplicación nativa [46].



**Figura 3.5 Arquitectura de una PWA**  
Elaborado por: El Investigador

En la figura 3.5 se muestra la arquitectura de la PWA, donde diferentes dispositivos pueden acceder a un navegador web y este mediante un service worker intercepta las peticiones y las distribuye dependiendo del estado de la conexión, ya sea a internet o a la cache del navegador para obtener respuesta de su petición.

#### 3.1.3.2 Ventajas, desventajas y Características principales de una PWA

##### Ventajas

- Son multiplataforma, por lo que son accesibles desde cualquier navegador y sistema operativo.
- Adaptan sus funcionalidades al navegador web utilizado.
- Permiten informar a los usuarios mediante notificaciones push.

- Su descarga no es necesaria.
- Se pueden utilizar sin conexión a Internet.
- Su velocidad de carga es similar al de las aplicaciones nativas y superior al de las aplicaciones web.
- Su navegación es sencilla, intuitiva e inmersiva, similar a la de las aplicaciones nativas.
- El usuario puede compartir y almacenar datos en la propia aplicación.
- Las actualizaciones se realizan automáticamente, sin molestias para los usuarios.
- Su diseño responsive se adapta a cualquier dispositivo.
- No se requiere desarrollar una aplicación para cada sistema operativo.
- Utilizan un protocolo seguro (HTTPS).
- Se puede guardar un acceso directo en la pantalla de inicio, como en las aplicaciones nativas.
- No es necesaria una tienda de aplicaciones.
- Se pueden compartir como enlace mediante un programa de mensajería instantánea.
- Funcionan en segundo plano.

### **Desventajas**

- No pueden acceder a todas las funcionalidades de los dispositivos.
- No todos los navegadores web soportan las PWA.
- Los usuarios de una PWA no pueden decidir cuándo actualizarla.

### **Características**

Las principales características de una PWA que fueron tomadas en consideración para el desarrollo del proyecto de investigación son las siguientes:

- **Universalidad:** Funciona sin problemas para cada usuario, independientemente de su navegador web.
- **Responsiva:** Debe funcionar con cualquier dispositivo, ordenador portátil, tablet o teléfono inteligente
- **Diseño:** Debe imitar las aplicaciones nativas, con una interactividad simple para funciones avanzadas.
- **Seguridad:** Las PWA deben usar siempre HTTPS para mantener seguros los datos del usuario.
- **Actualizaciones:** Se mantienen actualizados y ofrecen las últimas versiones de un servicio o sitio.
- **Fuera de línea:** Las PWA deben funcionar en gran medida incluso si el dispositivo está fuera de línea (modo avión o fuera de la cobertura de una red de internet)
- **Instalación:** Su instalación es posible realizarla desde cualquier navegador que soporte la tecnología PWA, sin pasos adicionales y sin necesidad de descargas desde tiendas de aplicaciones.
- **Compartir:** Las PWA solo requiere una única URL para ser compartidas [47].

### **3.1.3.3 Comparación entre enfoques de desarrollo Nativo, Híbrido y PWA**

#### **Nativo**

Son aplicaciones desarrolladas con herramientas y lenguajes de programación dedicados a una determinada plataforma, por lo que los programadores deben ajustarse a los lenguajes y herramientas específicos necesarios para desarrollar con éxito una aplicación.

#### **Híbrido**

Las aplicaciones desarrolladas con el enfoque híbrido utilizan el motor de un navegador web en el dispositivo móvil e incrustan el contenido HTML en un contenedor web nativo, como Cordova, Ionic, PhoneGap, MoSync, estos contenedores proporcionan un envoltorio nativo que contiene los códigos basados en



la web y también una API genérica de JavaScript que sirve como puente de la solicitud de servicio desde el código basado en la web hasta la API de la plataforma correspondiente.

## **PWA**

Las PWA son un tipo especial de aplicaciones web que se basan en el concepto de una sola aplicación para todas las plataformas. Estas aplicaciones no requieren instalación antes de usar y se sirven desde un servidor remoto a través de un Protocolo de transferencia de hipertexto (HTTPS) seguro.

<b>Características</b>	<b>Nativo</b>	<b>Hibrido</b>	<b>PWA</b>
<b>Instalable</b>	Sí	Sí	Sí
<b>Capacidad fuera de línea</b>	Limitado	Limitado	Sí
<b>Comprobable antes de la instalación</b>	No	Sí	Sí
<b>Disponibilidad en el mercado de aplicaciones</b>	Sí	Sí	Sí
<b>Disponibilidad multiplataforma</b>	No	Sí	Sí
<b>Acceso al hardware y la plataforma</b>	Sí	Sí	Limitado
<b>Sincronización en segundo plano</b>	Sí	Sí	Sí
<b>Capa de seguridad</b>	No	No	Sí
<b>Capacidad de enlace</b>	No	No	Sí
<b>Actualizado constantemente</b>	No	No	Sí
<b>Capacidad de escritorio</b>	No	No	Sí

**Tabla 3.4 Cuadro de comparación de enfoques de desarrollo Nativo, Hibrido y PWA**  
Elaborado por: [48]

En la tabla 3.4 se comparan los enfoques de desarrollo Nativo, Hibrido y PWA, donde la PWA supera en características a los otros enfoques, convirtiéndola en una solución viable para desarrollar proyectos accesibles, seguros y fáciles de mantener.

### **3.1.3.4 Tecnologías para el desarrollo de aplicaciones web progresivas**

Las tecnologías utilizadas para la construcción de PWA son las siguientes:

- Lenguaje de Marcado de Hipertexto Versión 5 (HTML5), Hojas de Estilo en Cascada (CSS) y JavaScript se utilizan para crear la interfaz de usuario, la cual es la parte estática de PWA. Esta interfaz de usuario, que se conoce como el shell de la aplicación, el cual se almacena en el caché del dispositivo móvil del usuario evitando así retrasos al cargar y errores que puedan suceder debido a una mala conexión de red.

- El lado del cliente utiliza llamadas a servicios web para cargar y representar el contenido tal como se hace en las aplicaciones nativas.
- El service worker es un código JavaScript que el navegador ejecuta en segundo plano.
- Manifest File, su propósito es exponer ciertas configuraciones modificables a los desarrolladores de aplicaciones, donde al proporcionar un extracto de datos en formato JSON, es posible almacenar en caché esta información con la ayuda del service worker y luego usar app shell para cargar reglas CSS. El service worker almacena en caché el extracto JSON y físicamente en la shell de la aplicación. Este shell de aplicación es un contenedor de autocontenido que tiene todas las hojas de estilo, scripts, imágenes, fuentes y salidas HTML necesarias para que un usuario presente una página completa [23].
- Los frameworks como Angular o librerías como React se pueden utilizar para desarrollar PWA. Estos frameworks tienen algunos service worker integrados que hacen innecesario reconstruirlos para cada PWA [49].

A continuación, se detalla una tabla comparativa de tecnologías de desarrollo web, en los que es posible desarrollar PWAs, para determinar el mejor a ser usado en el proyecto de investigación.

<b>Categoría</b>	<b>Angular</b>	<b>React</b>	<b>Vue</b>
<b>Tipo</b>	Framework	Biblioteca	Framework
<b>Lenguaje</b>	TypeScript	JavaScript	JavaScript
<b>Flexibilidad</b>	Baja	Alta	Alta
<b>Curva de aprendizaje</b>	Difícil	Moderada	Baja
<b>Herramientas de terceros</b>	No	Si	Si
<b>Documentación</b>	Detallada	Escasa	Detallada
<b>Tiempo de arranque</b>	Lento	Rápido	Rápido

**Tabla 3.5 Cuadro comparativo de tecnologías para el desarrollo de PWAs**

**Elaborado por: El investigador**

Como resultado del análisis de la tabla 3.5 se determinó que la tecnología a usarse para el desarrollo de la PWA para el monitoreo de mascotas se lo realizará con la librería React ya que esta cuenta con una curva de aprendizaje moderada, además su flexibilidad al ser alta permitirá que al desarrollar la PWA en esta sea posible incorporar herramientas de terceros y así facilitar la codificación de la aplicación.

### 3.1.4 Determinación de la tecnología frond-end

React al ser una librería tiene un problema, el cual es no tener un marco de trabajo, esto con el tiempo se convierte en problema grave, ya que para solucionarlo se necesita la instalación de más librerías de terceros que tienen su propia manera de trabajar.

Para solucionar el problema de no tener un marco de trabajo en React, la empresa Vercel desarrollo la tecnología Next.js, la cual utiliza la librería React como base para desarrollar aplicaciones web.

Next.js para el desarrollo del proyecto de investigación será de mucha ayuda, ya que este al tener un marco de trabajo definido incorpora herramientas muy importantes a ser usadas, una de ellas es la posibilidad de realizar aplicaciones web progresivas con simples modificantes a la configuración inicial que Next.js proporciona al desarrollador.

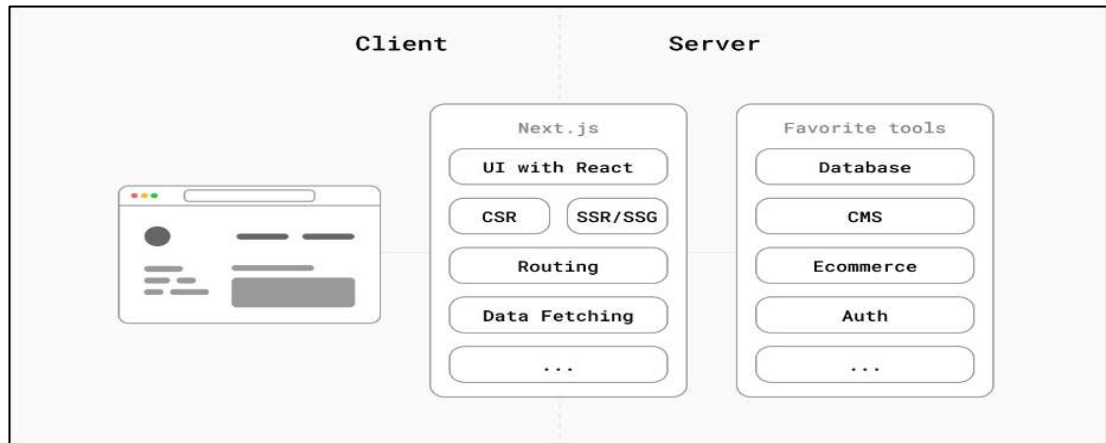
El enrutamiento basado en carpetas y archivos que usa Next.js es una ventaja importante, ya que con solo crear carpetas y organizarlas se creara un enrutamiento optimo, con posibilidad de desarrollar enrutamientos dinámicos.

Next.js tiene la capacidad para Generar Sitios Estáticos o Static Site Generation (SSG),es decir que el HTML se genera antes de cada petición, como al momento de crear el build de producción, también es posible usar la Representación del Lado del Servidor o Server-side Rendering (SSR), donde el HTML se genera en el servidor durante cada petición, estas dos posibilidades de pre-renderizado hacen que Next.js al funcionar en producción tenga una velocidad muy rápida para su carga en los navegadores, además es importante mencionar que es posible usar la combinación de ambos según la necesidad de cada aplicación web.

Algunas ventajas adicionales de usar Next.js son las siguientes:

- Los datos se pueden recuperar desde cualquier lugar. El estado, los ciclos de vida de los hooks de React o la población de datos inicial se admiten mediante la exportación de un componente de función que usa hooks o componentes de clase.
- Transpilación automática y agrupación (con webpack y babel).

- Es posible usar cualquier solución CSS.
- Es posible implementar un servidor flexible con Express o cualquier otro servidor HTTP de Node.js.
- Admite el uso de librerías y herramientas de terceros.
- Es compatible con TypeScript.



**Figura 3.6 Elementos de Next.js**  
Elaborado por: [50]

En la figura 3.6 se detalla como Next.js distribuye los elementos que lo componen del lado servidor y de lado cliente.

### 3.1.5 Determinación de la tecnología back-end

El termino back-end se refiere al área lógica de toda aplicación web, la cual es la parte o rama del desarrollo web encargada de que toda la lógica de una página funcione de forma correcta.

El back-end no está visible a ojos del usuario y no incluye ningún tipo de elemento gráfico.

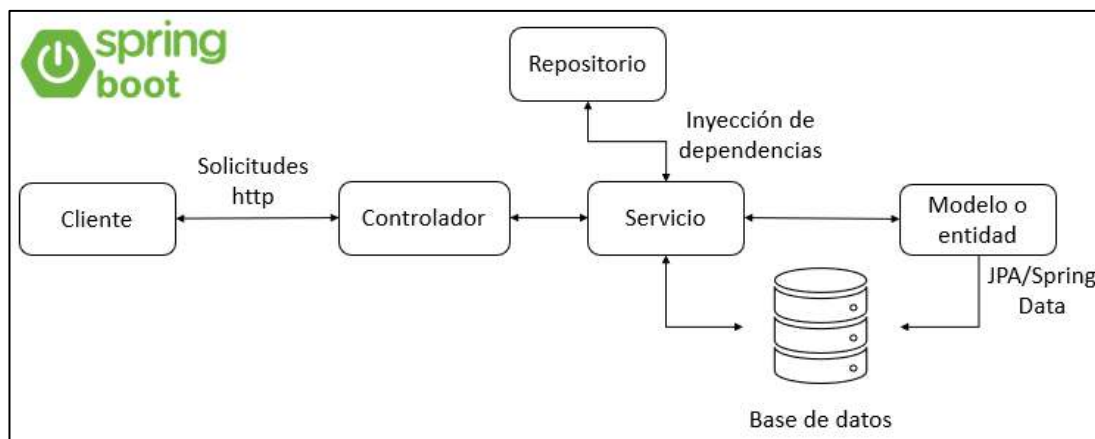
Las funciones que se gestionan en el back-end son las siguientes:

- Se llevan a cabo todas las funciones que hagan más simple el proceso de desarrollo.
- Las acciones de lógica.
- Las conexiones con las bases de datos.

- Se usan las librerías del servidor web, ya sea para comprimir las imágenes de la web, para implementar temas de caché u otros requerimientos.
- Se mantiene la seguridad de los sitios o aplicaciones web.
- Se pueden optimizar los recursos a fin de que las páginas resulten más ligeras.

Para el desarrollo del back-end encargado de proporcionar los servicios web tipo REST que serán llamados o consumidos por la PWA, se ha determinado a la tecnología Spring Boot como la herramienta a ser usada por las siguientes razones.

- Su gratuidad, ya que es una plataforma de lenguaje Java de código abierto
- Funciona sobre la Máquina Virtual de Java (JVM), por lo que la aplicación desarrollada se puede ejecutar en cualquier sistema operativo.
- Permite realizar microservicios, lo cual da la posibilidad para que la aplicación pueda separarse en varias tramas, esto permite realizar aplicaciones especializadas y fácilmente reutilizables dando mucha potencia y flexibilidad a los sistemas de información.
- Es rápido, ya que permite tener la estructura de la aplicación, la configuración de Spring framework y tener todas las dependencias en un formato legible en un archivo pom.xml, para su gestión en Maven de manera automática, por lo que desarrollar y desplegar una aplicación es prácticamente de inmediato.
- Cuenta con una comunidad grande de programadores que proporcionan soluciones a los inconvenientes que se producen al desarrollar aplicaciones.
- Da posibilidad de usar Java, Kotlin o Groovy para desarrollar nuevas aplicaciones, migrar o reutilizar código.



**Figura 3.7 Arquitectura de flujo de Spring Boot**  
Elaborado por: El investigador

La figura 3.7 describe como es el funcionamiento de Spring Boot, donde al desarrollar en código el modelo o entidad esta es usada para crear las respectivas tablas en la base de datos, el repositorio donde se extienden las operaciones CRUD de JPA Repository tiene comunicación con la capa de servicio y estas son capaces de realizar operaciones en la base de datos, además las funciones realizadas en la capa de servicio son usadas para desarrollar los servicios web tipo REST en el controlador de la aplicación y estos a través de solicitudes http son consumidos por la aplicación cliente.

### 3.1.6 Determinación de la metodología de desarrollo

Para determinar la metodología de desarrollo a utilizar a continuación se muestra un cuadro comparativo, en el cual se especificará las características de tres metodologías ágiles.

Características	Kanban	Scrum	Extreme Programming (XP)
Tamaño de proyecto	No importa el tamaño del proyecto	No importa el tamaño del proyecto	Proyectos pequeños y medianos
Tamaño de equipo	Tamaño del equipo irrelevante	Varios equipos de menos de 10 personas	Mínimo 3 y máximo 10 participantes
Iteraciones	Iteraciones regulares o bajo demanda	Sprint de 4 semanas	Iteraciones de 1 a 3 semanas
Estilo de desarrollo	Gradual y evolutivo	Iterativo y rápido	Iterativo y rápido
Coordinador de proyecto	Trabajo de equipo	Scrum Master	XP Coach

<b>Estándares de codificación</b>	No	No	Si
<b>Gestión de los requisitos</b>	Gestionado mediante el tablero Kanban	Requisitos Gestionados en forma de artefactos a través de Sprint Backlog y Product Backlog.	Gestionada en forma de tablas de historia de usuarios.

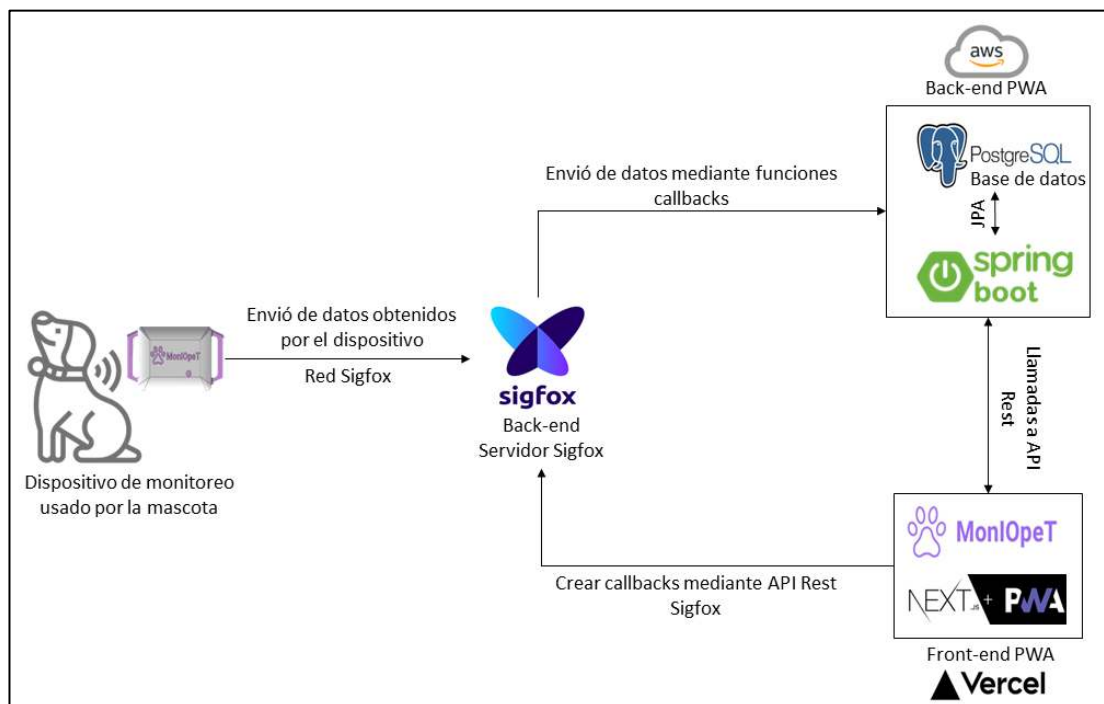
**Tabla 3.6 Comparación de metodologías ágiles**  
Elaborado por: El investigador

Gracias a la tabla 3.6 se pudo determinar que la mejor metodología para el desarrollo del proyecto de investigación es la metodología XP, ya que se enfoca en proyectos pequeños donde los integrantes que intervienen en el desarrollo no son numerosos. En el caso del presente proyecto de investigación serian tres: el XP coach, el programador y el cliente, donde las iteraciones se determinarán a través de las historias de usuarios que surgirán en la planificación del proyecto.

### 3.2 Desarrollo de la propuesta

#### 3.2.1 Funcionamiento de la PWA

Para identificar a la PWA que será desarrollada en el presente proyecto de investigación se asignó un nombre el cual es: MonIOpeT.



**Figura 3.8 Funcionamiento de la Aplicación web Progresiva con IoT usando tecnología Sigfox**  
Elaborado por: El investigador

En la figura 3.8 se detalla el funcionamiento de la PWA que se desarrolla en el presente proyecto de investigación, donde desde el dispositivo Ufox conformado por dos módulos Arduino, los cuales son: el módulo GPS NEO-6M y el módulo GY-906 sensor de temperatura infrarrojo MLX90614, se envían las lecturas obtenidas por estos y son alojadas en el servidor de Sigfox.

Mediante la PWA el usuario es capaz de asignar una función callback para crear el medio por el cual los datos que son enviados al back-end de Sigfox, se envíen automáticamente al back-end desarrollado en Spring Boot y con ello la PWA tenga la posibilidad de llamar los datos necesarios para realizar el monitoreo de la ubicación y la temperatura de su mascota.

### 3.2.2 Desarrollo de prototipo del dispositivo de monitoreo usando tecnología Sigfox

El prototipo del dispositivo de monitoreo para el presente proyecto es fundamental, ya que por medio de este los datos obtenidos de la mascota como lo son sus coordenadas y temperatura son los usados para realizar su monitoreo.

Los materiales usados para su desarrollo son los siguientes:

Material	Precio
Placa Ufox	45,00\$
Módulo GPS NEO-6M	10,00\$
Módulo GY-906 (Sensor de temperatura infrarrojo MLX90614)	16,00\$
Batería lipo 1200 mah	12,00\$
Cargador batería litio ultrafire 18650 Tp4056	2,00\$
Impresión del modelo 3D	20,00\$
Materiales extra	20,00\$
<b>Total</b>	<b>125,00\$</b>

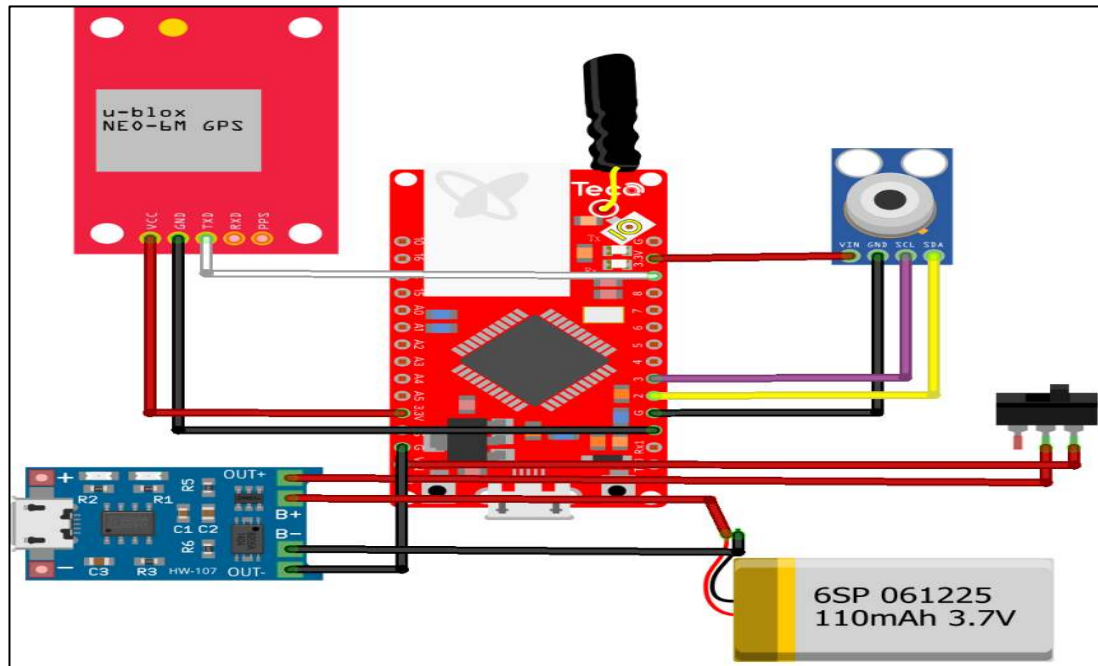
**Tabla 3.7 Costo total de materiales utilizados en el desarrollo del prototipo**  
Elaborado por: El investigador

La tabla 3.7 muestra que, la placa Ufox al estar basada en un Arduino Leonardo es posible incorporarle módulos adicionales, los cuales al ser accesibles en tiendas de electrónica facilitan el desarrollo de prototipos de dispositivos.



### 3.2.2.1 Esquema de conexiones

Un esquema de conexiones ayuda para comprender como esta realizado el circuito, que permite el funcionamiento de un dispositivo, por lo que a continuación se presenta la conexión usada para el prototipo del dispositivo de monitoreo.



**Figura 3.9 Esquema de conexiones del prototipo**  
Elaborado por: El investigador

La figura 3.9 detalla como la placa Ufox es el punto central de la conexión de los módulos, donde el pin TXD del módulo GPS NEO-6M es conectado al pin 9 de la placa Ufox y sus pines GND y VCC a los pines de energía 3.3V y GND de la placa, donde además los pines SCL y SDA del módulo GY-906 (Sensor de temperatura infrarrojo MLX90614) son conectados a los pines de la placa Ufox 3 y 2 respectivamente.

Para que el dispositivo tenga un medio de encendido y apagado se incorporó un cargador de baterías litio ultrafire 18650 Tp4056, el cual administra energía a la batería lipo de 1200 mah y esta al estar cargada suministrará energía al dispositivo controlado por un switch, conectando su parte positiva al pin VIN de la placa Ufox.

### 3.2.2.2 Código usado en el prototipo

El código usado para el desarrollo del dispositivo de monitoreo es el siguiente:

En la parte inicial del código se incluyen las librerías y variables que serán usadas en la sección de setup y loop

```
#include <LowPower.h>
#include <Ufox.h>
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include <Adafruit_MLX90614.h>

#define RXLED 17

Ufox wisol;
Adafruit_MLX90614 thermometer;
TinyGPS gps;
SoftwareSerial ss(9, 11);

float latitude, longitude;
long lastTx = -70000;
```

En la sección de setup al iniciar el programa se verifica que el termómetro infrarrojo esté funcionando, para a continuación iniciar la librería de Ufox (wisol) activando el dispositivo.

```
void setup() {
  Serial.begin(115200);
  if (!thermometer.begin()) {
    wisol.RST();
  }
  wisol.begin(9600);
  ss.begin(9600);
  pinMode(RXLED, OUTPUT);
}
```

Al iniciar la sección loop se leen los datos obtenidos por el módulo GPS, el sensor infrarrojo y la batería que esta aun disponible para ser usada por el dispositivo, al obtener los datos se los transforma en valores hexadecimales y se proporciona un payload que será usado por el back-end de Sigfox, para finalizar se envían los datos al servidor de Sigfox cada 10,18 minutos con la función wisol.SEND.

```
void loop() {
  delay(300);
  bool newData = false;
  unsigned long chars;
  unsigned short sentences, failed;
  uint16_t battery;
  uint16_t temperature = thermometer.readObjectTempC();
  while (ss.available()) {
    char c = ss.read();
    if (gps.encode(c))
      newData = true;
  }
  if (newData) {
    unsigned long age;
    gps.f_get_position(&latitude, &longitude, &age);
    latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude;
    longitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : longitude;
  }
  for (int x = 0; x < 10; x++) {
    temperature;
  }
  if (millis() - lastTx > 610000) {
    lastTx = millis();
    wisol.RST();
    digitalWrite(RXLED, LOW);
    battery = wisol.VOLT();
    Serial.print(latitude);
    Serial.print(" ");
    Serial.print(longitude);
    Serial.print(" ");
    Serial.print(temperature);
    Serial.print(" ");
    Serial.print(battery);
    Serial.println(" ");
    if (latitude >= 0 || longitude >= 0) {
      Serial.println("Coordenadas no validas");
      digitalWrite(RXLED, HIGH);
      wisol.SLEEP();
      delay(3000);
    } else {
      char buff[30] = "";
    }
  }
}
```

```

/*
    formatear a cadena y convertir los datos a valores hexadecimales,
    con un Payload Lat::float:32 Long::float:32 temp::int:16 bat::uint:16
*/

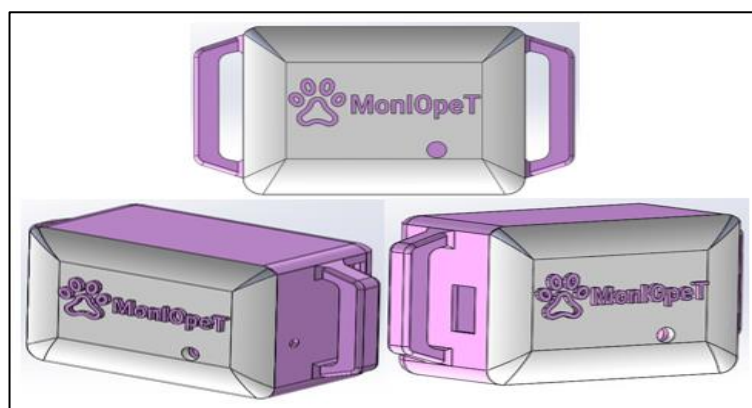
sprintf(buff, "%081x%081x%04x%04x", latitude, longitude, temperature, battery);
/*
    int=16bits=2bytes
    float float, fLon; float=32bits=4bytes
    Total=12Bytes
*/
Serial.print("Enviando: ");
Serial.println(buff);
Serial.println(wisol.SEND(buff)); //Envío de datos a Sigfox
digitalWrite(RXLED, HIGH);
wisol.SLEEP();
delay(3000);
}
}

```

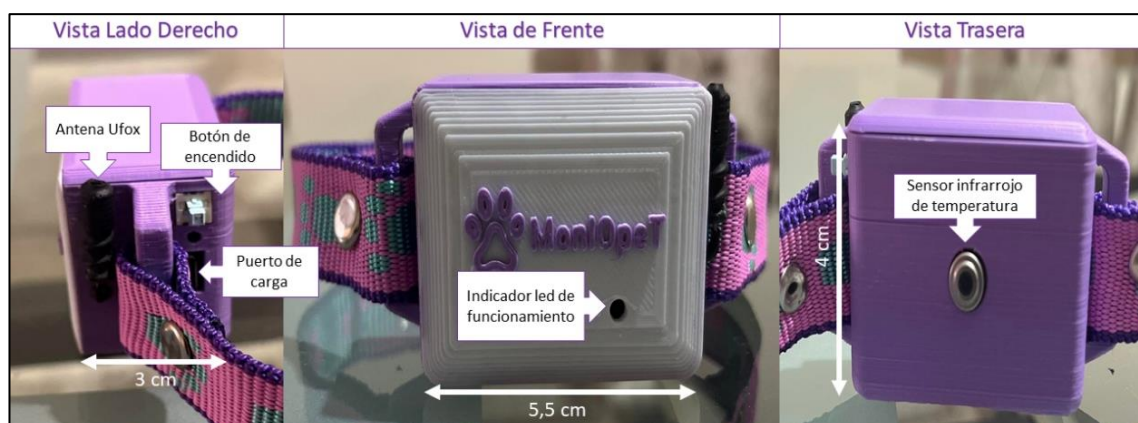
El dispositivo al encenderse usando el código desarrollado envía los datos obtenidos en constante funcionamiento durante 8 horas, pasado este tiempo el dispositivo necesita ser cargado.

### 3.2.2.3 Diseño 3D usado para la caja del prototipo del dispositivo

Para que los circuitos del dispositivo de monitoreo tengan una protección exterior se ha desarrollado un modelo 3D de caja, la cual será impresa en Filamento de Ácido Poliláctico (PLA).



**Figura 3.10** Diseño digital de la caja del prototipo  
Elaborado por: El investigador



**Figura 3.11** Diseño 3D impreso de la caja del prototipo con correa y vistas  
Elaborado por: El investigador

Las figuras 3.10 y 3.11 muestran el diseño usado para la impresión en PLA de la caja que cubrirá el circuito realizado del prototipo de dispositivo de monitoreo, como se puede apreciar, esta cuenta con una correa la cual ayudará para que el dispositivo pueda ser llevado por la mascota que será monitorizada.

#### 3.2.2.4 Configuración del servidor Sigfox

Para la comunicación del prototipo de dispositivo de monitoreo con el servidor de Sigfox es necesario realizar una configuración, ya que en el back-end de este servicio da la posibilidad de realizar varias acciones que ayudan al desarrollo de arquitecturas IoT.

#### Activación de placa Ufox y creación de cuenta Sigfox

Teca la empresa desarrolladora de la placa Ufox tiene un método de activación de sus productos, este proceso se muestra a continuación:

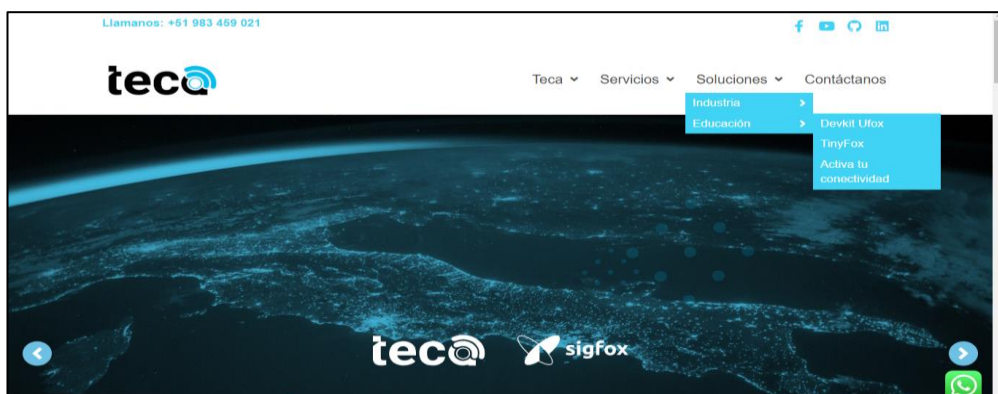


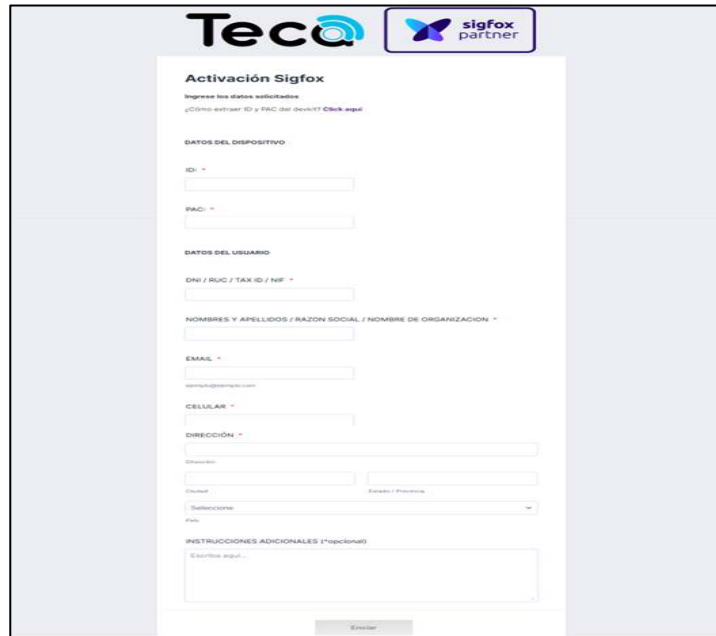
Figura 3.12 Página oficial de la Empresa Teca  
Elaborado por: [43]

En la figura 3.12 se muestra que tiene en la sección de soluciones y en la opción educación la posibilidad de activar la conexión de Sigfox.



**Figura 3.13 Pagina activación de conectividad Sigfox.**  
Elaborado por: [43]

Al ingresar a la opción activa tu conexión de la figura 3.12, en la figura 3.13 se muestra la opción activa tu conectividad aquí.

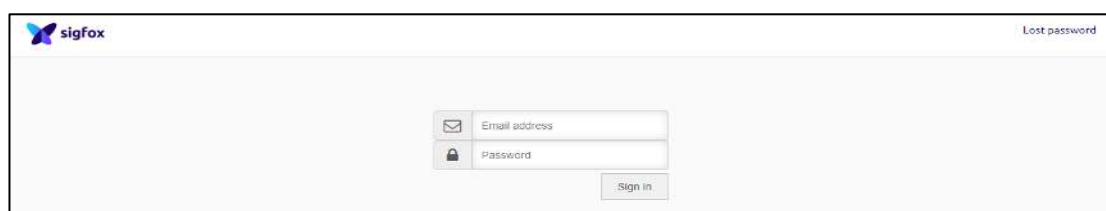


**Figura 3.14 Formulario de registro de placa Ufox y creación de cuenta Sigfox**  
Elaborado por: [43]

En la figura 3.14 se observa el formulario de registro del dispositivo Ufox, la información necesaria para el registro es el ID y el PAC de la placa Ufox, esta información se la obtiene al conectar la placa a una computadora y ejecutar el código proporcionado por la documentación de Ufox [44] en el IDE de Arduino. Es importante tener en cuenta que los datos del usuario ingresados serán usados para la creación de la cuenta back-end de Sigfox.

### Back-end Sigfox

El back-end de Sigfox es una página en la cual existen funciones que ayudan a organizar las actividades que realiza un dispositivo con tecnología Sigfox.

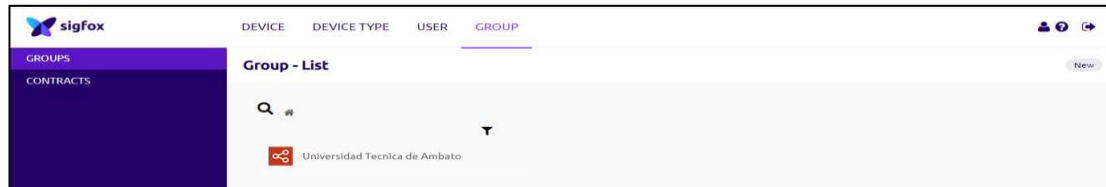


**Figura 3.15 Login back-end Sigfox**  
Elaborado por: [51]

En la figura 3.15 se muestra el formulario de ingreso al back-end de Sigfox.

## Usuario y API

Al crear la cuenta de Sigfox se proporciona un grupo por defecto, este grupo es el encargado de albergar los usuarios que usaran el back-end de Sigfox.



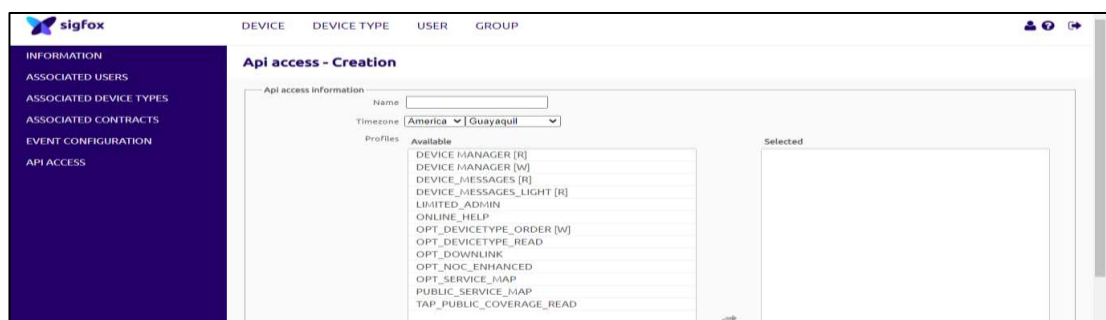
**Figura 3.16 Grupo back-end Sigfox**  
Elaborado por: [51]

En la figura 3.16 se muestra el grupo usado para el desarrollo del proyecto de investigación, en este caso Universidad Técnica de Ambato.



**Figura 3.17 Información Grupo back-end Sigfox**  
Elaborado por: [51]

En la figura 3.17 se muestra que al seleccionar un grupo se observa su información y la opción API ACCES, la cual gestiona el acceso a los métodos REST de Sigfox.



**Figura 3.18 Permisos API back-end Sigfox**  
Elaborado por: [51]

En la figura 3.18 se muestra que al seleccionar API ACCES y crear un acceso nuevo podemos agregarle un nombre y seleccionar que permisos tendrá.

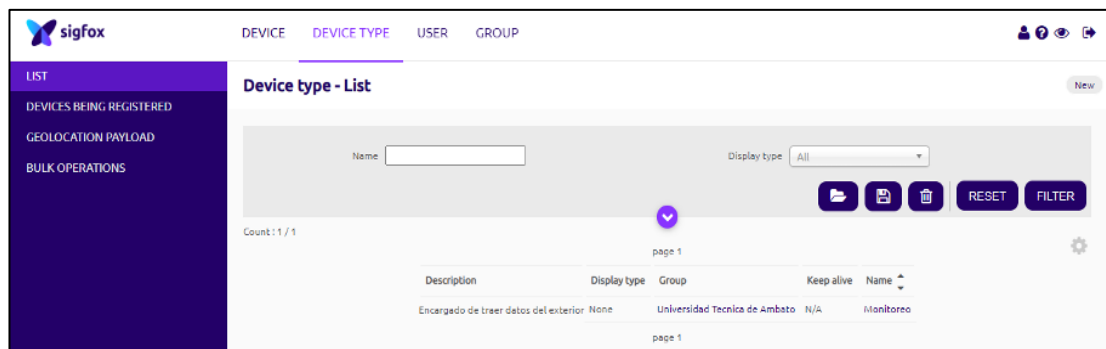


**Figura 3.19 Información API back-end Sigfox**  
Elaborado por: [51]

En la figura 3.19 se muestra que al crear el API ACCES se despliega automáticamente un login y password, los cuales serán usados por la PWA para tener acceso y crear funciones callback.

### Tipo de dispositivo

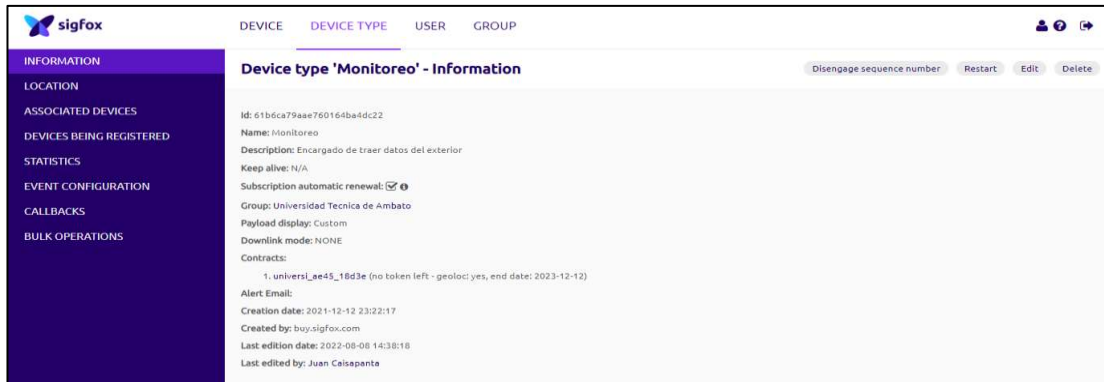
Los tipos de dispositivos ayudan para identificar que función realiza cada dispositivo en el back-end de Sigfox.



**Figura 3.20 Lista de tipos de dispositivos back-end Sigfox**  
Elaborado por: [51]

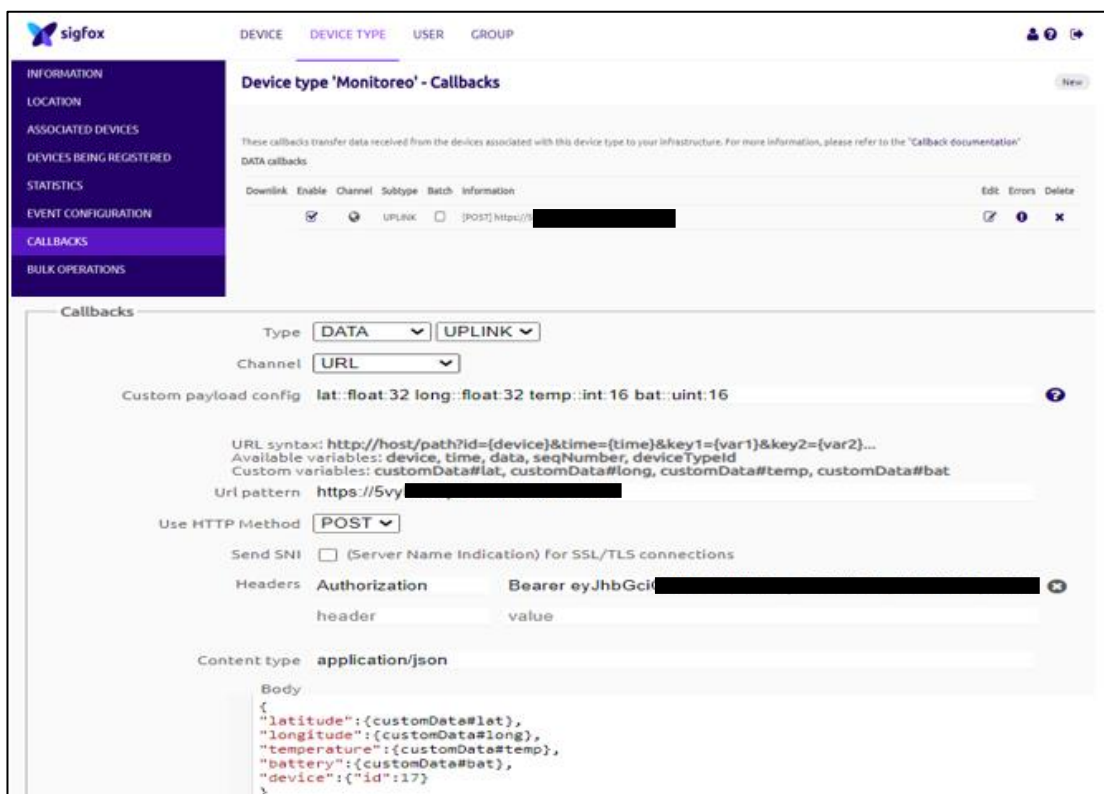
En la figura 3.20 se muestra el listado de tipos de dispositivos que han sido creados. Un tipo de dispositivo se crea por defecto al crear la cuenta de Sigfox en este caso se asignó el nombre de Monitoreo.





**Figura 3.21 Información del tipo de dispositivo back-end Sigfox**  
Elaborado por: [51]

En la figura 3.21 se muestra que al pulsar click en el nombre del tipo de dispositivo se muestra su información, se puede observar que el back-end de Sigfox asigna un Id el cual es usado como identificador para las funciones callback.



**Figura 3.22 Callbacks back-end Sigfox**  
Elaborado por: [51]

En la figura 3.22 se muestra la lista de funciones callback realizadas, en el proyecto de investigación se las usa para tener comunicación entre el back-end de Sigfox y el back-end de la PWA.

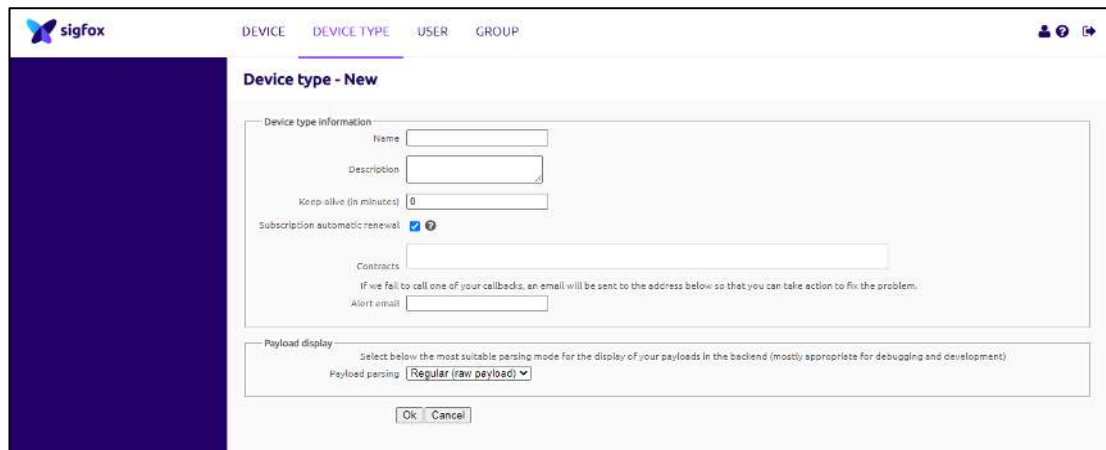


En caso de no tener creado un tipo de dispositivo se puede crear fácilmente uno, el proceso se muestra a continuación:



**Figura 3.23 Selección del grupo al que pertenecerá el tipo de dispositivo back-end Sigfox**  
Elaborado por: [51]

En la figura 3.23 se muestra la selección de grupo en el cual se basará el back-end de Sigfox para asignar un tipo de dispositivo.

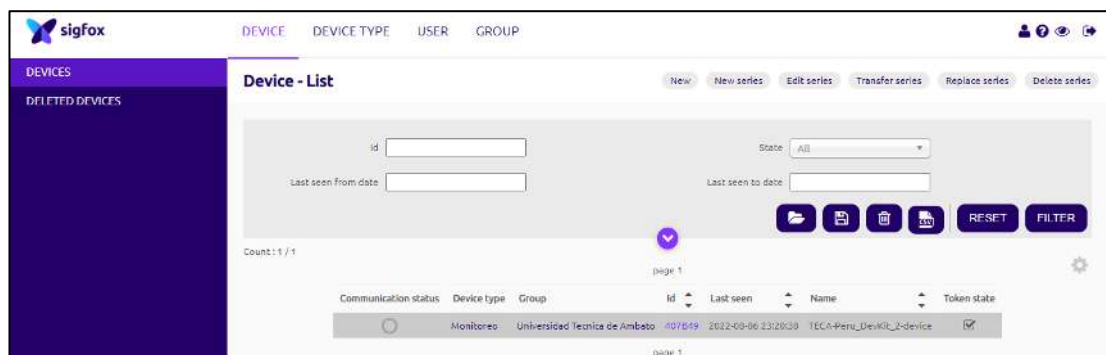


**Figura 3.24 Formulario de creación de nuevo tipo de dispositivo back-end Sigfox**  
Elaborado por: [51]

En la figura 3.24 se muestra el formulario que se despliega una vez seleccionado el grupo al que pertenecerá el nuevo tipo de dispositivo.

## Mensajes recibidos

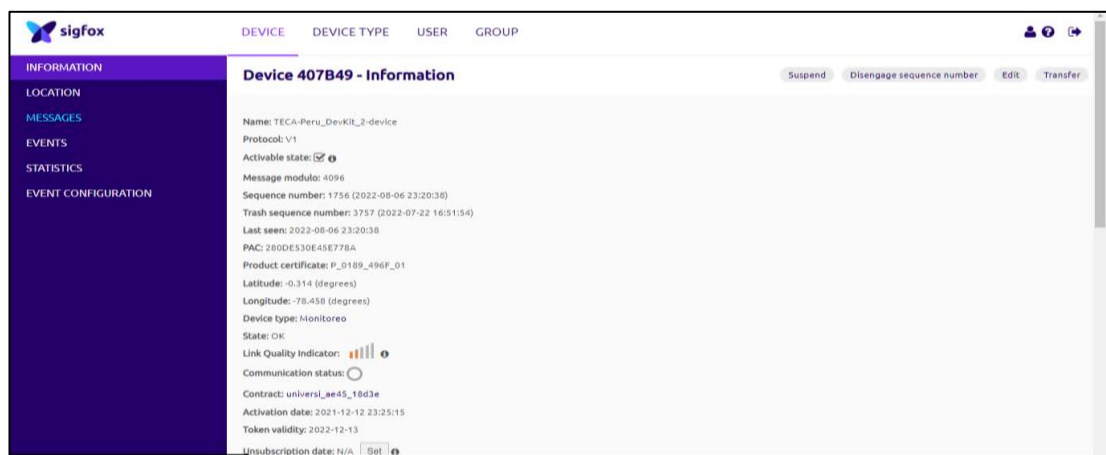
El dispositivo Sigfox al enviar un mensaje con los datos que está adquiriendo, se almacenan en la sección de mensajes del dispositivo en el back-end de Sigfox.



**Figura 3.25 Lista de dispositivos del usuario back-end Sigfox**  
**Elaborado por: [51]**

En la figura 3.25 se muestra la lista de dispositivos que tiene el usuario, donde se puede visualizar la información de cada uno de ellos, como el estado de comunicación entre el dispositivo y el back-end de Sigfox, su tipo, el grupo al que pertenece, su identificador (Id), la fecha del último mensaje, su nombre asignado y el estado del token que permite el funcionamiento del servicio Sigfox en el dispositivo.

