Universidad
Inca Garcilaso de la Vega
Nuevos Tiempos. Nuevas Ideas
Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones

Aplicación móvil colaborativa para la búsqueda de mascotas caninas perdidas en Pueblo Libre mediante GPS y notificaciones por email y SMS

Tesis para optar el Título de Ingeniero de Sistemas y Cómputo

Héctor Manuel Reluz Llaque

Asesor

MSc. Ing. Hector Hernan Henriquez Taboada

Lima – Perú
Junio del 2022

Este trabajo está dedicado a mis tíos Carlos, Mónica, Pepe y Orestes, que siempre me han ayudado de forma desinteresada, sin esperar nada a cambio, para ellos este humilde trabajo.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

En la presente tesis se planteó una solución para el proceso manual que conlleva la publicación de una mascota perdida, así como herramientas GPS para optimizar la búsqueda, para lo cual se utilizó una aplicación móvil desarollada con React Native la cual facilita el proceso de desarrollo al permitir su exportación a los dos sistemas operativos móviles más usados, Android y iOS. Además, se usó PHP con el framework Laravel para dar el servicio de APIs que el aplicativo móvil consumiría. Se decidió hacer uso de la metodología Mobile-D para el desarrollo de la solución tecnológica por ser estar enfocada en el desarrollo móvil y ser sólida.

Palabras clave: mascota, móvil, gps, react native, php, mobile-d

ABSTRACT

In this work it was raised a solution for the manual process that most pets' owners must do when their pets are lost, and using GPS technology for optimizing the search. To implement everything mentioned before a mobile app was developed using React Native, a framework that helps to avoid the development for two different projects because it allows you to export from one to Android and iOS, furthermore, PHP was used with the Laravel framework to serve the API that the mobile app will consume. Mobile-D was chosen as a methodology for developing this solution, because it is primarily focused in the mobile development.

Keywords: pet, mobile, gps, react native. php and mobile-d

INTRODUCCIÓN

No es inusual para nadie ver anuncios en la calle con los rostros de mascotas que se encuentran perdidas con un número de contacto, lo verdaderamente inusual es observar que alguien tome nota de los datos de contacto del dueño.

Uno podría cruzarse con la mascota extraviada, pero al no recordar las características física de la mascota, y no tener un lugar donde poder confirmarlo rápidamente, se perdería información importante de donde fue vista la mascota por útlima vez, sin mencionar que el dueño podría estar buscando en un lugar o dirección equivocado.

Una aplicación móvil que permita la centralización de información con las características de la mascota, con las posiciones GPS para visualizar donde fue vista por última vez, así como los datos de contacto permitiría no solamente un fácil acceso, sino llegar a las personas que se encuentren cerca y con el interés de ayudar.

La presente tesis está constituída por los siguientes capítulos:

Capítulo I: Planteamiento del Problema; en este capítulo se detallará la problemática de las mascotas perdidas que es el motivo de la investigación, explicando el problema general, los objetivos generales y específicos, la justificación y los alcances.

Capítulo II: Marco Teórico; el segundo capítulo indica los antecendentes investigados que serán los referentes para la investigación, toda la información recolectada servirá para ampliar los beneficios y corregir las desventajas encontradas.

Capítulo III: Metodología de la Investigación; el tercer capítulo explica las razones de la elección de la metodología Mobile-D.

Capítulo IV: Desarrollo de la Solución Tecnológica; el penúltimo capítulo sigue paso a paso las fases de Mobile-D para llevar a cabo el desarrollo de la aplicativo móvil y del backend para las APIS, siguiendo a detalle las buenas prácticas brindadas por esta metodlogía especializada en el desarrollo móviles.

Capítulo IV: Validación de la Solución Tecnológica; el último capítulo aclara y puntualiza la validación de los objetivos del primer capítulo.

Para concluir, se indica las recomendaciones y conclusiones que podrían llevar a futuras investigaciones a una mejora, además de la bibliografía y anexos que se usaron.

CAPÍTULO 1: PLANTEAMIENTO DEL PROBLEMA

1.1 Situación Problemática

Los animales domésticos comúnmente conocidos como mascotas forman al día de hoy parte de muchas familias, y son considerados por muchos de ellos un integrante más de su núcleo.

Al día de hoy en el Perú, según Kantar (2018) la empresa líder mundial de datos, casi la mitad de los hogares cuenta con una mascota (49% de los hogares), tomando los datos de INEI (2017), podemos detallar que hay como mínimo 3,772,461 mascotas con hogar.

En el Perú, existen muchas instituciones, asociaciones y ONGS dedicadas a la protección de animales. Dentro de las instituciones se encuentra SERFOR (Servicio Nacional Forestal y de Fauna Silvestre), el cual tiene como rol principal ayudar a la gestión y administración de la flora y fauna silvestre en el país, RENIAN (Registro Nacional de Identidad Animal), una institución que tiene como meta el registro de animales sean domésticos o no. En las asociaciones nos encontramos con ASPPA (Asociación Peruana de Protección a los Animales), es una organización integrada por diferentes profesionales que comparten el objetivo de proteger animales de la crueldad humana, además está ARBA (Asociación para el Rescate y Bienestar de los Animales), la cual busca promover y sensibilizar a la sociedad para mejorar la calidad de vida de los animales, otra asociación es WUF, una asociación sin fines de lucro que busca mejorar la calidad de vida de los caninos a través de iniciativas sostenibles. Cabe mencionar que hay muchos albergues como Milagros Perrunos, Mishi Wasi, Vidas De 4 Patas, que se dedican a rescatar perros y gatos de la calle o situación de peligro.

No existe hoy en día en el Perú ninguna institución, asociación u organización sin fines de lucro que se preocupe o vele por la búsqueda de las mascotas perdidas, sólo existen algunas empresas privadas como SoyWako, Mascotaperdida112 o Buscandomimascota que a través de publicidad en redes sociales tratan de sectorizar la búsqueda.

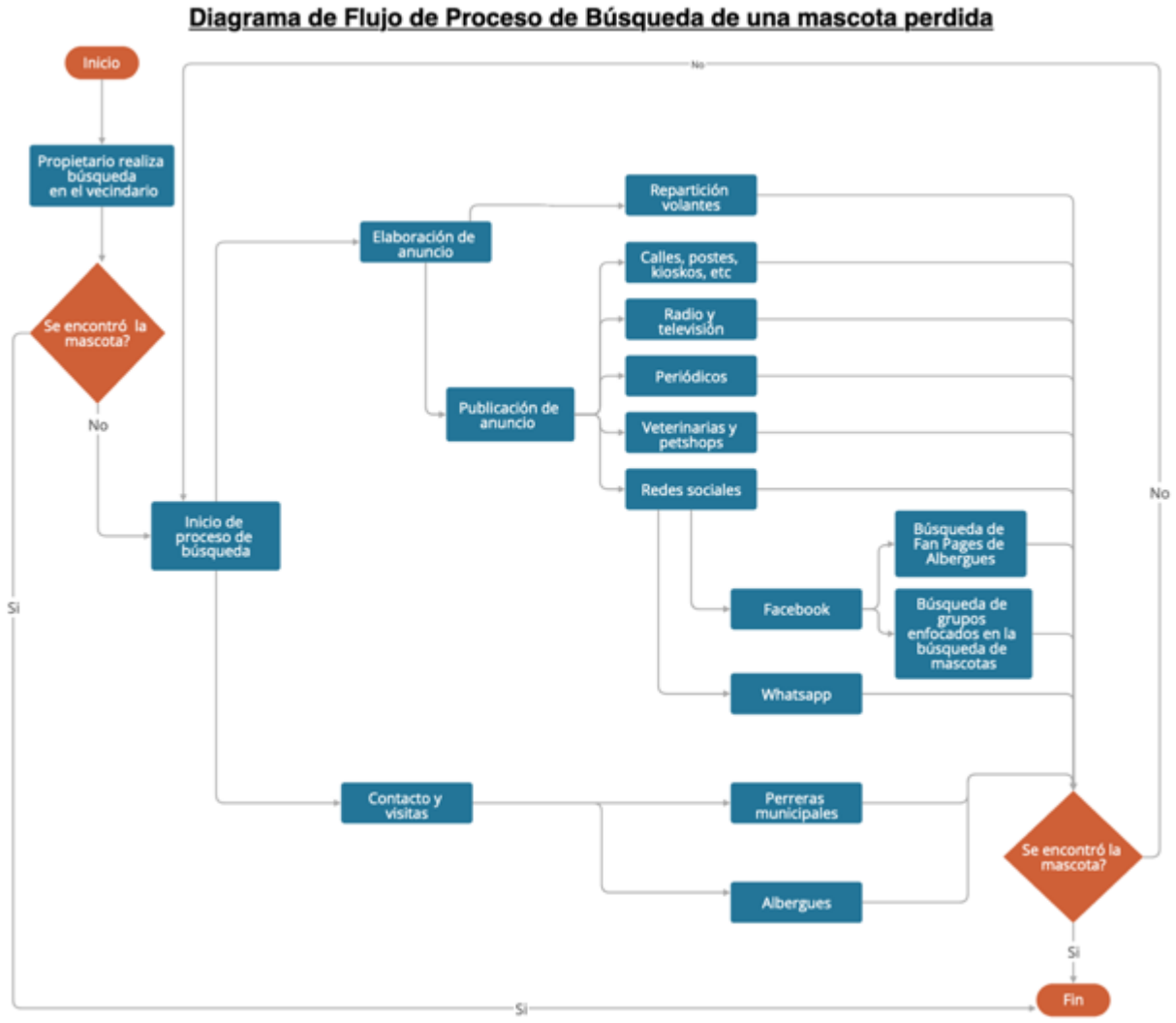
De acuerdo con AVMA (2021), la Asociación Médica Veterinaria Americana, 1 de cada 3 mascotas se pierden en algún momento de sus vidas, después de estos datos no suena tan increíble la información que Andina.pe (2018) comparte “Unas 420,000 mascotas, entre perros y gatos principalmente, se pierden anualmente en el Perú y son muy pocas las que son encontradas o devueltas a sus dueños, debido a que es muy difícil identificarlas.”.

Por ello, no es sorpresa la desesperación a la que se enfrentan cuando sus animales desaparecen y empiezan una búsqueda que pocas veces es exitosa debido a que, el alcance y difusión de sus anuncios o publicaciones en distintas redes sociales, no siempre llega a las personas interesadas en querer ayudar o dentro de un área de búsqueda específica de la mascota perdida, además del proceso de búsqueda que realizan (Ver figura 1.1) manualmente puede tomar y consumir mucho tiempo, y al pasar un cierto periodo deben realizarlo una y otra vez.

Además muchos de estos anuncios no llegan siempre al público objetivo, el cual muchas veces se encuentra fuera del radio geográfico de búsqueda de cada mascota. Sumado a esto, la poca rapidez de contacto en caso se reporte la mascota encontrada no se efectua a tiempo. Además de ello cuando sucede un avistamiento de una mascota, puede tomar un tiempo considerable hasta que contacte al dueño de la mascota, si es que ha visto el anuncio previamente, de lo contrario si ve el anuncio después, puede tomar muchísimo más, si el dueño es contactado se acercará al lugar específicado por el que brindó la información pero la mascota habrá cambiado de ubicación seguramente, entrando nuevamente en un círculo de esperar otro contacto e ir nuevamente al lugar brindado.

Figura 1-1

Problemática del proceso de búsqueda de mascotas



Fuente. Elaboración propia.

1.2 Formulación del Problema

El problema surge por la falta de eficacia con el proceso manual de difusión que conlleva el publicar los anuncios de las mascotas perdidas en diversos medios, tantos físicos como virtuales; lo cual consta de varios pasos que demandan gran cantidad de tiempo en su elaboración y constante actualización.

1.3 Objetivos

- General:

  Desarrollar una aplicación móvil para optimizar la búsqueda de mascotas perdidas en Pueblo Libre a través de puntos GPS para el rastreo de avistamientos.

- Específicos:

  - Permitir la creación de anuncios para mascotas perdida que sean visible por todos los usuarios de la aplicación.
  - Permitir registrar el avistamiento de una mascota para permitir trazar un recorrido de donde ha sido visto y cambiar el rango de búsqueda.
  - Notificar por email y SMS al dueño de la mascota en caso de un avistamiento.
  - Implementar la aplicación con la metodología MOBILE-D.

1.4 Justificación

La justificación de esta tesis reside en que sus resultados permitirán a través de la aplicación móvil centralizar la información de los anuncios de mascotas perdidas, así como facilitar la actualización del anuncio, notificar al dueño con avistamientos marcado por terceros, lo cual permitirá cambiar la dirección y rango de búsquda y conllevará a disminuir drásticamente el tiempo de búsqueda .
Hay 3 puntos importantes que justifican esta tesis el impacto social, económico y ecológico:

1. Social:

   Según Kat Albrecht (2018), un rescatista de personas que se dedicó a estudiar el comportamiento de gatos y perros perdidos explica que mucho de los perros que se pierden, incluso los que normalmente no son temerosos en casa se aterrorizan cuando se extravían. Mientras que algunos perros se podrán calmar cuando alguna persona se acerca, otros perros continuarán corriendo de todos, incluso de sus dueños.

   Esto podría lidiar a que el canino se aleje más aún del área de búsqueda y que al pasar las horas y días el animal por miedo o por hambre puede dañar personas otras mascotas o hasta enfermarse por no ser alimentado, perjudicando el aspecto social.

