



**UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

**Construcción de una plataforma móvil que facilite
el proceso de atención de urgencias veterinarias.**

GUSTAVO ALBERTO ROJAS MUÑOZ

Profesor Guía: RODRIGO PAVEZ MADARIAGA

Memoria para optar al título de
Ingeniero Civil en Computación

Curicó – Chile
Marzo, 2020

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Two circular official stamps and handwritten signatures. The left stamp is from the 'DIRECCIÓN SISTEMA DE BIBLIOTECAS UNIVERSIDAD DE TALCA' and the right stamp is from the 'SISTEMA DE BIBLIOTECAS CAMPUS CURICO'.

Curicó, 2022

Dedicado a mi familia y a mi perro Capuchino (Q.E.P.D).

AGRADECIMIENTOS

Agradecer a mi padre y mi madre por apoyarme en todo momento en mi etapa como estudiante. Agradezco todos los valores, enseñanzas y palabras de apoyo que ellos me han dado. Doy gracias por enseñarme que nunca hay que rendirse y siempre hay seguir adelante sin importar lo difícil que sea el camino y si uno se cae en el transcurso, aprender a levantarse para seguir creciendo. También agradecer todo el apoyo económico que ellos me han brindado, sin lugar a dudas puedo decir que nunca me faltó nada y todo es gracia a ellos. Además quiero agradecerles por enseñarme el importante valor de querer y cuidar a los animales, ya que es gracia a este sentimiento que se gesta la creación de una herramienta que ayude a las mascotas.

Agradezco a mi hermano, por ayudarme en mi etapa como estudiante, ya que él me dio ese gran empujón que me permitió avanzar en la universidad. En definitiva, la ayuda de él, fue un elemento fundamental que me permitió llegar donde estoy ahora. También quiero agradecerle, los buenos momentos que pasamos, yo creo enormemente que si él no habría estado acá, mi vida como estudiante habría sido solitaria.

Quiero dar las gracias a mi abuela, ya que ella me abrió las puertas de su hogar para mi. Agradezco enormemente a ella , porque me dio un techo, comida y buenos momentos, además de aguantar mis enojos por eso le agradezco todo la paciencia y cariño que tuvo conmigo.

Doy gracias a mi amigos de la universidad, agradecer los buenos momentos que vivimos juntos en esta difícil, pero bonita etapa, las palabras de apoyo que me dieron por todas esas cosas doy mil gracias a ellos. Yo creo firmemente que la amistad es un factor importante para el éxito en la etapa universitaria y en la vida en general.

Por último agradecer a todas las personas que conocí en esta etapa. El hecho de conocerlas, me ayudó para llegar hasta acá, gracias por los buenos momentos que vivimos, todas esas cosas me ayudaron a crecer como la persona que soy hoy en día. En definitiva muchas gracias a todos.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	VIII
Índice de Tablas	XVII
Resumen	XIX
1. Introducción	20
1.1. Contexto	20
1.2. Definición del problema	21
1.3. Propuesta de solución	22
1.4. Objetivos	22
1.5. Alcances	23
1.6. Resumen del capítulo	23
2. Marco Teórico	24
2.1. Conceptos básicos	24
2.2. La sociedad y las mascotas	25
2.3. Mascotas preferida por los chilenos	25
2.4. Preocupación de los chilenos hacia las mascotas	26
2.5. Urgencias veterinarias	27
2.5.1. Proceso de atención de urgencias	28
2.5.2. Tipo de urgencias frecuentes	32
2.5.3. Información útil en las urgencias	33
2.6. Trabajo relacionados	34
2.6.1. VetMapp	34
2.6.2. Google Maps	35

2.6.3. WhatsApp	35
2.7. Tecnologías a utilizar	36
2.7.1. Patrón de Diseño Modelo-Vista-Controlador (MVC)	36
2.7.2. Android	37
2.7.3. Android Studio	39
2.7.4. Java	40
2.7.5. XML	42
2.7.6. Firebase	43
2.7.7. Git	44
2.7.8. Google Maps	45
2.8. Metodología iterativa incremental	46
2.9. Metodología de evaluación del proyecto	48
2.10. Resumen del capítulo	50
3. Metodología	51
3.1. Notificar urgencias a veterinarios	51
3.2. Aplicación metodología de desarrollo	53
3.2.1. Fase de análisis	53
3.2.2. Fase de diseño	55
3.2.3. Fase de implementación	57
3.2.4. Fase de pruebas	57
3.3. Aplicación metodología de evaluación	58
3.3.1. Encuesta	58
3.3.2. Experimentación	59
3.4. Planificación del proyecto	60
3.4.1. Planificación del desarrollo de software	60
3.4.2. Planificación de la evaluación del proyecto.	61
3.5. Resumen del capítulo	63
4. Implementación de la solución	64
4.1. Plan de trabajo	64
4.2. Requisitos	67
4.2.1. Casos de uso	68
4.3. Arquitectura	69

4.3.1.	Arquitectura física	70
4.3.2.	Arquitectura lógica	71
4.3.3.	Diagrama de clases	72
4.3.4.	Diagrama de secuencia	74
4.3.5.	Base de datos Firebase	75
4.3.6.	Implementación Firebase realtime database	76
4.3.7.	Implementación Firebase Authentication	76
4.3.8.	Implementación Firabase Storage	77
4.3.9.	Implementación Firebase Cloud Messasing	78
4.4.	Desarrollo Aplicación móvil	80
4.4.1.	Incremento 1	80
4.4.2.	Incremento 2	80
4.4.3.	Incremento 3	84
4.4.4.	Incremento 4	88
4.4.5.	Incremento 5	93
4.4.6.	Incremento 6	98
4.4.7.	Incremento 7	101
4.4.8.	Incremento 8	102
4.5.	Pruebas	105
4.6.	Resumen Capítulo	108
5.	Resultados	109
5.1.	Resultados objetivo general	109
5.2.	Resultados objetivos específicos	113
5.2.1.	Comprobación del objetivo específico n°1	114
5.2.2.	Comprobación del objetivo específico n°2	119
5.2.3.	Comprobación del objetivo específico n°3	124
5.3.	Resumen Capítulo	126
6.	Conclusión y trabajos futuros	127
6.1.	Conclusión general	127
6.2.	Trabajos futuros	128
	Glosario	130

Bibliografía	132
Anexos	
A: Resultados pruebas experimentación con la aplicación móvil	136
B: Resultado encuesta objetivos específicos	141
C: Navegación aplicación móvil	145
C.1. Navegación dueño de mascota	145
C.1.1. Registro e inicio de sesión	146
C.1.2. Creación perfil de mascota	146
C.1.3. Creación petición de urgencia para una mascota	147
C.1.4. Editar perfil de mascota	147
C.1.5. Ver detalles de la confirmación de la urgencia	148
C.1.6. Cancelar petición de urgencia que un dueño de mascota ha realizado.	149
C.1.7. Ver el detalle de una clínica veterinaria	149
C.1.8. Ver perfil del dueño de mascota	150
C.1.9. Compartir perfil mascota con otro dueño	150
C.1.10. Recibir solicitud de compartir mascota	151
C.1.11. Aviso cancelación petición de urgencia para el dueño de mascota.	151
C.1.12. Cerrar sesión	152
C.2. Navegación veterinario	153
C.2.1. Registro e inicio de sesión veterinario.	153
C.2.2. Asociación a una clínica ya existente	154
C.2.3. Recepción solicitud de asociación.	154
C.2.4. Editar información básica clínica veterinaria.	155
C.2.5. Editar horarios clínica veterinaria.	156
C.2.6. Editar posición clínica veterinaria.	156
C.2.7. Agregar y remover servicios clínica veterinaria.	157
C.2.8. Listar veterinarios que trabajan en una clínica.	157
C.2.9. Aviso cancelación petición de urgencia para el veterinario. . .	158
C.2.10. Revisar detalle petición de urgencia.	159
C.2.11. Perfil veterinario	160

C.2.12. Cancelar petición que el veterinario ha aceptado	160
C.2.13. Crear una nueva clínica veterinaria	161
C.2.14. Salir de la sesión	162
D: Documento de requisitos	163
D.1. Introducción	163
D.1.1. Propósito	163
D.1.2. Alcances	164
D.1.3. Contexto	164
D.2. Descripción general	165
D.2.1. Características de los usuarios	165
D.2.2. Suposiciones y dependencias	165
D.2.3. Descripción del modelo	166
D.3. Requisitos del sistema	176
D.3.1. Requisitos de usuario	176
D.3.2. Requisitos de sistema	195
D.3.3. Matriz de trazado requisitos de usuario vs. requisitos de sistema	223

ÍNDICE DE FIGURAS

	página
2.1. Porcentaje de hogares chilenos que tiene mascota entre los años 2014 a 2018. Fuente: Microestudio GfK: los chilenos y sus mascotas 2018 .	25
2.2. Tipo de mascota que hay en los hogares de Chile. Fuente: Microestudio GfK: los chilenos y sus mascotas 2018	26
2.3. Gráfico que muestra el porcentaje promedio de atenciones normales y de urgencia en 10 clínicas veterinarias de la ciudad de Curicó.	27
2.4. Gráfico que muestra el comportamiento de los dueños cuando su mascota presenta una urgencia.	28
2.5. Gráfico que muestra el comportamiento de los dueños cuando llegan a la clínica veterinaria y hay demora en la atención.	29
2.6. Patrón de diseño Modelo-Vista-Controlador. Fuente: Elaboración propia.	37
2.7. MarketShare del sistema operativo Android en Chile entre enero y agosto de 2019. Fuente: Datos obtenidos del sitio StatCounter.	39
2.8. Lenguajes de programación, Scripting y de Marcado más ocupados en 2019. Fuente: Datos obtenidos de StackOverflow. Fecha: consultado el 21-09-2019.	42
2.9. Ciclo desarrollo metodología incremental. Fuente: Elaboración propia.	46
2.10. Fases de la experimentación en ingeniería de software.	49
3.1. Representación del flujo de envío de notificaciones entre un dueño de mascota y veterinario.	52
3.2. Patrón Modelo-Vista-Controlador utilizando la configuración de carpetas que Android Studio crea defecto.	56
3.3. Tareas de una iteración de proyecto.	61
3.4. Tareas de la planificación evaluación del proyecto.	61
4.1. Caso de uso asociado al requisito RU009.	69
4.2. Arquitectura física	70
4.3. Arquitectura lógica	71
4.4. Diagrama de clases del proyecto	73

4.5. Diagrama de secuencia de la interacción entre el dueño de mascota y el veterinario.	74
4.6. Formato JSON de un objeto almacenado en Firebase. Fuente: Elaboración propia	75
4.7. Declaración de las librerías necesarias para usar Firebase en Android Studio.	76
4.8. Declaración de librería que permita autenticar a un usuario.	76
4.9. Panel de control de Firebase Authentication.	77
4.10. Librería que permite trabajar con Firebase Storage	77
4.11. Explorador de carpetas de Firebase Storage.	78
4.12. Sentencia que permita usar las funcionales que Firebase Cloud Messaging entrega.	79
4.13. Estructura en formato <i>JSON</i> de la notificación.	79
4.14. Menú principal dueño de mascota.	81
4.15. Formulario para crear una mascota.	82
4.16. Datos de la mascota almacenados en la base de datos.	83
4.17. Sentencias para implementar la funcionalidades de la librería <i>Glide</i>	83
4.18. Listado de archivos que han sido almacenados en Firebase storage.	84
4.19. Menú para visualizar las clínicas veterinarias en el mapa	85
4.20. Diálogo de selección de mascota para la petición de una urgencia.	86
4.21. Pantalla para agregar síntomas para la petición de urgencia.	87
4.22. Opciones para proporcionar una imagen y video de la mascota para la urgencia.	88
4.23. Sentencia para usar los métodos de <i>SphericalUtils</i>	90
4.24. Firma del método que permite calcular distancia entre dos puntos geográficos.	90
4.25. Datos de la petición de urgencia almacenadas en la base de datos.	92
4.26. Menú principal veterinario.	93
4.27. Notificación avisando al veterinario que hay una mascota que tiene una atención de urgencia.	94
4.28. Información básica de la mascota en la petición de urgencia.	95
4.29. Información relacionada a la urgencia de la mascota en la petición de urgencia.	96
4.30. Opciones para aceptar o cancelar la petición de urgencia.	97

4.31. Lista de peticiones aceptadas por un veterinario.	98
4.32. Notificación que el dueño de mascota recibe cuando un veterinario acepta la petición de urgencia.	99
4.33. Detalles confirmación de la petición de urgencia.	100
4.34. Sentencia para implementar la librería <i>TagView</i>	100
4.35. Librería para implementar la funcionalidad de zoom.	101
4.36. Zoom realizado a la imagen de la petición de urgencia.	102
4.37. Menú para visualizar las clínicas veterinarias en el mapa	103
4.38. Diálogo de búsqueda de clínicas veterinarias que estén abiertas y/o que ofrezcan ciertos servicios.	103
4.39. Mensaje de éxito de la operación junto con la mascota creada en la lista.	105
4.40. Registro en formato JSON en la base de datos de la mascota creada.	106
4.42. Notificación de confirmación que recibe un dueño de mascota cuando un veterinario acepta la petición	106
4.41. Registro en formato JSON en la base de datos de la mascota creada.	107
4.43. Registro en formato JSON de la petición de urgencia en donde se resalta la fecha de aceptación de ésta.	107
5.1. Tiempo promedio de búsqueda sin y con el uso de la aplicación móvil.	113
5.2. Gráfico que muestra los resultados sobre la facilidad que tiene el dueño de mascota de crear una mascota en la aplicación móvil.	114
5.3. Gráfico que muestra los resultados sobre la facilidad que tiene el dueño de ingresar la información de la mascota en la aplicación móvil.	115
5.4. Gráfico que muestra los resultados sobre la facilidad que tiene el dueño de mascota de crear una mascota en la aplicación móvil.	115
5.5. Gráfico que muestra los resultados respecto a la facilidad que tiene el usuario en seleccionar una mascota para la urgencia.	116
5.6. Gráfico que muestra los resultados en relación al agrado que tiene el usuario al proporcionar información de los síntomas de una mascota.	116
5.7. Gráfico que muestra los resultados respecto a si la aplicación móvil permite ingresar una imagen de la mascota al momento de crear una urgencia.	117

5.8. Gráfico de los resultados con respecto a si la aplicación móvil permite ingresar un vídeo de la mascota al momento de crear una urgencia. . 117

5.9. Gráfico que muestra los resultados sobre cantidad de información que proporciona el dueño de mascota ayuda al veterinario en la urgencia. 118

5.10. Gráfico que muestra los resultados obtenidos al consultar a los veterinarios sobre la información relacionada a la urgencia de la mascota. 119

5.11. Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascota si el sistema de notificaciones ha servido para avisar a un veterinario. 120

5.12. Gráfico que muestra los resultados obtenidos al consultar a dueños de mascota sobre la claridad del mensaje de confirmación. 120

5.13. Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascota sobre la utilidad de este tipo de comunicación con el veterinario. 121

5.14. Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascota sobre las indicaciones que entrega la aplicación para dirigirse a la clínica veterinaria que atenderá la urgencia. 121

5.15. Gráfico que muestra los resultados obtenidos al consultar a los veterinarios sobre si el sistema de notificaciones es útil para avisar si una mascota tiene una urgencia. 122

5.16. Gráfico que muestra los resultados obtenidos al consultar a los veterinarios sobre la claridad del mensaje de la notificación cuando el dueño avisa de la urgencia. 123

5.17. Gráfico que muestra los resultados obtenidos al consultar a los veterinarios sobre la utilidad de este tipo de comunicación con el dueño de mascota. 123

5.18. Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascotas sobre si la aplicación sirve para indicar donde se encuentran las clínicas veterinarias. 124

5.19. Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascota sobre la utilidad de la información que muestra un marcador. 125

5.20. Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascota sobre la utilidad de la información en detalle de una clínica veterinaria. 125

C.1. Diagrama de navegación de la aplicación móvil para un dueño de mascota.	145
C.2. Proceso de registro e inicio de sesión de un dueño de mascota.	146
C.3. Proceso creación del perfil de una mascota.	146
C.4. Registro en formato JSON de la petición de urgencia en donde se resalta la fecha de aceptación de ésta.	147
C.5. Proceso de editar el perfil de una mascota.	147
C.6. Proceso de ver el detalle de la petición de urgencia.	148
C.7. Proceso de cancelar una petición de urgencia.	149
C.8. Proceso para visualizar información detallada de una clínica veterinaria.	149
C.9. Proceso para ver y editar el perfil de un usuario dueño de mascota.	150
C.10. Proceso de compartir la mascota con otro dueño de mascota.	150
C.11. Proceso de recibir una solicitud cuando un dueño comparte el perfil de una mascota	151
C.12. Proceso de recibir una notificación cuando un veterinario cancela una petición que aceptó	151
C.13. Proceso de salir de la sesión.	152
C.14. Proceso buscar clínicas veterinarias abiertas y/o por servicios que éstas ofrezcan.	152
C.15. Navegabilidad de un veterinario que no se encuentra asociado a ninguna clínica.	153
C.16. Proceso de registro e inicio de sesión de un veterinario.	153
C.17. Proceso de asociarse con una clínica veterinaria.	154
C.18. Proceso de aceptación o rechazo de asociación a una clínica veterinaria.	154
C.19. Navegabilidad de un veterinario que ya se encuentra asociado a una clínica.	155
C.20. Proceso de editar la información básica de una clínica veterinaria.	155
C.21. Proceso de editar los horarios de apertura y cierre de una clínica veterinaria.	156
C.22. Proceso de editar la ubicación de una clínica veterinaria.	156
C.23. Proceso de agregar o eliminar servicios que ofrece una clínica veterinaria.	157
C.24. Proceso de listar los veterinarios que trabajan o pertenecen a una clínica veterinaria.	157
C.25. Proceso de notificar que un dueño de mascota ha cancelado la petición.	158

C.26. Proceso de revisar una petición de urgencia.	159
C.27. Proceso de ver el perfil de un veterinario	160
C.28. Proceso cuando el médico veterinario decide cancelar una petición que él ya acepto.	160
C.29. Proceso de crear una clínica veterinaria.	161
C.30. Proceso de cerrar la sesión de un veterinario.	162
D.1. Caso de uso 1	166
D.2. Caso de uso 2	166
D.3. Caso de uso 3	167
D.4. Caso de uso 4	167
D.5. Caso de uso 5	168
D.6. Caso de uso 6	168
D.7. Caso de uso 7	169
D.8. Caso de uso 8	169
D.9. Caso de uso 9	170
D.10. Caso de uso 10	170
D.11. Caso de uso 11	171
D.12. Caso de uso 12	171
D.13. Caso de uso 13	172
D.14. Caso de uso 14	172
D.15. Caso de uso 15	173
D.16. Caso de uso 16	173
D.17. Caso de uso 17	174
D.18. Caso de uso 18	174
D.19. Caso de uso 19	175
D.20. Caso de uso 20	175
D.21. Caso de uso 21	176
D.22. Requisito de usuario 1.	176
D.23. Requisito de usuario 2	177
D.24. Requisito de usuario 3	177
D.25. Requisito de usuario 4	178
D.26. Requisito de usuario 5.	178
D.27. Requisito de usuario 6	179

D.28.Requisito de usuario 7	179
D.29.Requisito de usuario 9	180
D.30.Requisito de usuario 10	180
D.31.Requisito de usuario 11.	181
D.32.Requisito de usuario 12	181
D.33.Requisito de usuario 13	182
D.34.Requisito de usuario 14	182
D.35.Requisito de usuario 15.	183
D.36.Requisito de usuario 16	183
D.37.Requisito de usuario 17	184
D.38.Requisito de usuario 18	184
D.39.Requisito de usuario 19	185
D.40.Requisito de usuario 20	185
D.41.Requisito de usuario 21.	186
D.42.Requisito de usuario 22	186
D.43.Requisito de usuario 23	187
D.44.Requisito de usuario 24	187
D.45.Requisito de usuario 25.	188
D.46.Requisito de usuario 26	188
D.47.Requisito de usuario 27	189
D.48.Requisito de usuario 28	189
D.49.Requisito de usuario 29	190
D.50.Requisito de usuario 30	190
D.51.Requisito de usuario 31.	191
D.52.Requisito de usuario 32	191
D.53.Requisito de usuario 33	192
D.54.Requisito de usuario 34	192
D.55.Requisito de usuario 35.	193
D.56.Requisito de usuario 36	193
D.57.Requisito de usuario 37	194
D.58.Requisito de usuario 38	194
D.59.Requisito de sistema 1.	195
D.60.Requisito de sistema 2.	195
D.61.Requisito de sistema 3.	196

D.62.Requisito de sistema 4. 196

D.63.Requisito de sistema 5. 197

D.64.Requisito de sistema 6. 197

D.65.Requisito de sistema 7. 198

D.66.Requisito de sistema 8. 198

D.67.Requisito de sistema 9. 199

D.68.Requisito de sistema 10. 199

D.69.Requisito de sistema 11. 200

D.70.Requisito de sistema 12. 200

D.71.Requisito de sistema 13. 201

D.72.Requisito de sistema 14. 201

D.73.Requisito de sistema 15. 202

D.74.Requisito de sistema 16. 202

D.75.Requisito de sistema 17. 203

D.76.Requisito de sistema 18. 203

D.77.Requisito de sistema 19. 204

D.78.Requisito de sistema 20. 204

D.79.Requisito de sistema 21. 205

D.80.Requisito de sistema 22. 205

D.81.Requisito de sistema 23. 206

D.82.Requisito de sistema 24. 206

D.83.Requisito de sistema 25. 207

D.84.Requisito de sistema 26. 207

D.85.Requisito de sistema 27. 208

D.86.Requisito de sistema 28. 208

D.87.Requisito de sistema 29. 209

D.88.Requisito de sistema 30. 209

D.89.Requisito de sistema 31. 210

D.90.Requisito de sistema 32. 210

D.91.Requisito de sistema 33. 211

D.92.Requisito de sistema 34. 211

D.93.Requisito de sistema 35. 212

D.94.Requisito de sistema 36. 212

D.95.Requisito de sistema 37. 213

D.96.Requisito de sistema 38.	213
D.97.Requisito de sistema 39.	214
D.98.Requisito de sistema 40.	214
D.99.Requisito de sistema 41.	215
D.100Requisito de sistema 42.	215
D.101Requisito de sistema 43.	216
D.102Requisito de sistema 44.	216
D.103Requisito de sistema 45.	217
D.104Requisito de sistema 46.	217
D.105Requisito de sistema 47.	218
D.106Requisito de sistema 48.	218
D.107Requisito de sistema 49.	219
D.108Requisito de sistema 50.	219
D.109Requisito de sistema 51.	220
D.110Requisito de sistema 52.	220
D.111Requisito de sistema 53.	221
D.112Requisito de sistema 54.	221
D.113Requisito de sistema 55.	222
D.114Matriz de trazabilidad de requisitos e usuarios vs. requisitos de sistema.	223

ÍNDICE DE TABLAS

	página
3.1. Estructura de especificación de requisitos.	54
4.1. Especificación de un requisito funcional del proyecto.	67
4.2. Especificación de un requisito de software del proyecto.	68
4.3. Plan de pruebas de la creación de una mascota.	105
4.4. Plan de pruebas del envío de una petición de urgencia que realiza un dueño de mascota.	106
4.5. Plan de prueba para confirmación de una petición de urgencia.	107
5.1. Prueba de experimentación con el dueño de mascota 1.	110
5.2. Prueba de experimentación con el dueño de mascota 2.	110
5.3. Prueba de experimentación con el dueño de mascota 3.	111
5.4. Prueba de experimentación con el dueño de mascota 4.	111
5.5. Tabla de comparaciones de tiempos obtenidos en la encuesta y con la aplicación móvil.	112
A.1. Prueba de experimentación con el dueño de mascota 5.	136
A.2. Prueba de experimentación con el dueño de mascota 6.	137
A.3. Prueba de experimentación con el dueño de mascota 7.	137
A.4. Prueba de experimentación con el dueño de mascota 8.	138
A.5. Prueba de experimentación con el dueño de mascota 9.	138
A.6. Prueba de experimentación con el dueño de mascota 10.	139
A.7. Prueba de experimentación con el dueño de mascota 11.	139
A.8. Prueba de experimentación con el dueño de mascota 12.	140
B.1. Resultados de la encuesta Likert, realizada a los dueños de mascota relacionada a la facilidad de suministrar información de la mascota a la clínica veterinaria.	141
B.2. Resultados de la encuesta realizada a los veterinarios relacionada a la facilidad de suministrar información de la mascota a la clínica veteri- naria usando escala Likert.	141
B.3. Resultados de la encuesta realizada a los dueños de mascota relaciona- da a la facultad de proporcionar síntomas de la mascota en la urgencia.	142

B.4. Resultados de la encuesta realizada a los veterinarios relacionada a la facilidad de suministrar información de la mascota a la clínica veterinaria.	142
B.5. Resultados de la encuesta realizada a los dueños de mascota en relación al sistema de notificaciones de la aplicación	143
B.6. Resultados de la encuesta realizada a los veterinarios en relación al sistema de notificaciones de la aplicación.	143
B.7. Resultados de la encuesta realizada a los dueños de mascota en relación a la visualización de las clínicas veterinarias.	144
D.1. Descripción de los usuarios de la aplicación móvil	165

RESUMEN

El presente documento detalla el proceso de creación de una solución tecnológica que ayude a los usuarios, que son dueños de una mascota, a encontrar una clínica veterinaria apta para atender una urgencia. Para ello se pretende brindar un aviso oportuno a los médicos veterinarios de la situación, ya que con esto, ellos pueden preparar los elementos necesarios y así lograr una atención más efectiva. En definitiva la herramienta pretende dar una sensación de tranquilidad al dueño de una mascota, sabiendo que hay alguien que está dispuesto a atender la urgencia.

Para solucionar la problemática expuesta anteriormente, se pretende crear una aplicación móvil, en donde un usuario que tenga mascota puede enviar notificaciones a los veterinarios con información de los síntomas de la mascota, así ellos, con la información previamente suministrada por el dueño, determinar si la clínica veterinaria se encuentra en condiciones de atender la urgencia. Lo anterior buscar evitar o reducir el tiempo involucrado en la búsqueda de una clínica veterinaria apta para responder a nuestra urgencia. Por otro lado, permitiría a la clínica veterinaria estar al tanto y preparado para recibir la urgencia.

El documento presenta todo lo esencial para solucionar la problemática expuesta, abordando las bases necesarias para comprender el propósito del proyecto, además de mostrar toda la teoría, herramientas tecnológicas y metodológicas que se necesitan para implementar la solución.

El proyecto se desarrolla usando la metodología iterativa incremental, en donde el trabajo fue separado en 8 incrementos de 2 semanas cada uno para construir la aplicación móvil. Cada incremento involucra aplicar la fases de análisis, diseño, construcción y validación del software. Para evaluar los objetivos del proyecto se usa el proceso de experimentación mediante la aplicación móvil y encuestas para obtener datos relacionado a las urgencias veterinarias. Por su parte estos datos son relevantes para aplicar la validez de este proyecto.

Por último en el documento se exponen los resultados obtenidos en la fase de experimentación y se realiza un análisis de ellos. Finalmente se hace una conclusión del proyecto en general y de los resultados obtenidos, además se mencionan trabajos futuros o funcionalidades que se pretenden desarrollar con la aplicación móvil.

1. Introducción

Actualmente en Chile existe una cantidad importante de hogares que tiene una mascota, las cuales pasan a ser un miembro más al interior de las familias y cuando estas presentan problemas graves de salud, los dueños tienen que buscar una clínica veterinaria que pueda atender lo antes posible la urgencia y así no comprometer más la vida de la mascota afectada. Lo anterior muchas veces resulta problemático porque las clínicas veterinarias por diversas razones no pueden atender la urgencia. Frente a esta situación se plantea una solución que busca dar una rápida respuesta a la urgencia veterinaria.

El objetivo del presente capítulo es dar a conocer las bases para entender el problema a tratar, mencionando el origen, motivación, solución propuesta, objetivos a cumplir y alcances del trabajo a realizar.

1.1. Contexto

Un estudio realizado por la encuesta Adimark ¹ en 2018 arrojó que seis de cada diez hogares en Chile tiene una mascota, ya sea perro, gato o de otro tipo [1, página 8]. Al existir un gran número de mascotas en nuestro entorno, es altamente probable que requieran asistencia veterinaria producto de una urgencia médica, la cual si no es tratada a tiempo puede agravar o conllevar la muerte de esta, por lo que es fundamental encontrar rápidamente una clínica veterinaria que le pueda prestar atención médica y así salvar o mejorar la condición de salud de la mascota.

En estos casos las personas elijen la clínica veterinaria más cercana o la que frecuentan regularmente, lo cual no siempre significa que esta sean apta para atender la

¹Adimark, actualmente GfK, es una empresa de investigación de mercado y opinión pública.

urgencia, ya sea porque no cuenta con el implemento necesario, están ocupadas atendiendo otros casos o simplemente están cerradas. Lo anterior provocaría la búsqueda de otra clínica veterinaria y eso ocasionaría una pérdida de tiempo, lo que podría ser valioso para la vida de la mascota.

Algunas personas cuando tienen una mascota que necesita la atención de un médico veterinario optan por usar redes sociales para informar la situación. Sin embargo estos medios no siempre son efectivo, porque no todos los médicos veterinarios están dispuestos a ofrecer su número de teléfono para este tipo de servicios, ya que se puede prestar para malos usos o simplemente puede que el médico veterinario no preste atención a estas herramientas, por lo que la comunicación de la situación entre el dueño de la mascota con el veterinario sea ignorada.

Muchas veces en las clínicas veterinarias llegan casos de urgencia sin previo aviso, esto resulta un problema para los médicos veterinarios, porque al no contar con información previa de la situación o con un aviso anticipado del dueño de la mascota, se hace más difícil actuar en el momento. Atender una urgencia requiere preparar una sala para atención de la mascota, realizar exámenes, preparar insumos necesarios en caso de una posible cirugía, etc, lo cual lleva tiempo que el veterinario se podría ahorrar si conociera de forma previa cierta información de la urgencia.

1.2. Definición del problema

El problema se genera cuando las mascotas presentan una urgencia y los dueños de éstas no saben que clínica veterinaria los puede atender, esto ocasionaría una pérdida de tiempo para el dueño de la mascota, ya que ese tiempo se podría ocupar para dirigirse concretamente a una clínica veterinaria que se encuentre en condiciones de realizar la atención de la urgencia. La principal motivación para realizar este proyecto es presentar una real alternativa de búsqueda para la atención veterinaria que entregue una mayor tranquilidad a los dueños de mascotas frente a una situación angustiante. Para lograr una atención veterinaria más efectiva, es necesario que los médicos veterinarios cuenten con información de la situación que presenta la mascota, y de esta manera estén más preparados al momento de atender la urgencia.

1.3. Propuesta de solución

Desarrolla una aplicación móvil para sistemas operativos Android que involucre dos áreas de trabajo: Un área orientada a los dueños de mascotas y otra área orientada para los médicos veterinarios. A continuación se detalla cada parte de la solución propuesta.

- La primera parte consiste en que los usuarios dueños de mascotas podrán enviar un aviso, en forma de notificación, con información detallada de la urgencia de la mascota, la que llegará a los médicos veterinarios con el fin de que conozcan y analicen la situación de la mascota.
- La otra parte de la aplicación es para los médicos veterinarios que podrán recibir el aviso, en forma de notificación, para luego ver el detalle de ésta y con la información recibida determinar si la clínica esta en condiciones para atender la urgencia de la mascota.

1.4. Objetivos

Objetivo general

- Reducir el tiempo de búsqueda de una clínica veterinaria apta para la atención de urgencia de una mascota, a través de una aplicación móvil.

Objetivos específicos

- Construir un entorno que permita entregar fácilmente información de la mascota a la clínica veterinaria cuando ocurra una urgencia.
- Establecer una metodología que permita la interacción entre veterinarios y dueños de mascota a partir de una aplicación móvil.
- Construir un entorno que permita visualizar la ubicación de las clínicas veterinarias.

1.5. Alcances

- La aplicación móvil solo se limita a trabajar con urgencias de mascotas.
- La aplicación móvil solo considera mascotas de tipo perro o gato.
- La aplicación móvil busca clínicas veterinarias en la misma ciudad donde el dueño de la mascota generó la petición de atención de urgencia.
- La aplicación móvil necesita conexión a internet para funcionar.
- La aplicación móvil es gratuita.
- La aplicación móvil está en idioma español.
- La aplicación móvil se desarrolla para dispositivos Android.
- La aplicación móvil no gestiona la atención veterinaria a domicilio.
- La aplicación móvil no considera el medio de pago de la atención veterinaria.

1.6. Resumen del capítulo

En este capítulo se hace un análisis del contexto del problema, es decir, responder a la interrogante que origina el problema a resolver, describiendo las acciones que realiza un dueño de una mascota cuando se le presenta una urgencia. Además el capítulo aborda la definición del problema, indicando cual es la problemática a resolver y la importancia que tiene esto para las personas que tienen mascota y para los médicos veterinarios.

En definitiva este capítulo presenta la bases necesarias para guiar todo el trabajo que hay realizar para llegar la solución propuesta, en donde se mencionan los objetivos que se pretenden lograr con la realización del proyecto junto con los respectivos alcances de este, que sirven para limitar la acciones que debe realizar la solución.

2. Marco Teórico

En el presente capítulo se abordan temas relacionados al comportamiento de los chilenos frente a las mascotas en los últimos años, también se da a conocer la cantidad de urgencias que atienden las clínicas veterinarias, además de conocer los tipos de urgencias que sufren a menudo gatos y perros, por último se mencionan las herramientas tecnológicas y metodologías de trabajo utilizadas para desarrollar la solución propuesta.

2.1. Conceptos básicos

- **Urgencia veterinaria** son aquellos accidentes o enfermedades de las mascotas, las que pueden ocurrir cuando menos se espera y generalmente necesitan de la atención de un médico veterinario [2].

Este es el principal motivo por el cuál se decide construir una aplicación móvil para ayudar a los dueños de las mascota a encontrar un veterinario que pueda atender la urgencia.

- **Alerta** es un tipo de comunicación máquina - persona que es importante y es sensible al tiempo. Una alerta puede ser un recordatorio del calendario o una notificación de un nuevo mensaje [3].

Este tipo de comunicación puede alertar a los médicos veterinarios de que una mascota necesita una pronta atención.

2.2. La sociedad y las mascotas

En los hogares de Chile hay una importante cantidad de mascotas, así lo refleja un estudio realizado por Adimark en el año 2018, dicho estudio muestra que un 64 % de los hogares chilenos existe por lo menos una mascota [1, página 9] . Además mencionar que el promedio de los últimos 5 años se ha mantenido con leves variaciones tal y como se muestra en la Figura 2.1.



Figura 2.1: Porcentaje de hogares chilenos que tiene mascota entre los años 2014 a 2018. Fuente: Microestudio GFK: los chilenos y sus mascotas 2018

2.3. Mascotas preferida por los chilenos

El estudio realizado por Adimark, en 2018, arroja que los perros son el tipo de mascota más preferido por los chilenos, ya que un 52 % de los hogares en Chile tiene uno, mientras que un 25 % de los hogares tiene un gato como mascota [1, página 10].

Para desarrollar la aplicación móvil hay que tener en cuenta el tipo de mascota que los médicos veterinarios pueden atender con mayor frecuencia, ya que al contar con esta información podrán evaluar de mejor forma la situación. Además, conociendo que tipo de mascota tienen los chilenos en sus hogares, la aplicación móvil puede

proporcionar información de la urgencia a la clínica veterinaria según sea el tipo de mascota que presenta esta situación.



Figura 2.2: Tipo de mascota que hay en los hogares de Chile. Fuente: Microestudio GfK: los chilenos y sus mascotas 2018

En la Figura 2.2 muestra parte de los resultados obtenidos en la encuesta, indicando que un 36% de los hogares no tiene ninguna mascota y sólo un 5% de los hogares tiene algún otro tipo de mascota. Entonces como la mayoría de los hogares tiene perro o gato, la aplicación solo se enfoca en las mascotas mencionadas anteriormente. Cabe destacar que si bien la aplicación a construir pondrá su foco solamente en perros y gatos, la información que se debe agregar para la urgencia es genérica para cualquier mascota y solamente puede variar en la información básica de la mascota, como por ejemplo clasificación de la edad, del tamaño, etc.

2.4. Preocupación de los chilenos hacia las mascotas

Es de conocimiento público el lamentable maltrato que han sufrido algunos animales por parte de sus dueños o por personas ajenas, tanto así que ha llevado a las autoridades promulgar una ley que sanciona estos tipos de comportamientos. Este es el caso de la “Ley Cholito” (Ley n° 21.020) que debe su nombre coloquial en memoria a una mascota víctima de maltratos. Esta ley llamada “Ley de Tenencia Responsable de Mascotas y Animales de Compañía” sanciona:

“Toda acción u omisión (ocasional o reiterada), que injustificadamente causara daño, dolor o sufrimiento a un animal.”

Luego de impulsada la ley n° 21.020, se ha visto un incremento en la preocupación por la salud y bienestar de la mascotas, aumentando la preferencia de llevar a la mascota a un recinto clínico veterinario.

De los hogares que poseen mascotas, un 81 % está interesado en la protección de animales. Además aquellos hogares donde hay una mascota, un 70 % lleva a estas al veterinario [1, página 12]. Las cifras obtenidas por el estudio Adimark 2018 demuestran que los chilenos se muestran cada vez más preocupados por la tenencia responsable de animales.

2.5. Urgencias veterinarias

son aquellos accidentes o enfermedades de las mascotas, las que pueden ocurrir cuando menos se espera y generalmente necesitan de la atención de un médico veterinario [2].

Las clínicas veterinarias semanalmente reciben una cantidad importante de atenciones a mascotas, mayoritariamente perros y gatos, en donde un cierto porcentaje del total de esas atenciones corresponde a urgencias. Según una encuesta realizada a 10 clínicas veterinarias de la ciudad de Curicó, se observa que en promedio un 16.25 % de aproximadamente 70 atenciones semanales son consideradas urgencias.

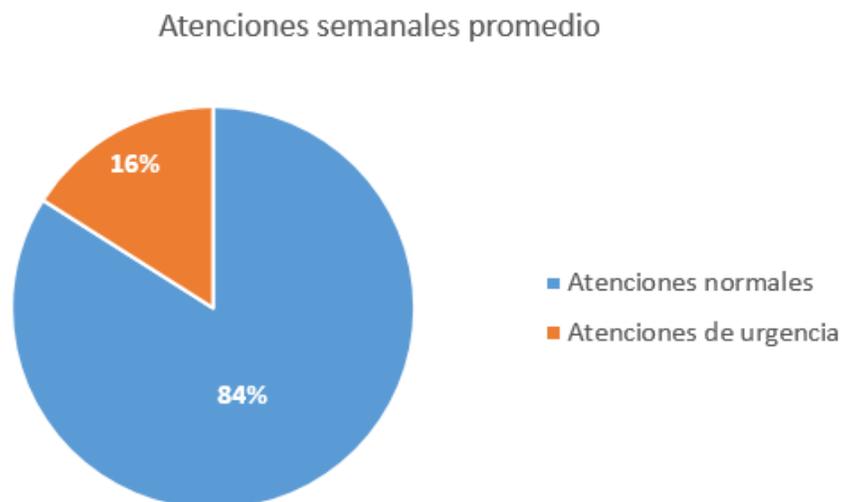


Figura 2.3: Gráfico que muestra el porcentaje promedio de atenciones normales y de urgencia en 10 clínicas veterinarias de la ciudad de Curicó.

La Figura 2.3 muestra, que de las 10 clínicas veterinarias encuestadas, el porcentaje promedio de atenciones de urgencias semanales es de 16 %. Si esto se lleva a un nivel regional, el porcentaje promedio podría ser similar o mayor. Cabe destacar que si bien la aplicación móvil sólo se enfoca en las atenciones de urgencia, en un futuro se puede trabajar con el porcentaje de atenciones restantes, lo cual resultaría que la aplicación se masifique.

2.5.1. Proceso de atención de urgencias

Preparativos parte del dueño de mascota

Cuando las personas tienen una mascota que presenta una situación en donde requiera la atención de un médico veterinario lo antes posible, los dueños tienden a mostrar dos tipos de comportamientos: dirigirse a una clínica veterinaria sin hacer un previo aviso de la urgencia o llamar primero a la clínica y informar de la situación y estado de la mascota.

Si su mascota sufriera una urgencia. ¿Qué haría usted?

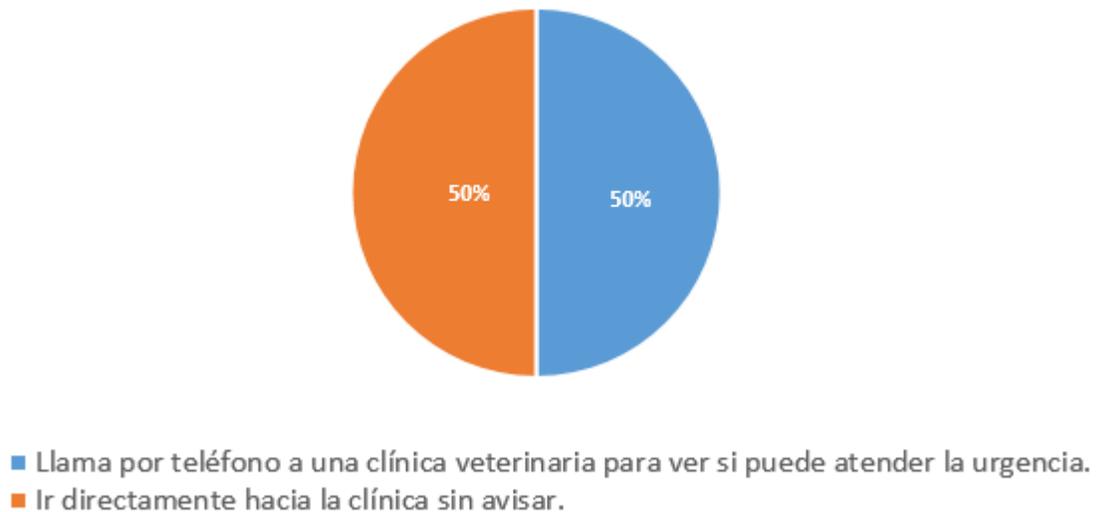


Figura 2.4: Gráfico que muestra el comportamiento de los dueños cuando su mascota presenta una urgencia.

La Figura 2.4 muestra los resultados de una encuesta realizada a 14 personas, en la cual se pregunta que harían ellos si tuvieran una mascota que presenta una

urgencia. La mitad de los encuestados respondieron que se dirigen hacia una clínica veterinaria sin hacer un previo aviso de la situación, mientras que la mitad restante llama primeramente a la clínica antes de ir a ella.

Además del proceso de aviso o no de la urgencia, también los dueños muestran otro comportamiento al momento de llegar a la clínica veterinaria, en donde toman dos decisiones según la disponibilidad de ésta.

Si usted llevara a su mascota, que presenta una urgencia, a una clínica veterinaria y ésta lamentablemente se encuentra ocupada, lo que no permite que su mascota sea atendida inmediatamente, ¿Que haría usted?.



Figura 2.5: Gráfico que muestra el comportamiento de los dueños cuando llegan a la clínica veterinaria y hay demora en la atención.

La Figura 2.5 muestra una encuesta realizada a las mismas 14 personas del estudio anterior, en donde cerca de dos tercios de los encuestados buscaría otra clínica si ésta no pudiera atenderlos, mientras que el tercio restante esperaría en la sala para ser atendido por un veterinario.

Proceso de atención del médico veterinario

El proceso de atención de la urgencia comienza cuando el veterinario atiende al dueño de la mascota, generalmente esta comunicación se lleva a cabo mediante

contacto telefónico. La información básica a obtener de la mascota se centra en la siguientes preguntas.

- ¿Que es lo que ha ocurrido?
- ¿Cuándo ha pasado?
- ¿Que se ha realizado hasta el momento?
- ¿Que problema presenta el paciente?

Cualquier duda respecto a la estabilidad del paciente debe ser considerada de atención prioritaria por el personal veterinario y deberá ser conducido a la clínica inmediatamente.

En la evaluación telefónica deberán ser evaluados los cuatros principales sistemas orgánicos (respiratorio, cardiovascular, sistema nervios central y sistema urogenital). Algunos síntomas frecuentes que pueden afectar algunos de estos sistemas y que requieren atención urgente son los siguientes:

- Dificultad respiratoria.
- Traumatismo.
- Vomito o diarrea severa.
- Sangrado visible.
- Palidez de mucosas .
- Debilidad extrema o incapacidad motora.
- Incapacidad de orinar.
- Ingestión de toxinas o cuerpos extraños.
- Dolor severo.

Los dueños pueden parecer muy angustiados y confundidos al teléfono, incluso considerando la ausencia de las circunstancias descritas, cualquier dueño preocupado por la situación clínica de su mascota debe ser tenido en consideración y se le debe recomendar una visita a la clínica veterinaria lo antes posible.

Una vez que la mascota ha llegado a la clínica veterinaria, debe ser evaluado por una persona del equipo médico. Una exploración sistemática y completa es absolutamente necesaria para no obviar ningún problema significativo en el animal. EL protocolo de rutina debe incluir lo siguiente.

1. Triage presencial (Termino referido a clasificar a los paciente en términos de la severidad de su cuadro clínico de forma presencial).
2. Evaluación primaria.
3. Evaluación secundaria.
4. Tratamiento definitivo.

La clasificación de los pacientes según el nivel de urgencia son los siguientes:

- Paciente catastrófico o en riesgo vital. Es aquél que debe recibir tratamiento inmediato, en segundos. Por ejemplo: los que presentan fallo respiratorio, traumatismo torácico, para cardiorrespiratoria u obstrucción de vías aéreas.
- Paciente critico o muy grave. Es aquél cuya atención debe llevarse a lo cabo de unos pocos minutos. Por ejemplo: pacientes con *shock*, sangrado, pero que presentan una función pulmonar adecuada.
- Paciente grave. Aquél cuya atención debe realizar en pocas horas. Se incluyen pacientes con fracturas abiertas, heridas profundas o quemaduras, pero que no estén en *shock* ni presentan alteración en el estado mental.
- Paciente menos grave. Aquél que debe realizarse la atención en las primeras 24 horas. Por ejemplo: cojeras, anorexia, vómitos, etc.

Luego en la atención primaria se debe evaluar, en no mas e 3 a 5 minutos, el ABCD del paciente, es decir, los síntomas que determinan si puede fallecer o no en pocos minutos [4].

- A (*Airway*): La vía aérea.
- B (*Breathing*): La respiración.

- C (*Circulation*): El sistema cardiovascular.
- D (*Disability*): La presencia de déficit en el sistema nervioso central.