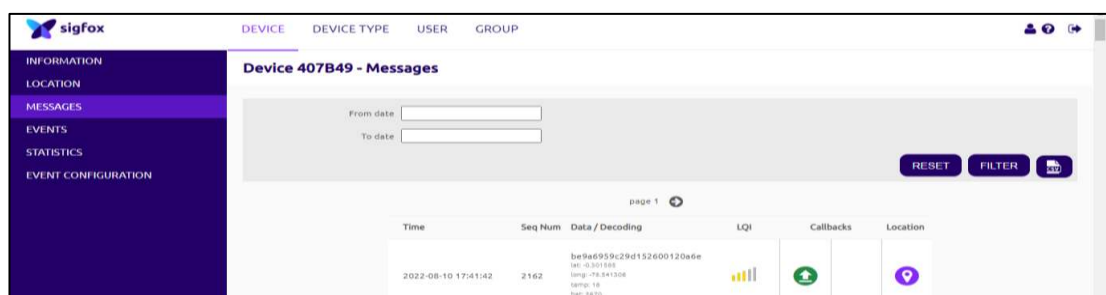
Al pulsar click en el Id del dispositivo se puede ingresar a su información para administrarla.



**Figura 3.26 Información de dispositivo back-end Sigfox**  
**Elaborado por: [51]**

En la figura 3.26 se muestra que al ingresar al Id del dispositivo podemos observar principalmente su estado, la secuencia del número de mensajes recibidos, su latitud y longitud de donde fue enviado el mensaje, la cantidad de señal que hubo, la fecha de activación del dispositivo y la fecha de expiración del token de validación para su funcionamiento.

Además, se muestra la opción de ingreso a los mensajes recibidos en el back-end de Sigfox.



**Figura 3.27 Lista de mensajes enviados por el dispositivo back-end Sigfox**  
**Elaborado por: [51]**

En la figura 3.27 se muestra que al ingresar a la opción mensajes se observan los datos enviados por el dispositivo, estos están codificados en hexadecimales y tienen un tamaño de 12 bytes que es lo que la tecnología Sigfox admite, junto a la fecha y hora de los datos recibidos, la señal que está recibiendo el dispositivo, además el estado de la función callback ejecutada y la ubicación de donde se recibieron los datos.

### **3.2.3 Aplicación de la metodología ágil XP al desarrollo de la PWA**

La metodología de desarrollo previamente escogida para el desarrollo de la PWA es XP, ya que su método de trabajo por fases ayudará a la mejor organización del proyecto de investigación.

#### **3.2.3.1 Análisis y requerimientos de la PWA**

La información obtenida a través de la encuesta realizada a dueños de mascotas en la ciudad de Quito dio un enfoque de los requerimientos y procesos que se deben integrar en la PWA.

La PWA contará con dos perfiles de usuarios, siendo estos el perfil administrador y el perfil dueño de mascotas.

El perfil administrador se encargará de gestionar la información de todos los usuarios, mascotas, dispositivos, establecimientos, medicinas, profesionales, especies, razas, tipos de establecimientos, tipos de medicinas y especialidades de los profesionales de la PWA.

En el perfil dueño de mascotas, el usuario podrá actualizar sus datos, como también gestionar los dispositivos de monitoreo e información de sus mascotas.

Una de las ventajas de una PWA es su funcionamiento multiplataforma, por lo que la interfaz del perfil administrador estará enfocada en pantallas de computadora y para el perfil dueño de mascotas en dispositivos móviles.

#### **3.2.3.2 Definición de roles**

Los roles de la metodología XP fueron definidos ajustándose al desarrollo del proyecto de investigación, estos roles son los siguientes:

<b>Nombre</b>	<b>Descripción</b>	<b>Rol</b>
Juan Caisapanta	Encargado de realizar la planificación, diseño, codificación, pruebas y producción de la PWA	Programador
Ing. Clay Aldás	Experto encargado impartir conocimiento que ayudara al desarrollo de la PWA	Coach
Fernanda Toapaxi	Encargada de revisar el perfecto funcionamiento de la PWA una vez puesta en fase de producción.	Cliente Tester

**Tabla 3.8 Definición de roles XP**  
Elaborado por: El investigador

### 3.2.3.3 Fase de planificación

La fase de planificación es la primera fase de la metodología XP, en esta fase se organiza la elaboración de las historias de usuarios, las cuales son el resultado de la comunicación continua del equipo de desarrollo y el cliente, además se establece el tiempo de desarrollo y que prioridad tienen las tareas a realizarse.

#### Historias de usuario

Las historias de usuario se generan de acuerdo con los requerimientos de la PWA establecidos por la encuesta a los dueños de mascotas en la ciudad de Quito y documentos que aportan información sobre la gestión de la información de mascotas.

El formato que será usado para el desarrollo de las historias de usuario es el siguiente:

<b>Historia de Usuario</b>	
<b>Código:</b>	<b>Usuario:</b>
<b>Nombre de historia:</b>	
<b>Prioridad en negocio:</b>	<b>Riesgo en desarrollo:</b>
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b>	
<b>Descripción:</b>	
<b>Observación:</b>	

**Tabla 3.9 Plantilla para Historias de usuario**  
Elaborado por: El investigador

En la tabla 3.9 se muestran los siguientes elementos

- Código: Identificador de la historia de usuario.
- Usuario: Persona que proporciona la información.
- Nombre de la historia: Título de la historia de usuario.

- **Prioridad en negocio:** Grado de prioridad basado en la necesidad del negocio, estas pueden ser (Alta, Media y Baja).
- **Riesgo en desarrollo:** Valor de complejidad que una historia de usuario representa basado en la probabilidad de fallo en el proceso de desarrollo, estas pueden ser (Alto, Medio, Bajo).
- **Puntos estimados:** Valor calculado en base a la cantidad de esfuerzo necesario para completar una historia de usuario, donde su punto máximo de dificultad es 5 y el más bajo 1.
- **Iteración asignada:** Número de iteración asignada a cada historia de usuario.
- **Programador responsable:** Persona encargada del desarrollo de cada historia de usuario
- **Descripción:** Se describe con un lenguaje natural sobre lo que debe hacer cada historia
- **Observación:** Información de inconvenientes presentadas en la historia de usuario.

Las historias de usuario a ser desarrolladas son las siguientes:

<b>Historia de Usuario</b>	
<b>Código:</b> H01	<b>Usuario:</b> Programador
<b>Nombre de historia:</b> Estructura del proyecto	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> Establecer la estructura del proyecto, donde se determinará como se usarán las tecnologías seleccionadas para el desarrollo de la PWA.	
<b>Observación:</b> Ninguna	

**Tabla 3.10 Historia de usuario H01 Estructura del proyecto**  
Elaborado por: El investigador

<b>Historia de Usuario</b>	
<b>Código:</b> H02	<b>Usuario:</b> Programador
<b>Nombre de historia:</b> Diseño de la base de datos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Juan Caisapanta	

<b>Descripción:</b> Diseñar la base de datos considerando las entidades que intervienen en el desarrollo de la PWA.
<b>Observación:</b> Ninguna

Tabla 3.11 Historia de usuario H02 Diseño de la base de datos  
Elaborado por: El investigador

Historia de Usuario	
<b>Código:</b> H03	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Registro de usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota ingresará sus datos personales junto a una contraseña.	
<b>Observación:</b> Todos los campos a ingresar son obligatorios y la contraseña será de 6 caracteres. El administrador no necesitará registrarse en la PWA	

Tabla 3.12 Historia de usuario H03 Registro de usuario  
Elaborado por: El investigador

Figura 3.28 Boceto preliminar de la historia de usuario H03 Registro de usuario  
Elaborado por: El investigador

Historia de Usuario	
<b>Código:</b> H04	<b>Usuario:</b> Administrador, Dueño de mascota
<b>Nombre de historia:</b> Inicio de sesión	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> Los usuarios ingresarán su correo electrónico registrado y su contraseña para iniciar sesión.	

**Observación:** El correo electrónico es verificado y la contraseña debe ser de 6 caracteres.

**Tabla 3.13 Historia de usuario H04 Inicio de sesión**  
Elaborado por: El investigador

Logo

**BIENVENIDO**

Por favor ingrese su correo electrónico y contraseña

Correo Electrónico

Contraseña

Olvidó su contraseña?

**INICIAR SESIÓN**

Nuevo en la plataforma? [Regístrate](#)

**Figura 3.29 Boceto preliminar de la historia de usuario H04 Inicio de sesión**  
Elaborado por: El investigador

Historia de Usuario	
<b>Código:</b> H05	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Olvido de contraseña	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota ingresará su correo electrónico para recibir su nueva contraseña.	
<b>Observación:</b> La nueva contraseña de 6 caracteres es enviada al correo electrónico ingresado por el usuario.	

**Tabla 3.14 Historia de usuario H05 Olvido de Contraseña**  
Elaborado por: El investigador

Logo

**BIENVENIDO**

Ingrese su correo electrónico para obtener su contraseña

Correo Electrónico

T: ----

**SOLICITAR** **CANCELAR**

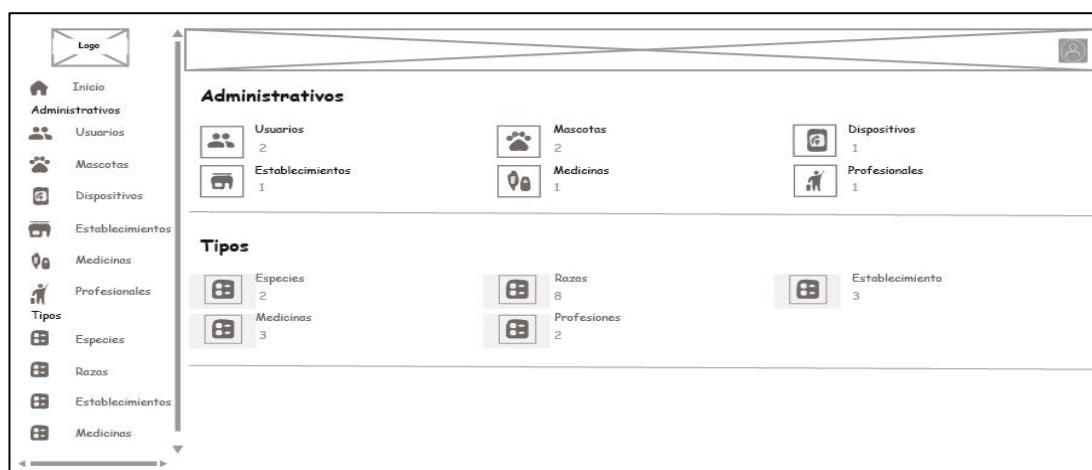
Nuevo en la plataforma? [Regístrate](#)

**Figura 3.30 Boceto preliminar de la historia de usuario H05 Olvido de contraseña**  
Elaborado por: El investigador

Historia de Usuario	
<b>Código:</b> H06	<b>Usuario:</b> Administrador

<b>Nombre de historia:</b> Panel de administración	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario administrador observará las opciones a administrar, además de un conteo total de los datos de cada sección.	
<b>Observación:</b> Las opciones a administrar se despliegan en un menú y el total de los datos de cada sección en cartas.	

**Tabla 3.15 Historia de usuario H06 Panel de administración**  
Elaborado por: El investigador



**Figura 3.31 Boceto preliminar de la historia de usuario H06 Panel de administración**  
Elaborado por: El Investigador

<b>Historia de Usuario</b>	
<b>Código:</b> H07	<b>Usuario:</b> Administrador
<b>Nombre de historia:</b> Gestión de usuarios	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario administrador podrá visualizar todos los usuarios y además guardar, actualizar y eliminar la información de un usuario.	
<b>Observación:</b> Solo el usuario administrador podrá realizar las acciones de gestión de los usuarios.	

**Tabla 3.16 Historia de usuario H07 Gestión de usuarios**  
Elaborado por: El investigador

ID	Nombre	Apellido	Correo Electrónico	Celular	Rol	Acciones
01	Usuario 1	Usuario 1	Usuario_1@hotmail.com	0987654321	User	VER
	Usuario 2	Usuario 2	Usuario_2@hotmail.com	0987654321	User	VER



**Figura 3.32 Boceto preliminar de la sección 1 de la historia de usuario H07 Gestión de usuarios**  
Elaborado por: El investigador

**Figura 3.33 Boceto preliminar de la sección 2 de la historia de usuario H07 Gestión de usuarios**  
Elaborado por: El investigador

**Figura 3.34 Boceto preliminar de la sección 3 de la historia de usuario H07 Gestión de usuarios**  
Elaborado por: El investigador

<b>Historia de Usuario</b>	
<b>Código:</b> H08	<b>Usuario:</b> Administrador
<b>Nombre de historia:</b> Gestión de mascotas	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario administrador podrá visualizar todas las mascotas y además guardar, actualizar y eliminar la información de una mascota.	
<b>Observación:</b> Solo el usuario administrador podrá realizar las acciones de gestión de las mascotas.	

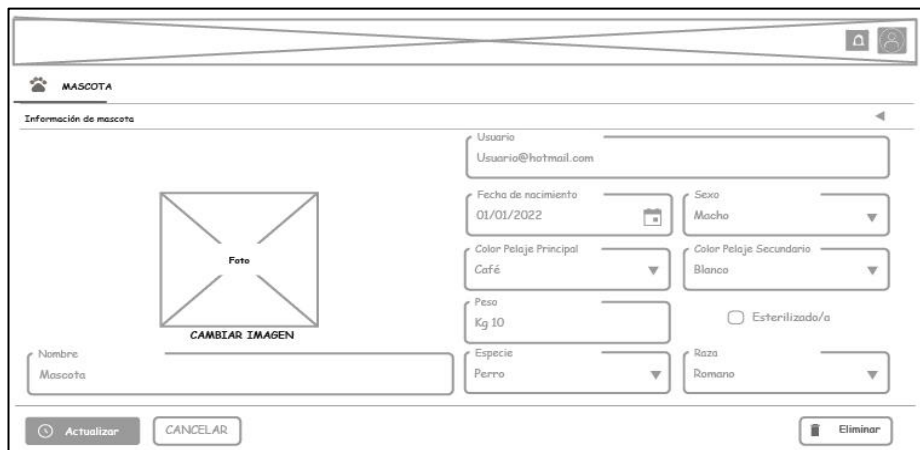
**Tabla 3.17 Historia de usuario H09 Gestión de mascotas**  
Elaborado por: El investigador



**Figura 3.35 Boceto preliminar de la sección 1 de la historia de usuario H08 Gestión de mascotas**  
Elaborado por: El investigador



**Figura 3.36 Boceto preliminar de la sección 2 de la historia de usuario H08 Gestión de mascotas**  
Elaborado por: El investigador



**Figura 3.37 Boceto preliminar de la sección 3 de la historia de usuario H08 Gestión de mascotas**  
Elaborado por: El investigador

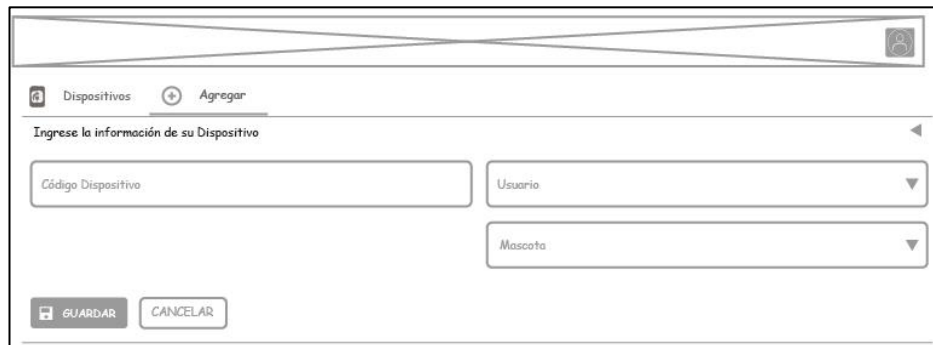
<b>Historia de Usuario</b>	
<b>Código:</b> H09	<b>Usuario:</b> Administrador
<b>Nombre de historia:</b> Gestión de dispositivos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	

<b>Descripción:</b> El usuario administrador podrá visualizar todos los dispositivos y además guardar y eliminar un dispositivo.
<b>Observación:</b> Solo el usuario administrador podrá realizar las acciones de gestión de los dispositivos.

**Tabla 3.18 Historia de usuario H08 Gestión de dispositivos**  
Elaborado por: El investigador



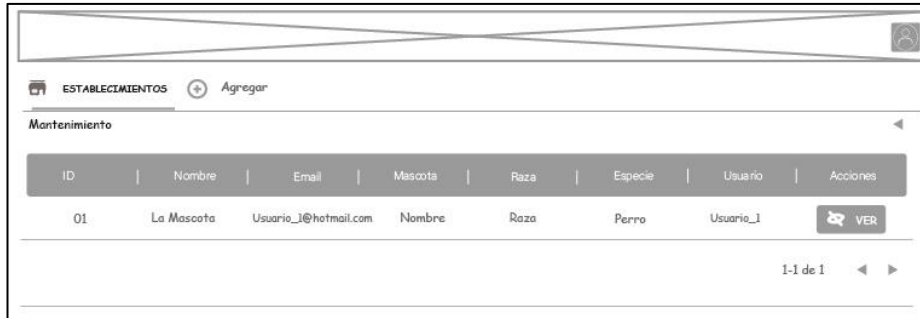
**Figura 3.38 Boceto preliminar de la sección 1 de la historia de usuario H09 Gestión de dispositivos**  
Elaborado por: El investigador



**Figura 3.39 Boceto preliminar de la sección 2 de la historia de usuario H09 Gestión de dispositivos**  
Elaborado por: El investigador

<b>Historia de Usuario</b>	
<b>Código:</b> H10	<b>Usuario:</b> Administrador
<b>Nombre de historia:</b> Gestión de establecimientos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario administrador podrá visualizar todos los establecimientos y además guardar, actualizar y eliminar la información de un establecimiento.	
<b>Observación:</b> Solo el usuario administrador podrá realizar las acciones de gestión de los establecimientos.	

**Tabla 3.19 Historia de usuario H10 Gestión de establecimientos**  
Elaborado por: El investigador



**Figura 3.40 Boceto preliminar de la sección 2 de la historia de usuario H10 Gestión de establecimientos**  
**Elaborado por: El investigador**



**Figura 3.41 Boceto preliminar de la sección 2 de la historia de usuario H10 Gestión de establecimientos**  
**Elaborado por: El investigador**



**Figura 3.42 Boceto preliminar de la sección 3 de la historia de usuario H10 Gestión de establecimientos**  
**Elaborado por: El investigador**

<b>Historia de Usuario</b>	
<b>Código:</b> H11	<b>Usuario</b> Administrador
<b>Nombre de historia:</b> Gestión de medicinas	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	

**Descripción:** El usuario administrador podrá visualizar todas las medicinas y además guardar, actualizar y eliminar la información de una medicina.

**Observación:** Solo el usuario administrador podrá realizar las acciones de gestión de las medicinas.

**Tabla 3.20 Historia de usuario H11 Gestión de medicinas**

**Elaborado por: El investigador**

ID	Nombre	Aplicador	Fecha de aplicación	Mascota	Usuario	Acciones
01	Nombre mascota	Nombre	01/01/2000	Raza	Usuario_1@hotmail.com	VER

**Figura 3.43 Boceto preliminar de la sección 1 de la historia de usuario H11 Gestión de medicinas**

**Elaborado por: El investigador**

Ingrese la información de la Medicina Aplicada

**CAMBIAR IMAGEN**

Nombre:  Fabricante:

Usuario:

Mascota:

Lote:  Aplicador:

Tipo Analgésico:

Fecha de aplicación:

Fecha Producción:  Fecha Expiración:

Descripción:

**Figura 3.44 Boceto preliminar de la sección 2 de la historia de usuario H11 Gestión de medicinas**

**Elaborado por: El investigador**

Información de medicamento (nombre medicamento)

**CAMBIAR IMAGEN**

Nombre:  Fabricante:

Usuario:

Mascota:

Lote:  Aplicador:

Tipo Analgésico:

Fecha de aplicación:

Fecha Producción:  Fecha Expiración:

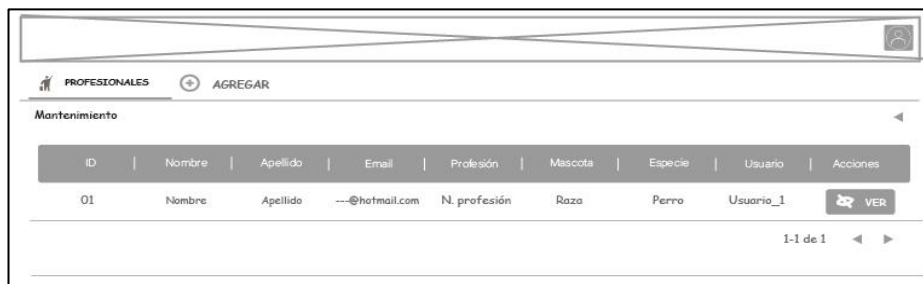
Descripción:

**Figura 3.45 Boceto preliminar de la sección 3 de la historia de usuario H11 Gestión de medicinas**

**Elaborado por: El investigador**

Historia de Usuario	
<b>Código:</b> H12	<b>Usuario:</b> Administrador
<b>Nombre de historia:</b> Gestión de profesionales	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario administrador podrá visualizar todos los profesionales y además guardar, actualizar y eliminar la información de un profesional.	
<b>Observación:</b> Solo el usuario administrador podrá realizar las acciones de gestión de los profesionales.	

**Tabla 3.21 Historia de usuario H12 Gestión de profesionales**  
Elaborado por: El investigador



**Figura 3.46 Boceto preliminar de la sección 1 de la historia de usuario H12 Gestión de profesionales**  
Elaborado por: El investigador



**Figura 3.47 Boceto preliminar de la sección 2 de la historia de usuario H12 Gestión de profesionales**  
Elaborado por: El investigador



**Figura 3.48 Boceto preliminar de la sección 3 de la historia de usuario H12 Gestión de profesionales**  
**Elaborado por: El investigador**

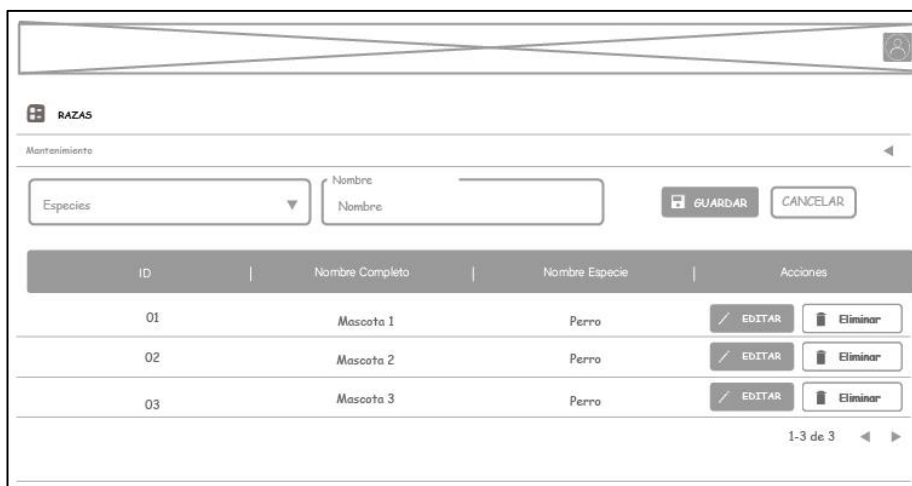
<b>Historia de Usuario</b>	
<b>Código:</b> H13	<b>Usuario:</b> Administrador
<b>Nombre de historia:</b> Gestión de especies	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario administrador podrá visualizar todas las especies y además guardar, actualizar y eliminar la información de una especie.	
<b>Observación:</b> Solo el usuario administrador podrá realizar las acciones de gestión de las especies.	

**Tabla 3.22 Historia de usuario H13 Gestión de especies**  
**Elaborado por: El investigador**

**Figura 3.49 Boceto preliminar de la historia de usuario H13 Gestión de especies**  
**Elaborado por: El investigador**

<b>Historia de Usuario</b>	
<b>Código:</b> H14	<b>Usuario:</b> Administrador
<b>Nombre de historia:</b> Gestión de razas	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario administrador podrá visualizar todas las razas y además guardar, actualizar y eliminar la información de una raza.	
<b>Observación:</b> Solo el usuario administrador podrá realizar las acciones de gestión de las razas.	

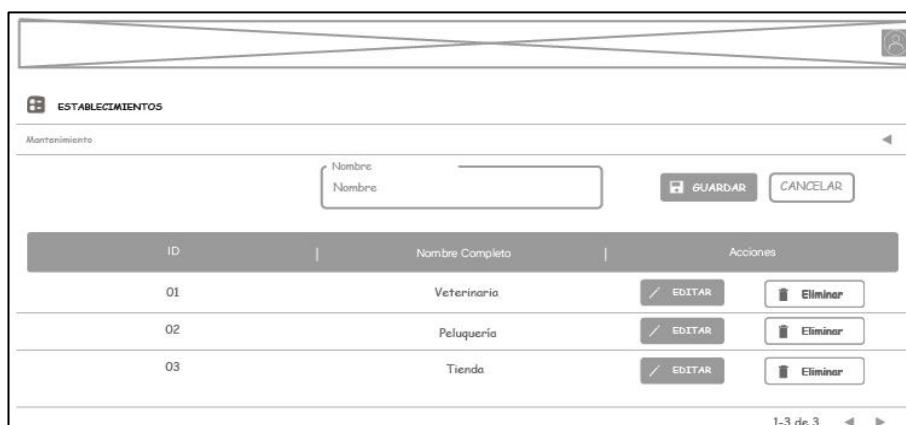
**Tabla 3.23 Historia de usuario H14 Gestión de razas**  
**Elaborado por: El investigador**



**Figura 3.50 Boceto preliminar de la historia de usuario H14 Gestión de razas**  
Elaborado por: El investigador

<b>Historia de Usuario</b>	
<b>Código:</b> H15	<b>Usuario:</b> Administrador
<b>Nombre de historia:</b> Gestión de tipos de establecimientos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario administrador podrá visualizar todos los tipos de establecimientos, además guardar, actualizar y eliminar la información de un tipo de establecimiento.	
<b>Observación:</b> Solo el usuario administrador podrá realizar las acciones de gestión de los tipos de establecimientos.	

**Tabla 3.24 Historia de usuario H15 Gestión de tipos de establecimientos**  
Elaborado por: El investigador



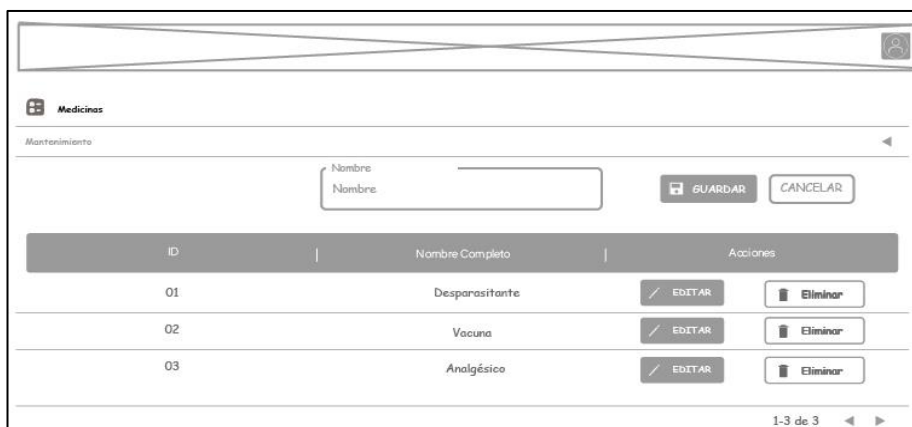
**Figura 3.51 Boceto preliminar de la historia de usuario H15 Gestión de tipos de establecimientos**  
Elaborado por: El investigador

<b>Historia de Usuario</b>	
<b>Código:</b> H16	<b>Usuario:</b> Administrador
<b>Nombre de historia:</b> Gestión de tipos de medicinas	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio



<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario administrador podrá visualizar todos los tipos de medicinas, además guardar, actualizar y eliminar la información de un tipo de medicina.	
<b>Observación:</b> Solo el usuario administrador podrá realizar las acciones de gestión de los tipos de medicinas.	

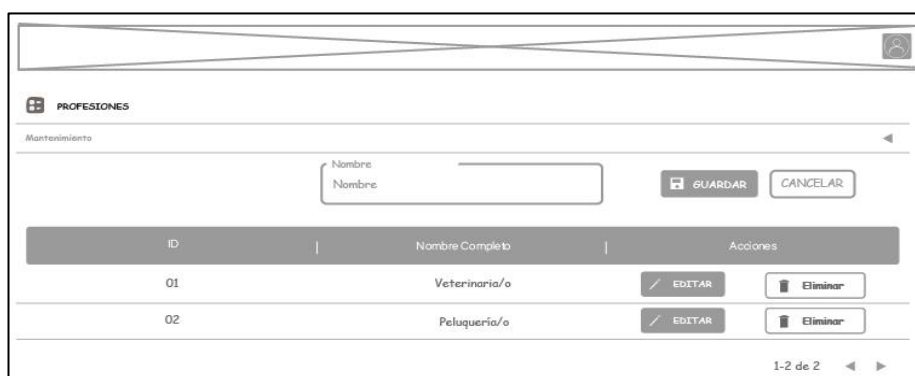
**Tabla 3.25 Historia de usuario H16 Gestión de tipos de medicinas**  
Elaborado por: El investigador



**Figura 3.52 Boceto preliminar de la historia de usuario H16 Gestión de tipos de medicinas**  
Elaborado por: El investigador

<b>Historia de Usuario</b>	
<b>Código:</b> H17	<b>Usuario:</b> Administrador
<b>Nombre de historia:</b> Gestión de profesiones	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario administrador podrá visualizar todos los tipos de profesiones, además guardar, actualizar y eliminar la información de un tipo de profesión.	
<b>Observación:</b> Solo el usuario administrador podrá realizar las acciones de gestión de las profesiones.	

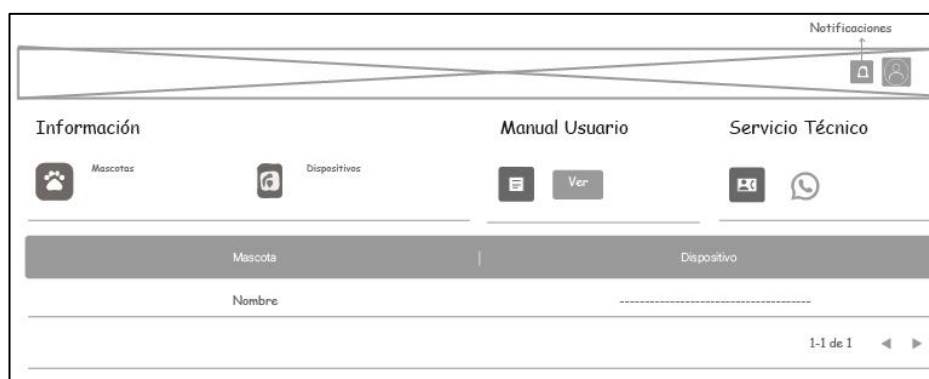
**Tabla 3.26 Historia de usuario H17 Gestión de profesiones**  
Elaborado por: El investigador



**Figura 3.53 Boceto preliminar de la historia de usuario H17 Gestión de profesiones**  
Elaborado por: El investigador

Historia de Usuario	
<b>Código:</b> H18	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Panel de usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota observará las opciones a administrar, la descarga del manual de usuario de la PWA, contacto de servicio técnico, una tabla con el código del dispositivo y su mascota asignada, además de un conteo total de los dispositivos y mascotas que fueron agregadas.	
<b>Observación:</b> Las opciones a administrar se despliegan en un menú y la descarga del manual de usuario, el contacto de servicio técnico y el total de los datos de los dispositivos y mascotas en cartas.	

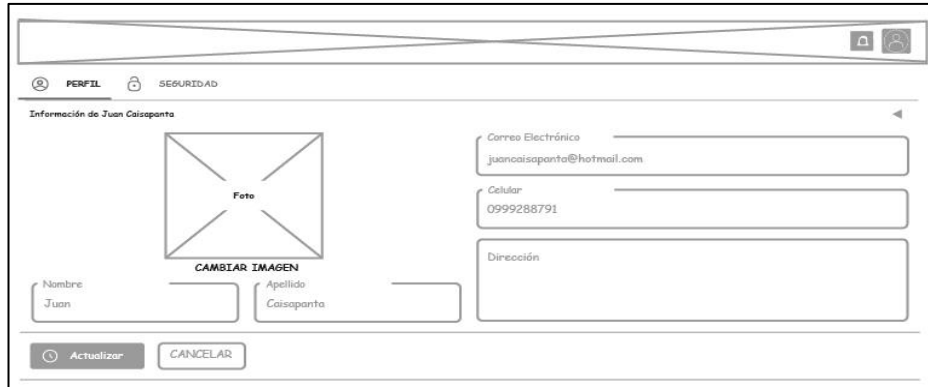
**Tabla 3.27 Historia de usuario H18 Panel de usuario**  
Elaborado por: El investigador



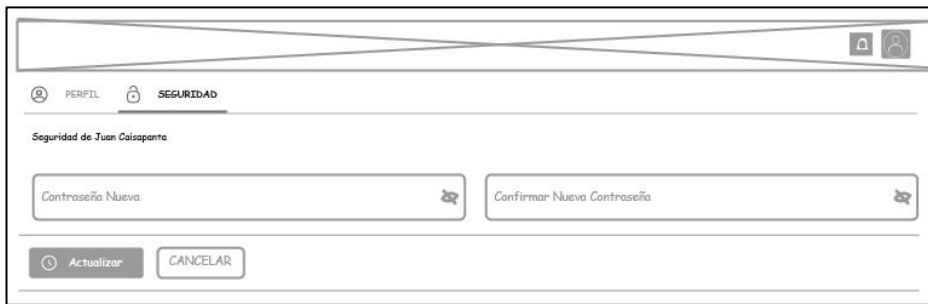
**Figura 3.54 Boceto preliminar de la historia de usuario H18 Panel de usuario**  
Elaborado por: El investigador

Historia de Usuario	
<b>Código:</b> H19	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Configuración cuenta de usuario	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota actualizará su información la cual fue proporcionada al registrarse en la PWA.	
<b>Observación:</b> La configuración de la cuenta del usuario dueño de la mascota se lo realizará a través de formularios controlados y su contraseña se actualizará por medio de dos campos, los cuales son nueva contraseña y confirmación de la nueva contraseña.	

**Tabla 3.28 Historia de usuario H19 Configuración cuenta de usuario**  
Elaborado por: El investigador



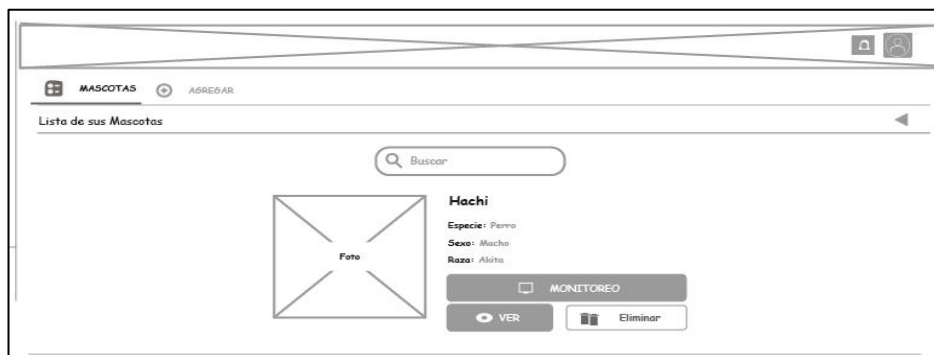
**Figura 3.55 Boceto preliminar de la sección 1 de la historia de usuario H19 Configuración cuenta de usuario**  
Elaborado por: El investigador



**Figura 3.56 Boceto preliminar de la sección 2 de la historia de usuario H19 Configuración cuenta de usuario**  
Elaborado por: El investigador

Historia de Usuario	
<b>Código:</b> H20	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Gestión de las mascotas del usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota podrá visualizar todas sus mascotas, además guardar, actualizar y eliminar la información de sus mascotas.	
<b>Observación:</b> Solo el usuario dueño de la mascota podrá realizar las acciones de gestión de sus mascotas	

**Tabla 3.29 Historia de usuario H20 Gestión de mascotas de usuario**  
Elaborado por: El investigador



**Figura 3.57 Boceto preliminar de la sección 1 de la historia de usuario H20 Gestión de las mascotas del usuario**  
**Elaborado por: El investigador**

**Figura 3.58 Boceto preliminar de la sección 2 de la historia de usuario H20 Gestión de las mascotas del usuario**  
**Elaborado por: El investigador**

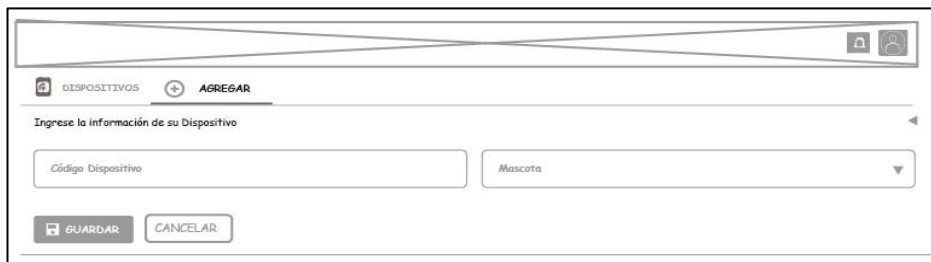
**Figura 3.59 Boceto preliminar de la sección 3 de la historia de usuario H20 Gestión de las mascotas del usuario**  
**Elaborado por: El investigador**

<b>Historia de Usuario</b>	
<b>Código:</b> H21	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Gestión de los dispositivos del usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota podrá visualizar todos sus dispositivos asignados a una mascota, además guardar y eliminar la información del dispositivo de una mascota.	
<b>Observación:</b> Solo el usuario dueño de la mascota podrá realizar las acciones de gestión de sus dispositivos.	

**Tabla 3.30 Historia de usuario H21 Gestión de dispositivos del usuario**  
**Elaborado por: El investigador**



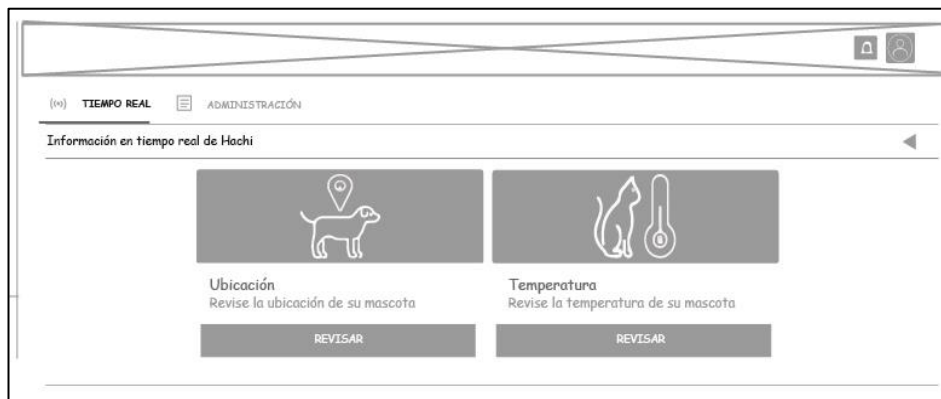
**Figura 3.60 Boceto preliminar de la sección 1 de la historia de usuario H21 Gestión de los dispositivos del usuario**  
Elaborado por: El investigador



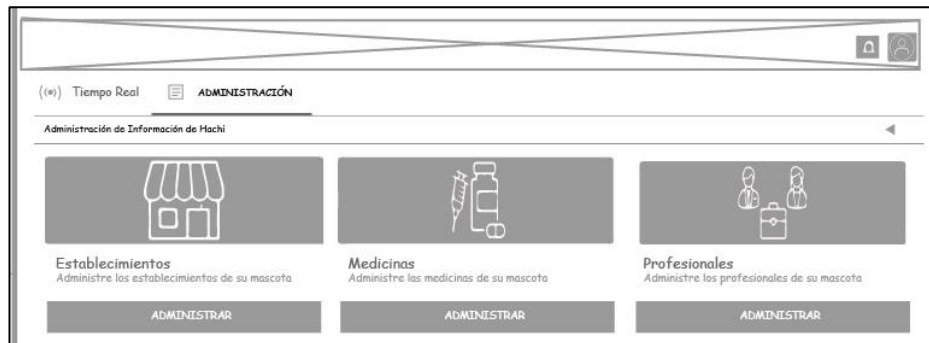
**Figura 3.61 Boceto preliminar de la sección 2 de la historia de usuario H21 Gestión de los dispositivos del usuario**  
Elaborado por: El investigador

<b>Historia de Usuario</b>	
<b>Código:</b> H22	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Secciones de monitorización	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota observará las opciones a administrar como lo son la ubicación, temperatura, establecimientos, medicinas y profesionales de su mascota.	
<b>Observación:</b> Las opciones se mostrarán en cartas.	

**Tabla 3.31 Historia de usuario H22 Secciones de monitorización**  
Elaborado por: El investigador



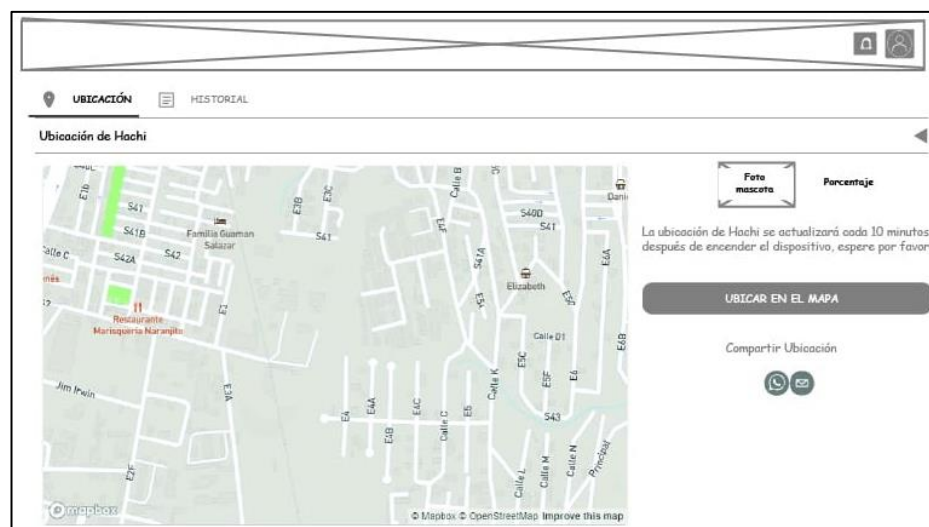
**Figura 3.62 Boceto preliminar de la sección 1 de la historia de usuario H22 Secciones de monitorización**  
**Elaborado por: El investigador**



**Figura 3.63 Boceto preliminar de la sección 2 de la historia de usuario H22 Secciones de monitorización**  
**Elaborado por: El investigador**

<b>Historia de Usuario</b>	
<b>Código:</b> H23	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Ubicación de la mascota del usuario en tiempo real	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 5	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota observará la ubicación de su mascota en tiempo real en un mapa de MapBox y con la posibilidad de compartir su ubicación.	
<b>Observación:</b> La ubicación de la mascota se podrá localizar a través del mapa proporcionado por Google Maps.	

**Tabla 3.32 Historia de usuario H23 Ubicación de la mascota del usuario en tiempo real**  
**Elaborado por: El investigador**

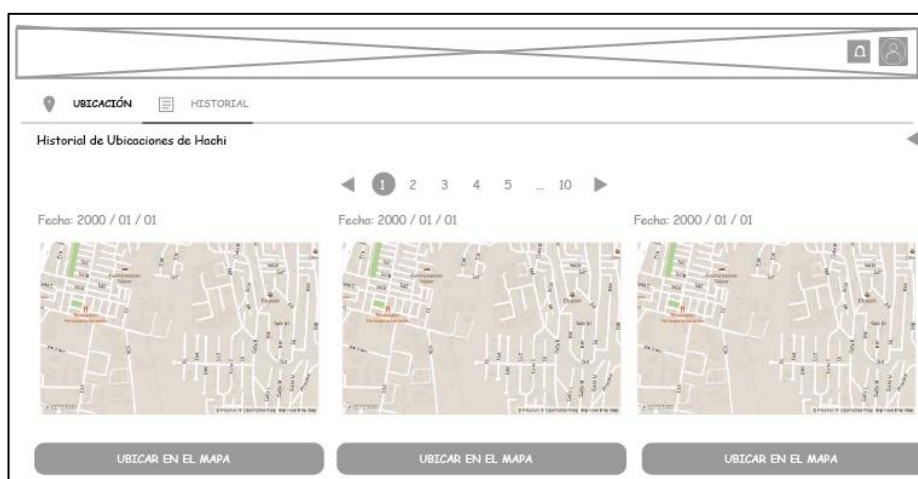


**Figura 3.64 Boceto preliminar de la historia de usuario H23 Ubicación de la mascota del usuario en tiempo real**  
**Elaborado por: El investigador**

<b>Historia de Usuario</b>	
<b>Código:</b> H24	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Historial de ubicaciones de la mascota del usuario en tiempo real	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota observará el historial de las ubicaciones de su mascota en mapas de MapBox en tiempo real.	
<b>Observación:</b> Ninguna.	

**Tabla 3.33 Historia de usuario H24 Historial de ubicaciones de la mascota del usuario en tiempo real**

Elaborado por: El investigador



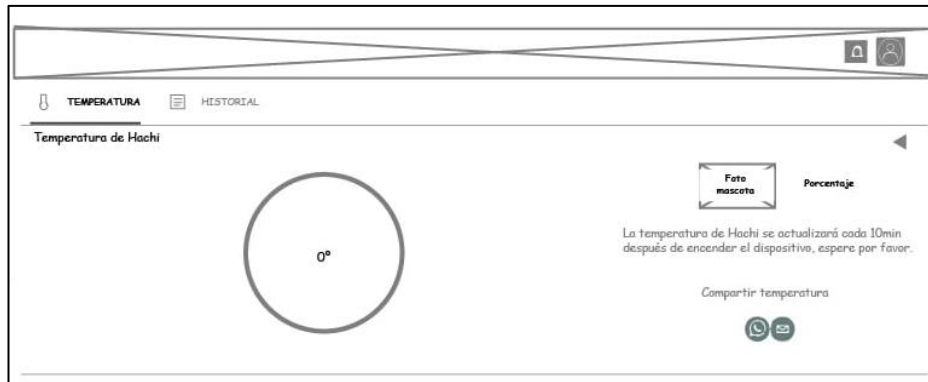
**Figura 3.65 Boceto preliminar de la historia de usuario H24 Historial de ubicaciones de la mascota del usuario en tiempo real**

Elaborado por: El investigador

<b>Historia de Usuario</b>	
<b>Código:</b> H25	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Temperatura de la mascota del usuario en tiempo real	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 5	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota observará la temperatura de su mascota en un gráfico en tiempo real determinado por colores y con la posibilidad de compartir su temperatura	
<b>Observación:</b> Ninguna.	

**Tabla 3.34 Historia de usuario H25 Temperatura de la mascota del usuario en tiempo real**

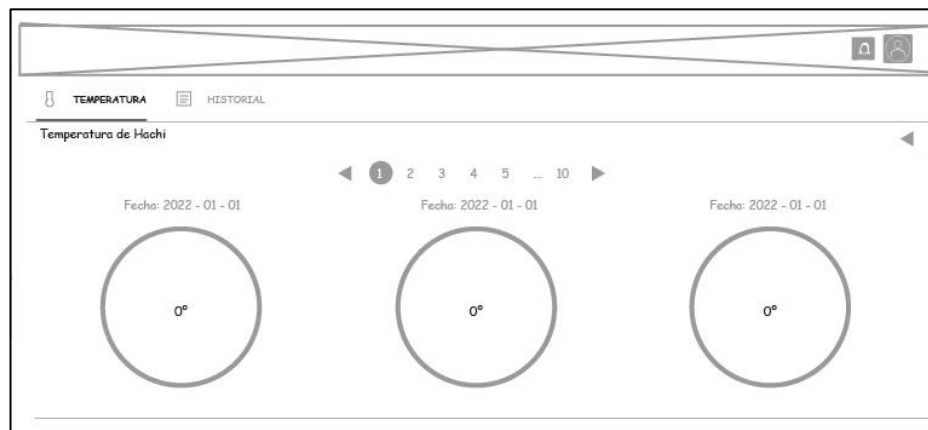
Elaborado por: El investigador



**Figura 3.66 Boceto preliminar de la historia de usuario H25 Temperatura de la mascota del usuario en tiempo real**  
Elaborado por: El investigador

<b>Historia de Usuario</b>	
<b>Código:</b> H26	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Historial de temperaturas de la mascota del usuario en tiempo real	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota observará el historial de las temperaturas de su mascota en gráficos determinado por colores en tiempo real.	
<b>Observación:</b> Ninguna.	

**Tabla 3.35 Historia de usuario H26 Historial de temperaturas de la mascota del usuario en tiempo real**  
Elaborado por: El investigador



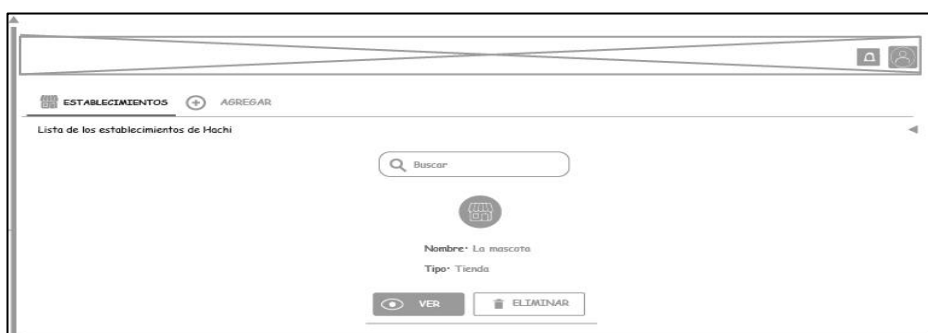
**Figura 3.67 Boceto preliminar de la historia de usuario H26 Historial de temperaturas de la mascota del usuario en tiempo real**  
Elaborado por: El investigador

<b>Historia de Usuario</b>	
<b>Código:</b> H27	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Gestión de establecimientos de la mascota	

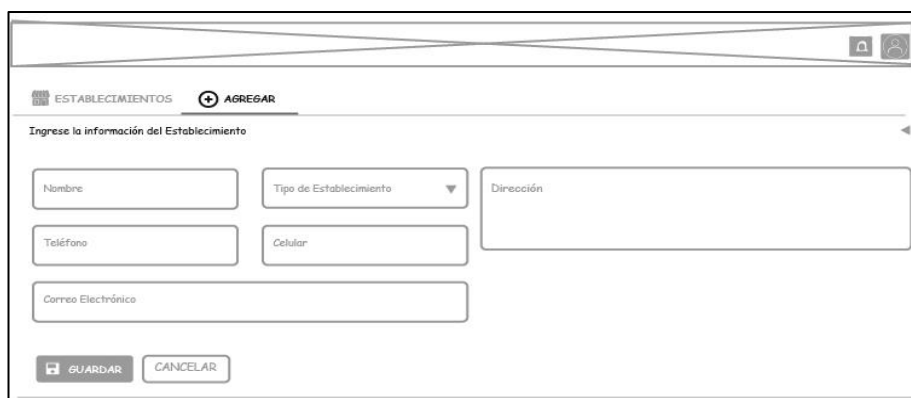


<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota podrá visualizar todos los establecimientos de su mascota, además guardar, actualizar y eliminar la información del establecimiento que frecuenta su mascota.	
<b>Observación:</b> Solo el usuario dueño de la mascota podrá realizar las acciones de gestión de los establecimientos de su mascota.	

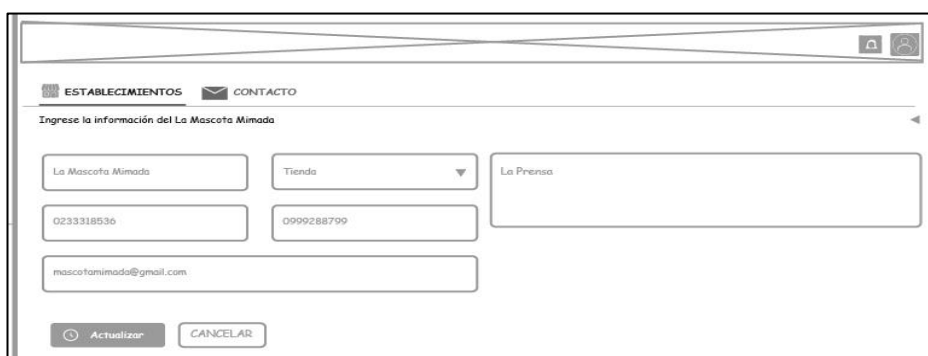
**Tabla 3.36 Historia de usuario H27 Gestión de establecimientos de la mascota**  
Elaborado por: El investigador



**Figura 3.68 Boceto preliminar de la sección 1 de la historia de usuario H27 Gestión de establecimientos de la mascota**  
Elaborado por: El investigador



**Figura 3.69 Boceto preliminar de la sección 2 de la historia de usuario H27 Gestión de establecimientos de la mascota**  
Elaborado por: El investigador



**Figura 3.70 Boceto preliminar de la sección 3 de la historia de usuario H27 Gestión de establecimientos de la mascota**  
Elaborado por: El investigador

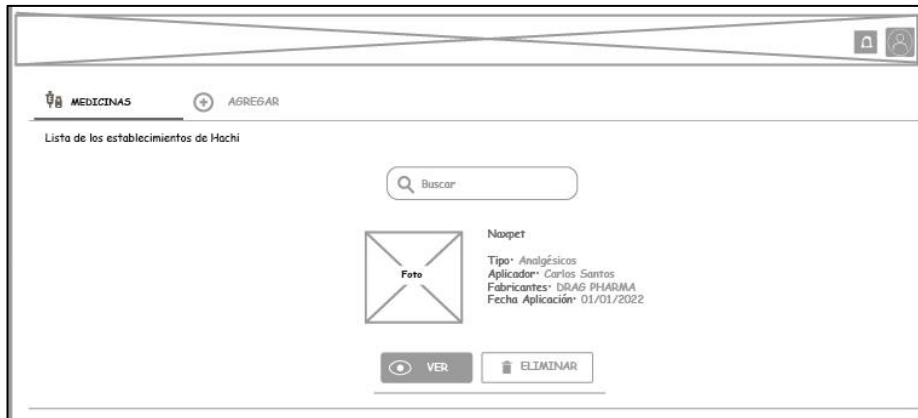
Historia de Usuario	
<b>Código:</b> H28	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Contactar establecimiento de la mascota	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota podrá conectar al establecimiento que frecuenta su mascota gracias a la información proporcionada.	
<b>Observación:</b> Podrá conectarse con el establecimiento que frecuenta su mascota por medio de correo electrónico o WhatsApp.	