2. Económico:

   En el aspecto económico, mencionando nuevamente a Andina.pe Andina.pe (2018), “Unas 420,000 mascotas, entre perros y gatos principalmente, se pierden anualmente en el Perú”, esta cifra de mascotas extraviadas tiene un impacto en la economía porque altera la dieta de alimentos y otros insumos a los proveedores de estos productos.

3. Ecológico:

   En el aspecto ecológico, podemos destacar que los desechos de animales contaminan parques y calles.

   Para finalizar, podemos destacar que la aplicación móvil recogerá el proceso actual que tiene que pasar una persona para encontrar su mascota de forma manual y automatizarlo, también permitirá realizar una edición o actualización del mismo sin necesidad de actualizarlo en todos lados.

   Además, esta aplicación podría ser usada en cualquier ciudad del mundo.

1.5 Alcance

La implementación de la aplicativo móvil será para cualquier usuario, no es necesario tener una mascota para registrarse, ya que usuarios sin mascotas pueden también ayudar con las búsquedas, cualquier usuario.

Para el aplicativo móvil, se desarrollará utilizando el lenguaje de programación Javascript con el framework React Native, para las APIS se utilizará PHP con el framework Laravel y la base de datos será MySQL

Módulo de Usuarios: Este módulo permitirá el acceso de usuarios y el control de acceso al sistema.

- Login de usuarios.
- Registro de usuarios

Módulo de Mascotas: Este módulo permitirá a los usuarios realizar el registro y ver el listado de sus mascotas

- Registrar una mascota
- Listar mis mascotas

Módulo de Anuncios: Este módulo permitirá el registro y publicación de anuncios

- Registro de anuncios
- Visualización del anuncio

Módulo de Búsqueda: Este módulo permitirá mostrar a los usuarios las mascotas extraviadas cerca a su ubicación

- Listar mascotas perdidas
- Visualizar los avistamientos marcados en el mapa
- Reportar el avistamiento de mascota perdida
- Marcar una mascota como encontrada

Módulo de Notificaciones: Este módulo permitirá notificar

- Notificar avistamiento por SMS
- Notificar avistamiento por email

CAPÍTULO 2: MARCO TEÓRICO

2.1 Antecedentes de la investigación

- **GeoPetFinder: Aplicación para dispositivos móviles para la búsqueda de perros extraviados en la ciudad de Bogotá (Almario, M. y Rubiano, K., 2017)**

  En esta tesis se formuló el problema de investigación, la búsqueda de perros perdidos en Bogotá, el cual destaca el incremento de mascotas por parte de los ciudadanos, así como el aumento de caninos callejeros en proporción con el de mascotas perdidas.

  La solución que propone esta tesis es realizar un prototipo de aplicativo móvil que permite visualizar a través del API de Google Maps los reportes de los usuarios en la ciudad de Bogotá que facilitarían la búsqueda de perros perdidos.

  La conclusión detalla que es una aplicación fiable para las tareas pero que no es operacional en todo momento, es un prototipo.

- **Aplicación Móvil para adopción de mascotas abandonadas "PELUDITOS.COM" (Mendez, A., Villafañe, A., Martinez, J. y Criollo, J., 2019)**

  En esta tesis se formuló el problema de investigación, la búsqueda de perros perdidos en Bogotá detalla el aumento de perros callejeros en la ciudad, se estima que por cada 100 perros hay 38 que deambulan en las calles, los cuales presentan un peligro para la sociedad. Ante esta problemática el estado ha impulsado la adopción de estos canes, pero sin mayor éxito alguno.

  La solución que propone esta tesis es realizar una aplicación móvil que permita la adopción de los canes, los usuarios podrán tener acceso al detalle de cada canino y decidir cual de ellos adoptar.

  La conclusión detalla que se cumplió con el objetivo general a través de un aplicativo Android que permite la adopción en una app de fácil uso.

- **Desarrollo de un sistema web y móvil de registro y control de mascotas del gobierno autónomo descentralizado municipal de Otavalo, para las plataformas iOS y Android (Andrade, K., 2018)**

En esta tesis se formuló el problema de investigación, el registro de datos sobre mascota en el municipio de Otavalo se viene realizando de forma manual, lo cual conlleva mucho tiempo, además de tener información repetida y de difícil acceso.

La solución que propone esta tesis es realizar un sistema web y móvil para el registro de las mascotas del municipio.

La conclusión detalla que, gracias al API de Google, se logró implementar los módulos de Georreferenciación tanto en la web como en el móvil, SCRUM permitió ofrecer un producto de calidad, la digitalización de datos ha mejorado la gestión, la implementación de cada módulo logró optimizar los proceso. En general, gracias a un análisis de datos en una encuesta, se concluye que el sistema cumple con las expectativas.

- **Sistema web basado en la gestión de mascotas y su geolocalización en caso de extravío en la Municipalidad Distrital de Puente Piedra (Peña, J., 2020)**

  Se detalla en esta tesis el problema por el que muchos dueños de mascotas pasan en el distrito de Puente Piedra al perder sus mascotas, que son robadas y que no tuvieron éxito en su búsqueda.

  La solución que propone la tesis es contar con un sistema web que permita hacer la gestión de las mascotas y su geolocalización por parte de la municipalidad y medir su influencia.

  La conclusión detalla que la influencia fue positiva por la municipalidad para la captación de clientes, aunque no específica los clientes de que, también informa que el sistema web realizado hecho con la metodología SCRUM con éxito. Respecto al software elaborado, se pudo ver que opera con éxito la gestión de mascotas, además de mapas que muestran las ubicaciones de la mascota, pero no hayinformación de como se actualizará dicha información en el mapa. En general, parece ser positiva la herramienta en la municipalidad pero no se explica como se solucionaría el problema del extravío de mascotas

- **Mejorar el proceso de adopción en las mascotas de los albergues de lima mediante una aplicación web (Cortez, A., Ocares, C., 2020)**

  La anterior tesis mencionada trata un problema social muy desantedido que es la baja adopción de mascotas que impacta en el largo tiempo que se debe esperar a que sea adoptada, desencadenando gastos y costos para mantenerlo, parte de ese problema es realizado por la poca difusión de anuncios.

El trabajo propone como solución la creación de un sitio web donde se podrá gestionar de mejor manera la información de las mascotas para direccionarla hacia los correctos canalaes de difusión y aumentar la tasa de adopción.

Como conclusiones se informa de que aumentó la difusión a la que llegaron los anuncios de adoptar mascotas, y como consecuencia incrementó la cantidad de adopciones, todo esto a través de haceindo prioridades unas mascotas sobre otras dependiendo del tiempo en el que se encuentre en los albergues, y para terminar ayuda a brindar mejor información de mascotas a los solicitantes.

2.2 Marco conceptual

APLICACIÓN MÓVIL

Una aplicación móvil es una aplicación de software desarrollada para usar con un dispositivo móvil tales como Smartphone o una tablet. Una gran cantidad de personas piensa que las aplicaciones móviles son recientes, del siglo XXI basadas en smartphone. Sin embargo, datan de los ochenta:

Tabla 2-1

Progreso del aplicativo móvil por años

1984	Se vio el lanzamiento de Psion Organiser (una computadora de bolsillo). Este fue el "World's First Practical Pocket Computer", la cual estaba cargada con aplicaciones como una calculadora y un reloj
1994	IBM introdujo "Simon". Este asistente personal digital podía recibir y enviar faxes y correos. También tenía aplicaciones tales como un directorio, calendario y agenda de reuniones. Muchas personas piensan que este fue el primer smarthpone en el mundo.
2002	Blackberry lanzó su primer smarthpone. Este dispositivo agregó la función de email. Rápidamente este requerimiento se convirtió en algo indispensable en todos los dispositivos
2007	iPhone lanzó iPhone. Al siguiente año desplegaron el App Store. Inicialmente tuvo 500 aplicaciones, 25% de estas eran gratuitas de descargar. El cual fue inmediatamente un éxito. En las primeras 72 horas, fueron descargadas 10 millones de aplicaciones. Google Play Store fue lanzado unos meses después. (Inicialmente llamada como Android Market).

Fuente. Elaboración propia. Tomado de Outsystems (2020)

Esto señaló el inicio del fenómeno de aplicaciones móviles que conocimos hoy.

Existen tres tipos de aplicaciones móviles:

- Aplicaciones Nativas
- Aplicaciones Web Progresivas
- Aplicaciones Hibridas

1. Aplicaciones Nativas

   Una aplicación nativa está hecha y escrita para una plataforma específica o dispositivo (iOS en el caso de Apple o Android en el caso de Google son prácticamente el 99% de dispositivos móviles).

   Debido a esto, las aplicaciones nativas sacan una gran ventaja de las especificaciones de hardware y software (como por ejemplo la cámara). Estas son usualmente de alto rendimiento y ofrecen al usuario del dispositivo móvil una mejor experiencia.

   Sin embargo, el usuario debes descargarla la aplicación desde la tienda de la plataforma. Esto normalmente es una barrera para algunas personas. Las aplicaciones móviles, por definición no son multiplataforma, requieren un desarrollo único para sistema. Así, las aplicaciones iOS son escritas en Objetive-C o Swift, mientras que las aplicaciones Android usan Java. Esto presenta una complejidad en el desarrollo ya sea por el costo de contratar un desarrollador para cada plataforma o el costo. Outsystems (2020)

2. Aplicaciones Web Progresivas

   Una aplicación web móvil necesita una web que funcione en un navegador puede ser adaptada. A diferencia de las aplicaciones nativas, las aplicaciones web progresivas (PWAs) funcionará en cualquier plataforma móvil. Como las aplicaciones nativas, las PWAs funcionan offline, permiten el envio de notificaciones, y acceso al hardware del dispositivo, tales como las cámaras o GPS. La experiencia de usuario es similar a las aplicaciones nativas en móvil y dispositivios de escritorio sin necesidad de descargar, con gran beneficio, las PWAs corren bien con una conexión lenta.

   Para añadir, desde la perspectiva del usuario final, las PWAs son fáciles de usar en dispositivos móviles y son ligeras. Y desde la perspectiva de un desarrollador, son más fáciles de cambiar que las aplicaciones nativas, y son más fácil de mantener. Además, a diferencia de las aplicaciones nativas, usan solamente un repositorio de código para todos los dispositivos . Outsystems (2020)

3. Aplicaciones Híbridas

   Una aplicación híbrida como su nombre sugiere combina aspectos de nativo y web apps. Se puede pensar de las híbridas como web apps puestas en un contenedor de una aplicación nativa. Como las aplicaciones nativas e híbridas deben ser descargadas desde la tienda. Una vez instalada en la aplicación móvil, el shell usa un navegador integrado para acceder a las

capacidades de la plataforma nativa. Las aplicaciones híbridas tiene muchas ventajas, y algunas limitaciones. Outsystems (2020)

Como las web apps, las aplicaciones híbridas son más fáciles de desarrollar que las nativas. Las nativas además no necesitan ser escritas en una plataforma específica. Ambos factores la hacen mucho más fáciles y baratas de desarrollar una aplicación híbrida. Esta puede ser algo importante a considerar ya que mucha oferta en el mercado de desarrolladores que no se puede satisfacer. Un beneficio de la aplicación híbrida es que no necesita validación una vez que es aprobada por la tienda. Esto significa que cualquiera actualización no actualiza el código nativo. Sin embargo, las aplicaciones híbridas no ofrecen una experiencia de usuario del nivel que una nativa. Además no puede tomar todas las ventajas de las características de una plataforma móvil. Outsystems (2020)

MASCOTA

Una mascota, o un animal de compañía, es un animal que se tiene principlamente para la compañía de una persona o entretenimiento, a diferencia de un animal de trabajo, ganado o laboratorio. Las mascotas populares son consideradas por su apariencia, inteligencia o sus personalidades fácilmente identificable, pero algunas mascotas son aceptadas por motivos altruistas y aceptadas por un dueño sin importar ninguna de estas.

Dos de los más populares mascotas son perros y gatos. Otros animales de compañía conocidos son conejos, hurones, cerdos, roedores, tales como jerbos, hámsteres, chinchillas, ratas, ratones y cobayos; mascotas digitales como tamagotchis; mascotas aves como loros, paseriformes o aves de corral; mascotas reptiles tales como tortugas, caimanes, cocodrilos, lagartijas y serpientes; mascotas acuáticas como peces, caracoles, anfibios como ranas y salamandras; y mascotas artrópodas como tarántulas y cangrejos ermitaños. Pequeñas mascotas pueden ser agrupadas como mascotas de bolsillos, mientras las equino y bovino agrupa animales más grandes de compañía.