La evaluación secundaria se inicia tras el tratamiento inicial y la estabilización del paciente. Ésta incluye de un examen físico más completo, la obtención de un historial clínico mas completo y la realización de pruebas diagnosticas. Después de la evaluación secundaria, el veterinario debe ser capaz de tomar decisiones en lo que respecta al plan terapéutico, pronostico y opciones de manejo del paciente en cuestión. Con toda la información recogida durante la fases previas, y tras informar al propietario sobre el estado de la mascota, se adoptará el tratamiento definitivo.

2.5.2. Tipo de urgencias frecuentes

Los médicos veterinarios reciben un variado tipo de urgencias, los cuales depende si se trata de un gato o perro. En el caso de los gatos las urgencias más recurrentes son las siguientes:

- **Peleas con otros animales.** Los gatos por naturaleza son territoriales. Muchas veces estos pelean con otros gatos que son invasores de su territorio, provocándose entre ellos heridas que algunas veces son graves. En otras ocasiones los gatos son atacados por perros, cuyas mordidas resultan de carácter grave para ellos.
- **Golpes por caídas.** Los gatos son ágiles, muchas veces este tipo de mascota merodean por los techos de las casas y los techos tiene una altura considerable, entonces cualquier caída de estos animales al suelo, les puede provocar fracturas y roturas de huesos.
- **Intoxicación.** Aunque en menor cantidad, los gatos igual son afectados por intoxicaciones, ya sea por comer algún alimento o beber agua con algún tipo de químico, como herbicidas, también se da el caso de intoxicación por comer productos que resultan dañinos para el gato.

En el caso de los perros alguno de los tipo de urgencia que se dan comúnmente son:

- **Mordidas con otros perros.** Los perros son territoriales y al defender su territorio pelean con otros, quedando mal heridos, lo que requiere una atención inmediata por parte de un médico veterinario.
- **Intoxicación.** Los perros son más propensos a intoxicarse debido a su naturaleza, ya que ellos comen, lamen y olfatean cualquier cosa que encuentren en el hogar o en la calle, quedando más expuestos a esa amenaza.
- **Atropellos.** Los perros son altamente propensos a atropellos por parte de vehículos, ya que por la fuerza del impacto los dejan malheridos, con fracturas y daños que podrían comprometer su movilidad e incluso su vida.
- **Golpes de calor.** En verano los perros son propensos a sufrir este tipo de situaciones por una exposición prolongada al sol.

2.5.3. Información útil en las urgencias

Cuando se presenta este tipo de situaciones es conveniente que los médicos veterinarios sepan datos básicos de la mascota y síntomas que esta presenta, así ellos están mejor preparados para atender la urgencia. Algunos datos básicos que hay que conocer en caso de perros y gatos son los siguientes:

- Tipo de mascota (perro o gato)
- Género (macho o hembra)
- Si esta castrado o esterilizada en caso que sea macho o hembra respectivamente
- Edad, para los médicos veterinarios es útil saber esta información, porque según la edad de la mascota, el tratamiento puede variar.
- En caso de los perros es útil saber si este es de tamaño: pequeño, mediano o grande, porque así los veterinarios pueden preparar de mejor forma el espacio adecuado de atención conociendo el tamaño del perro. Además es útil saber esta información, porque los veterinarios pueden preparar dosis de medicamentos o anestésicos sujeta al peso de la mascota.

La información anteriormente mencionada, junto con describir la situación actual de la mascota, conocer si el dueño ha suministrado algún tipo de medicamento, los

síntomas que presenta la mascota y cuanto tiempo ha pasado desde que manifiesta éstos, pueden ser de suma importancia para que el médico veterinario realice los protocolos necesarios para atender de mejor manera la urgencia.

2.6. Trabajo relacionados

A modo de ver los enfoques existentes para resolver el problema, se elabora una lista con aplicaciones con su descripción, ventajas y desventajas de cada enfoque.

2.6.1. VetMapp

Se escogió VetMapp como trabajo relacionado porque es una aplicación que está enfocada para que el usuario (dueño de una mascota) pueda buscar a través de un mapa (como Google Maps) clínicas veterinarias, facilitando la consulta y permitiendo realizar preguntas a veterinarios.

A continuación se nombran algunas ventajas de la aplicación VetMapp.

- Muestra las clínicas veterinarias en un mapa con un marcador diferente. Cada marcador representa el estado o servicios que una clínica veterinaria presenta.
- Presenta un servicio de ruta. Muestra la ruta desde la ubicación del usuario hasta la clínica veterinaria.
- Permite crear cuenta de dueños de mascotas y veterinarios.
- Le da la opción al usuario crear perfiles de la mascota.
- Permite la comunicación entre veterinario y dueño de la mascota a través de correo electrónico.

Ahora se nombran algunas desventajas de la aplicación VetMapp.

- No contempla el manejo de urgencias en la aplicación.
- La aplicación original esta pensando para Turquía.
- La aplicación no tiene muy buen funcionamiento cuando se cambia de idioma.
- No muestra clínicas veterinarias en Chile.

Toda la información sobre VetMapp fue obtenida de la página oficial de la aplicación [5].

2.6.2. Google Maps

Google Maps, si bien no es una aplicación enfocada al área de las urgencias veterinarias, cuenta con varias funcionalidades que son interesantes a la hora de buscar una clínica veterinaria para atender a una mascota.

A continuación se nombran algunas ventajas de la plataforma Google Maps.

- Permite buscar clínicas veterinarias en un sector dado, mostrando la ubicación de la clínica en el mapa.
- Muestra información de calificación, horarios, comentarios.
- Según la ubicación del usuario puede trazar una ruta hacia la clínica veterinaria.

Ahora se nombran algunas desventajas de la plataforma Google Maps.

- Algunas veces la información de los sitios que aparecen en Google Maps es incorrecta.
- La forma más sencilla para comunicarse con la clínica es a través del teléfono, lo cual es más difícil detallar información de una urgencia y no siempre es seguro que alguien conteste.

2.6.3. WhatsApp

WhatsApp es una aplicación de mensajería instantánea y mensajería multimedia para smartphones que es ampliamente utilizado por la gente. Actualmente muchas empresas usan esta aplicación para comunicarse con los clientes, ya sea para realizar consultas y/o agendar horas de atención. Algunos ejemplos de negocios que usan WhatsApp son: clínicas dentales, locales de comida, peluquerías, clínicas veterinarias, etc. A continuación se presenta algunas ventajas de esta aplicación.

- Envío de mensajería de texto y multimedia (vídeos e imágenes) de manera ilimitada.
- Permite compartir ubicación a través de Google Maps.

- Notificaciones de aviso cuando se hacen recepción de mensajes.

Ahora se presentan algunas desventajas de WhatsApp

- Dificultad de enviar información a varios médicos veterinarios.
- La información se recibe de forma más desordenada.
- Al ser una aplicación de mensajería de múltiple uso (no especifica para atención de urgencia veterinaria), algunas veces los usuarios de esta no están atentos a los mensajes, por lo que éstos son vistos fuera de tiempo.
- No permite mostrar información de clínicas veterinarias.

2.7. Tecnologías a utilizar

En esta sección se aborda la definición de las tecnologías a usar para implementar el proyecto, además de nombrar la distintas alternativas que existen para la herramienta usada y la justificación de la elección de ésta.

2.7.1. Patrón de Diseño Modelo-Vista-Controlador (MVC)

El patrón de diseño modelo vista controlador (MVC) tiene como principal objetivo separar la lógica de negocio de la lógica de presentación. La capa modelo representa donde los datos son enviados desde la base de datos para luego ser mostrados en una pantalla. La capa de vista representa los componentes visuales que el usuario ve en una pantalla de un smartphone o computador como por ejemplo botones, campos de textos, etc. Mientras que la capa controlador maneja los eventos gatillados por el usuario y los comunica con la capa modelo [6].

La Figura 2.6 se muestra el flujo del patrón de diseño Modelo-Vista-Controlador. En el contexto de desarrollo del proyecto, en ambiente Android, el modelo se encarga de recibir los datos provenientes de la base de datos. La vista es representado por los archivos .xml que el IDE ¹ genera. Es en esos archivos donde se aloja todo lo que el usuario visualiza al ejecutar la aplicación. Los controladores son archivos .java

¹Integrated Development Environment, en español entorno de desarrollo integrado. Es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador el desarrollo de software.

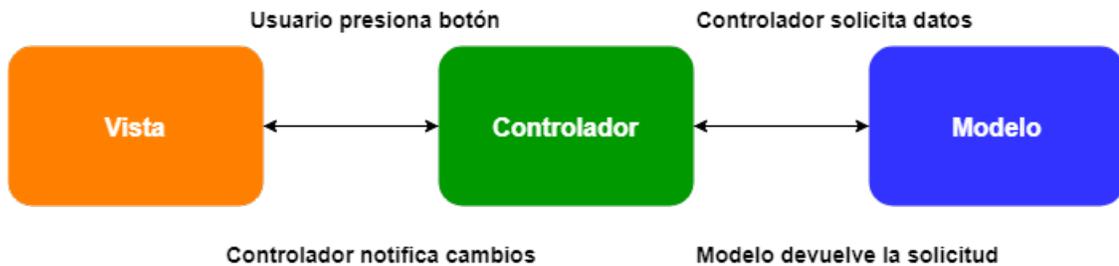


Figura 2.6: Patrón de diseño Modelo-Vista-Controlador. Fuente: Elaboración propia.

generados cuando se crea un componente de la capa vista, dichos archivos manejan los eventos generados por el usuario cuando interactúa con la vista con el fin comunicar los datos recibidos por la capa modelo y mostrarlo en la capa vista.

2.7.2. Android

Descripción

Android es un sistema operativo y una plataforma de software, para teléfonos móviles, basado en Linux. Además, también usan este sistema operativo (aunque no es muy habitual), tablets, netbooks, reproductores de música e incluso PC's. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual Dalvik ². Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, *widgets*, o incluso, modificar el propio sistema operativo, dado que Android es de código libre [7].

La tecnología Android cuenta con un SDK ³ que permite la creación de aplicación móviles con acceso a las siguientes características del dispositivo [8].

- Localización basada en servicio de GPS.
- Control multimedia para grabar con la cámara y el micrófono.
- Librerías para grabar audio y vídeo
- Integración con mapas para localización.

²Una variación de la máquina de Java con compilación en tiempo de ejecución.

³Abreviación en inglés de Software Development Kit

Existen muchas otras características que ofrece el SDK de Android, pero las nombradas anteriormente son las fundamentales para el correcto funcionamiento de la aplicación que se desea desarrollar.

Alternativas

No solamente se puede desarrollar una aplicación móvil para el sistema operativo Android de Google, existe otra alternativa el cual consiste en desarrollar para el sistema operativo iOS de la marca Apple. iOS, al igual que Android, cuenta con un SDK que permite a terceros desarrollar aplicaciones móviles nativas con acceso a distintos componentes del dispositivo móvil, como por ejemplo: la cámara, GPS, archivos, giroscopio, etc.

Justificación elección

Android es multiplataforma, puede ser desarrollado usando cualquier sistema operativo Windows (mayor a XP), Mac OS X (10.5.8 o mayor) y Linux (con kernel 2.6 o superior) [9]. En cambio para desarrollar en iOS se necesita computadores Macintosh que tengan sistema operativo Mac OS X 10.6.

Según StatCounter, sitio web que se encarga de recoger estadísticas en tiempo real, el Marketshare ⁴ de sistema operativos en Chile es de 83,08% para Android y un 16,35% para iOS entre el mes de enero a agosto de 2019 [10].

En la figura 2.7 es evidente que el sistema operativo que predomina en smartphones en Chile es Android.

⁴También conocido como participación del mercado, es el resultado de calcular las ventas de un producto en comparación con las ventas totales del mercado.

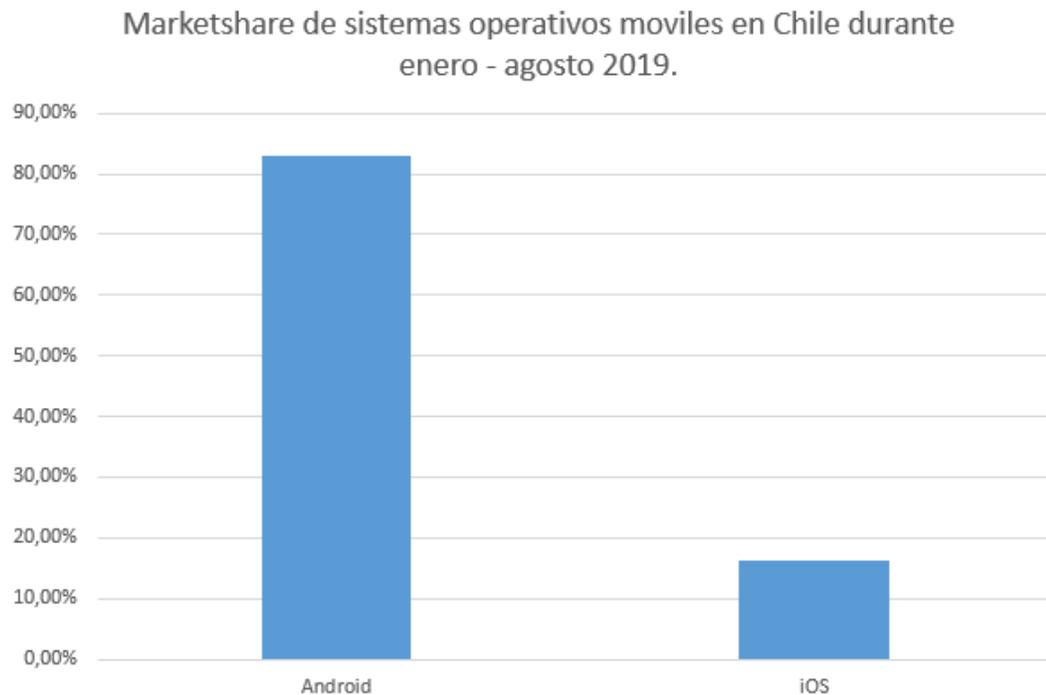


Figura 2.7: MarketShare del sistema operativo Android en Chile entre enero y agosto de 2019. Fuente: Datos obtenidos del sitio StatCounter.

2.7.3. Android Studio

Descripción

Android Studio es un IDE que fue desarrollado por Google y que permite al desarrollador usar una herramienta que facilita el proceso de creación de una aplicación en Android. Android Studio es el IDE oficial para desarrollar en esta plataforma [11].

Android Studio provee la siguientes herramientas que ayudan a agilizar el proceso de desarrollo:

- Tecnología IntelliJ, lo que permite auto-completar código.
- Integración con sistema de control de versiones.
- Variados menús y opciones para la creación de diferentes tipos de archivos (.java, .xml, etc), creación de componentes Android (Activities, Fragments, Layout files, etc).

Alternativas

Aparte del IDE Android Studio, también existe la posibilidad de desarrollar con Eclipse. Este IDE cuenta con varias herramientas como por ejemplo [12]:

- Proveedor de Git
- Proveedor de herramientas de Java
- Recomendaciones de código por desarrolladores Java
- Editores y herramientas de XML.

Por otra parte si se quiere desarrollar para iOS una elección es el IDE Xcode. Xcode provee un variado abanico de herramientas [13]:

- Editor de código
- Documentación completa
- Depurador gráfico
- Integración continua

Justificación elección

Como se mencionó en la descripción de la herramienta Android Studio, este es el IDE oficial para desarrollar en Android. Al ser el IDE oficial y de la empresa Google tiene más aceptación y cuenta con más referencia en caso de problemas con la herramienta. Además, cuenta con actualizaciones continuas, administrador de máquinas virtuales, estadísticas de compatibilidad con distintas versiones del sistema operativo Android.

2.7.4. Java

Descripción

Java es un lenguaje de programación orientado a objetos, desarrollado por James Gosling y su equipo en Sun Microsystems en California [14].

Este lenguaje es la base para la construcción de una aplicación en la plataforma Android, ya que para desarrollar una aplicación Android se tiene que aplicar conceptos de extensión de clases, implementaciones de interfaces, sobrescritura de métodos, entre otros conceptos propios del lenguaje Java y del paradigma orientado a objetos.

Alternativas

En 2012 apareció un lenguaje de programación llamado Kotlin. Este es un moderno lenguaje compatible con Android que arregla muchos errores de Java como *null pointer exceptions* o código excesivo. Kotlin es un lenguaje inspirado por Swift, Scala, Groovy, C# y mucho otros lenguajes [15].

Justificación elección

A pesar de que Kotlin corrige errores y problemas que tiene Java, se escogió este último como lenguaje de programación porque es más antiguo que Kotlin. El código fuente de Kotlin fue liberado en 2012, mientras que Java apareció en el año 1996, por ende la cantidad de referencia de Java es muchísima mayor que de Kotlin. Además es muy importante mencionar que se eligió Java por sobre los otros lenguajes, porque éste tuvo presente más que ningún otro lenguaje en el proceso de formación universitaria, por ende el dominio técnico será mucho mayor que de cualquier otro lenguaje de programación. Otro criterio para escoger Java fue por la gran cantidad de documentación que existe en la web, por lo tanto desarrollar con Java es rápido y fácil en caso de encontrarse con errores y/o no saber como realizar una cierta tarea.

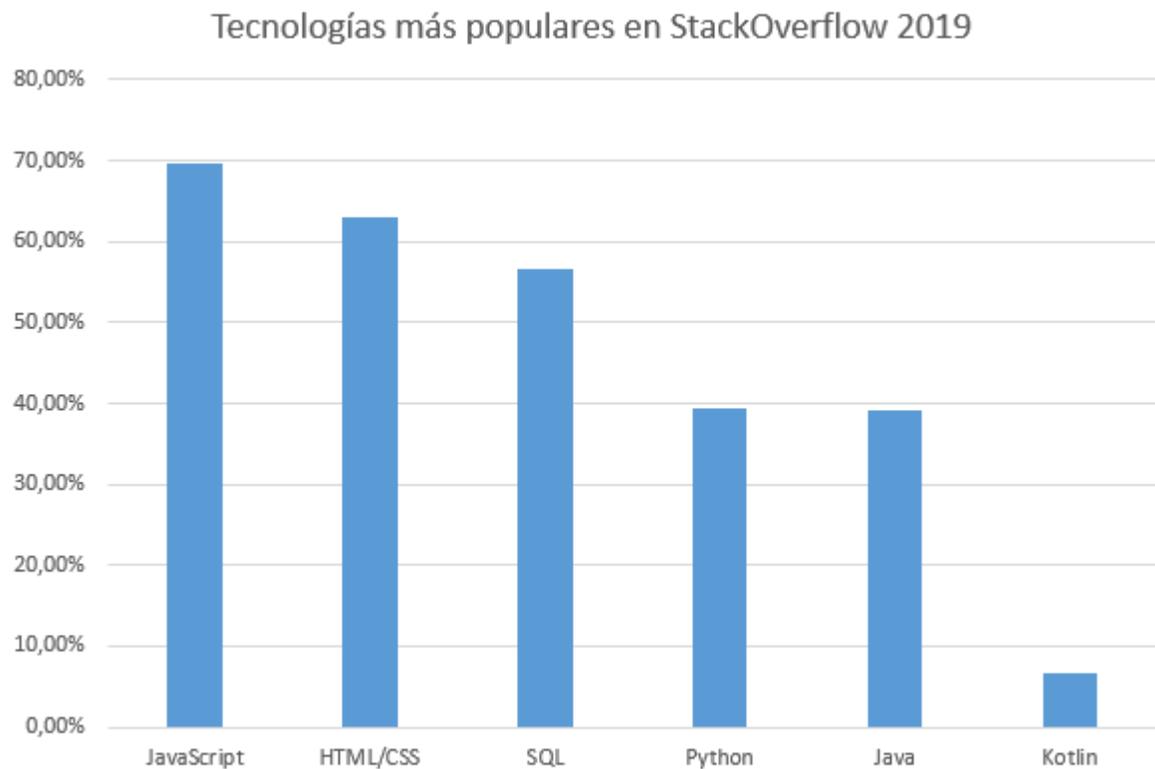


Figura 2.8: Lenguajes de programación, Scripting y de Marcado más ocupados en 2019. Fuente: Datos obtenidos de StackOverflow. Fecha: consultado el 21-09-2019.

La Figura 2.8 muestra que un casi un 40 % de los encuestados utiliza el lenguaje Java, mientras que menos del 10 % usa Kotlin.

2.7.5. XML

Descripción

Extensible Markup Language es un lenguaje parecido a HTML que fue diseñado para almacenar, transportar información y para ser auto descriptivo. La diferencia entre HTML y XML es que el primero fue diseñado para mostrar información mientras que el segundo fue diseñado para llevar información [16].

Este lenguaje de etiquetas es fundamental a la hora de construir una aplicación móvil en Android, ya que la parte visual usa XML. Las vistas tienen componentes visuales como lo son botones, textos, imágenes, entre otros. Además, estos componentes tienen características que son representados en etiquetas del lenguaje XML.

2.7.6. Firebase

Descripción

Firebase real-time database es una base de datos de Google hosteada en la nube. Los datos son almacenados como objetos JSON (JavaScript Object Notation) y son sincronizados en tiempo real por cada cliente conectado (web, iOS, Android, entre otros). Esta base de datos puede ser accedida directamente desde clientes y no hay necesidad de un servidor intermediario. Los datos además pueden ser persistidos localmente si el cliente de la aplicación esta fuera de servicio (sin conexión a internet), lo cual permite eventos en tiempo real que son constantemente gatillados que reflejan el estado de actualización de la información al usuario. Cuando el dispositivo recobra la conexión, Firebase sincroniza los cambios de los datos locales con las modificaciones remotas que ocurren mientras el cliente estaba sin servicio, mezclando cualquier conflicto automáticamente [17].

Firebase además entrega otras herramientas útiles para el desarrollo de proyecto como por ejemplo:

- Servicio de autenticación de usuarios.
- Servicio de almacenamiento de archivos en la nube (imágenes, vídeos, etc).
- Servicios de notificaciones Firecloud messages (FCM).
- Hostings.

Firebase utiliza una base de datos NoSQL, la que se refiere a una ecléctica y cada vez mayor familia de grupos de sistemas de administración de datos no relacionales; donde las base de datos no son construidas principalmente en tablas y generalmente no se usa SQL para la manipulación de datos [18].

Alternativas

Para la capa de acceso de datos (backend) existen variadas alternativas como por ejemplo usar MySQL, PostgreSQL o SQLite. Las tres herramientas mencionadas anteriormente son sistema de gestión base de datos (SGBD) relacionales, es decir, los datos se almacenan en estructura de datos llamadas tablas y a través de relaciones entre dichas tablas se pueden hacer consultas mediante un lenguaje de consulta como lo es SQL (Structured Query Language).

Justificación elección

Firebase es una opción muy recomendable sobre todo para el desarrollo móvil. Firebase cuenta con herramientas como: autenticación de usuarios, base de datos en tiempo real, almacenamiento de archivos en la nube, notificaciones push, entre otras. Todas las funcionalidades nombradas anteriormente disminuyen la carga del desarrollo, ya que si se usara una base de datos como MySQL se tendrían que implementar módulos de autenticación efectivos, hacer conversiones de formato de los archivos a otro tipo de formato para que puedan ser almacenados en las base de datos, levantar un servidor para hacer notificaciones, etc. Entonces realizar esas tareas sería una carga importante que demandaría tiempo.

Además, Firebase, en su documentación, cuenta con un completo tutorial para su integración al SDK de Android. Además en la web hay referencias sobre esta tecnología, lo que lo hace perfecto en caso que se tenga que realizar una búsqueda por una funcionalidad o error que pueda surgir durante el proceso de desarrollo. A lo anterior también se suma que en Youtube hay una gran cantidad de vídeo tutoriales sobre Firebase.

2.7.7. Git

Descripción

Para el sistema de control de versiones se usa la herramienta Git. Esta es una herramienta de control de versiones desarrollada por Linus Torvalds, quien creó el Kernel de Linux. La ventaja de trabajar con un sistema control de versiones como Git, es que se puede volver atrás en el desarrollo, esto es útil cuando se tiene errores en el código y no se sabe con certeza la fuente del error. Además se puede trabajar con un flujo de trabajo llamado GitFlow, el cual es super popular entre los equipos de desarrollo, porque permite implementar funcionalidades de forma paralela en ramas que tienen solo un propósito. Lo anterior facilita en gran medida los procesos de codeo, *testing* e integración con otros módulos o implementación que se van desarrollando de forma paralela.

Alternativas

Existen otras herramientas de control de versiones como por ejemplo: CSV (Concurrent System Version), Mercurial , entre otros. Estos dos últimos son los más conocidos a parte de Git.

Justificación elección

Para el desarrollo del proyecto se utiliza la herramienta Git simplemente por el hecho de tener más experiencia con dicha herramienta, ya que en cursos anteriores de desarrollo de software se ha utilizado este sistema de control de versiones. También se escogió trabajar con Git, porque se puede usar una herramienta gráfica llamada GitKraken, la cual permite visualizar de mejor forma las distintas ramas con las que cuenta un proyecto. Además usar GitKraken junto con GitFlow, ayuda a agilizar el desarrollo, ya que GitKraken aporta el entorno gráfico que facilita la visualización de las ramas del proyecto y Gitflow aporta con una metodología que permite usar las ramas de Git de una forma más ordenada.

2.7.8. Google Maps

Descripción

Google Maps es un proveedor web de mapas que entrega información detallada acerca de regiones geográficas y sitios alrededor del mundo[19]. Además Google Maps ofrece imágenes de mapas desplazables, fotografías por satélite del mundo, calculador de ruta a diferentes ubicaciones del globo, ya sea tanto a pie, auto, bicicleta, transporte público con Google Street View, condiciones de tráfico en tiempo real gracias a Google Traffic, navegador GPS, incluso ofrece la posibilidad de visualizar en un mapa 3D imágenes de la Luna y el planeta Marte. El SDK de Google Maps cuenta con una API ⁵ que permite integrar los servicios en un entorno web o móvil. El SDK permite integrar el mapa en la aplicación, creación de marcadores, cálculo de ruta, cálculo de distancia entre coordenadas, visualizar la posición actual del usuario, dibujo de figuras en el mapa, como polígonos, cuadrados, etc.

⁵Application Programming Interface

Alternativas

Existen otras alternativas de servicios que proveen mapas como por ejemplo OpenStreetMap, BingMap (Microsoft) y MapBox,

Justificación elección

Google Maps es una muy buena alternativa, ya que como Android es de Google y este proveedor de mapas también es del mismo fabricante, entonces puede haber mayor compatibilidad al trabajar con las herramientas. Además Google Maps cuenta con guías en sus páginas de como integrar sus servicios con Android, lo cual facilita mucho el desarrollo. Google tiene una comunidad importante de personas, entonces a la hora de buscar una referencia el resultado es más rápido.

2.8. Metodología iterativa incremental

El desarrollo incremental es una variante del modelo de cascada, que consiste en una serie de ciclos de vida de cascada en donde el proyecto de desarrollo de software se divide en segmentos llamados incrementos [20].

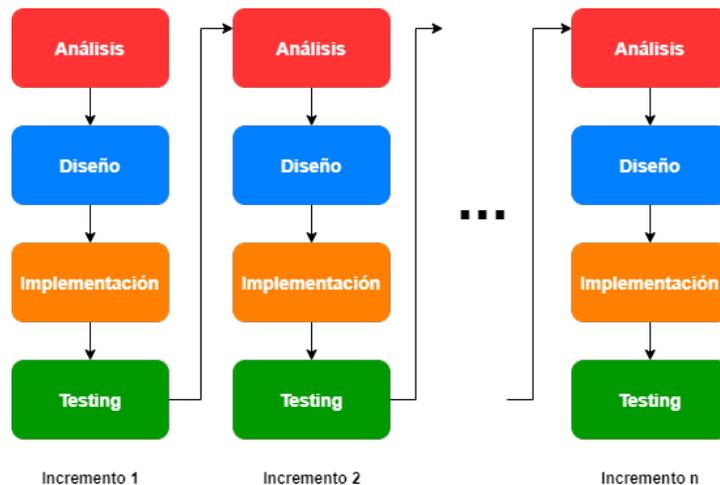


Figura 2.9: Ciclo desarrollo metodología incremental. Fuente: Elaboración propia.

Como se muestra en la Figura 2.9 cada incremento ejecuta las fases de desarrollo del modelo tradicional como lo es la fase de requerimientos, diseño, implementación, verificación y liberación de manera iterativa. Además cada incremento debe entregar

un artefacto de software funcional. La carga de trabajo se planifica, como se menciona anteriormente, en segmentos llamados iteraciones, cuya duración varía entre dos a cuatro semanas. A continuación se describen las fases a realizar en cada incremento.

1. *Análisis*. En esta fase se capturan los requisitos del proyecto, es decir, las funcionalidades que el software o sistema deberá realizar. Se pueden obtener los requisitos mediante reuniones con el cliente, observando los procesos de un negocio, mediante la realización de encuestas, entre otros métodos. Una vez capturado los requisitos, se separan en requisitos de usuarios y de software o de sistema. Los requisitos de usuario, son las acciones que debe realizar el sistema en lenguaje natural, mientras que los requisitos de software, son las acciones en lenguaje técnico, que ayudan a los desarrolladores a tener un mejor entendimiento de lo que debe realizar el sistema a construir.
2. *Diseño*. En esta fase se abstraen los requisitos capturados para obtener los modelos del sistema a construir. En esta fase se realizan diseños a bajo y alto nivel. Dentro de los diseños a bajo nivel se encuentran la realización de: diagramas de clases, diagramas entidad relación, diseño de algoritmos, entre otros mientras que en el diseño a alto nivel se realizan diagramas de arquitectura física y lógica además de diseño de interfaces gráficas. Estos modelos se realizan tomando en cuenta conceptos de modularidad, extensibilidad, rendimiento, escalabilidad, etc.
3. *Implementación*. En esta fase se representan todos los modelos realizados en la fase anterior, tanto de alto y bajo nivel, en código. En desarrollos iterativos, esta fase dura aproximadamente entre 2 a 4 semanas.
4. *Testing*. Esta fase se encarga de revisar de que el código no tenga fallos, es decir, que no hayan errores durante la ejecución del software así como también que las entradas y salidas sean las correctas, para ello se realizan pruebas de caja negra y caja blanca, además de pruebas unitarias, pruebas de integración y de sistema.

El modelo iterativo incremental presenta una serie de ventajas con respecto al modelo tradicional en cascada, los cuales se listan a continuación.

1. Reduce el riesgo del modelo en cascada agrupando los requisitos en incrementos para luego implementarlos de manera iterativa.
2. Es más flexible que el modelo cascada, ya que permite agregar nuevos requisitos que son posteriormente implementados en los siguientes incrementos.
3. Se tiene un software funcional en un periodo que va desde 2 a 4 semanas.
4. Como es una metodología que se centra en objetivos, entonces en las primeras iteraciones se entregan artefactos de software que aporten valor al cliente.
5. Los usuarios pueden utilizar los incrementos iniciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.
6. Los requisitos que aportan mayor valor, son los que más se prueban, entonces es muy probable que las funcionalidades que son importantes tengan un porcentaje de fallo pequeño.

2.9. Metodología de evaluación del proyecto

Los Estudios Empíricos son necesarios para mejorar los procesos, métodos y herramientas para el desarrollo y mantenimiento de software. Los estudios empíricos más ocupados en ingeniería de software son: experimentación, caso de estudio y encuestas [21].

- **Experimentación:** La característica más importante es que permiten gran nivel de control. El objetivo es variar una o mas variables y controlar otras, para apreciar que efecto tienen sobre una determinada variable de salida [22].
- **Caso de estudio:** Son utilizados para monitorear proyectos o actividades. Normalmente su propósito es seguir un determinado atributo o establecer relaciones entre determinados atributos. El nivel de control es mucho menor que en un experimento. Es un estudio de tipo observacional. Se realizan on-line (en entornos industriales)[22].
- **Encuesta:** Normalmente se hacen en retrospectiva, cuando por ejemplo, una técnica o herramienta se ha estado usando durante cierto tiempo. Tanto los

datos cualitativos como cuantitativos se recogen a través de cuestionarios o entrevistas. Permiten obtener conclusiones descriptivas y exploratorias[22].

El proceso de experimentación en ingeniería de software cuenta con cuatro fases como se muestra en la Figura 2.10.



Figura 2.10: Fases de la experimentación en ingeniería de software.

1. **Definición:** Consiste en especificar en términos generales los aspectos principales del experimento a utilizar indicando objeto en estudio, enfoque de calidad así como también el contexto [23].
2. **Diseño:** Esta fase consiste en como se llevará a cabo el experimento indicando los sujetos que participan y los instrumentos para la realizar o ejecutar el experimento [23].
3. **Ejecución:** La ejecución consiste en llevar a cabo el experimento. En esta fase se tiene listo los materiales a utilizar, los equipos, así como el espacio donde se llevará a cabo. Antes de comenzar el experimento, los participantes reciben una serie de instrucciones, así como los materiales con los que trabajarán durante la sesión del experimento [23].
4. **Análisis:** Tras ejecutar el experimento, se recolectan una serie de métricas generadas por los sujetos. En esta actividad del proceso, las métricas son analizadas y contrastadas con el objetivo previamente definido [23].

2.10. Resumen del capítulo

En este capítulo se muestran y definen aspectos que son relevantes para entender el contexto del proyecto. Es importante conocer la cantidad aproximada de mascotas que hay en los hogares en Chile. Según un estudio de Adimark, un 64 % de los hogares en Chile tiene una mascota, eso se puede traducir como en seis de cada diez hogares hay una mascota. Además es relevante conocer el tipo de mascota que más hay en los hogares, el mismo estudio mostró que los chilenos tienen más perros que gatos, mientras que un porcentaje menor, al rededor de 5 %, tiene otro tipo de mascota. También es importante saber que existen en Chile leyes que penan el maltrato animal mediante la ley 21.020 más comúnmente conocida como “Ley Cholito”, que multa a los dueños de mascota que infringen algún tipo de maltrato animal.

Es relevante conocer la cantidad de urgencias semanales que reciben las clínicas veterinarias, ya que con esto se sabe la frecuencia de este tipo de atenciones. Además es fundamental saber el tipo de urgencias que una clínica veterinaria atiende generalmente, porque con esta información se puede lograr que la aplicación sea útil, entregando a los veterinarios información importante.

Por último, se habla de las tecnologías que serán ocupadas para el desarrollo del proyecto, dando énfasis a la descripción de la tecnología, alternativas existente y una justificación para la elección de esta.

3. Metodología

La metodología es muy importante para la realización de cualquier proyecto, ya que ésta indica los pasos a seguir para la realización del proyecto. En el presente capítulo se da a conocer cómo se utiliza lo mencionado en el marco teórico en el desarrollo de la solución, además la forma de utilizar las metodologías de desarrollo de software junto con la de evaluación y de qué modo se planifica el proyecto.

3.1. Notificar urgencias a veterinarios

Como se ha mencionado en reiterada ocasiones es importante que los médicos veterinarios cuenten con información previa a atender la urgencia, como por ejemplo: si la mascota a atender es perro o gato, la edad, el tamaño (pequeño, mediano o grande en caso de ser un can), los síntomas que ésta presenta y cuánto tiempo ha transcurrido desde la aparición de éstos. Por esta razón, una forma de dar cuenta al médico veterinario que hay una urgencia es mediante el envío de notificaciones con la información que el dueño debe proporcionar.

Para enviar las notificaciones se usa Firebase Cloud Messaging (FCM), que es un servicio que Firebase provee. Esta permite el envío de notificaciones a dispositivos móviles mediante el uso de un *token*, el cual es un identificador que el mismo FCM genera para asociarlo a un dispositivo móvil que ejecuta la aplicación. Entonces para lograr el envío de notificaciones entre dueños de mascota y veterinarios, se tiene que almacenar el *token* de cada usuario, cuando éstos ingresen a la aplicación, y recuperarlos de la base de datos cuando un usuario quiera enviar una notificación.

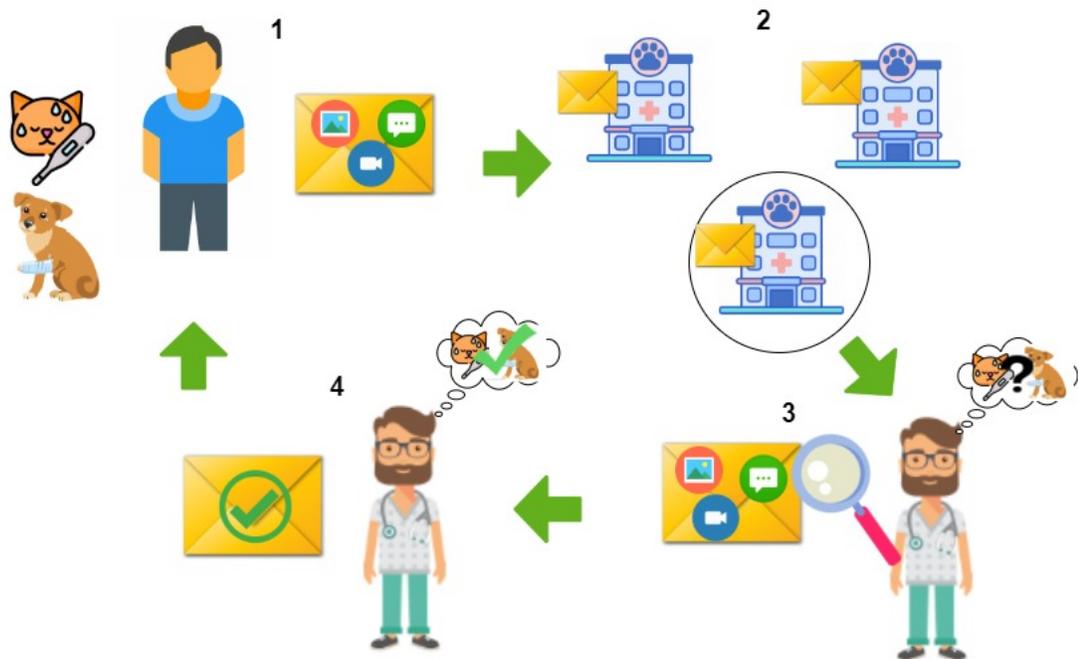


Figura 3.1: Representación del flujo de envío de notificaciones entre un dueño de mascota y veterinario.

En la Figura 3.1 se muestra el flujo separado en cuatro pasos de la notificación enviada por el dueño de la mascota cuando realiza una petición de urgencia para los veterinarios. Para entender de mejor manera el flujo mostrado, se explica el funcionamiento de cada paso.

1. En el primer paso del flujo, el dueño de la mascota proporciona información de la urgencia, como por ejemplo: síntomas que presenta ésta, además se puede adjuntar opcionalmente una imagen y/o vídeo que permita al veterinario entender mejor la situación o aclarar detalles. Finalmente el dueño envía el mensaje, en forma de notificación, hacia las clínicas veterinarias.
2. Luego en el segundo paso, el mensaje en forma de notificación llega a los médicos veterinarios que trabajan en las clínicas veterinarias y así ellos se percatan de la existencia de una urgencia.
3. Una vez que el veterinario abra la notificación, podrá visualizar el contenido de ésta y así analizar la urgencia de la mascota. Una vez que el veterinario termine

de leer el contenido de la notificación, decidirá si la clínica está en condiciones, o no, de atender la urgencia.

4. Si el veterinario acepta la petición, entonces al dueño de la mascota se le enviará una notificación indicando que el veterinario ha aceptado la petición y atenderá la urgencia. Con el conocimiento previo de la situación que presenta la mascota, el veterinario podrá preparar los elementos necesarios para recibir la urgencia mientras el dueño se dirige a la clínica veterinaria.

3.2. Aplicación metodología de desarrollo

El trabajo que se realiza se divide en ocho incrementos con una duración de dos semanas cada uno, ya que con esto se logra un prototipo funcional en menor tiempo y que en definitiva ayuda a no disminuir el ritmo de trabajo y la productividad del desarrollo. Los incrementos incorporan las fases de requisitos o análisis, diseño, implementación o desarrollo y pruebas. A continuación se explica como se trabaja en cada fase de la metodología aplicada al trabajo a realizar.

3.2.1. Fase de análisis

Para aplicar la metodología iterativa incremental primero se inicia con la fase de captura requisitos. Para ello se contacta con algunas personas que son dueños de mascotas y médicos veterinarios, luego se les comenta la idea de la solución propuesta con algunas funcionalidades que tendrá la aplicación móvil. Después los requisitos obtenidos son recopilados en un documento de especificación de requisitos en el cual se dividen en dos categorías:

- Requisitos de usuario
- Requisitos de sistema

Código - Nombre requisito.
Descripción:
Fuente:
Prioridad:
Estabilidad:
Fecha actualización:
Estado:
Incremento:
Tipo:
Usuario asociado:

Cuadro 3.1: Estructura de especificación de requisitos.

Para especificar los requisitos se usa una tabla que consta de diez campos como la que se muestra en el Cuadro 3.1. A continuación se describe el contenido de cada fila de la tabla [24].

- **Código - Nombre requisito:** Este es un código único que sirve para identificar o reconocer el requisito. Para los requisitos de usuarios se utiliza el formato RUXXX y para los de software RSXXX. Nombre en lenguaje normal del requisito.
- **Descripción:** Descripción del requisito. Qué aspectos involucra, en qué consiste, que herramientas, algoritmos, componentes gráficos a utilizar, etc. Este campo es de ayuda tanto para el usuario y el desarrollador.
- **Prioridad:** Prioridad asociada al requisito, esta puede ser crítica, deseable o innecesaria. Un requisito es crítico si afecta una operación crítica del negocio. Si existe alguna funcionalidad que se quiera incluir para mejorar los procesos actuales, entonces se habla de un requisito deseable. Finalmente si se trata de un requisito informativo o que puede esperar para etapas posteriores, el requisito es catalogado como innecesario.
- **Fuente:** Documento o persona desde la cual surgió el requisito.
- **Fecha actualización:** Fecha en que se actualizó el requisito.
- **Estado:** Estado actual del requisito dentro del desarrollo (Cumple, No Cumple, Ambiguo). Si un requisito se encuentra en un estado de “No cumple”, entonces

se pospone para el siguiente incremento. En caso que éste tenga una prioridad crítica, se opta por terminarlo de los primeros en el siguiente incremento.

- **Estabilidad:** Si el requisito está sujeto a cambios durante el ciclo de vida del software. El estándar ESA ¹ lo define como Estable o No Estable.
- **Usuario asociado:** Son los tipos de usuarios que están asociados al requisito.

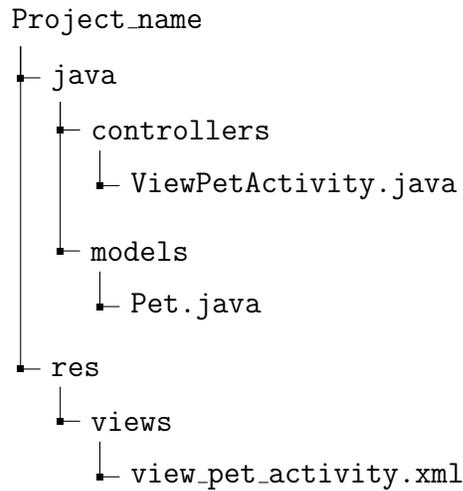
Una vez realizada la especificación de requisitos, con el formato especificado anteriormente, se seleccionan los requisitos que se implementan para cada incremento que involucra el desarrollo de la aplicación móvil.

3.2.2. Fase de diseño

Luego de obtener los requisitos, se realizan *mockups* de las interfaces gráficas de las principales pantallas que tendrá la aplicación móvil tanto para usuarios dueños de mascotas y veterinarios. También en esta fase se trabaja en la construcción y modificación de componentes de la arquitectura física y lógica de la aplicación móvil.

La arquitectura lógica del componente de los clientes, es decir, de los dispositivos móviles del usuario dueño de mascota y veterinario, se basa en un patrón de diseño Modelo Vista Controlador, el cual se utiliza para este proyecto. En Android Studio, por defecto se separan los archivos que componen las vistas de aquellos que pertenecen al controlador, es decir, cuando se crea un componente visual, por defecto el archivo se guarda en una carpeta llamada *res*, que hace referencia en inglés a *resources*. Mientras tanto los componentes del controlador, que manejan la lógica de la visual y la manipulación de las clases que interactúan con la lógica del desarrollo, se guardan en una carpeta llamada *java*. Los archivos que componen el modelo son guardados en una carpeta que el mismo desarrollador crea manualmente y que generalmente se nombra como *models*. A continuación se muestra la jerarquía que se utiliza para el patrón MVC en el IDE Android Studio.

¹ESA, European Spacial Agency, en español Agencia Espacial Europea



La jerarquía mostrada anteriormente se puede representar de manera más gráfica en la Figura 3.2.

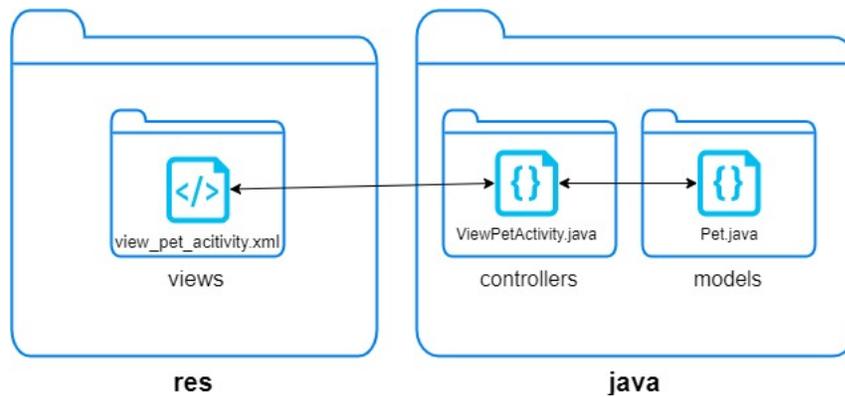


Figura 3.2: Patrón Modelo-Vista-Controlador utilizando la configuración de carpetas que Android Studio crea defecto.

Como el lenguaje Java es orientado a objetos, se trabaja en el desarrollo del diagrama de clases para representar el modelo de datos durante la fase de diseño en cada incremento. Los requisitos son propensos a cambiar o sufrir leves variaciones durante el desarrollo, entonces éste diagrama tiende a modificarse en cada incremento y también como se emplea el servicio de Firebase *database realtime*, el cual es un almacén de datos no relacional orientado a objetos, se utiliza el diagrama de clases para representar las relaciones de estos datos. Además la metodología de comunicación que se quiere lograr entre el dueño de mascota y veterinario en la aplicación móvil, se representa mediante un diagrama de secuencia. Este diagrama está relacionado con

el diagrama de clases de proyecto, por lo cual si el diagrama de clases cambia durante el proceso de diseño en el incremento, entonces el diagrama de secuencia también debe modificarse para mantener la consistencia entre estos.

Es importante mencionar que los modelos realizados en esta fase no se recopilan en un documento como en el caso de los requisitos que se guardan en un documento de especificación de requisitos.