**Tabla 3.37 Historia de usuario H28 Contactar establecimiento de la mascota**  
Elaborado por: El investigador

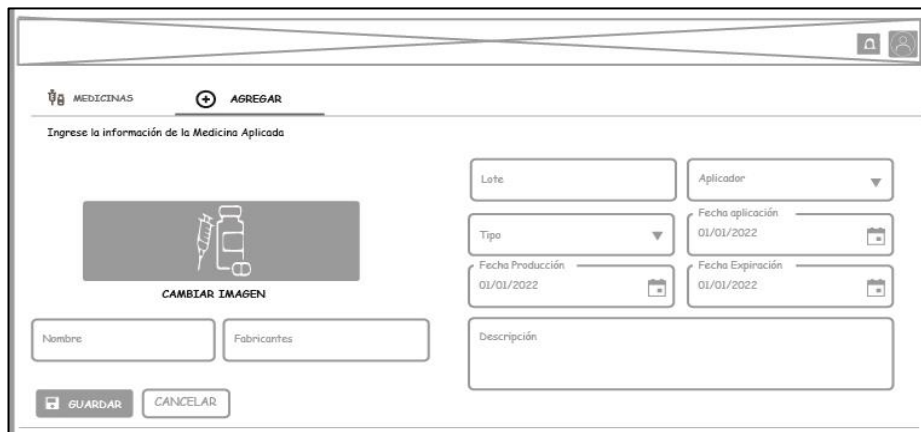
**Figura 3.71 Boceto preliminar de la historia de usuario H28 Contactar establecimiento de la mascota**  
Elaborado por: El investigador

Historia de Usuario	
<b>Código:</b> H29	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Gestión de medicinas de la mascota	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota podrá visualizar todas las medicinas de su mascota, además guardar, actualizar y eliminar la información de la medicina que usa su mascota.	
<b>Observación:</b> Solo el usuario dueño de la mascota podrá realizar las acciones de gestión de las medicinas de su mascota.	

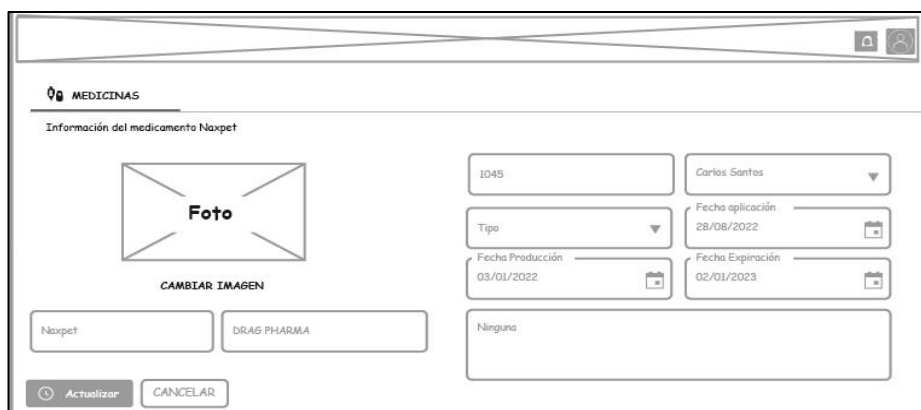
**Tabla 3.38 Historia de usuario H29 Gestión de medicinas de la mascota**  
Elaborado por: El investigador



**Figura 3.72 Boceto preliminar de la sección 1 de la historia de usuario H29 Gestión de medicinas de la mascota**  
**Elaborado por: El investigador**



**Figura 3.73 Boceto preliminar de la sección 2 de la historia de usuario H29 Gestión de medicinas de la mascota**  
**Elaborado por: El investigador**



**Figura 3.74 Boceto preliminar de la sección 3 de la historia de usuario H29 Gestión de medicinas de la mascota**  
**Elaborado por: El investigador**

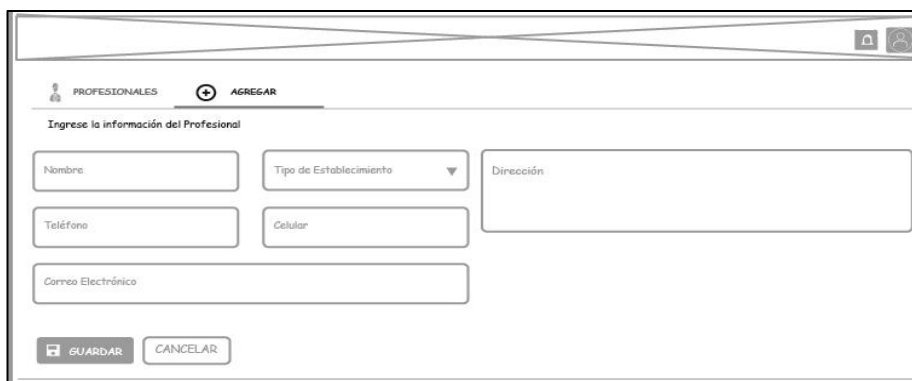
<b>Historia de Usuario</b>	
<b>Código:</b> H30	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Gestión de profesionales de la mascota	

<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota podrá visualizar todos los profesionales de su mascota, además guardar, actualizar y eliminar la información del profesional que visita su mascota.	
<b>Observación:</b> Solo el usuario dueño de la mascota podrá realizar las acciones de gestión de los profesionales de su mascota.	

**Tabla 3.39 Historia de usuario H30 Gestión de profesionales de la mascota**  
Elaborado por: El investigador



**Figura 3.75 Boceto preliminar de la sección 1 de la historia de usuario H30 Gestión de profesionales de la mascota**  
Elaborado por: El investigador



**Figura 3.76 Boceto preliminar de la sección 2 de la historia de usuario H30 Gestión de profesionales de la mascota**  
Elaborado por: El investigador



**Figura 3.77 Boceto preliminar de la sección 3 de la historia de usuario H30 Gestión de profesionales de la mascota**  
**Elaborado por: El investigador**

<b>Historia de Usuario</b>	
<b>Código:</b> H31	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Contactar profesional de la mascota	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota podrá conectar al profesional que frecuenta su mascota gracias a la información proporcionada.	
<b>Observación:</b> Podrá conectarse con el profesional que frecuenta su mascota por medio de correo electrónico o WhatsApp.	

**Tabla 3.40 Historia de usuario H31 Contactar profesional de la mascota**  
**Elaborado por: El investigador**

El boceto muestra una interfaz web con un menú superior que incluye 'PROFESIONALES' y 'CONTACTO'. Debajo, se encuentra un formulario titulado 'Contactar a Marta Sofis'. El formulario tiene un campo desplegable para 'Medio de contacto' y un campo de texto para 'Mensaje'. El mensaje de ejemplo dice: 'Saludos Marta Sofis. Soy Juan Caisapanta, quisiera separar una cita para mi perro Hashi. Mi número de celular es 0999288791. Espero su respuesta, excelente día.' Al final del formulario hay dos botones: 'ENVIAR' y 'CANCELAR'.

**Figura 3.78 Boceto preliminar de la historia de usuario H31 Contactar profesional de la mascota**  
**Elaborado por: El investigador**

<b>Historia de Usuario</b>	
<b>Código:</b> H32	<b>Usuario:</b> Dueño de la mascota
<b>Nombre de historia:</b> Consulta de mascotas por secciones	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Juan Caisapanta	
<b>Descripción:</b> El usuario dueño de la mascota podrá consultar que mascotas cumplen con la información de cada sección que administra estas son: establecimientos, medicinas y profesionales.	
<b>Observación:</b> Ninguna.	

**Tabla 3.41 Historia de usuario H32 Consulta de mascotas por secciones**  
**Elaborado por: El investigador**



**Figura 3.79 Boceto preliminar de la historia de usuario H32 Consulta de mascotas por secciones**  
Elaborado por: El investigador

### Estimación de historias de usuario

Creadas las historias de usuario se procede a estimar el tiempo que se requerirá para completar las actividades que se determinaron.

Código	Historia de Usuario	Tiempo estimado	
		Horas	Días
H01	Estructura del proyecto	56	7
H02	Diseño de la base de datos	56	7
H03	Registro de usuarios	12	3
H04	Inicio de sesión	12	3
H05	Olvido de contraseña	8	1
H06	Panel de administración	16	2
H07	Gestión de usuarios	12	3
H08	Gestión de mascotas	12	3
H09	Gestión de dispositivos	20	5
H10	Gestión de establecimientos	12	3
H11	Gestión de medicinas	12	3
H12	Gestión de profesionales	12	3
H13	Gestión de especies	4	2
H14	Gestión de razas	4	2
H15	Gestión de tipos de establecimientos	4	2
H16	Gestión de tipos de medicinas	4	2
H17	Gestión de profesiones	4	2
H18	Panel de usuario	16	2
H19	Configuración cuenta de usuario	8	2
H20	Gestión de las mascotas del usuario	8	2
H21	Gestión de los dispositivos del usuario	8	2
H22	Secciones de monitorización	8	1
H23	Ubicación de la mascota del usuario en tiempo real	16	2
H24	Historial de ubicaciones de la mascota del usuario en tiempo real	8	2
H25	Temperatura de la mascota del usuario en tiempo real	16	2

H26	Historial de temperaturas de la mascota del usuario en tiempo real	8	2
H27	Gestión de establecimientos de la mascota	15	3
H28	Contactar establecimiento de la mascota	6	2
H29	Gestión de medicinas de la mascota	15	3
H30	Gestión de profesionales de la mascota	15	3
H31	Contactar profesional de la mascota	6	2
H32	Consulta de mascotas por secciones	32	4
Tiempo estimado		445	87

**Tabla 3.42 Estimación de historias de usuario**  
Elaborado por: El investigador

### Plan de entregas

En el plan de entregas se agrupan las historias de usuario por iteraciones en las que el proyecto va a ser completado.

Iteración	Código	Historia de Usuario	Tiempo estimado	
			Horas	Días
1	H01	Estructura del proyecto	56	7
1	H02	Diseño de la base de datos	56	7
1	H03	Registro de usuarios	12	3
1	H04	Inicio de sesión	12	3
1	H05	Olvido de contraseña	8	1
2	H06	Panel de administración	16	2
2	H07	Gestión de usuarios	12	3
2	H08	Gestión de mascotas	12	3
2	H09	Gestión de dispositivos	20	5
2	H10	Gestión de establecimientos	12	3
2	H11	Gestión de medicinas	12	3
2	H12	Gestión de profesionales	12	3
2	H13	Gestión de especies	4	2
2	H14	Gestión de razas	4	2
2	H15	Gestión de tipos de establecimientos	4	2
2	H16	Gestión de tipos de medicinas	4	2
2	H17	Gestión de profesiones	4	2
3	H18	Panel de usuario	16	2
3	H19	Configuración cuenta de usuario	8	2
3	H20	Gestión de las mascotas del usuario	8	2
3	H21	Gestión de los dispositivos del usuario	8	2
4	H22	Secciones de monitorización	8	1
4	H23	Ubicación de la mascota del usuario en tiempo real	16	2
4	H24	Historial de ubicaciones de la mascota del usuario en tiempo real	8	2

4	H25	Temperatura de la mascota del usuario en tiempo real	16	2
4	H26	Historial de temperaturas de la mascota del usuario en tiempo real	8	2
4	H27	Gestión de establecimientos de la mascota	15	3
4	H28	Contactar establecimiento de la mascota	6	2
4	H29	Gestión de medicinas de la mascota	15	3
4	H30	Gestión de profesionales de la mascota	15	3
4	H31	Contactar profesional de la mascota	6	2
4	H32	Consulta de mascotas por secciones	32	4

**Tabla 3.43 Plan de entregas**  
Elaborado por: El investigador

### 3.2.3.4 Iteraciones

#### Iteración 1

Las historias de usuario que deben ser realizadas en la iteración 1 son las siguientes:

Iteración	Código	Historia de Usuario	Tiempo estimado	
			Horas	Días
1	H01	Estructura del proyecto	56	7
1	H02	Diseño de la base de datos	56	7
1	H03	Registro de usuarios	12	3
1	H04	Inicio de sesión	12	3
1	H05	Olvido de contraseña	8	1

**Tabla 3.44 Historias de usuario de la iteración 1**  
Elaborado por: El investigador

Las tareas para realizarse en la iteración 1 son las siguientes:

Tarea	
<b>Código de tarea:</b> T01	<b>Código de historia de usuario:</b> H01
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Nombre de tarea:</b> Definir estructura de carpetas para el desarrollo de los servicios web tipo REST.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 05/05/2022	<b>Fecha fin:</b> 8/05/2022
<b>Descripción:</b> Se desarrollará la estructura de carpetas que se adapte a la arquitectura de Spring Boot.	

**Tabla 3.45 Tarea T01 Definir estructura de carpetas para el desarrollo de los servicios web tipo REST**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T02	<b>Código de historia de usuario:</b> H01



<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Nombre de tarea:</b> Definir estructura de carpetas para el desarrollo de la PWA.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 09/05/2022	<b>Fecha fin:</b> 11/05/2022
<b>Descripción:</b> La estructura de carpetas para el desarrollo en Next.js se define automáticamente, pero es posible adicionar nuevas carpetas que ayuden al desarrollo y organización del proyecto.	

Tabla 3.46 Tarea T02 Definir estructura de carpetas para el desarrollo de la PWA  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T03	<b>Código de historia de usuario:</b> H02
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5
<b>Nombre de tarea:</b> Definir las entidades que intervendrán en el diseño de la base de datos.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 12/05/2022	<b>Fecha fin:</b> 18/05/2022
<b>Descripción:</b> Spring Boot permite crear las tablas de la base de datos por medio de las clases de entidades desarrolladas.	

Tabla 3.47 Tarea T03 Definir las entidades que intervendrán en el diseño de la base de datos  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T04	<b>Código de historia de usuario:</b> H03
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5
<b>Nombre de tarea:</b> Desarrollo de seguridad mediante json web token (JWT).	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 19/05/2022	<b>Fecha fin:</b> 19/05/2022
<b>Descripción:</b> En Spring Boot el método usado para la seguridad de la PWA es mediante JWT, donde un token será asignado a cada usuario registrado al momento de iniciar su sesión.	

Tabla 3.48 Tarea T04 Desarrollo de seguridad mediante json web token (JWT)  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T05	<b>Código de historia de usuario:</b> H03
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5
<b>Nombre de tarea:</b> Desarrollo del servicio web tipo REST para el registro de nuevos usuarios dueños de mascotas	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 20/05/2022	<b>Fecha fin:</b> 20/05/2022
<b>Descripción:</b> Desarrollo de clases y métodos para el registro de nuevos usuarios dueños de mascotas en Spring Boot.	

Tabla 3.49 Tarea T05 Desarrollo del servicio web tipo REST para el registro de nuevos usuarios dueños de mascotas  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T06	<b>Código de historia de usuario:</b> H03
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para el registro de nuevos usuarios dueños de mascotas.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 21/05/2022	<b>Fecha fin:</b> 21/05/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir el servicio web tipo REST de registro de nuevos usuarios dueños de mascotas en Next.js.	

Tabla 3.50 Tarea T06 Desarrollo de interfaz gráfica para el registro de nuevos usuarios dueños de mascotas

Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T07	<b>Código de historia de usuario:</b> H04
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para el inicio de sesión de los usuarios.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 22/05/2022	<b>Fecha fin:</b> 22/05/2022
<b>Descripción:</b> Desarrollo de clases y métodos para el inicio de sesión de los usuarios de la PWA en Spring Boot.	

Tabla 3.51 Tarea T07 Desarrollo de servicio web tipo REST para el inicio de sesión de los usuarios

Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T08	<b>Código de historia de usuario:</b> H04
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para el inicio de sesión de los usuarios.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 23/05/2022	<b>Fecha fin:</b> 24/05/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir el servicio web tipo REST de inicio de sesión de los usuarios de la PWA en Next.js.	

Tabla 3.52 Tarea T08 Desarrollo de interfaz gráfica para el inicio de sesión de los usuarios

Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T09	<b>Código de historia de usuario:</b> H05
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST en caso de olvido de contraseña de los usuarios dueños de mascotas.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 25/05/2022	<b>Fecha fin:</b> 25/05/2022
<b>Descripción:</b> Desarrollo de clases y métodos, además de la conexión al servidor SMTP para el envío de nueva contraseña mediante correo electrónico en caso de olvido de la contraseña de los usuarios dueños de mascotas en Spring Boot.	

**Tabla 3.53 Tarea T09 Desarrollo de servicio web tipo REST en caso de olvido de contraseña de los usuarios dueños de mascotas**  
Elaborado por: El investigador

<b>Tarea</b>	
<b>Código de tarea:</b> T10	<b>Código de historia de usuario:</b> H05
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica en caso de olvido de la contraseña de los usuarios dueños de mascotas.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 25/05/2022	<b>Fecha fin:</b> 25/05/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir el servicio web tipo REST de olvido de la contraseña de los usuarios dueños de mascotas en Next.js.	

**Tabla 3.54 Tarea T10 Desarrollo de interfaz gráfica en caso de olvido de la contraseña de los usuarios dueños de mascotas**  
Elaborado por: El investigador

## Iteración 2

Las historias de usuario que deben ser realizadas en la iteración 2 son las siguientes:

Iteración	Código	Historia de Usuario	Tiempo estimado	
			Horas	Días
2	H06	Panel de administración	16	2
2	H07	Gestión de usuarios	12	3
2	H08	Gestión de mascotas	12	3
2	H09	Gestión de dispositivos	20	5
2	H10	Gestión de establecimientos	12	3
2	H11	Gestión de medicinas	12	3
2	H12	Gestión de profesionales	12	3
2	H13	Gestión de especies	4	2
2	H14	Gestión de razas	4	2
2	H15	Gestión de tipos de establecimientos	4	2
2	H16	Gestión de tipos de medicinas	4	2
2	H17	Gestión de profesiones	4	2

**Tabla 3.55 Historias de usuario de la iteración 2**  
Elaborado por: El investigador

Las tareas para realizarse en la iteración 2 son las siguientes:

<b>Tarea</b>	
<b>Código de tarea:</b> T11	<b>Código de historia de usuario:</b> H06
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para el panel de administración.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 26/05/2022	<b>Fecha fin:</b> 27/05/2022

**Descripción:** Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para mostrar información de los datos ingresados en la PWA en Next.js.

**Tabla 3.56 Tarea T11 Desarrollo de interfaz gráfica para el panel de administración**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T12	<b>Código de historia de usuario:</b> H07
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para la gestión de los usuarios.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 28/05/2022	<b>Fecha fin:</b> 28/05/2022
<b>Descripción:</b> Desarrollo de clases y métodos para la gestión de los usuarios de la PWA en Spring Boot.	

**Tabla 3.57 Tarea T12 Desarrollo de servicio web tipo REST para la gestión de los usuarios**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T13	<b>Código de historia de usuario:</b> H07
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de los usuarios.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 29/05/2022	<b>Fecha fin:</b> 30/05/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST de gestión de usuarios en Next.js.	

**Tabla 3.58 Tarea T13 Desarrollo de interfaz gráfica para la gestión de los usuarios**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T14	<b>Código de historia de usuario:</b> H08
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para la gestión de mascotas.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 31/05/2022	<b>Fecha fin:</b> 31/05/2022
<b>Descripción:</b> Desarrollo de clases y métodos para la gestión de mascotas en Spring Boot.	

**Tabla 3.59 Tarea T14 Desarrollo de servicio web tipo REST para la gestión de mascotas.**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T15	<b>Código de historia de usuario:</b> H08
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de mascotas.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 01/06/2022	<b>Fecha fin:</b> 02/06/2022

**Descripción:** Desarrollo de componentes y métodos para consumir los servicios web tipo REST de gestión de mascotas en Next.js.

**Tabla 3.60 Tarea T15 Desarrollo de interfaz gráfica para la gestión de mascotas**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T16	<b>Código de historia de usuario:</b> H09
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para la gestión de dispositivos.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 03/06/2022	<b>Fecha fin:</b> 03/06/2022
<b>Descripción:</b> Desarrollo de clases y métodos para la gestión de dispositivos en Spring Boot.	

**Tabla 3.61 Tarea T16 Desarrollo de servicio web tipo REST para la gestión de dispositivos**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T17	<b>Código de historia de usuario:</b> H09
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para obtener datos enviados desde los dispositivos.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 04/06/2022	<b>Fecha fin:</b> 04/06/2022
<b>Descripción:</b> Desarrollo de clases y métodos para obtener los datos enviados desde los dispositivos en Spring Boot.	

**Tabla 3.62 Tarea T17 Desarrollo de servicio web tipo REST para obtener datos enviados desde los dispositivos**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T18	<b>Código de historia de usuario:</b> H09
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de los dispositivos.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 05/06/2022	<b>Fecha fin:</b> 07/06/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST de gestión de los dispositivos en Next.js.	

**Tabla 3.63 Tarea T18 Desarrollo de interfaz gráfica para la gestión de los dispositivos**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T19	<b>Código de historia de usuario:</b> H10
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para la gestión de establecimientos.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 08/06/2022	<b>Fecha fin:</b> 08/06/2022

**Descripción:** Desarrollo de clases y métodos para la gestión de establecimientos en Spring Boot.

**Tabla 3.64 Tarea T19 Desarrollo de servicio web tipo REST para la gestión de establecimientos**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T20	<b>Código de historia de usuario:</b> H10
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de establecimientos.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 09/06/2022	<b>Fecha fin:</b> 10/06/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST de gestión de establecimientos en Next.js.	

**Tabla 3.65 Tarea T20 Desarrollo de interfaz gráfica para la gestión de establecimientos**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T21	<b>Código de historia de usuario:</b> H11
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para la gestión de medicinas.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 11/06/2022	<b>Fecha fin:</b> 11/06/2022
<b>Descripción:</b> Desarrollo de clases y métodos para la gestión de medicinas en Spring Boot.	

**Tabla 3.66 Tarea T21 Desarrollo de servicio web tipo REST para la gestión de medicinas**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T22	<b>Código de historia de usuario:</b> H11
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de medicinas.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 12/06/2022	<b>Fecha fin:</b> 13/06/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST de gestión de medicinas en Next.js.	

**Tabla 3.67 Tarea T22 Desarrollo de interfaz gráfica para la gestión de medicinas**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T23	<b>Código de historia de usuario:</b> H12
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para la gestión de profesionales.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 14/06/2022	<b>Fecha fin:</b> 14/06/2022
<b>Descripción:</b> Desarrollo de clases y métodos para la gestión de profesionales en Spring Boot.	

Tabla 3.68 Tarea T23 Desarrollo de servicio web tipo REST para la gestión de profesionales  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T24	<b>Código de historia de usuario:</b> H12
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de profesionales.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 15/06/2022	<b>Fecha fin:</b> 16/06/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST de gestión de profesionales en Next.js.	

Tabla 3.69 Tarea T24 Desarrollo de interfaz gráfica para la gestión de profesionales  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T25	<b>Código de historia de usuario:</b> H13
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para la gestión de especies.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 17/06/2022	<b>Fecha fin:</b> 17/06/2022
<b>Descripción:</b> Desarrollo de clases y métodos para la gestión de especies en Spring Boot.	

Tabla 3.70 Tarea T25 Desarrollo de servicio web tipo REST para la gestión de especies  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T26	<b>Código de historia de usuario:</b> H13
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de especies.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 18/06/2022	<b>Fecha fin:</b> 18/06/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST de gestión de especies en Next.js.	

Tabla 3.71 Tarea T26 Desarrollo de interfaz gráfica para la gestión de especies  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T27	<b>Código de historia de usuario:</b> H14
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio para la gestión de razas.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 19/06/2022	<b>Fecha fin:</b> 19/06/2022
<b>Descripción:</b> Desarrollo de clases y métodos para la gestión de razas en Spring Boot.	

Tabla 3.72 Tarea T27 Desarrollo de servicio para la gestión de razas  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T28	<b>Código de historia de usuario:</b> H14
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de razas.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 20/06/2022	<b>Fecha fin:</b> 20/06/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST de gestión de razas en Next.js.	

Tabla 3.73 Tarea T28 Desarrollo de interfaz gráfica para la gestión de razas  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T29	<b>Código de historia de usuario:</b> H15
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para la gestión de tipos de establecimientos.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 21/06/2022	<b>Fecha fin:</b> 21/06/2022
<b>Descripción:</b> Desarrollo de clases y métodos para la gestión de tipos de establecimientos en Spring Boot.	

Tabla 3.74 Tarea T29 Desarrollo de servicio web tipo REST para la gestión de tipos de establecimientos  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T30	<b>Código de historia de usuario:</b> H15
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de tipos de establecimientos.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 22/06/2022	<b>Fecha fin:</b> 22/06/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST de gestión de tipos de establecimientos en Next.js.	

Tabla 3.75 Tarea T30 Desarrollo de interfaz gráfica para la gestión de tipos de establecimientos  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T31	<b>Código de historia de usuario:</b> H16
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para la gestión de tipos de medicinas.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 23/06/2022	<b>Fecha fin:</b> 23/06/2022
<b>Descripción:</b> Desarrollo de clases y métodos para la gestión de tipos de medicinas en Spring Boot.	

Tabla 3.76 Tarea T31 Desarrollo de servicio web tipo REST para la gestión de tipos de medicinas  
Elaborado por: El investigador



Tarea	
<b>Código de tarea:</b> T32	<b>Código de historia de usuario:</b> H16
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de tipos de medicinas.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 24/06/2022	<b>Fecha fin:</b> 24/06/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST de gestión de tipos de medicinas en Next.js.	

Tabla 3.77 Tarea T32 Desarrollo de interfaz gráfica para la gestión de tipos de medicinas  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T33	<b>Código de historia de usuario:</b> H17
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para la gestión de profesiones.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 25/06/2022	<b>Fecha fin:</b> 25/06/2022
<b>Descripción:</b> Desarrollo de clases y métodos para la gestión de profesiones en Spring Boot.	

Tabla 3.78 Tarea T33 Desarrollo de servicio web tipo REST para la gestión de profesiones  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T34	<b>Código de historia de usuario:</b> H17
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de profesiones.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 26/06/2022	<b>Fecha fin:</b> 26/06/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST de gestión de profesiones en Next.js.	

Tabla 3.79 Tarea T34 Desarrollo de interfaz gráfica para la gestión de profesiones  
Elaborado por: El investigador

### Iteración 3

Las historias de usuario que deben ser realizadas en la iteración 3 son las siguientes:

Iteración	Código	Historia de Usuario	Tiempo estimado	
			Horas	Días
3	H18	Panel de usuario	16	2
3	H19	Configuración cuenta de usuario	8	2
3	H20	Gestión de las mascotas del usuario	8	2
3	H21	Gestión de los dispositivos del usuario	8	2

Tabla 3.80 Historias de usuario de la iteración 3  
Elaborado por: El investigador

Las tareas para realizarse en la iteración 3 son las siguientes:

Tarea	
<b>Código de tarea:</b> T35	<b>Código de historia de usuario:</b> H18
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para el panel de usuario.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 27/06/2022	<b>Fecha fin:</b> 28/06/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para mostrar información de los datos ingresados por el usuario dueño de la mascota en Next.js.	

Tabla 3.81 Tarea T35 Desarrollo de interfaz gráfica para el panel de usuario  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T36	<b>Código de historia de usuario:</b> H19
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la configuración de la cuenta del usuario.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 29/06/2022	<b>Fecha fin:</b> 30/06/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para la configuración de la cuenta del usuario dueño de la mascota en Next.js.	

Tabla 3.82 Tarea T36 Desarrollo de interfaz gráfica para la configuración de la cuenta del usuario  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T37	<b>Código de historia de usuario:</b> H20
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de las mascotas del usuario.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 01/07/2022	<b>Fecha fin:</b> 02/07/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para la gestión de las mascotas del usuario en Next.js.	

Tabla 3.83 Tarea T37 Desarrollo de interfaz gráfica para la gestión de las mascotas del usuario  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T38	<b>Código de historia de usuario:</b> H21
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de los dispositivos del usuario.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 03/07/2022	<b>Fecha fin:</b> 04/07/2022

**Descripción:** Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para la gestión de los dispositivos del usuario en Next.js.

**Tabla 3.84 Tarea T38 Desarrollo de interfaz gráfica para la gestión de los dispositivos del usuario**

Elaborado por: El investigador

#### Iteración 4

Las historias de usuario que deben ser realizadas en la iteración 4 son las siguientes:

Iteración	Código	Historia de Usuario	Tiempo estimado	
			Horas	Días
4	H22	Secciones de monitorización	8	1
4	H23	Ubicación de la mascota del usuario en tiempo real	16	2
4	H24	Historial de ubicaciones de la mascota del usuario en tiempo real	8	2
4	H25	Temperatura de la mascota del usuario en tiempo real	16	2
4	H26	Historial de temperaturas de la mascota del usuario en tiempo real	8	2
4	H27	Gestión de establecimientos de la mascota	15	3
4	H28	Contactar establecimiento de la mascota	6	2
4	H29	Gestión de medicinas de la mascota	15	3
4	H30	Gestión de profesionales de la mascota	15	3
4	H31	Contactar profesional de la mascota	6	2
4	H32	Consulta de mascotas por secciones	32	4

**Tabla 3.85 Historias de usuario de la iteración 4**

Elaborado por: El investigador

Las tareas para realizarse en la iteración 4 son las siguientes:

Tarea	
<b>Código de tarea:</b> T39	<b>Código de historia de usuario:</b> H22
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para mostrar las secciones de monitorización.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 05/07/2022	<b>Fecha fin:</b> 05/07/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para mostrar las secciones de monitorización en Next.js.	

**Tabla 3.86 Tarea T39 Desarrollo de interfaz gráfica para mostrar las secciones de monitorización**

Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T40	<b>Código de historia de usuario:</b> H23
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para mostrar la ubicación de la mascota del usuario en tiempo real.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 06/07/2022	<b>Fecha fin:</b> 07/07/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para mostrar la ubicación de la mascota del usuario en tiempo real en Next.js.	

Tabla 3.87 Tarea T40 Desarrollo de interfaz gráfica para mostrar la ubicación de la mascota del usuario en tiempo real

Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T41	<b>Código de historia de usuario:</b> H24
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para mostrar el historial de ubicaciones de la mascota del usuario en tiempo real.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 08/07/2022	<b>Fecha fin:</b> 09/07/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para mostrar el historial de ubicaciones de la mascota del usuario en tiempo real. en Next.js.	

Tabla 3.88 Tarea T41 Desarrollo de interfaz gráfica para mostrar el historial de ubicaciones de la mascota del usuario en tiempo real

Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T42	<b>Código de historia de usuario:</b> H25
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para mostrar la temperatura de la mascota del usuario en tiempo real.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 10/07/2022	<b>Fecha fin:</b> 11/07/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para mostrar la temperatura de la mascota del usuario en tiempo real en Next.js.	

Tabla 3.89 Tarea T42 Desarrollo de interfaz gráfica para mostrar la temperatura de la mascota del usuario en tiempo real

Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T43	<b>Código de historia de usuario:</b> H26
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para mostrar el historial de temperaturas de la mascota del usuario en tiempo real.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 12/07/2022	<b>Fecha fin:</b> 13/07/2022

**Descripción:** Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para mostrar el historial de temperaturas de la mascota del usuario en tiempo real. en Next.js.

**Tabla 3.90 Tarea T43 Desarrollo de interfaz gráfica para mostrar el historial de temperaturas de la mascota del usuario en tiempo real**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T44	<b>Código de historia de usuario:</b> H27
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de establecimientos de la mascota.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 14/07/2022	<b>Fecha fin:</b> 16/07/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para la gestión de establecimientos de la mascota en Next.js.	

**Tabla 3.91 Tarea T44 Desarrollo de interfaz gráfica para la gestión de establecimientos de la mascota**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T45	<b>Código de historia de usuario:</b> H28
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para contactar el establecimiento de la mascota.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 17/07/2022	<b>Fecha fin:</b> 17/07/2022
<b>Descripción:</b> Desarrollo de clases y métodos, además de la conexión a servidor SMTP para él envío de correo electrónico al establecimiento de la mascota en Spring Boot.	

**Tabla 3.92 Tarea T45 Desarrollo de servicio web tipo REST para contactar el establecimiento de la mascota**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T46	<b>Código de historia de usuario:</b> H28
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para contactar el establecimiento de la mascota.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 18/07/2022	<b>Fecha fin:</b> 18/07/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para contactar el establecimiento de la mascota. en Next.js.	

**Tabla 3.93 Tarea T46 Desarrollo de interfaz gráfica para contactar el establecimiento de la mascota**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T47	<b>Código de historia de usuario:</b> H29
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de medicinas de la mascota.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 19/07/2022	<b>Fecha fin:</b> 21/07/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para la gestión de medicinas de la mascota en Next.js.	

Tabla 3.94 Tarea T47 Desarrollo de interfaz gráfica para la gestión de medicinas de la mascota  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T48	<b>Código de historia de usuario:</b> H30
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la gestión de profesionales de la mascota.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 22/07/2022	<b>Fecha fin:</b> 24/07/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para la gestión de profesionales de la mascota en Next.js.	

Tabla 3.95 Tarea T48 Desarrollo de interfaz gráfica para la gestión de profesionales de la mascota

Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T49	<b>Código de historia de usuario:</b> H31
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para contactar al profesional de la mascota.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 25/07/2022	<b>Fecha fin:</b> 25/07/2022
<b>Descripción:</b> Desarrollo de clases y métodos, además de la conexión a servidor SMTP para el envío de correo electrónico al profesional de la mascota en Spring Boot.	

Tabla 3.96 Tarea T49 Desarrollo de servicio web tipo REST para contactar al profesional de la mascota

Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T50	<b>Código de historia de usuario:</b> H31
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para contactar al profesional de la mascota.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 26/07/2022	<b>Fecha fin:</b> 26/07/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para contactar al profesional de la mascota en Next.js.	

**Tabla 3.97 Tarea T50 Desarrollo de interfaz gráfica para contactar al profesional de la mascota**  
Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T51	<b>Código de historia de usuario:</b> H32
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de servicio web tipo REST para la consulta de mascotas por secciones.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 27/07/2022	<b>Fecha fin:</b> 28/07/2022
<b>Descripción:</b> Desarrollo de clases y métodos para la consulta de mascotas por secciones. en Spring Boot.	

**Tabla 3.98 Tarea T51 Desarrollo de servicio web tipo REST para la consulta de mascotas por secciones**

Elaborado por: El investigador

Tarea	
<b>Código de tarea:</b> T52	<b>Código de historia de usuario:</b> H32
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Nombre de tarea:</b> Desarrollo de interfaz gráfica para la consulta de mascotas por secciones.	
<b>Programador responsable:</b> Juan Caisapanta	
<b>Fecha inicio:</b> 29/07/2022	<b>Fecha fin:</b> 30/07/2022
<b>Descripción:</b> Desarrollo de componentes y métodos para consumir los servicios web tipo REST necesarios para consulta de mascotas por secciones en Next.js.	

**Tabla 3.99 Tarea T52 Desarrollo de interfaz gráfica para la consulta de mascotas por secciones**  
Elaborado por: El investigador

### 3.2.3.5 Fase de diseño

La fase de diseño es la segunda fase de la metodología XP, en esta fase se realiza el diseño de acuerdo con las historias de usuario creadas, donde se usarán las tarjetas de clase, responsabilidad y colaboración (CRC), las cuales definen las responsabilidades de una clase y los colaboradores que participan.

#### Tarjetas CRC

Clase: Autenticación	
Responsabilidad	Colaboradores
Inicio de sesión Registro de usuario Olvido de contraseña Validación de token	Back-end: Entidad usuario Entidad rol Repositorio usuario Servicio usuario Controlador autenticación

	Front-end: React context autentificacion Librería axios Librería Material UI
<b>Observaciones:</b> El proceso de seguridad se realiza a través de JWT y para el proceso de olvido de contraseña se usa un correo electrónico enviado desde un servidor SMTP.	

**Tabla 3.100 Tarjeta CRC Clase Autentificación**  
 Elaborado por: El investigador

<b>Clase:</b> Notificación	
Responsabilidad	Colaboradores
Obtener notificaciones por usuario Eliminar notificación	Back-end: Entidad notificación Repositorio notificación Servicio notificación Controlador notificación
	Front-end: React context notificación Librería axios Librería Material UI
<b>Observaciones:</b> Ninguna.	

**Tabla 3.101 Tarjeta CRC Clase Notificación**  
 Elaborado por: El investigador

<b>Clase:</b> Usuario	
Responsabilidad	Colaboradores
Obtener usuarios Obtener usuario por correo electrónico Obtener usuario por id Actualizar usuario Actualizar contraseña Eliminar usuario	Back-end: Entidad usuario Repositorio usuario Servicio usuario Controlador usuario
	Front-end: React context usuario Librería axios Librería Material UI
<b>Observaciones:</b> Ninguna.	

**Tabla 3.102 Tarjeta CRC Clase Usuario**  
 Elaborado por: El investigador

<b>Clase:</b> Mascota	
Responsabilidad	Colaboradores
Obtener mascotas Obtener mascotas por id Obtener mascotas por especies Crear mascota Actualizar mascota	Back-end: Entidad mascota Repositorio mascota Servicio mascota Controlador mascota



Eliminar mascota	Front-end: React context mascota Librería axios Librería Material UI
<b>Observaciones:</b> Ninguna.	

**Tabla 3.103 Tarjeta CRC Clase Mascota**  
Elaborado por: El investigador

<b>Clase:</b> Dispositivo	
Responsabilidad	Colaboradores
Obtener dispositivos Obtener dispositivo por id Crear dispositivo Actualizar dispositivo Eliminar dispositivo	Back-end: Entidad dispositivo Repositorio dispositivo Servicio dispositivo Controlador dispositivo
	Front-end: React context dispositivo Librería axios Librería Material UI
<b>Observaciones:</b> Ninguna.	

**Tabla 3.104 Tarjeta CRC Clase Dispositivo**  
Elaborado por: El investigador

<b>Clase:</b> Detalle Dispositivo	
Responsabilidad	Colaboradores
Obtener detalles de los dispositivos Obtener detalle por dispositivo Crear detalle de un dispositivo Actualizar dispositivo Eliminar dispositivo	Back-end: Entidad dispositivo Entidad detalle dispositivo Entidad usuario Entidad notificación Repositorio detalle dispositivo Servicio detalle dispositivo Controlador detalle dispositivo
	Front-end: React context dispositivo Librería axios Librería Material UI
<b>Observaciones:</b> Los detalles de los dispositivos son los datos enviados por el dispositivo y guardados en la base de datos, al implementar la clase de notificación es posible realizar envíos de avisos por medio de correo electrónico al dueño de la mascota por si un dato enviado e ingresado tiene información útil, por ejemplo, si la mascota tiene temperatura baja o el dispositivo tiene la batería baja.	

**Tabla 3.105 Tarjeta CRC Clase Detalle Dispositivo**  
Elaborado por: El investigador

<b>Clase:</b> Establecimiento	
Responsabilidad	Colaboradores

Obtener establecimientos	Back-end:
Obtener establecimiento por id	Entidad establecimiento
Obtener establecimiento por nombre	Repositorio establecimiento
Obtener establecimiento por mascota	Servicio establecimiento
Crear establecimiento	Controlador establecimiento
Actualizar establecimiento	Front-end:
Eliminar establecimiento	React context establecimiento
Enviar correo electrónico al establecimiento	Librería axios
	Librería Material UI
<b>Observaciones:</b> El envío de correo electrónico al establecimiento se lo realiza mediante un servidor SMTP	

**Tabla 3.106 Tarjeta CRC Clase Establecimiento**  
Elaborado por: El investigador

<b>Clase:</b> Medicina	
<b>Responsabilidad</b>	<b>Colaboradores</b>
Obtener medicinas	Back-end:
Obtener medicina por id	Entidad medicina
Obtener medicina por nombre	Repositorio medicina
Obtener medicina por mascota	Servicio medicina
Crear medicina	Controlador medicina
Actualizar medicina	Front-end:
Eliminar medicina	React context medicina
	Librería axios
	Librería Material UI
<b>Observaciones:</b> Ninguna.	

**Tabla 3.107 Tarjeta CRC Clase Medicina**  
Elaborado por: El investigador

<b>Clase:</b> Profesional	
<b>Responsabilidad</b>	<b>Colaboradores</b>
Obtener profesionales	Back-end:
Obtener profesional por id	Entidad profesional
Obtener profesional por nombre	Repositorio profesional
Obtener profesional por mascota	Servicio profesional
Crear profesional	Controlador profesional
Actualizar profesional	Front-end:
Eliminar profesional	React context profesional
Enviar correo electrónico al profesional	Librería axios
	Librería Material UI
<b>Observaciones:</b> El envío de correo electrónico al profesional se lo realiza mediante un servidor SMTP.	

**Tabla 3.108 Tarjeta CRC Clase Profesional**  
Elaborado por: El investigador

<b>Clase:</b> Especie	
<b>Responsabilidad</b>	<b>Colaboradores</b>

Obtener especies Obtener ultima especie Obtener especie por id Crear especie Actualizar especie Eliminar especie	Back-end: Entidad especie Repositorio especie Servicio especie Controlador especie
	Front-end: React context especie Librería axios Librería Material UI
<b>Observaciones:</b> Ninguna.	

**Tabla 3.109 Tarjeta CRC Clase Especie**  
**Elaborado por: El investigador**

<b>Clase: Raza</b>	
<b>Responsabilidad</b>	<b>Colaboradores</b>
Obtener razas Obtener raza por especie Obtener raza por id Crear raza Actualizar raza Eliminar raza	Back-end: Entidad raza Repositorio raza Servicio raza Controlador raza
	Front-end: React context raza Librería axios Librería Material UI
<b>Observaciones:</b> Ninguna.	

**Tabla 3.110 Tarjeta CRC Clase Raza**  
**Elaborado por: El investigador**

<b>Clase: Tipo de Establecimiento</b>	
<b>Responsabilidad</b>	<b>Colaboradores</b>
Obtener tipos de establecimientos Obtener tipo de establecimiento por id Crear tipo de establecimiento Actualizar tipo de establecimiento Eliminar tipo de establecimiento	Back-end: Entidad tipo de establecimiento Repositorio tipo de establecimiento Servicio tipo de establecimiento Controlador tipo de establecimiento
	Front-end: React context tipo de establecimiento Librería axios Librería Material UI
<b>Observaciones:</b> Ninguna.	

**Tabla 3.111 Tarjeta CRC Clase Tipo de Establecimiento**  
**Elaborado por: El investigador**

<b>Clase: Tipo de Medicina</b>	
<b>Responsabilidad</b>	<b>Colaboradores</b>
Obtener tipos de medicinas Obtener tipo de medicina por id Crear tipo de medicina Actualizar tipo de medicina	Back-end: Entidad tipo de medicina Repositorio tipo de medicina Servicio tipo de medicina

Eliminar tipo de medicina	Controlador tipo de medicina
	Front-end: React context tipo de medicina Librería axios Librería Material UI
<b>Observaciones:</b> Ninguna.	

**Tabla 3.112 Tarjeta CRC Clase Tipo de Medicina**  
Elaborado por: El investigador

<b>Clase: Profesión</b>	
<b>Responsabilidad</b>	<b>Colaboradores</b>
Obtener profesiones Obtener profesión por id Crear profesión Actualizar profesión Eliminar profesión	Back-end: Entidad profesión Repositorio profesión Servicio profesión Controlador profesión
	Front-end: React context profesión Librería axios Librería Material UI
<b>Observaciones:</b> Ninguna.	

**Tabla 3.113 Tarjeta CRC Clase Profesión**  
Elaborado por: El investigador

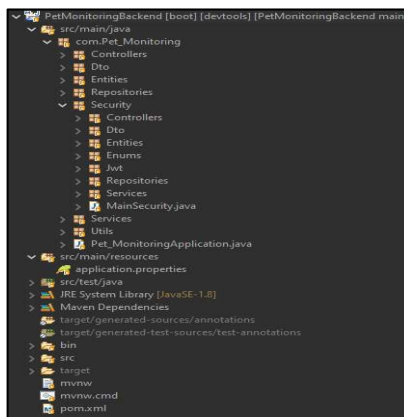
## Diseño de iteraciones

### Diseño iteración 1

### Historia de usuario H01 Estructura del proyecto

### Back-end Spring Boot

La estructura de carpetas a usarse para el desarrollo de los servicios web tipo REST que serán consumidos por la PWA son los siguientes:



### Figura 3.80 Estructura de carpetas para el desarrollo de servicios web tipo REST en Spring Boot

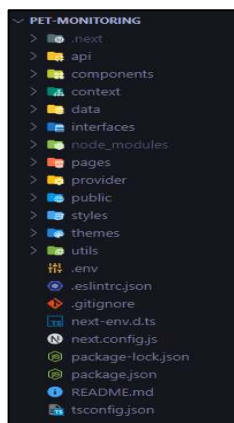
Elaborado por: El investigador

En la figura 3.80 se observa la estructura de carpetas para el desarrollo de los servicios web tipo REST, donde en:

- Security: Se gestiona la seguridad de la aplicación mediante JWT y Spring Security.
- Entities: Se determinan las clases que serán usadas para la creación de las tablas en la base de datos.
- Repositories: Se gestionan las clases donde se implementan las funciones CRUD.
- Services: Se instancian las clases de la carpeta Repositories
- Controller: Se usan los métodos de las clases de carpeta Services para desarrollar los servicios web tipo REST.
- application.properties: En este archivo se editan las configuraciones necesarias para la conexión a la base de datos y la conexión al servidor SMTP se lo realiza en el archivo.

### Front-end PWA MonIOpeT Next.js

Next.js al ser un framework proporciona una estructura determinada para usarse al crear un proyecto, pero es posible adicionar nuevas carpetas para que el desarrollo del proyecto sea más personalizado y sencillo para el desarrollador, la estructura para el desarrollo de la PWA es la siguiente:



**Figura 3.81 Estructura de carpetas para el desarrollo de la PWA MonIOpeT en Next.js**  
**Elaborado por: El investigador**

En la figura 3.81 se observa la estructura de carpetas para el desarrollo de la PWA en Next.js, donde en:

- **Api:** Se desarrollan las URLs a ser consumidas por la PWA.
- **Components:** Se gestionan los componentes de React.js.
- **Context:** Se gestionan las funciones globales a ser usadas por los componentes de React.js.
- **Data:** Se desarrollan las listas de objetos.
- **Interfaces:** Se desarrollan las interfaces a ser usadas para los objetos que son consumidos de los servicios web tipo REST.
- **Pages:** Se gestionan las rutas mediante la creación de carpetas y archivos y así haciendo más sencillo el enrutamiento de la PWA.
- **Provider:** Se gestionan los componentes que entregan las variables globales.
- **Public:** Se gestionan archivos estáticos y es donde se agrega el archivo `manifest.json` que es importante para el desarrollo de la PWA.
- **Styles:** Se gestionan los archivos globales para los estilos de la PWA.
- **Themes:** Se gestionan los temas de diseño que se pueden usar gracias a librerías de diseño de interfaz de usuario como Material UI.

## Historia de usuario H02 Diseño de la base de datos

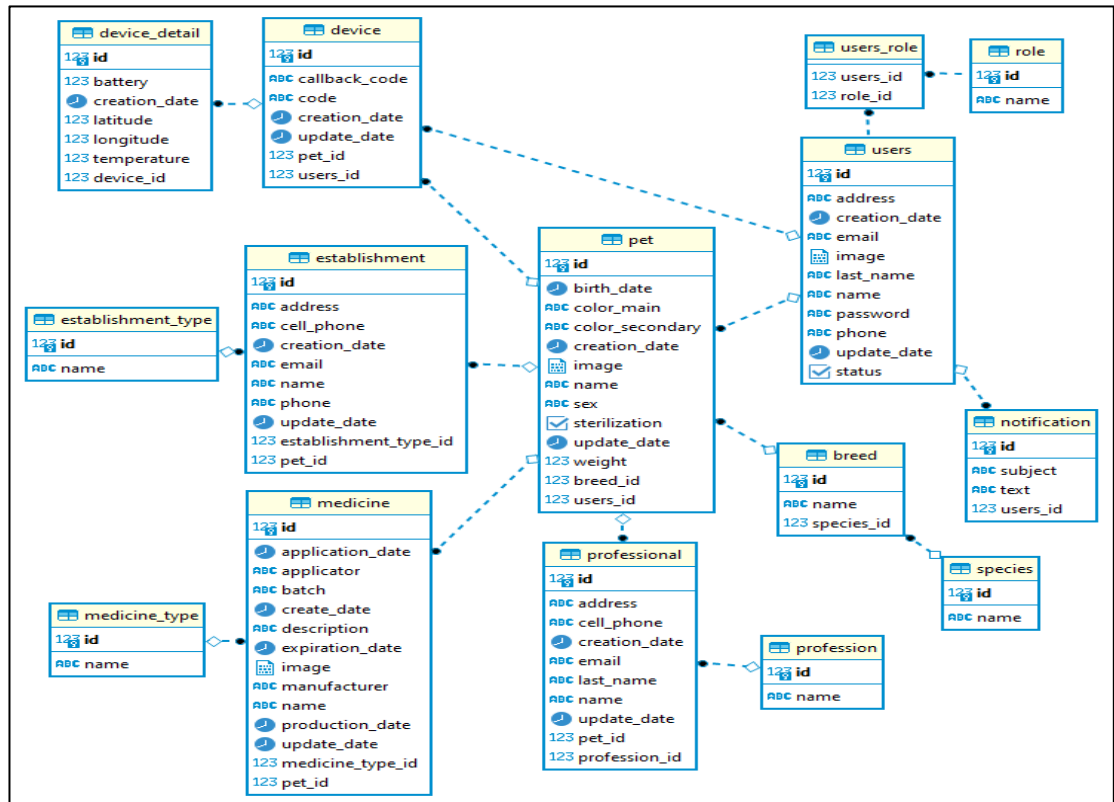


Figura 3.82 Esquema de la base de datos para la PWA MonIOpeT  
Elaborado por: El investigador

En la figura 3.82 se observa el esquema de la base de datos proporcionado por el software administrador de base de datos DBeaver, donde la base de datos está realizada en PostgreSQL.

## Historia de usuario H03 Registro de usuarios

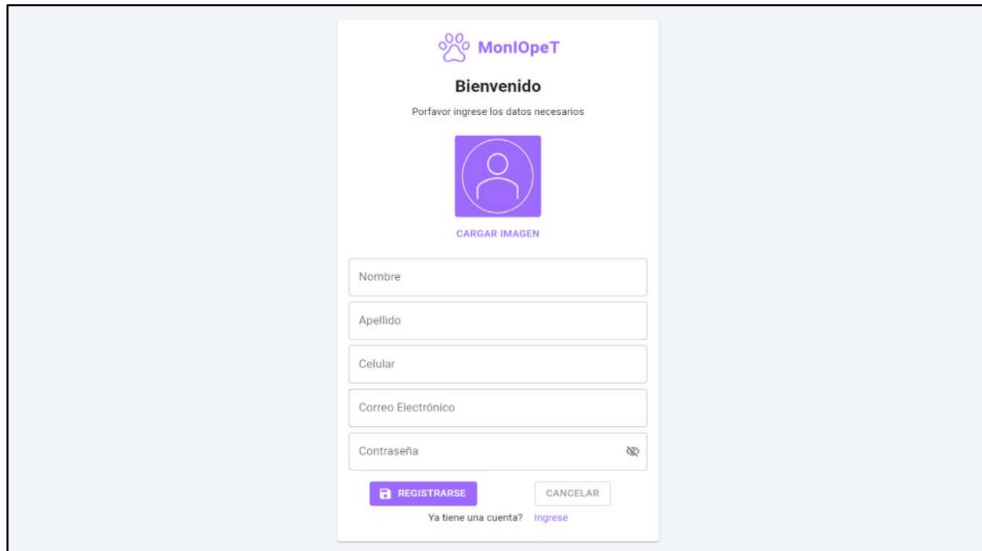
Para la facilidad de desarrollo del diseño de las interfaces de usuario de la PWA se usa la biblioteca Material UI, ya que proporciona componentes realizados en React.js compatibles con diseños de dispositivos móviles, los cuales dan la posibilidad de modificarlos para obtener un estilo propio.