Las mascotas proveen a sus dueños o guardianes beneficios físicos y emocionales. Caminar con un perro provee de ejercicio a ambos tanto el perro como al humano, aire fresco e interacción social. Mascotas puede dar compañía a la gente quienes vivan solos o gente adulta quienes no tienen interacción social con otras personas. Está medicamente probado clases de terapia con animales, mayormente perros o gatos, que son traídos para visitas para humanos confinados, como niños en hospitales o adultos mayores en casas de reposo. En la terapia de mascotas se utiliza animales entrenados para tratar diferentes tipos de objetivos como físicos, social, cognitivo o emocionales con los pacientes. (Wikipedia, 2020)

API

Una interfaz de programación de aplicación, o API, permite a compañías abrir los datos de sus aplicaciones y la funcionalidad de sus sistemas a desarrolladores externos, socios, y a áreas internas de la misma compañía. Esto permite a los productos y servicios comunicarse entre ellos y tomar ventaja de la información de otros y de la funcionalidad a través de sus documentos de interfaz. Los desarrolladores no necesitan saber como una API está implementada; ellos simplemente necesitan usar la interfaz para comunicar con otros productos y servicios. La API ha surgido en la pasada década, hasta el punto de que muchas de las web más populares de hoy en día no serían posibles de funcionar sin las APIs. Las API funcionan de la siguiente manera:

1. Un cliente inicia la llamada al API para obtener la información, también conocida como una petición. Esta petición es procesada desde una aplicación hacia el servidor a través de la API e incluye acciones de verbo (GET, POST, PUT), encabezados y en el algunos casos información en la parte del cuerpo de la petición.
2. Después de recibir una petición válida, la API hace la llamada a un programa externo o al servidor
3. El servidor envia la respuesta al API con la información requerida.
4. La API transfiere los datos a quien hizo la petición

(IBM, 2020)

MOBILE-D

Es una metodología orientada al desarrollo de aplicativos móviles, creada por Abrahamsson en 2004, surgió como una alternativa a las metodologías que no cumplían las expectativas para el desarrollo de aplicaciones móviles, el que era un desafío en aquel momento. Cabe resaltar además que para ese momento se sabía muy poco del ciclo de vida de los móviles. Debido a estas circunstancias se creo esta metodología.

Hoy en día básicamente cualquiera con las habilidades y conocimiento suficiente puede hacer desarrollo móvil, esto sin embargo requiere de un amplio conocimiento de las características y los desafíos que presentan las aplicaciones para móviles.

Las metodologías ágiles podrían parecer perfectas para el desarrollo móvil, sin embargo, las características de los móviles presentan algunas restricciones las actuales metodologías ágiles. El enfoque de la metodología Mobile-D propone solucionar esos problemas.

Para superar los desafíos presentados en el desarrollo móvil, Abrahamson y su equipo crearon Mobile-D. La propuesta está basada en las metodologías Extreme Programming (XP), Crystal y Rational Unified Process (RUP).

El objetivo de Mobile-D está optimizada para un equipo de menos de diez desarrolladores trabajando en un espacio juntos con el objetivo de entregar un aplicación móvil totalmente

funcional en un corto tiempo. Mobile-D ha sido desarrollada con la cooperación de tres compañías de desarrollo móvil. (VTT, 2004)

Un proyecto de desarrollo que sigue la metodología de Mobile-D está dividido en cinco iteraciones. Estas fases son: setup, core, core2, stabilize y wrap-up. En la traducción y práctica estas fases se han traducido y adaptado como:

1. Exploración

   El propósito de la fase de exploración es el planeamiento y establecimiento del proyecto a desarrollar. “Un buen planeamiento, es un trabajo prácticamente hecho al 50%” es un dicho en el mundo del desarrollo de software. La fase de exploración puede sin ningún problema hecho tanto al inicio como al final de las fases de Mobile-D e incluso puede cruzarse con la fase Iteración 0. La fase de exploración es una fase importante que establece los cimientos para una controlada implementación del proceso de desarrollo de software y la selección de ambientes. Diferentes grupos de stakeholders son indispensable para proveer una adecuada fase de exploración.

   a. Establecimiento del clientes

      El propósito de esta etapa es identificar y establecer los grupos de stakeholders que serán necesarios en varias tareas de la fase de exploración así como en soportar actividades durante el desarollor de software - excluyendo el equipo de desarrollo en sí - . La amplia variedad de experiencia y cooperación es necesaria para planear controladamente y hacer una implementación efectiva del producto

   b. Definición de Alcance

      El propósito de esta etapa es definir las metas para el actual proyecto respecto a los contenidos así como el tiempo y duración del mismo.

   c. Estblecimiento del proyecto

      El propósito de esta etapa es la de asignar recursos (tanto técnicos como humanos) necesarios para el desarrollo del proyecto. También establecer el proceso de referencia para el proyecto es una tarea importante. La fase del establecimiento del proyecto se realiza en orden de estar seguros que el equipo del proyecto puede iniciar el desarrollo de software sin retrasos causados por ekemplo falta de herramientas o el apropiado entrenamiento

2. Inicialización

   El propósito de la fase de inicialización es establecer el éxito de las fases del proyecto a realizar preparando y verificando todos los problemas críticos del desarrollo para que esten a disposición al final de la fase para implementar los requerimientos seleccionados por el cliente.

   a. Inicio de Proyecto

      El propósito de esta estapa es para

1. Establecer los recursos físicos y técnicos para el proyecto así como el ambiente para la monitozación del proyecto.
2. Entrenar al equipo del proyecto si es requerido y
3. Establecer las formas específicas del proyecto para comunicar con el grupo del cliente. Todas las tareas del inicio del proyecto incluyen la partcipación del equipo.

b. Día de Planeamiento en Iteración 0

El propóstio de esta fase es el inicio del proyecto es ganar un buen entendimiento del proyecto a desarrollar, para preparar y mejorar los planes para las fases del proyecto y preparar los planes para verificar y solucionar los problemas críticos al final de esta fase.

c. Día de Trabajo en Iteración 0

El propósito de esta etapa es probar y configurar aún más el entorno de desarrollo y asegurar que todo este listo para implementar el producto. También, el propósito es implementar una funcionalidad importante del sistema o resolver algún problema crítico de desarrollo sin producir código. También hacer investigaciones sobre tecnología son posibles en esta etapa. Si el equipo decide implementar alguna funcionalidad a este punto, no necesita ser algo con la mayor prioridad que ha sido definida por el cliente sino ser seleccionada por la importancia que tiene, por ejemplo la estructura de arquitectura del producto.

d. Día de Lanzamiento en Iteración 0

Este procedimiento es más general.y depende enormemente del proyecto y del ambiente donde el proyecto será implementado. Así, en Mobile-D no hay uan descripción específica para este punto

3. Producción

El propósito de la fase de producción es implementar los requerimientos del producto para implementar la funcionalidad de los requerimientos aplicando un ciclo de desarrollo iterativo.

a. Día de planeamiento

El propósito en el día de planeamiento es seleccionar y planear el contenido del trabajo para la iteración. Participando activamente para planear actividades, el cliente asegura que los requerimientos proveen valor al negocio para identificar esos requerimientos y sean entendidos correctamente.

b. Día de Trabajo

El propósito de esta etapa es hacer un entregable de un producto totalmente funcional .

c. Día de Lanzamiento

El propósito de esta etapa es hacer un lanzamiento de un producto totalmente funcional .

4. Estabilización

En esta fase requiere de que el sistema sea dividido en pequeños subsistemas. En el caso de un proyecto multi-equipo, el propósito de la tarea es integrar los subsistemas que son generaados por distintos equipos para ser unidos en solo producto.

a. Día de planeamiento

El propósito en el día de planeamiento es seleccionar y planear el contenido del trabajo para la iteración. Participando activamente para planear actividades, el cliente asegura que los requerimientos proveen valor al negocio para identificar esos requerimientos y sean entendidos correctamente.

b. Día de Trabajo

El propósito de esta etapa es hacer un entregable de un producto totalmente funcional .

c. Documentación Wrap-up

El propósito de tarea es producir documentación. Software sin documentación es un desastre. El código fuente es no el medio ideal para comunicar la estructura e interfaces del sistema. La documentación será producida para los stakeholders del proyectos y no para el equipo ágil.

d. Día de Lanzamiento

El propósito de esta etapa es hacer un lanzamiento de un producto totalmente funcional .

5. Prueba y Arreglo

El propósito del sistema de pruebas y arreglos es ver que el producto implementa correctamente las funcionalidades requeridas por el cliente, lo cual proveerá al equipo de feedback para hacer los arreglos de los errores encontrados.

a. Sistema del Testeo

El propósito de esta tarea es encontrar los defectos en producto del software después de la fase de implementación. El procedimiento del testeo provee la información de errores para la última fase del proceso de Mobile-D

b. Día de Planeamiento

El propósito en el día de planeamiento es seleccionar y planear el contenido del trabajo para la iteración. Participando activamente para planear actividades, el cliente asegura que los requerimientos proveen valor al negocio para identificar esos requerimientos y sean entendidos correctamente.

c. Día de Trabajo

El propósito de esta etapa es hacer un entregable de un producto totalmente funcional .

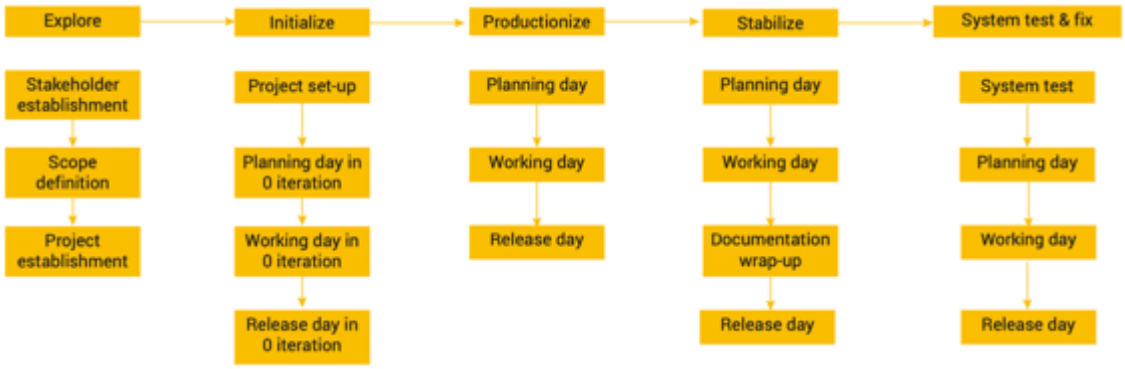
e. Día de Lanzamiento

El propósito de esta etapa es hacer un lanzamiento de un producto totalmente funcional .

(VTT, 2004)

Figura 2-1

Ciclo de desarrollo de Mobile-D



Fuente. Las cinco iteraciones de desarrollo de la metodología Mobile-D. Tomado de Amaya, Y (2013) Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles.

Cada fase consiste en tres diferentes tipos días de desarrollo: Día de Planeamiento (Planning Day), Día de Trabajo (Working Day) y Día de Lanzamiento (Release Day). Si multiples equipos están trabajando en un mismo requerimiento o producto, un Día de Integración (Integration Day) es también necesario. (VTT, 2004)

CAPÍTULO 3: MÉTODOLOGÍA DE LA INVESTIGACIÓN

3.1 Método

La metodología que se usará para esta investigación será MOBILE-D, una metodología que combina las prácticas de las metodologías tradicionales y ágiles para el desarrollo móvil.

Mobile-D permite llevar un desarrollo iterativo y re-adaptable, en el que es posible empezar con pequeños requerimientos, pero teniendo un producto con nuevas mejoras después de cada fase, además de una documentación de lo planeado, manteniendo con híbrido entre las metolodías ágiles y tradicionales gracias a las 5 fases con las que cuenta Mobile-D, las cuales no son obligatorias sino adaptables al desarrollo de cada proyecto.

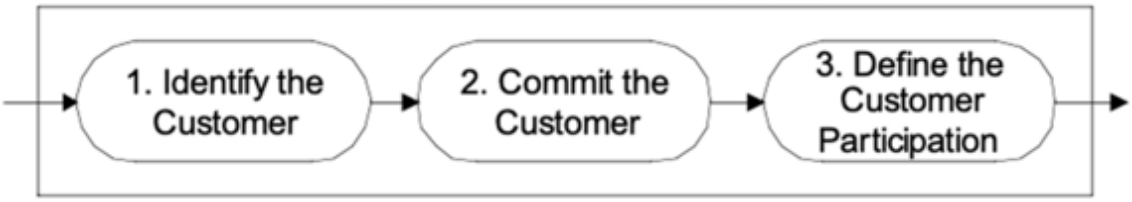
3.2 Adaptación de la Metodología

1. Fase de Exploración
   1.1. Definición de los Stakeholders
      1.1.1. Establecimiento de Clientes

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-1

Procedimiento de Establecimiento de Clientes (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-1

Objetivos del Establecimiento de Clientes (Mobile-D)

1. Identificar los clientes que participarán	Establecer los clientes que formarán parte de las fases de desarrollo. Los clientes deberán ser personas con un conocimiento tácito que ayudarán brindando la información para cada requerimiento.
2. Comprometer al cliente	Es esencial que el cliente sea parte de todo el ciclo de desarrollo, comprometerse a

	seguir todas las fases es esencial para conseguir un software de calidad.
3. Definir el modo de roles y responsabilidades	Se deberá comunicar al cliente las tareas y papel que desenvolverá en el ciclo

Fuente. Elaboración propia

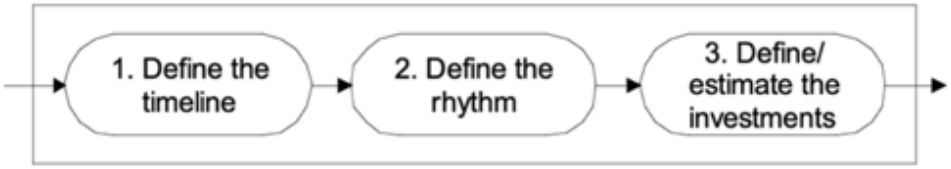
1.2. Definición del Alcance

1.2.1. Colección Inicial de Requerimientos

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-2

Procedimiento de Colección Inicial de Requerimientos (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-2