3.2.3. Fase de implementación

Esta fase consiste en traducir todos los modelos realizados en la etapa anterior en un software funcional. Para lograr esto primero se toman los *mockups*, es decir, los borradores de interfaces gráficas y convertirlos en un archivo *xml* con componentes gráficos, como por ejemplo: pantallas, botones, imágenes, entre otros.

Luego se traspasan las modificaciones realizadas en el diagrama de clases en código, para ello se debe actualizar los modelos hechos en lenguaje Java con los nuevos atributos o métodos del diagrama. Además se debe abstraer la información que entrega el diagrama de secuencia en la funcionalidad de comunicación entre dueño de mascota y veterinarios, ya que ésta funcionalidad se encuentra presente en casi todo el proceso de desarrollo por ser una actividad importante en la aplicación móvil.

Luego de haber completados los requisitos que se planificaron para un incremento en especial, se procede a realizar un *commit* de la rama asociada a la funcionalidad desarrollada mediante el software de control de versiones de código Git. Esto permite trabajar funcionalidades en ramas independientes, esto resulta útil, porque así los errores de una rama asociada a una funcionalidad no influyen en la otras y con esto se tiene un mejor control del código y de errores. Una vez hecho lo anterior, se procede a realizar la etapa de pruebas para asegurar que la implementaciones se encuentren libre de fallos.

3.2.4. Fase de pruebas

En esta etapa se procede a probar funcionalidades desarrolladas en el incremento. Para llevar a cabo esto se inicia testeando cada rama o *branch* por separado mediante pruebas de caja negra, en donde a las tareas realizadas se le asignan entradas y éstas deben responder con salidas esperadas o previamente definidas por el desarrollador,

entonces con este tipo de prueba se corrobora que la tarea desarrollada realiza correctamente su función. Una vez realizado lo anterior, se continua testeando pero ahora con la integración de las ramas, es decir, se prueba el funcionamiento en conjunto o en grupo de las tareas realizadas en el incremento.

Además de las pruebas unitarias, se hacen pruebas de flujo de la aplicación con usuarios reales, es decir, con dueños de mascotas y veterinarios con el fin de que ellos puedan detectar errores a medidas que ellos navegan y ejecutan acciones en la aplicación móvil. Luego de que las pruebas se realicen correctamente, se procede a dar por finalizado el incremento y se continua con el siguiente.

3.3. Aplicación metodología de evaluación

En este apartado se explica que tipo de pruebas se usan para medir el objetivo general del proyecto.

Como el objetivo general del proyecto es “Reducir el tiempo de búsqueda de una clínica veterinaria apta para la atención de urgencia de una mascota, a través de una aplicación móvil”, la variable tiempo es una variable cuantitativa, porque denota una cantidad, en este caso, cantidad de tiempo puede ser medido en segundos según el SI ².

Para el proyecto se utilizará evaluación mediante encuestas y experimentación. Las encuestas normalmente se hacen en retrospectiva, cuando por ejemplo, una técnica o herramienta se ha estado usando durante cierto tiempo. Tanto los datos cualitativos como cuantitativos se recogen a través de cuestionarios o entrevistas. Permiten obtener conclusiones descriptivas y exploratorias [22].

3.3.1. Encuesta

Para obtener la cantidad de tiempo que se demoran los dueños de mascotas en buscar una clínica veterinaria apta para atender la urgencia, se realiza una encuesta a los dueños de estas para obtener un tiempo de referencia que posteriormente se compara con los datos obtenidos mediante experimentación.

²Sistema Internacional de medidas

3.3.2. Experimentación

Para comprobar que los tiempos efectivamente se reducen se utiliza la experimentación, mediante la aplicación móvil a desarrollar como herramienta para experimentar frente una urgencia veterinaria. Cuando el dueño de la mascota crea una petición de urgencia para las clínicas veterinarias, la aplicación móvil toma la fecha (año-mes-día hora-minuto-segundo) cuando fue creada esta para guardarla en la base de datos, entonces cuando el médico veterinario acepte dicha petición, la aplicación móvil toma la fecha de aceptación para después guardarla en la base de datos y así calcular cuánto fue el tiempo transcurrido desde que se creó la petición hasta cuando esta fue aceptada.

A continuación se muestra como se aplica las fases de experimentación de ingeniería en la evaluación del proyecto.

1. **Definición:** En esta fase se determina cual va ser la variable de estudio, en este caso se va a estudiar el tiempo transcurrido entre que un dueño de mascota encuentra una clínica veterinaria apta para atender una urgencia. El objetivo principal de aplicar el experimento es comprobar si mediante el uso una aplicación móvil se reduce el tiempo mencionado anteriormente.
2. **Diseño:** Para llevar a cabo el experimento se define un ambiente de simulación de una urgencia, que involucra la participación de un usuario que tenga una mascota y un médico veterinario. Los participantes necesitan contar con un dispositivo móvil con sistema operativo Android y tengan el archivo con extension .apk ³ de la aplicación móvil. Se pretende usar un tiempo de experimentación estimado de entre una a dos semanas.
3. **Ejecución:** Una vez se compruebe que los participantes cuenten con aplicación instalada, se lleva cabo el experimento donde el dueño de mascota, dentro de los horarios en que las clínicas veterinarias atienden público, envía una petición de urgencia. Con esto, los médicos veterinarios pueden analizar el caso, dadas las condiciones en la que se encuentra la clínica al momento de recibir la petición y así ver si acepta o rechaza la urgencia.
4. **Análisis:** Cuando termine el proceso de ejecución del experimento, se calcula la diferencia de tiempo entre cuando la petición de urgencia fue enviada hasta

³Android Application Package, significado en español: Paquete de Aplicación Android

cuando fue aceptada. Una vez obtenido el tiempo anteriormente mencionado se procede a compararlo con el tiempo que el usuario indicó en la encuesta, donde se le pregunta cuando demoraba aproximadamente en encontrar una clínica veterinaria apta para atender la urgencia. Luego como último paso. se analiza si efectivamente existe una disminución entre estos tiempos.

3.4. Planificación del proyecto

El proceso de planificar un proyecto es importante, porque en este proceso se tiene un control de los recursos que se disponen para la realización del proyecto y con esto se tiene una mejor organización del mismo. Los recursos del proyecto se pueden definir como el tiempo que se requiere para completar el proyecto, personal disponible, asignación y responsabilidad de quien realiza cada tarea, recursos monetarios (en caso que los hubiera), etc.

La planificación se realiza con el formato de carta Gantt, además se utiliza el software llamado *Gantt Project* para la construcción de ésta y se divide en dos partes:

- Planificación del desarrollo de software.
- Planificación de la aplicación de la evaluación del proyecto.

3.4.1. Planificación del desarrollo de software

Como se mencionó en la Sección 3.1 para utilizar la metodología de desarrollo, se dividió la construcción del software en 8 incrementos con una duración de dos semanas cada uno. Cada incremento se compone de tareas de implementación, diseño y testing de lo realizado, además de mostrar la fecha de inicio, termino y la duración en días como se muestra en la Figura 3.3.

☐	• Incremento 2	20/05/19	2/06/19	14
☐	• Módulo perfil de mascota	20/05/19	23/05/19	4
	• Crear perfil mascota	20/05/19	22/05/19	3
	• Mostrar información mascota	23/05/19	23/05/19	1
	• Modificar información mascota	23/05/19	23/05/19	1
	• Asignar a veterinario privilegios admin	24/05/19	27/05/19	4
☐	• Módulo creación veterinaria	28/05/19	30/05/19	3
	• Agregar información clínica vete...	28/05/19	29/05/19	2
	• Mostrar información clínica	30/05/19	30/05/19	1
	• Modificar información clínica	30/05/19	30/05/19	1
	• Agregar veterinarios clínica	27/05/19	29/05/19	3
☐	• Testing	23/05/19	2/06/19	11
	• Testing perfil mascota	23/05/19	23/05/19	1
	• Testing asignación veterinario ad...	28/05/19	30/05/19	3
	• Testing creación clínica veterinaria	31/05/19	2/06/19	3

Figura 3.3: Tareas de una iteración de proyecto.

3.4.2. Planificación de la evaluación del proyecto.

La planificación de la evaluación del proyecto consta de cuatro tareas:

- Creación de encuesta para los dueños de las mascotas
- Difusión de la encuesta
- Contacto con clínicas veterinarias
- Periodo de prueba del proyecto

☐	• Planificación evaluación	1/10/19	29/10/19	29
☐	• Creación encuesta para dueños de masc...	1/10/19	2/10/19	2
	• Formular preguntas	1/10/19	2/10/19	2
	• Difundir encuesta a los dueños de masc...	3/10/19	9/10/19	7
	• Contacto con clínicas veterinarias	1/10/19	7/10/19	7
	• Período de prueba proyecto	10/10/19	29/10/19	20

Figura 3.4: Tareas de la planificación evaluación del proyecto.

En la Figura 3.4 se muestra el nombre, fecha de inicio, término y la duración en días de las tareas respectivamente. A continuación se explica en detalle cada tarea.

- La primera tarea consiste en crear una encuesta para que sea posteriormente aplicada a los dueños de mascota, el objetivo de esta es obtener una referencia de cuanto tiempo se demorarían ellos en encontrar una clínica veterinaria que sea apta para atender una urgencia veterinaria.
- La segunda tarea consiste en difundir la encuesta mencionada anteriormente para que los dueños de mascota la respondan y así obtener los resultado que luego serán comparados con los datos que entregue la aplicación móvil cuando esta sea probada por los usuarios.
- La tercera tarea consta en hacer contacto con distintas clínicas veterinarias todo esto para que tengan en consideración la existencia de una aplicación móvil que les avise cuando un dueño de mascota necesita una atención de urgencia.
- La cuarta y última tarea consiste que usuarios dueños de mascota y médicos veterinarios hagan uso de esta. Esta tarea tiene una duración de 20 días, lo cual deja un gran rango de días para que los usuarios la prueben.

3.5. Resumen del capítulo

En el presente capítulo se abordó la forma en que un dueño de mascota, mediante notificaciones con Firebase Cloud Messaging, envía peticiones de urgencia a las clínicas veterinarias para que los veterinarios que trabajan allí puedan analizar la situación y aceptar o no la petición realizada.

Además se dió a conocer el modo en que se utiliza el patrón Modelo-Vista-Controlador en un desarrollo Android con el IDE Android Studio.

Lo mencionado anteriormente dice relación a cómo lo expuesto en el Marco Teórico se ocupa en el desarrollo del proyecto.

Otro aspecto importante que se trató en el presente capítulo es la mención de la metodología de desarrollo de software junto con la de evaluación. La metodología de desarrollo que se presenta es la iterativa incremental, en la que el trabajo se divide en ocho incrementos con una duración de dos semanas cada uno. También se explica como se trabajaron las fases de análisis, diseño, implementación y pruebas en el desarrollo del proyecto.

En cuanto a la metodología de evaluación se ocupa la encuesta para obtener el tiempo de referencia en que un dueño de mascota encuentra una clínica veterinaria apta para atender una urgencia. Luego, usando experimentación, mediante la aplicación móvil, obtener otro tiempo de respuesta, para finalmente comprobar si los tiempo de respuesta obtenidos por la encuesta junto con la experimentación se reducen.

Por último se menciona el proceso de planificación para realizar el proyecto. La planificación del trabajo completo se divide en dos partes: la primera parte consiste en la planificación del desarrollo, que hace relación a la metodología de desarrollo, mientras que la segunda es acerca de la metodología de evaluación. En la de desarrollo se menciona como se estructura un incremento cualquiera, nombrando las tareas involucradas con sus respectiva fecha de inicio y término. Por su parte en la de evaluación se indica como se organiza las tareas involucradas como lo son la creación de encuestas y período de experimentación de la aplicación con su respectiva fecha de inicio y término.

4. Implementación de la solución

El presente capítulo trata de cómo, lo expuesto en el capítulo de metodología, se lleva cabo en la implementación del proyecto, en donde se aborda la puesta en práctica de la metodología de desarrollo y como se planificaron los incrementos que ayudaron a construir la aplicación. Dentro de la metodología de desarrollo se aborda cómo se ejecuta la fase de análisis (o requisitos), destacando los tipos de requisitos capturados. También se trata los distintos diagramas de la fase de diseño, que permiten llevar a cabo la implementación de la aplicación móvil. Además de exponer el cómo se construye la aplicación móvil, poniendo énfasis en las principales funcionalidades de ésta. Para por último mostrar las pruebas realizadas a la aplicación.

4.1. Plan de trabajo

En esta sección se aborda la planificación de cada incremento. Cabe recordar que para el proyecto se planificó con ocho incremento de 2 semanas cada uno, lo que da un tiempo de desarrollo de 4 meses.

- **Incremento n°1 (Fecha inicio: 06-05-19 - Fecha término: 19-05-2019)**
 - Construcción mockups.
 - Captura requisitos.
 - Diagrama de clases.
 - Implementación base de datos.
 - Desarrollo Registro de usuarios.
 - Desarrollo ingreso de usuarios.
 - Testing.

- **Incremento n°2 (Fecha inicio: 20-05-2019 - Fecha término: 02-06-2019)**
 - Desarrollo perfil mascota.
 - Desarrollo creación de clínica veterinaria.
 - Desarrollo agregar veterinarios a una clínica.
 - Asignar privilegios de administrador a un veterinario de una clínica.
 - Testing.

- **Incremento n°3 (Fecha inicio: 03-06-2019 - Fecha término: 16-06-2019)**
 - Implementar mapa de Google Maps.
 - Crear petición de urgencia para una mascota.
 - Enviar petición de urgencia.
 - Confirmar petición de urgencia.
 - Cancelar petición de urgencia.
 - Testing.

- **Incremento n°4 (Fecha inicio: 17-05-19 - Fecha término: 30-06-2019)**
 - Agregar síntomas de la mascota en la petición de urgencia
 - Agregar imagen de la mascota a la petición de urgencia.
 - Agregar vídeo de la mascota a la petición de urgencia.
 - Envío notificación a veterinarios.
 - Testing.

- **Incremento n°5 (Fecha inicio: 05-08-2019 - Fecha término: 18-08-2019)**
 - Mostrar clínica veterinaria en el mapa
 - Recibir petición de urgencia.

- Ver detalles de la urgencia.
 - Confirmar petición.
 - Testing.
- **Incremento n°6 (Fecha inicio: 19-08-2019 - Fecha término: 01-09-2019)**
 - Posición dueño de mascota.
 - Notificación de confirmación petición de urgencia.
 - Agregar veterinarios favorito.
 - Mostrar detalles clínicas veterinarias.
 - Seleccionar veterinarios para la petición de urgencia.
 - Testing.
- **Incremento n°7 (Fecha inicio: 03-09-2019 - Fecha término: 17-09-2019)**
 - Enviar petición de urgencia a una clínica veterinaria.
 - Zoom imagen en el detalle de la petición.
 - Compartir perfil de la mascota.
 - Reenviar petición de urgencia.
 - Testing.
- **Incremento n°8 (Fecha inicio: 18-09-2019 - Fecha término: 01-10-2019)**
 - Perfil dueño de mascota
 - Perfil veterinario
 - Eliminar veterinario de favoritos
 - Listar veterinarios de una clínica veterinaria
 - Eliminar veterinarios de una veterinaria
 - Buscador de clínicas veterinarias
 - Generar archivo APK.
 - Testing.

4.2. Requisitos

La ingeniería de requisitos permite la gestión adecuada de los requisitos de un proyecto de desarrollo software. Además, mejora la capacidad para realizar planificaciones de los procesos de proyectos de desarrollo software puesto que el conocer qué se tiene que desarrollar permite una efectiva proyección de las actividades, recursos, costos, tiempos, etc. del proyecto[25]. Los requisitos obtenidos en esta etapa se clasifican en dos categorías:

- **Requisitos de usuario:** Componen el proceso de organizar la información sobre las necesidades del usuario. Ofrecen al usuario una vista del problema pero no al desarrollador[25].
- **Requisitos de software:** Permite que el desarrollador construya un modelo de implementación que es necesario donde se muestra que es lo que el sistema debe hacer[25].

RU009 - Crear una petición de urgencia indicando los síntomas de la mascota.	
Descripción	El dueño de una mascota podrá crear una petición de urgencia, indicando información de los síntomas que presenta la mascota.
Fuente	Dueño de mascota
Prioridad	Crítica
Estabilidad	Estable
Fecha actualización	15-05-2019
Estado	Cumple
Incremento	1
Tipo	Funcional
Usuario asociado	Dueño de mascota

Cuadro 4.1: Especificación de un requisito funcional del proyecto.

En la tabla 4.1 se muestra un requisito de usuario del proyecto. El requisito se identifica con el código RU009, es decir, es un requisito de usuario y es el número nueve del listado. Este tipo de requisito da una idea al usuario de lo que debe realizar la aplicación móvil, pero al desarrollador.

Una vez obtenidos los requisitos de usuario, se puede proceder a realizar *mockups* de interfaces de usuario, diagramas de flujo de la aplicación, diagramas de clases, casos de uso, con el fin de que el desarrollador adquiriera un conocimiento más técnico de que debe realizar la aplicación y así obtener los requisitos de software.

RS012 - Implementar un campo de texto para agregar síntomas que la mascota posee.	
Descripción	El dueño podrá escribir en un <i>TextArea</i> los síntomas que presenta. Se opta por éste componente, porque permite ingresar gran cantidad de texto además que es multilinea.
Fuente	Dueño de mascota
Prioridad	Critica
Estabilidad	Estable
Fecha actualización	15-05-2019
Estado	Cumple
Incremento	1
Tipo	Funcional
Usuario asociado	Dueño de mascota
RU asociado	RU009

Cuadro 4.2: Especificación de un requisito de software del proyecto.

En la tabla 4.2 se muestra un requisito de software. Este tipo de requisitos da indicios al desarrollador de lo que debe realizar la aplicación móvil.

4.2.1. Casos de uso

En la realidad ningún sistema se encuentra totalmente aislado, puesto que siempre hay un actor, ya sea un ser humano o mecánico que interactúa con el sistema. Un caso de uso especifica el comportamiento de un sistema o una parte del mismo, y es una descripción de un conjunto de secuencias de acciones, donde cada secuencia representa la interacción de los elementos externos del sistema (sus actores) con el propio sistema. Un caso de uso representa un requerimiento funcional del sistema[26].

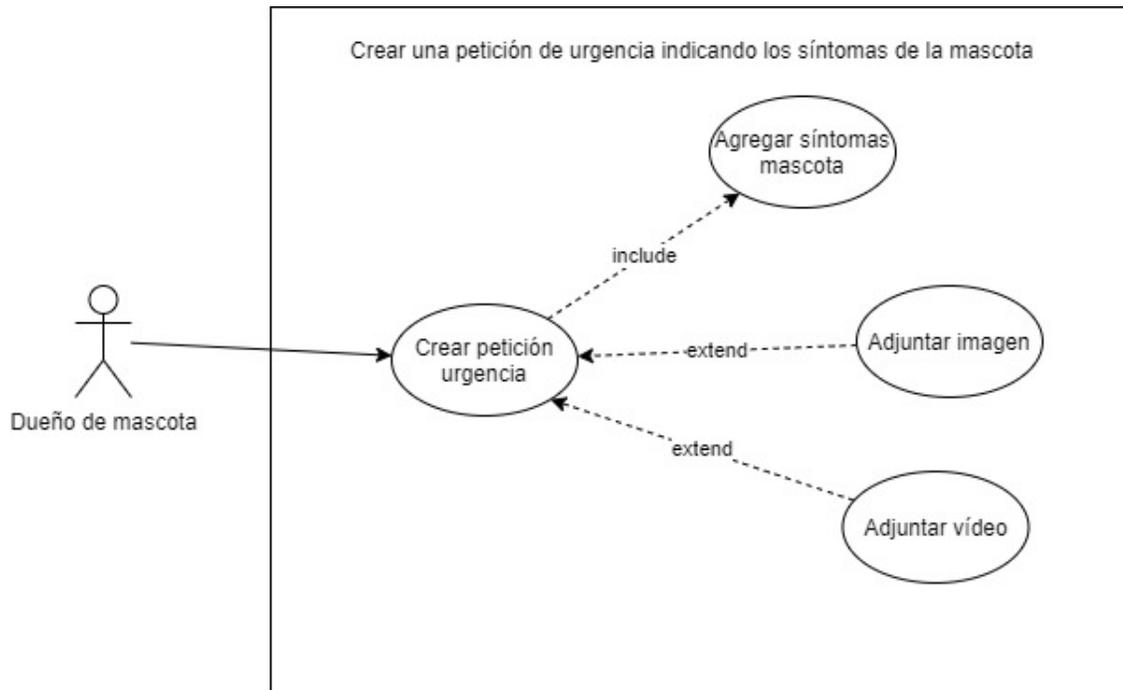


Figura 4.1: Caso de uso asociado al requisito RU009.

En la Figura 4.1 se ve el caso de uso para la funcionalidad de crear una alerta de petición. Para crear una petición el dueño (actor) debe suministrar información de los síntomas de la mascota (include). También a la petición se puede agregar una imagen y/o vídeo, pero estos últimos son opcionales (extend).

A modo de resumir el trabajo realizado, solamente se exponen en esta sección un reducido número de requisitos de usuario y de software con sus respectivos casos de uso. La totalidad de requisitos de usuario y de sistema, junto con los caso de uso y matriz de seguimiento, se pueden hallar en el documento de requisitos que se encuentra en el Anexo D.

4.3. Arquitectura

La arquitectura se define como: el nivel conceptual más alto de un sistema en su ambiente. Arquitectura es la organización fundamental de un sistema descrita en: Sus componentes, relación entre ellos y con el ambiente, principios que guían su diseño y evolución[27]. La arquitectura es importante porque entrega documentación temprana de las decisiones de diseño, restricción temprana de la implementación,

permite predecir cualidades del sistema, facilita o inhibe propiedades del sistema, etc.

4.3.1. Arquitectura física

La arquitectura física se encarga de modelar, a grande rasgos, la comunicación entre los distintos componentes físicos que involucra el proyecto, es decir, componentes que participan en la solución como lo son clientes, servidores, BD, etc. Para este proyecto se usa una arquitectura cliente - servidor. En dicha arquitectura los clientes son los dispositivos móviles, los que son usados tanto como por dueños de mascotas como veterinarios. A continuación se muestra la arquitectura cliente-servidor usada en el proyecto.

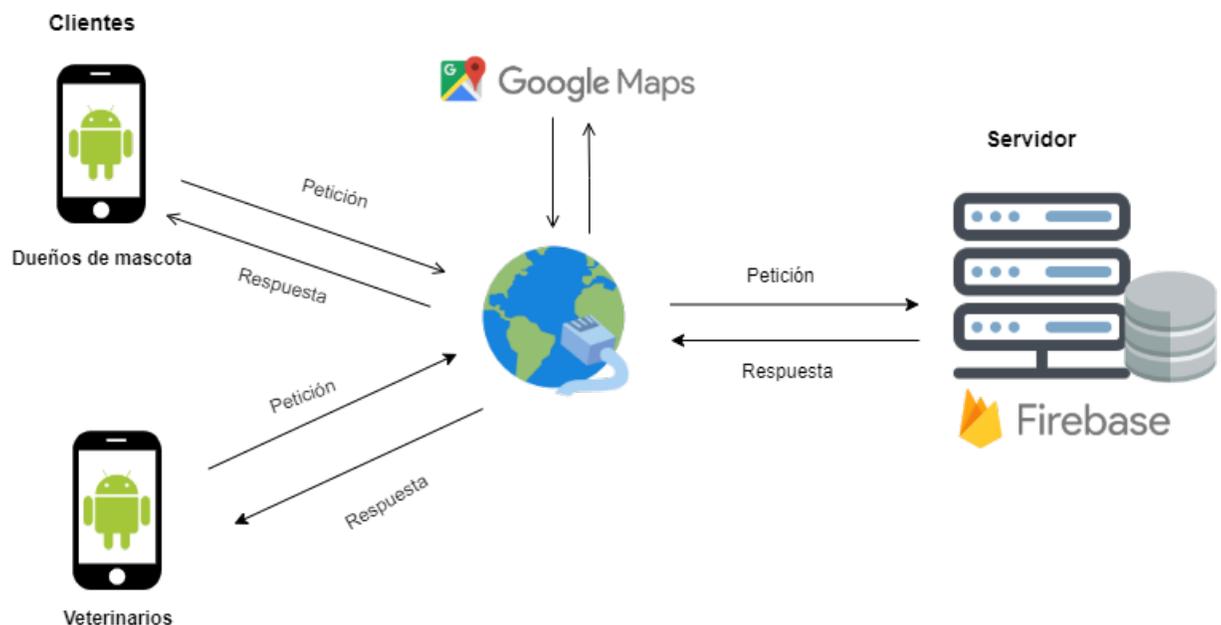


Figura 4.2: Arquitectura física

En la Figura 4.2 se muestra cómo los clientes interactúan con el servidor mediante peticiones y respuestas que se realizan a través de Internet. Cada vez que los clientes realizan peticiones al servidor (Firebase de Google), como por ejemplo, registrarse, ingresar a la aplicación, crear el perfil de una mascota, crear una urgencia, etc, serán procesadas por éste, para luego enviar una respuesta al cliente que puede ser un mensaje de aviso que indica que la petición fue procesada de forma exitosa o fallida.

4.3.2. Arquitectura lógica

La arquitectura lógica expresa cuales son los componentes lógicos (subsistemas o macro-funciones) que participan en la solución del proyecto y la relación entre ellos. Para visualizar la arquitectura lógica se descompone cada componente de la arquitectura física, como el cliente y el servidor, en subsistemas con el fin de comprender la manera en que interactúan éstos. El lado cliente está compuesto en su interior con un patrón de diseño Modelo-Vista-Controlador que el mismo Android Studio configura por defecto. Por otro lado, al ocupar la herramienta Firebase como servidor, no se tiene total comprensión de como se compone en su interior, pero se conoce que servicios este provee a la aplicación.

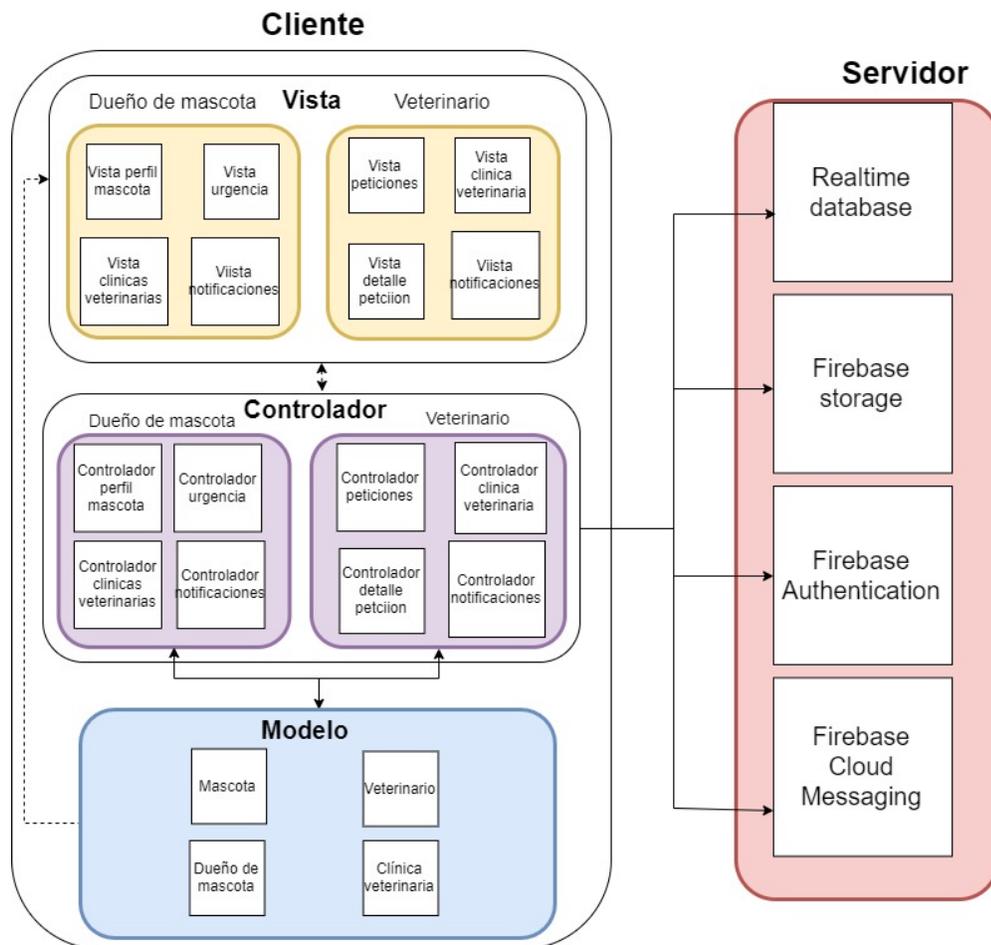


Figura 4.3: Arquitectura lógica

En la Figura 4.3 se muestra como el componente cliente se descompone en sub-

componentes al igual que el servidor.

El lado del cliente implementa un módulo de vistas, que a su vez se descompone en la vista para dueño de mascota y médico veterinario. Estos sub-componentes se encargan de mostrar información obtenida de las clases que se manejan en el modelo hacia los dispositivos móviles de los usuarios. Por otro lado se tiene el módulo controlador, que al igual que el módulo vista, se descompone en controlador para el dueño de mascota y veterinario.

El controlador se encarga de manejar los eventos que el usuario invoca, como por ejemplo realizar petición al servidor, CRUD de datos (Database realtime), solicitar archivos al servidor (Firebase storage), enviar y recibir notificaciones a los dispositivos móviles (Firebase Cloud Messaging) y autenticar a los usuarios (Firebase authentication). Además éste componente se encarga de ser el que transporta los datos suministrador por modelo para luego ser mostrado en la vista.

Por último el modelo se encarga de implementar las clases que manejan la lógica de la aplicación para posteriormente ser mostradas en la vista por medio del controlador. En este caso las principales clases que implementan esta lógica son: la clase Dueño de mascota, Mascota, Veterinario y Clínica veterinaria.

4.3.3. Diagrama de clases

La aplicación se construye bajo el paradigma de programación orientada a objetos (POO). Para realizar un buen diseño bajo este paradigma es correcto realizar un diagrama de clases. Un diagrama de clases es usado para describir una vista estática de una aplicación. Está compuesto de clases y relaciones. Una clase es una descripción de un concepto y que tiene atributos y operaciones asociadas, estas clases son representadas como rectángulos en el diagrama. Mientras que una relación representa una conexión lógica entre dos clases y se muestra gráficamente como una línea entre dos clases [28].

En POO, una clase es una descripción de un conjunto de objetos que en la vida real comparten los mismos atributos, operaciones, relaciones y semántica. En el contexto del proyecto a desarrollar, un ejemplo de una clase es una mascota, porque las mascotas comparten características en común, como por ejemplo, nombre, tipo, genero, etc, a estas características, en POO, se le conoce como atributos de una clase. En cuanto a la relación que presenta una clase mascota, se debe nombrar a la clase

dueño, porque para que un animal sea considerado mascota debe pertenecer a un dueño.

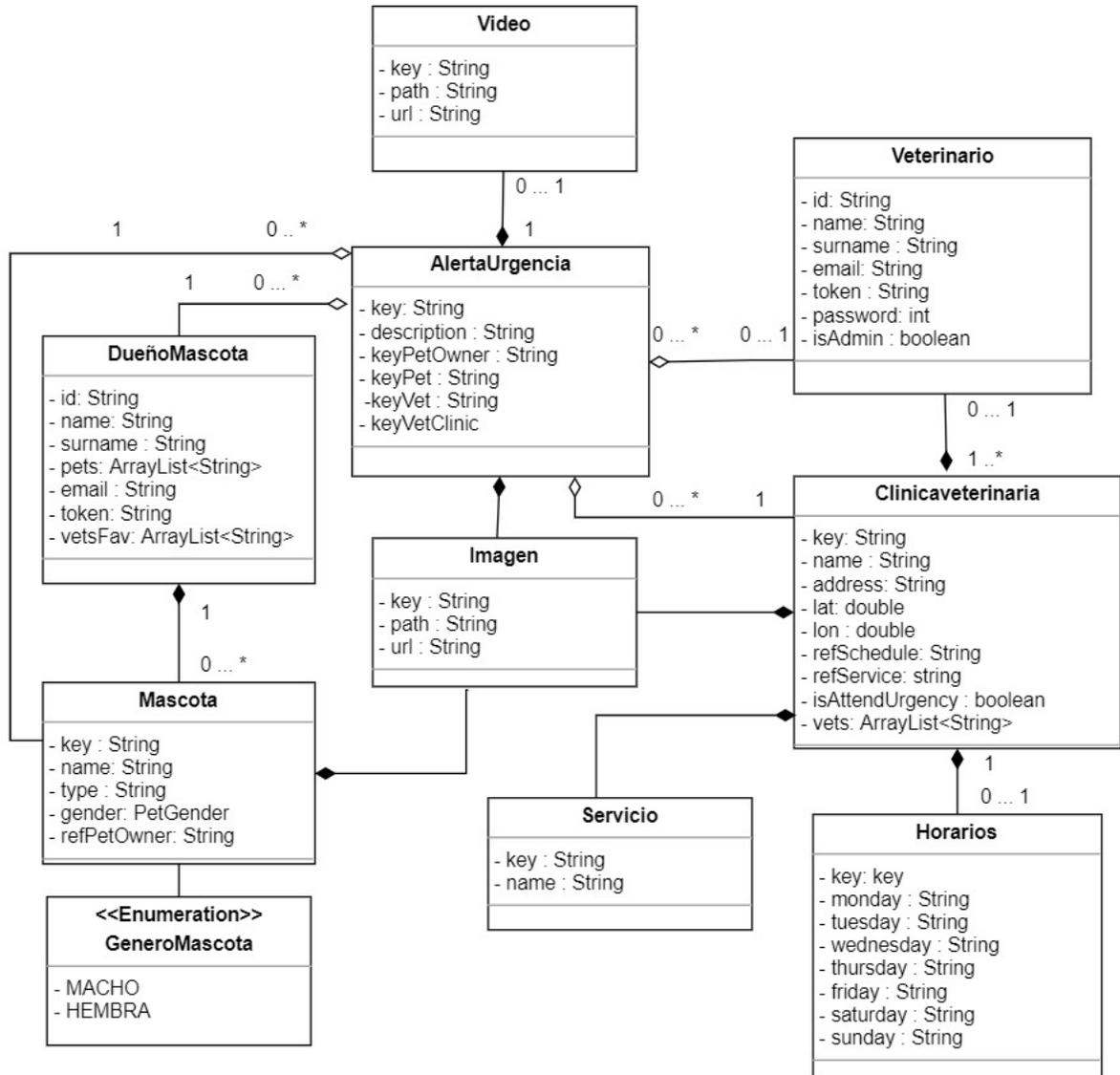


Figura 4.4: Diagrama de clases del proyecto

En POO existe dos tipos de relaciones: dinámicas y estáticas. Las relaciones dinámicas se le conoce con el nombre de composición (relación fuerte) y se simboliza con un rombo relleno negro, mientras que las estática se le conoce como agregación (relación débil) y se simboliza con un rombo blanco. En la Figura 4.4 se ve que la clase Mascota se relaciona con DueñoMascota, ya que éste último puede tener muchas mascotas. Además se relaciona con DueñoMascota mediante una composición

(relación fuerte), es decir, si una instancia de DueñoMascota se destruye, la instancia de Mascota asociado al dueño no tendría razón de existir, porque en el contexto del proyecto no existen mascota que no tengan dueños asociados. Por otro lado la clase DueñoMascota con AlertaUrgencia se relaciona mediante una agregación (relación débil), en donde si se destruye una instancia de AlertaUrgencia, la instancia DueñoMascota puede seguir funcionando normalmente.

4.3.4. Diagrama de secuencia

El diagrama de secuencia muestra la interacción que existe entre los objetos de la clases del proyecto mediante el envío y/o recepción de mensajes a lo largo del tiempo [29]. En resumen, se podría decir que un diagrama de secuencia es un diagrama de clase pero en movimiento.

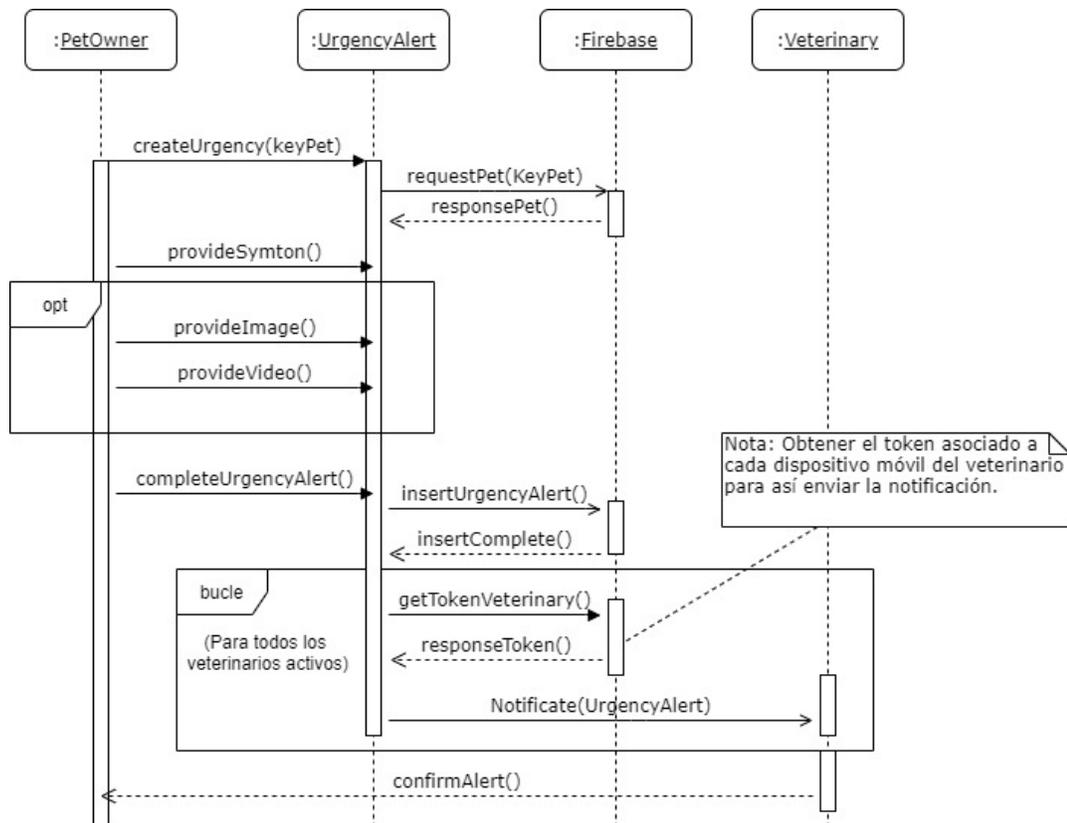


Figura 4.5: Diagrama de secuencia de la interacción entre el dueño de mascota y el veterinario.

La Figura 4.5 muestra la secuencia que existe cuando un dueño de mascota crea y

envía una petición de urgencia a los veterinarios. Cabe destacar que desde la versión 2.0 de UML se implementaron *marcos de interacción* [29]. En este caso el marco de interacción *opt* hace referencia a una secuencia opcional, mientras que *bucle* se refiere a una secuencia que se realiza una cierta cantidad de veces hasta que se cumpla una determinada condición.

4.3.5. Base de datos Firebase

Firebase provee una base de datos en tiempo real *NoSQL*. Los datos no son guardados en tablas como lo hacen los motores de base de datos relacionales como *MySQL* ó *PostgreSQL*, sino que se guardan como un objeto en formato *JSON*¹ y el dato almacenado se guarda con un formato clave-valor.

```
-M26D8WG1HTODQA6CHz9
{
  "agePet": "Adulto (Entre 3 a 10 años)",
  "dateAccepted": "",
  "dateCreated": "2020-03-10 22:29:31",
  "dateSymptom": "aproximadamente hace 1 día",
  "description": "Hola buenas tardes, hace unos días mi gata tier",
  "key": "-M26D8WG1HTODQA6CHz",
  "keyImage": "-M26DL1JK1BTv77Wz3e",
  "keyPet": "-LsZnLXBhBqdJBPKnWu",
  "keySender": "yY2XmpIjRFntBtmihWPHpOCKIKr",
  "keyVeterinary": "",
  "keyVeterinaryClinic": "",
  "keyVideo": "-M26DIhegjEE23xEDda",
  "procedure": "No, no he suministrado ningun medicamento porq",
  "sizePet": "",
  "state": true,
  "symptom": "dolor\nperdida apetito\nrespiración agitada\ndeca:",
  "typeEnvironment": ""
}
```

Figura 4.6: Formato JSON de un objeto almacenado en Firebase. Fuente: Elaboración propia

En la Figura 4.6 se muestra una instancia de la clase Urgencia almacenada en Firebase. Cuando dicho objeto se carga en la base de datos, éste se guarda en for-

¹JavaScript Object Notation, es un formato de intercambios de datos.

mato clave-valor, donde el nombre del atributo representa la clave y el contenido del atributo representa el valor. Cabe mencionar que si un atributo de un objeto tiene un valor nulo, entonces el valor se guarda como un string vacío en la base de datos.

4.3.6. Implementación Firebase realtime database

Para implementar los servicios de Firebase realtime database en un desarrollo con Android, se tiene que implementar las librerías de dicho servicio en un archivo que el IDE Android Studio crea cuando se inicia un proyecto. Este archivo es el *build.gradle*, el cual maneja las librerías que se pueden implementar en Android.

```
implementation 'com.google.firebase:firebase-core:16.0.9'  
implementation 'com.google.firebase:firebase-database:16.1.0'
```

Figura 4.7: Declaración de las librerías necesarias para usar Firebase en Android Studio.

En la Figura 4.7 se muestra la sentencia que permite usar el servicio de Firebase realtime en Android. Una vez realizado esto, la aplicación puede realizar el *CRUD* con la base de datos.

Como se trabaja con una base de datos no relacional, entonces se hace omisión del diagrama entidad relación y cualquier diagrama que se derive de este. Sin embargo el modelo más fiel o que más representa al diagrama entidad-relación ocupado en este proyecto, para modelar la base de datos, es el diagrama de clases que se muestra en la Figura 4.4.

4.3.7. Implementación Firebase Authentication

Para implementar el servicio de autenticación de un usuario en Firebase se tiene que implementar una librería en el archivo *build.gradle* que proporciona Android Studio.

```
implementation 'com.google.firebase:firebase-auth:16.2.1'
```

Figura 4.8: Declaración de librería que permita autenticar a un usuario.

La Figura 4.8 muestra la implementación de la librería de Firebase Authentication. Además este servicio cuenta con un panel que permite listar todos los usuarios que han sido registrados en el sistema, lo que permite tener un control de estos.



The screenshot shows the Firebase Authentication control panel. At the top, there is a search bar with the text "Buscar por dirección de correo electrónico, número de teléfono o UID de usuario" and a blue button labeled "Añadir usuario". Below the search bar is a table with the following columns: "Identificador", "Proveedores", "Fecha de creación", "Inicio de sesión", and "UID de usuario ↑". The table contains three rows of user data.

Identificador	Proveedores	Fecha de creación	Inicio de sesión	UID de usuario ↑
vet3@vetapp.com	✉	24 sept. 2019	24 sept. 2019	1IIRbnGuBuOXX4H5Wmo1yEXnm...
vet1@vetapp.com	✉	24 sept. 2019	24 sept. 2019	AjWjYvI9dufHqP2BFyTFpQpdFFx2
vet2@vetapp.com	✉	24 sept. 2019	24 sept. 2019	EvecMVE03EfjQZhkMWF1rXvd1NI2

Figura 4.9: Panel de control de Firebase Authentication.

En la Figura 4.9 se muestra una tabla con usuarios que pertenecen al sistema. En dicha tabla se muestra el identificador del usuario (en este caso el correo), la fecha de creación, fecha del último inicio de sesión y un código que funciona como identificador para cada usuario que el mismo Firebase crea. Firebase Authentication provee diversas formas de autenticar a un usuario, para este proyecto se implementa la autenticación mediante correo electrónico y contraseña. Existen otras formas como por ejemplo: mediante Facebook, Google, Twitter, Github, número telefónico, etc.

4.3.8. Implementación Firabase Storage

Para trabajar con este servicio se tiene que importar la siguiente librería en el archivo *build.gradle*.

```
implementation 'com.google.firebase:firebase-storage:16.1.0'
```

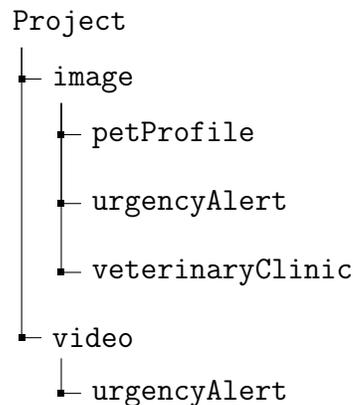
Figura 4.10: Librería que permite trabajar con Firebase Storage

La Figura 4.10 muestra la sentencia que permite usar este servicio. Al igual que los otros servicios mencionados anteriormente, Firebase Storage provee una interfaz de usuario que permite gestionar los archivos, como imágenes y vídeos, en carpetas para lograr así un mejor manejo y orden de estos. El funcionamiento es similar a los exploradores de archivos con los que cuentan los sistemas operativos como Windows y Ubuntu, en donde se pueden crear carpetas, subcarpetas y con esto lograr una jerarquía.



Figura 4.11: Explorador de carpetas de Firebase Storage.

La Figura 4.11 muestra el listado de carpetas usadas en el proyecto. Como la implementación del proyecto utiliza imágenes y vídeos, se opta por crear dos carpetas en donde cada una almacena los distintos archivos que se crean a medida que los usuarios interactúan con la aplicación. Dentro de las carpetas existen subcarpetas, las cuales se explican a continuación.



Como se muestra en la jeraquia anterior, la carpeta *image* contiene 3 subcarpetas, donde la primera almacena imágenes del perfil de la mascota, la segunda guarda imágenes cuando un usuario crea una petición de urgencia y por último la tercera contiene las imágenes de las clínicas veterinarias. Mientras tanto la carpeta *video* dentro contiene una subcarpeta, la cual guarda los vídeos que los usuarios proporcionan cuando crean una petición de urgencia.

4.3.9. Implementación Firebase Cloud Messasing

Al igual que los anteriores servicios, se debe implementar la siguiente sentencia en el archivo *build.gradle*.

```
implementation 'com.google.firebase:firebase-messaging:17.6.0'
```

Figura 4.12: Sentencia que permita usar las funcionales que Firebase Cloud Messaging entrega.

La Figura 4.12 muestra la implementación de la librería que permite usar los servicios de *Firebase Cloud Messaging* en Android Studio. Para el envío de notificaciones, hay que obtener un *token* del servidor que el servicio provee. Luego la notificación es enviada en un formato *JSON* mediante una petición *POST*, para ello se necesita obtener una *url* a la cual se hace la petición.

```
{ "to": "",  
  "data": {  
    "title": "",  
    "body": "",  
    "key": "",  
    "notificationType": ""  
  }  
}
```

Figura 4.13: Estructura en formato *JSON* de la notificación.