Ya que la PWA funciona en distintos dispositivos se presentará la vista de computadora y la vista en dispositivo móvil.

La interfaz del registro de usuarios permitirá que nuevos usuarios dueños de mascotas creen su cuenta de usuario proporcionando un nombre, apellido, número de celular, correo electrónico y contraseña.

Las vistas en computadora y dispositivo móvil del formulario de registro de nuevos usuarios dueños de mascotas son las siguientes:

### Vista en computadora



The screenshot shows a desktop view of the registration form for 'MonIopeT'. The form is centered on a light blue background. At the top, there is a purple paw print icon followed by the text 'MonIopeT'. Below this is the heading 'Bienvenido' and the instruction 'Porfavor ingrese los datos necesarios'. A purple square icon with a white person silhouette is shown, with the text 'CARGAR IMAGEN' underneath. The form contains five input fields: 'Nombre', 'Apellido', 'Celular', 'Correo Electrónico', and 'Contraseña' (with a password strength indicator icon). At the bottom, there are two buttons: a purple 'REGISTRARSE' button and a white 'CANCELAR' button. Below the buttons, there is a link: 'Ya tiene una cuenta? Ingrese'.

Figura 3.83 Vista en computadora del formulario de registro de nuevos usuarios dueños de mascotas

Elaborado por: El investigador

### Vista en dispositivo móvil



The screenshot shows a mobile view of the registration form for 'MonIopeT'. The form is centered on a light blue background. At the top, there is a purple paw print icon followed by the text 'MonIopeT'. Below this is the heading 'Bienvenido' and the instruction 'Porfavor ingrese los datos necesarios'. A purple square icon with a white person silhouette is shown, with the text 'CARGAR IMAGEN' underneath. The form contains five input fields: 'Nombre', 'Apellido', 'Celular', 'Correo Electrónico', and 'Contraseña' (with a password strength indicator icon). At the bottom, there are two buttons: a purple 'REGISTRARSE' button and a white 'CANCELAR' button. Below the buttons, there is a link: 'Ya tiene una cuenta? Ingrese'.

Figura 3.84 Vista en dispositivo móvil del formulario de registro de nuevos usuarios dueños de mascotas

Elaborado por: El investigador



## Historia de usuario H04 Inicio de sesión

El inicio de sesión permitirá al usuario acceder a su cuenta de usuario una vez registrado para gestionar las funciones de la PWA dependiendo de su perfil, ya sea usuario o administrador proporcionando su correo electrónico y su contraseña.

Las vistas en computadora y dispositivo móvil del formulario de inicio de sesión de los usuarios son las siguientes:

### Vista en computadora

Esta imagen muestra la interfaz de usuario para el inicio de sesión en una computadora. El formulario está centrado y contiene el logo 'MoniOpeT' con un ícono de pata. Debajo del logo, se lee 'Bienvenido' y 'Por favor ingrese con su correo electrónico y contraseña'. Hay dos campos de entrada: 'Correo Electrónico' y 'Contraseña' (con un ícono de ojo para alternar visibilidad). Debajo de los campos, hay un enlace 'Olvido su contraseña?' y un botón de 'INICIAR SESIÓN' con una flecha. En la parte inferior, se encuentra el texto 'Nuevo en la plataforma?' seguido de un enlace 'Regístrate'.

Figura 3.85 Vista en computadora del formulario de inicio de sesión de los usuarios  
Elaborado por: El investigador

### Vista en dispositivo móvil

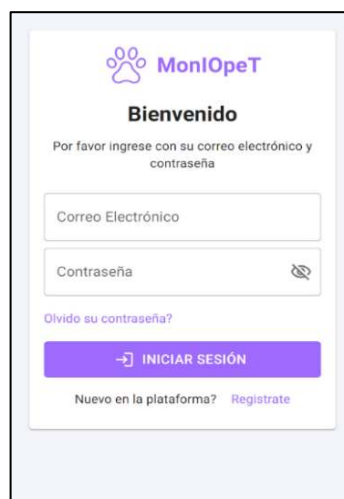
Esta imagen muestra la interfaz de usuario para el inicio de sesión en un dispositivo móvil. El formulario está adaptado al tamaño de la pantalla y contiene el mismo contenido que la versión de computadora: el logo 'MoniOpeT', el saludo 'Bienvenido', la instrucción de ingreso, los campos de 'Correo Electrónico' y 'Contraseña', el enlace 'Olvido su contraseña?', el botón 'INICIAR SESIÓN' y el enlace 'Regístrate'.

Figura 3.86 Vista en dispositivo móvil del formulario de inicio de sesión de los usuarios  
Elaborado por: El investigador

## Historia de usuario H05 Olvido de contraseña

El formulario de la interfaz de olvido de contraseña servirá para que el usuario pueda recuperar su contraseña al olvidarla, será necesario ingresar el correo electrónico con

el cual el usuario se registró en la PWA y su nueva contraseña será enviada a su correo electrónico.

Las vistas en computadora y dispositivo móvil del formulario de olvido de contraseña son las siguientes:

### Vista en computadora



The screenshot shows a desktop view of the password recovery form. At the top, there is a purple paw print icon followed by the text 'MoniOpeT'. Below this is the heading 'Bienvenido' and the instruction 'Ingrese su correo electrónico para obtener su contraseña'. A text input field labeled 'Correo Electrónico' is positioned below the instruction. Underneath the input field are two buttons: a purple button with a right-pointing arrow and the text 'SOLICITAR', and a light gray button with the text 'CANCELAR'. At the bottom of the form, there is a link that says 'Nuevo en la plataforma? [Regístrate](#)'.

**Figura 3.87 Vista en computadora del formulario de olvido de contraseña de los usuarios.  
Elaborado por: El investigador**

### Vista en dispositivo móvil



The screenshot shows a mobile view of the password recovery form. The layout is vertically centered and scaled to fit the screen. It features the same purple paw print icon and 'MoniOpeT' text at the top, followed by 'Bienvenido' and the instruction 'Ingrese su correo electrónico para obtener su contraseña'. The 'Correo Electrónico' input field is centered below the instruction. Below the input field are the 'SOLICITAR' and 'CANCELAR' buttons, with the 'SOLICITAR' button being purple and the 'CANCELAR' button being light gray. At the bottom, the link 'Nuevo en la plataforma? [Regístrate](#)' is also centered.

**Figura 3.88 Vista en dispositivo móvil del formulario de olvido de contraseña de los usuarios.  
Elaborado por: El investigador**

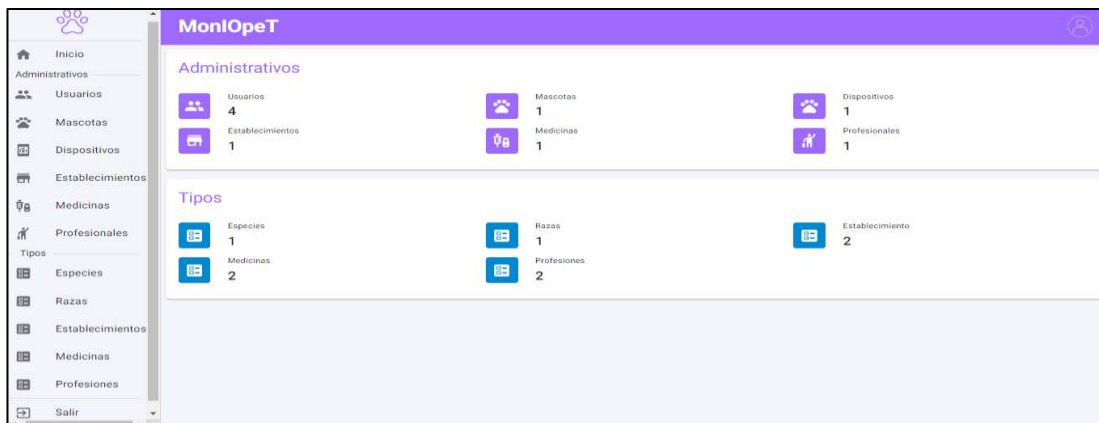
## Diseño iteración 2

### Historia de usuario H06 Panel de administración

El usuario administrador en el panel de administración podrá ver los totales de los datos ingresados en cada sección de la PWA, además podrá gestionar cada opción que se presenta en el menú.

Las vistas en computadora del panel de administración son las siguientes:

### Vista en computadora



**Figura 3.89 Vista en computadora del panel de administración**  
Elaborado por: El investigador

### Historia de usuario H07 Gestión de usuarios

El administrador podrá gestionar la información de todos los usuarios de la PWA mediante las funciones de listar, guardar, actualizar y eliminar.

La información de los usuarios se presentará en una tabla, donde el administrador podrá visualizar los datos principales de los usuarios, la información a guardar, actualizar y eliminar se lo realizará mediante formularios de ingreso de información.

Las vistas en computadora de la gestión de usuarios son las siguientes:

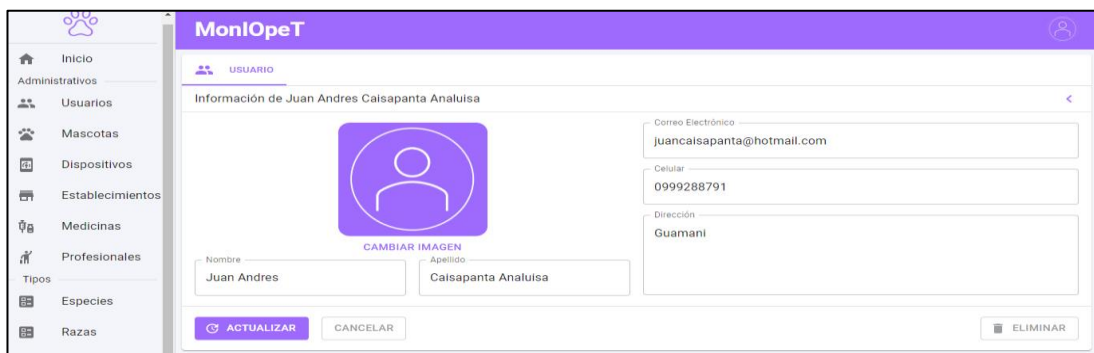
### Vista en computadora



**Figura 3.90 Vista en computadora de la tabla de información principal de los usuarios con opción de ver la información de cada usuario**  
Elaborado por: El investigador



**Figura 3.91 Vista en computadora del formulario para guardar un nuevo usuario**  
**Elaborado por: El investigador**



**Figura 3.92 Vista en computadora del formulario para la actualización y eliminación de información del usuario**  
**Elaborado por: El investigador**

## Historia de usuario H08 Gestión de mascotas

El administrador podrá gestionar la información de todas las mascotas de la PWA mediante las funciones de listar, guardar, actualizar y eliminar.

La información de las mascotas se presentará en una tabla, donde el administrador podrá visualizar los datos principales de las mascotas, la información a guardar, actualizar y eliminar se lo realizará mediante formularios de ingreso de información.

Las vistas en computadora de la gestión de mascotas son las siguientes:

### Vista en computadora



**Figura 3.93 Vista en computadora de la tabla de información principal de las mascotas con opción de ver la información de cada mascota**  
**Elaborado por: El investigador**

**Figura 3.94 Vista en computadora del formulario para guardar una nueva mascota**  
**Elaborado por: El investigador**

**Figura 3.95 Vista en computadora del formulario para la actualización y eliminación de información de la mascota**  
**Elaborado por: El investigador**

## **Historia de usuario H09 Gestión de dispositivos**

El administrador podrá gestionar la información de todos los de la PWA mediante las funciones de listar, guardar y eliminar.

La información de los dispositivos se presentará en una tabla, donde el administrador podrá visualizar los datos principales de los dispositivos y además eliminarlos, adicionalmente la información a guardar se lo realizará mediante formularios de ingreso de información.

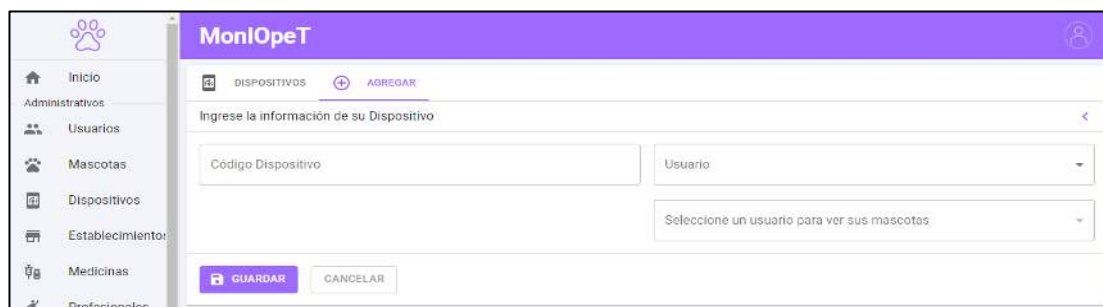
Los datos enviados por el dispositivo se presentan en una tabla, donde se visualiza las coordenadas de ubicación, la temperatura de la mascota y la batería que el dispositivo tenía en el momento de envío de los datos.

Las vistas en computadora de la gestión de dispositivos son las siguientes:

### Vista en computadora



**Figura 3.96** Vista en computadora de la tabla de información principal de los dispositivos con opción de ver y eliminar la información de cada dispositivo  
Elaborado por: El investigador



**Figura 3.97** Vista en computadora del formulario para guardar un nuevo dispositivo  
Elaborado por: El investigador

Id	Latitud	Longitud	Temperatura	Batería	Código de dispositivo
384	-0.301502	-78.54155	15	2670	61b6ca79aae760164ba4dc22
383	-0.301485	-78.541504	15	2660	61b6ca79aae760164ba4dc22
382	-0.301494	-78.54151	15	2675	61b6ca79aae760164ba4dc22
381	-0.300958	-78.54162	15	2695	61b6ca79aae760164ba4dc22
380	-0.301313	-78.541534	15	2695	61b6ca79aae760164ba4dc22
379	-0.301439	-78.54146	15	2710	61b6ca79aae760164ba4dc22
378	-0.301627	-78.541504	15	2710	61b6ca79aae760164ba4dc22
377	-0.30144	-78.541504	15	2720	61b6ca79aae760164ba4dc22

**Figura 3.98** Vista en computadora de la tabla de información de los datos enviados desde el dispositivo  
Elaborado por: El investigador

### Historia de usuario H10 Gestión de establecimientos

El administrador podrá gestionar la información de todos los establecimientos de la PWA mediante las funciones de listar, guardar, actualizar y eliminar.

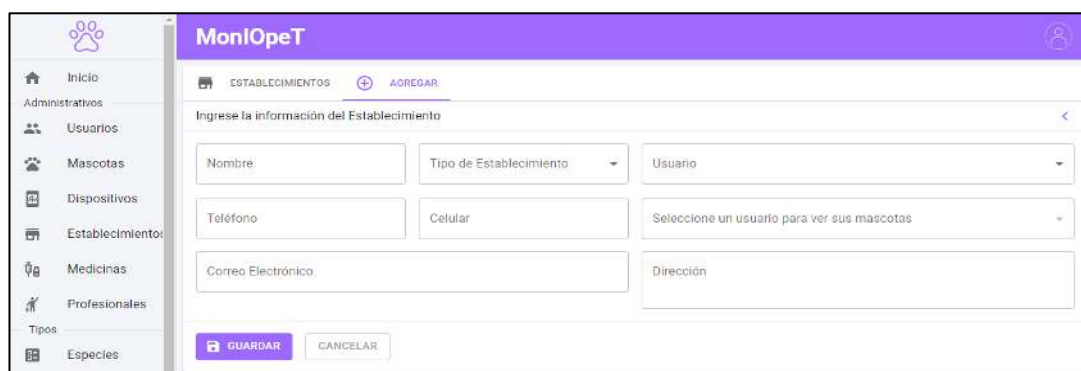
Los establecimientos se presentarán enlistados en una tabla y su información podrá ser gestionada mediante formularios.

Las vistas en computadora de la gestión de establecimientos son las siguientes:

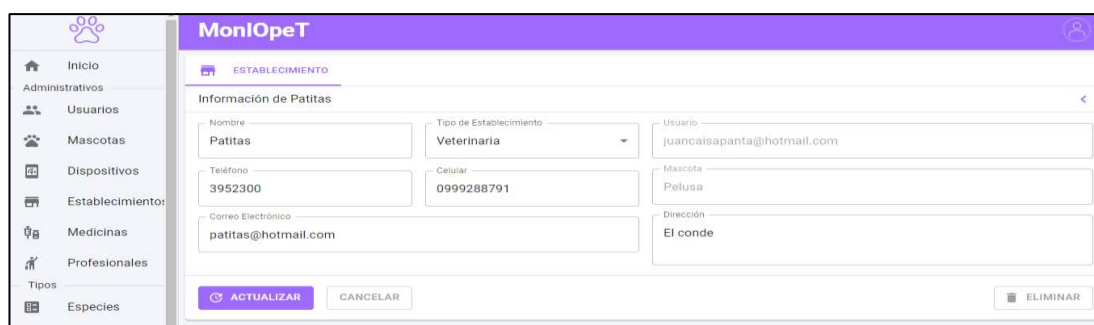
### Vista en computadora



**Figura 3.99 Vista en computadora de la tabla de información principal de los establecimientos con opción de ver la información de cada mascota**  
Elaborado por: El investigador



**Figura 3.100 Vista en computadora del formulario para guardar un nuevo establecimiento**  
Elaborado por: El investigador



**Figura 3.101 Vista en computadora del formulario para la actualización y eliminación de información del establecimiento**  
Elaborado por: El investigador

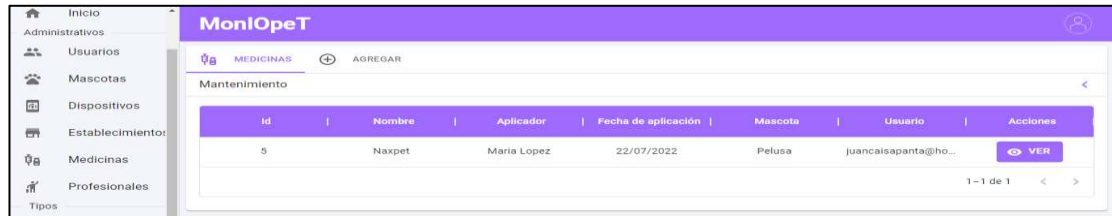
### Historia de usuario H11 Gestión de medicinas

El administrador podrá gestionar la información de todas las medicinas de la PWA mediante las funciones de listar, guardar, actualizar y eliminar.

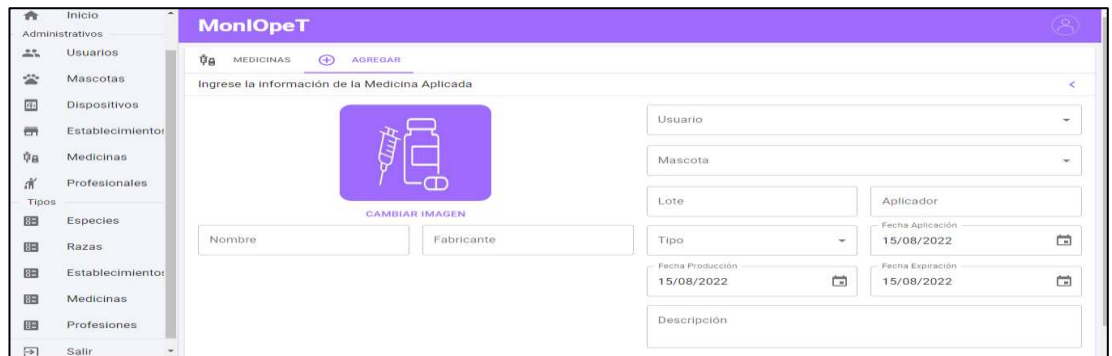
Las medicinas se presentarán enlistadas en una tabla y su información podrá ser gestionada mediante formularios.

Las vistas en computadora de la gestión de medicinas son las siguientes:

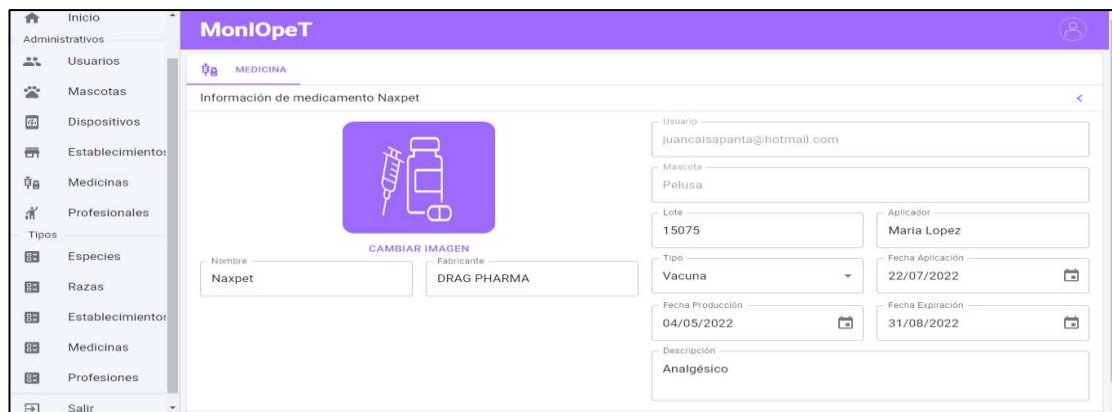
### Vista en computadora



**Figura 3.102 Vista en computadora de la tabla de información principal de las medicinas con opción de ver la información de cada medicina**  
Elaborado por: El investigador



**Figura 3.103 Vista en computadora del formulario para guardar una nueva medicina**  
Elaborado por: El investigador



**Figura 3.104 Vista en computadora del formulario para la actualización y eliminación de información de la medicina**  
Elaborado por: El investigador

### Historia de usuario H12 Gestión de profesionales

El administrador podrá gestionar la información de todos los profesionales de la PWA mediante las funciones de listar, guardar, actualizar y eliminar.



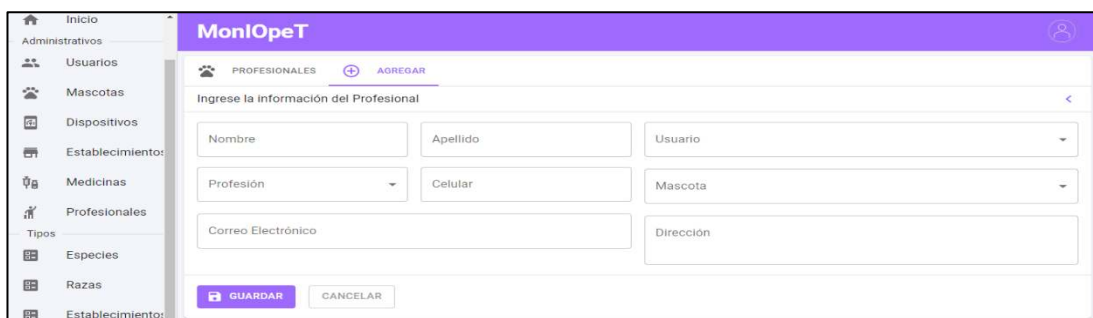
Los profesionales se presentarán enlistados en una tabla y su información podrá ser gestionada mediante formularios.

Las vistas en computadora de la gestión de profesionales son las siguientes:

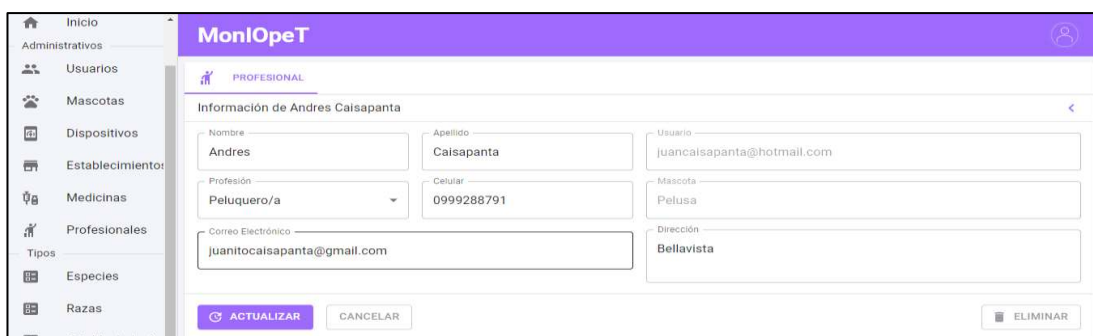
### Vista en computadora



**Figura 3.105** Vista en computadora de la tabla de información principal de los profesionales con opción de ver la información de cada profesional  
Elaborado por: El investigador



**Figura 3.106** Vista en computadora del formulario para guardar un nuevo profesional  
Elaborado por: El investigador



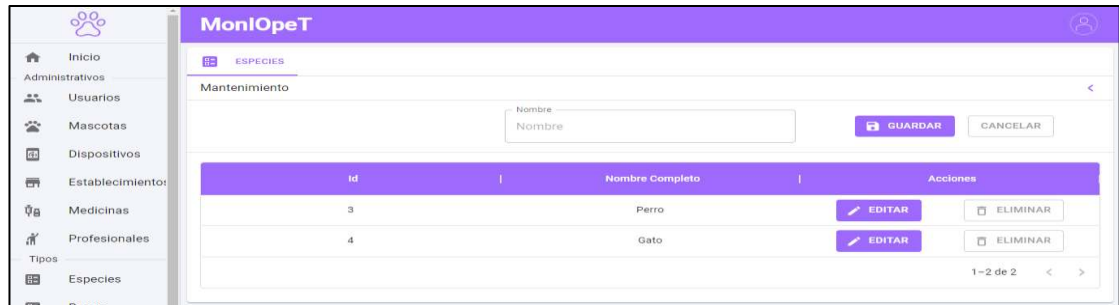
**Figura 3.107** Vista en computadora del formulario para la actualización y eliminación de información del profesional  
Elaborado por: El investigador

### Historia de usuario H13 Gestión de especies

El administrador podrá visualizar en una tabla la información principal de cada especie y gestionar todas las especies de la PWA mediante las funciones de guardar, actualizar y eliminar en un formulario.

Las vistas en computadora de la gestión de especies son las siguientes:

### Vista en computadora



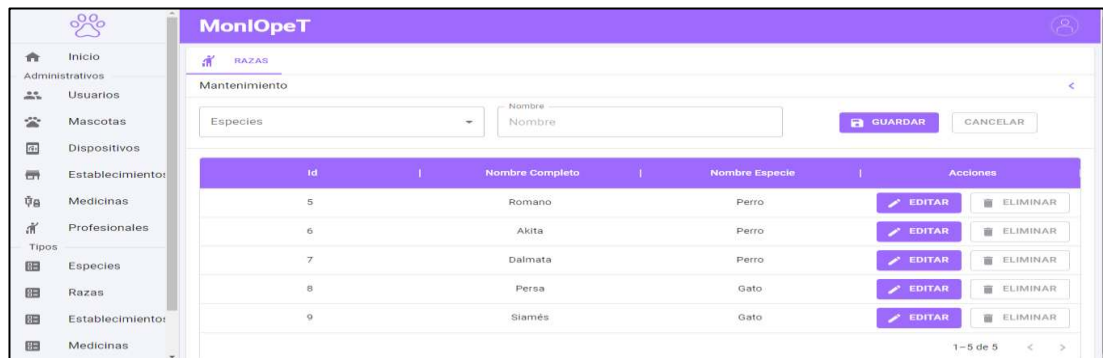
**Figura 3.108 Vista en computadora de la tabla y formulario para la gestión de especies**  
Elaborado por: El investigador

### Historia de usuario H14 Gestión de razas

El administrador podrá visualizar en una tabla la información principal de cada raza y gestionar todas las razas de la PWA mediante las funciones de guardar, actualizar y eliminar en un formulario.

Las vistas en computadora de la gestión de razas son las siguientes:

### Vista en computadora



**Figura 3.109 Vista en computadora de la tabla y formulario para la gestión de razas**  
Elaborado por: El investigador

### Historia de usuario H15 Gestión de tipos de establecimientos

El administrador podrá visualizar en una tabla la información principal de cada tipo de establecimiento y gestionar todos los tipos de establecimientos de la PWA mediante las funciones de guardar, actualizar y eliminar en un formulario.

Las vistas en computadora de la gestión de tipos de establecimientos son las siguientes:

## Vista en computadora



**Figura 3.110 Vista en computadora de la tabla y formulario para la gestión de los tipos de establecimientos**

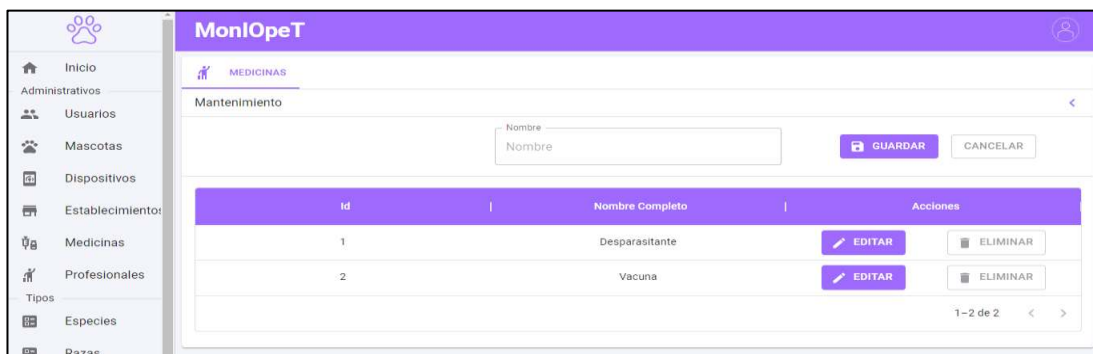
Elaborado por: El investigador

## Historia de usuario H16 Gestión de tipos de medicinas

El administrador podrá visualizar en una tabla la información principal de cada tipo de medicina y gestionar todos los tipos de medicinas de la PWA mediante las funciones de guardar, actualizar y eliminar en un formulario.

Las vistas en computadora de la gestión de tipos de medicinas son las siguientes:

## Vista en computadora



**Figura 3.111 Vista en computadora de la tabla y formulario para la gestión de los tipos de medicinas**

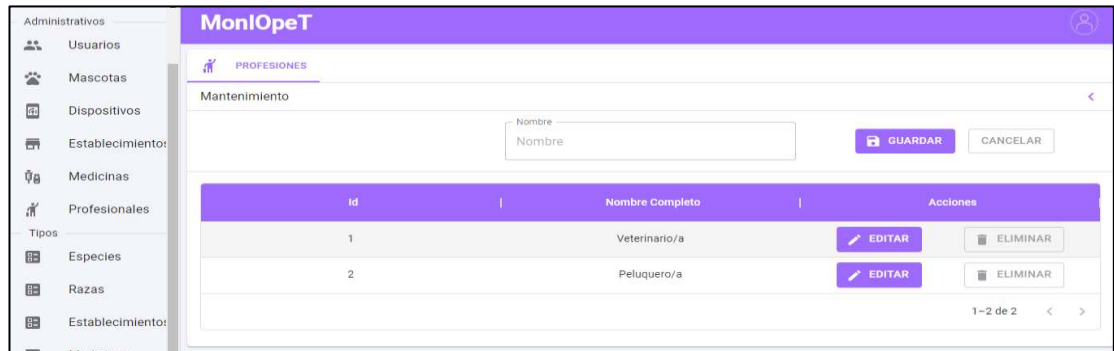
Elaborado por: El investigador

## Historia de usuario H17 Gestión de profesiones

El administrador podrá visualizar en una tabla la información principal de cada profesión y gestionar todas las profesiones de la PWA mediante las funciones de guardar, actualizar y eliminar en un formulario.

Las vistas en computadora de la gestión de profesiones son las siguientes:

## Vista en computadora



**Figura 3.112 Vista en computadora de la tabla y formulario para la gestión de profesiones**  
Elaborado por: El investigador

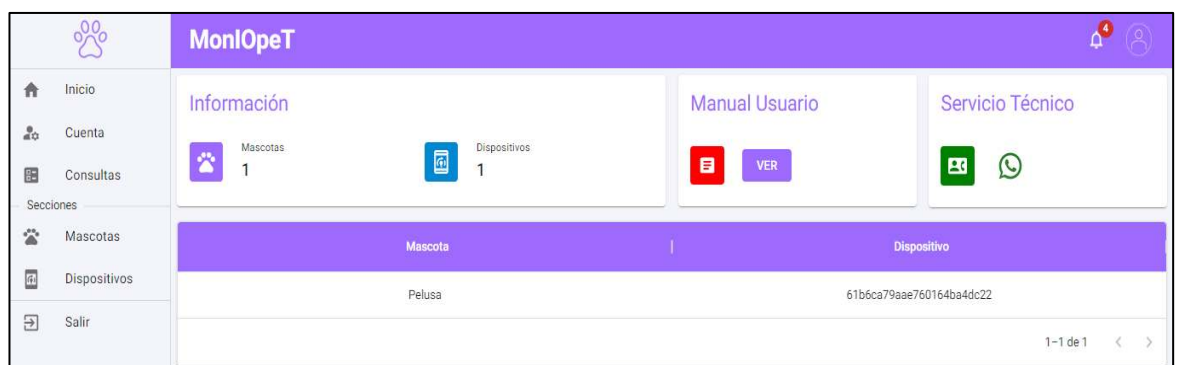
## Diseño iteración 3

### Historia de usuario H18 Panel de usuario

El usuario de la mascota al iniciar su sesión en la PWA observara su panel de usuario, en el cual se muestra cuantas mascotas y dispositivos tiene registrado el usuario, además se muestra la opción de ver el manual de usuario de la aplicación y el contacto de servicio técnico en caso de tener problemas con la PWA.

Las vistas en computadora y en dispositivos móviles del panel de usuario son las siguientes:

## Vista en computadora



**Figura 3.113 Vista en computadora del panel de usuario**  
Elaborado por: El investigador

## Vista en dispositivo móvil

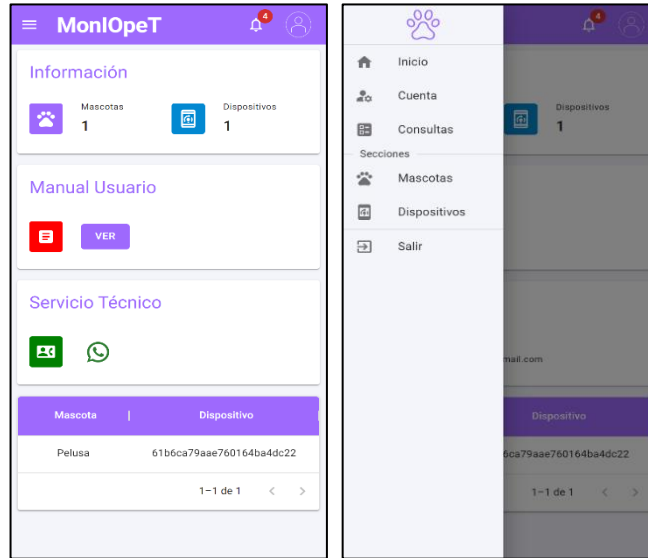


Figura 3.114 Vista en dispositivo móvil del panel y menú de opciones del usuario  
Elaborado por: El investigador

## Historia de usuario H19 Configuración cuenta de usuario

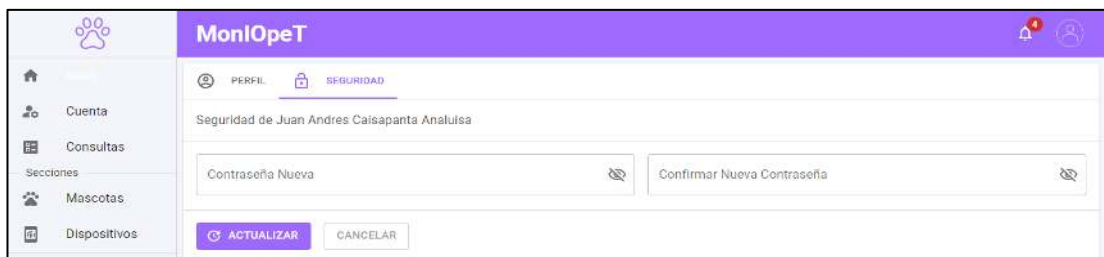
El usuario dueño de la mascota podrá actualizar la información con la que se registró en la PWA, además podrá cambiar su contraseña por medio de formularios.

Las vistas en computadora y en dispositivos móviles de la configuración de la cuenta de usuario son las siguientes:

## Vista en computadora

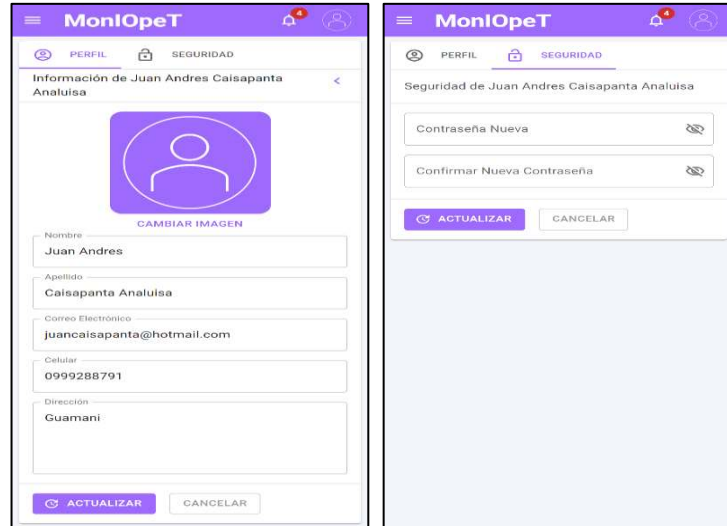


Figura 3.115 Vista en computadora del formulario de actualización de perfil de usuario  
Elaborado por: El investigador



**Figura 3.116 Vista en computadora del formulario para la actualización de contraseña del usuario**  
**Elaborado por: El investigador**

### Vista en dispositivo móvil



**Figura 3.117 Vista en dispositivo móvil de los formularios de actualización de perfil y contraseña del usuario**  
**Elaborado por: El investigador**

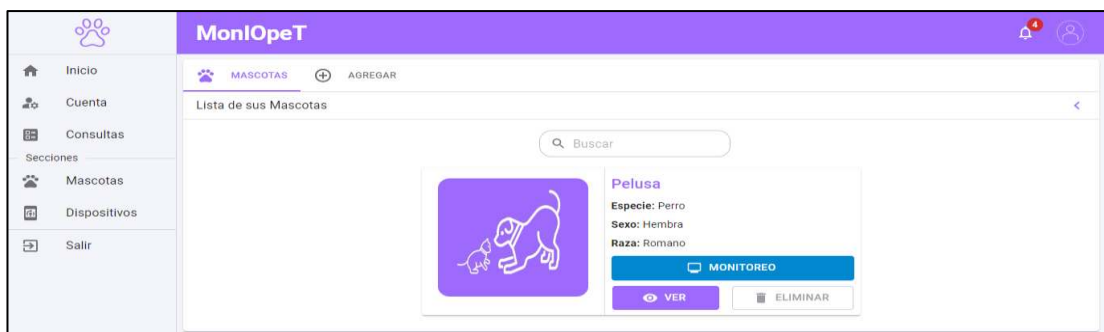
### Historia de usuario H20 Gestión de las mascotas del usuario

La información de las mascotas del usuario se presentará por medio de cartas, donde existen las opciones de ingresar a la sección de monitoreo, visualizar y eliminar la información de sus mascotas.

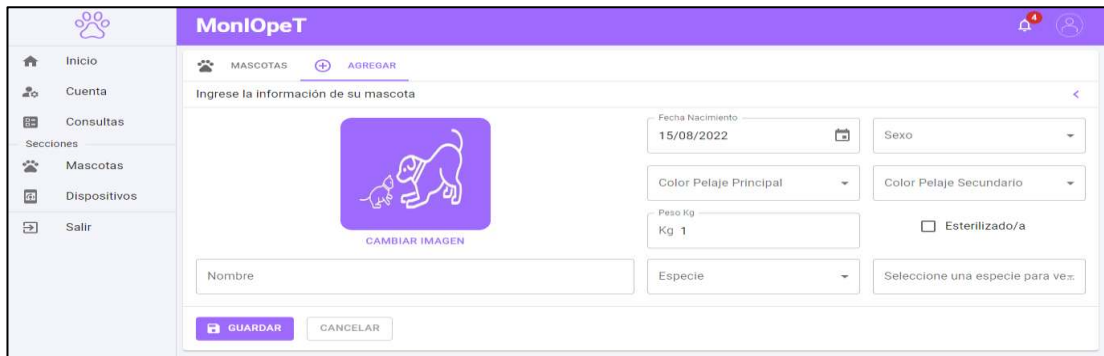
La información por guardar y actualizar se lo realizará mediante formularios de ingreso de información de la mascota del usuario.

Las vistas en computadora y en dispositivos móviles de la gestión de las mascotas del usuario son las siguientes:

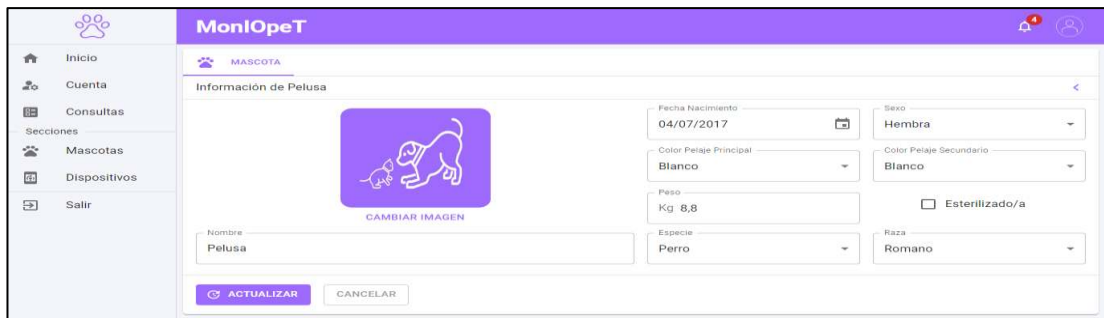
### Vista en computadora



**Figura 3.118 Vista en computadora de la información de las mascotas del usuario en cartas con opciones de ingreso a monitoreo, ver y eliminar la información**  
**Elaborado por: El investigador**

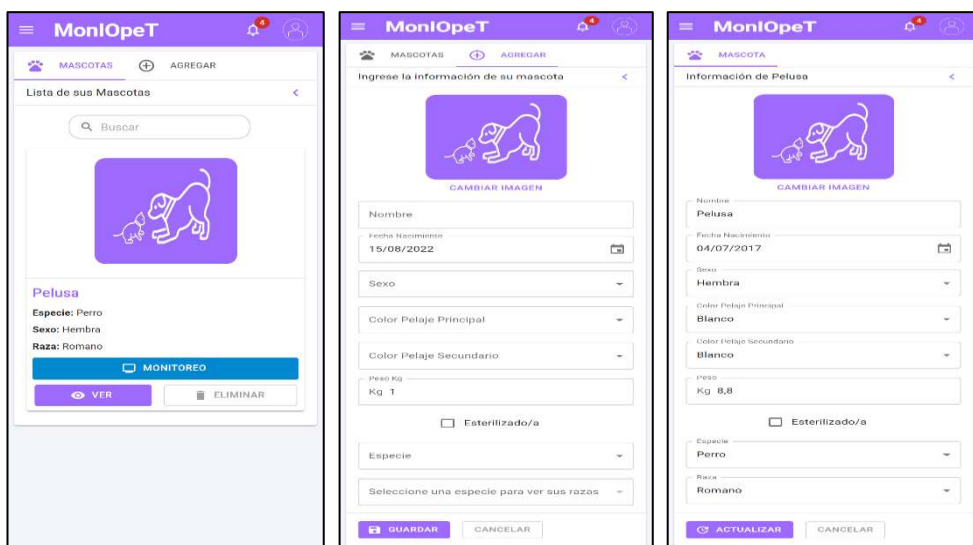


**Figura 3.119 Vista en computadora del formulario para guardar una nueva mascota del usuario**  
**Elaborado por: El investigador**



**Figura 3.120 Vista en computadora del formulario para actualizar la información de una mascota del usuario**  
**Elaborado por: El investigador**

### Vista en dispositivo móvil



**Figura 3.121 Vistas en dispositivo móvil de las opciones de ingreso al monitoreo, ver y eliminar la información de las mascotas del usuario presentada en cartas y los formularios para guardar y actualizar una mascota**  
**Elaborado por: El investigador**

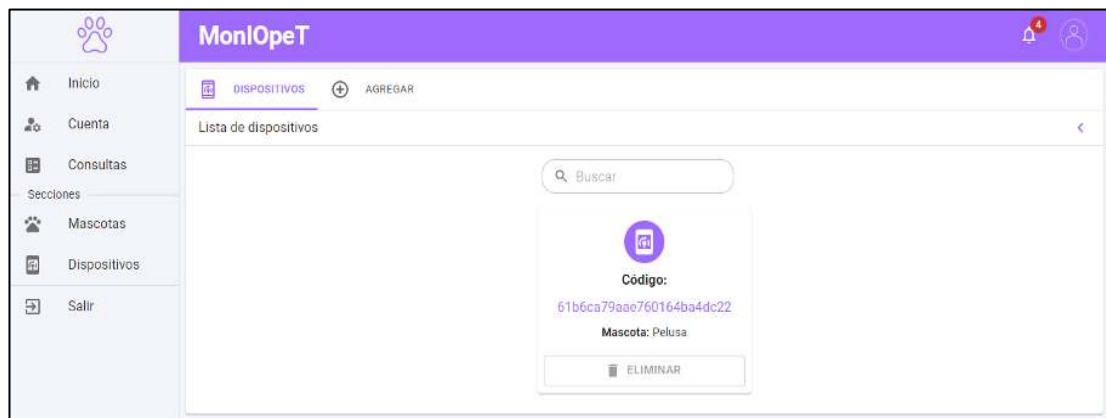
## Historia de usuario H21 Gestión de los dispositivos del usuario

La información de los dispositivos del usuario se presentará por medio de cartas, donde existen la opción de eliminar el dispositivo asignado a una mascota.

La información del dispositivo a guardar se lo realizará mediante formularios de ingreso de información, donde se agregará el código del dispositivo del usuario y la mascota a la cual se asignará el dispositivo.

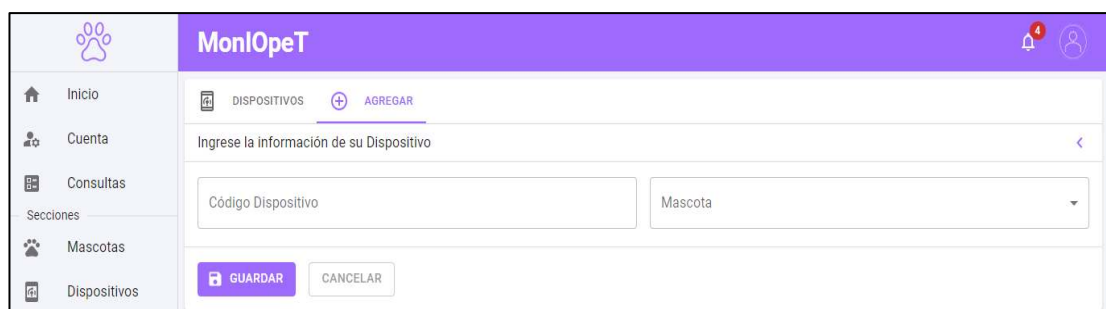
Las vistas en computadora y en dispositivos móviles de la gestión de los dispositivos del usuario son las siguientes:

### Vista en computadora



**Figura 3.122 Vista en computadora de la información de los dispositivos del usuario en cartas con opción de eliminar la información del dispositivo asignado a una mascota**

Elaborado por: El investigador

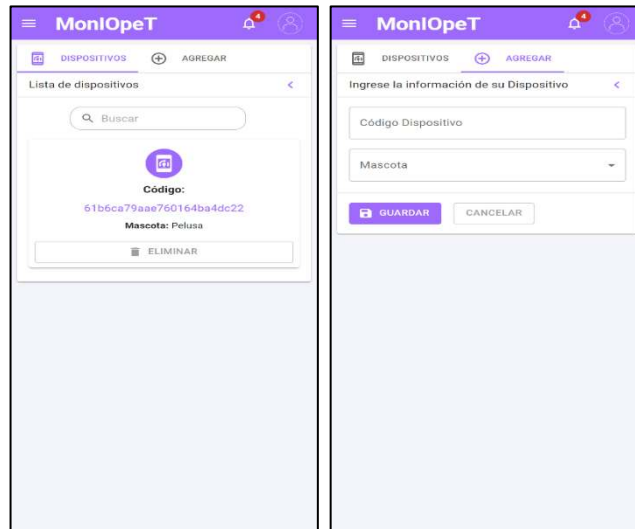


**Figura 3.123 Vista en computadora del formulario de ingreso de código del dispositivo del usuario y asignación de una mascota.**

Elaborado por: El investigador



## Vista en dispositivo móvil



**Figura 3.124 Vista en dispositivo móvil de la información del dispositivo asignado a una mascota, con opción a eliminar el dispositivo y el formulario de ingreso del código del dispositivo con la asignación de una mascota**  
Elaborado por: El investigador

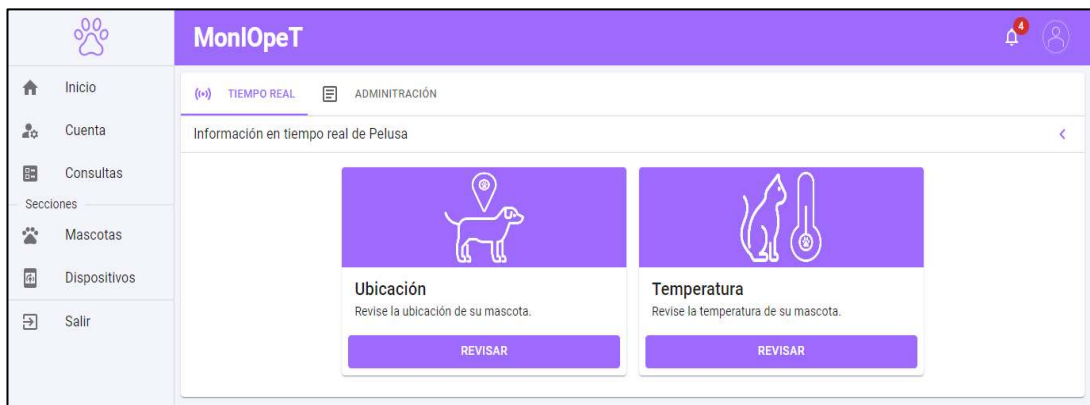
## Diseño iteración 4

### Historia de usuario H22 Secciones de monitorización

El usuario dueño de la mascota al seleccionar la opción de monitoreo de una mascota accederá a las opciones de ver la ubicación y temperatura de su mascota en tiempo real, además podrá gestionar los establecimientos que frecuenta, las medicinas que usa y los profesionales que visita su mascota.

Las vistas en computadora y en dispositivos móviles de las secciones de monitorización son las siguientes:

### Vista en computadora

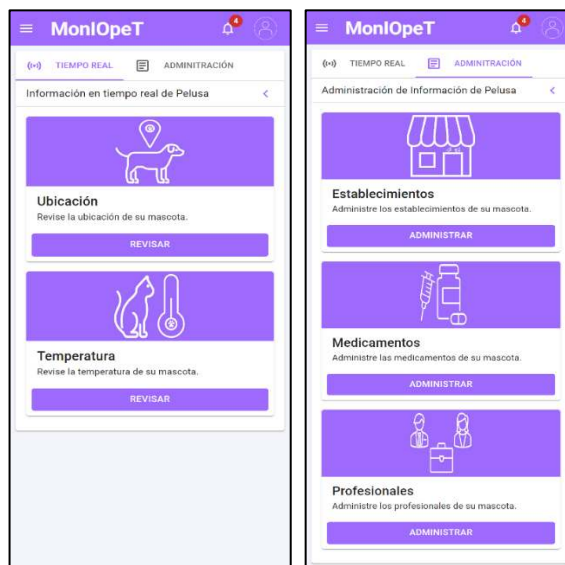


**Figura 3.125 Vista en computadora de las secciones de ubicación y temperatura en tiempo real de la mascota del usuario**  
Elaborado por: El investigador



**Figura 3.126 Vista en computadora de las secciones de gestión de establecimientos, medicinas y profesionales de la mascota del usuario**  
Elaborado por: El investigador

### Vista en dispositivo móvil



**Figura 3.127 Vistas en dispositivo móvil de las secciones de ubicación y temperatura en tiempo real, gestión de establecimientos, gestión de medicinas y gestión de profesionales**  
Elaborado por: El investigador

### Historia de usuario H23 Ubicación de la mascota del usuario en tiempo real

El usuario dueño de la mascota al ingresar en la opción de ubicación de la sección de monitoreo podrá visualizar la ubicación de su mascota en tiempo real por medio de un mapa con la posibilidad de abrir la ubicación en Google Maps, para revisar la ruta que debería seguir para llegar a su mascota, además podrá ver la batería disponible en el dispositivo de monitoreo y compartir la ubicación por medio de WhatsApp o correo electrónico.