Objetivos de la Colección Inicial de Requerimientos (Mobile-D)

1. Definir el cronograma	Se deberá definir las actividades que se realizarán, y que fases se adaptarán
2. Definir el ritmo	Teniendo en cuenta el cronograma ser deberá especificar el tiempo que tomará realizar cada una para ajustarlos en iteraciones.
3. Definir/estimar las presupuestos	Detallar los gastos que se estará invirtiendo en desarollar esta solución tecnológica

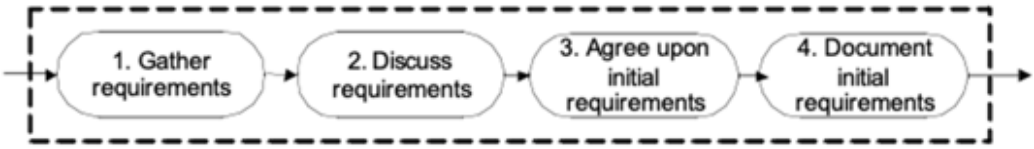
Fuente. Elaboración propia

1.2.2. Planeamiento Inicial del Proyecto

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-3

Procedimiento del Planeamiento Inicial del Proyecto (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-3

Objetivos del Planeamiento Inicial del Proyecto (Mobile-D)

1. Recoger los requerimientos	1. Crear el backlog de los requerimientos funcionales 2. Crear los mockups del aplicativo 3. Desarrollar el backend con los APIS para consumir por la aplicación 4. Desarrollar la aplicación móvil con los requerimientos definidos previamente 5. Realizar las pruebas y hacer los arreglos necesarios
2. Discutir requerimientos	Se discutirán con el dueño de una mascota, para asegurar los requerimientos sean los adecuados.
3. Acordar los requisitos iniciales	El dueño de la mascota identificará los requisitos más relevantes.
4. Documentar los requerimientos iniciales	Despues del acuerdo con el dueño se documentará todo para tener un backlog.

Fuente. Elaboración propia

1.3. Establecimiento del Proyecto

1.3.1. Selección del Ambiente

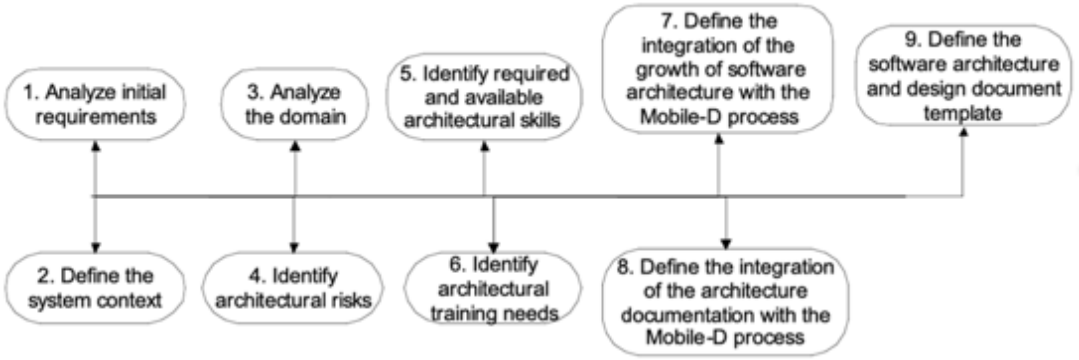
Se deberá describir el hardware, software que se usará para el desarollo.

1.3.2. Definición de Línea de Arquitectura

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-4

Procedimiento de la Definición de Línea de Arquitectura (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-4

Objetivos del Planeamiento Inicial del Proyecto (Mobile-D)

1. Analizar los requerimientos iniciales	Crear los mockups de la aplicación con un flujo básico que demuestre el objetivo global.
2. Definir el contexto del sistema	El aplicativo se desarrollará para iOS y Android
3. Analizar el dominio	Para la arquitectura se usará la Arquitectura React Native Como patrón Se usará el principio SOLID
4. Identificar los riesgos de arquitectura	No tener en cuenta las restricciones y las acciones por roles
5. Identificar habilidades arquitectonicas requeridas y disponibles	Estar familiarizado con el desarrollo de software para móviles y creación de APIs
6. Identificar las necesidades para el entrenamiento	La documentación de los frameworks Laravel y React Native cubrirá estas necesidades
7. Definir la integración del crecimiento gradual sistemático de la	La primera historia deberá ser una pantalla de la aplicación funcional que conecte con la API

	arquitectura de software con los procesos de Mobile-D	
8.	Definir la integración de la documentación de la arquitectura con el móvil	Documentación funcional de los flujos principales de la aplicación
9.	Definir una plantilla para la arquitectura de software	No aplica porque la arquitectura de software ya es brindada por el framework

Fuente. Elaboración propia

Herramientas: Microsoft Word, Balsamiq

Técnica: Levantamiento de Información

2. Fase de Inicialización

2.1. Configuración del Proyecto

2.1.1. Configuración del Ambiente

Por el lado del backend deberá definirse

- Sistma operativo y versión
- Servidor web y versión
- Framework, lenguaje y versión de ambos
- Base de datos y versión

Por el lado del móvil

- Framework, lenguaje y versión de ambos

2.1.2. Entrenamiento

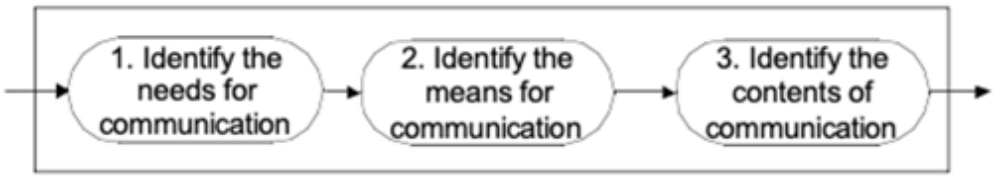
Se deberá contar con la documentación de los lenguajes, frameworks, metodologías que sean accessibles y fáciles de consultar.

2.1.3. Establecimiento de Comunicación con el Cliente

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-5

Procedimiento del Establecimiento de Comunicación con el Cliente (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Figura 3-6

Procedimiento del Establecimiento de Comunicación con el Cliente (Mobile-D)

1. Identificar las necesidades para la comunicación	Deberá acordarse con el cliente como se dará el seguimiento de los requerimientos y de que forma podrá visualizarlo. haya un nuevo feature.
2. Identificar las maneras por la comunicación	Deberá definirse por que medio se estará enviando los entregables. La comunicación será por correo
3. Identificar los contenidos de la comunicación	Deberá definirse que contenidos se entregará.

Fuente. Elaboración propia

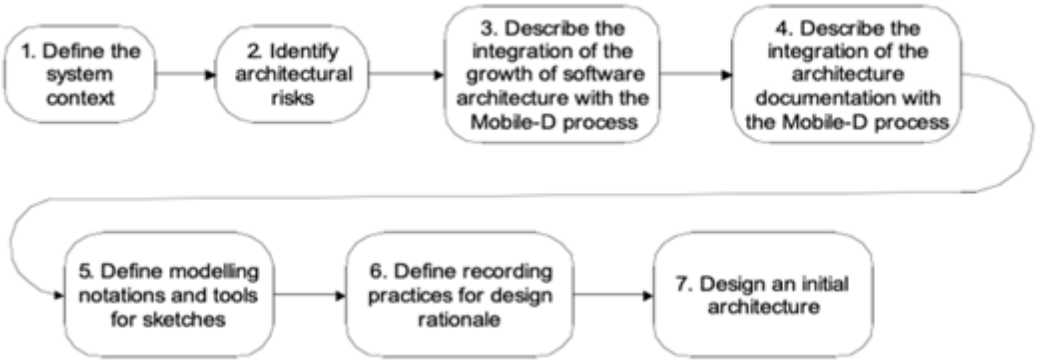
2.2. Día de Planeamiento en Iteración 0

2.2.1. Planificación de la Línea de Arquitectura

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-7

Procedimiento del Día de Planeamiento en Iteración 0 (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-5

Objetivos del Establecimiento de Comunicación con el Cliente (Mobile-D)

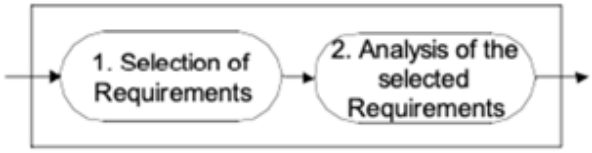
1. Definir el contexto del sistema	Los contextos del Backend y el API
2. Identificar los riesgos de arquitectura	Posibles riesgos que podrían afectar el uso de la aplicación móvil .
3. Describir la integración y crecimiento de la arquitectura de software con el proceso de Mobile-D	En caso exista alguna integración en el sistema deberá definirse en este punto.
4. Describir la integración de la arquitectura con documentación y proceso de Mobile-D	En caso el punto anterior se realice, deberá documentarse junto a la arquitectura.
5. Definir notaciones de modelado y herramientas para mockups	Se usará Balsamiq el modelado y mockups

Fuente. Elaboración propia

2.2.2. Análisis de los requerimientos iniciales

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-8: Procedimiento del Análisis de los requerimientos iniciales (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-6: Objetivos del Análisis de los requerimientos iniciales (Mobile-D)

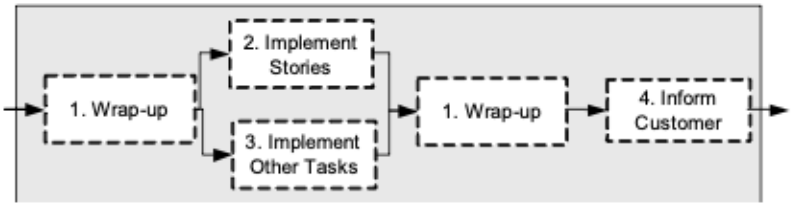
1. Selección de requerimientos	Deberá detallarse que requerimientos se realizarán en esta segunda fase.
2. Análisis de los requerimientos seleccionados	Detallar los requerimientos seleccionados con ayuda visual

Fuente. Elaboración propia

2.3. Día de Trabajo en Iteración 0

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-9: Procedimiento del Día de Trabajo en Iteración 0 (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-7: Objetivos del Día de Trabajo en Iteración 0 (Mobile-D)

1. Wrap-up	Se comunicará los progresos desde el planeamiento
2. Implementar historias de usuario	Deberá implementarse las historias definidas, tanto en el backend como en el móvil

Fuente. Elaboración propia

2.4. Día de Lanzamiento

Se mostrará lo trabajado en las anteriores etapas.

Herramientas: PHPStorm, Visual Studio Code

Técnica: Implementación de Software

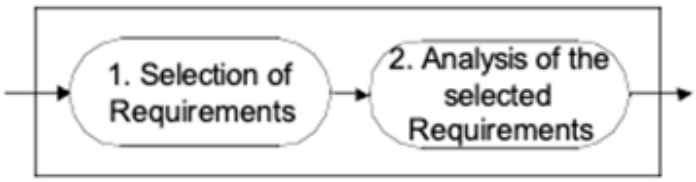
3. Fase de Producción

3.1. Día de Planeamiento

3.1.1. Análisis de Requerimientos

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-10: Procedimiento del Análisis de Requerimientos (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-8: Objetivos del Análisis de Requerimientos (Mobile-D)

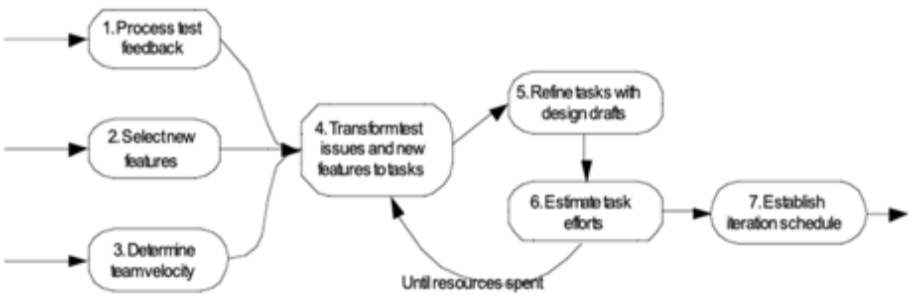
1. Selección de requerimientos	Definir los requerimientos que serán desarrollados para la penútlima fase.
2. Análisis de requerimientos	Detallar los requerimientos seleccionados con ayuda visual

Fuente. Elaboración propia

3.1.2. Planeamiento de Iteración

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-11: Procedimiento del Planeamiento Iteración (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-9: Objetivos del Planeamiento Iteración (Mobile-D)

1. Feedback del proceso de testeo	Recoger los resultados del testeo previo e identificar los defectos
2. Selección de nuevas características	El dueño definirá los nuevos historias a implementar
3. Transformar los problemas del testing en nuevas tareas	Identificar las tareas a implementar de acuerdo a las previamentes definidas por el dueño. Se dará prioridad a los defectos y mejoras.
4. Refinar tareas con diseños	Crear los mockups de la aplicación sobre las definididas en el paso 4
5. Estimar las nuevas tareas	Estimar las todas las tareas