La Figura 4.13 muestra la estructura de la notificación que es utilizada para este proyecto. A continuación se explica el uso de cada campo.

- *to*: Este campo asigna el *token* de cada usuario, es decir, la notificación se envía al dispositivo móvil que corresponde a dicho *token*. Cabe destacar que este campo es obligatorio en la estructura de la notificación.
- *data*: Este campo en su interior tiene el contenido de la notificación.
- *title*: Muestra el título de la notificación.
- *body*: Muestra el cuerpo de la notificación.
- *key*: Contiene la referencia de la información del objeto que se quiere mostrar cuando se abra la notificación. Este campo es creado para el propósito del proyecto.

- `notificationType`: Este campo guarda un código numérico que distingue el tipo de notificación, porque en el proyecto existen notificaciones para cada propósito y es bueno saber para el desarrollador de que tipo de notificación se está trabajando. Al igual que el campo anterior mencionado, éste es creado para el propósito del proyecto.

4.4. Desarrollo Aplicación móvil

La siguiente sección aborda la implementación de las principales funcionalidades de la aplicación móvil basada en los incrementos considerados para el proyecto. Las funcionalidades restantes con sus respectivas capturas, se pueden encontrar en el Anexo C del documento.

4.4.1. Incremento 1

El primer incremento del proyecto involucra la creación de la pantalla o *activity* de inicio de la aplicación móvil, esto permitirá a los usuarios poder registrarse e ingresar como dueños de mascota o veterinarios. El mecanismo de registro es mediante un formulario que implementa un par de campos de texto o *EditText*. Luego haber completado el formulario de registro y ésta se haya almacenado correctamente en la base de datos Firebase, el usuario puede ingresar a la aplicación mediante su correo electrónico y una contraseña.

El **primer incremento** tiene la finalidad entregar un software funcional, en donde los usuarios, dueños de mascotas y veterinarios, puedan registrarse e ingresar correctamente al sistema. Es importante recalcar que las capturas de pantalla del registro e ingreso a la aplicación para el dueño de mascota y veterinario, se pueden encontrar en el Anexo C.1.1 y C.2.1 respectivamente. Además destacar que para este incremento se utilizan los siguientes servicios de Firebase.

- Firebase realtime database.
- Firebase authentication.

4.4.2. Incremento 2

Una vez que el usuario dueño de mascota ingresa al sistema, él visualizará el menú principal de la aplicación, el cual consiste de un *Tabbed Activity*, es decir, una

pantalla que cuenta con pestañas, dichas pestañas son *fragments*, estos son porciones de interfaz de usuario que pueden componer una *activity*. A continuación se explica el propósito de cada pestaña.

- Urgencia, en esta pestaña el dueño de mascota podrá realizar una petición de urgencia para una mascota. También podrá visualizar en un mapa todas las clínicas veterinarias que están registradas en la aplicación, con el fin que el usuario pueda ver información detallada de la clínicas. Además el usuario tiene la opción de realizar una petición una clínica veterinaria en específico.
- Peticiones, en esta pestaña se listan las peticiones que el usuario ha realizado. También el usuario al presionar una petición puede ver el detalle de ésta.
- Mascotas, en esta última pestaña el usuario tiene la opción de crear una mascota. Además en ésta se listan las mascotas que el dueño posee. Si el usuario presiona una mascota de la lista, podrá editar la información de ésta.

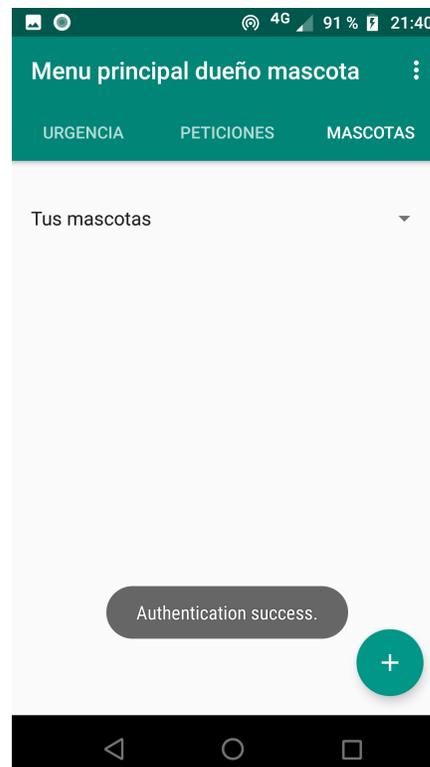


Figura 4.14: Menú principal dueño de mascota.

En la Figura 4.14 se muestra el menú principal del dueño de mascota, en donde está seleccionada la pestaña *mascotas*. Para que un dueño de mascota pueda crear peticiones de urgencia, primero debe crear una mascota y para ello debe presionar el *button* o botón circular que aparece en la parte inferior derecha de la figura mencionada. Una vez presionado el botón, se abre una *activity* o pantalla en donde aparece un formulario para ingresar información básica de la mascota.

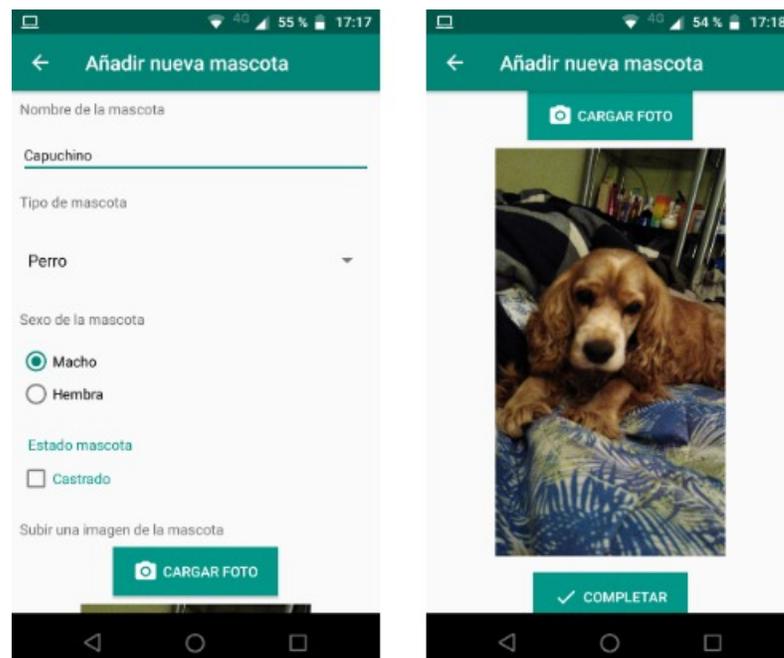


Figura 4.15: Formulario para crear una mascota.

En la Figura 4.15 se muestra el formulario que debe completar el usuario para crear una mascota. El formulario consta de los siguientes componentes gráficos:

- Un campo de texto o *EditText* para ingresar el nombre de la mascota.
- Un menú de opciones desplegable o *Spinner* para seleccionar el tipo de mascota (perro o gato).
- Un botón de opciones o *Radiobutton's* para escoger el sexo de la mascota y si ésta se encuentra esterilizada o castrada.
- Un botón que permite al usuario proporcionar una foto de la mascota, ya sea usando la galería del dispositivo móvil o utilizando la cámara del mismo.

Luego el usuario puede completar la creación de la mascota apretando el botón completar. Una vez que se almacenan los datos de la mascota correctamente en la base de datos, el sistema indica mediante un mensaje que la creación de la mascota ha sido exitosa o de lo contrario avisa que la creación ha fallado.



Figura 4.16: Datos de la mascota almacenados en la base de datos.

En la Figura 4.16 se muestran los datos de la mascota almacenados en Firebase teniendo en cuenta el formato JSON. Cabe destacar que para almacenar una imagen de una mascota en la base de datos, se usa el servicio Firebase storage. Este servicio permite almacenar archivos, ya sea imágenes y videos, para luego poder recuperarlos mediante una *url*. Además para facilitar la recuperación de una imagen que está almacenada en la base de datos se ocupa una librería llamada *Glide*.

```

implementation 'com.github.bumptech.glide:glide:4.9.0'
annotationProcessor 'com.github.bumptech.glide:compiler:4.9.0'

```

Figura 4.17: Sentencias para implementar la funcionalidades de la librería *Glide*

Firebase storage cuenta con un panel donde se puede visualizar, en una lista, todos los archivos que han sido almacenados. También permite gestionar carpetas dentro del panel, con esto, se hace más fácil organizar los distintos archivos con los que cuenta el proyecto.

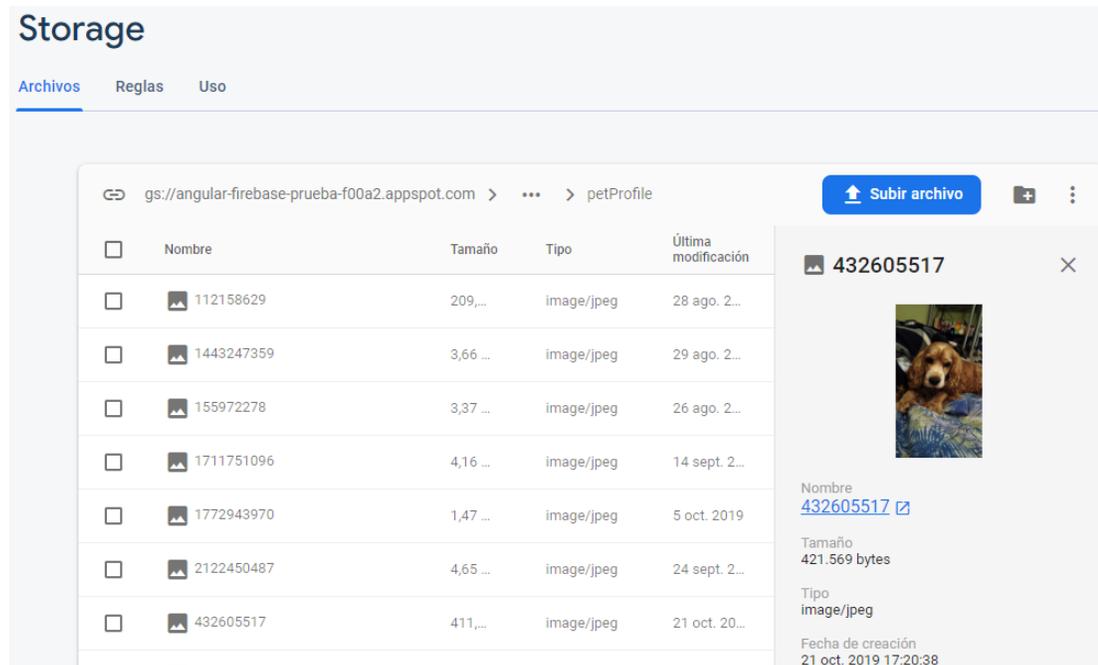


Figura 4.18: Listado de archivos que han sido almacenados en Firebase storage.

En la Figura 4.18 se observa el panel de Firebase storage, en donde se muestran archivos que han sido almacenados en la base de datos. En el listado aparece la imagen de la mascota que fue creada anteriormente, además en la parte derecha de la figura mencionada anteriormente, se muestra detalles del archivo como por ejemplo: el nombre, el formato, tamaño y la fecha de creación del mismo.

Es importante mencionar que el **segundo incremento** tiene la finalidad de entregar un software que permita al usuario crear un perfil de la mascota, en donde él ingrese información básica de la mascota además de una imagen de ésta. Además indicar que para este incremento se utilizaron los siguientes servicios de Firebase.

- Firebase realtime database.
- Firebase authentication.
- Firebase storage.

4.4.3. Incremento 3

Luego que el usuario haya creado una mascota, ahora puede crear una petición de urgencia. Para crear esta petición, el usuario debe presionar la pestaña urgencia del

Tabbed activity de la Figura 4.14. En dicha pestaña se muestra el mapa que Google Maps provee al desarrollador. El dueño de mascota puede enviar una petición de urgencia a una o varias clínicas veterinarias, que en el mapa se representan mediante marcadores o *markers*.

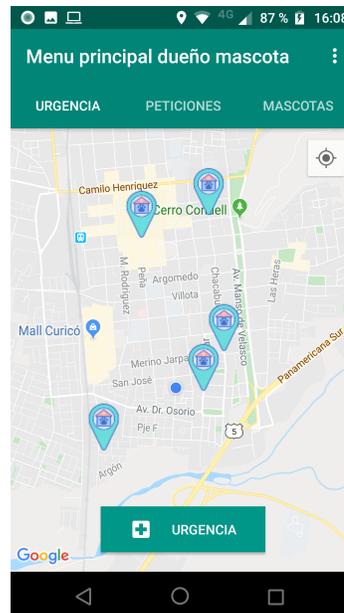


Figura 4.19: Menú para visualizar las clínicas veterinarias en el mapa

La Figura 4.19 muestra el menú que permite visualizar las clínicas veterinarias que se encuentran registradas en la aplicación. El usuario puede seleccionar una clínica veterinaria del mapa para crear la petición de urgencia, lo que permite al usuario enviar la urgencia a todos los veterinarios que trabajen en dicha clínica. También el dueño de mascota puede presionar el botón verde de urgencia, lo cual hará que la petición sea enviada a todas las clínicas veterinarias que se encuentren en la misma comuna del usuario cuando envía la urgencia.

Una vez se haya seleccionado el botón verde de urgencia, se tiene que seleccionar una mascota para la atención. La selección se realiza mediante un *FragmentManager* que muestra todas las mascotas que el usuario tiene en una lista o *ListView*. Cuando se presiona un ítem de esta lista, se abre una nueva *Activity* con un formulario para proporcionar los síntomas y otra información de la mascota seleccionada.

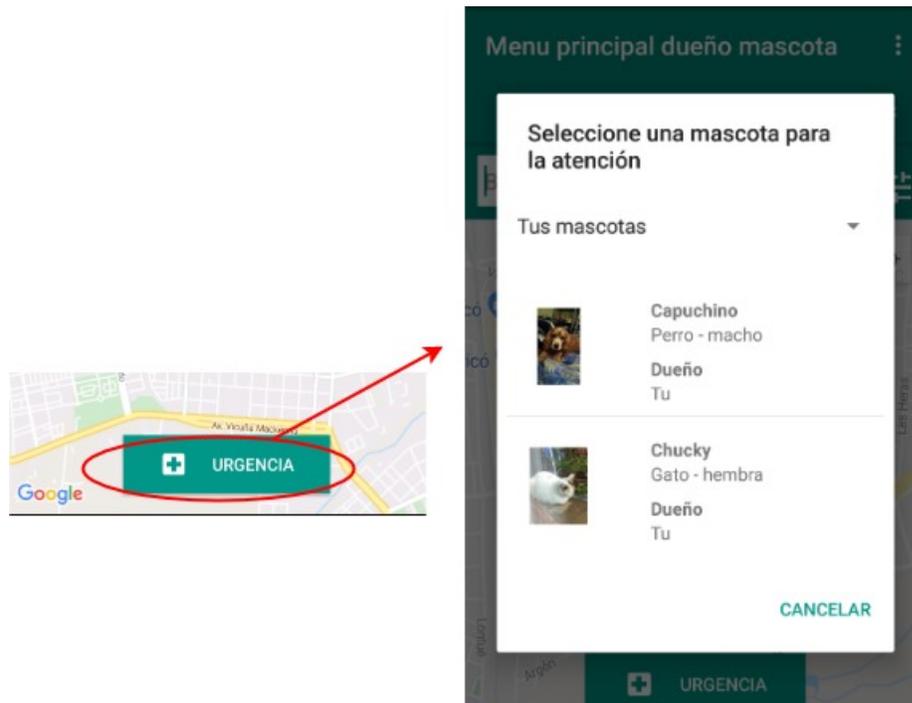


Figura 4.20: Diálogo de selección de mascota para la petición de una urgencia.

En la Figura 4.20 se muestra el diálogo de selección de mascota, que se despliega luego de presionar el botón de urgencia. La implementación de la petición de urgencia consiste en que el usuario proporcione información relevante para el veterinario a través de un formulario. A continuación se detalla la implementación de cada componente del formulario

- Un *spinner* que permite seleccionar la edad de la mascota. Cabe mencionar que las opciones de edad varían según sea el tipo de mascota. Por ejemplo, para perros se utiliza: menor a 1 año, entre 1 a 7 años y mayor a 7 años. Mientras que para gatos se implementa: menor a 2 años, de 3 a 10 años y mayor a 11 años.
- Un *spinner* que permite seleccionar el tamaño de la mascota. Las opciones implementadas son: raza pequeña, mediana y grande. Es importante decir que esta opción esta disponible solo para perros.
- Un *TextArea* o campo de texto obligatorio, que permite al usuario describir la situación de la urgencia. Este componente cuenta con un *Scrollbar* vertical, lo que permite ingresar una gran cantidad de texto.

- Un *TextArea* que permite al usuario ingresar los síntomas que presenta la mascota.
- Un *EditText* que permite al usuario proporcionar cuanto tiempo ha transcurrido desde que se presentaron los síntomas.
- Un *TextArea* que permite al usuario ingresar si ha administrado algún tipo de fármaco a la mascota y/o si ha realizado algún procedimiento previo.
- Dos botones, uno para subir una foto desde la galería, o con la cámara del dispositivo, y otro para grabar un vídeo con la cámara del dispositivo. Esta información es opcional y queda al criterio del usuario proporcionarla o no.

Atención para Capuchino

Indique edad de la mascota

Cachorro (Menor a 1 año)

Indique tamaño mascota

Pequeño

Descripción de la situación

Descripción de la urgencia

Síntomas mascota

Síntomas de la mascota

¿Cuánto tiempo ha transcurrido desde que notó los síntomas?

Ingrese cuánto tiempo aprox

¿Ha administrado algún fármaco o realizado algún procedimiento previo?

Indicar si ha hecho algo previamente

Figura 4.21: Pantalla para agregar síntomas para la petición de urgencia.

En la Figura 4.21 se muestra el formulario para que el usuario pueda agregar información mencionada anteriormente. La información que se muestra en la parte derecha de la anterior figura, es la más relevante para el médico veterinario, ya que con ella podrá tener una noción de la urgencia.

Es importante destacar que el **tercer incremento** tiene como propósito entregar un software que permita al usuario seleccionar una o varias clínicas veterinarias del

mapa a la cual irán dirigidas la petición de urgencia, además de proporcionar información de la urgencia al veterinario mediante un formulario. Además es importante mencionar que se utilizan los siguientes servicios de Google y Firebase.

- Firebase realtime database.
- Firebase authentication.
- Google Maps.

4.4.4. Incremento 4

Una vez el dueño de mascota haya proporcionado información relacionada a la descripción y síntomas de la urgencia, él puede complementar aún más la petición proporcionando una imagen y/o un vídeo de la mascota afectada. Esta información podría ser útil para el veterinario, porque puede ayudar a revelar síntomas o detalles que con información básica podría ser complejo de determinar.

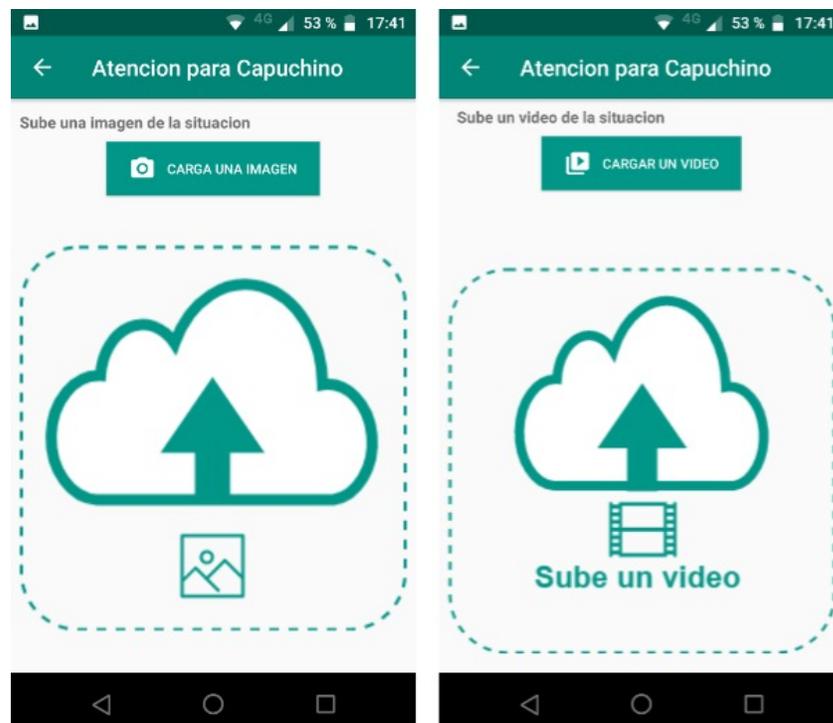


Figura 4.22: Opciones para proporcionar una imagen y video de la mascota para la urgencia.

La Figura 4.22 muestra la opción que permite subir una imagen y/o video a la petición de urgencia. Si el usuario presiona el botón verde para cargar una imagen, se abre un *Dialog* en donde el usuario selecciona si quiere cargar una imagen desde la galería del dispositivo o tomar una foto con la cámara del mismo. Si decide subir un vídeo, la aplicación solamente permite grabar uno con la cámara del teléfono y no proporcionarlo mediante la galería.

Luego de proporcionar la información necesaria para que el médico veterinario comprenda el motivo de la urgencia, el usuario puede enviar la petición de urgencia hacia las clínicas veterinarias. Es importante indicar que la petición llega a todas las clínicas veterinarias que pertenezcan a la misma comuna donde se crea la petición. Se menciona que la petición de urgencia llega a las clínicas veterinarias, pero en realidad esta petición es recibida por los veterinarios que trabajan en dichos recintos. Para que ellos reciban la petición, se debe comprobar cuales son las clínicas veterinarias que pertenecen a la comuna del dueño, para luego obtener los veterinarios que trabajan ahí y así adquirir el *token* de los dispositivos móvil para que la petición sea recibida en modo de notificación

El modo que se emplea para enviar la petición a las clínicas veterinarias es mediante un criterio de cercanía, es decir, la petición es recibida por las clínicas veterinarias que se encuentren más cerca respecto a la posición del usuario al momento de crear la petición. Es importante mencionar que por ahora, este es el único criterio de orden de envío con el que cuenta la aplicación móvil.

Algorithm 1 Compute the distance (in meters) from the pet owner position to veterinary clinic.

```

1: procedure CALCULATEDISTANCEFROMUSERTOVETCLINIC( $VC$ )
2:    $distanceArray \leftarrow \{\}$ 
3:   for each vetClinic  $vc$  in  $VC$  do
4:     if  $vc.city = userCity$  then
5:        $distance \leftarrow computeDistanceBetween(vc.position, userPosition)$ 
6:       add  $distance$  to  $distanceArray$ 
7:     end if
8:   end for
9:   return  $sortDistance(distanceArray)$     ▷ return the sorted distance array.
10: end procedure

```

El Algoritmo 1 muestra el pseudocódigo que permite calcular la distancia entre la posición del dueño de mascota y la distintas clínicas veterinarias. Para calcular

la distancia entre dos coordenadas geográficas se ocupa la librería *SphericalUtil* de Google Maps, el cual contiene un método estático que permite calcular la distancia (en metros) entre dos puntos geográficos.

```
implementation 'com.google.maps.android:android-maps-utils:0.5'
```

Figura 4.23: Sentencia para usar los métodos de *SphericalUtils*

La Figura 4.23 muestra la implementación de la librería en el archivo *build.gradle*. Luego de implementar la librería, ahora se puede usar el método que permite calcular la distancia entre 2 coordenadas geográficas.

```
static double computeDistanceBetween(LatLng from, LatLng to)  
Returns the distance between two LatLngs, in meters.
```

Figura 4.24: Firma del método que permite calcular distancia entre dos puntos geográficos.

En la Figura 4.24 se muestra la firma del método *computeDistanceBetween* que recibe 2 objetos como parámetros de tipo *LatLng*. El método retorna un dato de tipo *double* indicando la distancia en metros entre los puntos calculados. Para calcular la distancia entre el dueño, que crea la petición, y la clínica veterinaria, se necesita conocer la posición del dueño de mascota y la clínica. La posición del dueño se puede obtener mediante la *API* que Google Maps entrega. La posición de la clínica se obtiene mediante una consulta a la base de datos, ya que cuando un veterinario crea una nueva clínica veterinaria, el sistema le solicita que ubique ésta en el mapa mediante un marcador y al realizar la acción anterior, se obtiene los datos de latitud y longitud.

Para ordenar las distancias de menor a mayor que existe entre las clínicas veterinarias y el usuario, se utiliza el algoritmo de ordenamiento *insertsort*. En la línea 9 del Algoritmo 1, se invoca un método llamado *sortDistance*, éste toma las distancia obtenidas al realizar el cálculo y las ordena de menor a mayor. Con lo anterior, la aplicación envía la notificación primero a las clínicas veterinarias más cercanas al dueño de mascota.

Algorithm 2 Sorted distance array by using insertsort

```

1: procedure SORTDISTANCE(distanceArray)
2:    $n \leftarrow \text{size}(\text{distanceArray})$ 
3:   if  $n = 1$  then
4:     return distanceArray
5:   end if
6:   for  $i \leftarrow 1, n$  do
7:      $\text{currentDistance} \leftarrow \text{distanceArray}[i]$ 
8:     for  $j \leftarrow i - 1$ , down to 0 do
9:        $\text{distance} \leftarrow \text{distanceArray}[j]$ 
10:      if  $\text{currentDistance} < \text{distance}$  then
11:         $\text{exchange}(\text{distanceArray}[j], \text{distanceArray}[j + 1])$ 
12:      else
13:        break
14:      end if
15:    end for
16:  end for
17: end procedure

```

En el Algoritmo 2 se muestra el pseudocódigo del algoritmo de ordenamiento por inserción que es utilizado para ordenar las distancias de menor a mayor. *Insert Sort* es una técnica de ordenamiento que es utilizada para un número pequeño de elementos [30]. En las ciudades generalmente la cantidad de clínicas veterinarias que atienden urgencias no supera las 15, por lo tanto, se considera que esa cantidad es pequeña.

Cuando la petición es creada y enviada exitosamente, ésta se almacena en la base de datos Firebase.

```

-M26D8WG1HTODQA6CHz9
{
  "agePet": "Adulto (Entre 3 a 10 años)",
  "dateAccepted": "",
  "dateCreated": "2020-03-10 22:29:31",
  "dateSymptom": "aproximadamente hace 1 día",
  "description": "Hola buenas tardes, hace unos días mi gata tier",
  "key": "-M26D8WG1HTODQA6CHz",
  "keyImage": "-M26DL1JK1BTv77Wz3e",
  "keyPet": "-LsZnLXBhBqdJBPKnWu",
  "keySender": "yY2XmpIjRFntBtmihWPHp0CKIKr",
  "keyVeterinary": "",
  "keyVeterinaryClinic": "",
  "keyVideo": "-M26DIhegjEE23xEDda",
  "procedure": "No, no he suministrado ningún medicamento porque",
  "sizePet": "",
  "state": true,
  "symptom": "dolor\nperdida de apetito\nrespiración agitada\ndecaimp",
  "typeEnvironment": ""
}

```

Figura 4.25: Datos de la petición de urgencia almacenadas en la base de datos.

En la Figura 4.25 se muestra los datos de la petición de urgencia almacenados en la base de datos en formato *JSON*.

Cabe mencionar que el **cuarto incremento** tiene como objetivo, entregar un software que permita al dueño de mascota suministrar una imagen y/o vídeo de la mascota para la petición de urgencia, además de enviar esta petición a las clínicas veterinarias mediante un criterio de orden de envío. También mencionar que para este incremento se hace uso de las siguientes servicios.

- Firebase realtime database.
- Firebase authentication.
- Firebase storage.
- Firebase cloud messaging.
- Google Maps.

4.4.5. Incremento 5

La parte de la aplicación móvil del médico veterinario tiene dos estados: el primer estado es cuando un veterinario no está asociado a ninguna clínica veterinaria y el segundo es cuando un veterinario ya pertenece a una de las clínicas veterinarias. En esta parte sólo se aborda el funcionamiento del veterinario cuando está asociado a una clínica y recibe peticiones de urgencia. Los procesos de registro, ingreso de sesión y creación de la clínica veterinaria, se abordan en el Anexo B.2 del documento.

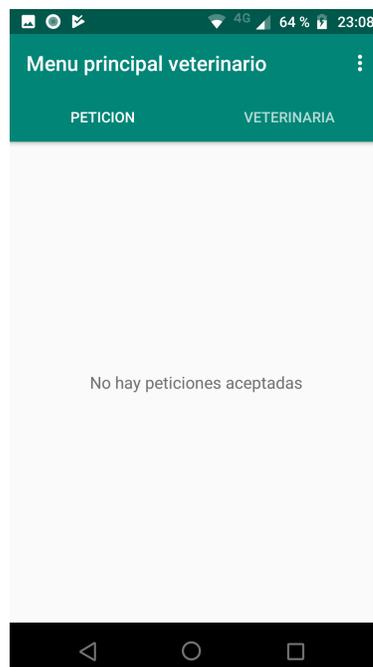


Figura 4.26: Menú principal veterinario.

En la Figura 4.26 se muestra el menú principal del veterinario. Cabe destacar que este menú principal es mostrado una vez el veterinario esté asociado a una clínica veterinaria, de lo contrario, el sistema muestra otro menú que permite asociarse a una clínica o crear una nueva. El menú principal se compone de una *tabbed activity*, en donde existen 2 pestañas o *fragments* las cuales son la pestaña de petición y veterinaria.

- La pestaña petición, es un *Fragment* que muestra un listado de peticiones que el veterinario ha aceptado. También al presionar una de éstas, puede ver el detalle.

- La pestaña veterinaria, muestra toda la información relacionada a la clínica donde trabaja.

Cuando el dueño de mascota envía una petición de urgencia, y la clínica veterinaria se encuentra dentro de la misma comuna donde se envió la petición, entonces el o los veterinarios que estén asociado a dicha clínica reciben la notificación. Es importante mencionar que como la petición llega a muchos veterinarios, el primero que acepte ésta, podrá tomar la solicitud y por lo tanto los demás veterinarios no podrán acceder a la información de la notificación.

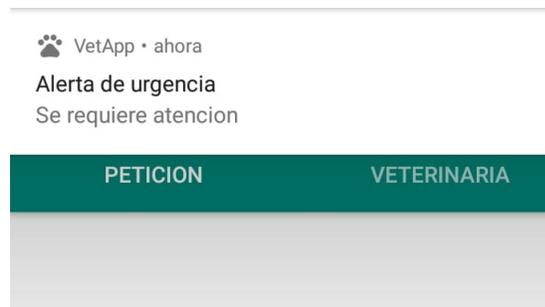


Figura 4.27: Notificación avisando al veterinario que hay una mascota que tiene una atención de urgencia.

La Figura 4.27 muestra la notificación que recibe un veterinario cuando un dueño de mascota envía una petición de urgencia. Cabe destacar que la notificación solamente la recibe el veterinario cuando la aplicación está en primer o segundo plano. Si el veterinario cierra la sesión y un dueño de mascota envía una petición, ésta no será recibida. Luego que la notificación es recibida, el veterinario puede ver, en una *activity* o pantalla, el detalle de la petición de urgencia presionando la notificación. La pantalla del detalle de la petición esta separada en 3 secciones. La primera sección muestra la información básica de la mascota en *TextView's*.

- Nombre de la mascota
- Tipo de mascota (perro o gato)
- Sexo
- Estado (castrado o esterilizada)
- Nombre del dueño

- Edad de la mascota por categoría. Las categorías de la edad varia según el tipo de mascota.
- Tamaño de la mascota. Este item sólo aplica cuando es perro.
- Una imagen de la mascota.



Figura 4.28: Información básica de la mascota en la petición de urgencia.

En la Figura 4.28 se muestra la información básica de la mascota anexada en la petición de urgencia.

La segunda sección muestra el detalle de la petición. Esta sección muestra lo siguiente:

- Fecha de envío de la petición. La fecha tiene un formato de día, mes, año y hora, minuto y segundo.
- El nombre del dueño de la mascota.
- Descripción de la situación de urgencia.
- Síntomas que presenta la mascota.

- Cuanto tiempo ha transcurrido desde la aparición de los síntomas.
- Si el dueño de la mascota suministró algún fármaco o si realizó algún procedimiento previo a la urgencia.
- Una imagen de la mascota (en caso que el dueño haya proporcionado una).
- Un vídeo de la mascota (en caso que el dueño haya grabado uno).

Detalles de la urgencia

Enviado por
Matias Pérez

Fecha envío
2020-03-01 13:24:25

Descripción de la situación
En este campo de texto sirve para que el usuario indique que paso con la mascota, que tipo de urgencia, etc

Sintomas de la mascota
No hay sintomas enviados

Tiempo transcurrido desde aparición sintomas
No hay informacion sobre cuando aparecieron los sintomas

Administracion de algun farmaco o procedimiento previo
No hay informacion de farmacos suministrados anteriormente o procedimiento realizados previamente

Figura 4.29: Información relacionada a la urgencia de la mascota en la petición de urgencia.

La Figura 4.29 muestra los detalles de la petición de urgencia. La imagen que proporciona el dueño de la mascota cuando crea la petición, es mostrada de forma más fácil usando la librería *Glide*. Cuando se guarda una imagen en Firebase, este servicio proporciona una *url* de la imagen, lo que junto a la librería facilita considerablemente la descarga y visualización de ésta.

Para aceptar o rechazar la petición de urgencia, se implementan 2 botones que se encuentra al final de la *activity*. Si el veterinario decide aceptar la petición, entonces se muestra un *dialog* que pregunta si desea realmente realizar tal acción. De lo contrario si el veterinario decide rechazar, también se pregunta a través de *dialog* si

desea realizar esa acción. Se pregunta por la realización de estas acciones, porque la decisión del veterinario es muy importante para el dueño de mascota, ya que él podría ver la confirmación e ir a la clínica, entonces la respuesta del veterinario tiene que ser bien analizada.



Figura 4.30: Opciones para aceptar o cancelar la petición de urgencia.

La Figura 4.30 muestra los botones que permiten al veterinario aceptar o rechazar la petición de urgencia. Una vez se haya aceptado la petición, ésta podrá ser vista en la pestaña petición que se muestra en la Figura 4.19, en donde el veterinario puede acceder a la información en detalle.

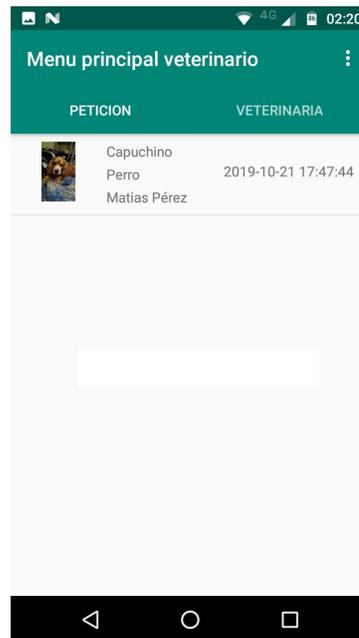


Figura 4.31: Lista de peticiones aceptadas por un veterinario.

La Figura 4.31 muestra la lista de peticiones de urgencia que han sido aceptadas por el médico veterinario. En ella aparece información de la mascota como por ejemplo: una imagen distintiva, nombre y tipo de mascota. Además del nombre del dueño y la fecha cuando fue enviada esta petición.

Es importante mencionar que el **quinto incremento** se enfoca en el veterinario, específicamente en la funcionalidad de recepción, visualización de los detalles y confirmación de la petición de urgencia que el dueño de mascota envía. Además recalcar que para este incremento se hace uso de las siguientes servicios.

- Firebase realtime database.
- Firebase authentication.
- Firebase storage.
- Firebase cloud messaging.

4.4.6. Incremento 6

Cuando un médico veterinario haya analizado y confirmado la petición, el dueño de mascota recibirá una notificación, la cual indica que la petición fue aceptada.

Para que el dueño de mascota reciba la notificación, se debe obtener el *token* de éste usuario para luego adjuntarlo a la estructura de la notificación en el campo *to* que se muestra en la Figura 4.13.

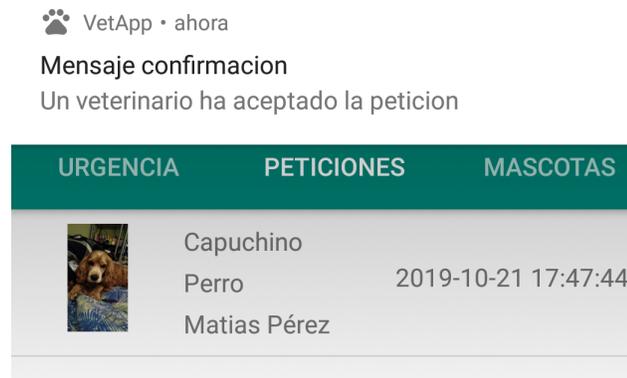


Figura 4.32: Notificación que el dueño de mascota recibe cuando un veterinario acepta la petición de urgencia.

En la Figura 4.32 se muestra la notificación que recibe el dueño de mascota cuando un médico veterinario acepta la petición de urgencia. Cuando el usuario presione la notificación, se abre una *activity* que muestra detalles de la confirmación de la petición. Esta pantalla se implementa para mostrar la siguiente información:

- Nombre de la mascota de la urgencia
- Fecha de envió de la petición
- Fecha de aceptación de la petición
- Tiempo transcurrido en minutos entre el envió y la aceptación.
- Nombre del médico veterinario que acepto la petición
- Nombre de la clínica veterinaria en donde se atenderá la mascota.
- Conjunto de servicios que ofrece la clínica veterinaria.
- Mapa donde se ubica la clínica veterinaria.

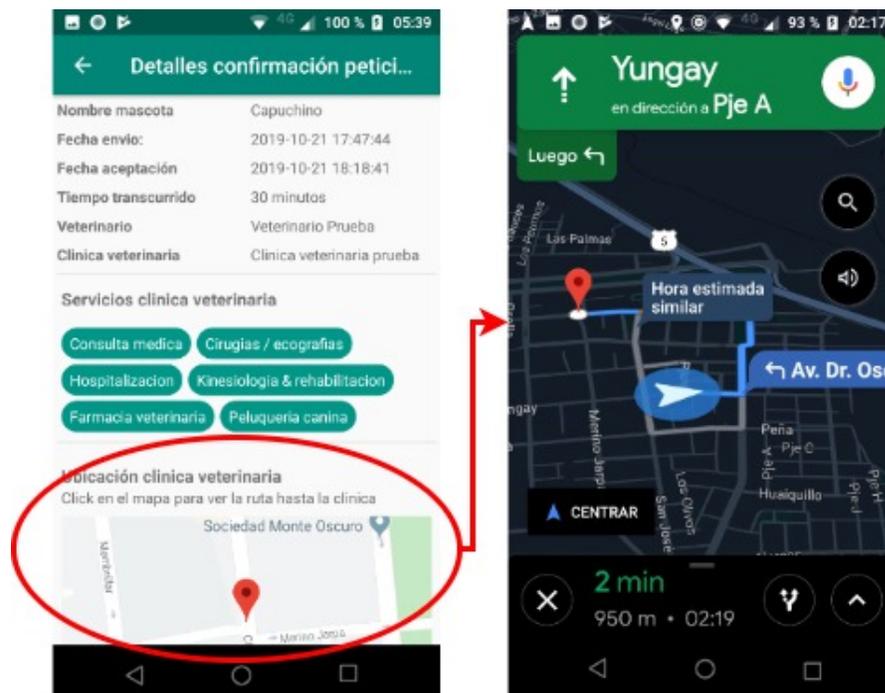


Figura 4.33: Detalles confirmación de la petición de urgencia.

La Figura 4.33 muestra detalles de la confirmación de la petición (izquierda) y la ruta hacia la clínica veterinaria (derecha). Es importante destacar que el tiempo transcurrido será la información a tomar en cuenta para ver si el objetivo del proyecto se cumple.

Los servicios que ofrece la clínica son mostrados en unos elipses llamados *Tag-View*, el cual no es un componente nativo de Android, por lo que dicha herramienta debe ser implementada mediante una librería disponible para el público.

```
implementation 'com.github.Cutta:TagView:1.3'
```

Figura 4.34: Sentencia para implementar la librería *TagView*.

La Figura 4.34 muestra la implementación de la librería en el archivo *build.gradle*. Estos tag's se ajustan automáticamente al ancho del *activity*, en caso de que el texto que contiene el tag sea lo suficientemente largo para no ajustarse correctamente a la pantalla.

El detalle de la confirmación contiene un mapa en donde se indica la ubicación de la clínica para la atención de la urgencia. Si dicho mapa es presionado por el usuario,

la aplicación abrirá Google Maps que viene instalada por defecto en los dispositivos móviles Android. Para que Google Maps muestre la ruta hacia la clínica, hay que especificar las coordenadas de ésta y el modo de ruta, que en este caso es, mediante un vehículo.

Es importante mencionar que el **sexto incremento** tiene la finalidad entregar un software que sea capaz de mostrar el mensaje de confirmación del veterinario, ver el detalle ésta y visualizar la ruta hacia la clínica veterinaria atenderá la urgencia de la mascota. Además recalcar que para este incremento se hace uso de las siguientes servicios.

- Firebase realtime database.
- Firebase authentication.
- Firebase storage.
- Firebase cloud messaging.
- Google Maps

4.4.7. Incremento 7

Cuando un veterinario recibe la notificación de urgencia por parte del dueño de mascota, ésta podría contener una imagen de la mascota que puede resultar útil al veterinario para visualizar algún detalle de la urgencia. Entonces para complementar aún más el detalle y a la imagen de la urgencia, se implementa la funcionalidad de zoom. Para usar esto, el veterinario presiona la imagen de la petición, una vez realizado lo anterior, se abre una *activity* donde él puede realizar esta acción.

```
implementation 'com.github.chrisbanes:PhotoView:2.1.3'
```

Figura 4.35: Librería para implementar la funcionalidad de zoom.

La Figura 4.35 muestra la sentencia que hay que escribir en el archivo *build.gradle* para usar la funcionalidad de zoom en Android. Esta funcionalidad es útil para el veterinario en caso que éste desee ver algún detalle de la mascota que solamente el zoom puede entregar.



Figura 4.36: Zoom realizado a la imagen de la petición de urgencia.

En la Figura 4.36 se muestra el zoom realizado a la imagen de la petición.

Es importante mencionar que el **séptimo incremento** tiene como principal tarea entregar un software que permita al veterinario hacer zoom a la imagen de la petición de urgencia. Además recalcar que para este incremento se hace uso de las siguientes servicios.

- Firebase realtime database.
- Firebase authentication.
- Firebase storage.

4.4.8. Incremento 8

Cabe mencionar que también el dueño de mascota tiene la opción de buscar una clínica por el nombre o mediante una búsqueda combinada que involucra filtrar por clínicas que se encuentren en horario de atención a público y/o servicios que ofrecen éstas.



Figura 4.37: Menú para visualizar las clínicas veterinarias en el mapa

En la Figura 4.37 se muestra el mapa con el buscador de clínicas veterinarias. Este buscador consta de un *EditText*, donde el usuario puede escribir el nombre de la clínica para luego presionar en el *ImageButton* con el ícono de lupa para iniciar la búsqueda. Además hay un último ícono que al presionarlo despliega un *DialogFragment* o simplemente un diálogo donde se puede buscar clínicas veterinarias mediante filtros combinados.

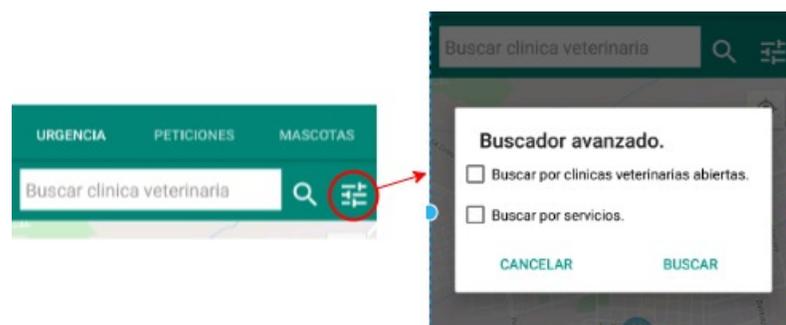


Figura 4.38: Diálogo de búsqueda de clínicas veterinarias que estén abiertas y/o que ofrezcan ciertos servicios.

La Figura 4.38 muestra el diálogo, donde el usuario puede buscar por clínicas veterinarias abiertas y/o que ofrezcan ciertos servicios que la mascota necesite para la urgencia. Para implementar la búsqueda por nombre y servicios, se hace lo siguiente:

1. Se obtiene el *string* que el usuario escribe en el componente visual *EditText* y se eliminan caracteres como por ejemplo tildes. Esta acción se realiza en caso que el usuario se olvide de colocar una tilde a una palabra, como puede ser el nombre de un servicio o el de una clínica veterinaria, y eso produzca una búsqueda que no entregue resultados.
2. Luego el *string* resultante se convierte todo a mayúscula con el método *toUpperCase()* de la clase *String* de Java.
3. Una vez el *string* esté en mayúscula y sin tildes, se compara con el nombre de la clínica veterinaria o el servicio a buscar. Para comparar se usa el método *contains()* de la clase *String*, que devuelve verdadero o falso si la palabra coincide en alguna parte con el nombre de la veterinaria o el servicio a buscar.
4. Por último se devuelve las clínicas veterinarias que coinciden con la búsqueda realizada por el usuario. De lo contrario se muestra un mensaje advirtiendo de que no hay resultados.

Una vez el usuario decida filtrar, o no, clínicas veterinarias, puede comenzar a crear la petición de urgencia presionando el botón verde de Urgencia que se muestra en la parte inferior de la Figura 4.37. Es importante mencionar que cuando el usuario presione dicho botón, la petición a crear será enviada a todas las clínicas veterinarias que hayan sido filtradas previamente por algún criterio de búsqueda que el dueño de mascota haya seleccionado. De lo contrario la petición será enviada todas las clínicas que aparecieron automáticamente en el mapa cuando el usuario seleccionó la pestaña Urgencia del *Tabbed activity*.

Es importante mencionar que el **octavo incremento** tiene como principal tarea entregar un software que permita al dueño de mascota hacer búsqueda por nombre de clínicas veterinarias, además de hacer una búsqueda mediante filtros de clínicas que se encuentren abiertas (en horario de atención de público) y/o por servicios que éstas ofrecen.

Además recalcar que para este incremento se hace uso de las siguientes servicios.

- Firebase realtime database.
- Firebase authentication.
- Firebase storage.

4.5. Pruebas

En esta sección se aborda el plan de pruebas realizado para probar las funcionalidades de la aplicación móvil. Cabe destacar que las pruebas que se tratan en esta sección son pruebas de caja negra. Además es importante mencionar que solo se muestran las pruebas a las funcionalidades más importantes de la aplicación móvil.