Las vistas en computadora y en dispositivos móviles de la ubicación de la mascota del usuario en tiempo real son las siguientes:

### Vista en computadora

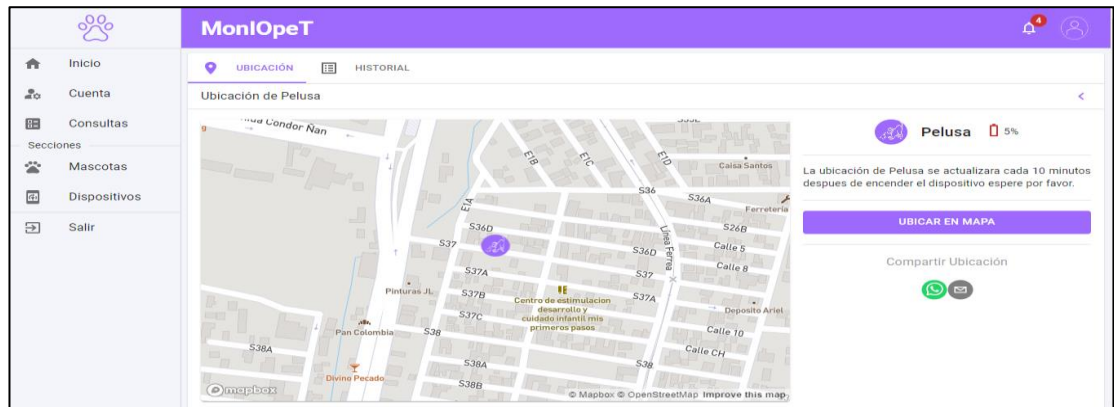


Figura 3.128 Vista en computadora de la ubicación en tiempo real de la mascota del usuario  
Elaborado por: El investigador

### Vista en dispositivo móvil

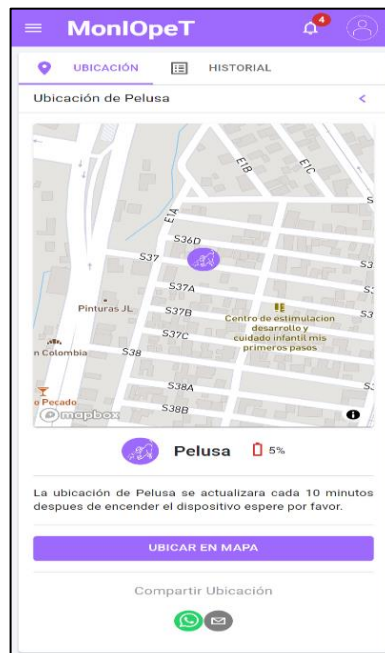


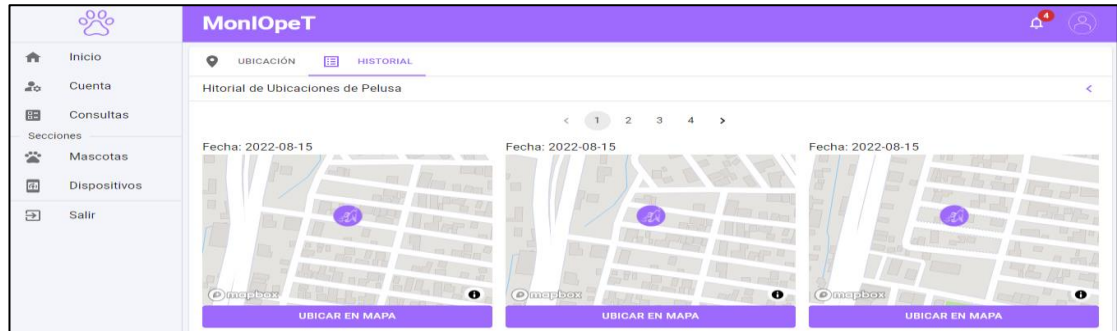
Figura 3.129 Vista en dispositivo móvil de la ubicación en tiempo real de la mascota del usuario  
Elaborado por: El investigador

### Historia de usuario H24 Historial de ubicaciones de la mascota del usuario en tiempo real

Las ubicaciones de la mascota del usuario se guardan para poder visualizar por fecha y mediante mapas los lugares donde se encontraba su mascota.

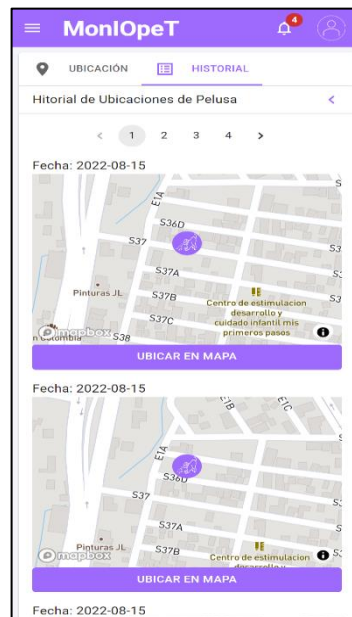
Las vistas en computadora y en dispositivos móviles del historial de ubicaciones de la mascota del usuario en tiempo real son las siguientes:

### Vista en computadora



**Figura 3.130 Vista en computadora del historial de ubicaciones de la mascota del usuario**  
Elaborado por: El investigador

### Vista en dispositivo móvil



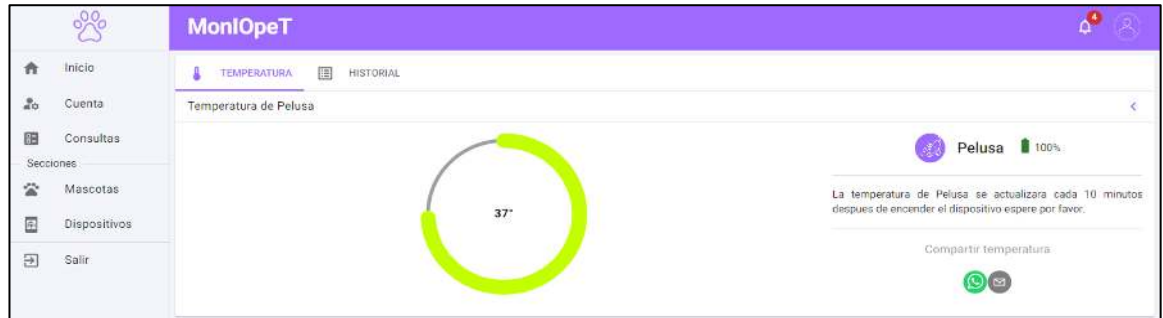
**Figura 3.131 Vista en dispositivo móvil del historial de ubicaciones de la mascota del usuario**  
Elaborado por: El investigador

### Historia de usuario H25 Temperatura de la mascota del usuario en tiempo real

El usuario dueño de la mascota al ingresar en la opción de temperatura de la sección de monitoreo podrá visualizar la temperatura de su mascota en tiempo real por medio de grafico que indicará la temperatura actual de la mascota, además podrá ver la batería disponible en el dispositivo de monitoreo y compartir la temperatura por medio de WhatsApp o correo electrónico.

Las vistas en computadora y en dispositivos móviles de la temperatura de la mascota del usuario en tiempo real son las siguientes:

### Vista en computadora



**Figura 3.132 Vista en computadora de la temperatura en tiempo real de la mascota del usuario**  
Elaborado por: El investigador

### Vista en dispositivo móvil



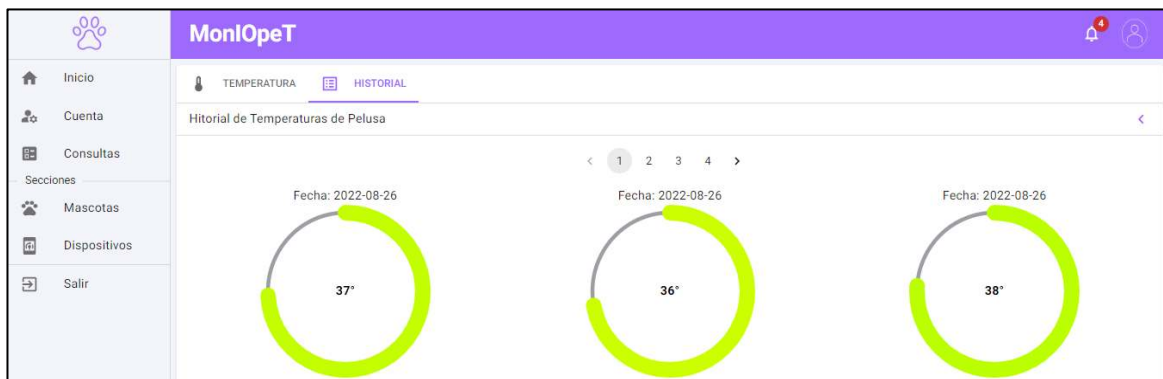
**Figura 3.133 Vista en dispositivo móvil de la temperatura en tiempo real de la mascota del usuario**  
Elaborado por: El investigador

### Historia de usuario H26 Historial de temperaturas de la mascota del usuario en tiempo real

Las temperaturas de la mascota del usuario se guardan para poder visualizar por fecha y mediante gráficos la temperatura su mascota.

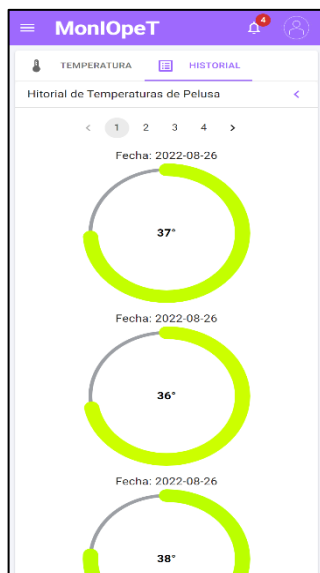
Las vistas en computadora y en dispositivos móviles del historial de temperaturas de la mascota del usuario en tiempo real son las siguientes:

## Vista en computadora



**Figura 3.134** Vista en computadora del historial de temperaturas de la mascota del usuario  
Elaborado por: El investigador

## Vista en dispositivo móvil



**Figura 3.135** Vista en dispositivo móvil del historial de temperaturas de la mascota del usuario  
Elaborado por: El investigador

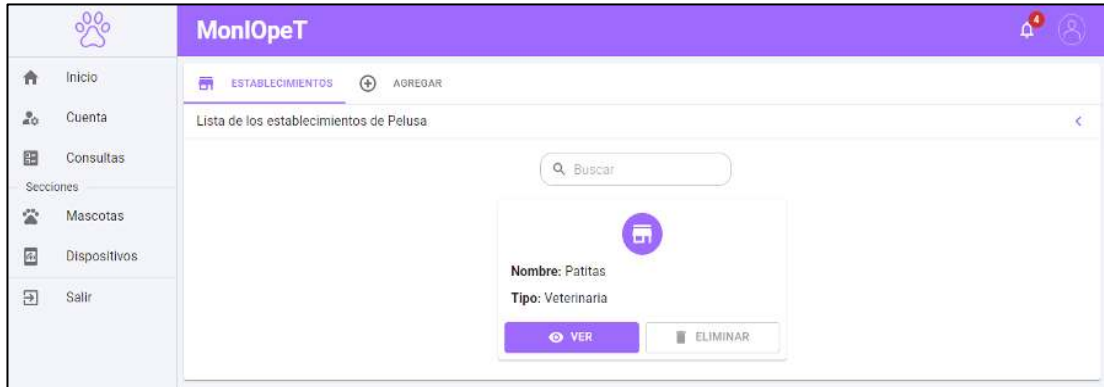
## Historia de usuario H27 Gestión de establecimientos de la mascota

La información de los establecimientos de la mascota del usuario se presentará por medio de cartas, donde existen las opciones de visualizar y eliminar la información de los establecimientos.

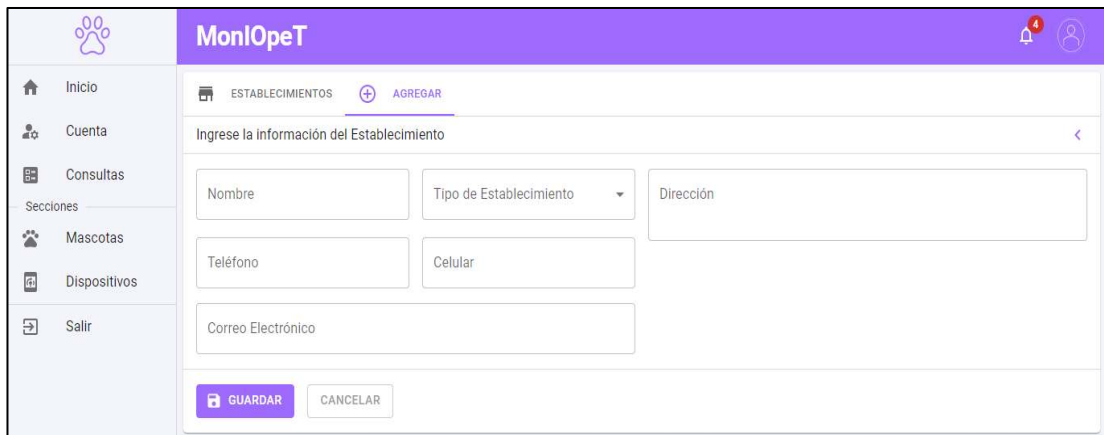
La información por guardar y actualizar se lo realizará mediante formularios de ingreso de información de los establecimientos de la mascota.

Las vistas en computadora y en dispositivos móviles de la gestión de establecimientos de la mascota del usuario son las siguientes:

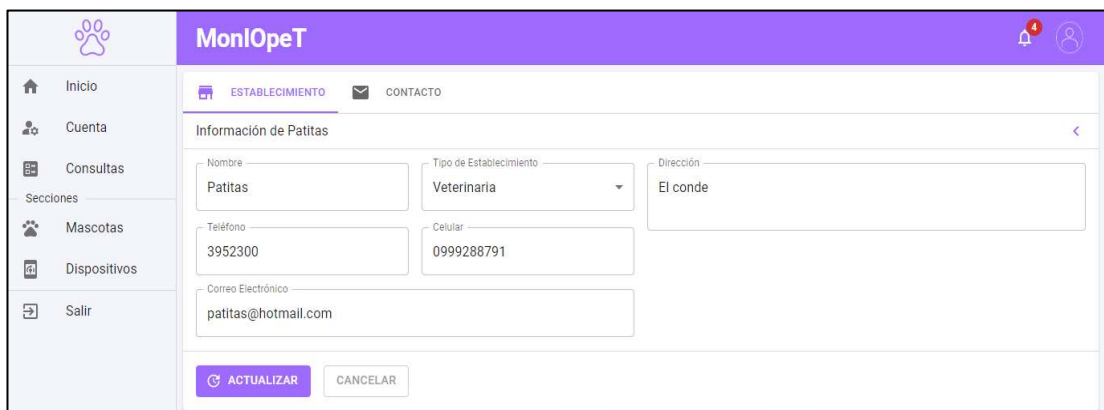
## Vista en computadora



**Figura 3.136** Vista en computadora de la información de los establecimientos de la mascota del usuario en cartas con opciones de ver y eliminar la información del establecimiento  
Elaborado por: El investigador

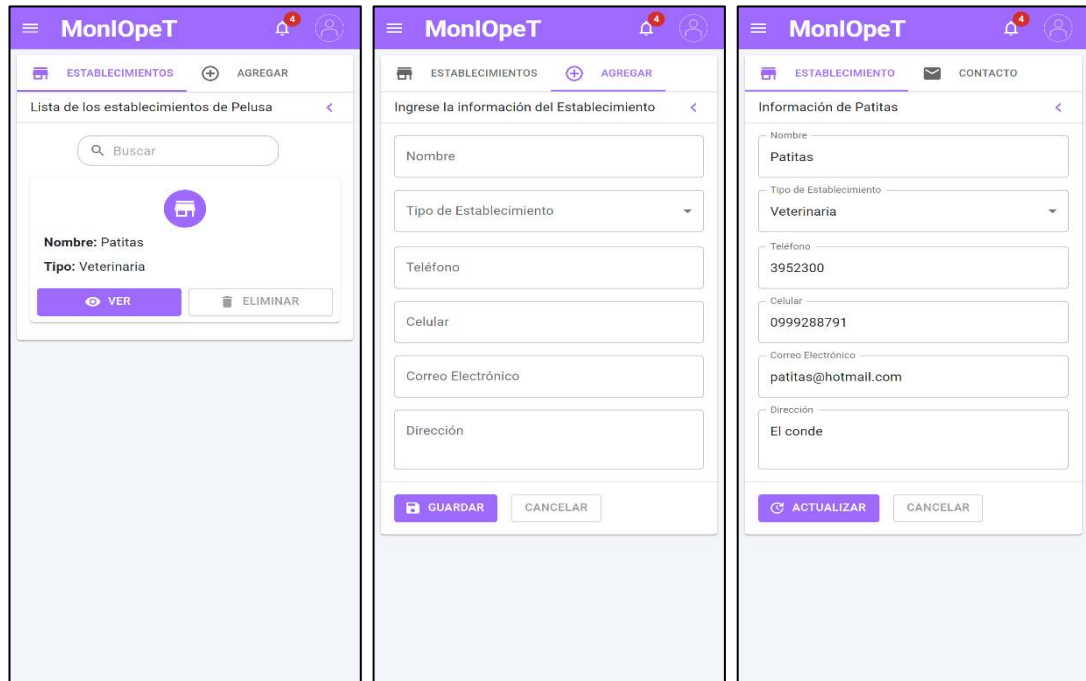


**Figura 3.137** Vista en computadora del formulario para guardar un nuevo establecimiento de la mascota del usuario  
Elaborado por: El investigador



**Figura 3.138** Vista en computadora del formulario para actualizar la información del establecimiento de la mascota del usuario  
Elaborado por: El investigador

## Vista en dispositivo móvil



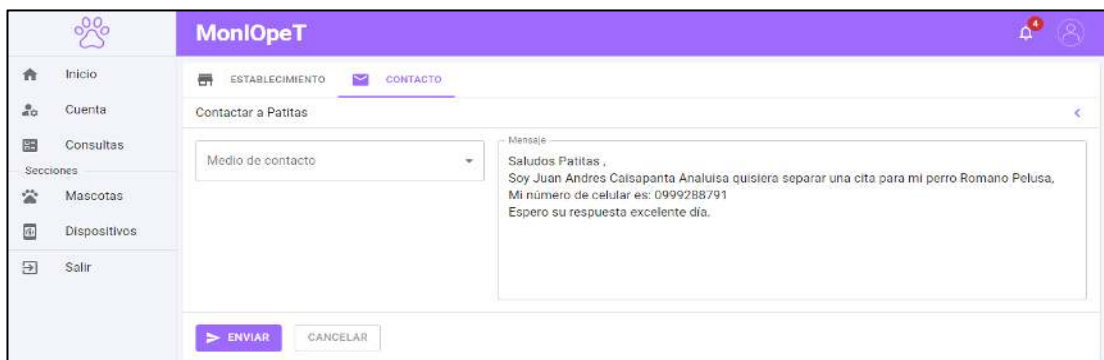
**Figura 3.139** Vistas en dispositivo móvil de las opciones de ver y eliminar la información de los establecimientos de la mascota del usuario presentada en cartas y los formularios para guardar y actualizar un establecimiento  
Elaborado por: El investigador

## Historia de usuario H28 Contactar establecimiento de la mascota

El usuario dueño de la mascota podrá contactar al establecimiento que frecuenta su mascota por medio de correo electrónico o WhatsApp.

La vista en computadora y en dispositivo móvil para contactar al establecimiento de la mascota son las siguientes:

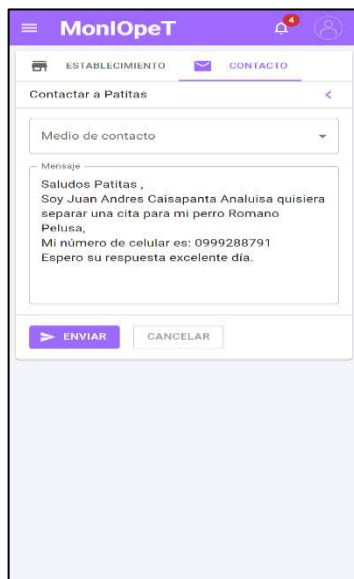
## Vista en computadora



**Figura 3.140** Vista en computadora del formulario para contactar al establecimiento de la mascota del usuario  
Elaborado por: El investigador



## Vista en dispositivo móvil



**Figura 3.141 Vista en dispositivo móvil del formulario para contactar al establecimiento de la mascota del usuario**  
Elaborado por: El investigador

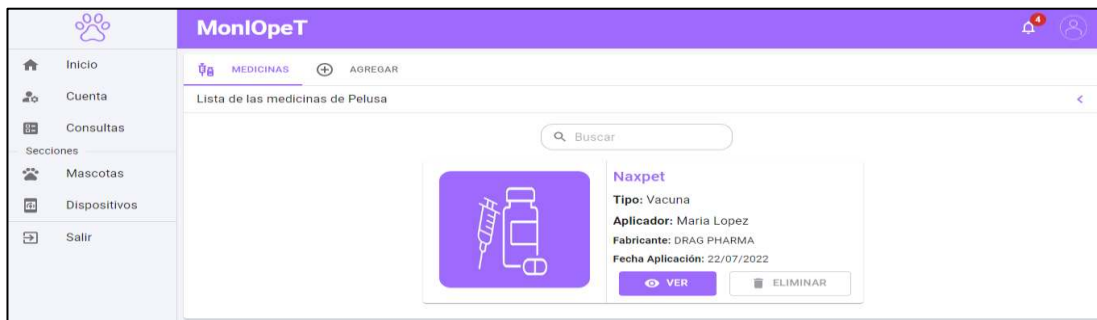
## Historia de usuario H29 Gestión de medicinas de la mascota

La información de las medicinas de la mascota del usuario se presentará por medio de cartas, donde existen las opciones de visualizar y eliminar la información de las medicinas.

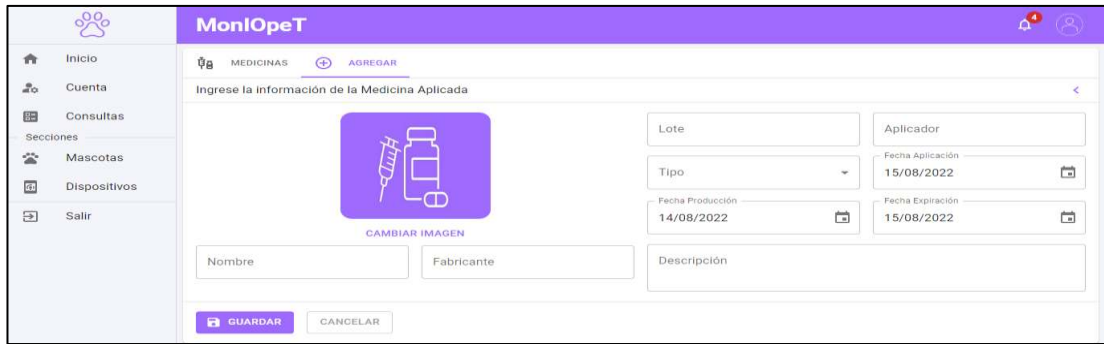
La información por guardar y actualizar se lo realizará mediante formularios de ingreso de información de las medicinas de la mascota.

Las vistas en computadora y en dispositivos móviles de la gestión de medicinas de la mascota del usuario son las siguientes:

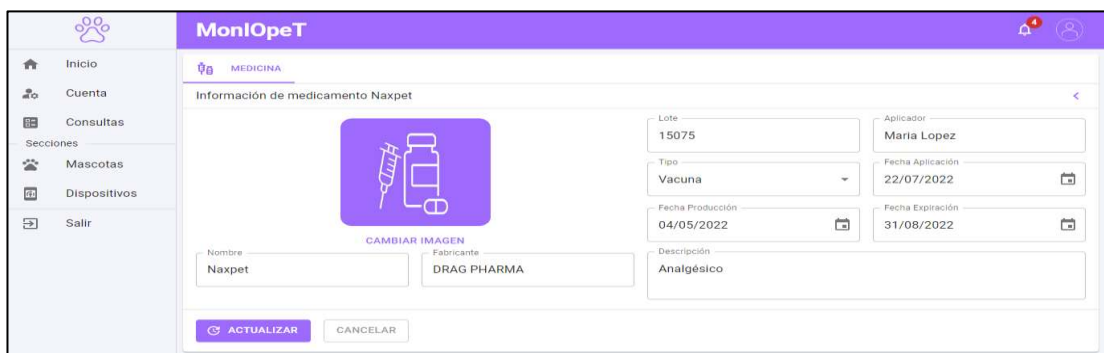
## Vista en computadora



**Figura 3.142 Vista en computadora de la información de las medicinas de la mascota del usuario en cartas con opciones de ver y eliminar la información de la medicina**  
Elaborado por: El investigador

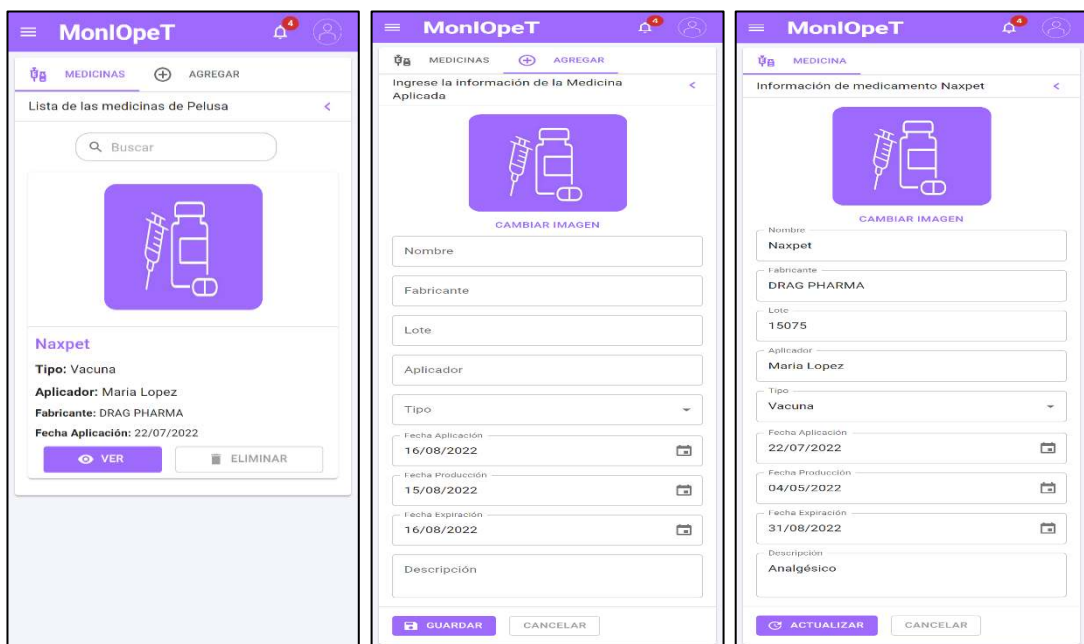


**Figura 3.143 Vista en computadora del formulario para guardar una nueva medicina de la mascota del usuario**  
**Elaborado por: El investigador**



**Figura 3.144 Vista en computadora del formulario para actualizar la información de la medicina de la mascota del usuario**  
**Elaborado por: El investigador**

### Vista en dispositivo móvil



**Figura 3.145 Vistas en dispositivo móvil de las opciones de ver y eliminar la información de las medicinas de la mascota del usuario presentada en cartas y los formularios para guardar y actualizar una medicina**  
**Elaborado por: El investigador**

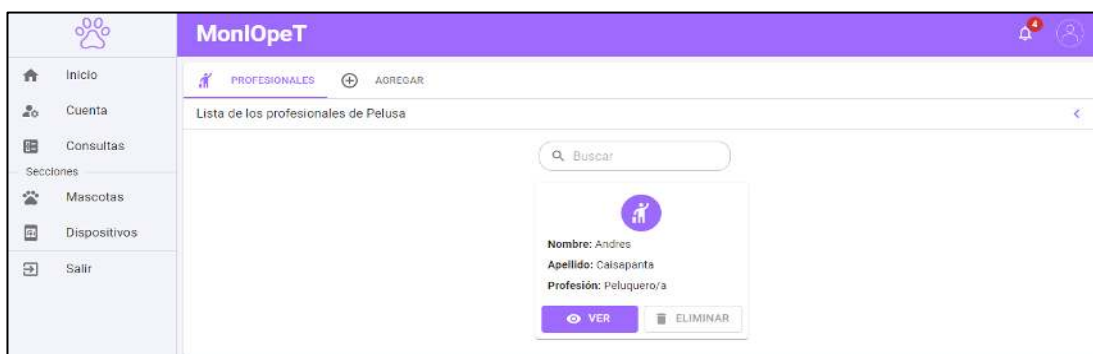
## Historia de usuario H30 Gestión de profesionales de la mascota

La información de los profesionales de la mascota del usuario se presentará por medio de cartas, donde existen las opciones de visualizar y eliminar la información de los profesionales.

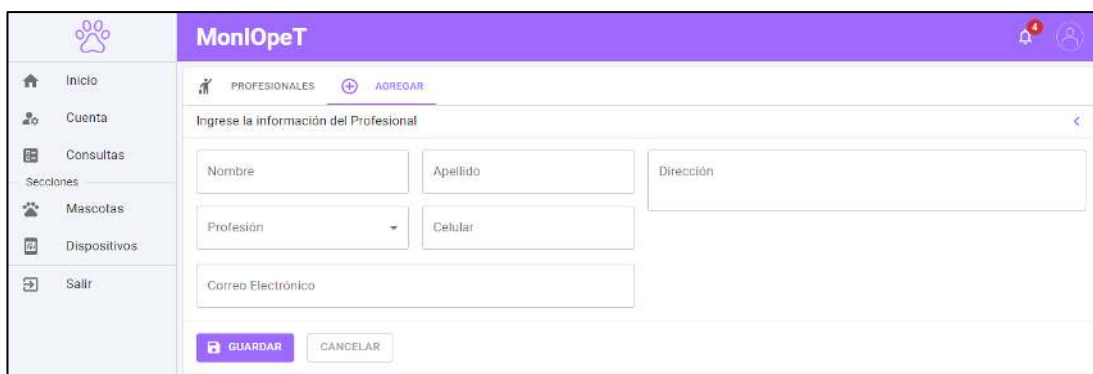
La información por guardar y actualizar se lo realizará mediante formularios de ingreso de información de los profesionales de la mascota.

Las vistas en computadora y en dispositivos móviles de la gestión de los profesionales de la mascota del usuario son las siguientes:

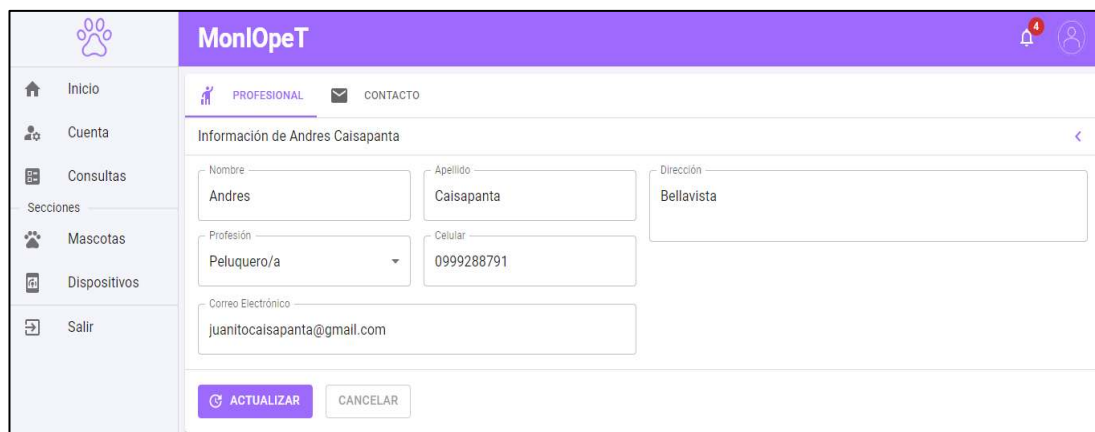
### Vista en computadora



**Figura 3.146 Vista en computadora de la información de los profesionales de la mascota del usuario en cartas con opciones de ver y eliminar la información del profesional**  
Elaborado por: El investigador

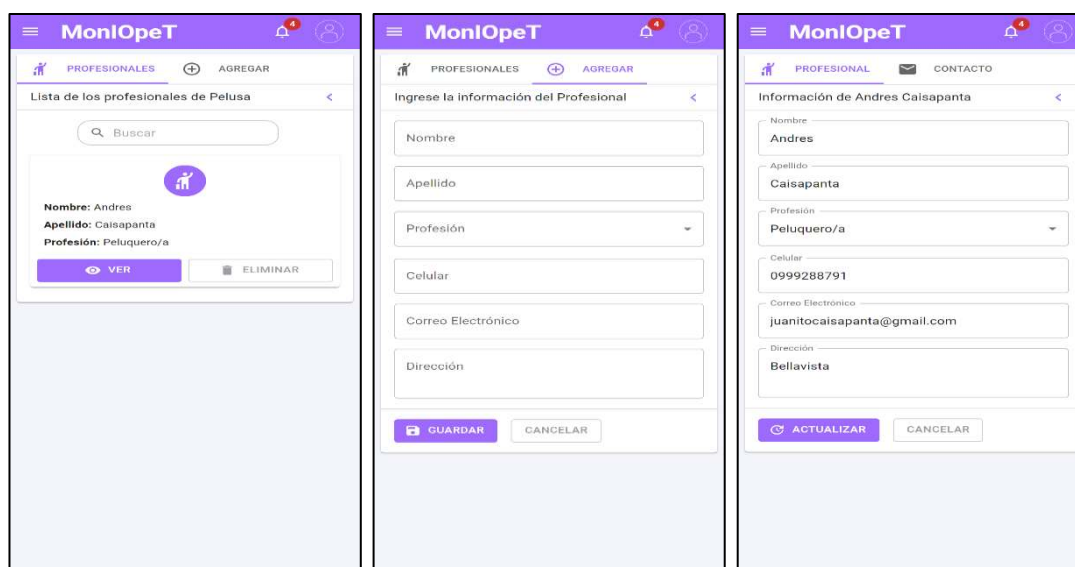


**Figura 3.147 Vista en computadora del formulario para guardar un nuevo profesional de la mascota del usuario**  
Elaborado por: El investigador



**Figura 3.148 Vista en computadora del formulario para actualizar la información del profesional de la mascota del usuario**  
**Elaborado por: El investigador**

### Vista en dispositivo móvil



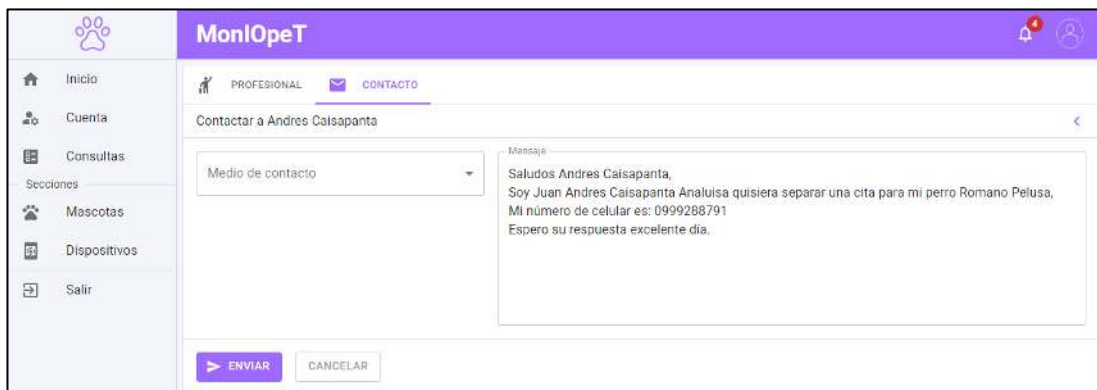
**Figura 3.149 Vistas en dispositivo móvil de las opciones de ver y eliminar la información de los profesionales de la mascota del usuario presentada en cartas y los formularios para guardar y actualizar un profesional**  
**Elaborado por: El investigador**

### Historia de usuario H31 Contactar profesional de la mascota

El usuario dueño de la mascota podrá contactar al profesional que regularmente visita su mascota por medio de correo electrónico o WhatsApp.

La vista en computadora y en dispositivo móvil para contactar al profesional de la mascota son las siguientes:

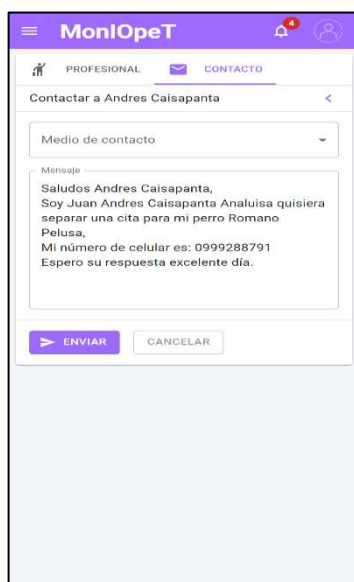
## Vista en computadora



**Figura 3.150 Vista en computadora del formulario para contactar al profesional de la mascota del usuario**

Elaborado por: El investigador

## Vista en dispositivo móvil



**Figura 3.151 Vista en dispositivo móvil del formulario para contactar al profesional de la mascota del usuario**

Elaborado por: El investigador

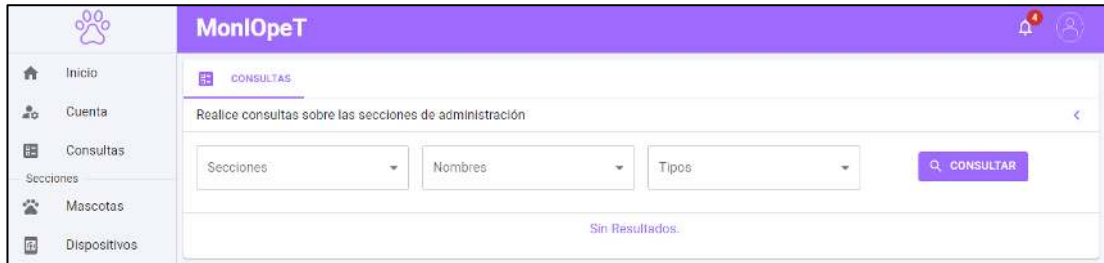
## Historia de usuario H32 Consulta de mascotas por secciones

El usuario dueño de la mascota podrá consultar que mascotas cumplen con la información de las secciones que gestiona o administra, estas son los establecimientos que frecuenta la mascota, las medicinas que usa la mascota y los profesionales que visita la mascota.

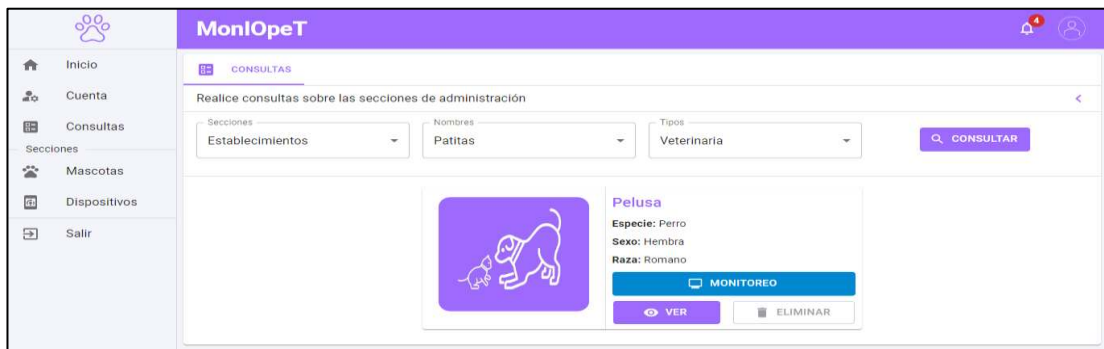
Al cumplir con la información de cada sección se presentará la tarjeta de la mascota la cual cumple con la información proporcionada.

Las vistas en computadora y en dispositivo móvil de la consulta de mascotas del usuario por secciones son las siguientes:

### Vista en computadora

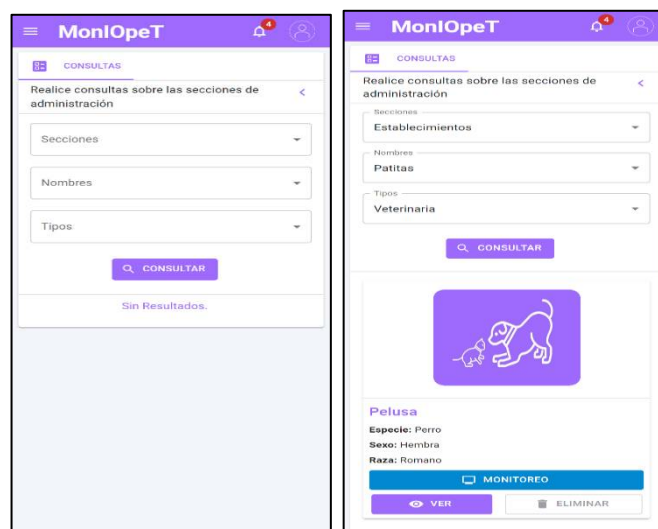


**Figura 3.152 Vista en computadora del formulario para la consulta de las mascotas del usuario por secciones**  
Elaborado por: El investigador



**Figura 3.153 Vista en computadora de la mascota consultada con la información proporcionada por secciones que gestiona el usuario**  
Elaborado por: El investigador

### Vista en dispositivo móvil



**Figura 3.154 Vistas en dispositivo móvil del formulario de consulta de las mascotas del usuario y el resultado de la consulta por secciones que gestiona el usuario presentado en cartas**  
Elaborado por: El investigador

### 3.2.3.6 Fase de Codificación

La fase de codificación es la tercera fase de la metodología XP, en esta fase se realiza la escritura del código con el que está desarrollado la PWA.

Los lenguajes usados para el desarrollo del back-end de la PWA es Java ya que se usará Spring Boot en el desarrollo de los servicios web tipo REST y el lenguaje usado para el desarrollo del front-end es TypeScript, con el que se desarrollará las interfaces graficas usando Next.js que es el framework de la librería React.js

A continuación, se mostrará el código usado en el desarrollo de las historias de usuario más relevantes de la PWA MonIOpeT.

#### Codificación de las historias de usuario más relevantes

##### Historia de usuario H01 Estructura del proyecto

Para empezar con el desarrollo de la PWA, primero se deben realizar las configuraciones necesarias para que el back-end y el front-end puedan trabajar conjuntamente, por lo que a continuación se muestra el código usado para la preparación del proyecto.

##### Código back-end Spring Boot

Al iniciar un proyecto en Spring Boot se realiza la modificación del archivo `application.properties`, donde se indican las variables con las que el back-end tendrá conexión a la base de datos, así como también al servidor SMTP para el servicio de correos y a JWT para el funcionamiento de la seguridad de la PWA.

El código usado para la modificación del archivo `application.properties` es el siguiente:

```
spring.datasource.url= ****
spring.datasource.username= ****
spring.datasource.password= ****
spring.jpa.hibernate.ddl-auto= update
spring.servlet.multipart.max-file-size= 5MB
spring.servlet.multipart.max-request-size= 5MB

#email
spring.mail.host= ****
spring.mail.port= 587
spring.mail.username= ****
spring.mail.password= ****

# Other properties
spring.mail.properties.mail.smtp.auth= true
spring.mail.properties.mail.smtp.connectiontimeout= 5000
spring.mail.properties.mail.smtp.timeout= 5000
spring.mail.properties.mail.smtp.writetimeout= 5000

# TLS , port 587
spring.mail.properties.mail.smtp.starttls.enable= true

# SSL, port 465
spring.mail.properties.mail.smtp.socketFactory.port = 465
spring.mail.properties.mail.smtp.socketFactory.class = javax.net.ssl.SSLSocketFactory
```

```
#security
jwt.secret = ****
```

## Código front-end PWA MonIOpeT Next.js

Next.js nos describe en su documentación que existe una librería que permite el desarrollo de una PWA de manera sencilla, ya que al instalar la librería “next-pwa” y modificando el archivo next.config.js el entorno estaría preparado para el desarrollo de una PWA.

El código usado para la modificación del archivo next.config.js es el siguiente:

```
const withPWA = require("next-pwa");
module.exports = withPWA({
  reactStrictMode: true,
  pwa: {
    dest: "public",
    register: true,
    skipWaiting: true,
    disable: process.env.NODE_ENV === "development",
  },
});
```

El archivo manifest.json es importante para el funcionamiento de la PWA, ya que en este se indica como funcionará y que iconos se usarán en los dispositivos donde se instale la PWA.

El código usado para el manifest.json de la PWA MonIOpeT es el siguiente:

```
{
  "theme_color": "#F4F5FA",
  "background_color": "#F4F5FA",
  "display": "standalone",
  "scope": "/",
  "start_url": "/",
  "name": "MonIOpeT",
  "short_name": "MonIOpeT",
  "description": "aplicaci\u00f3n",
  "icons": [
    {
      "src": "/icon-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "/icon-256x256.png",
      "sizes": "256x256",
      "type": "image/png"
    },
    {
      "src": "/icon-384x384.png",
      "sizes": "384x384",
      "type": "image/png"
    },
    {
      "src": "/icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

## Historia de usuario H03 Registro de usuarios

Como primera parte para el registro de nuevos usuarios, se realiza la configuración de seguridad en el back-end de la PWA, esto se realiza gracias a que Spring Boot cuenta con la herramienta de Spring Security que permite la seguridad en inicios de sesión de



usuario mediante configuración de métodos y clases, al finalizar la configuración el front-end será capaz de consumir el servicio web tipo REST generado.

## Código back-end Spring Boot

La clase `UserPrincipalDetails` es usada por Spring Security para la configuración del usuario al realizar el inicio de sesión en la PWA.

El código usado para el desarrollo de la clase `UserPrincipalDetails` es el siguiente:

```
@Data
@AllArgsConstructor
public class UserPrincipalDetails implements UserDetails {
    private static final long serialVersionUID = 1L;
    private Long id;
    private String name;
    private String last_name;
    private String email;
    private String password;
    private String address;
    private String phone;
    private Collection<? extends GrantedAuthority> authorities;
    public static UserPrincipalDetails build(Users user) {
        List<GrantedAuthority> authorities = user.getRole().stream().map(role -> new
SimpleGrantedAuthority(role.getName().name()).collect(Collectors.toList());
        return new UserPrincipalDetails(user.getId(), user.getName(), user.getLast_name(),
user.getEmail(),user.getPassword(), user.getAddress(), user.getPhone(), authorities);
    }
    @Override
    public Collection<? extends GrantedAuthority> getAuthorities() {
        return authorities;
    }
    @Override
    public String getPassword() {
        return password;
    }
    @Override
    public String getUsername() {
        return email;
    }
    @Override
    public boolean isAccountNonExpired() {
        return true;
    }
    @Override
    public boolean isAccountNonLocked() {
        return true;
    }
    @Override
    public boolean isCredentialsNonExpired() {
        return true;
    }
    @Override
    public boolean isEnabled() {
        return true;
    }
}
```

El método `loadUserByUsername` devuelve el usuario que accedió por medio de su correo electrónico.

El código usado para el desarrollo del método `loadUserByUsername` es el siguiente:

```
@Service
public class UserDetailsServiceImpl implements UserDetailsService {
    @Autowired
    UserService userService;
    @Override
    public UserDetails loadUserByUsername(String email) throws UsernameNotFoundException {
        Users user = userService.getByEmail(email).get();
        return UserPrincipalDetails.build(user);
    }
}
```

En la clase `JwtEntryPoint` se verifica que solo los usuarios con el token generado por JWT puedan acceder a los recursos.

El código usado para el desarrollo de la clase JwtEntryPoint es el siguiente:

```
@Component
public class JwtEntryPoint implements AuthenticationEntryPoint {
    private final static Logger logger = LoggerFactory.getLogger(JwtEntryPoint.class);
    @Override
    public void commence(HttpServletRequest req, HttpServletResponse res, AuthenticationException e) throws IOException {
        logger.error("fail en el método commence");
        res.sendError(HttpServletResponse.SC_UNAUTHORIZED, "no autorizado");
    }
}
```

En la clase JwtProvider se genera el token de JWT con la información del usuario, esta información es: el correo electrónico del usuario, la fecha de creación del token y los roles que tiene el usuario. Todos los datos obtenidos del usuario se cifran mediante una frase secreta con el algoritmo de una sola vía HS512 (código de autenticación de mensaje basado en hash con algoritmo que convierte el texto de cualquier longitud en una cadena de tamaño fijo).

El código usado para el desarrollo de la clase JwtProvider es el siguiente:

```
@Component
public class JwtProvider {
    private final static Logger logger = LoggerFactory.getLogger(JwtProvider.class);
    @Value("${jwt.secret}")
    private String secret;
    @Value("${jwt.expiration}")
    private int expiration;
    public String generateToken(Authentication authentication) {
        UserPrincipalDetails userPrincipalDetails = (UserPrincipalDetails) authentication.getPrincipal();
        return Jwts.builder().setSubject(userPrincipalDetails.getUsername()).setIssuedAt(new
Date().getTime(), userPrincipalDetails.getAuthorities()).signWith(SignatureAlgorithm.HS512, secret).compact();
    }
    public String getEmailFromToken(String token) {
        return Jwts.parser().setSigningKey(secret).parseClaimsJws(token).getBody().getSubject();
    }
    public boolean validateToken(String token) {
        try {
            Jwts.parser().setSigningKey(secret).parseClaimsJws(token);
            return true;
        } catch (MalformedJwtException e) {
            logger.error("Token mal formado");
        } catch (UnsupportedJwtException e) {
            logger.error("Token no soportado");
        } catch (ExpiredJwtException e) {
            logger.error("Token expirado");
        } catch (IllegalArgumentException e) {
            logger.error("token vacío");
        } catch (SignatureException e) {
            logger.error("Error en la firma del token");
        }
        return false;
    }
}
```

En la clase JwtTokenFilter se verifica el estado el token generado revisando si este no ha sido modificado.

El código usado para el desarrollo de la clase JwtTokenFilteres el siguiente:

```
public class JwtTokenFilter extends OncePerRequestFilter {
    private final static Logger logger = LoggerFactory.getLogger(JwtTokenFilter.class);
    @Autowired
    JwtProvider jwtProvider;
    @Autowired
    UserDetailsServiceImpl userDetailsService;
    @Override
    protected void doFilterInternal(HttpServletRequest req, HttpServletResponse res, FilterChain filterChain) throws
IOException, ServletException {
        try {
            String token = getToken(req);
            if (token != null && jwtProvider.validateToken(token)) {
                String email = jwtProvider.getEmailFromToken(token);
                UserDetails userDetails = userDetailsService.loadUserByUsername(email);
                UsernamePasswordAuthenticationToken auth = new UsernamePasswordAuthenticationToken(userDetails, null,
userDetails.getAuthorities());
                SecurityContextHolder.getContext().setAuthentication(auth);
            }
        }
    }
}
```

```

    } catch (Exception e) {
        logger.error("fail en el método doFilter " + e.getMessage());
    }
    filterChain.doFilter(req, res);
}
private String getToken(HttpServletRequest request) {
    String header = request.getHeader("Authorization");
    if (header != null && header.startsWith("Bearer "))
        return header.replace("Bearer ", "");
    return null;
}
}

```

En la clase MainSecurity se modifica la configuración de Spring Security, donde solo por medio de los roles de los usuarios se la PWA podrán acceder a sus datos.

El código usado para el desarrollo de la clase MainSecurity es el siguiente:

```

@Configuration
@EnableWebSecurity
@EnableGlobalMethodSecurity(prePostEnabled = true)
public class MainSecurity extends WebSecurityConfigurerAdapter {
    @Autowired
    UserDetailsServiceImpl userDetailsService;
    @Autowired
    JwtEntryPoint jwtEntryPoint;
    @Bean
    public JwtTokenFilter jwtTokenFilter() {
        return new JwtTokenFilter();
    }
    @Bean
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        auth.userDetailsService(userDetailsService).passwordEncoder(passwordEncoder());
    }
    @Bean
    @Override
    public AuthenticationManager authenticationManagerBean() throws Exception {
        return super.authenticationManagerBean();
    }
    @Override
    protected AuthenticationManager authenticationManager() throws Exception {
        return super.authenticationManager();
    }
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.cors().and().csrf().disable().authorizeRequests()
            .antMatchers("/auth/**").permitAll()
            .antMatchers("/user/**").hasAnyAuthority("User", "Admin")
            .antMatchers("/pet/**").hasAnyAuthority("User", "Admin")
            .antMatchers("/device/**").hasAnyAuthority("User", "Admin")
            .antMatchers("/species/**").hasAnyAuthority("User", "Admin")
            .antMatchers("/breed/**").hasAnyAuthority("User", "Admin")
            .antMatchers("/medicine/**").hasAnyAuthority("User", "Admin")
            .antMatchers("/professional/**").hasAnyAuthority("User", "Admin")
            .antMatchers("/profession/**").hasAnyAuthority("User", "Admin")
            .antMatchers("/notification/**").hasAnyAuthority("User", "Admin")
            .antMatchers("/medicineType/**").hasAnyAuthority("User", "Admin")
            .antMatchers("/device-detail/**").hasAnyAuthority("User", "Admin")
            .anyRequest().authenticated().and().exceptionHandling()
            .authenticationEntryPoint(jwtEntryPoint).and().sessionManagement()
            .sessionCreationPolicy(SessionCreationPolicy.STATELESS);
        http.addFilterBefore(jwtTokenFilter(), UsernamePasswordAuthenticationFilter.class);
    }
}

```

El método register mediante el método http post crea un nuevo usuario con la información proporcionada por el front-end de la PWA.

El código usado para el desarrollo del método register en el back-end es el siguiente:

```

@PostMapping("/register")
public ResponseEntity<?> register(@Valid @ModelAttribute UserDto userDto, BindingResult bindingResult, @RequestParam(required = false, value = "image") MultipartFile image) throws IOException {
    if (userService.existsByEmail(userDto.getEmail()))
        return new ResponseEntity<>(new Message("El correo electrónico ya existe"), HttpStatus.BAD_REQUEST);
    Users user = new Users();
    if (image == null) {
        user.setImage(Util.extractBytes("https://i.ibb.co/6DX9405/user.png"));
    } else {
        byte[] bytesImg = image.getBytes();
        user.setImage(bytesImg);
    }
    user.setName(userDto.getName());
    user.setLast_name(userDto.getLast_name());
    user.setEmail(userDto.getEmail());
}

```

```

        user.setPassword(passwordEncoder.encode(userDto.getPassword()));
        user.setAddress(userDto.getAddress());
        user.setPhone(userDto.getPhone());
        user.setCreation_date(Util.dateNow());
        Collection<Role> role = new ArrayList<>();
        role.add(roleService.getByRoleName(RoleName.USER).get());
        // role.add(roleService.getByRoleName(RoleName.ROLE_ADMIN).get());
        user.setRole(role);
        userService.createUser(user);
        return new ResponseEntity<>(new Message("usuario guardado"), HttpStatus.CREATED);
    }

```

## Código front-end PWA MonIOpeT Next.js

El método register en el front-end de la PWA envía mediante el método http post la información requerida para la creación de un nuevo usuario.

El código usado para el desarrollo del método register en el front-end es el siguiente:

```

const register = async (
  name: string,
  last_name: string,
  email: string,
  phone: string,
  password: string,
  image: any,
): Promise<{ isComplete: boolean }> => {
  return await petMonitoringApi.post("/auth/register",
    {
      name: name,
      last_name: last_name,
      email: email,
      phone: phone,
      password: password,
      image: image,
    },
    { headers: { "Content-Type": "multipart/form-data" } },
  ).then((response) => {
    const { token, user } = response.data;
    Cookies.set("token", token);
    dispatch({ type: "[Auth] - Login", payload: user });
    swalMessage("Listo", "Registro realizado con éxito", "success");
    return { isComplete: true };
  })
  .catch(() => {
    swalMessage("Error", "Error al registrarse - verifique la información proporcionada", "error");
    return { isComplete: true };
  });
};

```

## Historia de usuario H04 Inicio de sesión

En el inicio de sesión el usuario que va a ingresar a la PWA pasará por el filtro de JWT, donde se verificará la validez de su token y dependiendo de su rol será dirigido al panel que debe gestionar.

## Código back-end Spring Boot

El método getValidateToken verifica si el token con el que ingreso el usuario es válido, devolviendo true o false.

El código usado para el desarrollo del método getValidateToken es el siguiente:

```

@SuppressWarnings({ "unchecked", "rawtypes" })
@GetMapping("/validate-token/{token}")
public ResponseEntity<?> getValidateToken(@PathVariable("token") String token) {
    return new ResponseEntity(jwtProvider.validateToken(token), HttpStatus.OK);
}

```

El método login crea el token con el que iniciará sesión el usuario.

El código usado para el desarrollo del método login en el back-end es el siguiente:

```
@SuppressWarnings({ "unchecked", "rawtypes" })
@PostMapping("/login")
public ResponseEntity<JwtDto> login(@Valid @RequestBody LoginDto loginDto, BindingResult bindingResult) {
    if (bindingResult.hasErrors())
        return new ResponseEntity(new Message("campos mal puestos"), HttpStatus.BAD_REQUEST);
    UsernamePasswordAuthenticationToken authentication = authenticationManager.authenticate(new
    SecurityContextHolder.getContext().setAuthentication(authentication);
    String jwt = jwtProvider.generateToken(authentication);
    UserDetails userDetails = (UserDetails) authentication.getPrincipal();
    JwtDto jwtDto = new JwtDto();
    jwtDto.setToken(jwt);
    jwtDto.setBearer("Bearer");
    jwtDto.setUser(userDetails.getUsername());
    jwtDto.setAuthorities(userDetails.getAuthorities());
    return new ResponseEntity(jwtDto, HttpStatus.OK);
}
```

## Código front-end PWA MonIOpeT Next.js

El método checkToken será llamado por la mayoría de los métodos en el front-end de la PWA, ya que este verifica que el usuario que inicio la sesión no haya modificado su token de JWT.

El código para el desarrollo del método checkToken en el front-end es el siguiente:

```
const checkToken = async (): Promise<{ isComplete: boolean }> => {
    if (!Cookies.get("token")) {
        return { isComplete: true };
    }
    const token = Cookies.get("token") || "";
    return await petMonitoringApi.get(`/auth/validate-token/${token}`)
        .then(async (validate) => {
            if (validate.data === true) {
                const { sub: email } = jwt.decode(token) as { sub: string };
                return await petMonitoringApi.get(`/user/${email}`, {
                    headers: { Authorization: `Bearer ${token}` },
                })
                    .then((user) => {
                        dispatch({ type: "[Auth] - Login", payload: user.data });
                        return { isComplete: true };
                    })
                    .catch(() => {
                        Cookies.remove("token");
                        return { isComplete: true };
                    });
            }
            return { isComplete: true };
        })
        .catch(() => {
            Cookies.remove("token");
            return { isComplete: true };
        });
};
```

El método login envía por el método http post el correo y la contraseña obteniendo como respuesta su token generado en el back-end.

El código para el desarrollo del método login el front-end es el siguiente:

```
const login = async (email: string, password: string): Promise<{ isComplete: boolean }> => {
    return await petMonitoringApi.post("/auth/login", { email, password,
    })
        .then((response) => {
            if (response.data.authorities[0].authority === "Admin") {
                Cookies.set("token", response.data.token);
                checkToken();
                router.replace("/admin", undefined, { shallow: true });
                swalMessage("Bienvenido", "Ingreso realizado con éxito", "success");
            } else if (response.data.authorities[0].authority === "User") {
                Cookies.set("token", response.data.token);
                checkToken();
                router.replace("/users", undefined, { shallow: true });
                swalMessage("Bienvenido", "Ingreso realizado con éxito", "success");
            } else {

```

```

        swalMessage("Error", "Error al ingresar - verifique la información proporcionada", "error");
        router.replace("/auth/login", undefined, { shallow: true });
    }
    return { isComplete: true };
}
});
.catch(() => {
    swalMessage("Error", "Error al ingresar - verifique la información proporcionada", "error");
    return { isComplete: true };
});
});
};

```

Para que los usuarios sean redirigidos a su respectivo panel de gestión ya sea administrador o usuario, los middlewares verifican el rol del usuario antes de que la pagina se genere en el navegador.

Los códigos para el desarrollo de los middlewares del usuario y administrador en el front-end son los siguientes:

```

export async function middleware(req: NextRequest | any, ev: NextFetchEvent) {
    const { token = "" } = req.cookies;
    const url = req.nextUrl.clone();
    try {
        const data = jwt_decode(token) as IToken;
        if (data?.role[0].authority.toString() == "User" || !token) {
            url.pathname = "/auth/login";
            return NextResponse.rewrite(url);
        }
    } catch (error) {
        url.pathname = "/auth/login";
        return NextResponse.rewrite(url);
    }
    return NextResponse.next();
}

```

```

export async function middleware(req: NextRequest | any, ev: NextFetchEvent) {
    const { token = "" } = req.cookies;
    const url = req.nextUrl.clone();
    try {
        const data = jwt_decode(token) as IToken;
        if (data?.role[0].authority.toString() === "Admin" || !token) {
            url.pathname = "/auth/login";
            return NextResponse.rewrite(url);
        }
    } catch (error) {
        url.pathname = "/auth/login";
        return NextResponse.rewrite(url);
    }
    return NextResponse.next();
}

```

## Historia de usuario H05 Olvido de contraseña

En caso de que el usuario se olvide de su contraseña, existe la posibilidad de que se le entregue una contraseña nueva por medio de correo electrónico, la cual podrá ser actualizada en la configuración de la cuenta del usuario.

### Código back-end Spring Boot

El método forgetPassword se encarga de validar que el correo del usuario que olvido su contraseña exista, y al validar la existencia del correo electrónico, el back-end enviará una nueva clave al correo electrónico del usuario.

El código usado para el desarrollo del método forgetPassword en el back-end es el siguiente:

```

@SuppressWarnings({ "unchecked", "rawtypes" })

```

```

@GetMapping("/forget-password/{email}")
public ResponseEntity<> forgetPassword(@PathVariable("email") String email) {
    if (!userService.existsByEmail(email))
        return new ResponseEntity(new Message("El correo electrónico no existe"), HttpStatus.BAD_REQUEST);
    if (StringUtils.isBlank(email))
        return new ResponseEntity(new Message("El correo electrónico es obligatorio"),
        HttpStatus.BAD_REQUEST);
    String password = RandomStringUtils.randomAlphanumeric(6).toUpperCase();
    Users user = userService.getByEmail(email).get();
    user.setPassword(passwordEncoder.encode(password));
    userService.updateUser(user);
    userService.sendEmailUser("server.moniope@gmail.com", user.getEmail(), "Contraseña Olvidada", "Su nueva contraseña
es: " + password);
    return new ResponseEntity<>(new Message("Contraseña enviada"), HttpStatus.CREATED);
}

```

## Código front-end PWA MonIOpeT Next.js

El método forgetPassword en el front-end se encarga de consumir el servicio web tipo REST enviando el correo electrónico del usuario que olvidó su contraseña para ser validada por el back-end.

El código usado para el desarrollo del método forgetPassword en el front-end es el siguiente:

```

const forgetPassword = async (email: string, router: any): Promise<{ isComplete: boolean }> => {
    return await petMonitoringApi
        .get(`/auth/forget-password/${email}`)
        .then(() => {
            swalMessage("Listo", "Se ha enviado un correo con su nueva contraseña", "success");
            router.replace("/auth/login");
            return { isComplete: true };
        })
        .catch(() => {
            swalMessage("Error", "Error al enviar el correo - verifique la información proporcionada", "error");
            return { isComplete: true };
        });
};

```

## Historia de usuario H09 Gestión de dispositivos

Las funciones createDevice y createDeviceDetail son las más relevantes en el desarrollo de la gestión de dispositivos, ya que estos se encargan de crear datos después de pasar por varias validaciones.

## Código back-end Spring Boot

El código usado para el desarrollo del método createDevice encargado de crear un nuevo dispositivo en la PWA es el siguiente:

```

@PostMapping
public ResponseEntity<> createDevice(@RequestBody @Validated DeviceDto deviceDto, BindingResult bindingResult) {
    Device device = new Device();
    if (bindingResult.hasErrors())
        return new ResponseEntity<>(new Message("Campos invalidos"), HttpStatus.BAD_REQUEST);
    if (StringUtils.isBlank(deviceDto.getCode()))
        return new ResponseEntity<>(new Message("El código es necesario"), HttpStatus.BAD_REQUEST);
    if (deviceService.existsByDeviceCode(deviceDto.getCode()))
        return new ResponseEntity<>(new Message("El código no esta disponible"), HttpStatus.BAD_REQUEST);
    device.setCode(deviceDto.getCode());
    device.setCreation_date(Util.dateNow());
    device.setPet(deviceDto.getPet());
    device.setUsers(deviceDto.getUsers());
    deviceService.createDevice(device);
    return ResponseEntity.ok(device.getId());
}

```

## Código back-end Spring Boot de los detalles del dispositivo

El método createDeviceDetail crea y verifica un nuevo detalle con los datos enviados desde un dispositivo de monitoreo.

Las verificaciones comprueban si los datos cumplen con las condiciones necesarias para informar mediante correo electrónico al usuario dueño de la mascota si existe un problema con su mascota o con su dispositivo.

Las verificaciones realizadas a los datos enviados desde el dispositivo de monitoreo son las siguientes:

Si la batería es menor a 2700 voltios se enviará un correo electrónico al usuario dueño de la mascota avisando que la batería esta baja y necesita ser recargada

Las verificaciones de temperatura se determinaron gracias al artículo científico “Investigating the use of non-contact infrared thermometers in cats and dogs” donde las autoras realizaron mediciones de temperatura a gatos y perros con los termómetros infrarrojos Thermofocus y Rycom [52].

Los resultados obtenidos de 54 mediciones de temperatura en gatos determinaron que la temperatura de un gato con el termómetro infrarrojo Thermofocus osciló entre 31,0 y 38,8 °C con una mediana de 36,4 °C y con el termómetro infrarrojo Rycom osciló entre 32,1 y 38,2 °C con una mediana de 35,5 °C [52].

Los resultados obtenidos de 78 mediciones de temperatura en perros determinaron que la temperatura de un perro con el termómetro infrarrojo Thermofocus osciló entre 34,0 y 41,6 °C con una mediana de 38,1 °C y con el termómetro infrarrojo Rycom osciló entre 32,1 y 39,1 °C con una mediana de 37,4 °C [52].

Gracias a los resultados del artículo científico se concluyó que para las verificaciones de temperatura se consideraría las temperaturas de 34,0 a 39,0 °C como normales.

Se concluyó también, que si la temperatura de la mascota es menor a 34,0°C se enviará un correo electrónico al usuario dueño de la mascota avisando que la temperatura de su mascota esta baja y necesita ser revisada, ya que existe el riesgo de que la mascota este sufriendo de hipotermia, la cual es una enfermedad causada por la baja temperatura corporal del animal.



Y si la temperatura de la es mayor a 39,0°C, igualmente se enviará un correo electrónico al usuario dueño de la mascota avisando que la temperatura de su mascota es alta y necesita ser revisada, ya que existe el riesgo de que la mascota este sufriendo de hipertermia, la cual es una enfermedad causada por el aumento de temperatura corporal del animal.

El código usado para el desarrollo del método createDeviceDetail encargado de crear y verificar un nuevo detalle con los datos enviados desde un dispositivo de monitoreo por medio de las funciones callback del back-end de Sigfox en la PWA es el siguiente:

```

@SuppressWarnings({ "unchecked", "rawtypes" })
@PostMapping(produces = "application/json")
public ResponseEntity<?> createDeviceDetail(@RequestBody @Validated DeviceDetailDto deviceDetailDto) {
    DeviceDetail deviceDetail = new DeviceDetail();
    if (deviceDetailDto.getLatitude() > 0) {
        return new ResponseEntity(new Message("latitud no valida"), HttpStatus.BAD_REQUEST);
    }
    if (deviceDetailDto.getLongitude() > 0) {
        return new ResponseEntity(new Message("longitud no valida"), HttpStatus.BAD_REQUEST);
    }
    if (deviceDetailDto.getTemperature() == 0) {
        return new ResponseEntity(new Message("temperatura no valida"), HttpStatus.BAD_REQUEST);
    }
    if (deviceDetailDto.getBattery() == 0) {
        return new ResponseEntity(new Message("bateria no valida"), HttpStatus.BAD_REQUEST);
    }
    deviceDetail.setLatitude(deviceDetailDto.getLatitude());
    deviceDetail.setLongitude(deviceDetailDto.getLongitude());
    deviceDetail.setTemperature(deviceDetailDto.getTemperature());
    deviceDetail.setBattery(deviceDetailDto.getBattery());
    deviceDetail.setCreation_date(Util.dateNow());
    deviceDetail.setDevice(deviceDetailDto.getDevice());
    if (deviceDetailDto.getTemperature() > 39) {
        MessageDto user = deviceService.getUserDevice(deviceDetailDto.getDevice().getId());
        deviceService.sendEmailDevice("server.moniofet@gmail.com", user.getUser_email(), "Temperatura", "AVISO: La temperatura
de su mascota" + user.getPet_name() + " esta alta con riesgo de hipertermia y es de " + deviceDetailDto.getTemperature() + " °C");
        Notification notification = new Notification();
        Users users = userService.getByUserId(user.getUser_id()).get();
        notification.setSubject("Temperatura");
        notification.setText("AVISO: La temperatura de su mascota" + user.getPet_name() + " esta alta y es de " +
deviceDetailDto.getTemperature() + " °C");
        notification.setUsers(users);
        notification.setCreation_date(Util.dateNow());
        notificationService.createNotification(notification);
    }
    if (deviceDetailDto.getTemperature() < 34) {
        MessageDto user = deviceService.getUserDevice(deviceDetailDto.getDevice().getId());
        deviceService.sendEmailDevice("server.moniofet@gmail.com", user.getUser_email(), "Temperatura", "AVISO: La temperatura
de su mascota " + user.getPet_name() + " esta baja con riesgo de hipotermia y es de " + deviceDetailDto.getTemperature() + " °C");
        Notification notification = new Notification();
        Users users = userService.getByUserId(user.getUser_id()).get();
        notification.setSubject("Temperatura");
        notification.setText("AVISO: La temperatura de su mascota " + user.getPet_name() + " esta baja y es de " +
deviceDetailDto.getTemperature() + " °C");
        notification.setUsers(users);
        notification.setCreation_date(Util.dateNow());
        notificationService.createNotification(notification);
    }
    if (deviceDetailDto.getBattery() <= 2700) {
        MessageDto user = deviceService.getUserDevice(deviceDetailDto.getDevice().getId());
        deviceService.sendEmailDevice("server.moniofet@gmail.com", user.getUser_email(), "Bateria", "AVISO: La bateria del
dispositivo de su mascota " + user.getPet_name() + " esta baja, cargue el dispositivo");
        Notification notification = new Notification();
        Users users = userService.getByUserId(user.getUser_id()).get();
        notification.setSubject("Bateria");
        notification.setText("AVISO: La bateria del dispositivo de su mascota " + user.getPet_name() + " esta baja, cargue el
dispositivo");
        notification.setUsers(users);
        notification.setCreation_date(Util.dateNow());
        notificationService.createNotification(notification);
    }
    deviceDetailService.createDeviceDetail(deviceDetail);
    return ResponseEntity.ok(deviceDetail);
}

```

## Código front-end PWA MonIOpeT Next.js

En el front-end de la PWA por medio de los métodos http get, post, put y delete se realiza la gestión de dispositivos realizando la transferencia de datos con el back-end.

El servicio web tipo REST proporcionado por Sigfox tiene como medida de seguridad intermediaria implementado un servidor proxy por lo que es necesario realizar una configuración en el front-end de la PWA.

El archivo next.config.js implementada la configuración es el siguiente:

```
const withPWA = require("next-pwa");

module.exports = withPWA({
  reactStrictMode: true,
  pwa: {
    dest: "public",
    register: true,
    skipWaiting: true,
    disable: process.env.NODE_ENV === "development",
  },
  async rewrites() {
    return [
      {
        source: "/v2/:slug*",
        destination: `${process.env.API_SIGFOX}/v2/:slug`,
      },
    ];
  },
});
```

El código usado para el desarrollo del método addDevice encargado de enviar por el método http post la información de un dispositivo para ser creado es el siguiente:

```
const addDevice = async (code: string, user: number, pet: number, clearForm: ()=>void): Promise<{ isComplete: boolean }> => {
  const token = Cookies.get("token") || "";
  let id: any;
  return await petMonitoringApi
    .post(
      '/device',
      {
        code,
        users: { id: user },
        pet: { id: pet },
      },
      {
        headers: {
          Authorization: `Bearer ${token}`,
        },
      },
    )
    .then(async (device) => {
      id = device.data;
      if (device.data) {
        await sigfoxCallbackApi
          .post('/device-types/${code}/callbacks', {
            channel: "URL",
            callbackType: 0,
            callbackSubtype: 2,
            payloadConfig: "lat::float:32 long::float:32 temp::int:16 bat::uint:16",
            enabled: true,
            url: `${process.env.NEXT_PUBLIC_API_PET_MONITORING}/device-detail`,
            httpMethod: "POST",
            headers: {
              Authorization: `Bearer ${token}`,
            },
            sendSni: false,
            bodyTemplate:
              '{\n"latitude":{customData#lat},\n"longitude":{customData#long},\n"temperature":{customData#temp},\n"battery":{customData#bat},\n"device":{"id":' +
                device.data +
                "}"\n}',
            contentType: "application/json",
          })
          .then(async (callback) => {
            if (callback.data) {
              await petMonitoringApi.put(
                '/device/${id}',
                {
                  callback_code: callback.data.id,
                },
                {
                  headers: {
                    Authorization: `Bearer ${token}`,
                  },
                },
              )
              .then(async (message) => {
                if (message.data) {
                  checkToken();
                  clearForm();
                  swalMessage("Success", "Dispositivo agregado", "success");
                }
              });
            }
          })
      }
    });
}
```