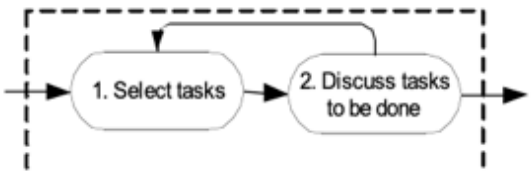
Fuente. Elaboración propia

3.2. Día de Trabajo

3.2.1. Wrap-up

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-12 Procedimiento de Wrap-up (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-10: Objetivos de Wrap-up (Mobile-D)

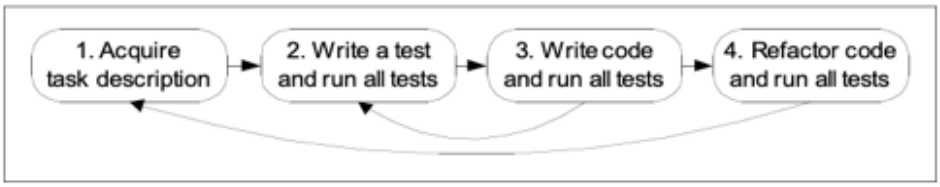
1. Seleccionar tareas	Seleccionar las tareas a ser implementadas
2. Discutir tareas a realizar	Ver a detalle la tarea y el diseño de la misma

Fuente. Elaboración propia

3.2.2. TDD (Desarrollo Guiado por Pruebas)

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-13: Procedimiento de TDD (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-11: Objetivos de TDD (Mobile-D)

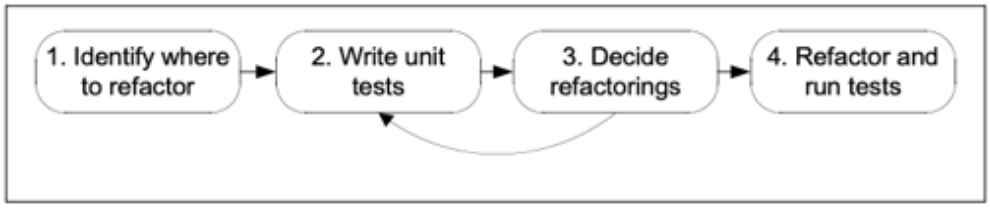
1. Adquirir descripción de la tarea	Escribir el nombre del requerimiento que se usará como definición
2. Escribir un test y correr todos los tests	Se escribe un test con la funcionalidad que se espera probar
3. Escribir el código y el correr los tests	Se codifica la funcionalidad para hacer pasar el test
4. Refactorizar el codigo y correr los tests	Después de tener el test exitoso se procede a refactorizar

Fuente. Elaboración propia

3.2.3. Refactorización

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-14: Procedimiento de Refactorización (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

Tabla 3-12: Objetivos de Refactorización (Mobile-D)

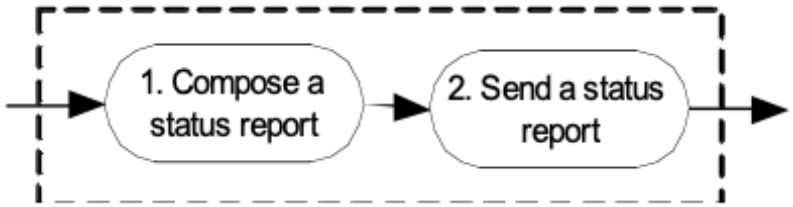
1. Identificar donde refactorizar	Identificar bloques de código repetidos, y bloques de código que sean resaltado por el IDE
2. Escribir test unitarios	Escribir un test antes de la refactorización
3. Decidir que refactorizar	Decidir que factorizar
4. Refactorizar y correr los tests	Correr los test para verificar que no se alteró el comportamient

Fuente. Elaboración propia

3.2.4. Informar al Cliente

Se procede a desarrollar los objetivos otorgados por Mobile-D

Figura 3-15: Procedimiento para Informar el Cliente (Mobile-D)



Fuente. Tomado de los documentos de Mobile-D (VTT, 2004)

3.3. Día de Lanzamiento

Se mostrará lo trabajado en las anteriores etapas.

4. Fase de Estabilización

4.1. Día de Planeamiento

4.1.1. Análisis de Requerimientos

Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en la Fase de Producción > Día de Planeamiento > Análisis de requerimientos.

4.1.2. Planeamiento Iteración

Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en la Fase de Producción > Día de Planeamiento > Planeamiento Iteración.

4.2. Día de Trabajo

4.2.1. Wrap-up

Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en Fase de Producción > Día de Trabajo > Wrap-up

4.2.2. TDD (Desarrollo Guíado de Pruebas)

Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en Fase de Producción > Día de Trabajo > TDD

4.2.3. Refactorización

Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en Fase de Producción > Día de Trabajo > Refactorización

4.2.4. Informar al Cliente

Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en Fase de Producción > Día de Trabajo > Informar al Cliente

4.3. Día de Lanzamiento

Se mostrará lo trabajado en las anteriores etapas.

5. Fase de Pruebas y Arreglos

5.1. Pruebas del Sistema

Se procederá a realizar las pruebas y a solucionar los errores encontrados, y las correciones que serás observadas por el dueño, así como por el equipo.

5.2. Día de Planeamiento

5.2.1. Análisis de Requerimientos

Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en la Fase de Producción > Día de Planeamiento > Análisis de requerimientos.

5.2.2. Planeamiento Iteración

Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en la Fase de Producción > Día de Planeamiento > Planeamiento Iteración.

5.3. Día de Trabajo

5.3.1. Wrap-up

Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en Fase de Producción > Día de Trabajo > Wrap-up

5.3.2. TDD (Desarrollo Guíado de Pruebas)

Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en Fase de Producción > Día de Trabajo > TDD

5.3.3. Refactorización
Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en Fase de Producción > Día de Trabajo > Refactorización

5.3.4. Informar al Cliente
Se vuelven a desarrollar los objetivos otorgados por Mobile-D, se repite el procedimiento mencionado en Fase de Producción > Día de Trabajo > Informar al Cliente

5.3.5. Día de Lanzamiento
Se mostrará lo trabajado en las anteriores etapas.

CAPÍTULO 4: DESARROLLO DE LA SOLUCIÓN TECNOLÓGICA

En este capítulo se procederá al desarrollo de la aplicación móvil bajo la metodología de Mobile-D

4.1 Descripción de las actividades realizadas

1. Fase Exploración
   1.1. Definición de Stakeholders
      1.1.1. Establecimiento de Clientes
         1.1.1.1. Identificar los clientes que participarán

            El cliente sería el dueño de una mascota que haya pasado la experiencia de perder su mascota.

         1.1.1.2. Comprometer al cliente

            El dueño de la mascota se compromete a ser parte del ciclo de desarrollo de software con la metodología Mobile-D.

         1.1.1.3. Definir el modo de roles y responsabilidades

            El dueño será el actor que ayudará a brindar los requerimientos bajo su experiencia.

   1.2. Definición de Alcance
      1.2.1. Colección Inicial de Requerimientos
         1.2.1.1. Definir el Cronograma

Figura 4-1: Cronograma de actividades

Fuente. Elaboración Propia. Cronograma detallando las fases de Mobile-D.

         1.2.1.2. Definir el ritmo

            Se realizarán 4 iteraciones de 2 semanas. Se iniciará la primera semana de Mayo, terminando la cuarta semana de Junio.

         1.2.1.3. Definir/estimar los presupuestos

Figura 4-2: Estimación de presupuesto de costos del desarrollo del proyecto

Presupuesto

Fecha presupuesto:	13-05-2022		
DESCRIPCIÓN	PRECIO	Cantidad	TOTAL
Tarifa por Hora de Desarrollador de Software	$ 50.00	160	$ 8,000.00
Macbook Pro 2019	$ 3,000.00	1	$ 3,000.00
Licencia PHPSTORM	$ 199.00	1	$ 199.00
Cuenta Apple Developer	$ 99.00	1	$ 99.00
Costo de luz	$ 50.00	1	$ 50.00
		Total en dólares	$11,348

Fuente. Elaboración Propia. Presupuesto general de los costos que llevará el proyecto.

1.2.2. Planeamiento Inicial del Proyecto

1.2.2.1. Recoger los requerimientos

1. Crear el backlog de requerimientos funcionales
    - Registrar Usuario
    - Login Usuario
    - Registrar mascota
    - Listar mis mascotas
    - Publicar anuncio de mascota perdida
    - Listar anuncios de mascotas perdidas
    - Reportar avistamiento de Mascota Perdida
    - Notificar al dueño de mascota por email y SMS

2. Crear los mockups del aplicativos

- Registrar Usuario

Figura 4-3: Mockup de la aplicación móvil para Registrar Usuario

Fuente. Elaboración Propia. Visualización de la acción para que un usuario se registre

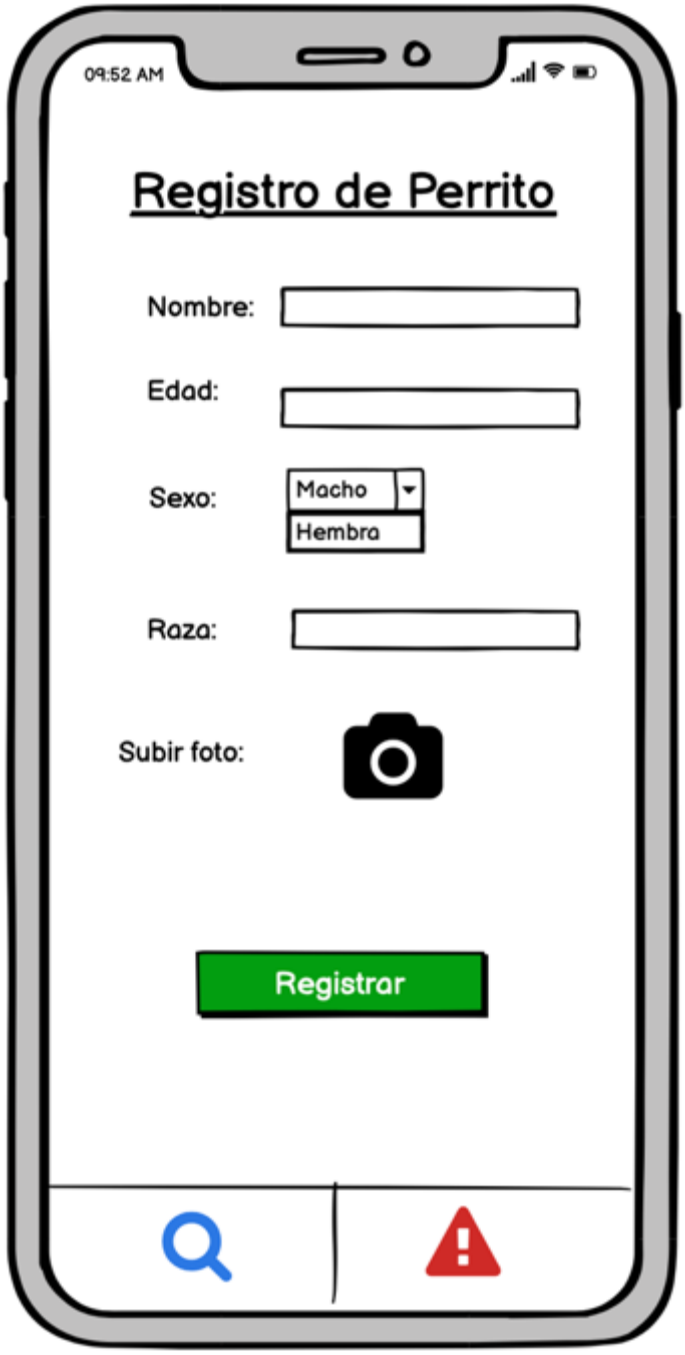
- Login Usuario

Figura 4-4: Mockup de la aplicación móvil para ingresar a la aplicación (Login)

Fuente. Elaboración Propia. Visualización de la acción para que un usuario se ingrese a la aplicación después de ingresar sus datos.

- Registrar Mascota

Figura 4-5: Mockup de la aplicación móvil para el Registro de una Mascota



Fuente. Elaboración Propia. Visualización de la para registrar una mascota.

- Listar Mascotas

Figura 4-6: Mockup de la aplicación móvil para listar las mascotas

Fuente. Elaboración Propia. Visualización de la lista de mascotas.

- Publicar anuncio de Mascota Perdida

Figura 4-7: Mockup de la aplicación móvil para publicar anuncio

Fuente. Elaboración Propia. Visualización del formulario para reportar una mascota como perdida.

- Reportar avistamiento de Mascota Perdida

Figura 4-8: Mockup de la aplicación móvil para mostrar las mascotas perdidas

Fuente. Elaboración Propia. Visualización de la pantalla que muestra las mascotas perdidas

Figura 4-9: Mockup de la aplicación para marcar el avistamiento de una mascota

Fuente. Elaboración Propia. Visualización del formulario para reportar una mascota como perdida.

- Alertar Dueño de mascota por email y SMS

Figura 4-10: Mockup de la aplicación para la alerta de SMS

Fuente. Elaboración Propia. Visualización de la alerta SMS cuando hubo un avistamiento de una mascota

Figura 4-11: Mockup de la aplicación para la alerta Email

Fuente. Elaboración Propia. Visualización de la alerta email cuando hubo un avistamiento de una mascota

3. Desarrollar el backend con los APIs para consumir por la aplicación

   El backend será el encargado de desarrollar y proveer los requerimientos del backlog para servirlos como API que será consumido por el aplicativo móvil.