Test creación de una mascota	
Propósito	Corroborar que la mascota se guarda y se crea exitosamente cuando el usuario completa el formulario de creación.
Prerrequisitos	El usuario tiene que haber ingresado al sistema.
Entradas	Datos de la mascota (Nombre, tipo, genero, estado, imagen)
Salidas	Mensaje de éxito de la operación, registro creado en la base de datos y mascota mostrada en lista de mascotas del usuario.

Cuadro 4.3: Plan de pruebas de la creación de una mascota.

A modo de comprobar las salidas esperadas del Cuadro 4.3, se adjuntan las figuras 4.39 y 4.40 que demuestran los resultados obtenidos.



Figura 4.39: Mensaje de éxito de la operación junto con la mascota creada en la lista.

```

-LsZnLXBhBqdJBPKnWuz
  gender: "HEMBRA"
  key: "-LsZnLXBhBqdJBPKnWu"
  keyImage: "-LsZnLXCuCKzBUf162C"
  keyPetOwner: "yY2XmpIjRFNtBtmihWPHpOCKIKr"
  name: "Chucky"
  neutered: true
  refImage: "https://firebasestorage.googleapis.com/v0/b/anç
  type: "Gato"
    
```

Figura 4.40: Registro en formato JSON en la base de datos de la mascota creada.

Test envió petición urgencia al veterinario.	
Propósito	Comprobar que la petición es recibida por el veterinario mediante una notificación al dispositivo móvil, ya que con esto se alerta al veterinario que existe una urgencia.
Prerrequisitos	El usuario tiene que haber ingresado al sistema y tiene que haber seleccionado una mascota para la petición.
Entradas	Información de la urgencia (síntomas de la mascota, imagen y/o vídeo) y <i>token</i> del dispositivo móvil del veterinario
Salidas	Notificación de aviso.

Cuadro 4.4: Plan de pruebas del envío de una petición de urgencia que realiza un dueño de mascota.

Para verificar la salida del Cuadro 4.4, se adjunta la Figura 4.41 que muestra el resultado obtenido.

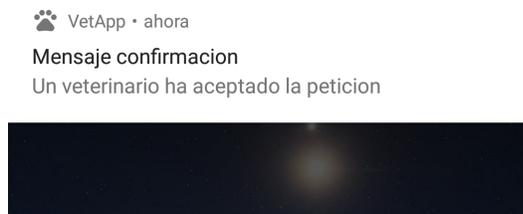


Figura 4.42: Notificación de confirmación que recibe un dueño de mascota cuando un veterinario acepta la petición



Figura 4.41: Registro en formato JSON en la base de datos de la mascota creada.

Test confirmación petición urgencia.	
Propósito	Verificar que cuando el veterinario acepta la petición de urgencia, el dueño de mascota reciba la confirmación mediante una notificación en su dispositivo móvil.
Prerrequisitos	El usuario tiene que haber ingresado al sistema y tiene que haber enviado la petición de urgencia.
Entradas	<i>token</i> del dispositivo móvil del dueño de mascota.
Salidas	Notificación de aviso y modificación del campo fecha de aceptación en la base de datos del registro petición de urgencia.

Cuadro 4.5: Plan de prueba para confirmación de una petición de urgencia.

```

-M26D8WG1HTODQA6CHz9
- agePet: "Adulto (Entre 3 a 10 años
- dateAccepted: "2020-03-10 22:31:3" ←
- dateCreated: "2020-03-10 22:29:3"
- dateSymptom: "aproximadamente hace 1 di
- description: "Hola buenas tardes, hace unos dias mi gata tier
- key: "-M26D8WG1HTODQA6CHz
- keyImage: "-M26DL1JK1BTV77Wz3e
- keyPet: "-LsZnLXBhBqdJBPKnWu
- keySender: "yY2XmpIjRFNtBtmihWPHp0CKIKr
- keyVeterinary: ""
- keyVeterinaryClinic: ""
- keyVideo: "-M26DIhegjEE23xEDda
- procedure: "No, no he suministrado ningun medicamento porqu
- sizePet: ""
- state: true
- symptom: "dolor\nperdida apetito\nrespiración agitada\ndeca:
- typeEnvironment: ""

```

Figura 4.43: Registro en formato JSON de la petición de urgencia en donde se resalta la fecha de aceptación de ésta.

4.6. Resumen Capítulo

El capítulo muestra el proceso de implementación de la metodología de desarrollo de software aplicado al proyecto. La etapa de análisis abordó como se inició con la captura de los requisitos, los cuales se clasificaron en requisitos de usuario y de sistema, además de mostrar un ejemplo de como se especifica un requisito en una tabla. La etapa de diseño abordó los diagramas a utilizar, como lo son el de arquitectura física, lógica y el diagrama de clases y de secuencia. Estos son los prerrequisitos para desarrollar la aplicación en la siguiente fase y le dan una guía al desarrollador de como se comportará el sistema. La etapa de construcción implicó mostrar la navegación del usuario dueño de mascota y veterinario. También se mostró un flujo de las funcionalidades mas relevantes del sistema con capturas de la aplicación desarrollada, para que así, se tenga un mejor entendimiento del funcionamiento de la aplicación móvil. La etapa de prueba trató sobre planes de pruebas ocupados para verificar que las funcionalidades desarrolladas retornarán las salidas esperadas.

Además un aspecto no menos importante que se presentó en este capítulo fue el de cómo se realizó la planificación de los 8 incrementos (de 2 semanas de duración cada uno) junto con sus respectivas fecha de inicio, término y las tareas que comprendía cada incremento del proyecto.

5. Resultados

El objetivo de este capítulo, es abordar los resultados obtenidos mediante el proceso de experimentación con el uso de la aplicación móvil, por parte de dueños de mascotas y veterinarios, además de mostrar los resultados arrojados por medio del proceso de encuestas realizada a los mismos usuarios, con el fin de poder realizar un análisis de estos datos, que en definitiva, darán indicios de el cumplimiento o no del objetivo general y de los objetivos específicos del proyecto.

5.1. Resultados objetivo general

En esta sección se abordan los resultados obtenidos mediante el uso de la aplicación móvil, con el fin de comprobar que el objetivo general del proyecto se cumple. La encuesta tiene como principal misión, recolectar el tiempo aproximado en que los usuarios que poseen mascotas logran encontrar una clínica veterinaria apta para atender una urgencia. Una vez obtenidos los resultados de la encuesta, se comparan con los tiempos obtenidos mediante la experimentación con la aplicación móvil construida. Cabe mencionar que algunos resultados entregados por los usuarios de la encuesta se encuentran en un rango de tiempo, entonces para calcular un solo valor de esta medición, se calculó el promedio como tiempo aproximado.

Para saber cual es el tiempo que emplean los dueños de mascota en buscar una clínica veterinaria apta para una urgencia, se ha realizado la siguiente pregunta.

- Aproximadamente ¿Cuánto tiempo cree usted que se demoraría en encontrar una clínica veterinaria que pueda atender la urgencia de su mascota?.

A continuación se exponen algunos resultados individuales relacionados a las pruebas realizadas en el proceso de experimentación para comprobar que el objetivo general del proyecto se cumple.

Experimentación con el dueño de mascota n°1	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 30 minutos
Nombre mascota	Rayas
Veterinario	veterinario de prueba
Tiempo de búsqueda sin app	30 minutos
Tiempo de búsqueda con app	18 minutos
Resultado prueba	✓ Se cumple el objetivo con una reducción de 12 minutos.

Cuadro 5.1: Prueba de experimentación con el dueño de mascota 1.

Experimentación con el dueño de mascota n°2	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 8 minutos.
Nombre mascota	Blanca
Veterinario	veterinario de prueba
Tiempo de búsqueda sin app	8 minutos
Tiempo de búsqueda con app	32 minutos
Resultado prueba	✗ No se cumple el objetivo, ya que hubo un aumento en los tiempos de 24 de minutos con la app.

Cuadro 5.2: Prueba de experimentación con el dueño de mascota 2.

Experimentación con el dueño de mascota n°3	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 30 minutos.
Nombre mascota	Cindy
Veterinario	veterinario de prueba
Tiempo de búsqueda sin app	30 minutos
Tiempo de búsqueda con app	25 minutos
Resultado prueba	✓ Se cumple el objetivo, ya que hubo un ahorro de 5 minutos en los tiempos de búsqueda con la app.

Cuadro 5.3: Prueba de experimentación con el dueño de mascota 3.

Experimentación con el dueño de mascota n°4	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 60 minutos.
Nombre mascota	Tacha
Veterinario	veterinario de prueba
Tiempo de búsqueda sin app	60 minutos
Tiempo de búsqueda con app	22 minutos
Resultado prueba	✓ Se cumple el objetivo, ya que hubo un ahorro de 38 minutos en los tiempos de búsqueda con la app.

Cuadro 5.4: Prueba de experimentación con el dueño de mascota 4.

Los experimentos restantes, se exponen en el Anexo A del documento. A continuación se comparan los tiempos reunidos a través de la encuesta junto con resultados obtenidos mediante el uso de la aplicación móvil, para luego analizar e interpretar los resultados.

Tiempo búsqueda sin uso el aplicación móvil (minutos)	Tiempo búsqueda con el uso aplicación móvil (minutos)	Reducción de tiempo (minutos)
30	18	↓ 12
8	32	↑ 24
30	25	↓ 5
60	22	↓ 38
60	28	↓ 32
20	38	↑ 18
20	8	↓ 12
20	13	↓ 7
17	15	↓ 2
40	23	↓ 18
60	17	↓ 43
37	12	↓ 25

Cuadro 5.5: Tabla de comparaciones de tiempos obtenidos en la encuesta y con la aplicación móvil.

El Cuadro A.8 muestra la comparación de tiempo que existe cuando se busca una clínica veterinaria apta para una urgencia, primero sin el uso de la aplicación y después con el uso de ésta, además en la tabla las flechas verdes indica una reducción de tiempo y las rojas un aumento de éste. De la tabla se desprende que el 84% de los dueños de mascotas, equivalente a 10 de ellos, tuvieron una reducción de tiempo cuando usaron la aplicación móvil para encontrar una clínica veterinaria apta para atender una urgencia.

Como la encuesta aplicada para tomar los tiempos de búsqueda sin el uso de la aplicación no fue dirigida a un grupo etario en específico, si no que fue dirigida a un público general, algunos tiempos que se muestran en la Figura A.8 pueden parecer atípicos (tiempos muy bajos o tiempos muy altos), esto puede deberse a los siguientes factores:

- La persona quizás no tiene experiencia buscando clínica veterinarias usando alguna herramienta tecnológica.
- La persona tiene un contacto rápido con una clínica veterinaria.

- La persona nunca ha tenido un urgencia veterinaria.
- Un caso extremo, donde la persona encuestada se colocó en un contexto de búsqueda en un fin de semana o un día festivo.

Todos los factores mencionados anteriormente pueden afectar la respuesta del encuestado.

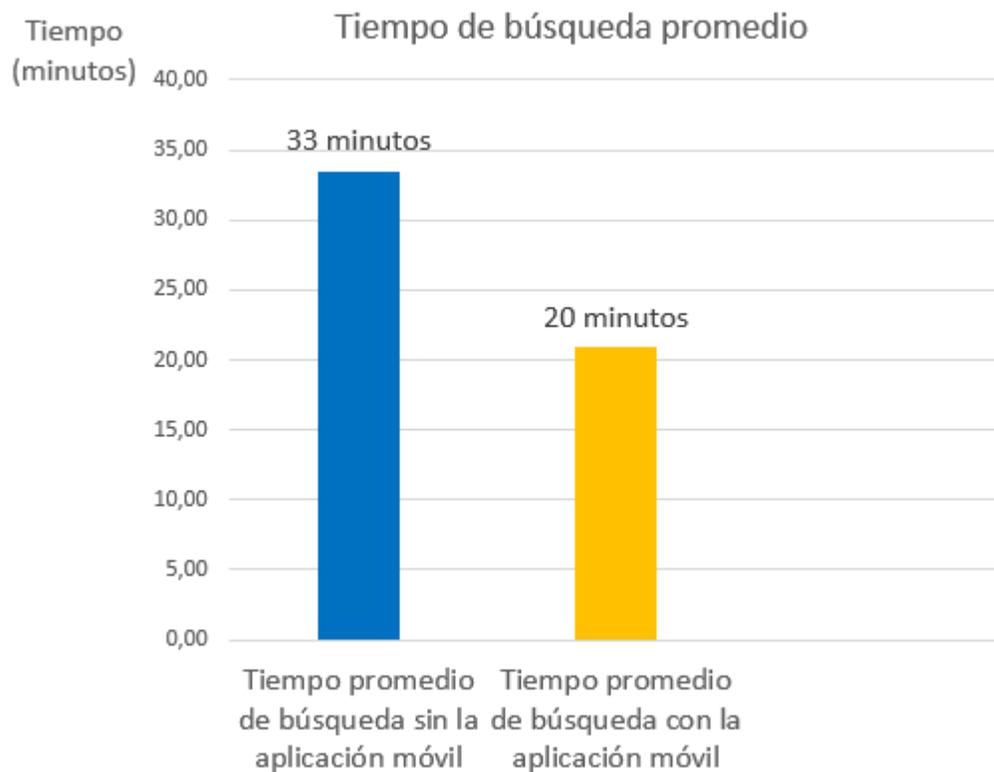


Figura 5.1: Tiempo promedio de búsqueda sin y con el uso de la aplicación móvil.

La Figura 5.1 muestra los tiempos de búsqueda promedio de clínica veterinarias para una urgencia con y sin el uso de la aplicación móvil. Del gráfico se desprende, que el promedio de búsqueda con el uso de la aplicación disminuye en 13 minutos, es decir, una reducción promedio equivalente a un 39 %.

5.2. Resultados objetivos específicos

Esta sección se analizan los resultados obtenidos en la encuesta aplicada a dueños de mascota y veterinarios, con el propósito de verificar que los objetivos específicos

del proyecto se cumplen. Es importante mencionar que la encuesta fue aplicada a 12 dueños de mascotas y solamente a 4 veterinarios.

Como el proyecto involucra a dos tipos de usuarios, se aplica una encuesta para cada uno. Además mencionar que el mecanismo para recolectar los datos es mediante una escala Likert de 1 a 5, donde 1 representa *Muy en desacuerdo* y 5 *Muy de acuerdo*. La encuesta del dueño de mascota consta de 3 secciones, donde cada una hace referencia a un objetivo específico. La encuesta del veterinario funciona de la misma manera, pero con menos secciones, ya que algunos objetivos específicos van dirigido a un solo tipo de usuario.

5.2.1. Comprobación del objetivo específico n°1

El primer objetivo específico del proyecto está definido como:

- Construir un entorno que permita entregar fácilmente información de la mascota a la clínica veterinaria cuando ocurra una urgencia.

Para comprobar el objetivo recién mencionado se realizó una encuesta a dueños de mascotas que utilizaron la aplicación móvil. A continuación se muestran los resultados.

¿Le ha parecido fácil encontrar el menú para crear una mascota en la aplicación móvil?

12 respuestas

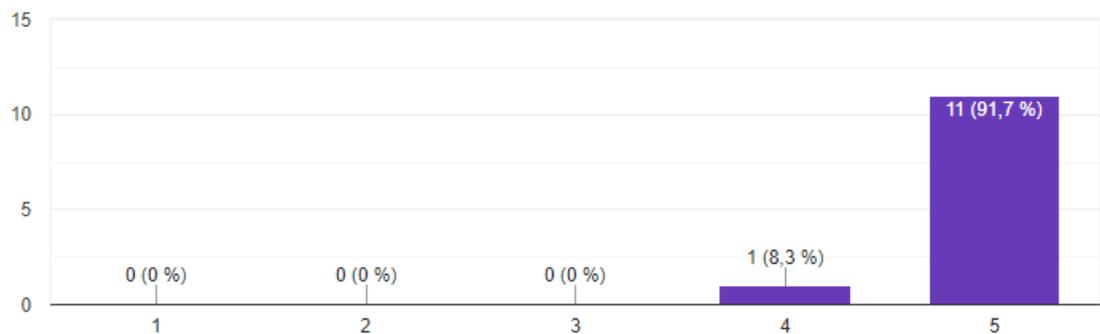


Figura 5.2: Gráfico que muestra los resultados sobre la facilidad que tiene el dueño de mascota de crear una mascota en la aplicación móvil.

La Figura 5.2 muestra que 11 dueños de mascotas, equivalentes a casi 92% del total de encuestas, respondieron que estaban muy de acuerdo de que el menú para

crear una mascota es fácil de encontrar en la aplicación móvil. Mientras que 1 persona estuvo solamente de acuerdo.

¿La información relacionada a la mascota es fácil de agregar a la aplicación móvil?



12 respuestas

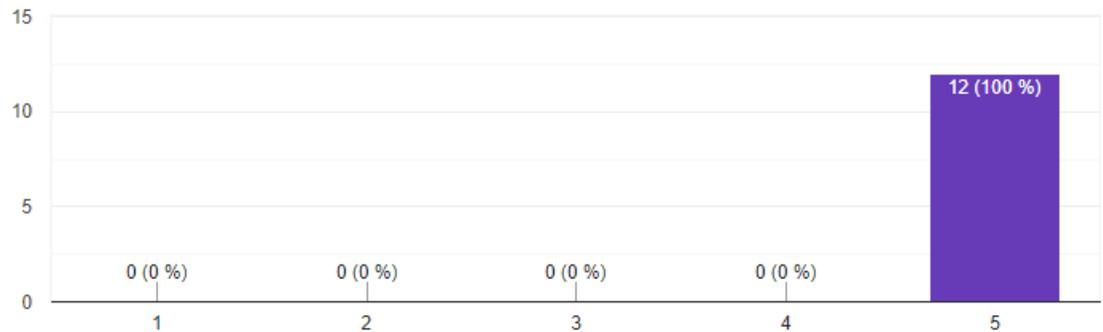


Figura 5.3: Gráfico que muestra los resultados sobre la facilidad que tiene el dueño de ingresar la información de la mascota en la aplicación móvil.

La Figura 5.3 muestra que 12 dueños de mascotas, equivalentes al total de los encuestados, respondieron que estaban muy de acuerdo de que el formulario construido para ingresar la información básica de la mascota resulta sencillo.

¿La cantidad de información solicitada para agregar una mascota a la aplicación es adecuada?

12 respuestas

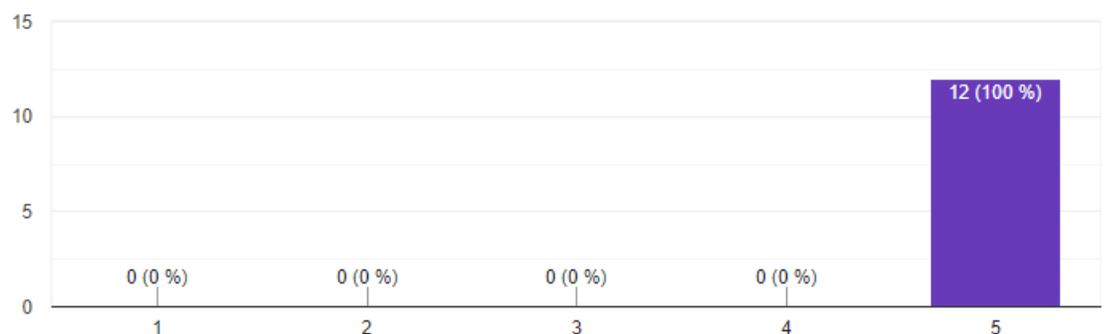


Figura 5.4: Gráfico que muestra los resultados sobre la facilidad que tiene el dueño de mascota de crear una mascota en la aplicación móvil.

La Figura 5.4 muestra que 12 dueños de mascotas, equivalentes al total de los

encuestados, respondieron que estaban muy de acuerdo de que no resulta engorroso ingresar la información básica de la mascota al momento de crearla en la aplicación.

¿La ha parecido fácil seleccionar una mascota para crear una urgencia?

12 respuestas

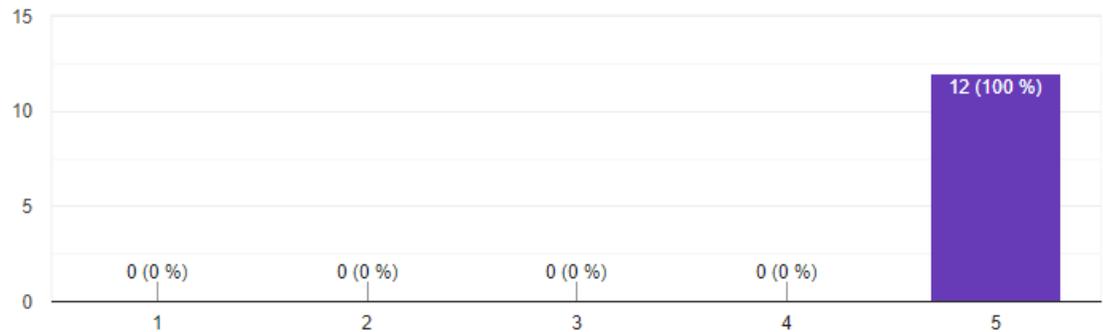


Figura 5.5: Gráfico que muestra los resultados respecto a la facilidad que tiene el usuario en seleccionar una mascota para la urgencia.

En la Figura 5.11 se muestra que todos los dueños de mascotas encuestados, estuvieron muy de acuerdo en que, es fácil seleccionar una mascota para la crear la urgencia.

¿La forma de proporcionar los síntomas de la mascota es de su agrado?

12 respuestas

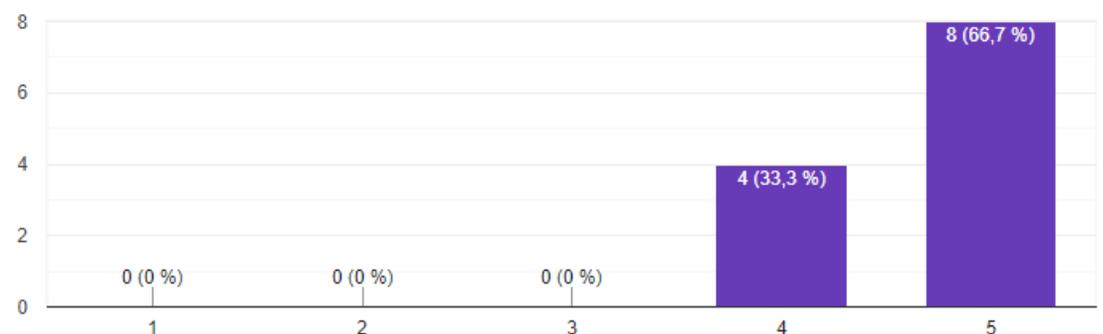


Figura 5.6: Gráfico que muestra los resultados en relación al agrado que tiene el usuario al proporcionar información de los síntomas de una mascota.

La Figura 5.12 releva que el aproximadamente el 66 % de las personas encuestadas

estuvo muy de acuerdo sobre la forma de proporcionar los síntomas de la mascota en la aplicación. El 33 % restante de los encuestados solamente estuvo de acuerdo.

¿La aplicación le ha permitido ingresar una imagen de la mascota al momento de crear una urgencia?

12 respuestas

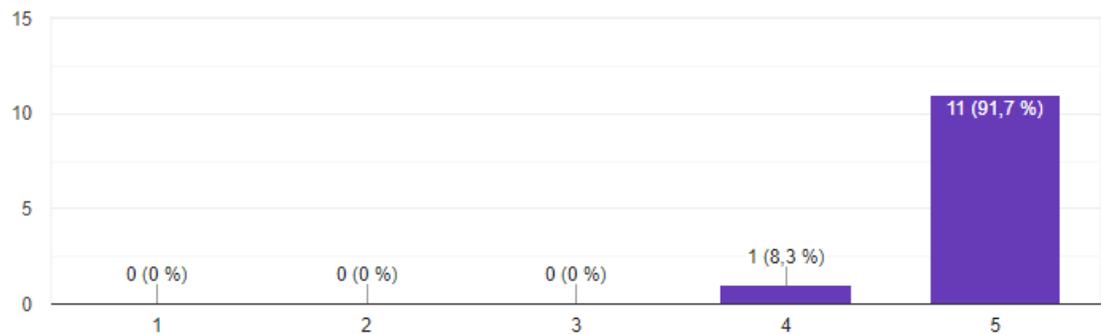


Figura 5.7: Gráfico que muestra los resultados respecto a si la aplicación móvil permite ingresar una imagen de la mascota al momento de crear una urgencia.

En la Figura 5.13 se observa que la mayoría de los dueños de mascotas encuestados, respondieron que estaban muy de acuerdo en relación a la facultad que entrega la aplicación de proporcionar una imagen de la mascota, al momento de crear una urgencia.

¿La aplicación le ha permitido proporcionar un vídeo de la mascota al momento de crear una urgencia?

12 respuestas

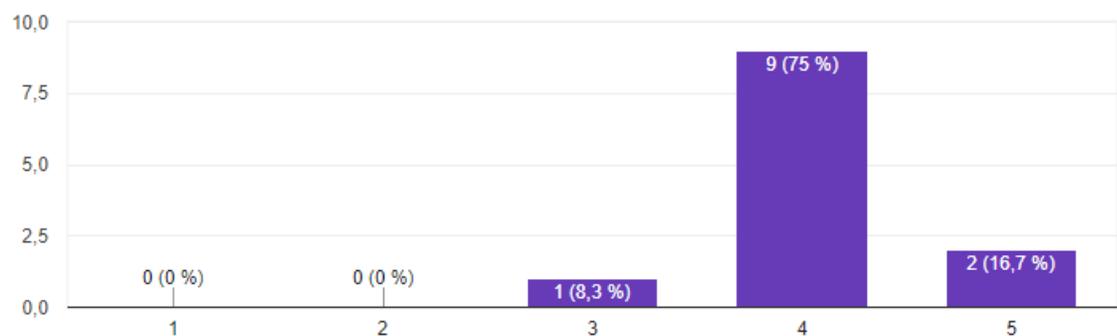


Figura 5.8: Gráfico de los resultados con respecto a si la aplicación móvil permite ingresar un vídeo de la mascota al momento de crear una urgencia.

La Figura 5.8 muestra que un 75 % de los encuestados, estuvo solamente de acuerdo sobre si la aplicación permitía subir un vídeo al momento de crear la urgencia. El restante 8 % y 16 % estuvo neutral y muy de acuerdo respectivamente.

Para corroborar aún más los resultados se muestra los resultados de la encuesta aplicado a los veterinarios.

¿La cantidad de información básica sobre la mascota es suficiente para la urgencia?

4 respuestas

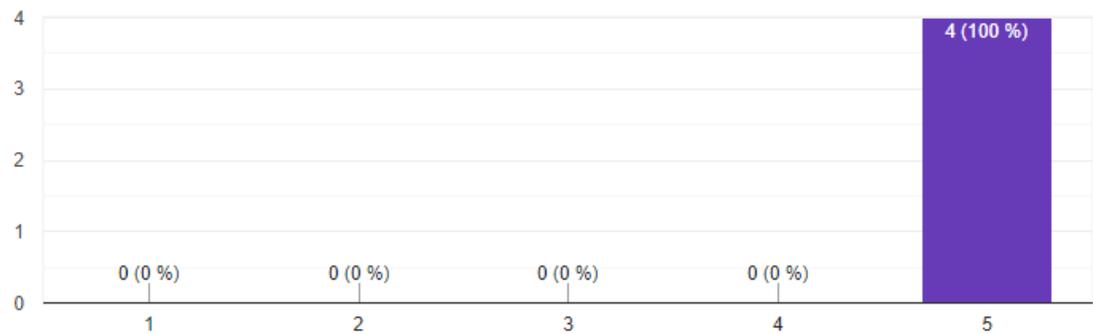


Figura 5.9: Gráfico que muestra los resultados sobre cantidad de información que proporciona el dueño de mascota ayuda al veterinario en la urgencia.

La Figura 5.9 revela que del total de veterinarios que respondieron la encuesta, todos están muy de acuerdo de que la cantidad de información básica de la mascota, es apta para comprender la urgencia.

¿La información de los síntomas de la mascota es suficiente para comprender la urgencia?

4 respuestas

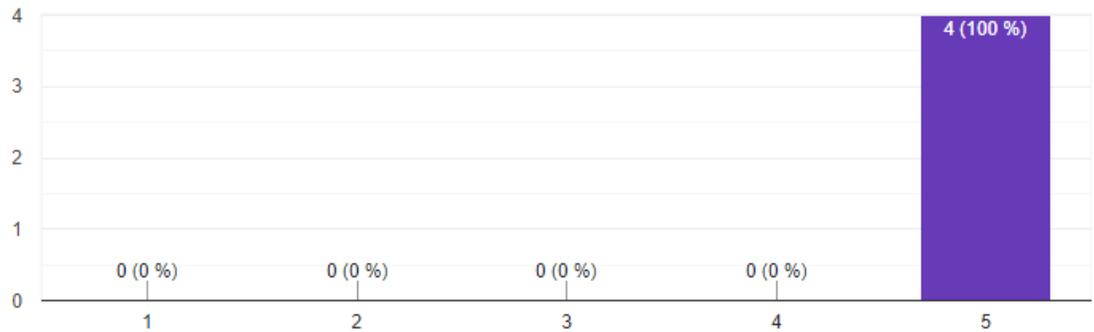


Figura 5.10: Gráfico que muestra los resultados obtenidos al consultar a los veterinarios sobre la información relacionada a la urgencia de la mascota.

La Figura 5.9 muestra que todos veterinarios que usaron la aplicación móvil estuvieron muy de acuerdo a que los campos que proporciona la aplicación, son de ayuda para comprender la urgencia de la mascota.

5.2.2. Comprobación del objetivo específico n°2

La segunda sección de la encuesta está enfocada en comprobar el siguiente objetivo:

- Establecer una metodología que permita la interacción entre veterinarios y dueños de mascota.

El sistema de notificaciones le ha servido para avisar de que un veterinario ha aceptado la urgencia?

12 respuestas

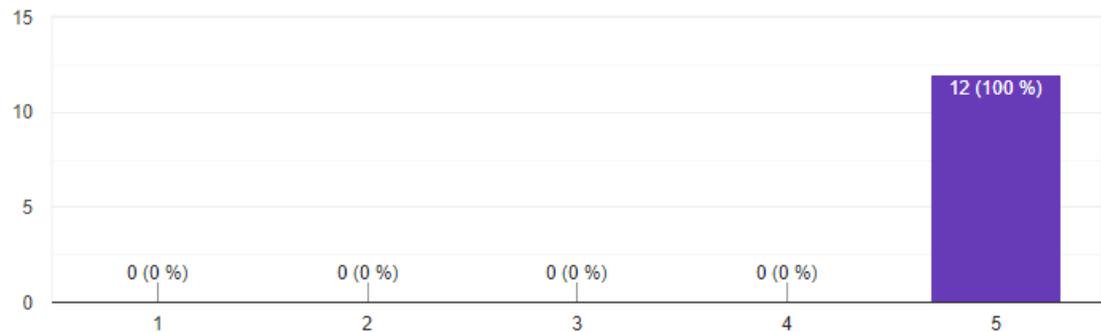


Figura 5.11: Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascota si el sistema de notificaciones ha servido para avisar a un veterinario.

La Figura 5.11 muestra que el total de encuestados estuvo muy de acuerdo a que el sistema de notificación, si sirve para avisar al veterinario que hay una mascota que necesita atención.

Cuando la aplicación le notifica que un veterinario ha tomado la urgencia, el mensaje de la notificación es fácil de entender?

12 respuestas

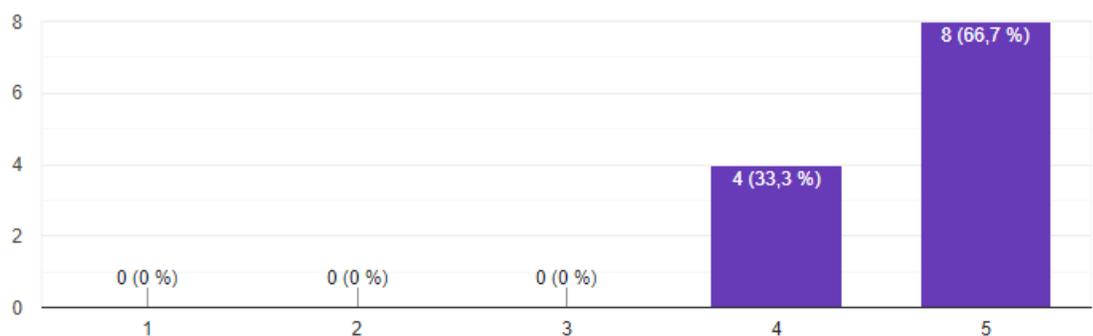


Figura 5.12: Gráfico que muestra los resultados obtenidos al consultar a dueños de mascota sobre la claridad del mensaje de confirmación.

La Figura 5.12 muestra que más de la mitad de los dueños de mascotas encuestados, es decir, un 66% estuvo muy de acuerdo sobre el mensaje de confirmación que

envía el veterinario es fácil de comprender. El resto de encuestados estuvo solamente de acuerdo.

¿Encuentra útil este tipo de comunicación con el veterinario para una urgencia?

12 respuestas

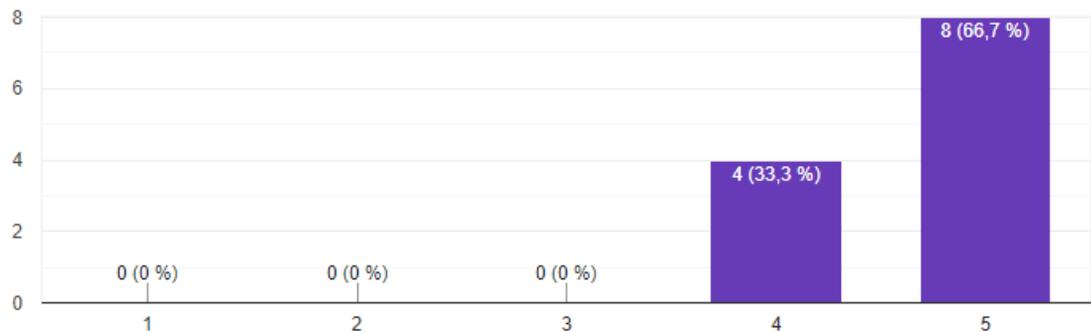


Figura 5.13: Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascota sobre la utilidad de este tipo de comunicación con el veterinario.

La Figura 5.13 muestra que el 66 % de los encuestados, estuvo muy de acuerdo con este tipo de interacción con el veterinario, mientras que el restante 33 %, solo se mostró de acuerdo.

La notificación cuando un veterinario confirma la urgencia, ¿Le indica de forma clara hacia que clínica veterinaria debe dirigirse?

12 respuestas

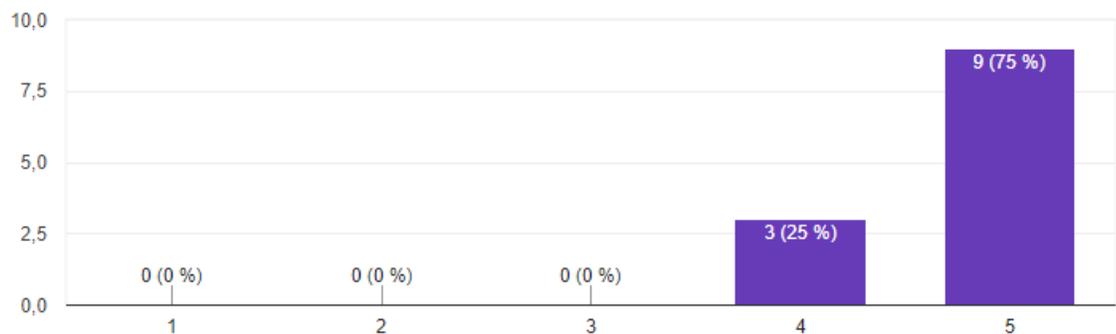


Figura 5.14: Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascota sobre las indicaciones que entrega la aplicación para dirigirse a la clínica veterinaria que atenderá la urgencia.

La Figura 5.14 revela que el 75 % de los dueños de mascota se mostró muy de acuerdo de que la aplicación indica de forma clara a que clínica hay que dirigirse cuando el veterinario acepta la urgencia. El 25 % restante, se mostró solamente de acuerdo.

Para corroborar aún más los resultados se muestra los resultados de la encuesta aplicado a los veterinarios.

¿El sistema de notificaciones le ha avisado de que una mascota tiene una urgencia?

4 respuestas

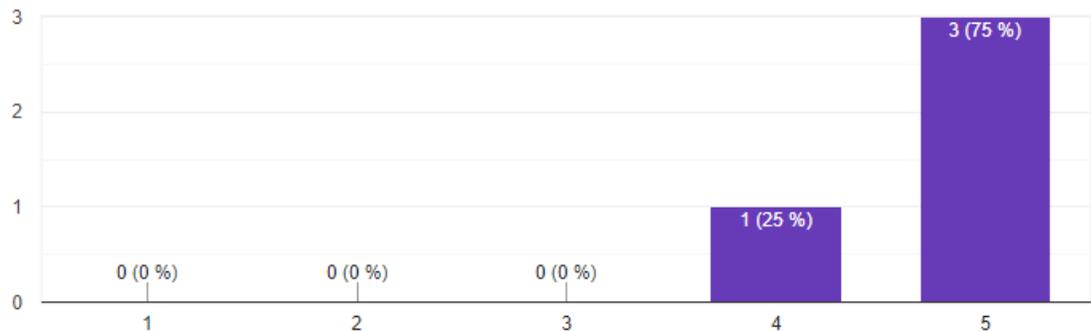


Figura 5.15: Gráfico que muestra los resultados obtenidos al consultar a los veterinarios sobre si el sistema de notificaciones es útil para avisar si una mascota tiene una urgencia.

La Figura 5.15 muestra que el 75 % de los encuestados se mostró muy de acuerdo sobre el funcionamiento del sistema de notificaciones para avisar que hay una mascota necesita una atención. El resto de encuestados solamente estuvo de acuerdo.

Cuando la aplicación le notifica que una mascota tiene urgencia, el mensaje de la notificación es fácil de entender?

4 respuestas

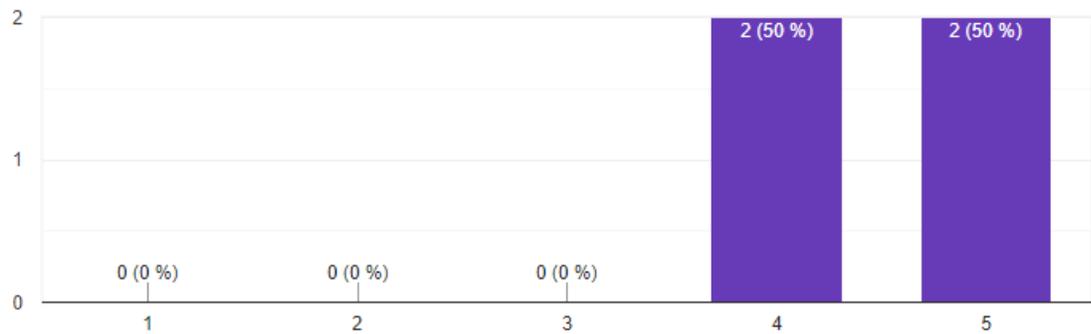


Figura 5.16: Gráfico que muestra los resultados obtenidos al consultar a los veterinarios sobre la claridad del mensaje de la notificación cuando el dueño avisa de la urgencia.

La Figura 5.16 muestra que la mitad de los veterinarios encuestados estuvo muy de acuerdo sobre la claridad del mensaje de aviso por parte del dueño de mascota. La otra mitad se mostró de acuerdo solamente.

Encuentra útil esta forma de avisar al dueño de mascota de que hay un veterinario dispuesto a atender la urgencia

4 respuestas

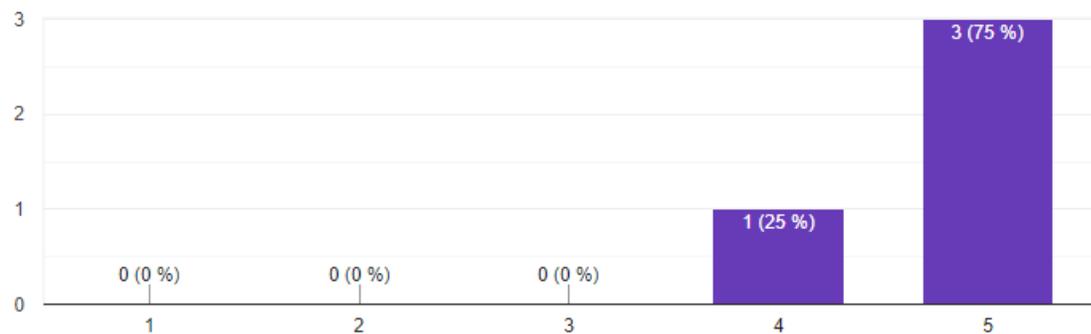


Figura 5.17: Gráfico que muestra los resultados obtenidos al consultar a los veterinarios sobre la utilidad de este tipo de comunicación con el dueño de mascota.

La Figura 5.17 revela que el 75 % de los veterinarios encuestados, encuentra muy

útil este tipo de comunicación con el dueño de mascota cuando ocurre una urgencia. El porcentaje restante solo se muestra de acuerdo.

5.2.3. Comprobación del objetivo específico n°3

La tercera sección de la encuesta tiene como fin comprobar el tercer objetivo específico, el cual se define de la siguiente manera:

- Visualizar la ubicación de las clínicas veterinarias.

¿El mapa le ha servido para saber donde se encuentran las clínicas veterinarias?

12 respuestas

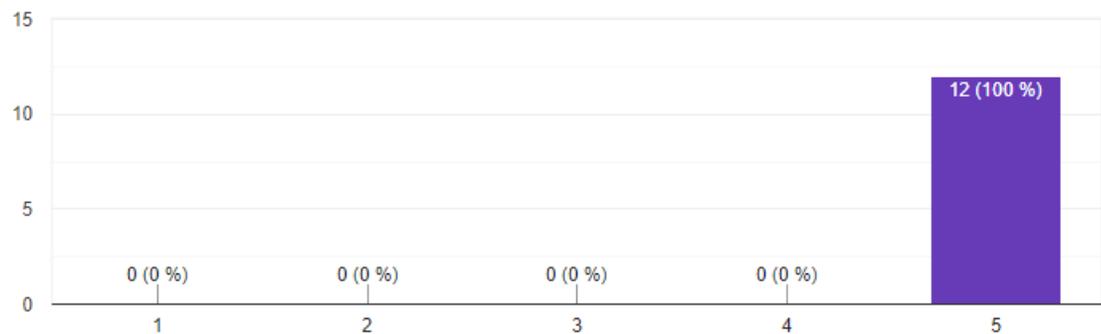


Figura 5.18: Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascotas sobre si la aplicación sirve para indicar donde se encuentran las clínicas veterinarias.

La Figura 5.18 muestra que el 100% de los dueños de mascotas encuestados estuvieron muy de acuerdo sobre esta funcionalidad.

¿La información que muestra el marcador le ha sido de utilidad?

12 respuestas

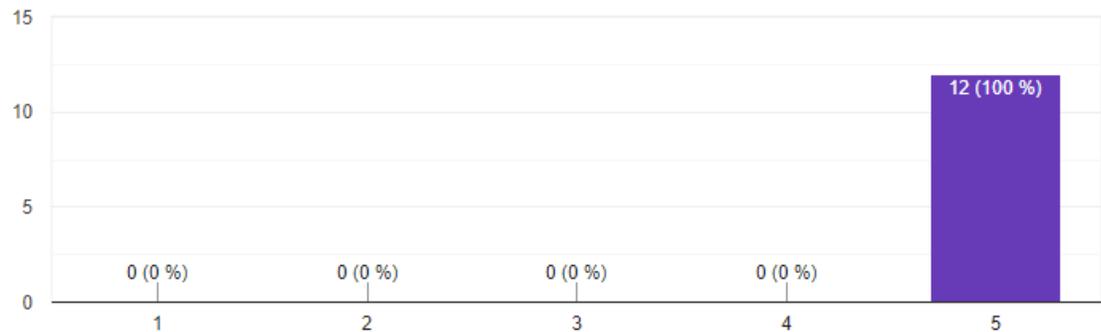


Figura 5.19: Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascota sobre la utilidad de la información que muestra un marcador.

La Figura 5.19 muestra que el 100% de los dueños de mascotas encuestados estuvieron muy de acuerdo sobre esta funcionalidad.

¿La información en detalle de la clínica veterinaria le ha sido de utilidad?

12 respuestas

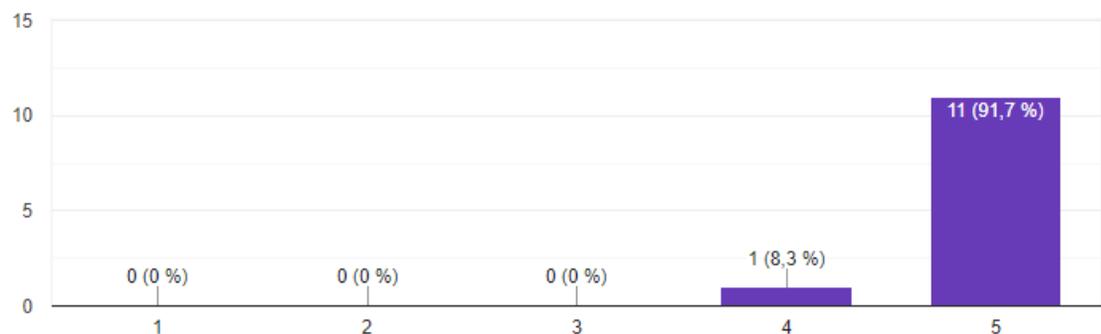


Figura 5.20: Gráfico que muestra los resultados obtenidos al consultar a los dueños de mascota sobre la utilidad de la información en detalle de una clínica veterinaria.

En la Figura 5.20 se muestra que casi la totalidad de encuestados, es decir, un 91% de dueños de mascotas, estuvo muy de acuerdo sobre la utilidad de la información detallada de la clínica veterinaria. Mientras tanto el porcentaje restante de encuestado se mostró de acuerdo solamente.

5.3. Resumen Capítulo

En este capítulo se hace un análisis de los tiempos promedio de búsqueda de clínicas veterinarias para una urgencia, obtenidos sin el uso de la aplicación móvil y después haciendo uso de ésta. Para luego hacer una comparación entre estos tiempo y comprobar si hubo una reducción de ellos.

Por otro lado se hace una revisión a los datos obtenidos por medio de las encuestas aplicada a dueños de mascotas y veterinarios, con el fin de realizar un análisis de estos y comprobar si las funcionalidades más importante de la aplicación funcionan de forma correcta y ayudan a ambos usuarios con sus propósitos.

6. Conclusión y trabajos futuros

6.1. Conclusión general

Para concluir el trabajo realizado es importante mencionar el objetivo general del proyecto

- Reducir el tiempo de búsqueda de una clínica veterinaria apta para la atención de urgencia de una mascota, a través de una aplicación móvil.