```

    .catch(async () => {
      await petMonitoringApi.delete(`/device/${id}`, {
        headers: {
          Authorization: `Bearer ${token}`,
        },
      });
      swalMessage("Error", "Error al agregar el dispositivo - revise su código", "error");
    });
  }
  return { isComplete: true };
})
.catch(async () => {
  swalMessage("Error", "Error al agregar el dispositivo - revise su código", "error");
  return { isComplete: true };
});
};

```

### 3.2.3.7 Fase de pruebas

La fase de codificación es la cuarta fase de la metodología XP, en esta fase se realizan las pruebas necesarias para determinar el funcionamiento exitoso de la PWA.

Las pruebas que se van a realizar son: pruebas de aceptación según las historias de usuario determinadas, pruebas de rendimiento de la PWA y pruebas de usabilidad por parte del administrador y el usuario dueño de la mascota.

#### Pruebas de aceptación

Prueba de Aceptación	
<b>Código:</b> P01	<b>Historia de usuario:</b> H01
<b>Nombre:</b> Estructura del proyecto	
<b>Descripción:</b> Organización de las carpetas para el desarrollo del back-end y front-end de la PWA	
<b>Condiciones de ejecución:</b> Creación de los proyectos back-end y front-end	
<b>Entrada:</b> Carpetas creadas por defecto	
<b>Resultado esperado:</b> Organizar las carpetas de los proyectos back-end y front-end aumentando el número de carpetas y modificando archivos por defecto para hacer más sencillo el desarrollo de la PWA.	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.114 Prueba de aceptación P01 Estructura del proyecto  
Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P02	<b>Historia de usuario:</b> H02
<b>Nombre:</b> Diseño de la base de datos	
<b>Descripción:</b> Diseñar una base de datos que cumpla con los requerimientos del proyecto	
<b>Condiciones de ejecución:</b> Iniciar proyecto en Spring Boot	
<b>Entrada:</b> Entidades creadas en Spring Boot	
<b>Resultado esperado:</b> Base de datos diseñada en PostgreSQL	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.115 Prueba de aceptación P02 Diseño de la base de datos  
Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P03	<b>Historia de usuario:</b> H03
<b>Nombre:</b> Registro de usuarios	
<b>Descripción:</b> Un nuevo usuario dueño de la mascota se registra en la PWA MonIOpeT.	
<b>Condiciones de ejecución:</b> El usuario dueño de la mascota no debe estar registrado en la PWA	
<b>Entrada:</b> El usuario dueño de la mascota ingresa su información en un formulario junto a una contraseña de 6 caracteres.	
<b>Resultado esperado:</b> Al completarse el registro del usuario dueño de la mascota se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.116 Prueba de aceptación P03 Registro de usuarios

Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P04	<b>Historia de usuario:</b> H04
<b>Nombre:</b> Inicio de sesión	
<b>Descripción:</b> Algún usuario registrado en la PWA MonIOpeT inicia su sesión.	
<b>Condiciones de ejecución:</b> El usuario debe estar registrado en la PWA	
<b>Entrada:</b> El usuario ingresa su correo electrónico en un formulario junto a su contraseña de 6 caracteres.	
<b>Resultado esperado:</b> Al ingreso a la PWA del usuario se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.117 Prueba de aceptación P04 Inicio de sesión

Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P05	<b>Historia de usuario:</b> H05
<b>Nombre:</b> Olvido de contraseña	
<b>Descripción:</b> El usuario dueño de la mascota en caso de olvidar su contraseña la PWA envía una nueva su correo electrónico registrado en la PWA.	
<b>Condiciones de ejecución:</b> El usuario debe estar registrado en la PWA	
<b>Entrada:</b> El usuario ingresa su correo electrónico en un formulario.	
<b>Resultado esperado:</b> Al enviarse la nueva contraseña al correo electrónico del usuario se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.118 Prueba de aceptación P05 Olvido de contraseña

Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P06	<b>Historia de usuario:</b> H06
<b>Nombre:</b> Panel de administración	
<b>Descripción:</b> El usuario con el rol administrador debe iniciar su sesión en la PWA MonIOpeT para ver las opciones que tiene para administrar y ver información de los datos de la PWA	
<b>Condiciones de ejecución:</b> El usuario con el rol administrador debió iniciar sesión en la PWA	

<b>Entrada:</b> El usuario con el rol administrador ve una interfaz con el menú de opciones a administrar.
<b>Resultado esperado:</b> El usuario puede ingresar a la gestión de las entidades que intervienen en la PWA.
<b>Evaluación de prueba:</b> Satisfactoria

**Tabla 3.119 Prueba de aceptación P06 Panel de administración**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P07	<b>Historia de usuario:</b> H07
<b>Nombre:</b> Gestión de usuarios	
<b>Descripción:</b> El usuario administrador puede gestionar la información de todos los usuarios de la PWA	
<b>Condiciones de ejecución:</b> El usuario debe tener el rol administrador.	
<b>Entrada:</b> El usuario administrador puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de los usuarios	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de los usuarios se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.120 Prueba de aceptación P07 Gestión de usuarios**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P08	<b>Historia de usuario:</b> H08
<b>Nombre:</b> Gestión de mascotas	
<b>Descripción:</b> El usuario administrador puede gestionar la información de todas las mascotas de la PWA	
<b>Condiciones de ejecución:</b> El usuario debe tener el rol administrador.	
<b>Entrada:</b> El usuario administrador puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de las mascotas.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de las mascotas se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.121 Prueba de aceptación P08 Gestión de mascotas**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P09	<b>Historia de usuario:</b> H09
<b>Nombre:</b> Gestión de dispositivos	
<b>Descripción:</b> El usuario administrador puede gestionar la información de todos los dispositivos de la PWA	
<b>Condiciones de ejecución:</b> El usuario debe tener el rol administrador.	
<b>Entrada:</b> El usuario administrador puede realizar las siguientes actividades: ver, guardar y eliminar por medio de formularios la información de los dispositivos.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de los dispositivos se muestra una alerta exitosa.	

**Evaluación de prueba:** Satisfactoria

**Tabla 3.122 Prueba de aceptación P09 Gestión de dispositivos**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P10	<b>Historia de usuario:</b> H10
<b>Nombre:</b> Gestión de establecimientos	
<b>Descripción:</b> El usuario administrador puede gestionar la información de todos los establecimientos de la PWA	
<b>Condiciones de ejecución:</b> El usuario debe tener el rol administrador.	
<b>Entrada:</b> El usuario administrador puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de los establecimientos.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de los establecimientos se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.123 Prueba de aceptación P10 Gestión de establecimientos**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P11	<b>Historia de usuario:</b> H11
<b>Nombre:</b> Gestión de medicinas	
<b>Descripción:</b> El usuario administrador puede gestionar la información de todas las medicinas de la PWA	
<b>Condiciones de ejecución:</b> El usuario debe tener el rol administrador.	
<b>Entrada:</b> El usuario administrador puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de las medicinas.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de las medicinas se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.124 Prueba de aceptación P11 Gestión de medicinas**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P12	<b>Historia de usuario:</b> H12
<b>Nombre:</b> Gestión de profesionales	
<b>Descripción:</b> El usuario administrador puede gestionar la información de todos los profesionales de la PWA	
<b>Condiciones de ejecución:</b> El usuario debe tener el rol administrador.	
<b>Entrada:</b> El usuario administrador puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de los profesionales.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de los profesionales se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.125 Prueba de aceptación P12 Gestión de profesionales**  
Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P13	<b>Historia de usuario:</b> H13
<b>Nombre:</b> Gestión de especies	
<b>Descripción:</b> El usuario administrador puede gestionar la información de todas las especies de la PWA	
<b>Condiciones de ejecución:</b> El usuario debe tener el rol administrador.	
<b>Entrada:</b> El usuario administrador puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de las especies.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de las especies se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.126 Prueba de aceptación P13 Gestión de especies

Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P14	<b>Historia de usuario:</b> H14
<b>Nombre:</b> Gestión de razas	
<b>Descripción:</b> El usuario administrador puede gestionar la información de todas las razas de la PWA	
<b>Condiciones de ejecución:</b> El usuario debe tener el rol administrador.	
<b>Entrada:</b> El usuario administrador puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de las razas.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de las razas se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.127 Prueba de aceptación P14 Gestión de razas

Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P15	<b>Historia de usuario:</b> H15
<b>Nombre:</b> Gestión de tipos de establecimientos	
<b>Descripción:</b> El usuario administrador puede gestionar la información de todos los tipos de establecimientos de la PWA	
<b>Condiciones de ejecución:</b> El usuario debe tener el rol administrador.	
<b>Entrada:</b> El usuario administrador puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de los tipos de establecimientos.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de los tipos de establecimientos se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.128 Prueba de aceptación P15 Gestión de tipos de establecimientos

Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P16	<b>Historia de usuario:</b> H16
<b>Nombre:</b> Gestión de tipos de medicinas	

<b>Descripción:</b> El usuario administrador puede gestionar la información de todos los tipos de medicinas de la PWA
<b>Condiciones de ejecución:</b> El usuario debe tener el rol administrador.
<b>Entrada:</b> El usuario administrador puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de los tipos de medicinas.
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de los tipos de medicinas se muestra una alerta exitosa.
<b>Evaluación de prueba:</b> Satisfactoria

**Tabla 3.129 Prueba de aceptación P16 Gestión de tipos de medicinas**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P17	<b>Historia de usuario:</b> H17
<b>Nombre:</b> Gestión de profesiones	
<b>Descripción:</b> El usuario administrador puede gestionar la información de todas las profesiones de la PWA	
<b>Condiciones de ejecución:</b> El usuario debe tener el rol administrador.	
<b>Entrada:</b> El usuario administrador puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de las profesiones.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de las profesiones se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.130 Prueba de aceptación P17 Gestión de profesiones**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P18	<b>Historia de usuario:</b> H18
<b>Nombre:</b> Panel de usuario	
<b>Descripción:</b> El dueño de la mascota con el rol usuario debe iniciar su sesión en la PWA MonIOpeT para ver las opciones que tiene para administrar y monitorear la información de su mascota en la PWA.	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA	
<b>Entrada:</b> El usuario ve una interfaz con el menú de opciones a administrar.	
<b>Resultado esperado:</b> El usuario puede actualizar su información, gestionar la información de sus mascotas y dispositivos, además ver y gestionar las secciones de monitoreo de su mascota.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.131 Prueba de aceptación P18 Panel de usuario**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P19	<b>Historia de usuario:</b> H19
<b>Nombre:</b> Configuración cuenta de usuario	
<b>Descripción:</b> El dueño de la mascota con el rol usuario puede actualizar la información con la que se registró en la PWA MonIOpeT	



<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA
<b>Entrada:</b> El usuario puede ingresar su nueva información en los formularios de actualización de información.
<b>Resultado esperado:</b> Al completarse alguna actualización de información del usuario se muestra una alerta exitosa.
<b>Evaluación de prueba:</b> Satisfactoria

Tabla 3.132 Prueba de aceptación P19 Configuración cuenta de usuario  
Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P20	<b>Historia de usuario:</b> H20
<b>Nombre:</b> Gestión de las mascotas del usuario	
<b>Descripción:</b> El dueño de la mascota con el rol usuario puede gestionar la información de sus mascotas ingresadas en la PWA MonIOpeT	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA	
<b>Entrada:</b> El usuario puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de sus mascotas.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de las mascotas del usuario se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.133 Prueba de aceptación P20 Gestión de las mascotas del usuario  
Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P21	<b>Historia de usuario:</b> H21
<b>Nombre:</b> Gestión de dispositivos del usuario	
<b>Descripción:</b> El dueño de la mascota con el rol usuario puede gestionar la información de sus dispositivos de monitoreo ingresados en la PWA MonIOpeT	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA	
<b>Entrada:</b> El usuario puede realizar las siguientes actividades: ver, guardar y eliminar por medio de formularios la información de su dispositivo asignado a una mascota.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de los dispositivos del usuario se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.134 Prueba de aceptación P21 Gestión de dispositivos del usuario  
Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P22	<b>Historia de usuario:</b> H22
<b>Nombre:</b> Secciones de monitorización	
<b>Descripción:</b> El dueño de la mascota con el rol usuario debe ingresar a la opción de monitoreo de su mascota para ver las secciones que puede gestionar.	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA y elegir la opción de monitoreo de su mascota	

<b>Entrada:</b> El usuario en la interfaz de secciones de su mascota verá las opciones a gestionar.
<b>Resultado esperado:</b> El usuario ver las secciones de monitoreo de su mascota que lo redireccionará a sus respectivas páginas de gestión.
<b>Evaluación de prueba:</b> Satisfactoria

**Tabla 3.135 Prueba de aceptación P22 Secciones de monitorización**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P23	<b>Historia de usuario:</b> H23
<b>Nombre:</b> Ubicación de la mascota del usuario en tiempo real	
<b>Descripción:</b> El dueño de la mascota con el rol puede ver en tiempo real la ubicación de su mascota y compartirla.	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA y elegir la opción de ubicación en las secciones de monitoreo de su mascota	
<b>Entrada:</b> El usuario encenderá su dispositivo de monitoreo asignado a una mascota.	
<b>Resultado esperado:</b> El usuario observara la ubicación de su mascota enviada desde el dispositivo de monitoreo, la información obtenida es posible compartirla en WhatsApp o en correo electrónico.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.136 Prueba de aceptación P23 Ubicación de la mascota del usuario en tiempo real**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P24	<b>Historia de usuario:</b> H24
<b>Nombre:</b> Historial de ubicaciones de la mascota del usuario en tiempo real	
<b>Descripción:</b> El dueño de la mascota con el rol puede ver en tiempo real el historial de ubicaciones de su mascota y revisar cada ubicación.	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA y elegir la opción de ubicación en las secciones de monitoreo de su mascota	
<b>Entrada:</b> El usuario deberá tener ubicaciones guardadas de su mascota.	
<b>Resultado esperado:</b> El usuario observará todas las ubicaciones de su mascota por su fecha de ingreso y revisarlas en Google Maps.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.137 Prueba de aceptación P24 Historial de ubicaciones de la mascota del usuario en tiempo real**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P25	<b>Historia de usuario:</b> H25
<b>Nombre:</b> Temperatura de la mascota del usuario en tiempo real	
<b>Descripción:</b> El dueño de la mascota con el rol puede ver en tiempo real la temperatura de su mascota y compartirla.	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA y elegir la opción de temperatura en las secciones de monitoreo de su mascota	

<b>Entrada:</b> El usuario encenderá su dispositivo de monitoreo asignado a una mascota.
<b>Resultado esperado:</b> El usuario observará la temperatura de su mascota enviada desde el dispositivo de monitoreo, la información obtenida es posible compartirla en WhatsApp o en correo electrónico.
<b>Evaluación de prueba:</b> Satisfactoria

Tabla 3.138 Prueba de aceptación P25 Temperatura de la mascota del usuario en tiempo real  
Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P26	<b>Historia de usuario:</b> H26
<b>Nombre:</b> Historial de temperaturas de la mascota del usuario en tiempo real	
<b>Descripción:</b> El dueño de la mascota con el rol puede ver en tiempo real el historial de temperaturas de su mascota.	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA y elegir la opción de temperatura en las secciones de monitoreo de su mascota	
<b>Entrada:</b> El usuario deberá tener temperaturas guardadas de su mascota.	
<b>Resultado esperado:</b> El usuario observará todas las temperaturas de su mascota por su fecha de ingreso.	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.139 Prueba de aceptación P26 Historial de temperaturas de la mascota del usuario en tiempo real  
Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P27	<b>Historia de usuario:</b> H27
<b>Nombre:</b> Gestión de establecimientos de la mascota	
<b>Descripción:</b> El dueño de la mascota con el rol usuario puede gestionar la información de los establecimientos que frecuenta su mascota ingresada en la PWA MonIOpeT	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA y elegir la opción de establecimientos en las secciones de monitoreo de su mascota	
<b>Entrada:</b> El usuario puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de los establecimientos que frecuenta su mascota.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de los establecimientos de su mascota se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

Tabla 3.140 Prueba de aceptación P27 Gestión de establecimientos de la mascota  
Elaborado por: El investigador

Prueba de Aceptación	
<b>Código:</b> P28	<b>Historia de usuario:</b> H28
<b>Nombre:</b> Contactar establecimiento de la mascota	
<b>Descripción:</b> El dueño de la mascota con el rol usuario puede contactar al establecimiento que frecuenta su mascota ingresada en la PWA MonIOpeT	

<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA y elegir la opción de ver establecimiento en la sección de establecimientos en la opción de monitoreo de su mascota.
<b>Entrada:</b> El usuario debe elegir entre contactar al establecimiento por medio de WhatsApp o correo electrónico e ingresar el mensaje de contacto o editar el presentado por defecto
<b>Resultado esperado:</b> El mensaje de contacto al establecimiento se envía por WhatsApp o correo electrónico.
<b>Evaluación de prueba:</b> Satisfactoria

**Tabla 3.141 Prueba de aceptación P28 Contactar establecimiento de la mascota**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P29	<b>Historia de usuario:</b> H29
<b>Nombre:</b> Gestión de medicinas de la mascota	
<b>Descripción:</b> El dueño de la mascota con el rol usuario puede gestionar la información de las medicinas que usa su mascota ingresada en la PWA MonIOpeT	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA y elegir la opción de medicinas en las secciones de monitoreo de su mascota	
<b>Entrada:</b> El usuario puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de las medicinas que usa su mascota.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de las medicinas de su mascota se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.142 Prueba de aceptación P29 Gestión de medicinas de la mascota**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P30	<b>Historia de usuario:</b> H30
<b>Nombre:</b> Gestión de profesionales de la mascota	
<b>Descripción:</b> El dueño de la mascota con el rol usuario puede gestionar la información de los profesionales que visita su mascota ingresada en la PWA MonIOpeT	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA y elegir la opción de profesionales en las secciones de monitoreo de su mascota	
<b>Entrada:</b> El usuario puede realizar las siguientes actividades: ver, guardar, actualizar y eliminar por medio de formularios la información de los profesionales que visita su mascota.	
<b>Resultado esperado:</b> Al completarse alguna de las actividades de gestión de los profesionales de su mascota se muestra una alerta exitosa.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.143 Prueba de aceptación P30 Gestión de profesionales de la mascota**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P31	<b>Historia de usuario:</b> H31
<b>Nombre:</b> Contactar profesional de la mascota	
<b>Descripción:</b> El dueño de la mascota con el rol usuario puede contactar al profesional que visita su mascota ingresada en la PWA MonIOpeT	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA y elegir la opción de ver profesional en la sección de profesionales en la opción de monitoreo de su mascota.	
<b>Entrada:</b> El usuario debe elegir entre contactar al profesional por medio de WhatsApp o correo electrónico e ingresar el mensaje de contacto o editar el presentado por defecto	
<b>Resultado esperado:</b> El mensaje de contacto al profesional se envía por WhatsApp o correo electrónico.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.144 Prueba de aceptación P31 Contactar profesional de la mascota**  
Elaborado por: El investigador

<b>Prueba de Aceptación</b>	
<b>Código:</b> P32	<b>Historia de usuario:</b> H32
<b>Nombre:</b> Consulta de mascotas por secciones	
<b>Descripción:</b> El dueño de la mascota con el rol usuario puede consultar que mascotas cumplen con las características de cada sección que administra.	
<b>Condiciones de ejecución:</b> El dueño de la mascota con el rol usuario debió iniciar sesión en la PWA	
<b>Entrada:</b> El usuario ingresará el nombre y tipo de cada sección de administración	
<b>Resultado esperado:</b> Al completarse la consulta se devolverá la mascota que cumple con los datos de la sección ingresada.	
<b>Evaluación de prueba:</b> Satisfactoria	

**Tabla 3.145 Prueba de aceptación P32 Consulta de mascotas por secciones**  
Elaborado por: El investigador

## **Pruebas de rendimiento**

Las pruebas de rendimiento de la PWA se las realizará en la herramienta desarrollada en el lenguaje Java JMeter. El cual es un proyecto de Apache que cuenta con funciones que ayudan a realizar pruebas de carga, estrés y estabilidad de aplicaciones Web

### **Prueba de carga realizada a la PWA MonIOpeT**

Para realizar la prueba de carga, se realizó una muestra de las actividades que realizaría un usuario en la PWA

Reporte resumen											
Nombre:		Reporte resumen									
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo		Navegar...		Log/Mostrar sólo:		Escribir en Log Sólo Errores		Éxitos		Configurar	
% Error	Etiqueta	# Muestras	Media	Min	Max	Desv. Estándar	Rendimiento	Kb/sec	Sent KB/sec	Media de Byt...	
0,00%	-10	1	3	3	3	0,00	333,3/sec	638,67	121,74	1962,0	
0,00%	-11	1	2	2	2	0,00	500,0/sec	2325,68	185,06	4763,0	
0,00%		13	3	2	7	1,59	309,5/sec	875,44	157,62	2896,2	
0,00%	-12	1	2	2	2	0,00	500,0/sec	67,38	167,48	138,0	
0,00%	-13	1	3	3	3	0,00	333,3/sec	44,92	114,58	138,0	
0,00%	-14	1	1	1	1	0,00	1000,0/sec	134,77	343,75	138,0	
0,00%	-15	1	1	1	1	0,00	1000,0/sec	134,77	354,49	138,0	
0,00%	-17	1	2	2	2	0,00	500,0/sec	67,38	175,78	138,0	
0,00%	-20	1	2	2	2	0,00	500,0/sec	67,38	190,92	138,0	
0,00%	-28	1	2	2	2	0,00	500,0/sec	67,38	182,38	138,0	
0,00%	-29	1	2	2	2	0,00	500,0/sec	5242,19	191,89	10736,0	
0,00%	-30	1	2	2	2	0,00	500,0/sec	67,38	175,78	138,0	
0,00%	-34	1	3	3	3	0,00	333,3/sec	44,92	127,28	138,0	
0,00%	-35	1	2	2	2	0,00	500,0/sec	67,38	188,48	138,0	
0,00%	-36	1	2	2	2	0,00	500,0/sec	3472,17	209,47	7111,0	
0,00%	-37	1	2	2	2	0,00	500,0/sec	67,38	190,43	138,0	
0,00%	-61	1	2	2	2	0,00	500,0/sec	67,38	171,39	138,0	
0,00%	-62	1	5	5	5	0,00	200,0/sec	1300,78	75,78	6660,0	
0,00%	-63	1	2	2	2	0,00	500,0/sec	2325,68	187,01	4763,0	
0,00%	Total	31	2	1	7	1,29	738,1/sec	1750,88	315,24	2429,1	

**Figura 3.155 Resultados de la prueba de carga realizada a la PWA MonIOpeT**  
Elaborado por: El investigador

En la figura 3.155 se puede observar que el tiempo promedio total de las solicitudes http con una muestra es de 2 ms, es decir 0.002 segundos y con 0% de errores.

### Prueba de estrés realizada a la PWA MonIOpeT

Para realizar la prueba de estrés, se realizaron 1000 muestras de las actividades de un usuario en la PWA.

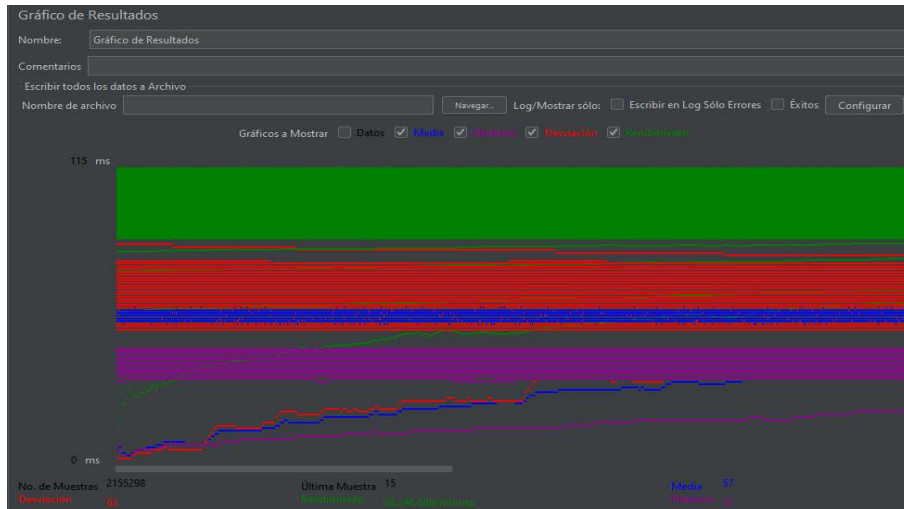
Reporte resumen											
Nombre:		Reporte resumen									
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo		Navegar...		Log/Mostrar sólo:		Escribir en Log Sólo Errores		Éxitos		Configurar	
% Error	Etiqueta	# Muestras	Media	Min	Max	Desv. Estándar	Rendimiento	Kb/sec	Sent KB/sec	Media de Byt...	
0,00%	-10	1000	2154	10	5185	927,49	95,2/sec	182,38	34,76	1962,0	
0,00%	-11	1000	1989	8	3107	770,33	86,2/sec	401,12	31,82	4763,0	
0,00%		13000	1613	7	8884	1388,30	495,7/sec	1402,10	252,44	2896,2	
0,00%	-12	1000	789	17	1513	278,88	75,9/sec	10,23	25,41	138,0	
0,00%	-13	1000	848	7	2169	367,01	64,9/sec	8,73	22,32	138,0	
0,00%	-14	1000	719	30	1363	188,78	63,2/sec	8,52	21,73	138,0	
0,00%	-15	1000	744	25	1804	243,18	56,9/sec	7,66	20,16	138,0	
0,00%	-17	1000	894	90	2167	499,47	50,0/sec	7,28	18,99	138,0	
0,00%	-20	1000	833	99	1892	240,33	50,0/sec	6,83	19,41	138,0	
0,00%	-28	1000	832	129	1794	271,64	49,0/sec	6,65	18,99	138,0	
0,00%	-29	1000	2152	797	6555	1350,51	47,1/sec	493,83	18,68	10736,0	
0,00%	-30	1000	832	140	2162	288,13	48,3/sec	6,54	17,05	138,0	
0,00%	-34	1000	1081	1	2157	481,07	48,3/sec	6,51	18,46	138,0	
0,00%	-35	1000	837	2	2175	491,57	49,0/sec	6,66	18,63	138,0	
0,00%	-36	1000	948	282	2414	376,06	50,0/sec	349,87	21,11	7111,0	
0,00%	-37	1000	863	78	1882	479,87	56,5/sec	7,62	21,53	138,0	
0,00%	-61	1000	806	12	1848	376,27	58,7/sec	7,91	20,11	138,0	
0,00%	-62	1000	3120	633	7039	1658,53	55,6/sec	361,83	21,08	6660,0	
0,00%	-63	1000	910	3	6400	974,73	71,6/sec	333,19	26,79	4763,0	
0,00%	Total	31000	1352	1	9884	1179,01	1182,1/sec	2305,20	504,89	2429,1	

**Figura 3.156 Resultados de la prueba de estrés realizada a la PWA MonIOpeT**  
Elaborado por: El investigador

En la figura 3.156 se puede observar que, el tiempo promedio total de las solicitudes http con 1000 muestras es de 1352 ms es decir 1.352 segundos y con 0% de errores.

### Prueba de estabilidad realizada a la PWA MonIOpeT

Para realizar la prueba de estabilidad, se realizaron 50 muestras de las actividades de un usuario en la PWA ejecutándose constantemente.

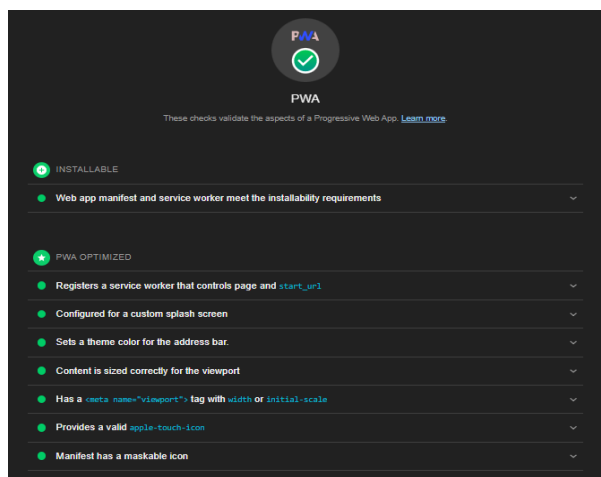


**Figura 3.157 Resultados de la prueba de estabilidad realizada a la PWA MonIOpeT**  
Elaborado por: El investigador

En la figura 3.157 se puede observar que en un tiempo de 30 minutos con 50 muestras ejecutándose constantemente el rendimiento de color verde, donde se procesan 96.246,088 solicitudes por minuto es estable.

### **Prueba de rendimiento realizada con la herramienta de Google Lighthouse a la PWA MonIOpeT**

Google proporciona una herramienta para determinar el rendimiento de una PWA indicando si está preparada a ser usada en producción.



**Figura 3.158 Resultados de la prueba de rendimiento realizada con la herramienta de Google Lighthouse a la PWA MonIOpeT.**  
Elaborado por: El investigador

En la figura 3.158 se puede observar que como resultado la PWA es instalable en dispositivos y esta optimizada.

## Pruebas de usabilidad

Se realizaron pruebas de usabilidad dirigido hacia los usuarios a la cual está dirigida la PWA MonIOpeT.

Se utilizó la siguiente escala de Likert para medir el nivel de satisfacción de los usuarios.

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

A continuación, se muestra los resultados obtenidos de las pruebas de usabilidad realizadas.

<b>Resultados prueba de usabilidad</b>	
<b>Tipo/Rol:</b> Administrador	<b>Código:</b> PU01
<b>Preguntas</b>	<b>Respuestas</b>
¿La PWA MonIOpeT le permite realizar la gestión de todos los datos de forma rápida?	Totalmente de Acuerdo
¿La interfaz de la PWA MonIOpeT le parece agradable?	Totalmente de Acuerdo
¿Las secciones de la PWA MonIOpeT están organizadas?	Totalmente de Acuerdo
¿La PWA MonIOpeT le parece fácil de usar?	Totalmente de Acuerdo
¿El tiempo de respuesta de las acciones de la PWA MonIOpeT son rápidas?	Totalmente de Acuerdo
¿La PWA MonIOpeT le muestra mensajes si alguna acción se realizó con éxito?	Totalmente de Acuerdo

**Tabla 3.146 Prueba de usabilidad PU01 Administrador**  
Elaborado por: El investigador

<b>Resultados prueba de usabilidad</b>	
<b>Tipo/Rol:</b> Usuario	<b>Código:</b> PU02
<b>Preguntas</b>	<b>Respuestas</b>



¿La PWA MonIOpeT permite la actualización de su información en caso de necesitarlo?	Totalmente de Acuerdo
¿La PWA MonIOpeT le permite realizar la gestión de la información de sus mascotas y dispositivos de forma rápida?	Totalmente de Acuerdo
¿La PWA MonIOpeT le permite realizar las actividades de monitoreo de su mascota de forma rápida?	Totalmente de Acuerdo
¿La interfaz de la PWA MonIOpeT le parece agradable?	Totalmente de Acuerdo
¿Las secciones de la PWA MonIOpeT están organizadas?	Totalmente de Acuerdo
La PWA MonIOpeT le parece fácil de usar	Totalmente de Acuerdo
¿El tiempo de respuesta de las acciones de la PWA MonIOpeT son rápidas?	Totalmente de Acuerdo
¿La PWA MonIOpeT le muestra mensajes si alguna acción se realizó con éxito?	Totalmente de Acuerdo
¿Recomendaría a otra persona usar la PWA MonIOpeT?	Totalmente de Acuerdo

**Tabla 3.147 Prueba de usabilidad PU02 Usuario**  
**Elaborado por: El investigador**

<b>Resultados prueba de usabilidad</b>	
<b>Tipo/Rol:</b> Usuario	<b>Codigo:</b> PU03
<b>Preguntas</b>	<b>Respuestas</b>
¿La PWA MonIOpeT permite la actualización de su información en caso de necesitarlo?	Totalmente de Acuerdo
¿La PWA MonIOpeT le permite realizar la gestión de la información de sus mascotas y dispositivos de forma rápida?	Totalmente de Acuerdo
¿La PWA MonIOpeT le permite realizar las actividades de monitoreo de su mascota de forma rápida?	Totalmente de Acuerdo
¿La interfaz de la PWA MonIOpeT le parece agradable?	Totalmente de Acuerdo
¿Las secciones de la PWA MonIOpeT están organizadas?	Totalmente de Acuerdo
La PWA MonIOpeT le parece fácil de usar	De acuerdo
¿El tiempo de respuesta de las acciones de la PWA MonIOpeT son rápidas?	Totalmente de Acuerdo
¿La PWA MonIOpeT le muestra mensajes si alguna acción se realizó con éxito?	Totalmente de Acuerdo

¿Recomendaría a otra persona usar la PWA MonIOpeT?	Totalmente de Acuerdo
--	-----------------------

**Tabla 3.148 Prueba de usabilidad PU03 Usuario**  
**Elaborado por: El investigador**

Gracias a los resultados de las tablas 3.146, 3.147 y 3.148 se puede concluir que la PWA MonIOpeT para los roles administrador y usuario la gestión de información es rápida, las interfaces diseñadas son agradables, los tiempos de respuesta son rápidas, la organización de las secciones son las adecuadas y es fácil de usar.

### **3.2.3.8 Fase de producción**

La fase de producción es la quinta y última fase de la metodología XP, en esta fase el back-end y el front-end desarrollado y probado de la PWA MonIOpeT se despliega en la Web gracias a plataformas dedicadas a organizar, optimizar y presentar la PWA a sus clientes.

El back-end en el cual se desarrollaron los servicios web tipo REST en la herramienta Spring Boot se desplegará en la plataforma AWS, donde gracias al servicio Elastic Beanstalk permite el despliegue de aplicaciones con simples configuraciones.

El front-end en el cual se desarrolló la PWA MonIOpeT en Next.js, se desplegará en la plataforma Vercel, ya que esta esta optimizada para el despliegue de proyectos desarrollados en Next.js.

### **Despliegue Back-end Spring Boot**

Los pasos seguidos para el despliegue en el servicio Elastic Beanstalk de AWS es el siguiente:

#### **1 Registro en Amazon Web Services (AWS)**

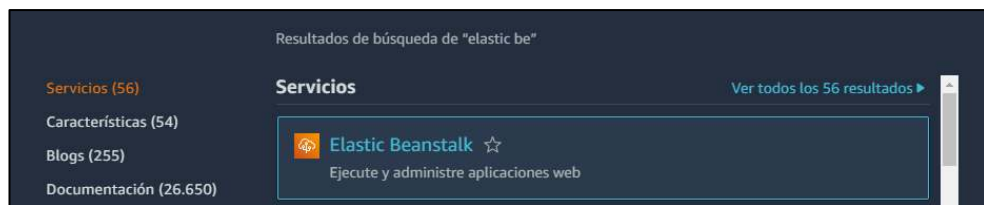
AWS al registrar un usuario requiere de una forma de pago ya que la plataforma cobrará un precio por el número de servicios usados, al registrarse por primera vez hay la posibilidad de tener 3 meses de uso gratuito en servicios seleccionados.



**Figura 3.159 Registro en Amazon Web Services (AWS)**  
Elaborado por: El investigador

## 2 Buscar el servicio Elastic Beanstalk

El servicio Elastic Beanstalk permite ejecutar y administrar aplicaciones web con simples configuraciones.



**Figura 3.160 Lista de mensajes enviados por el dispositivo back-end Sigfox**  
Elaborado por: El investigador

## 3 Crear entorno

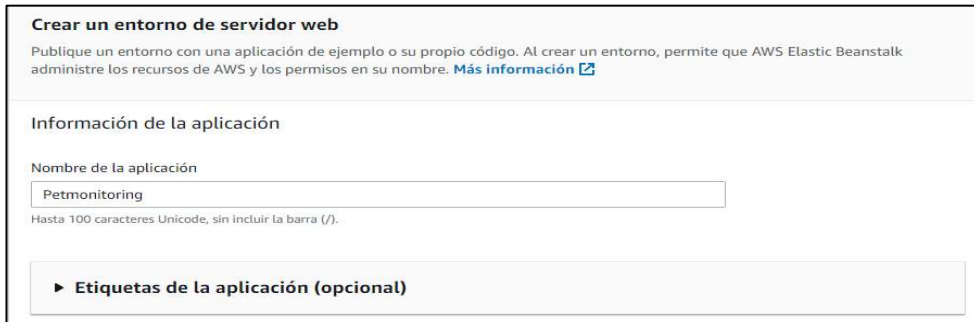
Un entorno es un conjunto de recursos de AWS que ejecutan versiones de una aplicación.



**Figura 3.161 Elección de entorno para una aplicación web**  
Elaborado por: El investigador

## 4 Creación de un entorno de servidor web

La información de la aplicación requiere un nombre para la aplicación en este caso Petmonitoring.



**Crear un entorno de servidor web**  
Publique un entorno con una aplicación de ejemplo o su propio código. Al crear un entorno, permite que AWS Elastic Beanstalk administre los recursos de AWS y los permisos en su nombre. [Más información](#)

**Información de la aplicación**

Nombre de la aplicación  
Petmonitoring  
Hasta 100 caracteres Unicode, sin incluir la barra (/).

► **Etiquetas de la aplicación (opcional)**

**Figura 3.162 Ingreso de nombre de la aplicación para la creación de un entorno de servidor web**  
Elaborado por: El investigador

El entorno para ser creado requiere de un nombre, la plataforma en la que funcionara y el modo que se cargara la aplicación web.



**Información del entorno**  
Elija el nombre, el subcampo y la descripción para su entorno. Estas no se pueden cambiar más adelante.

Nombre del entorno  
Petmonitoring-env

Campo  
Dejar en blanco para generar valor aut. us-east-1:elasticbeanstalk  
Comprobar disponibilidad

Descripción

**Figura 3.163 Ingreso de nombre del entorno**  
Elaborado por: El investigador

El entorno requiere una plataforma en la que servirán los servicios web tipo REST, estos están desarrollados en el lenguaje Java por lo que escogemos el tipo de plataforma Java.



**Plataforma**

**Plataforma administrada**  
Plataformas publicadas y mantenidas por Amazon Elastic Beanstalk. [Obtenga más información](#)

**Plataforma personalizada**  
Plataformas creadas por usted y que le pertenecen.

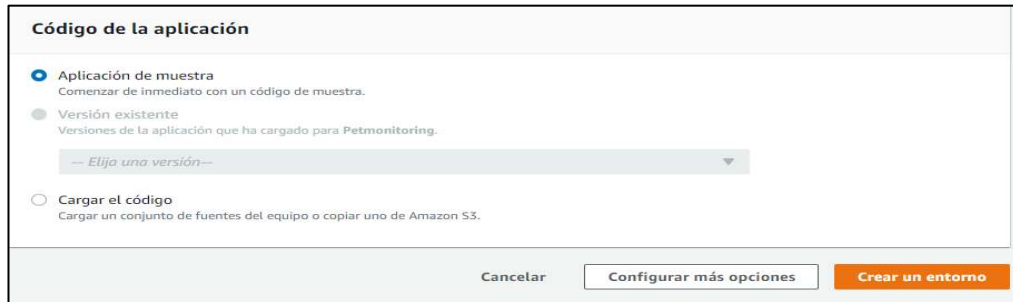
Plataforma  
Java

Ramificación de la plataforma  
Corretto 17 running on 64bit Amazon Linux 2

Versión de la plataforma  
3.3.1 (Recommended)

**Figura 3.164 Elección de la plataforma en la que esta desarrollada la aplicación web**  
Elaborado por: El investigador

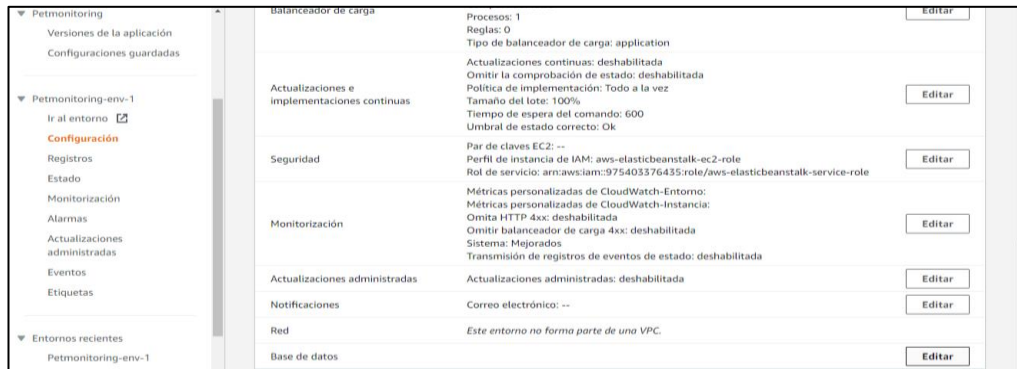
Para la carga de la aplicación elegimos la aplicación muestra para que al crearse el entorno sea posible ver su funcionamiento.



**Figura 3.165 Elección de modo de carga de la aplicación web y creación de entorno**  
Elaborado por: El investigador

## 5 Configuración de la base de datos en el entorno creado

Dado que para el funcionamiento de los servicios web tipo REST es necesario una base de datos esta se configurará en el entorno creado.



Balancedador de carga	Procesos: 1 Reglas: 0 Tipo de balancedador de carga: application	Editar
Actualizaciones e implementaciones continuas	Actualizaciones continuas: deshabilitada Omitir la comprobación de estado: deshabilitada Política de implementación: Todo a la vez Tamaño del lote: 100% Tiempo de espera del comando: 600 Umbral de estado correcto: Ok	Editar
Seguridad	Par de claves EC2: -- Perfil de instancia de IAM: aws-elasticbeanstalk-ec2-role Rol de servicio: arn:aws:iam:975403376435:role/aws-elasticbeanstalk-service-role	Editar
Monitorización	Métricas personalizadas de CloudWatch-Entorno: Métricas personalizadas de CloudWatch-Instancia: Omitir HTTP 4xx: deshabilitada Omitir balancedador de carga 4xx: deshabilitada Sistema: Mejorados Transmisión de registros de eventos de estado: deshabilitada	Editar
Actualizaciones administradas	Actualizaciones administradas: deshabilitada	Editar
Notificaciones	Correo electrónico: --	Editar
Red	Este entorno no forma parte de una VPC.	Editar
Base de datos		Editar

**Figura 3.166 Opciones de configuración del entorno de la aplicación web**  
Elaborado por: El investigador

## 6 Ingreso de características de la base de datos

Dado que los servicios web tipo REST fueron desarrollados con la base de datos PostgreSQL, ingresamos las características que se necesitaría para que la base de datos sea suficiente para su perfecto funcionamiento.

**Configuración de la base de datos**  
 Seleccione un motor y tipo de instancia para la base de datos de su entorno.

**Motor**  
 postgres

**Versión del motor**  
 14.3

**Clase de instancia**  
 db.t4g.micro

**Almacenamiento**  
 Elija un número entre 5 GB y 1024 GB.  
 10

**Nombre de usuario**  
 postgres

**Contraseña**  
 \*\*\*\*\*

**Disponibilidad**  
 Baja (una AZ)

**Política de eliminación de bases de datos**  
 Esta política se aplica al desactivar una base de datos o al terminar el entorno asociado a ella.

**Crear instantánea**  
 Elastic Beanstalk guarda una instantánea de la base de datos y, a continuación, la elimina. Puede restaurar una base de datos a partir de una instantánea al agregar una base de datos a un entorno de Elastic Beanstalk o al crear una base de datos independiente. Es posible que se apliquen cargos por almacenar instantáneas de base de datos.

**Retener**  
 La base de datos desactivada seguirá estando disponible y operativa fuera de Elastic Beanstalk.

**Eliminar**  
 Elastic Beanstalk termina la base de datos. La base de datos dejará de estar disponible.

Cancelar Continuar Aplicar

**Figura 3.167 Configuración de la información de la base de datos PostgreSQL en el entorno de la aplicación web**  
 Elaborado por: El investigador

## 7 Carga de la aplicación desarrollada en Spring Boot

Una vez creado el entorno es posible cargar la aplicación con extensión .jar generada por Spring Boot con los servicios web tipo REST.

**Cargar e implementar**

Para implementar una versión anterior, vaya a la página de versiones de la aplicación.

**Cargar aplicación**  
 Elegir archivo  
 Nombre del archivo: MonitoreoMascotas-0.0.1-SN...  
 Etiqueta de versión: pet-monitoring-api

**Preferencias de implementación**  
 La versión de la aplicación se implementará con la política **Todo a la vez**.  
 Número actual de instancias: 1

Cargando la versión de la aplicación... Cancelar Implementar

**Figura 3.168 Carga de la aplicación web desarrollada en Spring Boot**  
 Elaborado por: El investigador

Después de cargar la aplicación es posible ver el estado de la aplicación.

Elastic Beanstalk > Entornos > Petmonitoring-env

**Petmonitoring-env**  
 Petmonitoring-env:eba-4274hj28.us-east-1.elasticbeanstalk.com (e-bpbdayuc3n)  
 Nombre de la aplicación: pet-monitoring

Actualizar Acciones

**Estado**  
 Aceptar Causas

**Versión en ejecución**  
 pet-monitoring-api  
 Cargar e implementar

**Plataforma**  
 Corretto 8 running on 64bit Amazon Linux 2/3.3.1  
 Cambiar

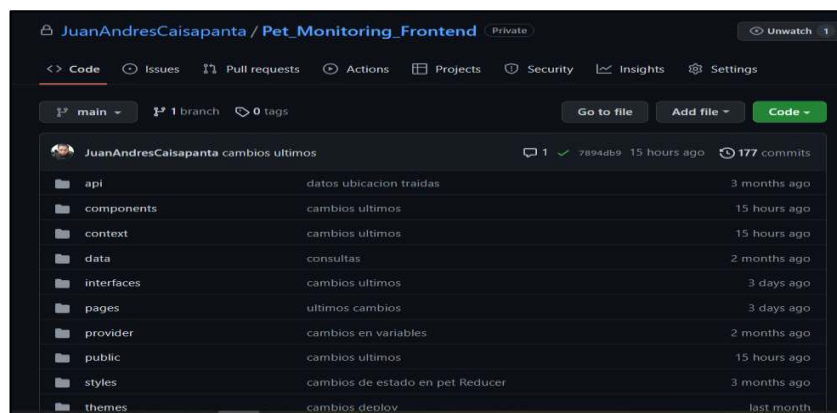
**Figura 3.169 Estado de la aplicación web con los servicios web tipo REST en AWS**  
Elaborado por: El investigador