4. Desarrollar la aplicación móvil con los requerimientos definidos previamente

   El aplicativo móvil desarrollará las interfaces de los mockups y consumirá los servicios brindado por el backend.

5. Realizar las prueba y arreglos necesarios

   Se deberán realizar el desarrollo de las APIs con Test Driven Development para crear requerimientos funcionales, probados lógicamente y que podrá ser corrido en cualquier momento.

1.2.2.2. Discutir requerimientos

No se encontró ninguna objeción por parte del cliente respecto a los requerimientos antes mencionados, dando todo como conforme y estando de acuerdo.

1.2.2.3. Acordar los requisitos iniciales

Todos los requisitos del backlog

1.2.2.4. Documentar los requerimientos iniciales

Se deberá un formato adecuado al backlog para que desde previo al inicio al desarrollo de cada requerimiento se tenga claro que se desea tener como objetivo.

1.2.3. Establecimiento del Proyecto

1.2.3.1. Selección del Ambiente

Para el desarrollo del aplicativo móvil se requerirá un hardware con estos requerimientos mínimos:

- Sistema Operativo: MAC OS X con la versión Monterrey o superior
- Memoria Ram: 16 GB 2400 MHz DDR4 o superior
- Procesador: 2.3GHz 8-Core Intel Core i7 o superior
- Gráfica: 512 MB o Superior

Figura 4-12: Características de MacBook Pro 2019

Fuente. Elaboración Propia. Este equipo servirá para el desarrollo del aplicativo móvil y APIS. Elaboración Propia

Se deberá además de contar con la licencia de las siguientes herramientas:

- IDE PhpStorm 2020.2 o superior

Figura 4-13: Licencia del IDE PhpStorm 2020.2

Fuente. Elaboración Propia. PhpStorm es el IDE que se usará para la creación de APIS. Elaboración Propia

- Cuenta de desarrollador de iOS

Figura 4-14: Cuenta de desarollador iOS

Fuente. Elaboración Propia. La cuenta de iOS servirá para exportar la aplicación a .ipa, un formato que maneja Apple. Elaboración Propia

Además de las siguientes herramientas que son de acceso gratuito:

- Visual Studio Code

Figura 4-15: Editor Visual Studio Code 1.67.1

Fuente. Elaboración Propia. El editor Visual Studio Code se usará para el desarrollo del aplicativo móvil

- Android Studio
- Android Virtual Device Manager
- Postman

2. Fase de Inicialización

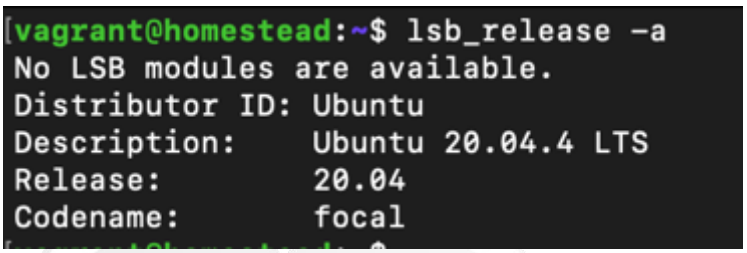
2.1. Configuración del Proyecto

2.1.1. Configuración del Ambiente

1. Para la configuración del ambiente del backend para las APIS se usará:

- Ubuntu Server 20.04 LTS. Se usa LTS por ser una versión estable y con soporte de hasta cinco años

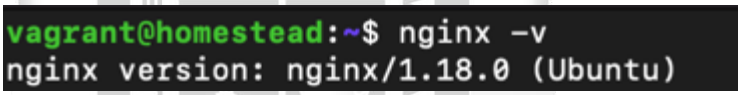
Figura 4-16: Versión del sistema operativo Ubuntu



Fuente. Elaboración Propia. Sistema operativo en el que correrá el backend para dar el servicio de las APIs

- Nginx versión 1.18.0

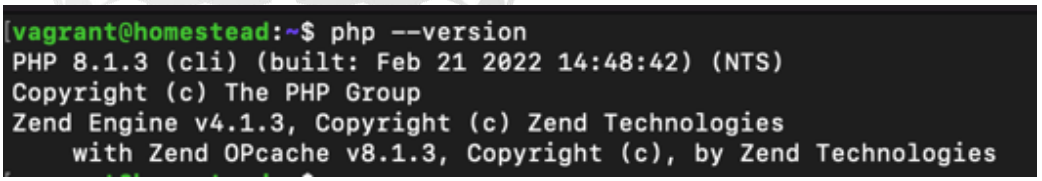
Figura 4-17: Versión del servidor NGINX



Fuente. Elaboración Propia. Servidor web del backend

- PHP versión 8.1.3

Figura 4-18: Versión de PHP



Fuente. Elaboración Propia. Lenguaje de programación para desarrollo de web o APIs.

- MySQL versión 8.0.28

Figura 4-19: Versión de MySQL

Fuente. Elaboración Propia. Base de datos relacional.

- Laravel Framework versión 8

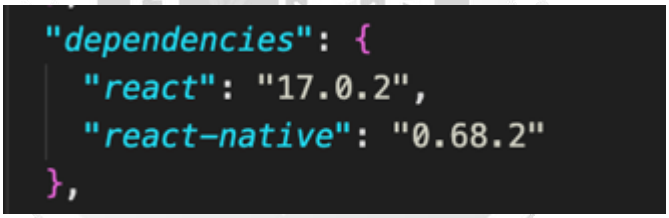
Figura 4-20: Versión de Laravel

Fuente. Elaboración Propia. Framework PHP.

2. Para la configuración del ambiente para móviles se usará:

- React Native versión 0.68.2

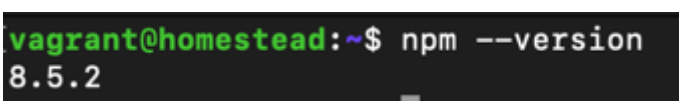
Figura 4-21: Versión de React Native



Fuente. Elaboración Propia. Framework Javascript para la creación de aplicaciones móviles.

- NPM versión 8.5.2

Figura 4-22: Versión de NPM



Fuente. Elaboración Propia. Manejador de paquetes.

- Javascript con Typescript

2.1.2. Entrenamiento

Se procede a listar la documentación oficial de los lenguajes programación que serán consultados constantemente

Tabla 4-1: Documentación de lenguajes de programación

PHP	https://www.php.net/manual/en/index.php
Laravel Framework	https://laravel.com/docs/8.x/
React Native	https://reactnative.dev/docs/0.61/enviroment-setup
Mobile-D	http://virtual.vtt.fi/virtual/agile/mobiled.html
Android	https://developer.android.com/docs
Swift	https://www.swift.org/documentation/

Fuente. Elaboración Propia. Lenguajes con links de su respectiva documentación.

2.1.3. Establecimiento de Comunicación con el Cliente

a. Identificar las necesidades para la comunicación

La comunicación será con un entregable con los requerimientos acordados después de cada fase de Mobile-D.

b. Identificar las maneras por la comunicación

La comunicación será por correo electrónico.

c. Identificar los contenidos de la comunicación

Se darán los progresos de cada iteración en un video o screenshots de las pantallas que contenga los detalles de los avances y dificultades encontrados.

2.2. Día de Planeamiento en Iteración 0

2.2.1. Planificación de la Línea de Arquitectura

a. Definitir el contexto del sistema

Tabla 4-2: Contexto del Sistema

Móvil	Android y iOS
API	Ubuntu Server

Fuente. Elaboración propia.

b. Identificar los riesgos de arquitectura

1. Arquitectura REST

- Cambiar la forma de envio de datos, si se envia en JSON y luego en XML puede dejar inactiva la aplicación.

- No manejar consistencia en los datos y rutas, puede afectar a los usuarios que no actualicen la aplicación.

2. Arquitectura Cliente-Servidor
    - No contactar con un fiable sistema de autenticación puede comprometer el sistema
    - No restringir la cantidad de peticiones, puede llevar a una sobrecarga al servidor y elevación de costos

c. Describir la integración y crecimiento de la arquitectura de software con el proceso de Mobile-D

La integración y arquitectura de crecimiento sólo sucederá de parte del backend (API).

d. Describir la integración de la arquitectura con documentación y proceso de Mobile-D

La integración de la aplicación móvil con el backend (API) sucederá al comunicarse a través de una arquitectura cliente-servidor.

e. Definir notaciones de modelado y herramientas para mockups

Se usará Balsamiq el modelado y mockups

3. Fase de Producción

3.1. Día de Planeamiento

3.1.1. Análisis de Requerimiento

a. Selección de requerimientos

- Registrar Usuarios (API)
- Registrar Usuarios (Móvil)
- Login de Usuarios (API)
- Login de Usuarios (Móvil)
- Registrar Mascotas (API)
- Registrar Mascotas (Móvil)
- Listar Mascotas (API)
- Listar Mascotas (Móvil)

b. Análisis de requerimientos seleccionados

- Registrar Usuarios (API)

  Este requerimiento es uno de los esenciales, el cual permitirá al usuario registrarse y poder acceder a la aplicación. Se realizará a través del método POST.

- Registrar Usuarios (Móvil)

A través del aplicativo móvil se procederá a consumir el API para poder enviar una petición POST para registrar el usuario.

- Login de Usuarios (API)

  Después de desarrollarse este requerimiento el usuario podrá autenticarse en la aplicación, previo registro. Se realizará a través del método POST.

- Login de Usuarios (Móvil)

  A través del aplicativo móvil se procederá a consumir el API para enviar una petición POST para autenticarse y recibir el token que se usará en el aplicativo móvil.

- Registrar Mascota(API)

  Este requerimiento sólo es accesible para los usuarios registrados.

  Los dueños de mascotas podrán registrar todas las mascotas que posean, para tener toda su información cargada en el sistema en caso alguna vez requiera hacer una publicación. Se realizará a través del método POST.

- Registrar Mascota(Móvil)

  A través del aplicativo móvil, con el token recibido por el login de usuario, se podrá hacer uso de esta API, con el que podrá registrarse a una mascota.

- Listar Mascotas (API)

  En este requerimiento los dueños de mascotas podrán visualizar las mascota registradas en la aplicación.

- Listar Mascotas (Móvil)

  A través del aplicativo móvil, con el token recibido por el login de usuario, se podrá hacer uso de esta API, con el que podrá listar todas las mascotas pertenecientes al usuario autenticado.

3.2. Día de Trabajo

3.2.1. Wrap-up

Se comunica al cliente que se tiene el ambiente listo para el desarrollo, el backlog de tareas y el dealle de cada requerimient, así como los mockups realizados de como se verá la aplicación en el aplicativo móvil

3.2.2. TDD (Desarrollo Guiado de Pruebas)

3.2.2.1. Adquirir descripción de la tarea

Las pruebas guiadas se desarollarán sólo en el backend (API), ya que son las que contienen toda la lógica y el almacenamiento de la información, estas serán :

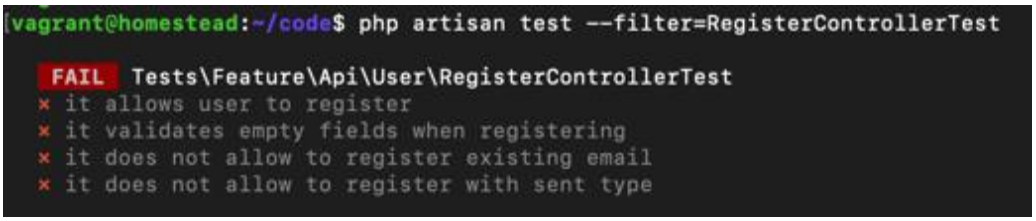
- Registrar Usuarios (API)
- Login de Usuarios (API)
- Registrar Mascota (API)

- Listar Mascotas (API)

3.2.2.2. Escribir un test y correr los tests

- Registrar Usuarios (API)

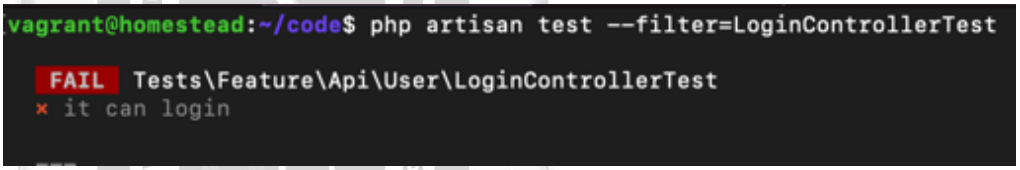
Figura 4-23: Test fallido para Registrar usuario



Fuente. Elaboración Propia. Se escribe la prueba para registrar un usuario.

- Login de Usuarios (API)

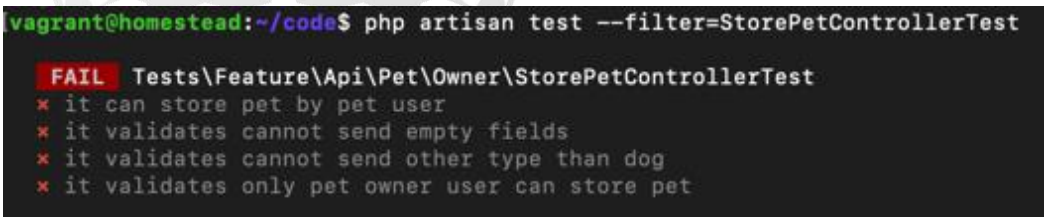
Figura 4-24: Test fallido para Login



Fuente. Elaboración Propia. Se escribe la prueba para el login.