Analizando los resultados obtenidos mediante el proceso de experimentación con la aplicación móvil, se puede comprobar que el tiempo promedio de búsqueda se redujo en un 40%, si se lleva ese porcentaje a una medida de tiempo, equivale a una reducción de 13 minutos en la búsqueda de una clínica apta para una urgencia veterinaria, también cabe destacar que el uso constante de esta aplicación puede reducir aún más los tiempos. Además del total de dueños de mascota participantes en el proceso de experimentación, un 84% de ellos tuvieron una reducción en sus tiempos gracias al uso de esta aplicación, y esto quiere decir, el proyecto cumple con su objetivo general. Si bien la cantidad de dueños de mascota y veterinarios que participaron en este proceso pudo ser aún mayor, los resultados expuesto dan un indicio que esta herramienta puede ser de ayuda en el área de atención veterinaria.

Por otro lado analizando los resultados de la encuesta aplicada a los dueños de mascota y veterinarios, se comprueba que los objetivos específicos del proyecto se cumplen, ya que ambos usuarios tuvieron una buena recepción sobre la facilidad y funcionalidad de la aplicación móvil.

La metodología iterativa incremental es una buena opción de desarrollo para este tipo de proyectos, donde al principio no se tiene una concepción clara de lo que se

quiere hacer. Además muchas veces hay retrasos en el desarrollo de ciertas funcionalidades, pero este tipo de metodología de desarrollo, es flexible en para esos casos. Además escoger este tipo de metodología que conllevan un más poco de documentación, más que *Scrum* por ejemplo, es útil para recordar algunos conceptos de diseño que son fundamentales en la ingeniería de software como es el desarrollo de diagrama de clases, de secuencia, arquitectura, entre otros. Ya que en realidad son estos conceptos los que diferencian a un ingeniero de un programador como tal.

Por último mencionar, que el desarrollo de este proyecto es muy útil para mi formación académica, ya que fue una oportunidad para aprender nuevas tecnologías como lo es el desarrollo móvil en Android y usar los servicios de Firebase. Y que en un futuro podrían ser de utilidad. También el hecho de interactuar con personas que no son del área de computación, ayuda a la parte de formación profesional, ya que muchas veces se tiene que trabajar en un ambiente donde hay poca claridad de lo que se quiere realizar.

6.2. Trabajos futuros

La aplicación móvil fue construida pensando solo en atenciones de urgencias de perros y gatos, pero dado el contexto de la aplicación, puede ser aplicada a diferentes áreas de la medicina veterinaria. A continuación se nombran trabajos o funcionalidades futuras del proyecto.

- Compatibilidad con iOS.
- Añadir ingreso de sesión con Google y Facebook.
- Implementar calificación de una clínica veterinaria y de veterinarios.
- Desarrollar un chat entre dueño de mascotas y veterinarios.
- Considerar el manejo de un módulo de pago en la aplicación móvil.
- Contemplar un módulo de gestión de citas en una clínica veterinaria.
- Desarrollar un módulo de historial clínico de una mascota.

- Considerar la atención de urgencia de otro tipo de mascotas, mediante el proceso de parametrización de especies e información relevante para veterinario según el tipo de éstas.
- Agregar un usuario de tipo secretaria(o) o ATC (Auxiliar técnico veterinario), para que se encargue de recepcionar las urgencias y luego envíe la información de la situación a un médico veterinario.

Glosario

Veterinario: Persona que se dedica a curar y prevenir las enfermedades de los animales.

Urgencia veterinaria: Son aquellos accidentes o enfermedades de las mascotas y estas pueden ocurrir cuando menos se espera y generalmente necesitan de la atención de médico veterinario.

Adimark: Es una empresa de investigación de mercado y opinión pública.

SDK: Software Development Kit, es generalmente un conjunto de herramientas de desarrollo de software que permite a un desarrollador de software crear una aplicación informática para un sistema concreto.

IDE: Integrated Development Environment, es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

API: Application Programming Interface, es un conjunto funciones, métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

MVC: Modelo-Vista-Controlador, es un patrón de arquitectura de software que tiene como propósito separar la lógica de negocio de la interfaz del usuario.

Token: Es un identificador que permite a un usuario o sistema tener acceso a ciertos recursos de una aplicación.

JSON: JavaScript Object Notation.

Activity: Es una pantalla o vista que compone una aplicación móvil en Android.

Tabbed Activity: Es una pantalla o vista que se compone de pestañas.

Fragment: Es una porción de interfaz gráfica que puede componer una activity.

Spinner: Es una lista de desplegable que permite al usuario seleccionar una opción.

EditText: Es un campo de texto que sirve para ingresar información como por ejemplo: nombre, fecha, correo, etc.

TextArea: Es un campo de texto extenso, que generalmente se ocupa para ingresar una cantidad de texto considerable en los formularios.

ImageButton: Es un botón que tiene forma de icono.

Bibliografía

- [1] MicroEstudio GfK. https://www.gfk.com/fileadmin/user_upload/country_one_pager/-CL/GfK_Los_chilenos_y_sus_mascotas.pdf. Consultado: 2019-06-19.
- [2] Vetmapp - la major manera de encontrar clínicas veterinarias. <http://www.vetmapp.com/es/>. Consultado: 2019-04-27.
- [3] Alert-messaging. <https://www.revolv.com/page/Alert-messaging>. Consultado: 2019-04-30.
- [4] Triage y evaluación inicial de las urgencias. <https://www.portalveterinaria.com/articoli/-articulos/24028/triaje-y-evaluacion-inicial-de-las-urgencias.html>. Consultado: 2020-05-27.
- [5] Vetmapp - la major manera de encontrar clínicas veterinarias. <http://www.vetmapp.com/es/>. Consultado: 2019-04-27.
- [6] Karina Sokolova, Marc Lemercier, and Ludovic Garcia. Android passive mvc: a novel architecture model for android application development. In *International Conference on Pervasive Patterns and Applications*, 2013.
- [7] M. Baez, A. Borrego, J. Cordero, L. Cruz, M. Gonzalez, F. Hernandez, D. Palomero, J Rodriguez de Llera, D. Sanz, M. Saucedo, P. Torralbo, and A. Zapata. Introduccion a android. page 3, 2012.
- [8] R. Meir. *Android 4 Application Development*. John Wiley Sons, Inc, Indianapolis, Indiana, 2012.

- [9] Mark H. Goadrich and Michael P. Rogers. Smart smartphone development: Ios versus android. In *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*, SIGCSE '11, page 607, New York, NY, USA, 2011. ACM.
- [10] Mobile operating system market share in chile - may 2019. <http://gs.statcounter.com/os-market-share/mobile/chile>. Consultado: 2019-06-15.
- [11] Google developers Android Studio. <https://developer.android.com/studio/releases/index.html>. Consultado: 2019-06-09.
- [12] Eclipse for android developers. <https://www.eclipse.org/downloads/packages/>. Consultado: 2019-06-15.
- [13] Xcode ide. <https://developer.apple.com/xcode/features/>. Consultado: 2019-06-17.
- [14] C Thomas Wu. *An Introduction to Object-Oriented Programming with Java*. Mc Graw Hill, 2010.
- [15] M. Moskala and I. Wodja. *Android Development With Kotlin*. Packt, 2017.
- [16] What is xml. https://www.w3schools.com/xml/xml_what.asp. Consultado 2019-06-10.
- [17] G. Martinez and P. Lopez. *Advanced Databases Project: real-time databases and Firebase*. Université Libre de Bruxelles, Brussels, 2018.
- [18] A. B. M. Moniruzzaman and Syed Akhter Hossain. Nosql database: New era of databases for big data analytics - classification, characteristics and comparison. *CoRR*, abs/1307.0191, 2013.
- [19] Google maps. <https://whatis.techtarget.com/definition/Google-Maps>. Consultado: 2019-06-18.
- [20] Haneen Hijazi, Thair Khmour, and Abdulsalam Alarabeyyat. A review of risk management in different software development methodologies. *International Journal of Computer Applications*, 45(7):9, 2012.

- [21] Félix García, Manuel Serrano, José A Cruz-Lemus, Marcela Genero, Coral Calero, and Mario Piattini. La experimentación en la docencia de ingeniería del software. *X Jornadas de Enseñanza Universitaria de la Informática (JENUI'04)*, pages 237–245, 2004.
- [22] Marcela Genero Bocco and José A. Cruz-Lemus. Experimentación en ingeniería de software. Technical report, Universidad de Castilla-La Mancha Escuela Superior de Informática Departamento de Tecnologías y Sistemas de Información, Ciudad Real - España.
- [23] Omar Gómez, Juan Ucán, and Gerzon Gómez. Aplicación del proceso de experimentación a la ingeniería de software. *Abstraction and Application Magazine*, 8, 2014.
- [24] Documento de requisitos de usuario/software sistema de ejemplo. https://www.u-cursos.cl/ingenieria/2007/1/CC51A/1/material_docente/bajar?id_material=119354, Abril 2014.
- [25] Adelaida Ramírez Fernández. Clasificaciones de tipos de requisitos para la mejora del proceso de desarrollo de software. Master's thesis, 2012.
- [26] C Gutierrez-Soto. casosuso. <http://www.dcc.uchile.cl/~clgutier/casosUso.doc>.
- [27] Mark W Maier, David Emery, and Rich Hilliard. Ansi/ieee 1471 and systems engineering. *Systems Engineering*, 7(3):257–270, 2004.
- [28] Helen C Purchase, Linda Colpoys, Matthew McGill, David Carrington, and Carol Britton. Uml class diagram syntax: an empirical study of comprehension. In *Proceedings of the 2001 Asia-Pacific symposium on Information visualisation-Volume 9*, page 3. Australian Computer Society, Inc., 2001.
- [29] Paul Kimmel. *Manual de UML*. McGraw-Hill, 2007.
- [30] Ashutosh Bharadwaj and Shailendra Mishra. Comparison of sorting algorithms based on input sequences. *International Journal of Computer Applications*, 78(14), 2013.

ANEXOS

A. Resultados pruebas experimentación con la aplicación móvil

Experimentación con el dueño de mascota n°5	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 60 minutos.
Nombre mascota	Nina
Veterinario	veterinario de prueba 1
Tiempo de búsqueda sin app	60 minutos
Tiempo de búsqueda con app	28 minutos
Resultado prueba	✓ Se cumple el objetivo, ya que hubo un ahorro de 32 minutos en los tiempos de búsqueda con la app.

Cuadro A.1: Prueba de experimentación con el dueño de mascota 5.

Experimentación con el dueño de mascota n°6	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 28 minutos.
Nombre mascota	Princesa
Veterinario	veterinario de prueba 1
Tiempo de búsqueda sin app	20 minutos
Tiempo de búsqueda con app	38 minutos
Resultado prueba	× No se cumple el objetivo, ya que hubo un aumento de 38 minutos en los tiempos de búsqueda con la app.

Cuadro A.2: Prueba de experimentación con el dueño de mascota 6.

Experimentación con el dueño de mascota n°7	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 20 minutos.
Nombre mascota	Lulu
Veterinario	veterinario de prueba
Tiempo de búsqueda sin app	20 minutos
Tiempo de búsqueda con app	8 minutos
Resultado prueba	✓ Se cumple el objetivo, ya que hubo un ahorro de 12 minutos en los tiempos de búsqueda con la app.

Cuadro A.3: Prueba de experimentación con el dueño de mascota 7.

Experimentación con el dueño de mascota n°8	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 20 minutos.
Nombre mascota	Antonis
Veterinario	veterinario de prueba 1
Tiempo de búsqueda sin app	20 minutos
Tiempo de búsqueda con app	13 minutos
Resultado prueba	✓ Se cumple el objetivo, ya que hubo un ahorro de 7 minutos en los tiempos de búsqueda con la app.

Cuadro A.4: Prueba de experimentación con el dueño de mascota 8.

Experimentación con el dueño de mascota n°9	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 17 minutos.
Nombre mascota	Eyun
Veterinario	veterinario de prueba
Tiempo de búsqueda sin app	17 minutos
Tiempo de búsqueda con app	15 minutos
Resultado prueba	✓ Se cumple el objetivo, ya que hubo un ahorro de 2 minutos en los tiempos de búsqueda con la app.

Cuadro A.5: Prueba de experimentación con el dueño de mascota 9.

Experimentación con sujeto el dueño de mascota n°10	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 40 minutos.
Nombre mascota	Charlie
Veterinario	veterinario de prueba 2
Tiempo de búsqueda sin app	40 minutos
Tiempo de búsqueda con app	23 minutos
Resultado prueba	✓ Se cumple el objetivo, ya que hubo un ahorro de 17 minutos en los tiempos de búsqueda con la app.

Cuadro A.6: Prueba de experimentación con el dueño de mascota 10.

Experimentación con el dueño de mascota n°11	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 60 minutos.
Nombre mascota	Carlos
Veterinario	veterinario de prueba 1
Tiempo de búsqueda sin app	60 minutos
Tiempo de búsqueda con app	17 minutos
Resultado prueba	✓ Se cumple el objetivo, ya que hubo un ahorro de 43 minutos en los tiempos de búsqueda con la app.

Cuadro A.7: Prueba de experimentación con el dueño de mascota 11.

Experimentación con el dueño de mascota n°12	
Objetivo prueba	El tiempo de búsqueda de una clínica veterinaria apta para una urgencia con la aplicación móvil, debe ser menor a 37 minutos.
Nombre mascota	Poly
Veterinario	veterinario de prueba 3
Tiempo de búsqueda sin app	37 minutos
Tiempo de búsqueda con app	12 minutos
Resultado prueba	✓ Se cumple el objetivo, ya que hubo un ahorro de 25 minutos en los tiempos de búsqueda con la app.

Cuadro A.8: Prueba de experimentación con el dueño de mascota 12.

B. Resultado encuesta objetivos específicos

Preguntas dueños de mascotas	Escala likert				
	1	2	3	4	5
¿Le ha parecido fácil encontrar el menú para crear una mascota en la aplicación móvil?	0	0	0	1	11
¿La información relacionada a la mascota es fácil de agregar a la aplicación móvil?	0	0	0	0	12
¿La cantidad de información solicitada para agregar una mascota a la aplicación es adecuada?	0	0	0	0	12

Cuadro B.1: Resultados de la encuesta Likert, realizada a los dueños de mascota relacionada a la facilidad de suministrar información de la mascota a la clínica veterinaria.

Preguntas veterinarios	Escala likert				
	1	2	3	4	5
¿La cantidad de información básica sobre la mascota es suficiente para la urgencia?	0	0	0	0	4

Cuadro B.2: Resultados de la encuesta realizada a los veterinarios relacionada a la facilidad de suministrar información de la mascota a la clínica veterinaria usando escala Likert.

Preguntas	Escala Likert				
	1	2	3	4	5
¿La ha parecido fácil seleccionar una mascota para crear una urgencia?	0	0	0	0	12
¿La forma de proporcionar los síntomas de la mascota es de su agrado?	0	0	0	4	8
¿La aplicación le ha permitido ingresar una imagen de la mascota al momento de crear una urgencia?	0	0	0	1	11
¿La aplicación le ha permitido proporcionar un vídeo de la mascota al momento de crear una urgencia?	0	0	1	9	2

Cuadro B.3: Resultados de la encuesta realizada a los dueños de mascota relacionada a la facultad de proporcionar síntomas de la mascota en la urgencia.

Preguntas	Escala Likert				
	1	2	3	4	5
¿La cantidad de información básica sobre la mascota es suficiente para la urgencia?	0	0	0	0	4

Cuadro B.4: Resultados de la encuesta realizada a los veterinarios relacionada a la facilidad de suministrar información de la mascota a la clínica veterinaria.

Preguntas	Escala Likert				
	1	2	3	4	5
¿El sistema de notificaciones le ha servido para avisar de que un veterinario ha aceptado la urgencia?	0	0	0	0	12
Cuando la aplicación le notifica que un veterinario ha tomado la urgencia, el mensaje de la notificación es fácil de entender?.	0	0	0	4	8
¿Encuentra útil este tipo de comunicación con el veterinario para una urgencia?.	0	0	0	4	8
La notificación cuando un veterinario confirma la urgencia, ¿Le indica de forma clara hacia que clínica veterinaria debe dirigirse?.	0	0	0	3	9

Cuadro B.5: Resultados de la encuesta realizada a los dueños de mascota en relación al sistema de notificaciones de la aplicación

Preguntas	Escala Likert				
	1	2	3	4	5
¿El sistema de notificaciones le ha avisado de que una mascota tiene una urgencia?	0	0	0	1	3
Cuando la aplicación le notifica que una mascota tiene urgencia, ¿El mensaje de la notificación es fácil de entender?.	0	0	0	2	2
¿Encuentra útil esta forma de avisar al dueño de mascota de que hay un veterinario dispuesto a atender la urgencia?.	0	0	0	1	3

Cuadro B.6: Resultados de la encuesta realizada a los veterinarios en relación al sistema de notificaciones de la aplicación.

Preguntas	Escala Likert				
	1	2	3	4	5
¿El mapa le ha servido para saber donde se encuentran las clínicas veterinarias?.	0	0	0	0	12
¿La información que muestra el marcador le ha sido de utilidad?.	0	0	0	0	12
¿La información en detalle de la clínica veterinaria le ha sido de utilidad?.	0	0	0	1	11

Cuadro B.7: Resultados de la encuesta realizada a los dueños de mascota en relación a la visualización de las clínicas veterinarias.

C. Navegación aplicación móvil

En la presente sección se aborda la navegabilidad de la aplicación móvil, la cual se separa en dos secciones: la primera sección corresponde a la navegación del usuario dueño de mascota. La segunda sección aborda la navegación del veterinario.

C.1. Navegación dueño de mascota

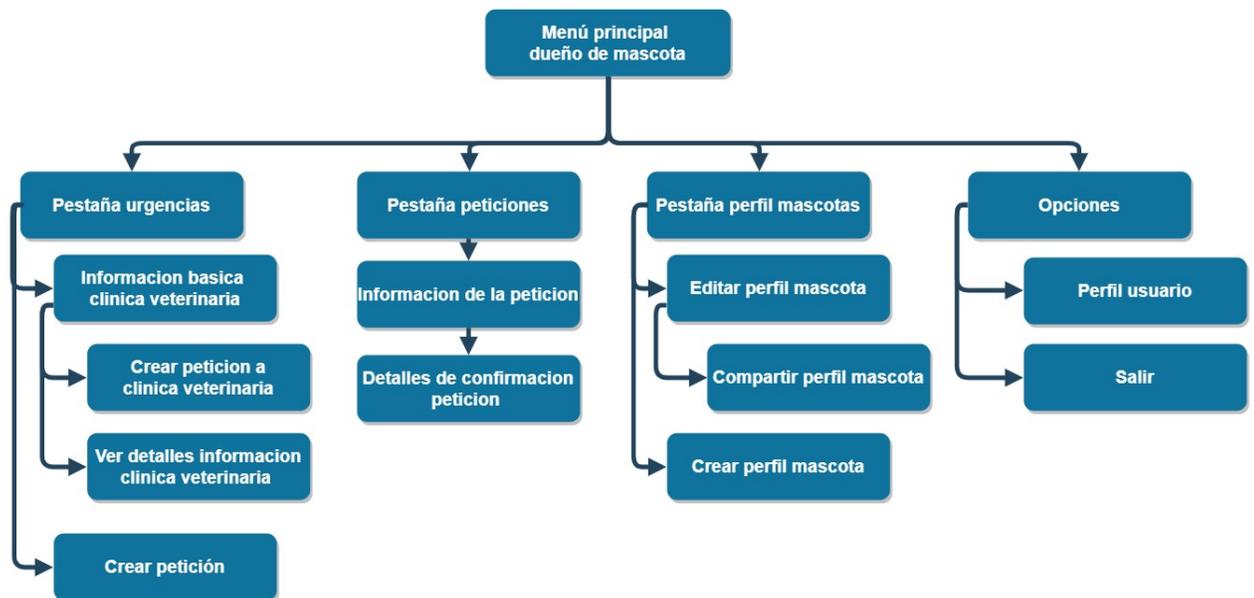


Figura C.1: Diagrama de navegación de la aplicación móvil para un dueño de mascota.

C.1.1. Registro e inicio de sesión

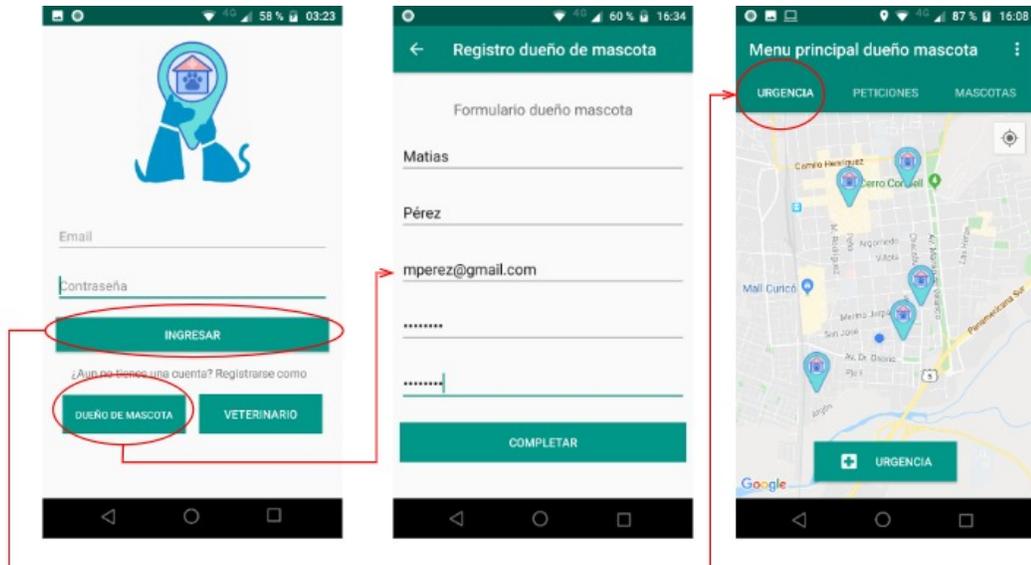


Figura C.2: Proceso de registro e inicio de sesión de un dueño de mascota.

C.1.2. Creación perfil de mascota

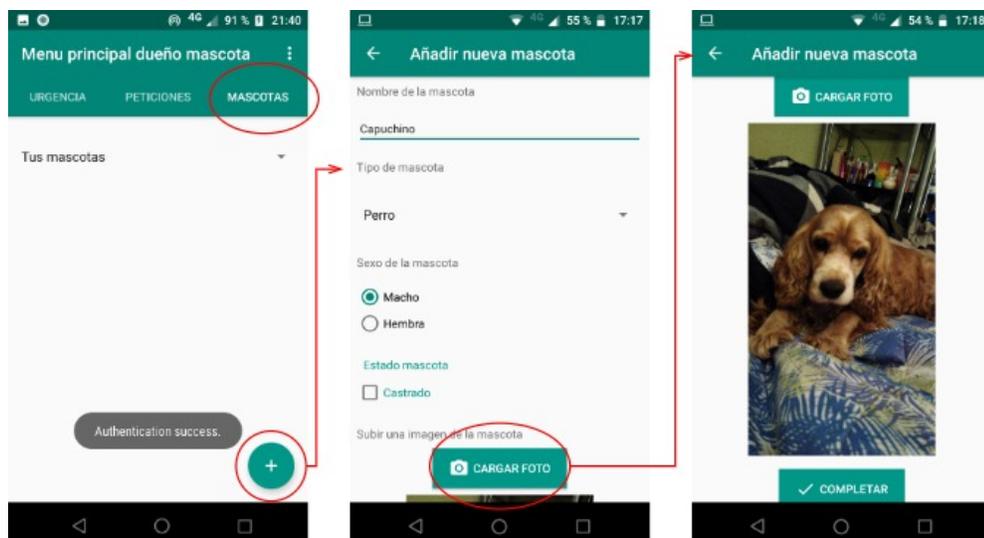


Figura C.3: Proceso creación del perfil de una mascota.

C.1.3. Creación petición de urgencia para una mascota

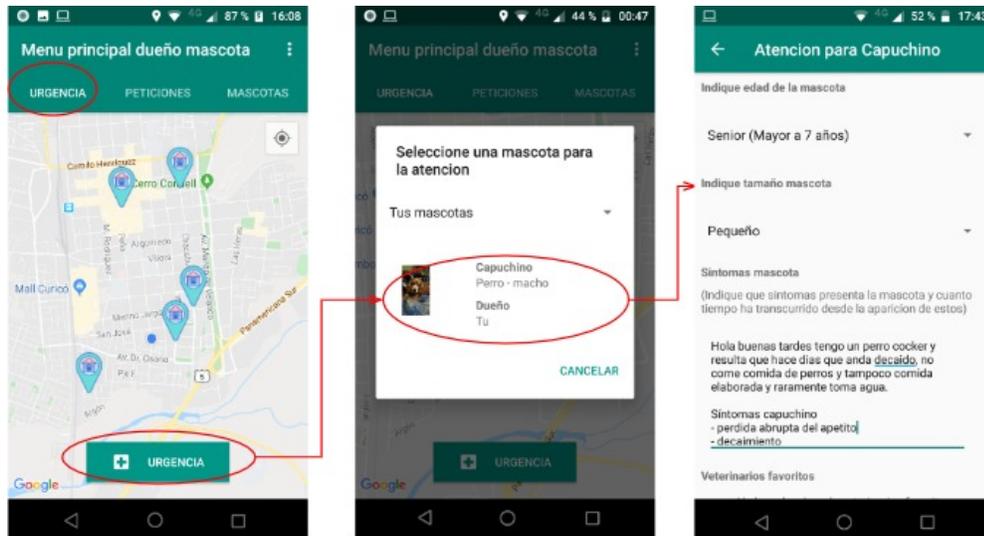


Figura C.4: Registro en formato JSON de la petición de urgencia en donde se resalta la fecha de aceptación de ésta.

C.1.4. Editar perfil de mascota

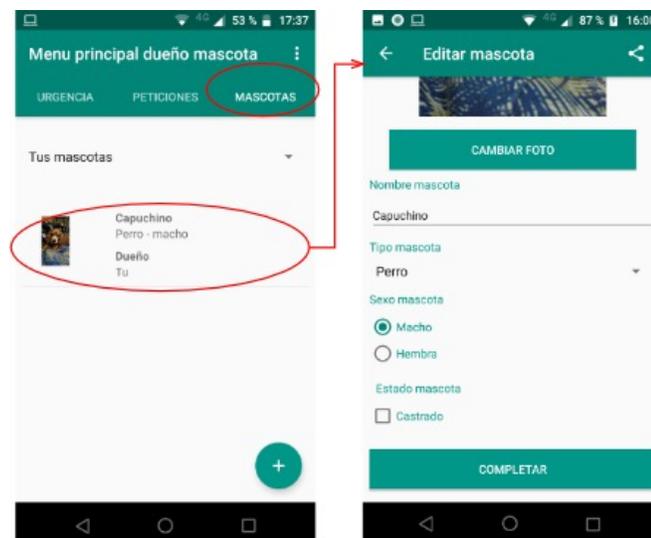


Figura C.5: Proceso de editar el perfil de una mascota.

C.1.5. Ver detalles de la confirmación de la urgencia

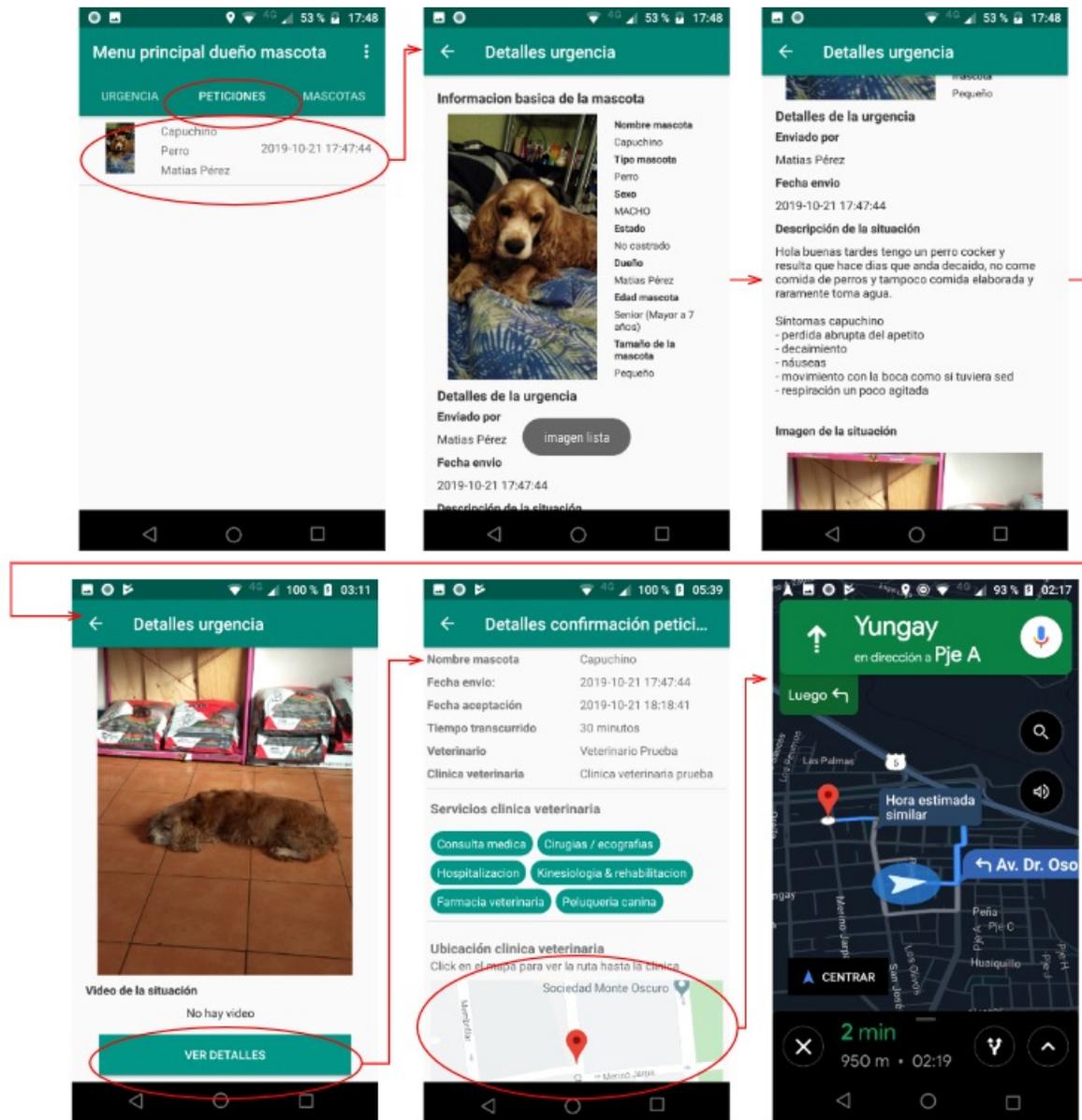


Figura C.6: Proceso de ver el detalle de la petición de urgencia.

C.1.6. Cancelar petición de urgencia que un dueño de mascota ha realizado.

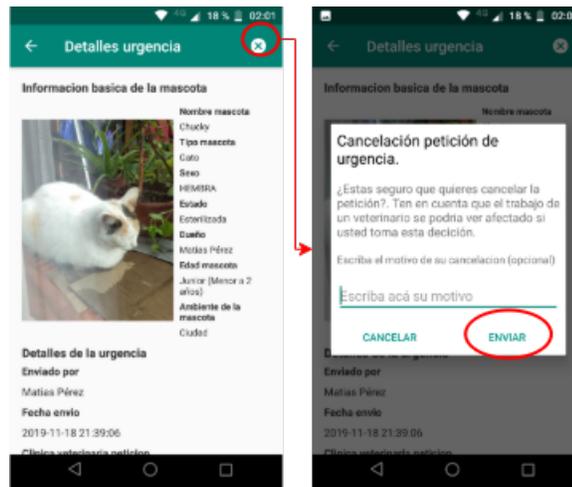


Figura C.7: Proceso de cancelar una petición de urgencia.

C.1.7. Ver el detalle de una clínica veterinaria

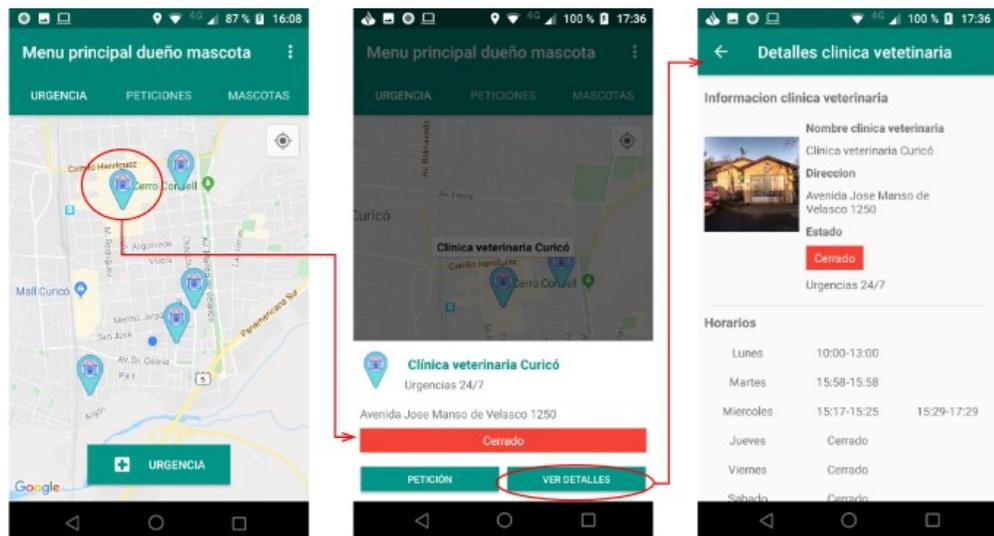


Figura C.8: Proceso para visualizar información detallada de una clínica veterinaria.

C.1.8. Ver perfil del dueño de mascota

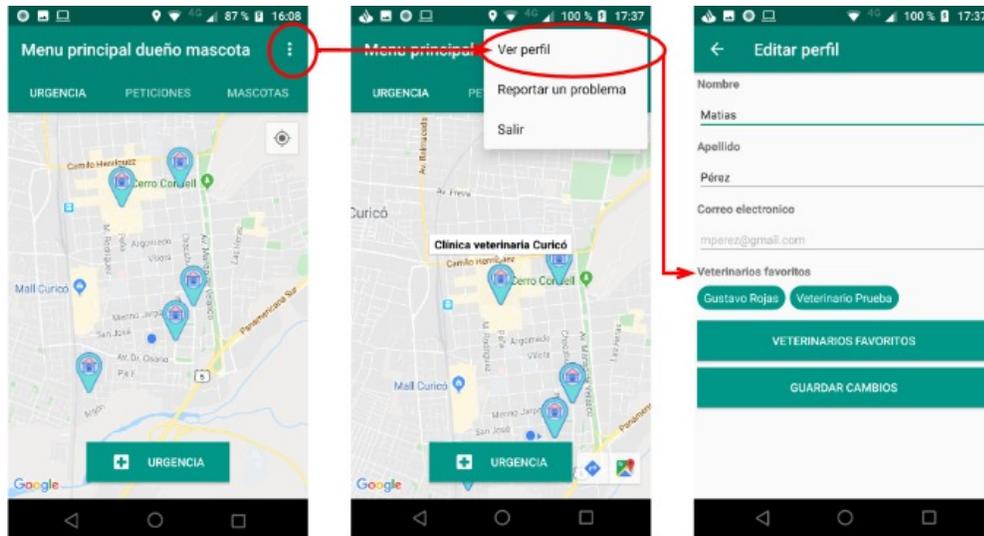


Figura C.9: Proceso para ver y editar el perfil de un usuario dueño de mascota.

C.1.9. Compartir perfil mascota con otro dueño

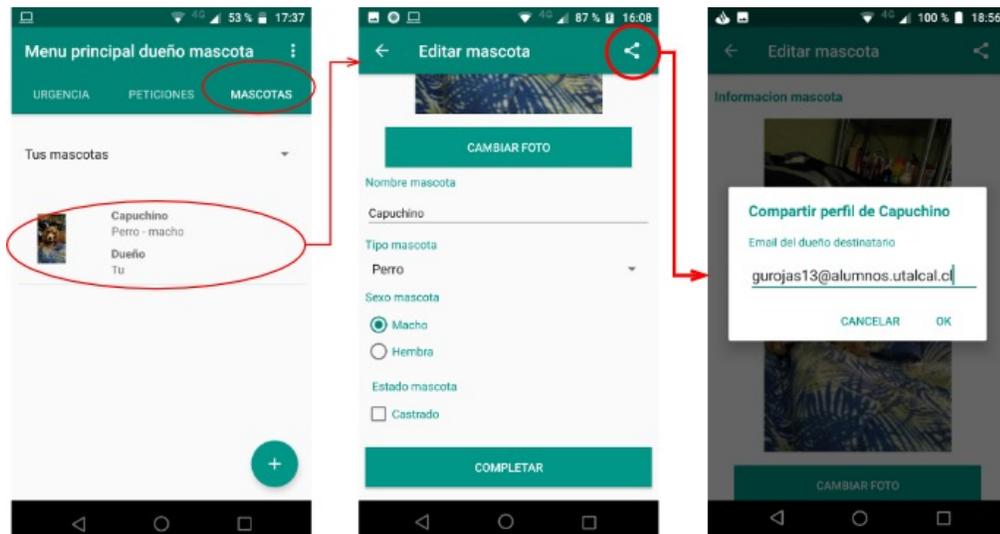


Figura C.10: Proceso de compartir la mascota con otro dueño de mascota.

C.1.10. Recibir solicitud de compartir mascota

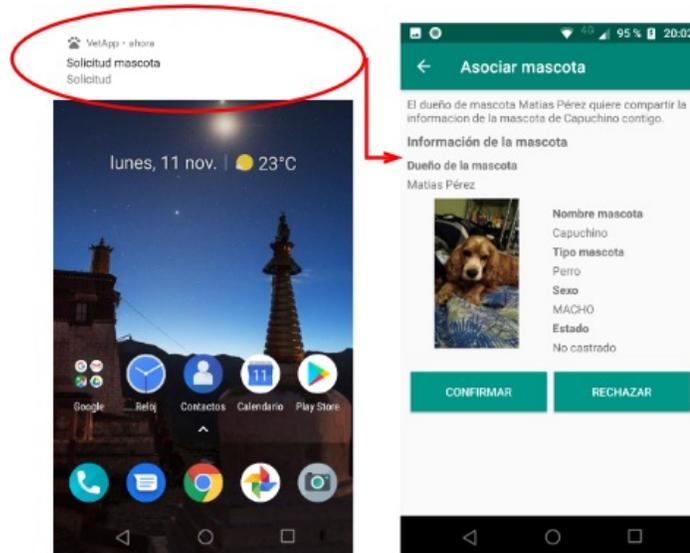


Figura C.11: Proceso de recibir una solicitud cuando un dueño comparte el perfil de una mascota

C.1.11. Aviso cancelación petición de urgencia para el dueño de mascota.

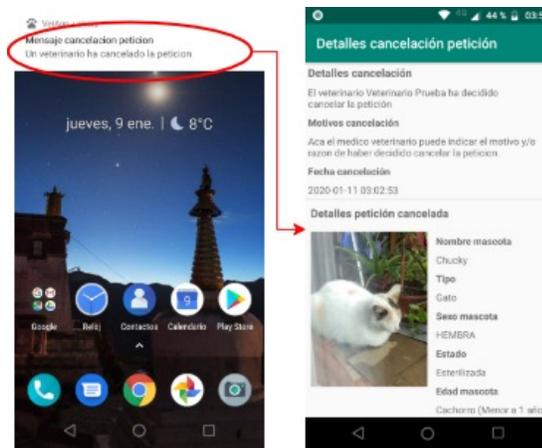


Figura C.12: Proceso de recibir una notificación cuando un veterinario cancela una petición que aceptó

C.1.12. Cerrar sesión

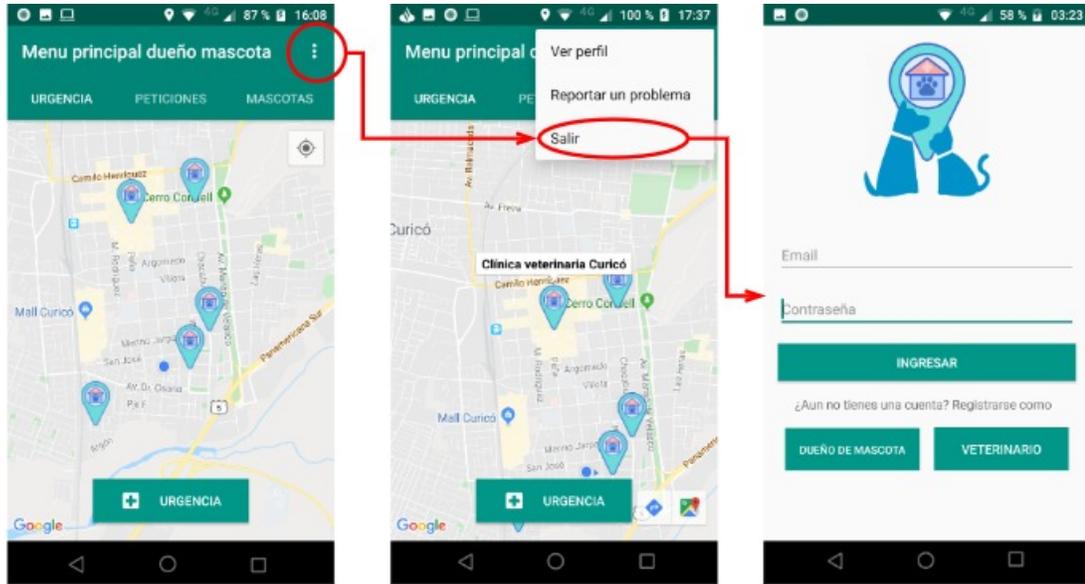


Figura C.13: Proceso de salir de la sesión.

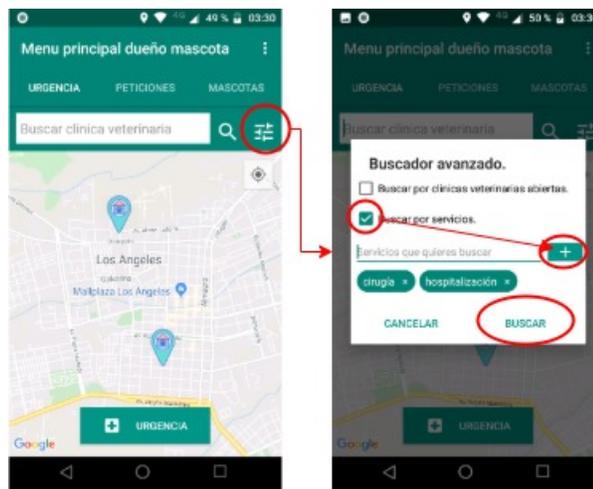


Figura C.14: Proceso buscar clínicas veterinarias abiertas y/o por servicios que éstas ofrezcan.

C.2. Navegación veterinario

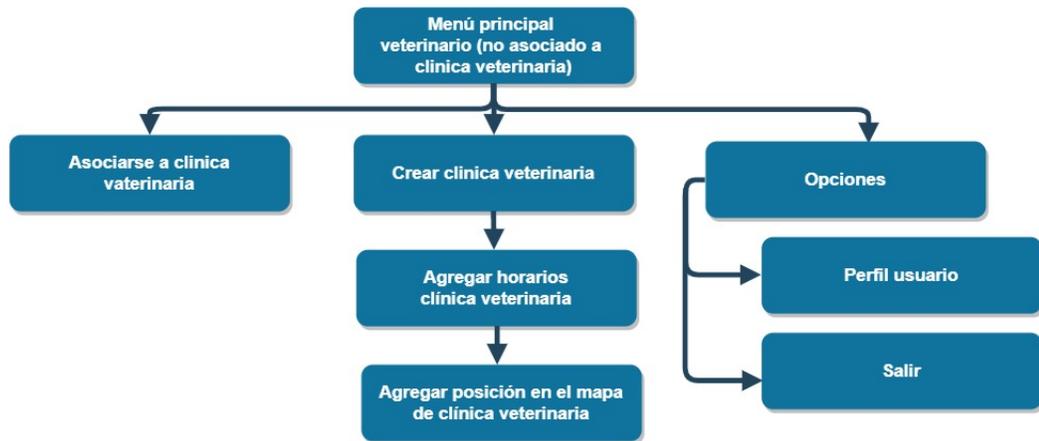


Figura C.15: Navegabilidad de un veterinario que no se encuentra asociado a ninguna clínica.

C.2.1. Registro e inicio de sesión veterinario.

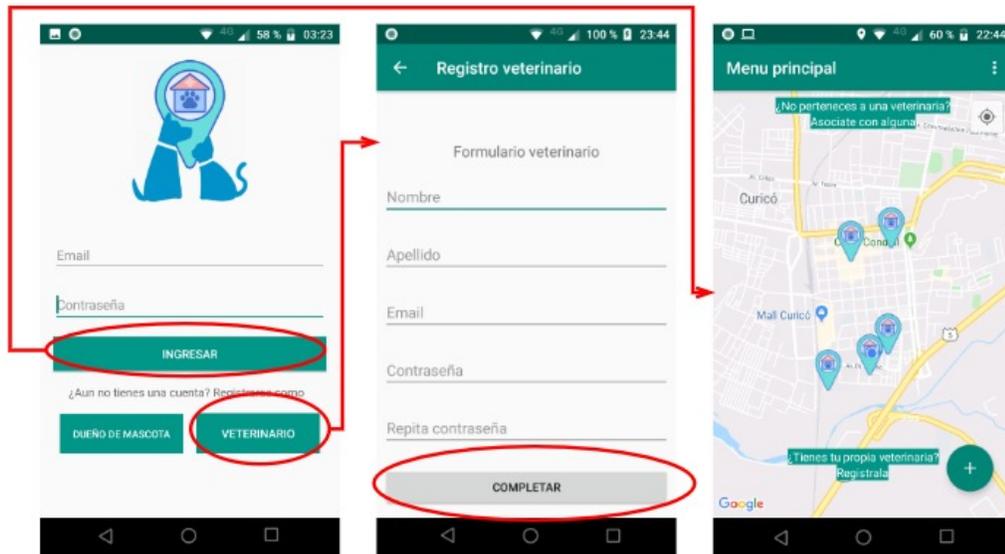


Figura C.16: Proceso de registro e inicio de sesión de un veterinario.