## Despliegue PWA MonIOpeT Next.js

Los pasos seguidos para el despliegue de la PWA MonIOpeT en la plataforma Vercel es el siguiente:

### 1 Repositorio en Github de la PWA MonIOpeT

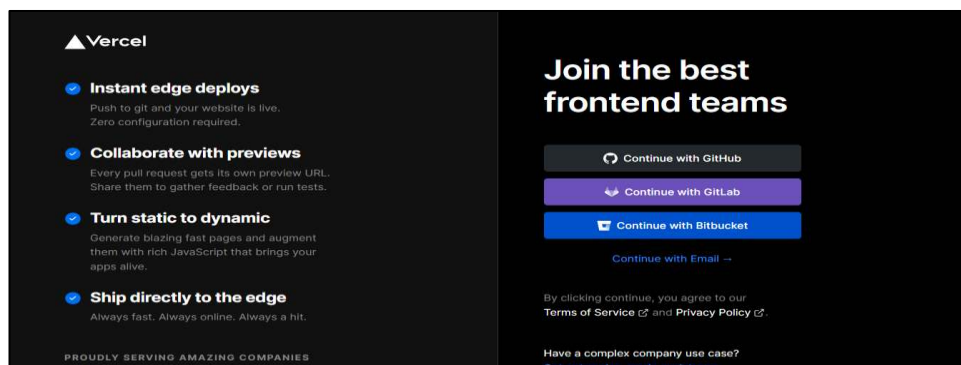
Vercel cuenta con la posibilidad de conectar repositorios de Github y desplegar los proyectos desarrollados en JavaScript.



**Figura 3.170 Repositorio de la PWA desarrollada en Next.js**  
Elaborado por: El investigador

### 2 Registro en plataforma Vercel

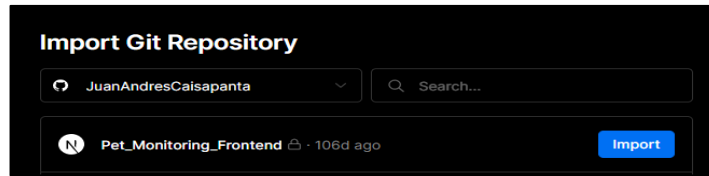
Vercel da la posibilidad de registrarse en la plataforma por medio de la cuenta de Github.



**Figura 3.171 Registro de usuario de la plataforma Vercel**  
Elaborado por: El investigador

### 3 Importación del repositorio de Github

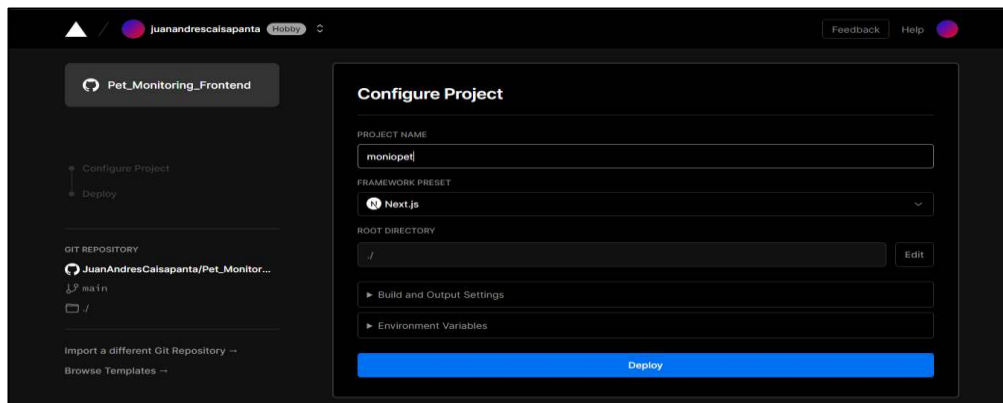
Con la cuenta de Github enlazada con la plataforma Vercel es posible importar el proyecto de la PWA.



**Figura 3.172 Importación proyecto PWA de Github**  
Elaborado por: El investigador

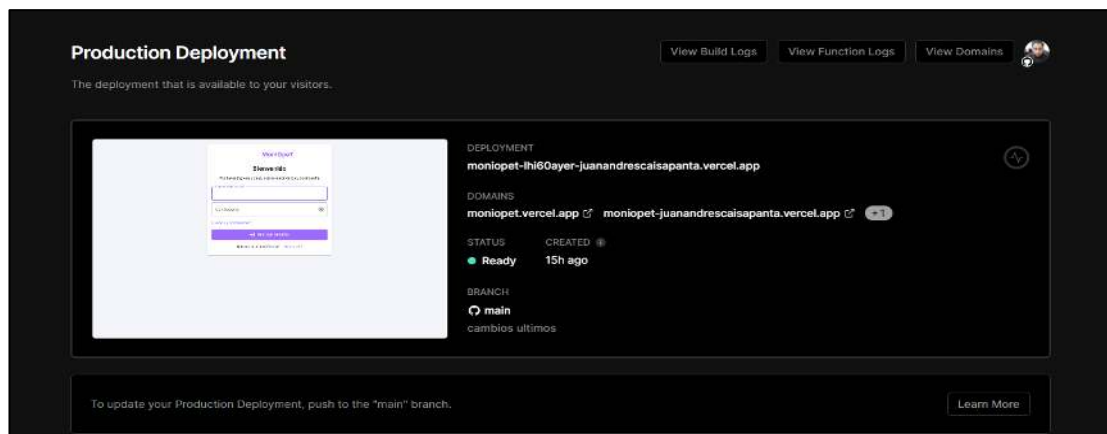
#### 4 Configuración y despliegue del proyecto

Configuramos el proyecto que será desplegado en este caso lo desplegamos con el nombre MonIOpeT



**Figura 3.173 Configuración proyecto PWA con nombre MonIOpeT**  
Elaborado por: El investigador

Una vez configurado el proyecto Vercel despliega la PWA con un enlace certificado https.



**Figura 3.174 Lista de mensajes enviados por el dispositivo back-end Sigfox**  
Elaborado por: El investigador

#### Instalación de la PWA MonIOpeT

Ya que MonIOpeT es una PWA esta puede ser instalada en varios ecosistemas, los cuales son los siguientes:



## Dispositivo computadora con sistema operativo Windows 10

Los pasos para instalar la PWA MonIOpeT en el sistema operativo Windows 10 como una aplicación de escritorio son los siguientes:

### 1 Abrir la página web moniopet.vercel.app en el navegador Google Chrome e instalar la PWA

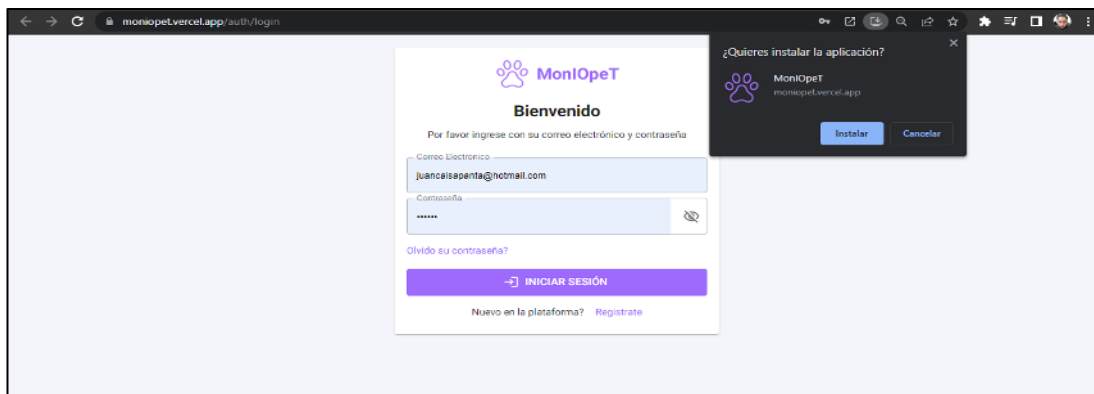


Figura 3.175 Opción de instalación de la PWA MonIOpeT en el navegador Google Chrome  
Elaborado por: El investigador

### 2 Ejecutar la aplicación con el icono generado en el escritorio del sistema operativo Windows 10

La aplicación se instala en el sistema operativo Windows 10 junto a un acceso directo en el escritorio del sistema.

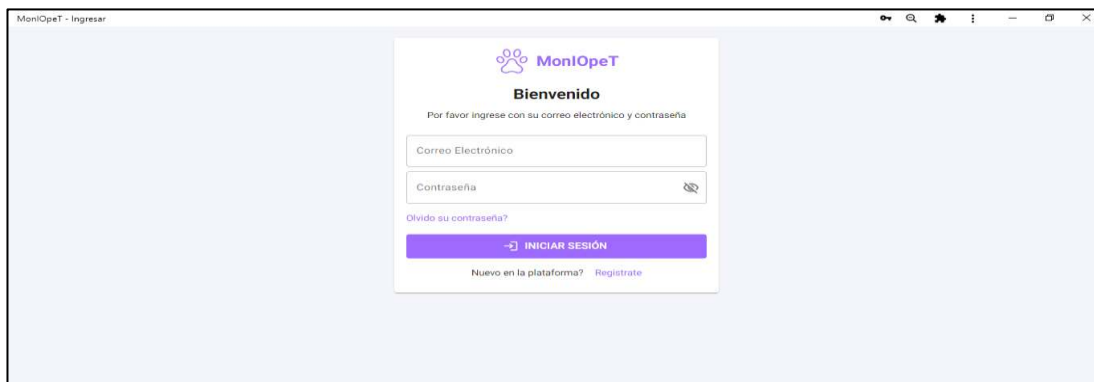


Figura 3.176 PWA MonIOpeT ejecutada como una aplicación de escritorio en Windows 10  
Elaborado por: El investigador

## Dispositivo móvil android OPPO Reno4 Z 5G

Los pasos para instalar la PWA MonIOpeT en un dispositivo móvil Android como una aplicación nativa.

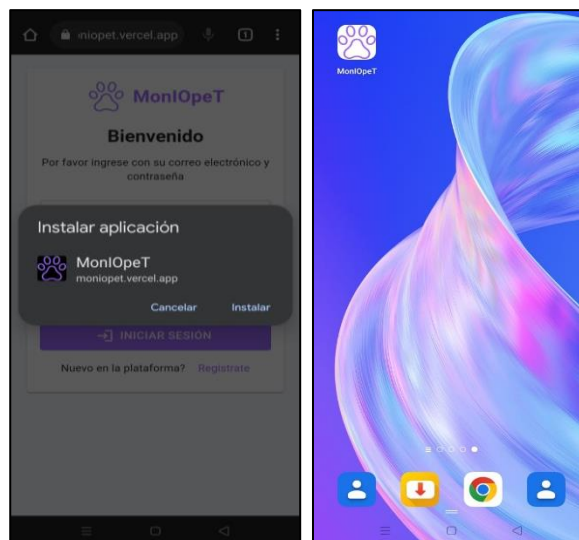
### 1 Abrir la página web moniopet.vercel.app en el navegador Google Chrome



**Figura 3.177 Opción de instalación de la PWA MonIOpeT en el navegador Google Chrome de sistema operativo móvil Android  
Elaborado por: El investigador**

## **2 Instalar la PWA MonIOpeT**

Al instalar la PWA en el dispositivo móvil Android se puede visualizar el icono de la PWA como una aplicación nativa.



**Figura 3.178 Instalación e icono de la PWA instalada MonIOpeT  
Elaborado por: El investigador**

Al pulsar el icono de la aplicación instalada se ejecuta la PWA MonIOpeT

## **3 Ejecutar la aplicación instalada**



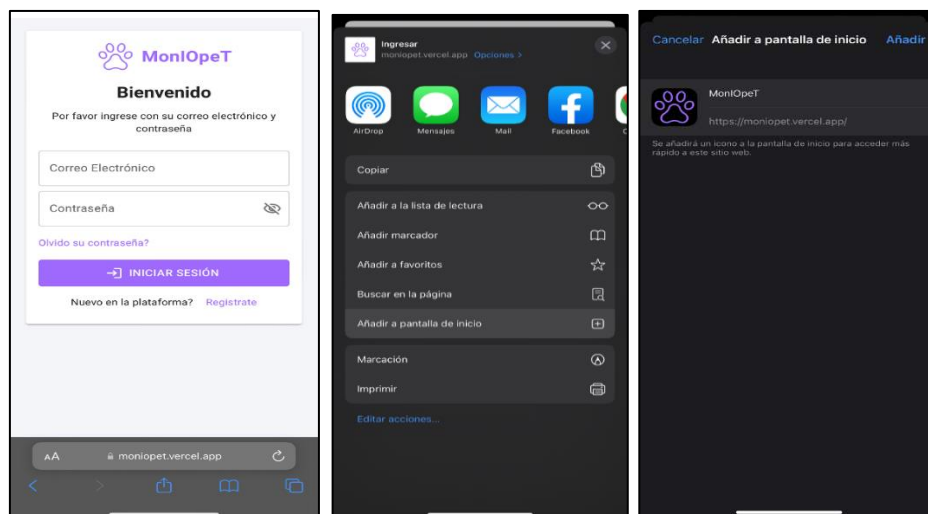
**Figura 3.179 PWA MonIOpeT ejecutada como una aplicación de nativa en dispositivo móvil Android**  
Elaborado por: El investigador

### Dispositivo móvil iPhone 13

Los dispositivos Apple no cuenta con la función de instalación de PWAs, pero es posible incorporarlas como un icono de una aplicación para ejecutarla cuando sea necesario.

Los pasos para realizar el icono de la PWA en el dispositivo iPhone son los siguientes:

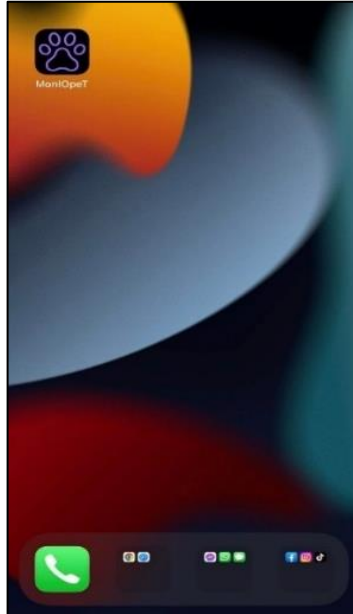
- 1 **Abrir la página web [moniopeT.vercel.app](https://moniopeT.vercel.app) en el navegador Safari y en las opciones del navegador seleccionar Añadir a la pantalla de inicio.**



**Figura 3.180 Pasos para añadir a la pantalla de inicio la PWA MonIOpeT**  
Elaborado por: El investigador

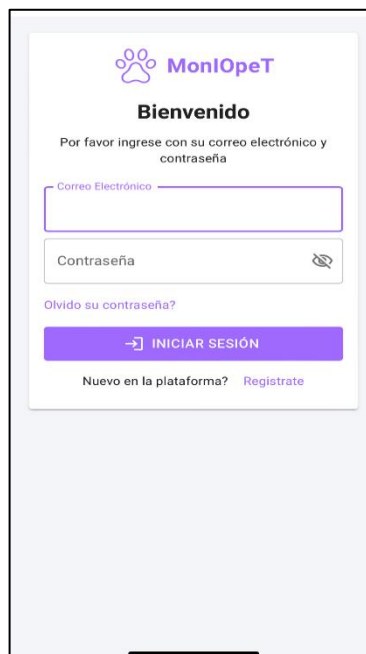
## 2 Pulsar el icono de la PWA MonIOpeT

El icono de la PWA se visualiza en la pantalla principal del dispositivo móvil iPhone.



**Figura 3.181** Icono de la PWA MonIOpeT en iPhone 13  
Elaborado por: El investigador

## 3 Ejecutar la PWA MonIOpeT



**Figura 3.182** PWA MonIOpeT ejecutada como una aplicación en dispositivo iPhone 13  
Elaborado por: El investigador

## CAPÍTULO IV CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

- Gracias al análisis de la tecnología LPWAN más accesible para el rastreo de mascotas en la ciudad de Quito, se escogió la tecnología Sigfox, ya que esta al ser de poco costo y con una cobertura que cubre la mayoría de la ciudad, es idónea para el desarrollo de un prototipo de dispositivo de monitoreo.
- Se logró desarrollar un prototipo de dispositivo de monitoreo de mascotas, gracias a que la placa de desarrollo Ufox usada en su construcción, está diseñada para trabajar con el servicio de Sigfox y cuenta con una documentación clara por parte de sus fabricantes.
- Al completar el desarrollo de la PWA MonIOpeT se verificaron las características principales de una PWA estudiadas en el presente proyecto de investigación, ya que este tiene un diseño responsivo que se adapta a diferentes dispositivos de distinto tamaño, también se evidencia su seguridad gracias al protocolo HTTPS que usa Vercel al desplegar PWAs con una URL que puede ser compartida, además al instalarse en los dispositivos puede ser actualizada constantemente y seguir funcionando al estar sin conexión a internet.
- La PWA con el nombre MonIOpeT, la cual ha sido desarrollada usando la herramienta Spring Boot para la realización de sus servicios web tipo REST y el framework Next.js para la realización de su interfaz gráfica en el presente proyecto de investigación, cumple con la actividad de obtener los datos de los dueños de mascotas y da la posibilidad de monitorear en tiempo real la ubicación, la temperatura de su mascota. Además, el dueño de la mascota puede gestionar los establecimientos que frecuenta, las medicinas que usa y los profesionales que visita su mascota en la ciudad de Quito.

### 4.2 Recomendaciones

- La red y servicio de Sigfox se expande gracias al número de proyectos que se están realizando en un territorio, por ello se recomienda investigar y desarrollar proyectos que ayuden a la expansión de esta tecnología en el Ecuador.

- Al realizar el prototipo de dispositivo para el monitoreo de mascotas se identificó que el módulo GPS NEO-6M de Arduino requiere de un tiempo estimado de 10 minutos en un lugar abierto para entregar la ubicación en la que se encuentra, por lo que se debe esperar al encender el dispositivo un tiempo hasta que los primeros datos en tiempo real sean reflejados en la PWA. Además, de ser posible en futuros proyectos utilizar un módulo GPS de Arduino de mejor calidad.
- En el desarrollo de la PWA con Next.js no se usó su funcionalidad como servidor, donde se da la posibilidad de realizar servicios web tipo REST por lo que se aconseja en futuros desarrollos de PWAs incorporar esta funcionalidad.
- Ya que la PWA desarrollada en el presente proyecto de investigación se divide en dos perfiles, los cuales son administrador y usuario, se sugiere usar la interfaz del perfil administrador en computadoras y la interfaz del perfil usuario en dispositivos móviles.
- El prototipo de monitoreo de mascotas usa la tecnología Sigfox como único medio de comunicación con la PWA MonIOpeT, por lo que se recomienda en futuras implementaciones incorporar otra tecnología que sea un soporte en caso de que Sigfox no funcione.

## REFERENCIA BIBLIOGRÁFICA

- [1] A. Abdulkarim, A. Khan, B. Goriman Khan, and E. Aklilu, “Stray Animal Population Control: Methods, Public Health Concern, Ethics, and Animal Welfare Issues,” *World Vet J*, vol. 11, no. 3, pp. 319–326, 2021, doi: 10.54203/scil.2021.wvj44.
- [2] B. L. Lee Yin B, “The development of pet GPS tracker using LORA.” [Online]. Available: <https://plh.utem.edu.my/cgi-bin/koha/opac-detail.pl?biblionumber=118593>
- [3] N. A. Pérez Vargas and B. D. Torres Pachar, “Desarrollo de un Prototipo para la Geolocalización y Monitorización de Frecuencia Cardiaca de Mascotas en la Ciudad de Quito,” Quito. [Online]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/19072/1/UPS%20-%20TTS072.pdf>
- [4] Ing. Anthony Limber Morán Cabezas, “Análisis e Implementación de un Sistema de Rastreo Satelital aplicado a mascotas mediante software libre con tecnología GPS y GSM,” Guayaquil. [Online]. Available: <http://repositorio.ucsg.edu.ec/bitstream/3317/16332/1/T-UCSG-POS-MTEL-192.pdf>
- [5] “Guayaquil, más de 1000 animales de compañía extraviados en el 2021 – FRAE.” <https://rescateanimal.org.ec/estudio-amp/>
- [6] “Monitoreo de áreas, especies y poblaciones para evaluar la efectividad de las acciones de manejo y conservación,” [Online]. Available: [http://www.cropwildrelatives.org/fileadmin/templates/cropwildrelatives.org/upload/In\\_situ\\_Manual/3-CWR\\_MANUAL\\_SPANISH-cap13.pdf](http://www.cropwildrelatives.org/fileadmin/templates/cropwildrelatives.org/upload/In_situ_Manual/3-CWR_MANUAL_SPANISH-cap13.pdf)
- [7] intrac, “The different functions of programme monitoring,” 2017. [Online]. Available: <https://www.intrac.org/wpcms/wp-content/uploads/2017/01/Monitoring.pdf>
- [8] C. Kuendig, M. Sabathy, and T. Biro, “Systems and sensors for monitoring purposes ,” Othmarsingen, Switzerland. [Online]. Available: [https://www.geosig.com/files/GeoSIG\\_Systems\\_and\\_Sensors\\_for\\_Monitoring\\_Purposes\\_2009.pdf](https://www.geosig.com/files/GeoSIG_Systems_and_Sensors_for_Monitoring_Purposes_2009.pdf)
- [9] N. Anis, A. Binti, and A. Shah, “Tracker-Pet Collar Application (T-PC APPS) to Track Pet Using Global-Positioning System (GPS),” 2020. [Online]. Available: <https://myfik.unisza.edu.my/www/fyp/fyp19sem2/report/047368.pdf>
- [10] K. K. Patel, S. M. Patel, and P. G. Scholar, “Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges,” *International Journal of Engineering Science and Computing*, 2016, [Online]. Available: <http://ijesc.org/>
- [11] S. Hernández Caballero, “Estudio en detalle de LoRaWAN. Comparación con otras tecnologías LPWAN considerando diferentes patrones de tráfico.” [Online]. Available:

<https://openaccess.uoc.edu/bitstream/10609/106369/7/shernandezc6TFM0120 memoria.pdf>

- [12] W. Alexis, P. Calderón, O. Yazit, and S. García, “Implementación de una Red LPWAN de Sensores para el Monitoreo de Condiciones Ambientales”, [Online]. Available: [https://www.cdtdegas.com/images/Descargas/Nuestra\\_revista/MetFlu14/9\\_RedLPWAN.pdf](https://www.cdtdegas.com/images/Descargas/Nuestra_revista/MetFlu14/9_RedLPWAN.pdf)
- [13] O. Bautista, J. Elizabeth, R. Reinoso, M. Augusta, I. L. Cueva, and R. Alcides, “Implementación de un prototipo de geoposicionamiento con tecnología Sigfox, aplicado a paradas inteligentes y rastreo de niños”, [Online]. Available: <http://repositorio.espe.edu.ec/bitstream/21000/25209/1/T-ESPE-044591.pdf>
- [14] A. David Rea Espinoza, “Desarrollo de un Rastreador Gps para Ubicación y Desplazamiento de Niños en Rutas o Zonas Configuradas en una Aplicación,” Quito. [Online]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/20714/1/UPS%20-%20TTS443.pdf>
- [15] “¿Qué es sigfox? – SIGFOX.” <https://sigfox.com.py/que-es-sigfox/>
- [16] Emmanuel M. Migabo, Karim D. Djouani, and Anish M. Kurien, “The Narrowband Internet of Things (NB-IoT) Resources Management Performance State of Art, Challenges, and Opportunities”, [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9097268>
- [17] A. M. Pinto Erazo, “Desarrollo de una Metodología de Análisis de Tecnologías de Radio IoT, para Escenarios Urbano y Rural, por Medio de Modelos Matemáticos.” [Online]. Available: <http://repositorio.utn.edu.ec/bitstream/123456789/10809/2/PG%20798%20TRABAJO%20GRADO.pdf>
- [18] SEMTECH, “LoRa® and LoRaWAN®: A Technical Overview,” 2020. [Online]. Available: [https://loradevelopers.semtech.com/uploads/documents/files/LoRa\\_and\\_LoRaWAN-A\\_Tech\\_Overview-Downloadable.pdf](https://loradevelopers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf)
- [19] M. Banzi and M. Shiloh, The Open Source Electronics Prototyping Platform Make: Getting Started with Arduino. [Online]. Available: [https://www.esc19.net/cms/lib011/TX01933775/Centricity/Domain/110/make\\_gettingstartedwitharduino\\_3rdedition.pdf](https://www.esc19.net/cms/lib011/TX01933775/Centricity/Domain/110/make_gettingstartedwitharduino_3rdedition.pdf)
- [20] R. Teja, “Lista de módulos compatibles con Arduino | Lista de módulos de Arduino.” <https://www.electronicshub.org/arduino-modules-list/>
- [21] M. R. Valarezo Pardo, J. A. Honores Tapia, A. S. Gómez Moreno, and L. F. Vences Sánchez, “Comparación de Tendencias Tecnológicas en Aplicaciones Web.” [Online]. Available: [https://www.3ciencias.com/wp-content/uploads/2018/09/Art\\_2.pdf](https://www.3ciencias.com/wp-content/uploads/2018/09/Art_2.pdf)
- [22] J. R. Molina Ríos, M. P. Zea Ordóñez, M. J. Contenido Segarra, and F. G. García Zerda, “Comparison of methodologies in web applications.” doi: 10.17993/3ctecno.2018.v7n1e25.1-19.



- [23] W. R. Oñate Calderón, “Implementación de una Aplicación Web Progresiva para la Gestión de Pruebas de Simulación para el Ingreso a Instituciones Militares y Universidades en el centro de Capacitación y Nivelación Académica Smartel.” [Online]. Available: <https://repositorio.uta.edu.ec/jspui/bitstream/123456789/31787/1/t1741si.pdf>
- [24] A. Gambhir and G. Raj, “Analysis of Cache in Service Worker and Performance Scoring of Progressive Web Application,” Proceedings on 2018 International Conference on Advances in Computing and Communication Engineering, ICACCE 2018, pp. 294–299, Aug. 2018, doi: 10.1109/ICACCE.2018.8441715.
- [25] G. Martínez Villalobos, G. D. Camacho Sánchez, and D. A. Biancha Gutiérrez, “Diseño de Framework Web para el Desarrollo Dinámico de Aplicaciones,” Scientia et Technica Año XVI, vol. 44, 2010, [Online]. Available: <https://www.redalyc.org/pdf/849/84917316032.pdf>
- [26] J. J. Valdivia Caballero, “Modelo de procesos para el desarrollo del front-end de aplicaciones web,” 2016, [Online]. Available: <https://revistas.ulima.edu.pe/index.php/Interfases/article/view/1245/1205>
- [27] B. M. Hamerník, “Development of Modern User Interfaces in Angular Framework.” [Online]. Available: <https://is.muni.cz/th/ilj4c/dp.pdf>
- [28] J. Johansson, “Create React App vs NextJS A comparison of two ReactJS based web application frameworks.” [Online]. Available: <https://www.diva-portal.org/smash/get/diva2:1573881/FULLTEXT01.pdf>
- [29] S. Plaza Estévez, N. Ramírez Lamela, and C. Acosta Morales, “API de servicios web orientados a accesibilidad,” Madrid. [Online]. Available: [https://eprints.ucm.es/id/eprint/38686/1/Memoria\\_API%20de%20servicios%20web%20de%20accesibilidad.pdf](https://eprints.ucm.es/id/eprint/38686/1/Memoria_API%20de%20servicios%20web%20de%20accesibilidad.pdf)
- [30] “JSON Web Token Introduction - jwt.io.” <https://jwt.io/introduction>
- [31] S. Ramírez Pérez, “Estudio del framework Spring, Spring Boot y Microservicios,” 2020. [Online]. Available: [https://ebuah.uah.es/dspace/bitstream/handle/10017/45107/TFM\\_Ramirez\\_Perez\\_2020.pdf?sequence=1&isAllowed=y](https://ebuah.uah.es/dspace/bitstream/handle/10017/45107/TFM_Ramirez_Perez_2020.pdf?sequence=1&isAllowed=y)
- [32] M. Sköld, “Active Database Management Systems for Monitoring and Control.” [Online]. Available: <http://www.it.uu.se/research/group/udbl/Theses/MartinSkoldPhD.pdf>
- [33] Z. Xi, “The Usability of PostgreSQL.” [Online]. Available: [https://www.theseus.fi/bitstream/handle/10024/30123/Zeng\\_Xi.pdf?sequence=1&isAllowed=y](https://www.theseus.fi/bitstream/handle/10024/30123/Zeng_Xi.pdf?sequence=1&isAllowed=y)
- [34] D. Chappell, “A short introduction to cloud platforms an enterprise-oriented view sponsored by microsoft corporation.” [Online]. Available: <http://www.davidchappell.com/CloudPlatforms--Chappell.pdf>

- [35] J. Varia and S. Mathew, “Overview of Amazon Web Services ,” 2014. [Online]. Available: [https://media.amazonwebservices.com/AWS\\_Overview.pdf](https://media.amazonwebservices.com/AWS_Overview.pdf)
- [36] “Introduction to Vercel – Vercel Docs.” <https://vercel.com/docs>
- [37] E. Tegegne, “Software Development Methodologies and Practices in Startups-Systematic Literature Review,” 2018, [Online]. Available: <http://jultika.oulu.fi/files/nbnfioulu-201804041421.pdf>
- [38] K. Saeedi and A. Visvizi, “Software Development Methodologies, HEIs, and the Digital Economy,” 2021. doi: 10.3390/educsci.
- [39] “Ciudades Inteligentes.” [Online]. Available: [https://habitat3.org/wp-content/uploads/Issue-Paper-21\\_ciudades-inteligentes.pdf](https://habitat3.org/wp-content/uploads/Issue-Paper-21_ciudades-inteligentes.pdf)
- [40] J. E. Orbea Bautista and M. A. Rosero Reinoso, “Implementación de un prototipo de geoposicionamiento con tecnología Sigfox, aplicado a paradas inteligentes y rastreo de niños.” [Online]. Available: <http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/25209/T-ESPE-044591.pdf?sequence=1&isAllowed=y>
- [41] G. G. Vallejo Cabezas and C. A. Cuesta Cruz, “Diseño de una Red Lpwan para la Implementación de Medidores Inteligentes en el Cantón San Cristóbal, Provincia de Galápagos.” [Online]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/21915/1/UPS%20-%20TTS621.pdf>
- [42] “Configuraciones de radio | compilación sigfox.” <https://build.sigfox.com/sigfox-radio-configurations-rc#geographical-zones>
- [43] “Teca IoT – El futuro es simple.” <https://teca.pe/>
- [44] “TECA-IOT/Ufox: Librería Ufox - Sigfox.” <https://github.com/TECA-IOT/Ufox>
- [45] “Coverage | Sigfox.” <https://www.sigfox.com/en/coverage>
- [46] A. Nilsson, “Performance and feature support of Progressive Web Applications: A performance and available feature comparison between Progressive Web Applications, React Native applications and native iOS applications.” 2022, [Online]. Available: <https://www.diva-portal.org/smash/get/diva2:1653815/FULLTEXT01.pdf>
- [47] T. S. Flavio Wilson and S. Y. Sairy Amauta, “Aplicación Web Progresiva para el Manejo de Inventario en la Farmacia de la Coordinación de Salud Zona 3.” [Online]. Available: [http://dspace.unach.edu.ec/bitstream/51000/9159/1/PWA\\_tesis\\_Segovia\\_Tapuy-Final%20%281%29.pdf](http://dspace.unach.edu.ec/bitstream/51000/9159/1/PWA_tesis_Segovia_Tapuy-Final%20%281%29.pdf)
- [48] O. Adetunji, C. Ajaegbu, N. Otuneme, and O. J. Omotosho, “Dawning of Progressive Web Applications (PWA): Edging Out the Pitfalls of Traditional Mobile Development,” American Academic Scientific Research Journal for Engineering, Technology, and Sciences, vol. 68, no. 1, pp. 85–99, May 2020, [Online]. Available:

[https://asrjetsjournal.org/index.php/American\\_Scientific\\_Journal/article/view/5812](https://asrjetsjournal.org/index.php/American_Scientific_Journal/article/view/5812)

- [49] “Exploring the role of progressive web applications in modern web development”, doi: 10.1088/1742-6596/1679/2/022043.
- [50] “What is Next.js? | Learn Next.js.” <https://nextjs.org/learn/foundations/about-nextjs/what-is-nextjs>
- [51] “Back-end Sigfox.” <https://backend.sigfox.com/auth/login>
- [52] E. J. Hall, A. Fleming, and A. J. Carter, “Investigating the use of non-contact infrared thermometers in cats and dogs,” <https://doi.org/10.12968/vetn.2019.10.2.109>, vol. 10, no. 2, pp. 109–115, Mar. 2019, doi: 10.12968/VETN.2019.10.2.109.



## ANEXOS

### Anexo 1

#### Cuestionario realizado a dueños de mascotas en la ciudad de Quito

### Encuesta sobre el monitoreo de mascotas en tiempo real

La presente encuesta es realizada a propietarios de mascotas con el fin de determinar la necesidad de una herramienta informática para la monitorización en tiempo real de mascotas en la ciudad de Quito.

 [juanitocaisapanta@gmail.com](mailto:juanitocaisapanta@gmail.com) (no compartidos) 

[Cambiar de cuenta](#)

**\*Obligatorio**

1.- ¿Cuáles considera usted que son las razones por las que existen animales callejeros? (Puede elegir varias opciones) \*

Robo

Abandono

Pérdida

2.- ¿Considera usted que la seguridad de una mascota es importante? \*

Sí

No

3.- ¿De qué especie de mascota usted ha sido dueño? (Puede elegir varias opciones) \*

Perro

Gato

Otro

4.- ¿Ha perdido alguna vez una mascota? \*

Sí

No

5.- ¿Le gustaría saber la ubicación y temperatura de su mascota en tiempo real? \*

Sí

No

6.- ¿Lleva usted el registro de los datos de su mascota en alguna plataforma tecnológica? \*

Sí

No

7.- ¿Tendría algún problema usted con que su mascota use un pequeño dispositivo en su cuello para su monitoreo en tiempo real? \*

Sí

No

8.- ¿Estaría usted interesado en una plataforma de monitoreo para su mascota en tiempo real de pago anual? \*

Sí

No

9.- ¿Cuál es el equipo tecnológico con el que usted accede a internet frecuentemente? (Puede elegir varias opciones) \*

Dispositivo Móvil

Laptop

Computadora de escritorio

**Enviar** [Borrar formulario](#)

Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](#) - [Términos del Servicio](#) - [Política de Privacidad](#)

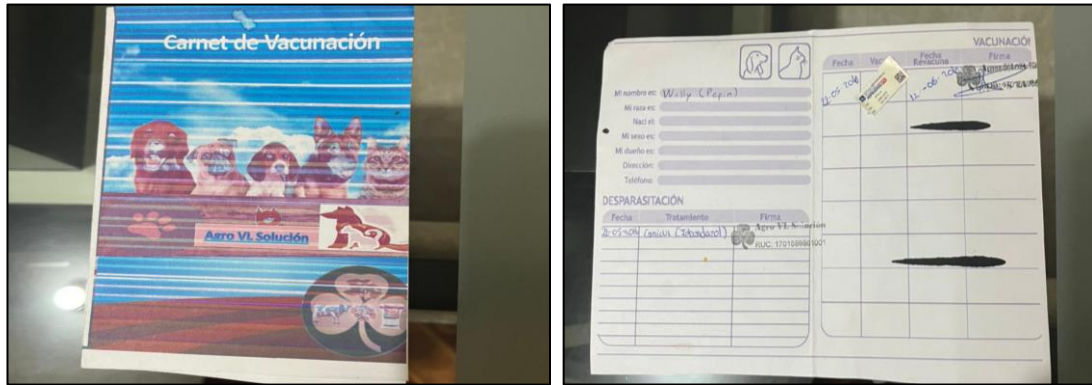
**Google** Formularios

La encuesta se aplicó a 409 dueños de mascotas en la ciudad de Quito

## Anexo 2

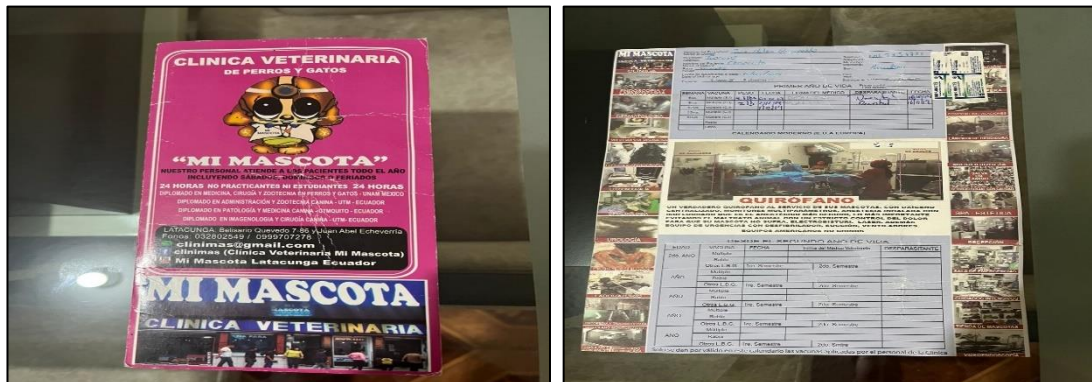
### Documentos usados para abstraer información

#### Documento 1



El documento corresponde a la clínica veterinaria Agro VL Solución.

#### Documento 2



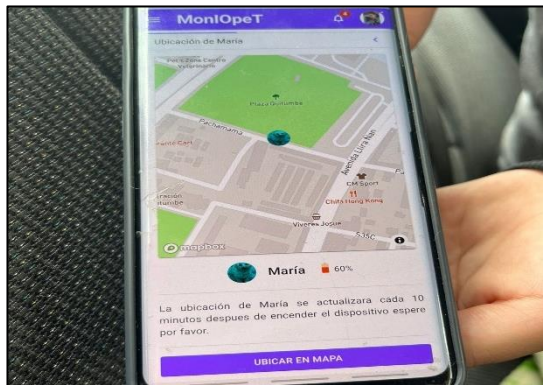
El documento corresponde a la clínica veterinaria Mi Mascota.



### Anexo 3

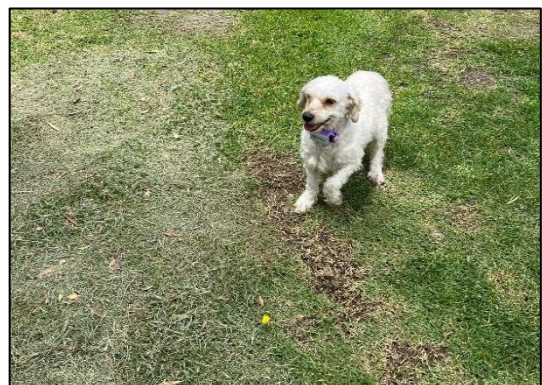
#### Uso de la aplicación en la ciudad de Quito

##### Sector Sur de Quito

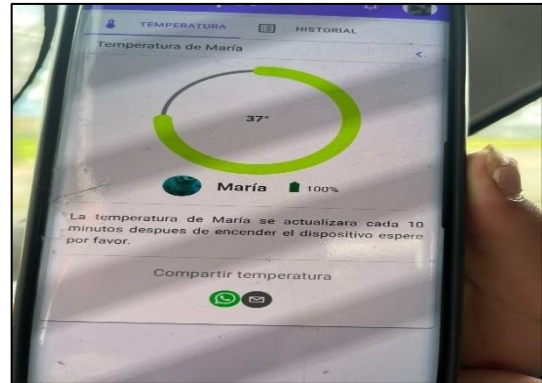


Las fotografías fueron realizadas en el sector Quitumbe.

##### Sector Centro de Quito

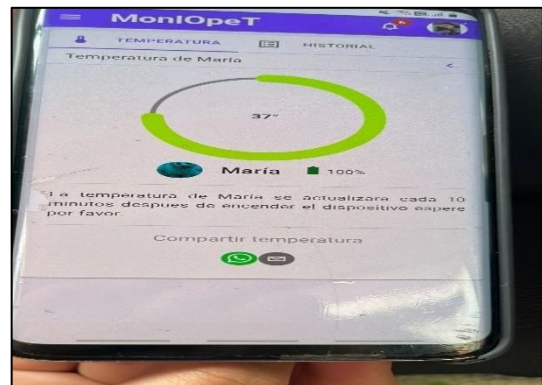
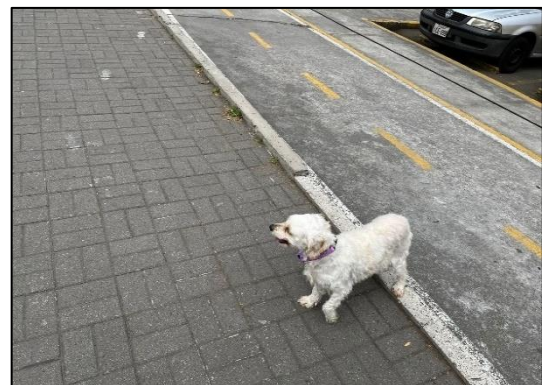
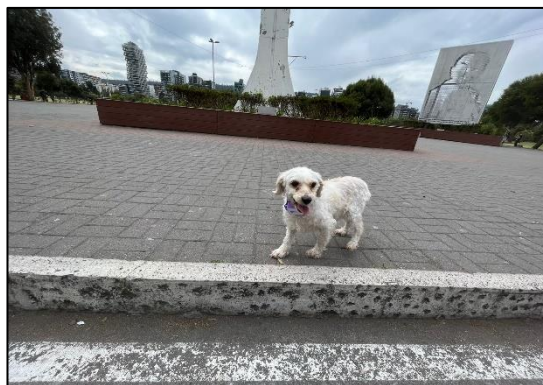






Las fotografías fueron realizadas en el parque El Ejido.

### Sector Norte de Quito



Las fotografías fueron realizadas en el parque La Carolina.



## Anexo 4

### Manual de usuario

#### Perfil usuario

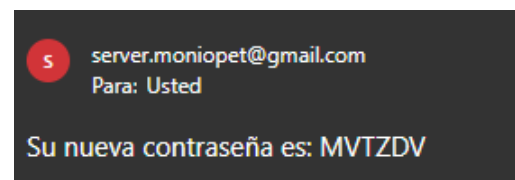
#### Registro e inicio de sesión

The image shows two side-by-side screenshots of the MonIOpeT user interface. The left screenshot is the registration page, titled 'Bienvenido', with the instruction 'Por favor ingrese los datos necesarios'. It features a profile picture upload area with a 'CARGAR IMAGEN' button. Below this are input fields for 'Nombre' (filled with 'Juan'), 'Apellido' (filled with 'Caisapanta'), 'Celular' (filled with '0999288791'), 'Correo Electrónico' (filled with 'juancaisapanta@hotmail.com'), and 'Contraseña' (masked with dots). A purple 'REGISTRARSE' button is at the bottom, along with a 'CANCELAR' button and a link 'Ya tiene una cuenta? Ingrese'. The right screenshot is the login page, also titled 'Bienvenido', with the instruction 'Por favor ingrese con su correo electrónico y contraseña'. It has input fields for 'Correo Electrónico' (filled with 'juancaisapanta@hotmail.com') and 'Contraseña' (masked with dots). A purple 'INICIAR SESIÓN' button is prominent, with a link 'Olvido su contraseña?' above it and 'Nuevo en la plataforma? Regístrate' below it.

El usuario dueño de la mascota al ingresar a la PWA MonIOpeT, tendrá que registrarse e ingresar la información requerida. Al registrarse el usuario podrá acceder a su cuenta de usuario al iniciar sesión.

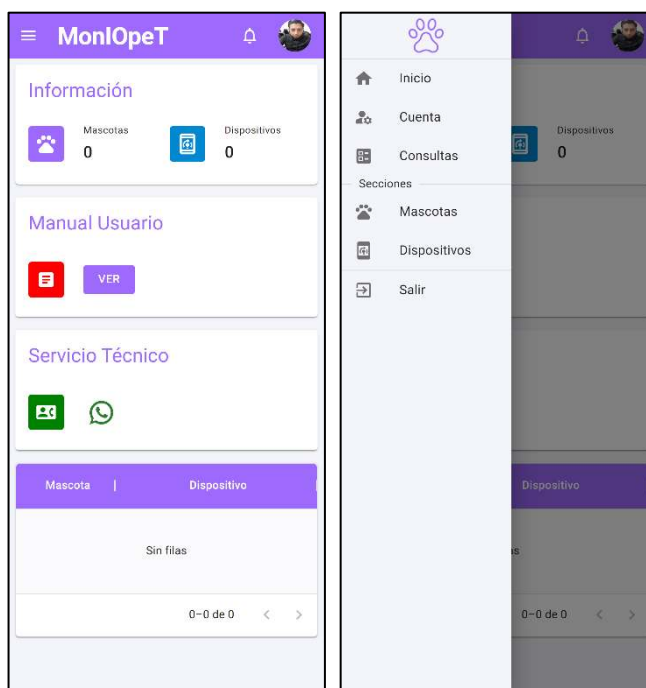
#### Olvido de contraseña

This screenshot shows the 'Olvido de contraseña' (Forgot Password) screen. It is titled 'Bienvenido' and instructs the user to 'Ingrese su correo electrónico para obtener su contraseña'. There is an input field for 'Correo Electrónico' containing 'juancaisapanta@hotmail.com'. Below the field is a purple 'SOLICITAR' button and a 'CANCELAR' button. At the bottom, there is a link 'Nuevo en la plataforma? Regístrate'.



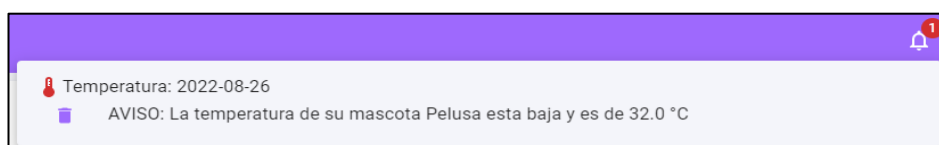
En caso de que el usuario ya se haya registrado con anterioridad y no recuerde su contraseña podrá solicitar una nueva accediendo a la opción **Olvido su contraseña?** En la página de inicio de sesión en la cual al ingresar su correo electrónico con el que se registró se enviará a su buzón de correos.

## Panel de usuario



Al iniciar sesión el usuario dueño de la mascota podrá ver el menú de opciones que puede gestionar y el panel donde se muestra la cantidad e información de las mascotas y dispositivos que tiene registrado, además de la posibilidad de ver el manual de usuario de la PWA y contactar al servicio técnico mediante WhatsApp.

## Notificaciones



El usuario dueño de la mascota puede ver mediante notificaciones el estado de su mascota y de su dispositivo. Las notificaciones enviadas también se guardan en el buzón del correo electrónico del usuario.

## Perfil de usuario

The image shows two screenshots of the MonOpeT mobile application. The left screenshot displays the 'Perfil' (Profile) page for user Juan Caisapanta. It includes a profile picture, a 'CAMBIAR IMAGEN' button, and input fields for Name (Juan), Surname (Caisapanta), Email (juancaisapanta@hotmail.com), Phone (0999288791), and Address. The right screenshot shows the 'Seguridad' (Security) page, featuring fields for 'Contraseña Nueva' (New Password) and 'Confirmar Nueva Contraseña' (Confirm New Password), along with 'ACTUALIZAR' (Update) and 'CANCELAR' (Cancel) buttons.

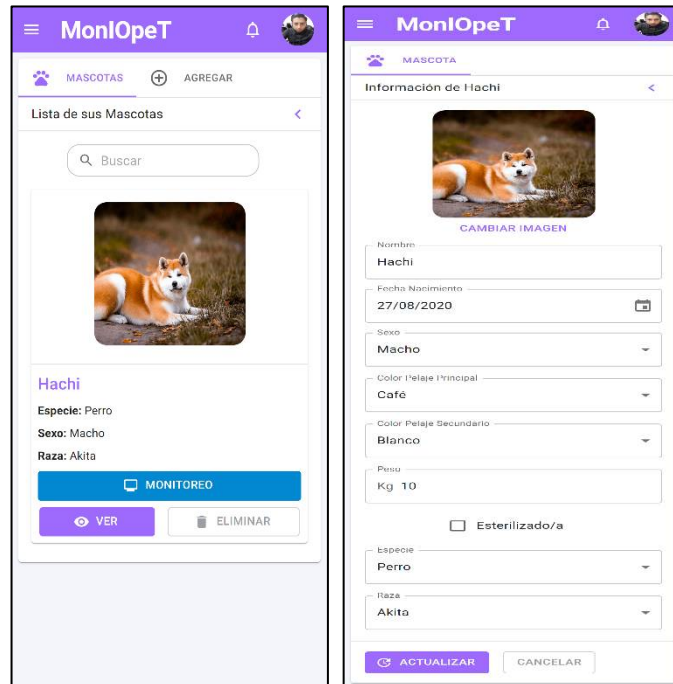
En la opción **Cuenta** del menú el usuario dueño de la mascota puede actualizar la información con la que se registró en la PWA.

## Gestión de las mascotas del usuario

The image displays the 'MASCOTAS' (Pets) section of the MonOpeT app, specifically the 'Agregar' (Add) form for registering a new pet. The form includes fields for Name (Hachi), Birth Date (23/08/2020), Sex (Macho), Primary Color (Café), Secondary Color (Blanco), Weight (Kg 9), and Species (Perro). There is also a checkbox for 'Esterilizado/a' (Sterilized) and a 'GUARDAR' (Save) button. Surrounding the main form are several dropdown menu options: 'Sexo' (Hembra, Macho), 'Color Pelaje Secundario' (Amarillo, Blanco, Café, Crema, Gris, Negro), 'Especie' (Perro, Gato), and 'Raza' (Akita, Dalmata, Mestizo, Romano, Shar pei).

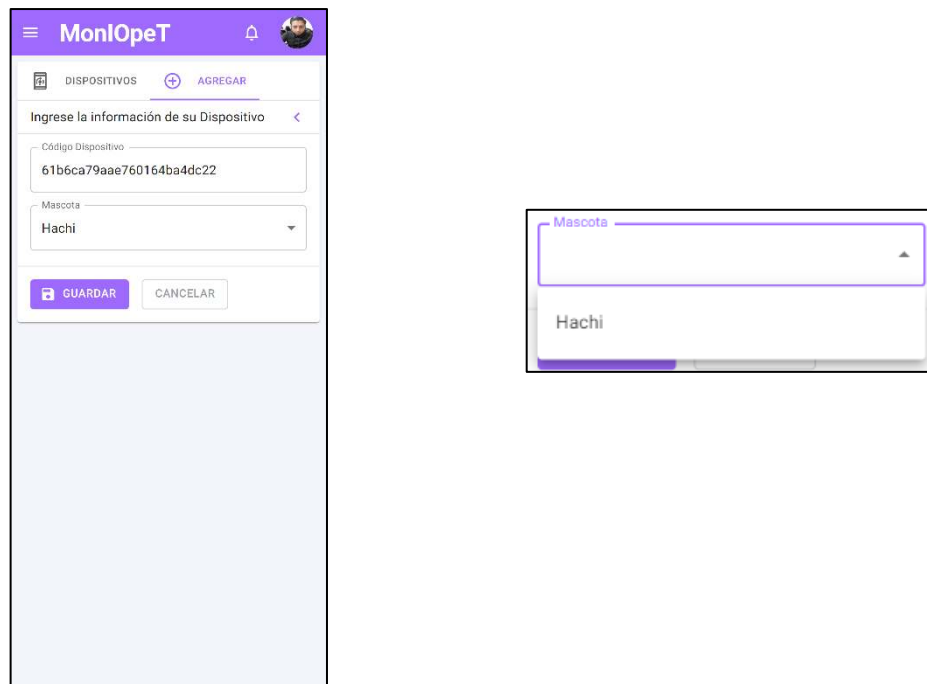
El usuario podrá registrar su mascota en la opción **Mascotas** del menú, mediante un formulario el cual entregará información que ayude a un registro más fácil.

## Información de la mascota



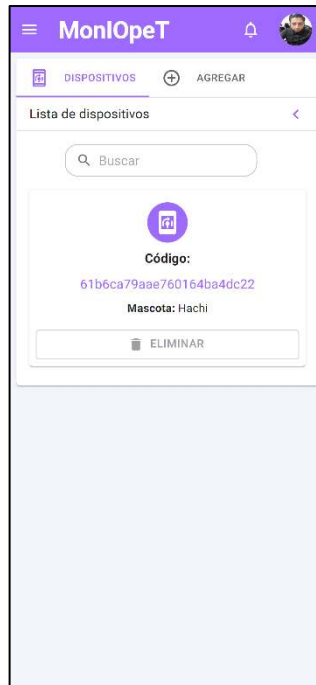
La mascota agregada tiene las opciones de ver su información y actualizarla, eliminar su información e ingresar a la sección de monitoreo.

## Gestión de los dispositivos del usuario



Para agregar un dispositivo de monitoreo es necesario seleccionar la opción del menú **Dispositivos** e ingresar el código que pertenece al dispositivo y seleccionar a la mascota a la cual se asignará dicho dispositivo.

## Información del dispositivo



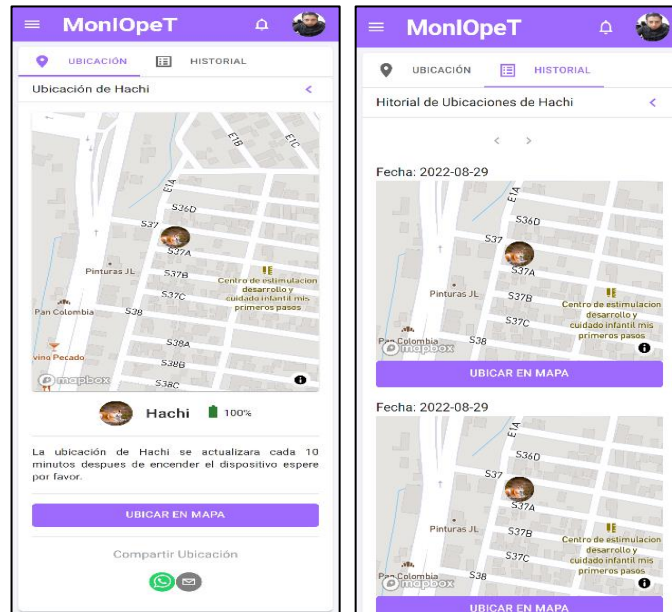
Al agregar con éxito el dispositivo se puede ver la información de este y la opción de eliminarlo para ser asignado a otra mascota.

## Secciones de monitoreo en tiempo real de la mascota



Al ingresar a la opción de **Monitoreo** de la mascota del usuario se muestra la sección de información en tiempo real, donde se puede seleccionar ver la ubicación o temperatura de la mascota.

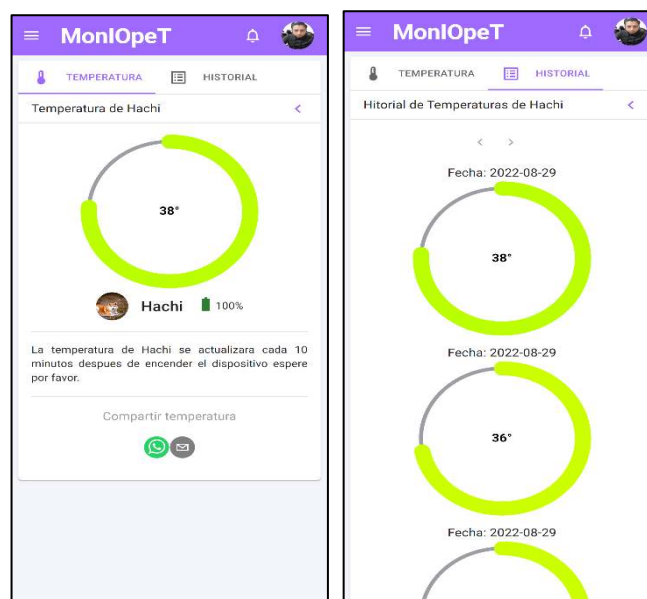
## Ubicación de la mascota en tiempo real de la mascota



La **Ubicación** de la mascota se muestra en un mapa que se actualiza cada cierto tiempo, la ubicación obtenida es posible compartirla por correo electrónico o WhatsApp y localizarla mediante el mapa de Google Maps que otorga una ruta con la cual llegar al objetivo.

Se presenta también el historial de ubicaciones, donde las ubicaciones que han sido obtenidas se guardan y son presentadas por la fecha en la que fue registrada.

## Temperatura de la mascota en tiempo real



La **Temperatura** de la mascota se muestra mediante un gráfico que cada cierto tiempo cambia de color dependiendo del estado de la temperatura, la temperatura obtenida es posible compartirla por correo electrónico o WhatsApp.

Se presenta también el historial de temperaturas, donde las temperaturas que han sido obtenidas se guardan y son presentadas por la fecha en la que fue registrada.

### Secciones administración de la mascota

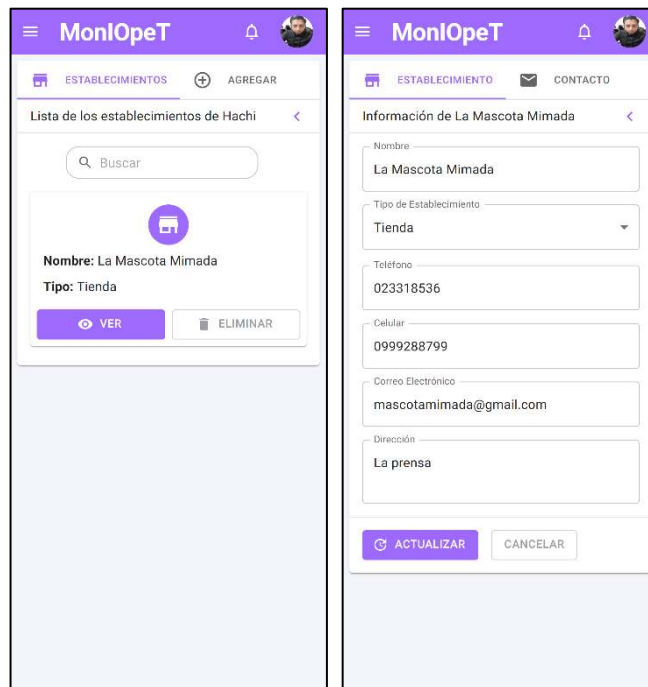


Al ingresar a la opción de monitoreo de la mascota del usuario se muestra la sección de administración de establecimiento, medicinas y profesionales de la mascota.

### Gestión de establecimientos de la mascota



Es posible agregar nuevos **Establecimientos** que frecuenta la mascota del usuario por medio de formularios.



Los establecimientos agregados se presentan con la posibilidad de ver la información de cada uno, donde estos se pueden eliminar y actualizar.

### Contacto al establecimiento de la mascota





Al ver la información del establecimiento es posible contactar al establecimiento mediante correo electrónico o WhatsApp con un mensaje por defecto o uno personalizado.

### Gestión de medicinas de la mascota

MoniOpeT

MEDICINAS AGREGAR

Ingrese la información de la Medicina Aplicada



CAMBIAR IMAGEN

Nombre: Naxpet

Fabricante: DRAG PHARMA

Lote: 1045

Aplicador: Carlos Santos

Tipo: Analgésico

Fecha Aplicación: 28/08/2022

Fecha Producción: 03/01/2022

Fecha Expiración: 02/01/2023

Descripción: Ninguna

GUARDAR CANCELAR

Tipo

Analgésico

Desparasitante

Vacuna

Fecha Expiración


Es posible agregar nuevas **Medicinas** que usa la mascota del usuario por medio de formularios.

MoniOpeT

MEDICINAS AGREGAR

Lista de las medicinas de Hachi

Buscar



Naxpet

Tipo: Analgésico

Aplicador: Carlos Santos

Fabricante: DRAG PHARMA


Fecha Aplicación: 28/08/2022

VER ELIMINAR

MoniOpeT

MEDICINA

Información de medicamento Naxpet



CAMBIAR IMAGEN

Nombre: Naxpet

Fabricante: DRAG PHARMA

Lote: 1045

Aplicador: Carlos Santos

Tipo: Analgésico

Fecha Aplicación: 28/08/2022

Fecha Producción: 03/01/2022

Fecha Expiración: 02/01/2023

Descripción: Ninguna

ACTUALIZAR CANCELAR

Las medicinas agregadas se presentan con la posibilidad de ver la información de cada una, donde estas se pueden eliminar y actualizar.

### Gestión de profesionales de la mascota

MoniOpeT

PROFESIONALES AGREGAR

Ingrese la información del Profesional

Nombre: Marta

Apellido: Solis

Profesión: Veterinario/a

Celular: 0999288792

Correo Electrónico: martasolis@hotmail.com

Dirección: El Conde

GUARDAR CANCELAR

Profesión

Peluquero/a

Veterinario/a

Es posible agregar nuevos **Profesionales** que visita la mascota del usuario por medio de formularios.

MoniOpeT

PROFESIONALES AGREGAR

Lista de los profesionales de Hachi

Buscar

Nombre: Marta

Apellido: Solis

Profesión: Veterinario/a

VER ELIMINAR

MoniOpeT

PROFESIONAL CONTACTO

Información de Marta Solis

Nombre: Marta

Apellido: Solis

Profesión: Veterinario/a

Celular: 0999288792

Correo Electrónico: martasolis@hotmail.com

Dirección: El Conde

ACTUALIZAR CANCELAR

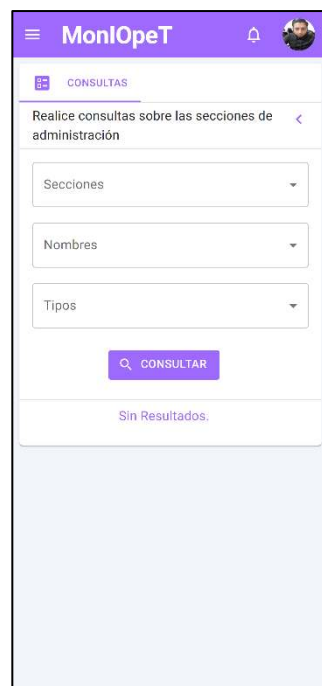
Los profesionales agregados se presentan con la posibilidad de ver la información de cada uno, donde estos se pueden eliminar y actualizar.

## Contacto al profesional de la mascota

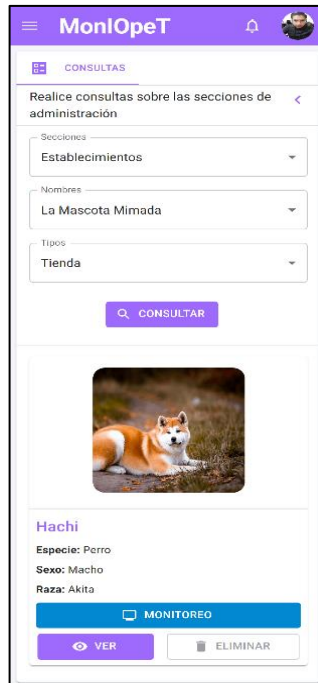


Al ver la información del profesional es posible contactar al profesional mediante correo electrónico o WhatsApp con un mensaje por defecto o uno personalizado.

## Consulta de mascotas por las secciones de administración



La opción **Consultas** del menú se refiere a obtener mascotas que cumplan con la información de los establecimientos, medicinas o profesionales.



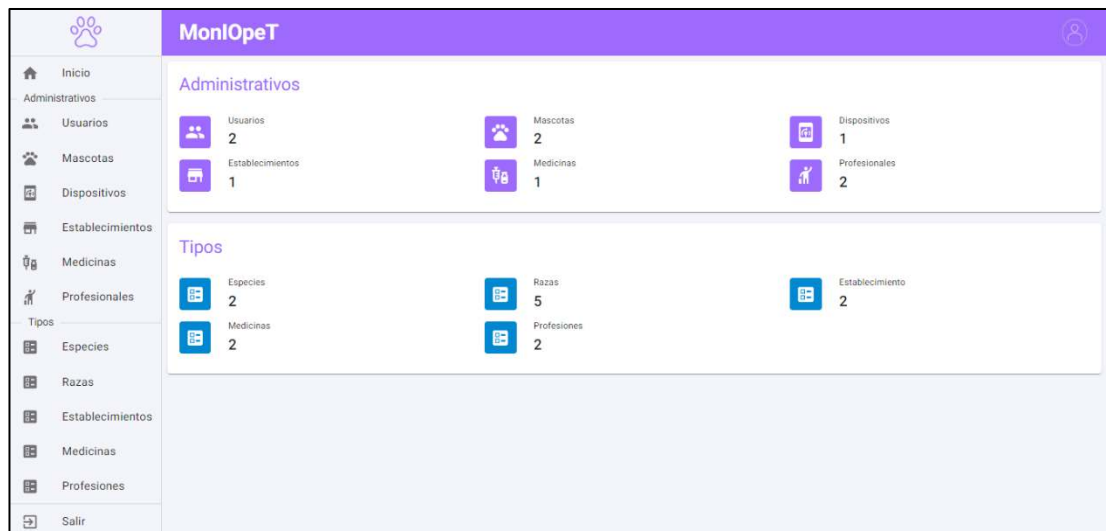
Al agregar la información requerida se obtendrá la mascota la cual cumple con los datos y poder realizar la gestión de misma.

## Perfil administrador



El usuario administrador no necesita registrarse ya que cuenta con un perfil personalizado.

## Panel del administrador

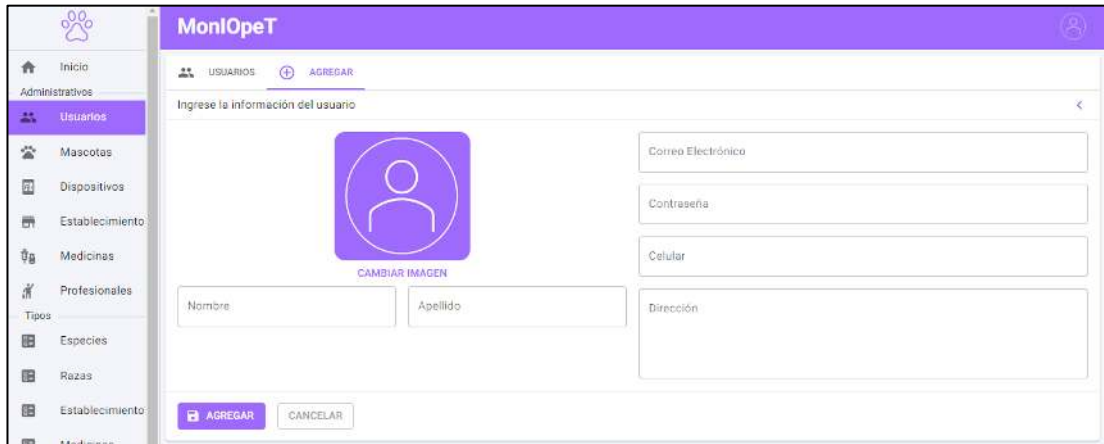


Al ingresar en el panel de administrador es posible ver la cantidad de datos que están agregados en la PWA MonIOpeT además del menú con las opciones a gestionar.

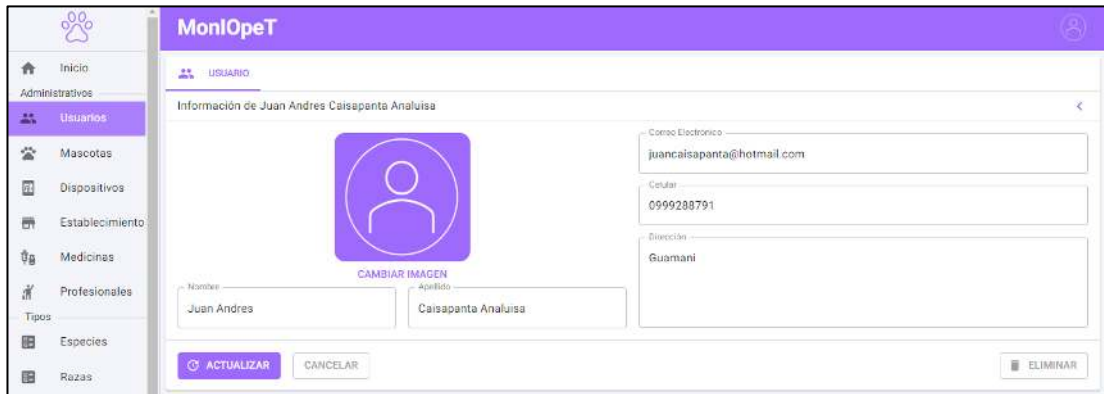
## Gestión de usuarios



Al ingresar a la opción de **Usuarios** se muestra una tabla con todos los usuarios de la PWA.

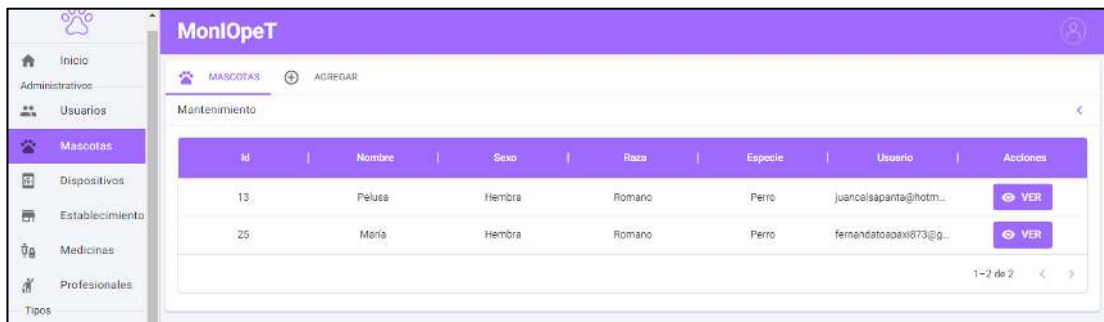


Es posible agregar un nuevo usuario mediante un formulario controlado.

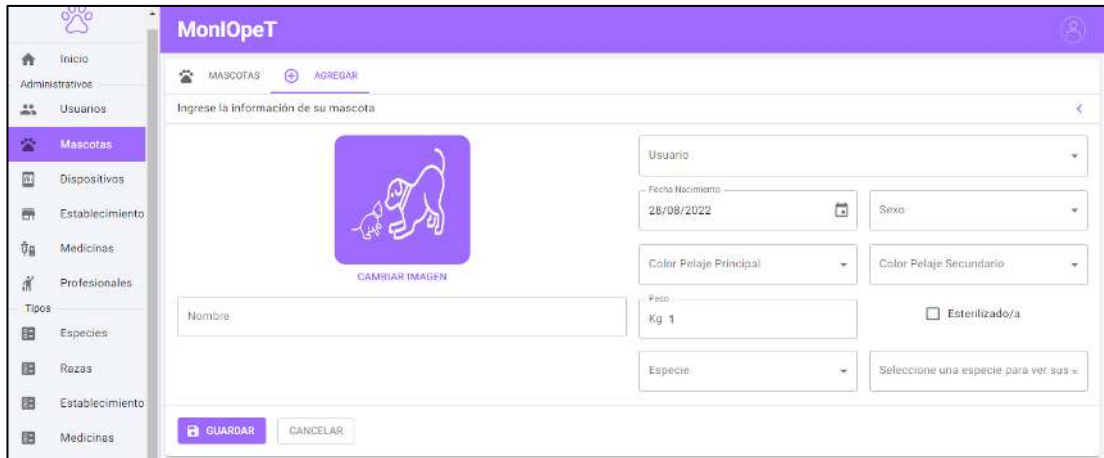


Es posible ver la información de un usuario y poder eliminarlo.

## Gestión de mascotas

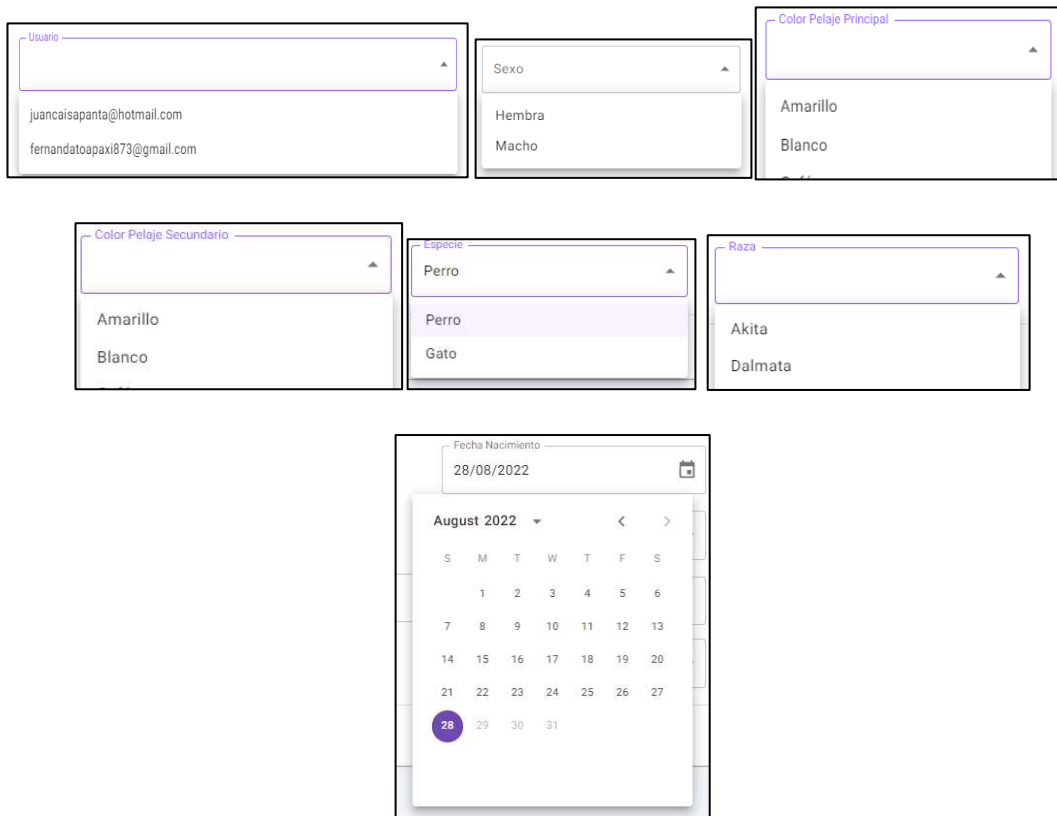


Al ingresar a la opción de **Mascotas** se muestra una tabla con todas las mascotas de la PWA.

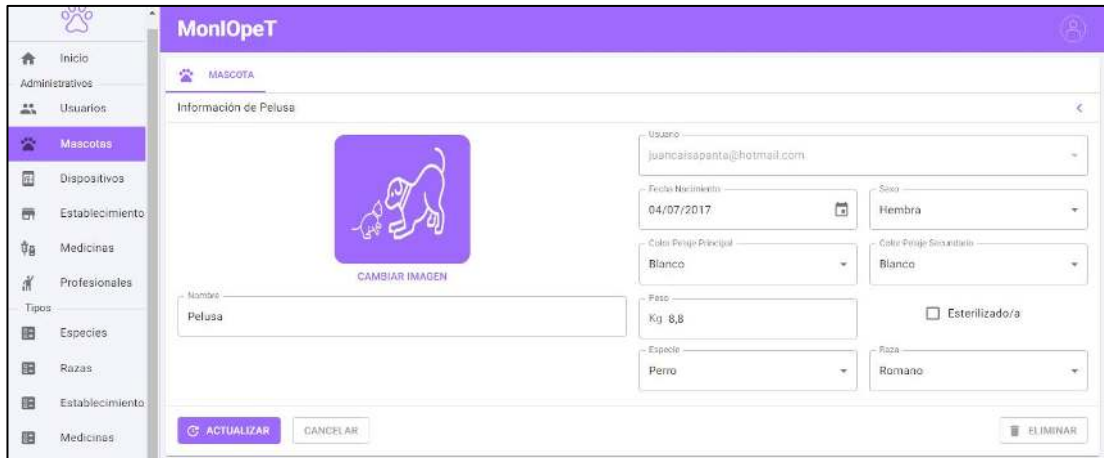


Es posible agregar una nueva mascota mediante un formulario controlado asignándole un usuario.

### Componentes del formulario de ingreso de información de una mascota



Los componentes de los formularios como las listas y calendarios facilitan el ingreso de información.

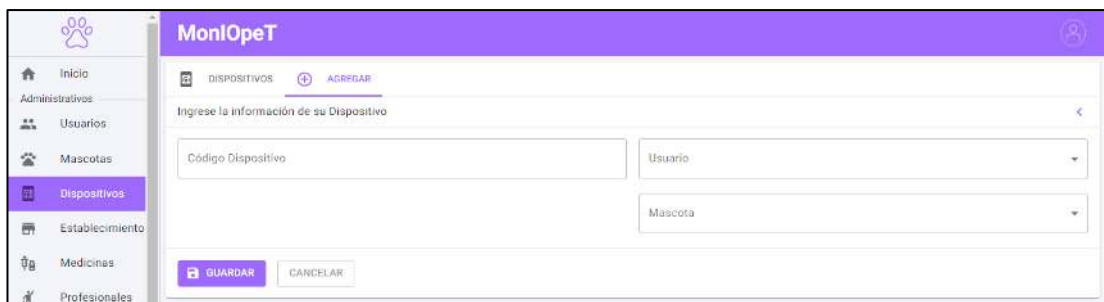


Es posible ver la información de una mascota y poder eliminarla.

### Gestión de dispositivos

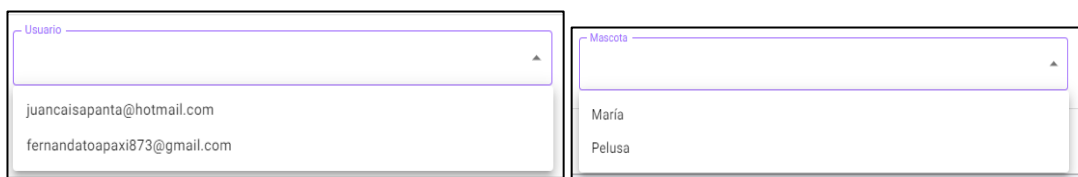


Al ingresar a la opción de **Dispositivos** se muestra una tabla con todos los dispositivos de la PWA con la posibilidad de eliminarlos.



Es posible agregar un nuevo dispositivo mediante un formulario controlado asignándole una mascota de un usuario.

### Componentes del formulario de ingreso de información de un dispositivo



Los componentes de los formularios como las listas facilitan el ingreso de información.



The screenshot shows the 'Historial' (History) section of the MoniOpeT application. It displays a table with the following data:

Id	Latitud	Longitud	Temperatura	Bateria	Código de dispositivo
404	-0.1829624	-78.48617	37	3050	61b6ca79eae760164ba4dc22
405	-0.2109073	-78.49904	37	3050	61b6ca79eae760164ba4dc22
406	-0.2109073	-78.49904	36	3050	61b6ca79eae760164ba4dc22
407	-0.295366	-78.54703	34	2965	61b6ca79eae760164ba4dc22
408	-0.1829624	-78.48617	36	2950	61b6ca79eae760164ba4dc22

Es posible ver la información que ha enviado un dispositivo.

### Gestión de establecimientos

The screenshot shows the 'Establecimientos' (Establishments) section of the MoniOpeT application. It displays a table with the following data:

Id	Nombre	Email	Mascota	Raza	Especie	Usuario	Acciones
2	Pelusa	pelusa@hotmail.com	Pelusa	Romano	Perro	juancaisapanta@ho...	VER

Al ingresar a la opción de **Establecimientos** se muestra una tabla con todos los establecimientos de la PWA.

The screenshot shows the 'Agregar' (Add) form in the MoniOpeT application. The form is titled 'Ingrese la información del Establecimiento' and contains the following fields:

- Nombre (Text input)
- Tipo de Establecimiento (Dropdown menu)
- Usuario (Dropdown menu)
- Teléfono (Text input)
- Celular (Text input)
- Mascota (Dropdown menu)
- Correo Electrónico (Text input)
- Dirección (Text input)

Buttons for 'GUARDAR' (Save) and 'CANCELAR' (Cancel) are located at the bottom of the form.

Es posible agregar un nuevo establecimiento mediante un formulario controlado asignándole una mascota de un usuario.

### Componentes del formulario de ingreso de información de un establecimiento

This image shows two dropdown menus from the form. The first is labeled 'Tipo de Establecimiento' and has options 'Peluqueria' and 'Veterinaria'. The second is labeled 'Usuario' and has options 'juancaisapanta@hotmail.com' and 'fernandatoapaxi873@gmail.com'.

This image shows a dropdown menu labeled 'Mascota' with options 'María' and 'Pelusa'.

Los componentes de los formularios como las listas facilitan el ingreso de información.

MoniOpeT

ESTABLECIMIENTO

Información de Patitas

Nombre: Patitas | Tipo de Establecimiento: Veterinaria | Usuario: juancaissapanta@hotmail.com

Teléfono: 3952300 | Celular: 0999288791 | Mascota: Pelusa

Correo Electrónico: patitas@hotmail.com | Dirección: El conde

ACTUALIZAR CANCELAR ELIMINAR

Es posible ver la información de un establecimiento y poder eliminarlo.

## Gestión de medicinas

MoniOpeT

MEDICINAS AGREGAR

Mantenimiento

Id	Nombre	Aplicador	Fecha de aplicación	Mascota	Usuario	Acciones
5	Navpet	María Lopez	22/07/2022	Pelusa	juancaissapanta@hotmail.com	VER

1-1 de 1

Al ingresar a la opción de **Medicinas** se muestra una tabla con todas las medicinas de la PWA.

MoniOpeT

MEDICINAS AGREGAR

Ingrese la información de la Medicina Aplicada

CAMBIAR IMAGEN

Nombre: | Fabricante: | Tipo: | Fecha Aplicación: 29/08/2022

Lote: | Aplicador: | Fecha Producción: 29/08/2022 | Fecha Expiración: 29/08/2022

Descripción: | Usuario: | Mascota:

GUARDAR CANCELAR

Es posible agregar una nueva medicina mediante un formulario controlado asignándole una mascota de un usuario.

## Componentes del formulario de ingreso de información de una medicina

Tipo

Desparasitante

Vacuna

Los componentes de los formularios como las listas facilitan el ingreso de información.

MoniOpeT

MEDICINA

Información de medicamento Naxpet

Usuario: juancaisapanta@hotmail.com

Mascota: Hachi

Lote: 1045 Aplicador: Carlos Santos

Tipo: Analgésico Fecha Aplicación: 28/08/2022

Fecha Producción: 03/01/2022 Fecha Expiración: 02/01/2023

Descripción: Ninguna

ACTUALIZAR CANCELAR ELIMINAR

Es posible ver la información de una medicina y poder eliminarla.

## Gestión de profesionales

MoniOpeT

PROFESIONALES AGREGAR

Mantenimiento

Id	Nombre	Apellido	Email	Profesión	Mascota	Especie	Usuario	Acciones
7	Andres	Caisapanta	juancaisapant...	Peluquero/a	Pelusa	Pero	juancaisapanta@...	VER
8	Hahaja	Nanabana	hahaha@gagaga...	Peluquero/a	Pelusa	Pero	juancaisapanta@...	VER

1-2 de 2

Al ingresar a la opción de **Profesionales** se muestra una tabla con todos los profesionales de la PWA.

MoniOpeT

PROFESIONALES AGREGAR

Ingrese la información del Profesional

Nombre Apellido Usuario

Profesión Celular Mascota

Correo Electrónico Dirección

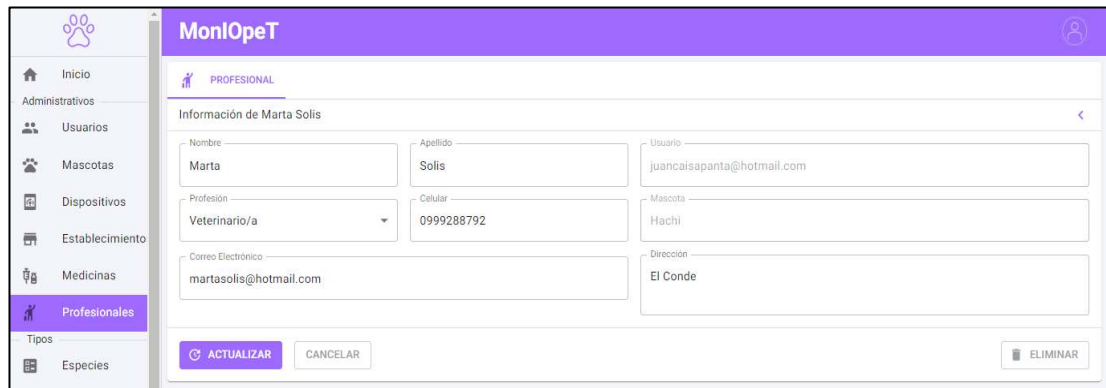
GUARDAR CANCELAR

Es posible agregar un nuevo profesional mediante un formulario controlado asignándole una mascota de un usuario.

## Componentes del formulario de ingreso de información de un profesional

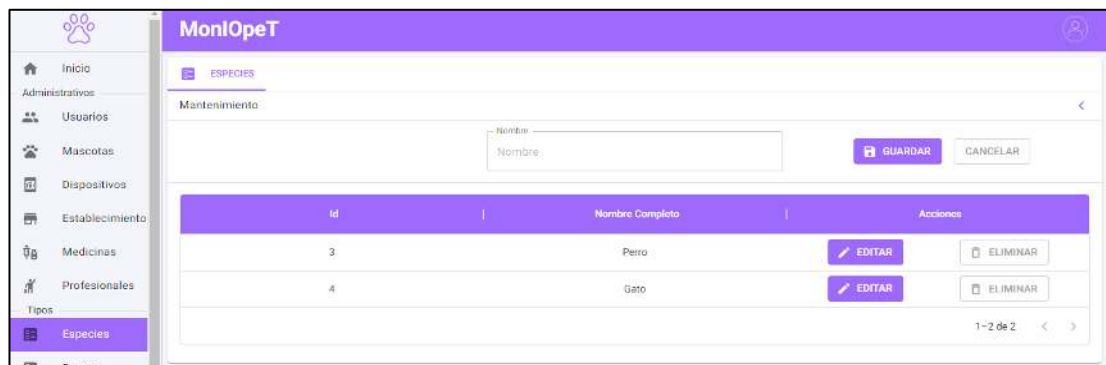


Los componentes de los formularios como las listas facilitan el ingreso de información.



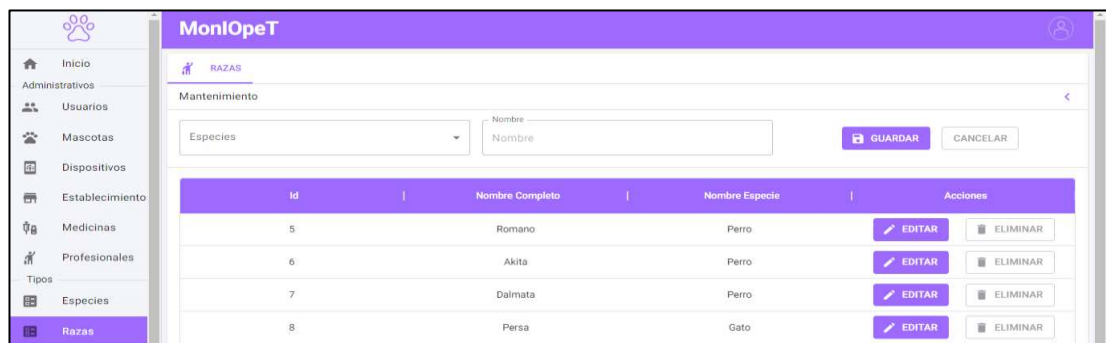
Es posible ver la información de un profesional y poder eliminarlo.

### Gestión de especies



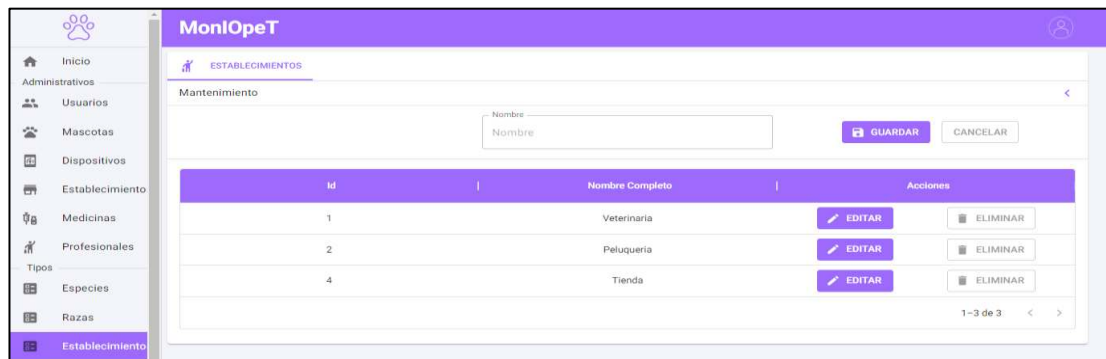
La gestión de **Especies** se lo realiza de manera sencilla mediante un campo en el cual se ingresa el nombre de la especie a guardar, con la posibilidad de editar y eliminar esta información seleccionando cada opción de la tabla.

### Gestión de razas



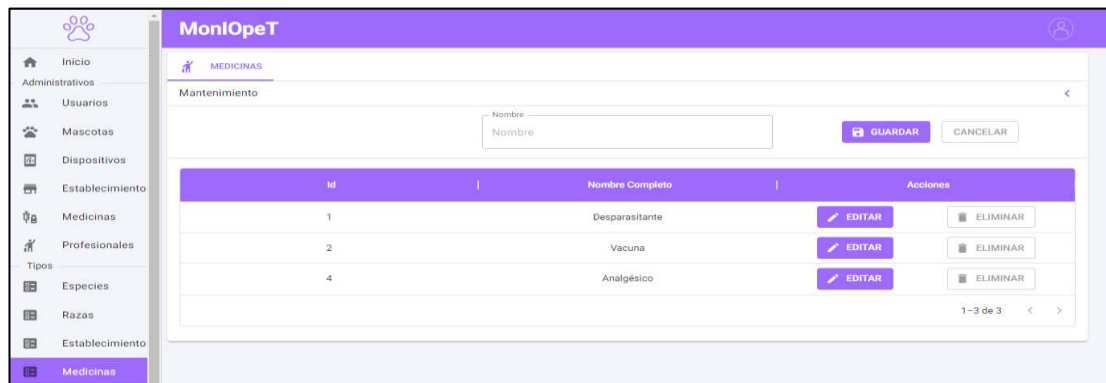
La gestión de **Razas** se lo realiza de manera sencilla mediante dos campos en los cuales se ingresa el nombre de la raza y se selecciona la especie de la raza a guardar, con la posibilidad de editar y eliminar esta información seleccionando cada opción de la tabla.

### Gestión de tipos de establecimientos



La gestión de los **Tipos de establecimientos** se lo realiza de manera sencilla mediante un campo en el cual se ingresa el nombre del tipo de establecimiento a guardar, con la posibilidad de editar y eliminar esta información seleccionando cada opción de la tabla.

### Gestión de tipos de medicinas



La gestión de los **Tipos de medicinas** se lo realiza de manera sencilla mediante un campo en el cual se ingresa el nombre del tipo de medicina a guardar, con la posibilidad de editar y eliminar esta información seleccionando cada opción de la tabla.

## Gestión de profesiones

MoniOpeT

PROFESIONES

Mantenimiento

Nombre:

Nombre:

GUARDAR CANCELAR

Id	Nombre Completo	Acciones
1	Veterinario/a	EDITAR ELIMINAR
2	Peluzero/a	EDITAR ELIMINAR

1-2 de 2 < >

La gestión de las **Profesiones** se lo realiza de manera sencilla mediante un campo en el cual se ingresa el nombre de la profesión a guardar, con la posibilidad de editar y eliminar esta información seleccionando cada opción de la tabla.