- Registrar Mascota (API)

Figura 4-25: Test fallido para Registrar Mascota



Fuente. Elaboración Propia. Se escribe la prueba para registrar mascota

- Listar Mascotas (API)

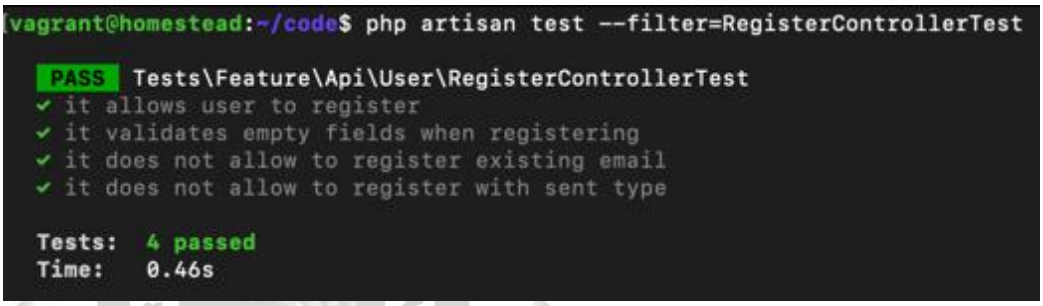
Figura 4-26: Test fallido para Listar Mascotas

Fuente. Elaboración Propia. Se escribe la prueba para listar las mascotas del dueño

3.2.2.3. Escribir el código y correr los tests

- Registrar Usuarios (API)

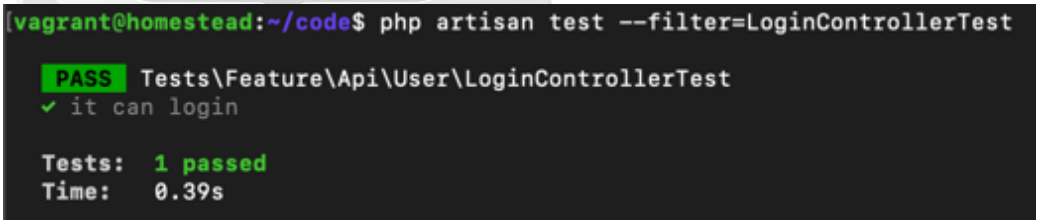
Figura 4-27: Test exitoso para Registrar Usuarios



Fuente. Elaboración Propia. Se escribe la prueba para registrar usuarios

- Login de Usuarios (API)

Figura 4-28: Test exitoso para el Login de Usuarios



Fuente. Elaboración Propia. Se escribe la prueba para el login de usuarios

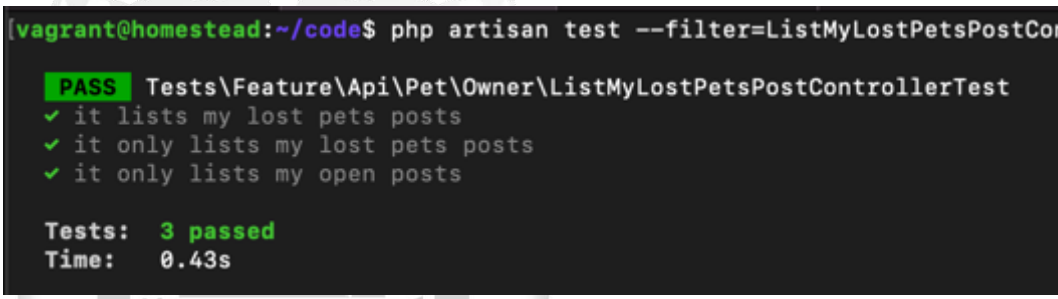
- Registrar Mascota (API)

Figura 4-29: Test exitoso para Registrar Mascota

Fuente. Elaboración Propia. Se escribe la prueba para registrar mascota

- Listar Mascotas (API)

Figura 4-30: Test exitoso para Listar Mascotas



Fuente. Elaboración Propia. Se escribe la prueba para listar mascota

3.2.2.4. Refactorizar el código y correr los tests

Se refactorizo métodos aplicando los conceptos SOLID y gracias a los test se pudo verificar que la integridad del sistema no se vio comprometida en lo que se refactorizó.

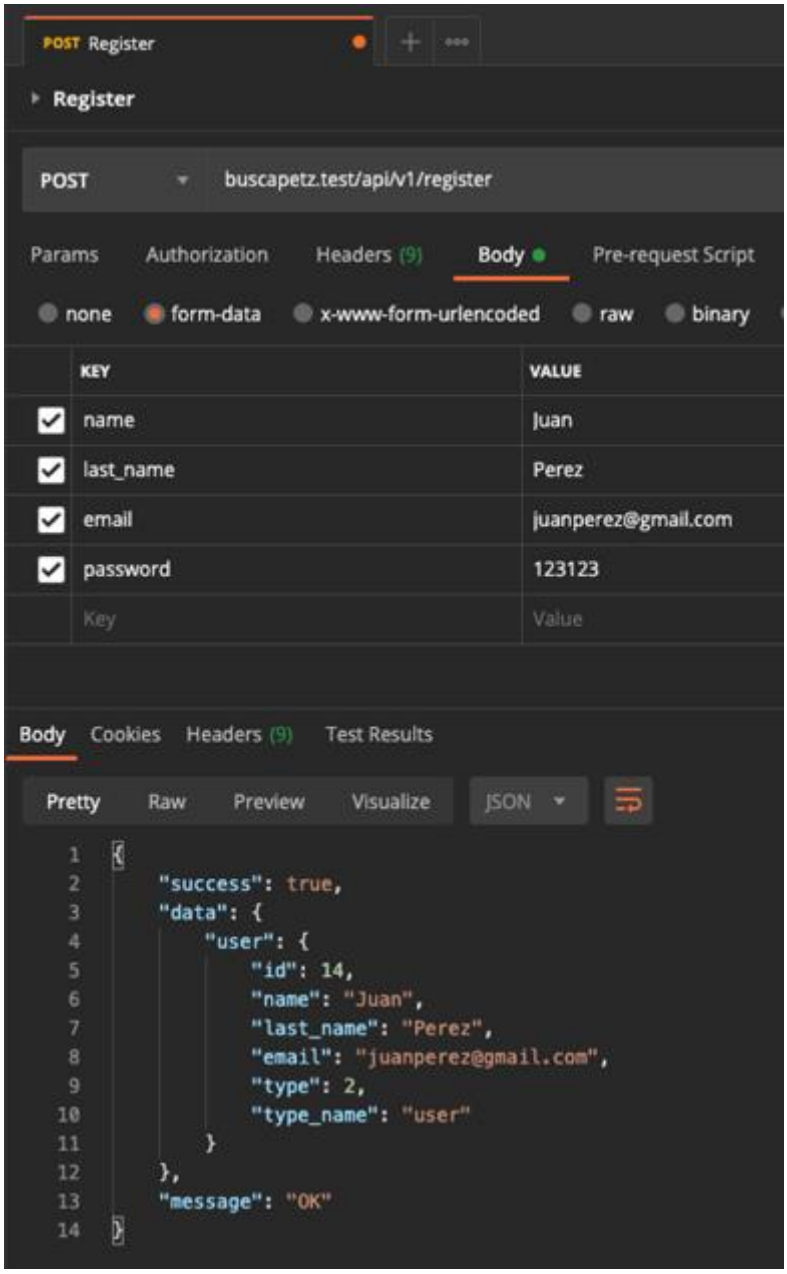
3.3. Día de Lanzamiento

Se procede a mostrar el funcionamiento de la API, así como la integración con el aplicatvo móvil:

1. Registrar Usuarios (API)

   Se muestra las API de registrar Usuario funcionando

Figura 4-31: Request al API para Registrar un usuario



Fuente. Elaboración Propia. Request al API existoso del registro de un usuario

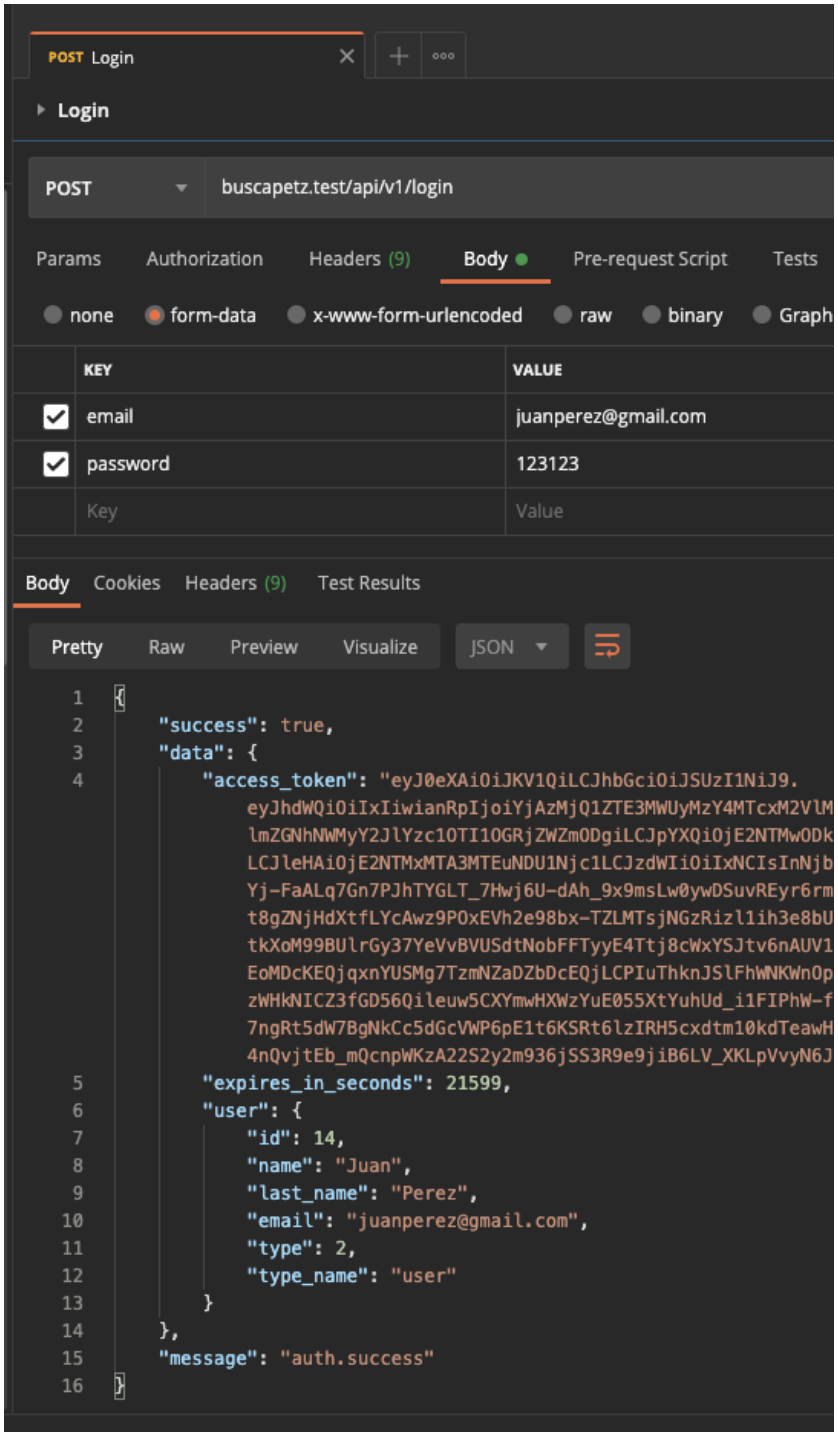
Figura 4-32: Pantalla de la aplicación móvil para el Registro

Fuente. Elaboración Propia. Pantalla de la aplicación móvil para registrarse y hacer uso del aplicativo.