C.2.2. Asociación a una clínica ya existente

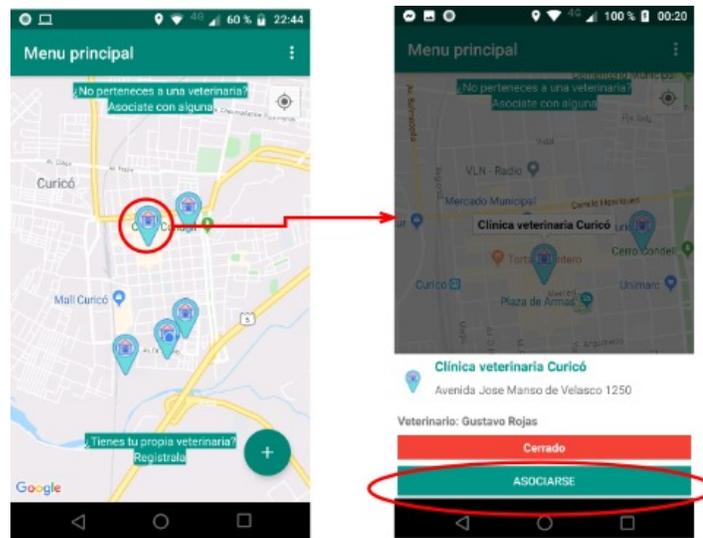


Figura C.17: Proceso de asociarse con una clínica veterinaria.

C.2.3. Recepción solicitud de asociación.

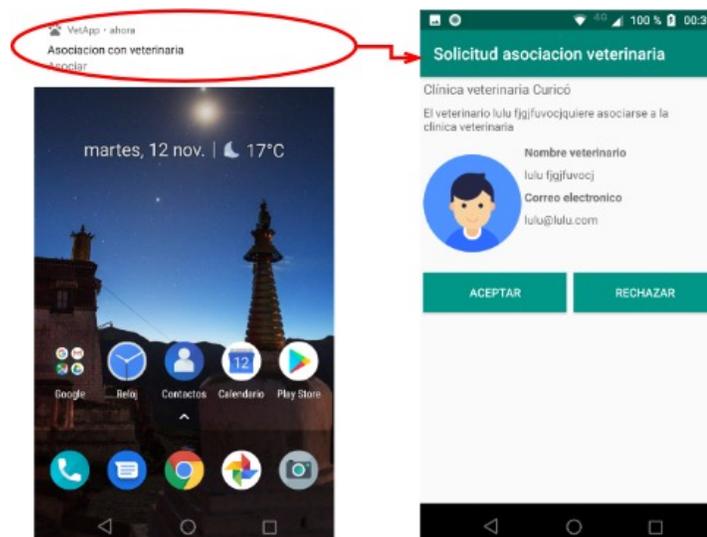


Figura C.18: Proceso de aceptación o rechazo de asociación a una clínica veterinaria.

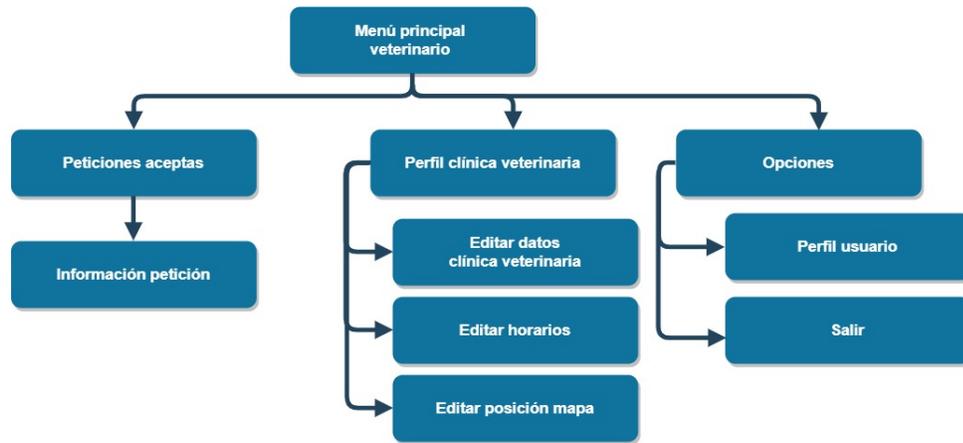


Figura C.19: Navegabilidad de un veterinario que ya se encuentra asociado a una clínica.

C.2.4. Editar información básica clínica veterinaria.

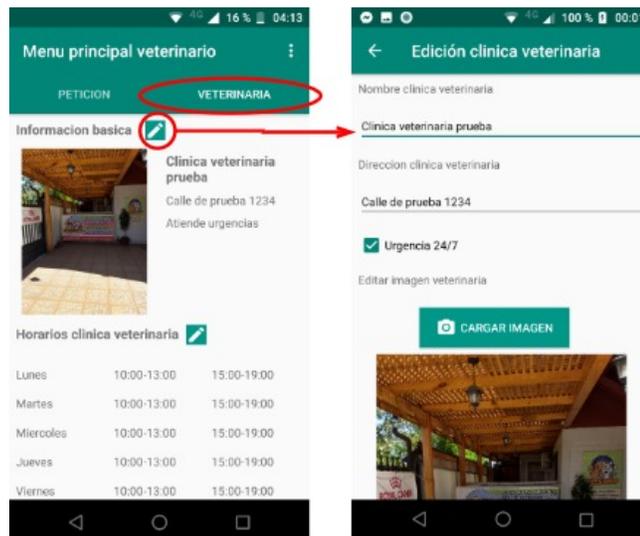


Figura C.20: Proceso de editar la información básica de una clínica veterinaria.

C.2.5. Editar horarios clínica veterinaria.

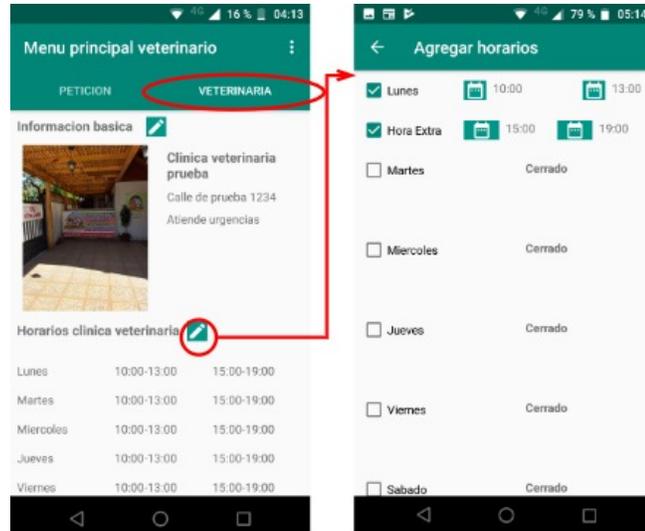


Figura C.21: Proceso de editar los horarios de apertura y cierre de una clínica veterinaria.

C.2.6. Editar posición clínica veterinaria.

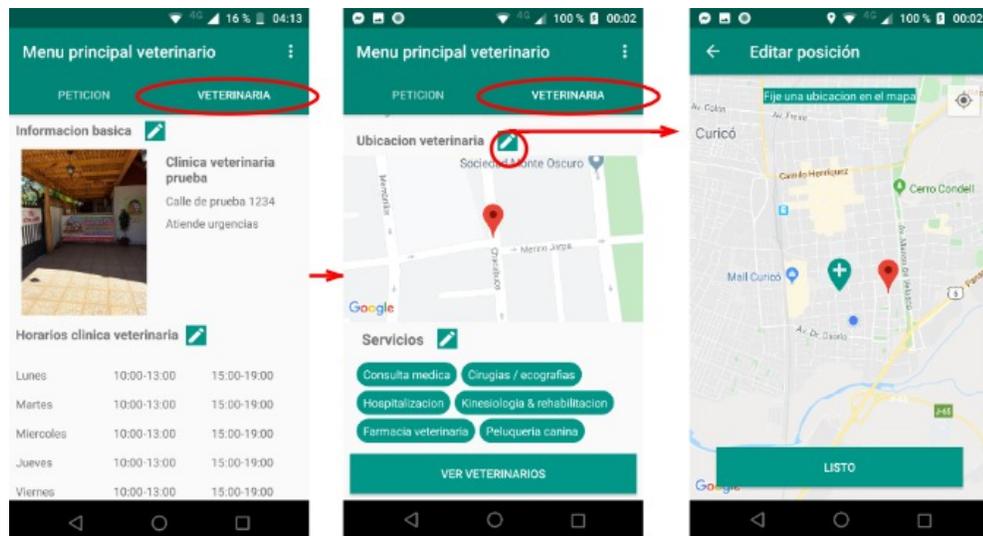


Figura C.22: Proceso de editar la ubicación de una clínica veterinaria.

C.2.7. Agregar y remover servicios clínica veterinaria.

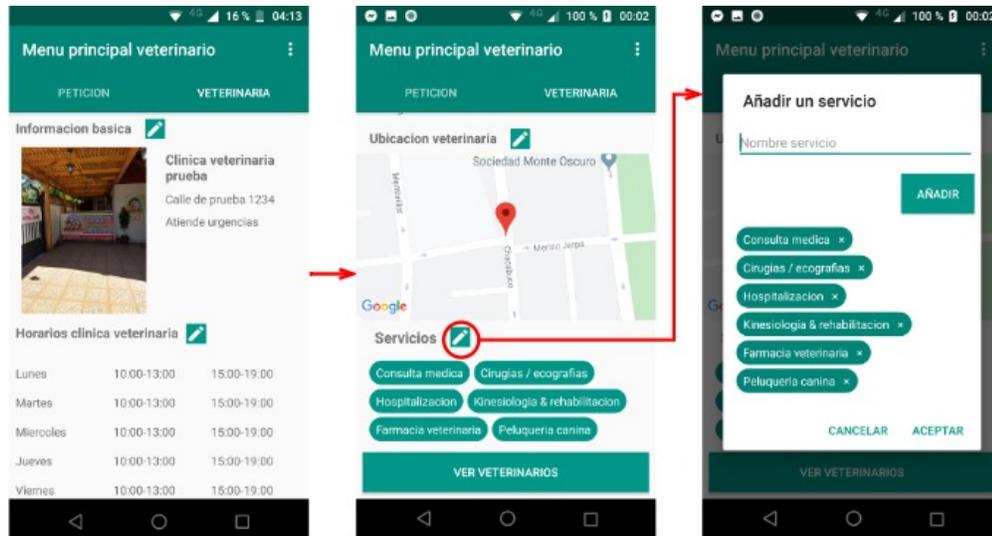


Figura C.23: Proceso de agregar o eliminar servicios que ofrece una clínica veterinaria.

C.2.8. Listar veterinarios que trabajan en una clínica.

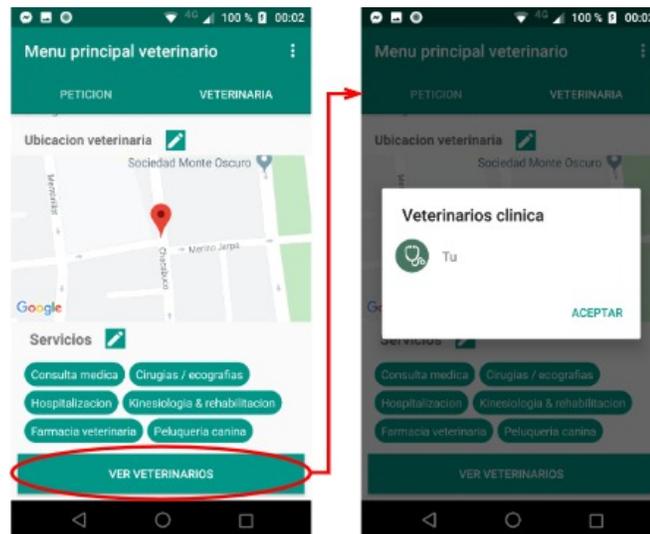


Figura C.24: Proceso de listar los veterinarios que trabajan o pertenecen a una clínica veterinaria.

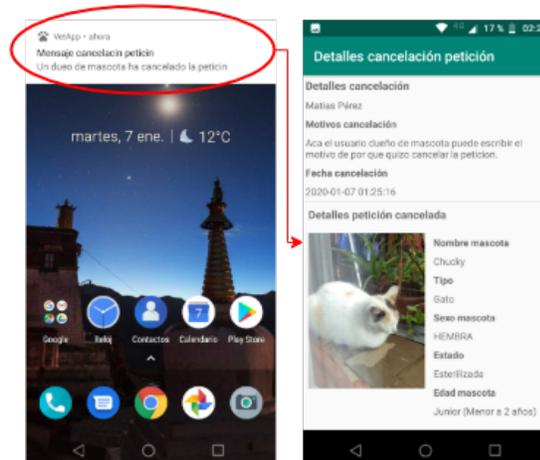
C.2.9. Aviso cancelación petición de urgencia para el veterinario.

Figura C.25: Proceso de notificar que un dueño de mascota ha cancelado la petición.

C.2.10. Revisar detalle petición de urgencia.

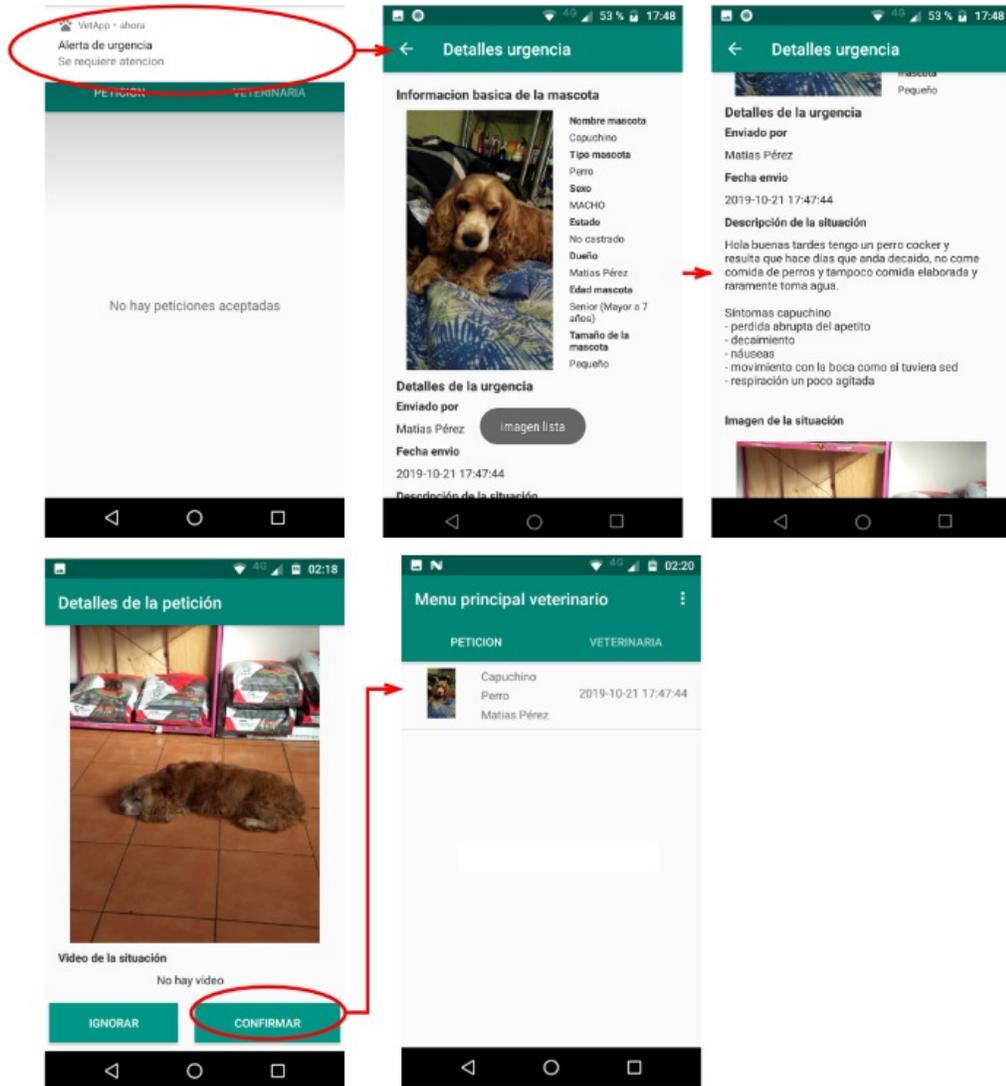


Figura C.26: Proceso de revisar una petición de urgencia.

C.2.11. Perfil veterinario

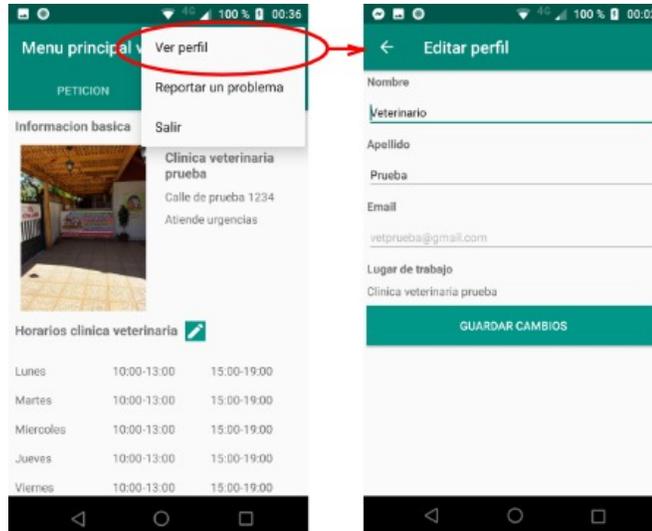


Figura C.27: Proceso de ver el perfil de un veterinario

C.2.12. Cancelar petición que el veterinario ha aceptado

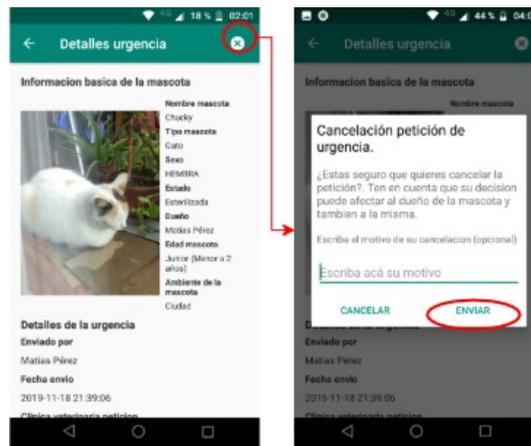


Figura C.28: Proceso cuando el médico veterinario decide cancelar una petición que él ya aceptó.

C.2.13. Crear una nueva clínica veterinaria

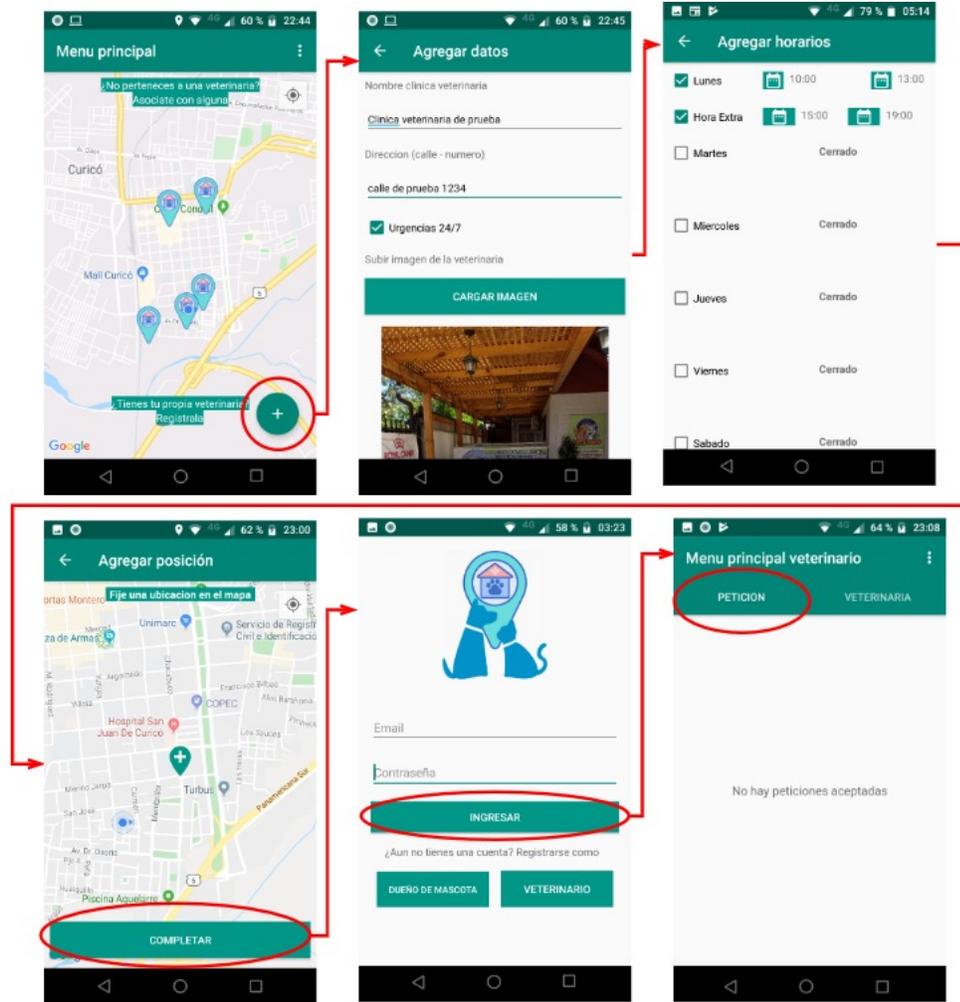


Figura C.29: Proceso de crear una clínica veterinaria.

C.2.14. Salir de la sesión

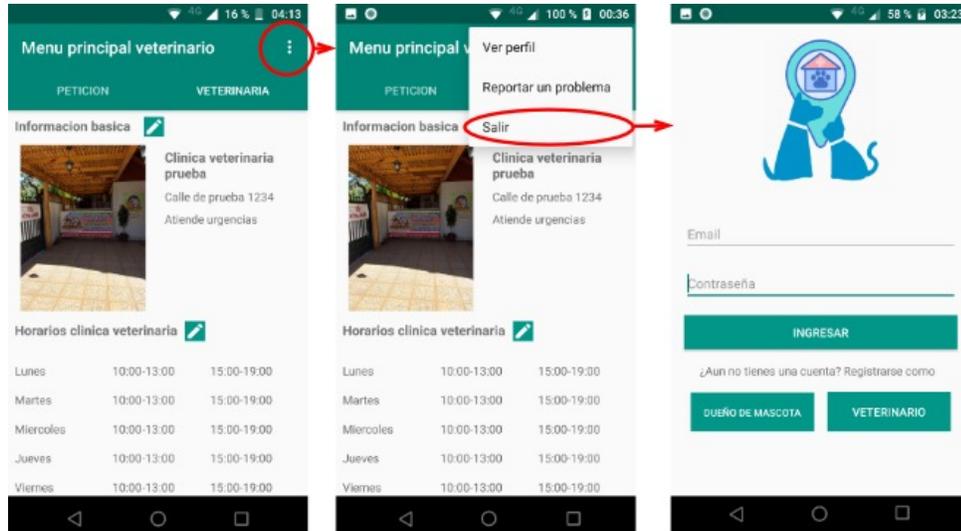


Figura C.30: Proceso de cerrar la sesión de un veterinario.

D. Documento de requisitos

D.1. Introducción

En el presente documento se dará a conocer el propósito, alcances y contexto de la implementación de la aplicación móvil.

D.1.1. Propósito

El propósito principal de implementar la aplicación móvil es reducir el tiempo de búsqueda que emplea un dueño de mascota en encontrar una clínica veterinaria apta para atender una urgencia.

La aplicación móvil permitirá avisar con información relevante, a uno o varios médicos veterinarios, de la existencia de una urgencia que presenta la mascota de un usuario. Con la información suministrada por el dueño, el médico veterinario puede preparar con anticipación la clínica veterinaria para atender la urgencia de manera más efectiva. Junto con el aviso, el dueño podrá entregar detalles de la urgencia, como por ejemplo: si la mascota se trata de un perro o gato, edad, descripción de la situación, síntomas, además el dueño puede complementar la información entregada con una imagen y/o video. Una vez que el veterinario reciba el aviso, podrá analizar la situación y avisar al dueño de que la clínica se encuentra apta para recibir la urgencia o no. También la aplicación móvil contará con un mapa en donde el dueño de la mascota podrá visualizar las clínicas veterinarias que se encuentran disponibles para enviar el aviso de la urgencia.

En definitiva la aplicación viene siendo una alternativa rápida para atención veterinaria que ayudaría a dar una sensación de tranquilidad a los dueño de mascota

frente a una situación angustiante.

D.1.2. Alcances

La aplicación móvil a construir cuenta con las siguientes consideraciones:

- La aplicación móvil solo se limita a trabajar con urgencias de mascotas.
- La aplicación móvil solo considera mascotas de tipo perro o gato.
- La aplicación móvil busca clínicas veterinarias en la misma ciudad donde el dueño de la mascota generó la petición de atención de urgencia.
- La aplicación móvil necesita conexión a internet para funcionar.
- La aplicación móvil es gratuita.
- La aplicación móvil está en idioma español.
- La aplicación móvil se desarrolla para dispositivos Android.
- La aplicación móvil no gestiona la atención veterinaria a domicilio.
- La aplicación móvil no considera el medio de pago de la atención veterinaria.

D.1.3. Contexto

Cuando una mascota necesita una atención urgente por parte de un médico veterinario, el dueño, busca clínicas veterinarias que puedan atender dicha situación. Pero algunas veces ocurre que los centros más cercanos no siempre son la mejor opción o la clínica que frecuentan se encuentra ocupada, entonces obliga al dueño a realizar una búsqueda, por lo que se emplea un cierto tiempo que puede ser primordial para el diagnóstico de la mascota. Además muchos dueños de mascota llevan su mascota de urgencia a la clínica veterinaria sin un previo aviso, lo que provoca que la atención sea más tardía. Es por las razones mencionadas anteriormente que la aplicación intenta reducir el tiempo que emplea una persona en buscar una clínica veterinaria apta para atender la urgencia de la mascota mediante un aviso temprano a los veterinarios. Un efecto que tiene el hecho de avisar la urgencia con anticipación,

es que los veterinarios pueden preparar los elementos necesarios para atender a la mascota.

El desarrollo de aplicación móvil se basa usando la metodología iterativa incremental, en la cual se usan 8 incrementos de 2 semanas cada uno. La solución se desarrolla usando el ambiente Android con el lenguaje de programación Java. Para mantener la persistencia de los datos, se usa Firebase realtime, la cual es una base de datos no relacional de Google.

D.2. Descripción general

D.2.1. Características de los usuarios

Tipo de usuarios	Descripción
Dueño de mascota	Este tipo de usuario es aquel que puede crear perfiles de mascotas y enviar peticiones de urgencia con información de ellas a los veterinarios.
Veterinario	Este tipo de usuario es aquel que trabaja en la clínicas veterinarias y son los encargados de recibir las peticiones de urgencia que un dueño de mascota envía.

Cuadro D.1: Descripción de los usuarios de la aplicación móvil

D.2.2. Suposiciones y dependencias

Se presentan a continuación las condiciones que se deben cumplir para que el sistema funcione correctamente.

- El sistema funciona solamente en sistemas operativos Android.
- Los veterinarios deben trabajar en una única clínica veterinaria.
- Los usuarios deben contar con una conexión a Internet para hacer uso de la aplicación.
- El dueño de mascota solo puede crear perfiles de perros y gatos.

D.2.3. Descripción del modelo

En esta sub-sección se muestra el funcionamiento del sistema usando diagramas de casos de uso.

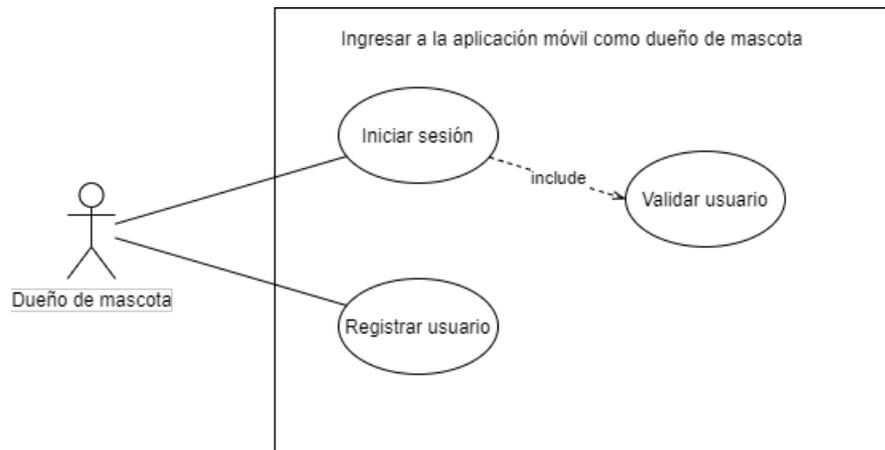


Figura D.1: Caso de uso 1

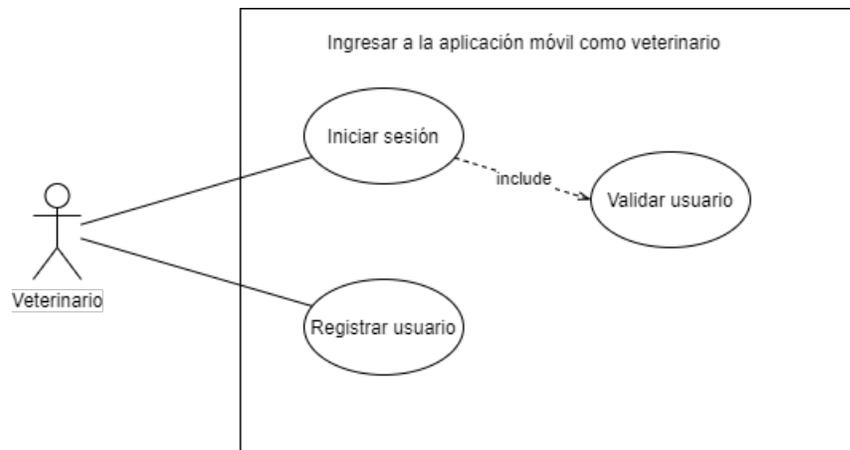


Figura D.2: Caso de uso 2

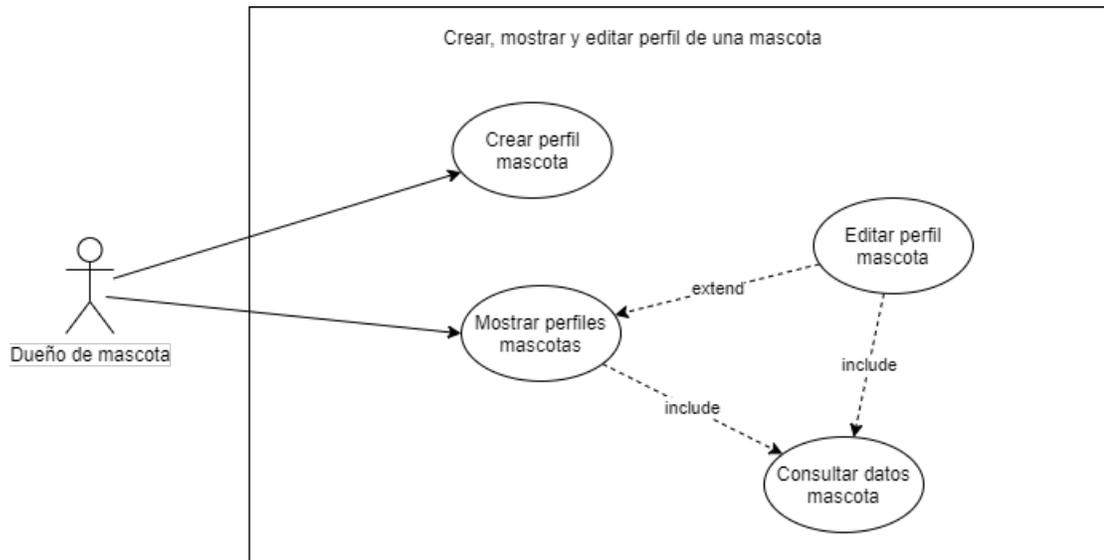


Figura D.3: Caso de uso 3

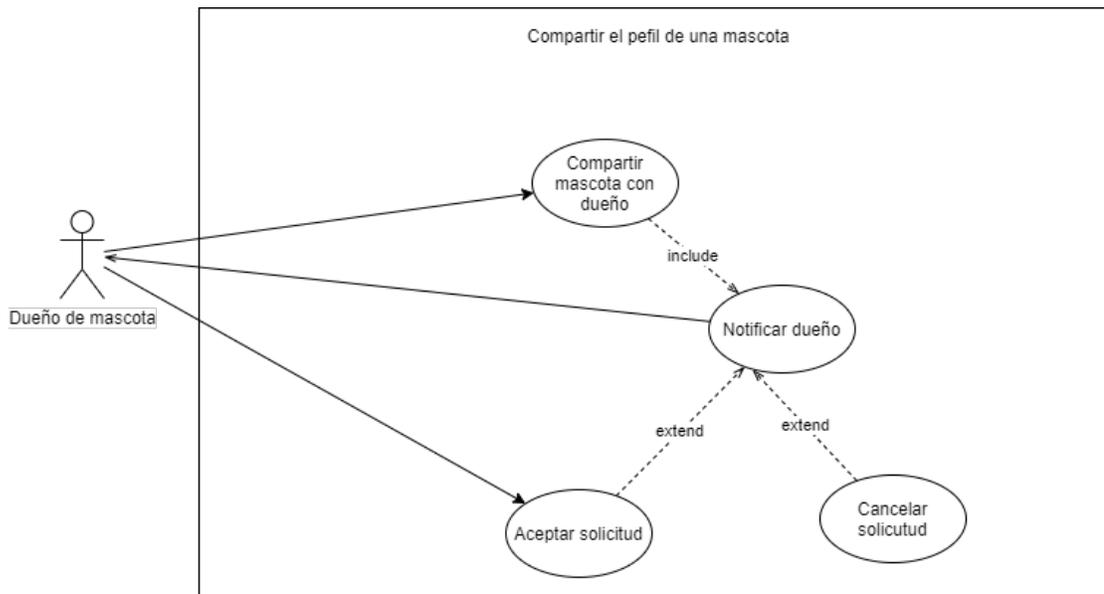


Figura D.4: Caso de uso 4

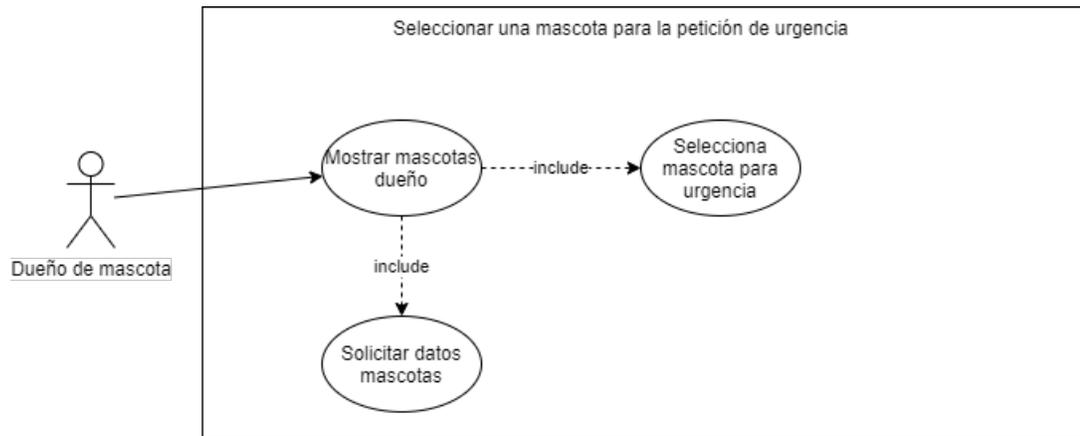


Figura D.5: Caso de uso 5

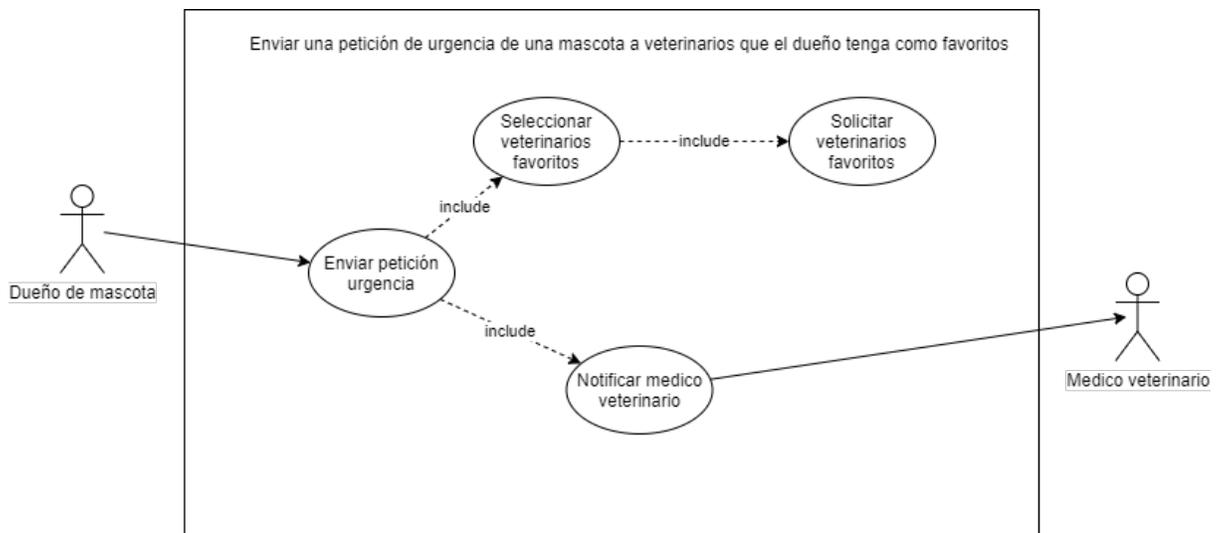


Figura D.6: Caso de uso 6

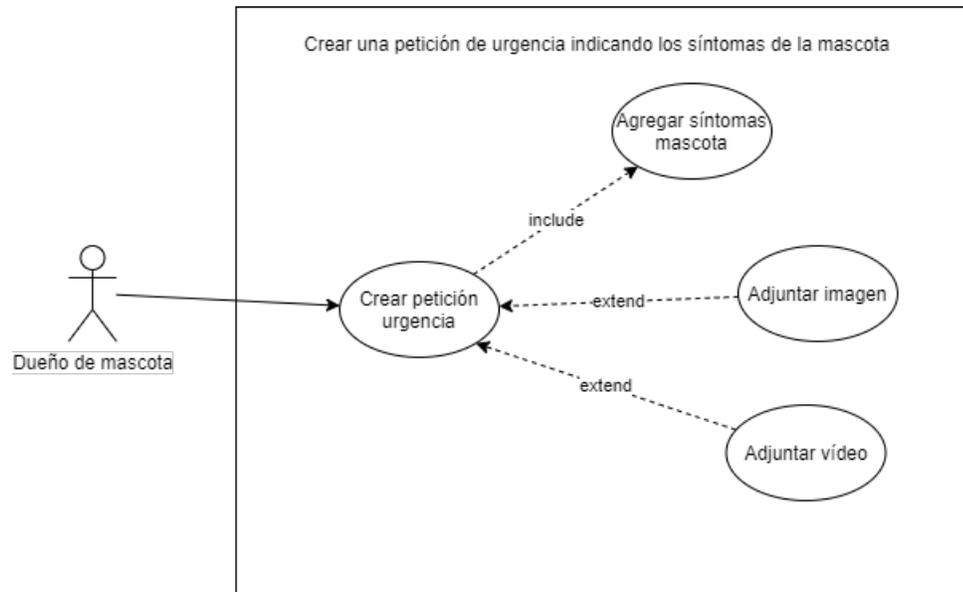


Figura D.7: Caso de uso 7

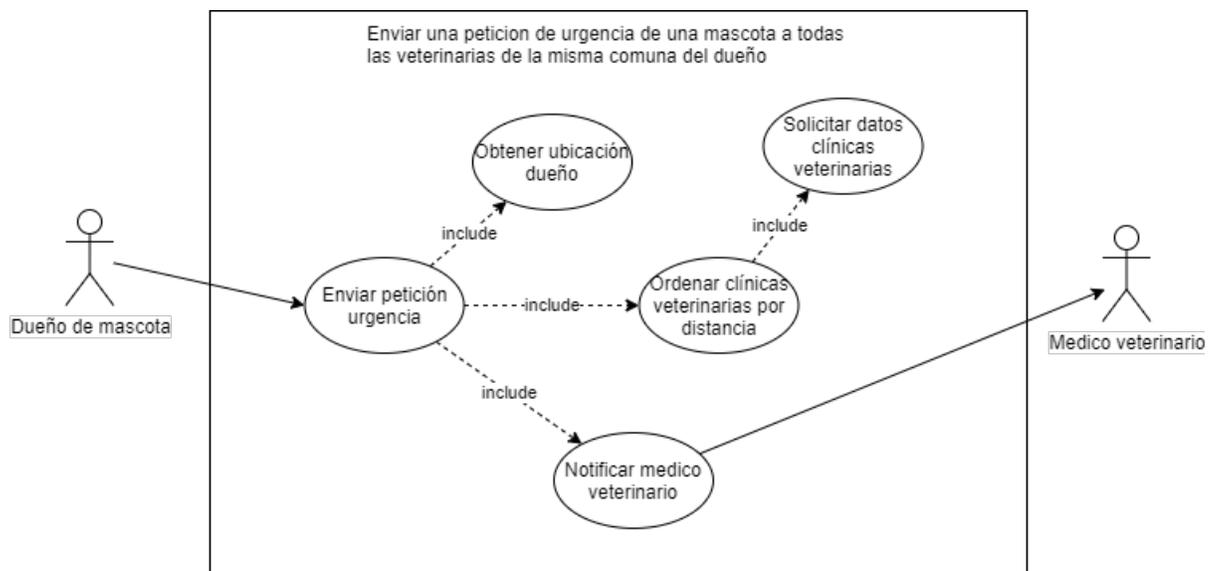


Figura D.8: Caso de uso 8

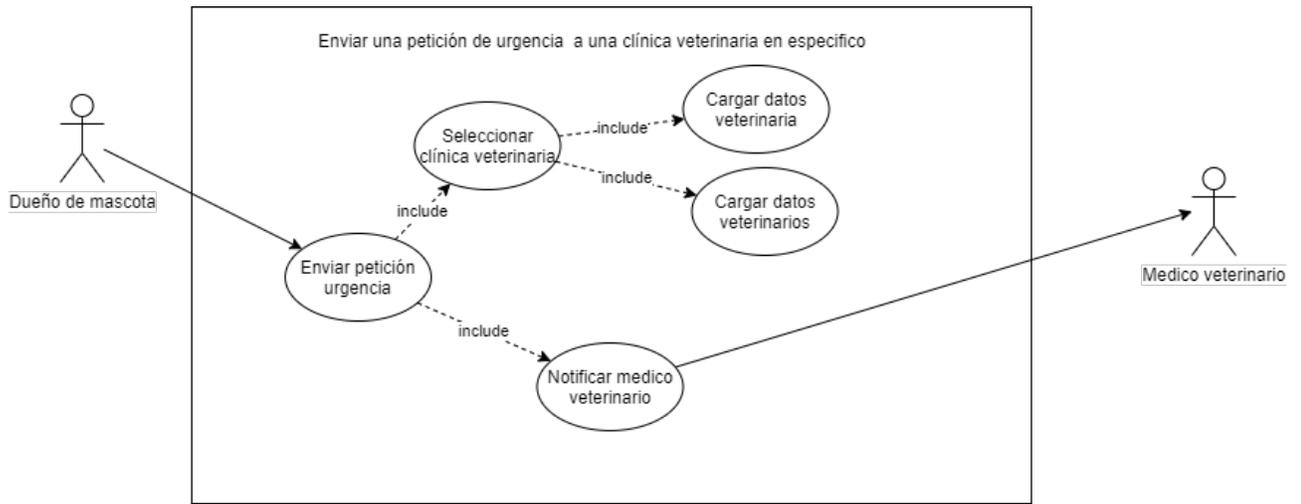


Figura D.9: Caso de uso 9

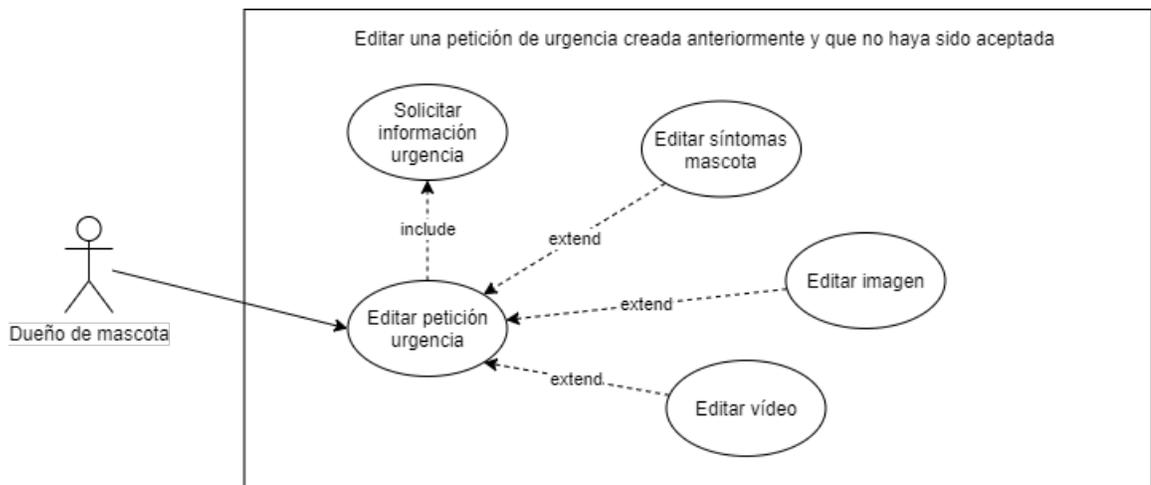


Figura D.10: Caso de uso 10

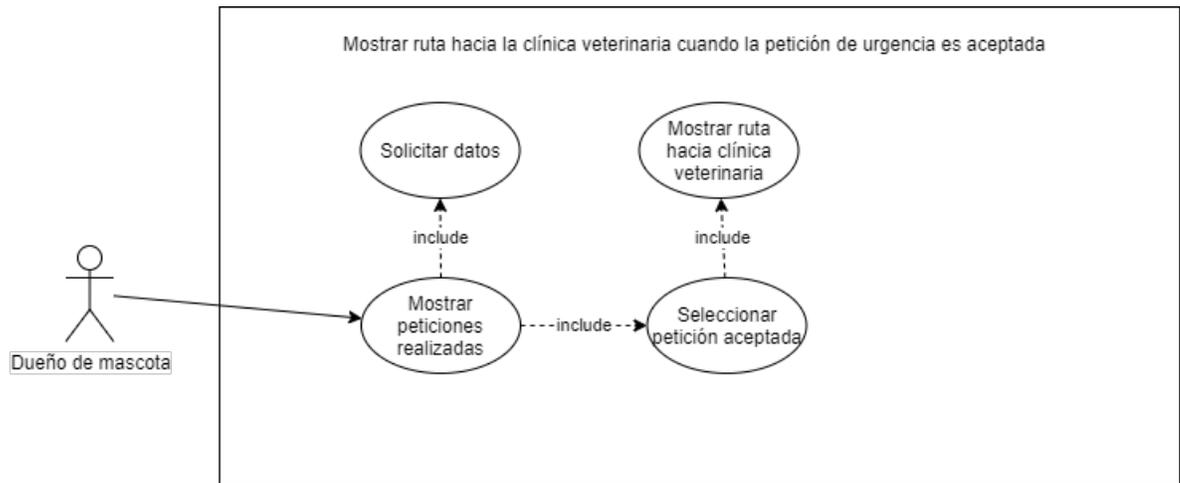


Figura D.11: Caso de uso 11

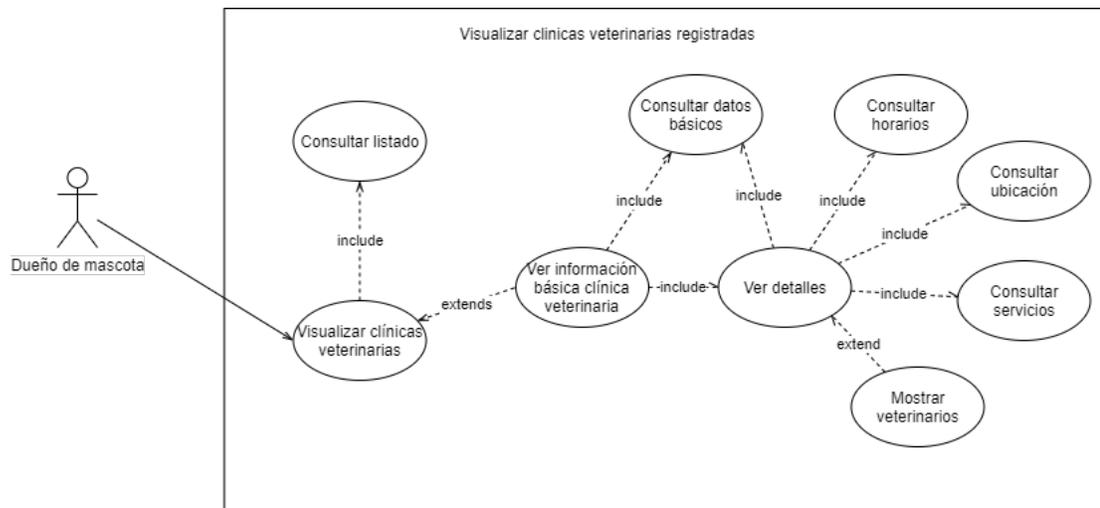


Figura D.12: Caso de uso 12

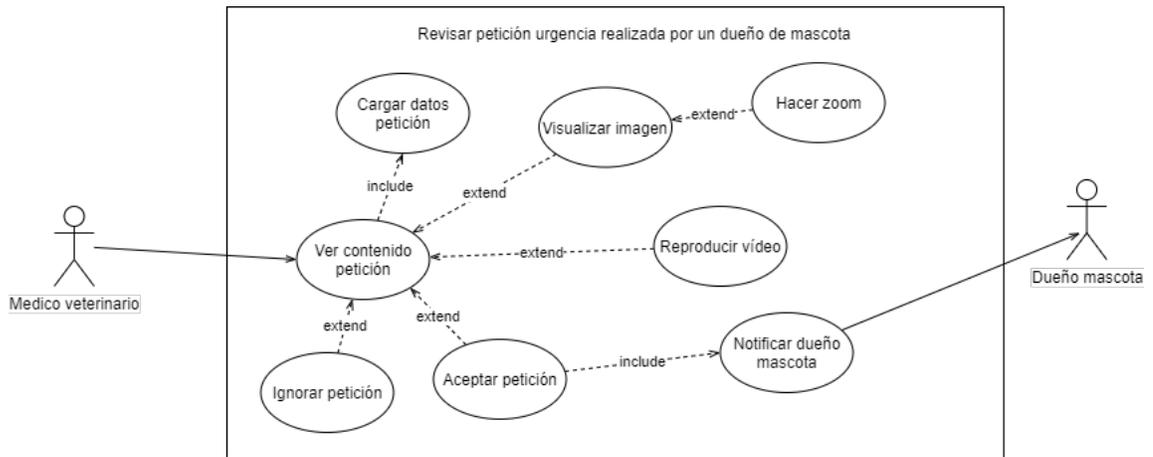


Figura D.13: Caso de uso 13

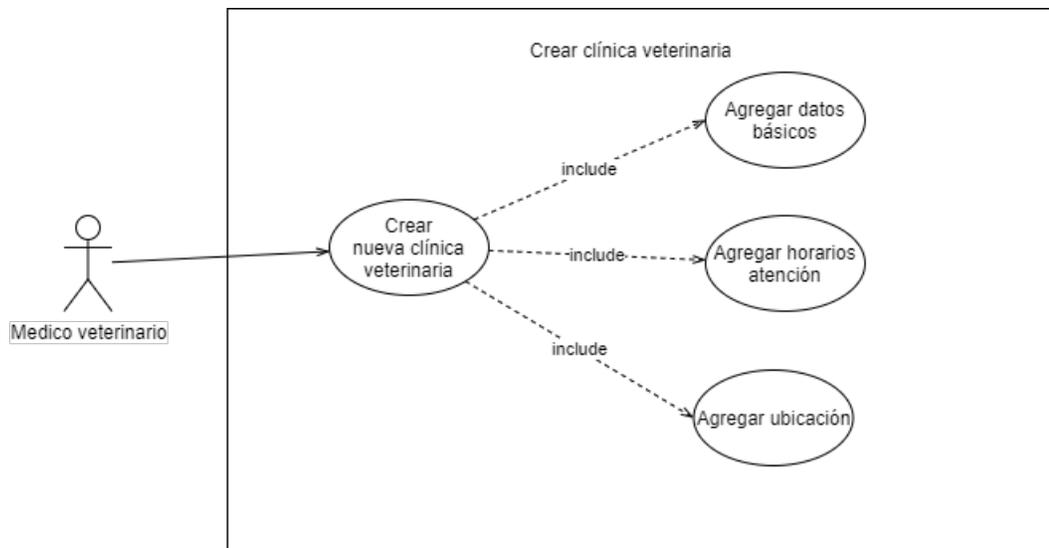


Figura D.14: Caso de uso 14

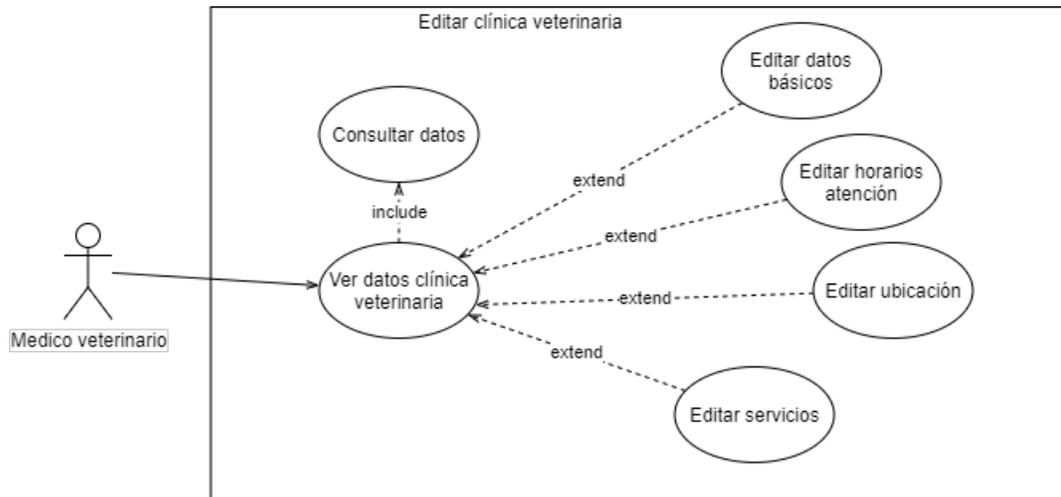


Figura D.15: Caso de uso 15

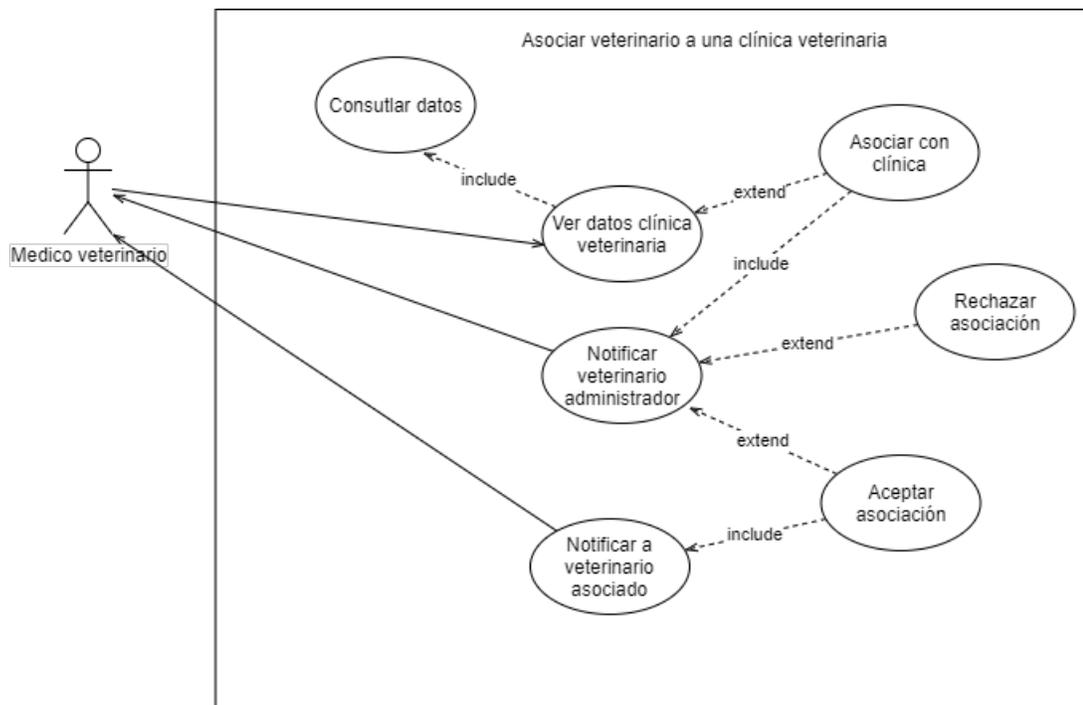


Figura D.16: Caso de uso 16

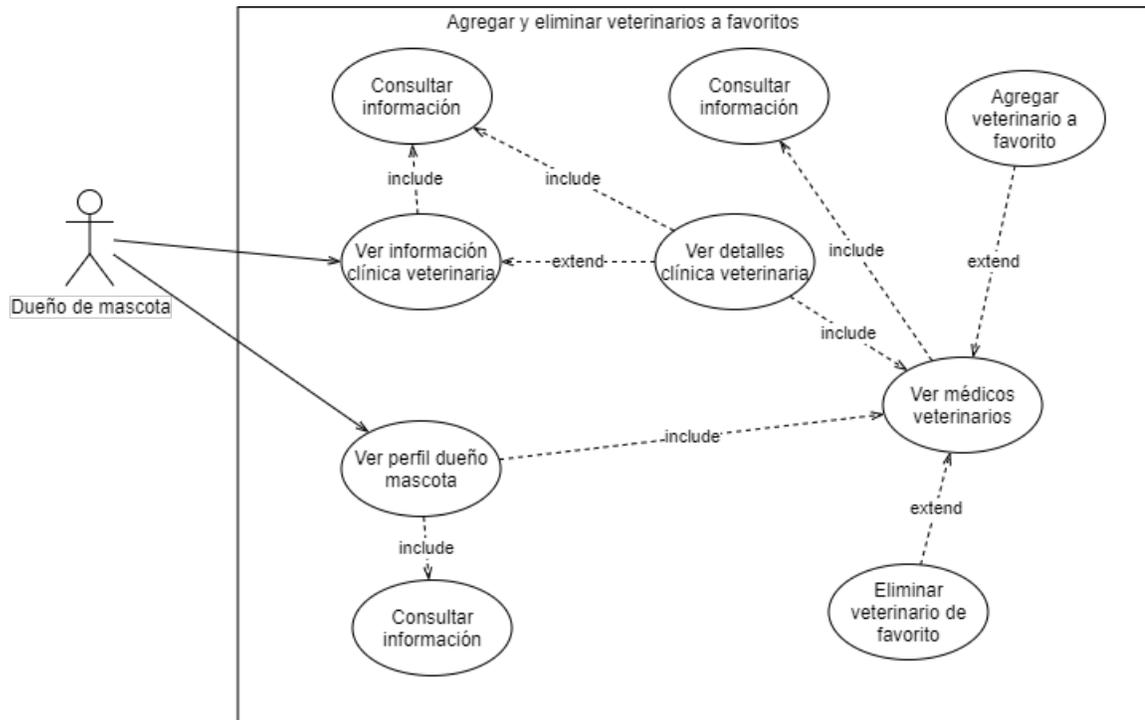


Figura D.17: Caso de uso 17

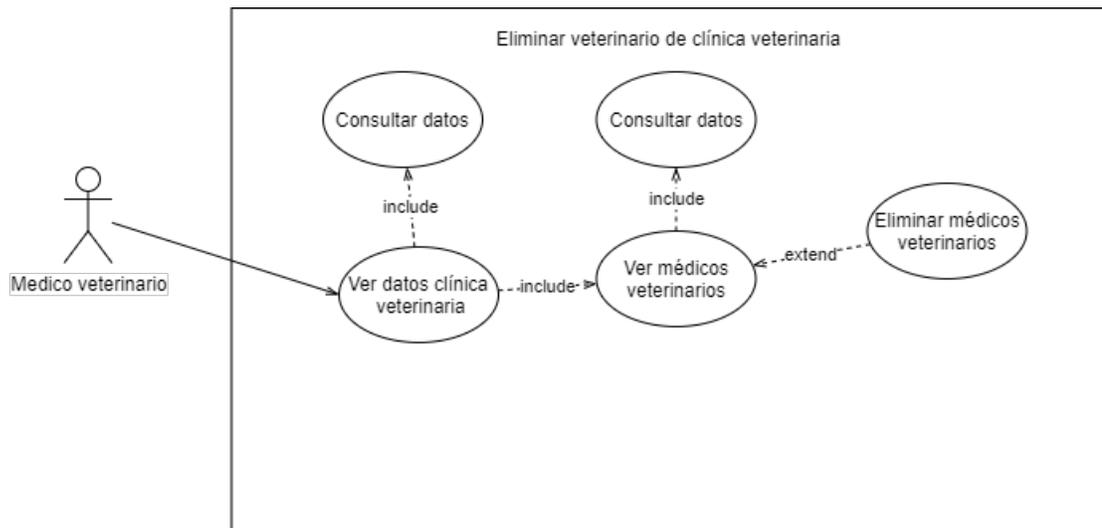


Figura D.18: Caso de uso 18

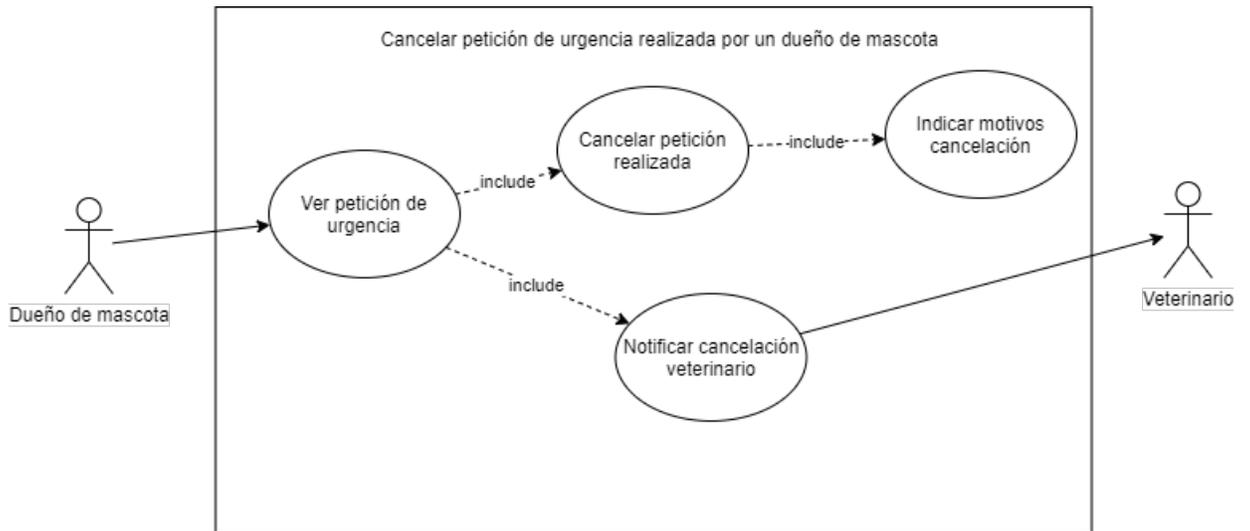


Figura D.19: Caso de uso 19

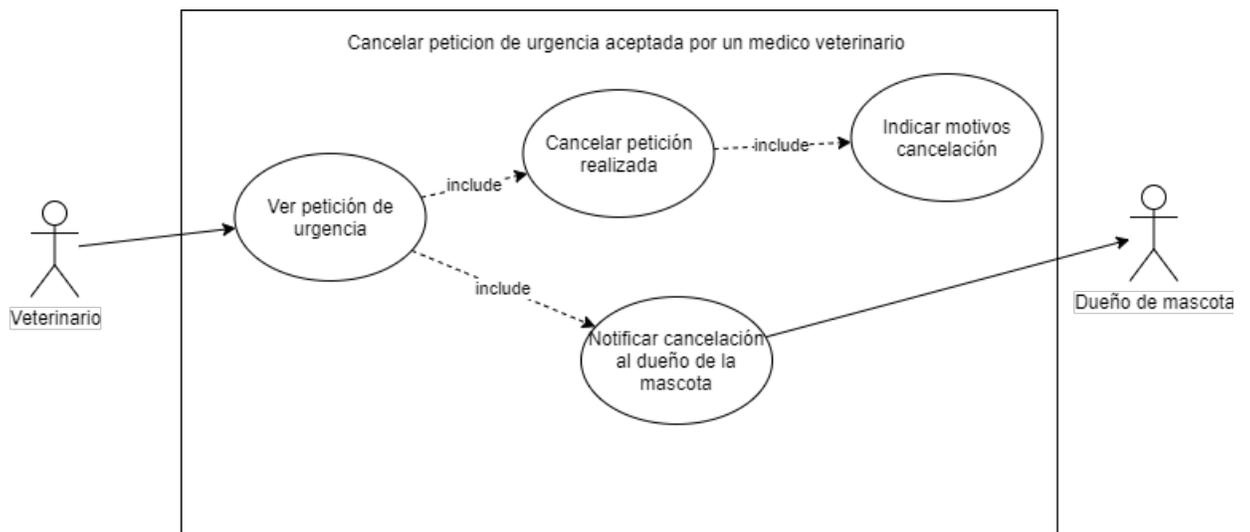


Figura D.20: Caso de uso 20

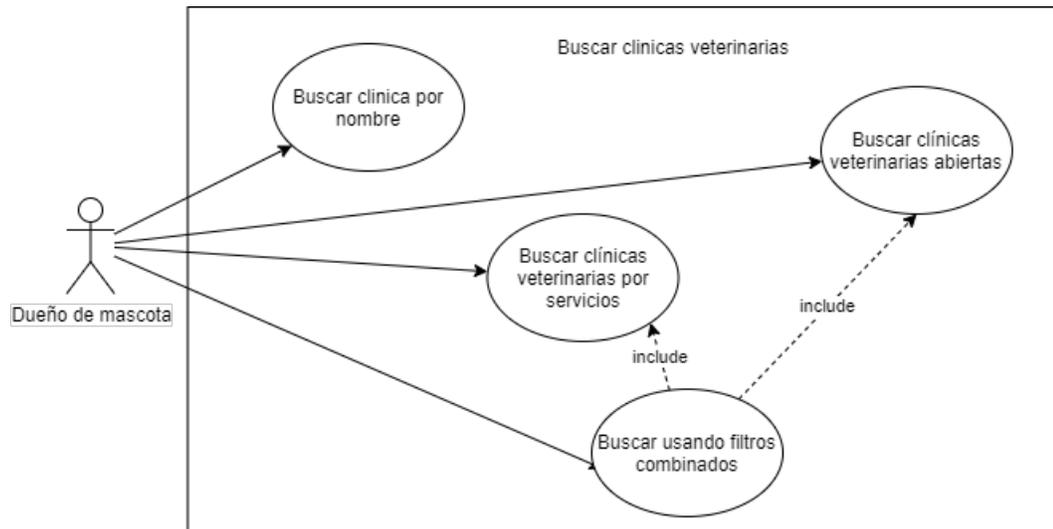


Figura D.21: Caso de uso 21

D.3. Requisitos del sistema

D.3.1. Requisitos de usuario

RU001 - Ingresar a la aplicación móvil como dueño de mascota
Descripción: Los dueños de mascota podrán ingresar a la aplicación móvil mediante un formulario de inicio de sesión que solicite un email como identificador de usuario y una contraseña
Fuente: Dueño de mascota
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.22: Requisito de usuario 1.

RU002 - Ingresar a la aplicación móvil como veterinario
Descripción: Los médicos veterinarios podrán ingresar a la aplicación móvil mediante un formulario de inicio de sesión que solicite un email como identificador de usuario y una contraseña.
Fuente: Veterinario
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.23: Requisito de usuario 2

RU003 - Ver mascotas que tiene un dueño
Descripción: El dueño podrá ver todas las mascotas que son de su propiedad.
Fuente: Dueño de mascota
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.24: Requisito de usuario 3

RU004 - Crear perfil de una mascota.
Descripción: El dueño de la mascota podrá crear el perfil de una mascota, suministrando información como el nombre, tipo de mascota, género, si esta esterilizada o castrado y una foto que identifique a la mascota.
Fuente: Dueño de mascota
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.25: Requisito de usuario 4

RU005 - Modificar perfil de una mascota.
Descripción: El dueño de la mascota podrá modificar el perfil de una mascota, actualizando información como el nombre, tipo de mascota, género, si esta esterilizada o castrado y la foto que identifique a la mascota.
Fuente: Dueño de mascota
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.26: Requisito de usuario 5.

RU006 - Compartir el perfil de una mascota.
Descripción: El dueño de la mascota podrá compartir la información de una mascota con otra persona, para que esta última pueda hacer peticiones de urgencia a veterinarios.
Fuente: Dueño de mascota
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización: 23-07-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.27: Requisito de usuario 6

RU007 - Seleccionar una mascota para la petición de urgencia.
Descripción: El dueño de la mascota podrá seleccionar entre sus mascota a cual de ellas irá dirigida la petición de urgencia.
Fuente: Desarrollador
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.28: Requisito de usuario 7

RU009 - Crear una petición de urgencia indicando los síntomas de la mascota
Descripción: El dueño de una mascota podrá crear una petición de urgencia dando información de los síntomas que presenta la mascota
Fuente: Dueño de mascota
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.29: Requisito de usuario 9

RU010 - Enviar una petición de urgencia de la mascota a todos los veterinarios que pertenezcan a una clínica veterinaria de la misma comuna del dueño
Descripción: El dueño de una mascota podrá crear una petición de urgencia solamente a los veterinarias que tiene agregados como favoritos.
Fuente: Desarrollador
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.30: Requisito de usuario 10

RU011 - Enviar una petición de urgencia de la mascota a veterinarios que el dueño tenga como favoritos
Descripción: El dueño de una mascota podrá crear una petición de urgencia solamente a los médicos veterinarios que tiene agregados como favoritos.
Fuente: Desarrollador.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 26-06-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.31: Requisito de usuario 11.

RU012 - Enviar una petición de urgencia a una clínica veterinaria en específico
Descripción: El dueño de una mascota podrá crear una petición de urgencia solamente a una clínica veterinaria para que solamente los veterinarios que trabajen en dicha clínica reciban la petición.
Fuente: Desarrollador.
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización: 26-08-2019
Estado: Cumple
Incremento: 2
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.32: Requisito de usuario 12

RU013 - Agregar una foto de la mascota a la petición de urgencia
Descripción: El dueño podrá agregar una foto de la situación de la mascota a la petición de urgencia.
Fuente: Dueño de mascota
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 23-07-2019
Estado: Cumple
Incremento: 2
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.33: Requisito de usuario 13

RU014 - Agregar un vídeo de la mascota a la petición de urgencia
Descripción: El dueño podrá agregar un vídeo de la situación de la mascota a la petición de urgencia.
Fuente: Dueño de mascota.
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 2
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.34: Requisito de usuario 14

RU015 - Editar una petición de urgencia creada anteriormente
Descripción: El dueño de una mascota podrá editar la información a la petición creada anteriormente modificando síntomas, video, imagen.
Fuente: Desarrollador.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 23-07-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.35: Requisito de usuario 15.

RU016 - Mostrar las peticiones de urgencia que el dueño de mascota ha creado
Descripción: El dueño de una mascota ver las peticiones de urgencia que él ha creado
Fuente: Desarrollador.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 23-07-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.36: Requisito de usuario 16

RU017 - Mostrar las peticiones de urgencia que el veterinario ha aceptado
Descripción: El médico veterinario podrá ver las peticiones de urgencia que él ha aceptado
Fuente: Veterinario.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 23-07-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Veterinario

Figura D.37: Requisito de usuario 17

RU018 - Mostrar ruta hacia la clínica veterinaria cuando la petición de urgencia es aceptada
Descripción: El dueño de una mascota podrá crear una petición de urgencia para una mascota en especial, indicando información de los síntomas de la mascota para que posteriormente llegue a los médicos veterinarios.
Fuente: Dueño de mascota
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 23-07-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.38: Requisito de usuario 18

RU019 - Visualizar en un mapa las clínicas veterinarias.
Descripción: La aplicación móvil mostrará en un mapa la localización de las clínicas veterinarias junto con información relevante para el dueño de una mascota.
Fuente: Dueño de mascota
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.39: Requisito de usuario 19

RU020 - Recibir petición de urgencia realizada por un dueño de mascota
Descripción: El veterinario podrá revisar la petición de urgencia junto con la información que esta petición trae.
Fuente: Veterinario.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 23-07-2019
Estado: Cumple
Incremento: 2
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.40: Requisito de usuario 20

RU021 - Aceptar petición de urgencia
Descripción: El veterinario podrá aceptar la petición de urgencia. Al aceptar la petición, se le enviará un aviso de vuelta al dueño de mascota sobre la confirmación realizada.
Fuente: Veterinario.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 23-07-2019
Estado: Cumple
Incremento: 2
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.41: Requisito de usuario 21.

RU022 - Crear clínica veterinaria
Descripción: El veterinario podrá crear una clínica veterinaria, donde se suministre información como el nombre, dirección, si atiende urgencias o no y una foto que muestre la clínica veterinaria.
Fuente: Veterinario
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.42: Requisito de usuario 22

RU023 - Agregar horarios a la clínica veterinaria
Descripción: El veterinario podrá agregar información de los días y las horas que la clínica veterinaria abre sus puertas para la atención.
Fuente: Veterinario
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 06-08-2019
Estado: Cumple
Incremento: 2
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.43: Requisito de usuario 23

RU024 - Agregar posición de la clínica veterinaria
Descripción: El veterinario podrá indicar la posición de la clínica veterinaria mediante un marcador de un mapa.
Fuente: Veterinario
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.44: Requisito de usuario 24

RU025 - Modificar clínica veterinaria
Descripción: El veterinario podrá modificar una clínica veterinaria, donde se suministre información como el nombre, dirección, si atiende urgencias o no y una foto que muestre la clínica veterinaria.
Fuente: Veterinario
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.45: Requisito de usuario 25.

RU026 - Modificar horarios a la clínica veterinaria
Descripción: El veterinario podrá modificar información de los días y las horas que la clínica veterinaria abre sus puertas para la atención.
Fuente: Veterinario
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.46: Requisito de usuario 26

RU027 - Modificar posición de la clínica veterinaria
Descripción: El veterinario podrá editar la posición de la clínica veterinaria mediante un marcador de un mapa.
Fuente: Veterinario
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.47: Requisito de usuario 27

RU028 - Asociar veterinario a una clínica veterinaria existente
Descripción: Un veterinario que no pertenece a ninguna clínica puede mandar un aviso al administrador una clínica veterinaria para asociarse a ella.
Fuente: Veterinario
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 26-08-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.48: Requisito de usuario 28

RU029 - Agregar médicos veterinarios a favoritos
Descripción: El dueño de mascota podrá agregar veterinarios a favoritos para que cuando cree una petición pueda seleccionar de sus veterinarios favoritos a cuales les llegará la petición.
Fuente: Desarrollador.
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización: 26-08-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.49: Requisito de usuario 29

RU030 - Eliminar médicos veterinarios de favoritos
Descripción: El dueño de mascota podrá eliminar veterinarios que tenga como favoritos.
Fuente: Desarrollador.
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización: 26-08-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.50: Requisito de usuario 30

RU031 - Mostrar médicos veterinarios asociados a una clínica veterinaria
Descripción: El veterinario administrador podrá eliminar un veterinario que trabaje en la clínica..
Fuente: Veterinario
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 26-08-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.51: Requisito de usuario 31.

RU032 - Eliminar médicos veterinarios asociados a una clínica veterinaria
Descripción: El veterinario administrador podrá eliminar un veterinario que trabaje en la clínica.
Fuente: Veterinario
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.52: Requisito de usuario 32

RU033 - Editar perfil de dueño de mascota
Descripción: El dueño de mascota podrá editar su información personal.
Fuente: Dueño de mascota
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota

Figura D.53: Requisito de usuario 33

RU034 - Editar perfil de médico veterinario
Descripción: El médico veterinario podrá editar su información personal
Fuente: Veterinario
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Médico veterinario

Figura D.54: Requisito de usuario 34

RU035 - Los usuarios deben proporcionar una contraseña con un número mínimo de caracteres
Descripción: El dueño de mascota y veterinario deben proporcionar una contraseña con una longitud aceptable.
Fuente: Desarrollador
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Restricción
Usuario asociado: Médico veterinario y dueño de mascota

Figura D.55: Requisito de usuario 35.

RU036 - La aplicación móvil sólo podrá ser usada en dispositivos que tengan sistema operativo Android
Descripción: El dueño de mascota y veterinario deben proporcionar una contraseña con una longitud aceptable.
Fuente: Desarrollador
Prioridad: Crítico
Estabilidad: No estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Restricción
Usuario asociado: Médico veterinario y dueño de mascota

Figura D.56: Requisito de usuario 36

RU037 - Cancelar una petición de urgencia
Descripción: El dueño de mascota podrá cancelar una petición de urgencia que haya realizado.
Fuente: Dueño de mascota
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 26-08-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota y veterinario.

Figura D.57: Requisito de usuario 37

RU038 - Agregar filtro de búsqueda de clinica veterinarias.
Descripción: El dueño de mascota podrá buscar de diferentes formas clínicas veterinarias
Fuente: Desarrollador
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 26-08-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado: Dueño de mascota.

Figura D.58: Requisito de usuario 38

D.3.2. Requisitos de sistema

RS001 - Implementar formulario registro de dueño de mascota
Descripción: Mediante el llenado de un formulario que solicite nombre, apellido, email, y contraseña del dueño de mascota, este quedará registrado en la aplicación móvil.
Prioridad: Crítico
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU001

Figura D.59: Requisito de sistema 1.

RS002 - Implementar formulario de ingreso dueño de mascota
Descripción: Los dueños de mascota podrán ingresar a la aplicación móvil llenando un formulario de ingreso mediante el email, como identificador único, y una contraseña.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU001

Figura D.60: Requisito de sistema 2.

RS003 - Implementar salir de la sesión dueño de mascota
Descripción: Los dueño de mascotas podrán cerrar la sesión en la aplicación móvil apretando un menú desplegable que se encontrará en la parte superior derecha de ésta.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU001

Figura D.61: Requisito de sistema 3.

RS004 - Implementar formulario registro médico veterinario
Descripción: Mediante el llenado de un formulario que solicite nombre, apellido, email, y contraseña del médico veterinario, éste quedará registrado en la aplicación móvil.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU002

Figura D.62: Requisito de sistema 4.

RS005 - Implementar formulario ingreso médico veterinario
Descripción: Los veterinarios podrán ingresar a la aplicación móvil llenando un formulario de ingreso mediante el email, como identificador único, y una contraseña
Prioridad: Crítico
Estabilidad: Estable
Fecha actualización: 15-09-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU002

Figura D.63: Requisito de sistema 5.

RS006 - Implementar salir de la sesión veterinario
Descripción: Los usuarios veterinario podrán cerrar la sesión en la aplicación móvil apretando un menú desplegable que se encontrará en la parte superior derecha de la aplicación móvil.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU002

Figura D.64: Requisito de sistema 6.

RS007 - Implementar listado con las mascota que el dueño tiene
Descripción: El dueño podrá visualizar en un listview las mascota que él posee. En el lista se mostrará el nombre y una foto de la mascota.
Prioridad: Crítico
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU003

Figura D.65: Requisito de sistema 7.

RS008 - Implementar formulario de creación de perfil de una mascota
Descripción: El dueño de mascota mediante la completación de un formulario podrá crear una mascota. El formulario tendrá campos como: nombre, tipo de mascota (perro o gato), género y una foto de la mascota.
Prioridad: Crítico
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU004

Figura D.66: Requisito de sistema 8.

RS009 - Implementar modificar perfil de una mascota.
Descripción: El dueño de la mascota podrá editar la información de esta, como por ejemplo: el nombre de la mascota, tipo de mascota, género, castrado o esterilizada y la foto.
Prioridad: Crítico
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU005

Figura D.67: Requisito de sistema 9.

RS010 - Indicar dueño de mascota al cual se quiere compartir el perfil de la mascota
Descripción: El dueño de una mascota podrá compartir la misma a otra persona indicando el email de ella mediante un campo de texto el cual será desplegado por un diálogo.
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización: 23-07-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU006

Figura D.68: Requisito de sistema 10.

RS011 - Notificar al dueño de mascota al cual se le quiere compartir el perfil de la mascota.
Descripción: Al dueño de mascota receptor se le enviará una notificación indicando que hay un dueño de mascota que quiere compartir el perfil. La notificación mostrará información básica de la mascota. El dueño receptor puede aceptar o rechazar la solicitud.
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización: 23-07-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU006

Figura D.69: Requisito de sistema 11.

RS012 - Seleccionar mascota para una petición de urgencia
Descripción: El dueño podrá seleccionar una mascota al cual quiere la petición de urgencia, para ello se desplegará un listview con todas las mascotas y mediante un click a ésta la podrá seleccionar para la petición.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU007

Figura D.70: Requisito de sistema 12.

RS013 - Implementar un campo de texto para agregar síntomas que la mascota posee.
Descripción: El dueño podrá escribir en un edittext los síntomas que presenta. El componente edittext debe poseer un scroll para un fácil chequeo de la información escrita.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU009

Figura D.71: Requisito de sistema 13.

RS014 - Obtener comuna del dueño de mascota al momento de enviar la petición de urgencia
Descripción: Se debe obtener la comuna del dispositivo móvil que usa el dueño de mascota al momento de enviar la petición
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU010

Figura D.72: Requisito de sistema 14.

RS015 - Obtener las clínica veterinarias que pertenecen a la comuna del dueño de mascota al momento de enviar la petición de urgencia
Descripción: Se debe obtener todas las clínicas veterinarias que pertenezcan a la comuna del dueño para que la petición de urgencia se envíe a los veterinarios que trabajan en dichas clínicas
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU010

Figura D.73: Requisito de sistema 15.

RS016 - Enviar la petición de urgencia a las clínicas veterinarias más cercanas respecto a la ubicación del dueño de la mascota.
Descripción: Se debe calcular la distancia en metros desde la ubicación del dueño de la mascota con las clínicas veterinarias. Las clínicas más cercanas se almacenarán al principio de un arreglo para que después se les envíe la notificación sobre la petición.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU010

Figura D.74: Requisito de sistema 16.

RS017 - Notificar a los médicos veterinarios de una petición de urgencia
Descripción: Se debe obtener los tokens de los veterinarios perteneciente a las clínicas veterinarias que pertenecen a la misma comuna donde el dueño de la mascota envía la petición de urgencia.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 15-05-2019
Estado: Cumple
Incremento: 1
Tipo:
Usuario asociado:
RU asociado: RU010

Figura D.75: Requisito de sistema 17.

RS018 - Seleccionar veterinarios favoritos para la petición de urgencia.
Descripción: Se desplegará una lista con los médicos veterinarios que un dueño de mascota tiene como favoritos y podrá seleccionarlos mediante un checkbox.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 26-08-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU011

Figura D.76: Requisito de sistema 18.

RS019 - Notificar a los médicos veterinarios que tiene el dueño de mascota como favoritos
Descripción: Se debe obtener los tokens de los veterinarios que un dueño de mascota tiene como favoritos para posteriormente hacer llegar la notificación al dispositivo que hace match con dicho token
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 26-08-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU011

Figura D.77: Requisito de sistema 19.

RS020 - Obtener médicos veterinarios que trabajan en la clínica veterinaria seleccionada por el dueño de mascota
Descripción: Se debe obtener los médicos veterinarios que trabajan en la clínica que el
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización: 26-08-2019
Estado: Cumple
Incremento: 1
Tipo: Funcional
Usuario asociado:
RU asociado: RU012

Figura D.78: Requisito de sistema 20.

RS021 - Notificar a los veterinarios de una clínica que el dueño de mascota seleccionó para enviar la petición de urgencia.
Descripción: Se debe obtener los <u>tokens</u> de los veterinarios que pertenecen a la clínica veterinaria que el dueño de mascota escogió para enviar la petición de urgencia.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU012

Figura D.79: Requisito de sistema 21.

RS022 - Tomar una foto usando la cámara del dispositivo móvil
Descripción: El dueño de mascota tendrá la opción de proporcionar una imagen de la mascota que necesita la atención mediante la cámara del celular.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU013

Figura D.80: Requisito de sistema 22.

RS023- Seleccionar una foto de la galería del dispositivo móvil
Descripción: El dueño de mascota podrá proporcionar una imagen de la mascota usando la galería de fotos del celular.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU013

Figura D.81: Requisito de sistema 23.

RS024 - Grabar un video con la cámara del dispositivo móvil
Descripción: El dueño de mascota podrá grabar con el celular un video para la petición de urgencia.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU014

Figura D.82: Requisito de sistema 24.

RS025 - Editar campos de la petición de urgencia
Descripción: El dueño de mascota podrá editar una petición de urgencia que haya creado. De la petición podrá editar los síntomas de la mascota, cambiar la imagen y/o video en caso que él haya proporcionado uno o ambos.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU015

Figura D.83: Requisito de sistema 25.

RS026 - Mostrar en un listview las peticiones enviadas por un dueño de mascota
Descripción: Listar en un listview las peticiones del dueño de mascota que haya enviado.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU016

Figura D.84: Requisito de sistema 26.

RS027 - Listar en un listview todas las peticiones de urgencia que un veterinario haya aceptado
Descripción: Mostrar en un componente listview las peticiones que un veterinario haya aceptado.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU017

Figura D.85: Requisito de sistema 27.

RS028 - Mostrar en un mapa la clínica veterinaria que atenderá la petición de urgencia
Descripción: El sistema debe mostrar en un mapa de Google Maps con un marcador la ubicación de la clínica que va a atender la urgencia.
Prioridad: Crítico
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU018

Figura D.86: Requisito de sistema 28.

RS029 - Mostrar la ruta hacia la clínica veterinaria
Descripción: Al presionar el mapa con la ubicación de la clínica, éste debe mostrar la ruta en vehicula hasta la veterinaria. Además debe mostrar el camino a seguir, el tiempo y la distancia.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU018

Figura D.87: Requisito de sistema 29.

RS030 - Mostrar en un mapa de Google maps las clínicas veterinarias registradas en el sistema
Descripción: En el mapa debe mostrarse mediante un marcador la ubicación de las clínicas veterinarias que están registradas en el sistema.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU019

Figura D.88: Requisito de sistema 30.

RS031 - Mostrar información básica de una clínica veterinaria en el marcador
Descripción: El usuario al presionar el marcador debe desplegar un diálogo en donde muestre información básica de la clínica, como por ejemplo: nombre de la clínica, dirección, si es cerrado o abierto.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU019

Figura D.89: Requisito de sistema 31.

RS032 - Recibir notificación de la petición de urgencia
Descripción: El veterinario debe recibir una notificación en el celular que le avise que hay una mascota que necesita una atención de urgencia.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU020

Figura D.90: Requisito de sistema 32.

RS033 - Ver detalle de la petición recibida
Descripción: El veterinario al presionar la notificación podrá revisar el detalle de la petición. En el detalle vendrá toda la información básica de la mascotas sumado a lo que el dueño de la mascota proporcionó para la petición (síntomas, imagen y/o video).
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU020

Figura D.91: Requisito de sistema 33.

RS034 - Aceptar petición de urgencia
Descripción: El veterinario tendrá la opción de aceptar la petición de urgencia recibida.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU021

Figura D.92: Requisito de sistema 34.

RS035 - Notificar la confirmación de la petición realizada por el dueño de la mascota
Descripción: El dueño de la mascota recibirá una notificación sobre la aceptación de la petición realizada.
Prioridad: Crítico
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU021

Figura D.93: Requisito de sistema 35.

RS036- Ver detalles de la confirmación de la petición de urgencia
Descripción: El dueño de la mascota al presionar sobre la notificación de confirmación podrá ver información como por ejemplo: el tiempo transcurrido desde que se envió la petición hasta cuando se aceptó, el veterinario que aceptó la petición, la clínica veterinaria que atenderá la urgencia.
Prioridad: Crítico
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU021

Figura D.94: Requisito de sistema 36.

RS037 - Crear clinica veterinaria
Descripción: El veterinario podrá crear una clínica veterinaria, para ello se requiere agregar datos como por ejemplo: el nombre de la clínica, dirección y una imagen distintiva de la clínica.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU022

Figura D.95: Requisito de sistema 37.

RS038 - Agregar horarios a la clínica veterinaria
Descripción: El veterinario puede seleccionar los días que atiende la clínica además de indicar los horarios de apertura y cierre.
Prioridad: Crítico
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU023

Figura D.96: Requisito de sistema 38.

RS039 - Agregar posición de la clínica veterinaria
Descripción: El veterinario podrá agregar la ubicación en un mapa de la clínica veterinaria.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU024

Figura D.97: Requisito de sistema 39.

RS040 - Modificar información de la clínica veterinaria
Descripción: El veterinario podrá editar el nombre, dirección y la imagen representativa de la clínica veterinaria.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU025

Figura D.98: Requisito de sistema 40.

RS041 - Modificar horarios de la clínica veterinaria
Descripción: El veterinario puede editar los horarios de apertura y cierre así como también los días en que atiende la clínica.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU026

Figura D.99: Requisito de sistema 41.

RS042 - Modificar la posición de la clínica veterinaria
Descripción: El veterinario puede cambiar la ubicación (latitud y longitud) de la clínica veterinaria.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU027

Figura D.100: Requisito de sistema 42.

RS043 - Mostrar opción para asociarse una clínica veterinaria
Descripción: El veterinario al presionar un marcador de una clínica veterinaria que está registrada en el sistema podrá ver un opción que le permite enviar un aviso al veterinario administrador de la clínica.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU028

Figura D.101: Requisito de sistema 43.

RS044 - Notificar al veterinario administrador de una clínica veterinaria sobre la solicitud de asociación
Descripción: El veterinario administrador recibirá una notificación en donde podrá ver la información del veterinario que quiere asociarse a la clínica veterinaria.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU028

Figura D.102: Requisito de sistema 44.

RS045 - Agregar veterinario a favorito
Descripción: El dueño de mascota podrá agregar veterinarios que trabajan en las clínicas veterinarias. Para agregar a favoritos, el dueño tendrá que ver los detalles de la clínica veterinaria.
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU029

Figura D.103: Requisito de sistema 45.

RS046 - Eliminar médicos veterinarios de favorito
Descripción: El dueño de la mascota puede remover veterinarios de su lista de favoritos.
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU030

Figura D.104: Requisito de sistema 46.

RS047 - Listar los veterinarios que pertenecen a una clínica veterinaria
Descripción: El veterinario administrador podrá listar a todos los veterinarios que trabajan en dicha clínica.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU031

Figura D.105: Requisito de sistema 47.

RS048 - Eliminar veterinarios que pertenecen a una clínica veterinaria.
Descripción: El veterinario administrador puede remover veterinarios que trabajan en una clínica veterinaria.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU032

Figura D.106: Requisito de sistema 48.

RS049 - Editar perfil del dueño de mascota
Descripción: El dueño de la mascota podrá modificar su información. Dentro de los campos que puede editar esta el nombre y el apellido.
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU033

Figura D.107: Requisito de sistema 49.

RS050 - Editar perfil del veterinario
Descripción: El veterinario podrá modificar su información. Dentro de los campos que puede editar esta el nombre y el apellido.
Prioridad: Deseable
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU034

Figura D.108: Requisito de sistema 50.

RS051 - Indicar motivos cancelación petición de urgencia
Descripción: El dueño de mascota podrá enviar un comentario breve al médico veterinario, para que éste conozca el motivo de la cancelación realizada por el dueño que creó la petición.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU037

Figura D.109: Requisito de sistema 51.

RS052 - Notificar al médico veterinario de la cancelación de la petición de urgencia.
Descripción: El veterinario le llegará una notificación indicando que un dueño de mascota ha decidido cancelar la petición de urgencia.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU037

Figura D.110: Requisito de sistema 52.

RS053 - Buscar clínicas veterinarias por nombre.
Descripción: El dueño de mascota podrá buscar por el nombre de una clínica veterinaria que pertenezca a su comuna.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU038

Figura D.111: Requisito de sistema 53.

RS054 - Buscar clínicas veterinarias que se encuentren abiertas.
Descripción: El dueño de mascota podrá buscar clínicas veterinarias que estén en su comunas y que se encuentren abiertas al público.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU038

Figura D.112: Requisito de sistema 54.

RS055 - Buscar clínicas veterinarias según servicios que estas ofrezcan.
Descripción: El dueño de mascota podrá buscar clínicas veterinarias que ofrecen servicios que el usuario ingresa al momento de realizar la búsqueda.
Prioridad: Crítica
Estabilidad: Estable
Fecha actualización:
Estado: Cumple
Incremento:
Tipo: Funcional
Usuario asociado:
RU asociado: RU038

Figura D.113: Requisito de sistema 55.