2. Login de Usuarios (API)

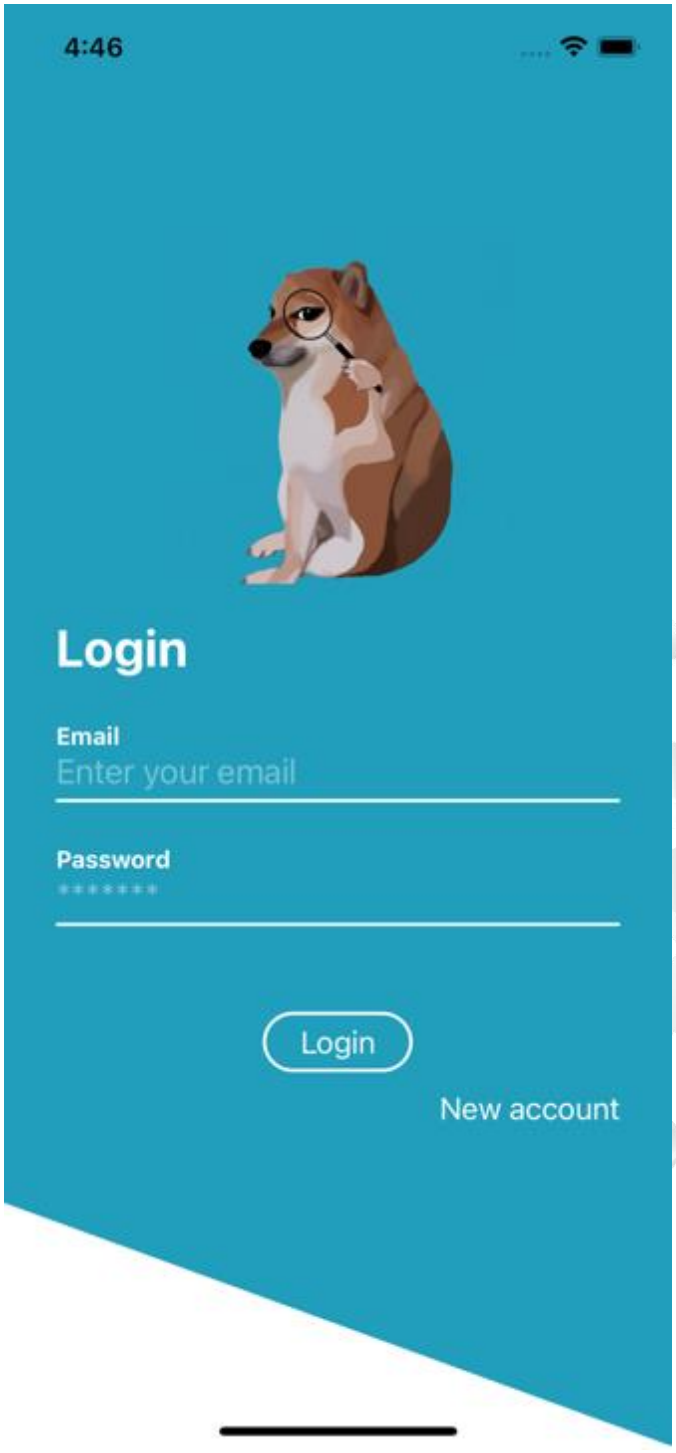
   Se muestra las API de login de Usuario funcionando:

Figura 4-33: Request al API para ingresar el sistema (Login)



Fuente. Elaboración Propia. Request al API existoso del login de un usuario

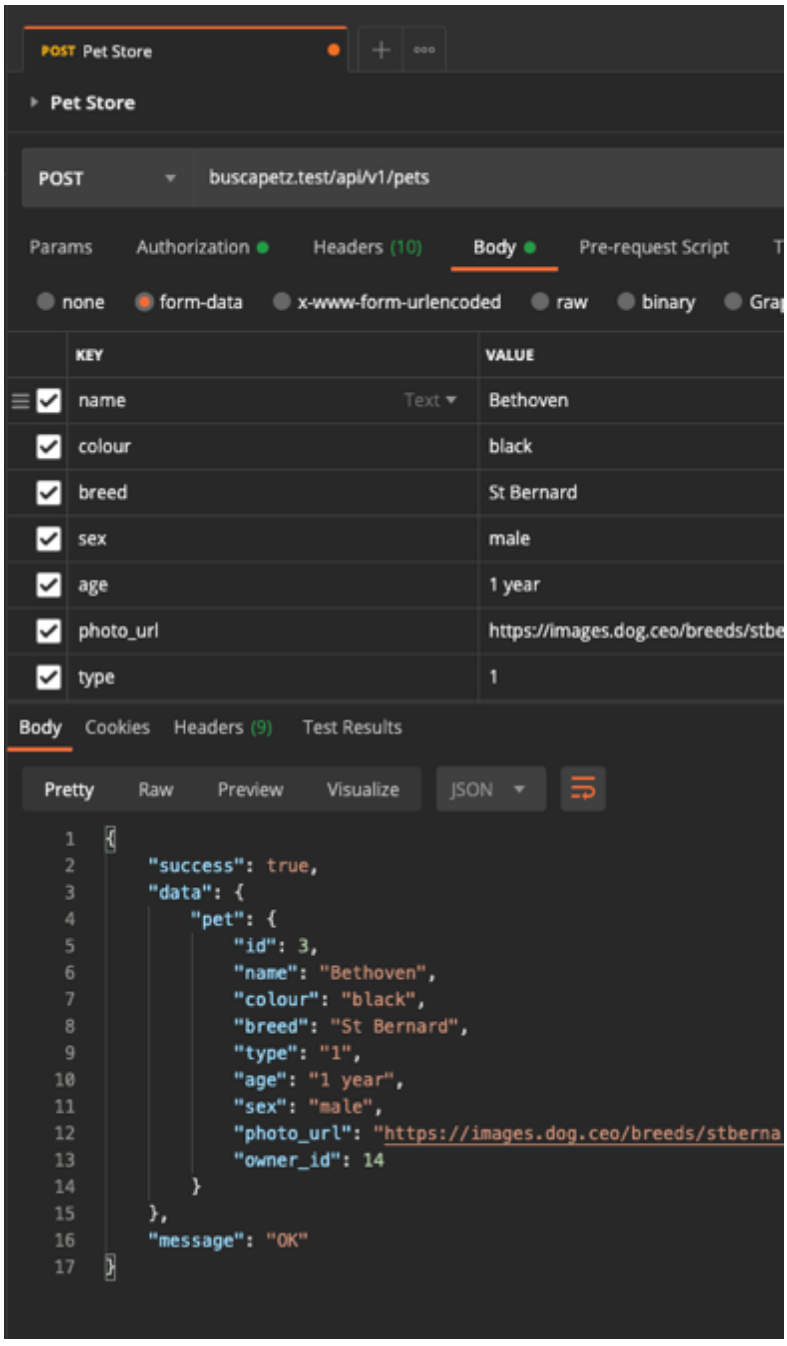
Figura 4-34: Pantalla de la aplicación móvil para el Login



Fuente. Elaboración Propia. Pantalla de la aplicación móvil para ingresar y hacer uso del aplicativo.

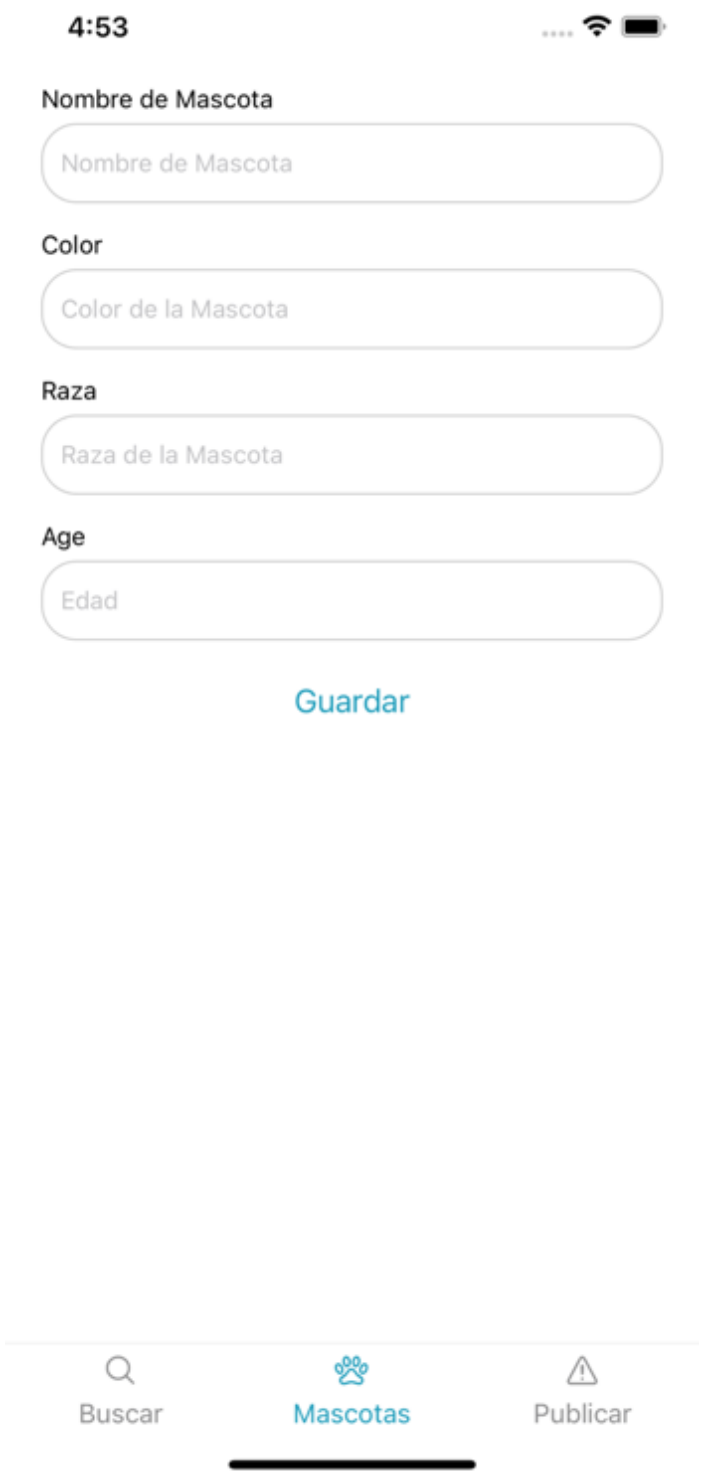
3. Registrar Mascota (API)
   Se muestra las API de login de Usuario funcionando:

Figura 4-35: Request al API para Registrar un Mascota



Fuente. Elaboración Propia. Request al API existoso al Registrar una Mascota

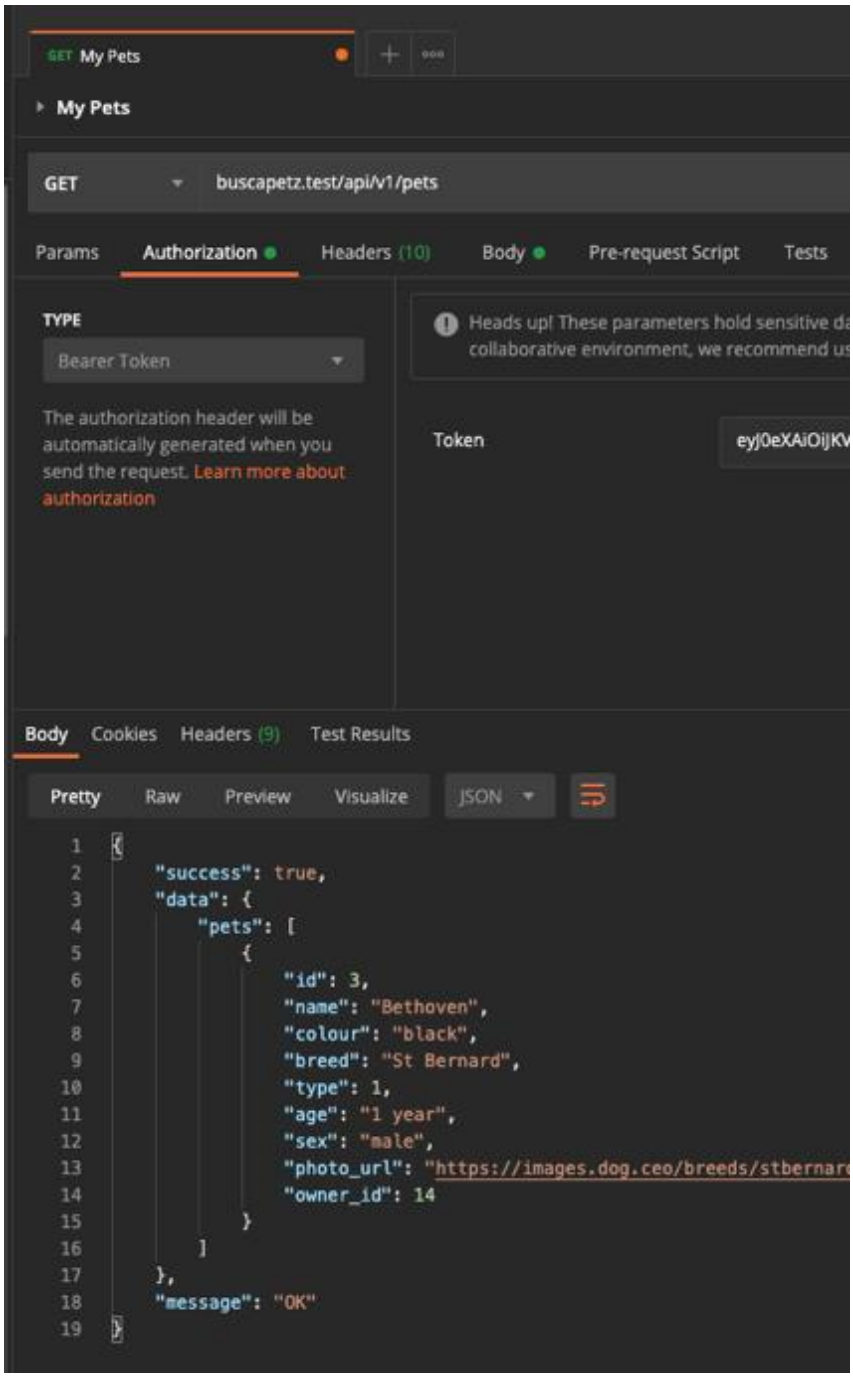
Figura 4-36: Pantalla de la aplicación móvil para registrar una Mascota



Fuente. Elaboración Propia. Pantalla de la aplicación móvil para registrar una mascota.

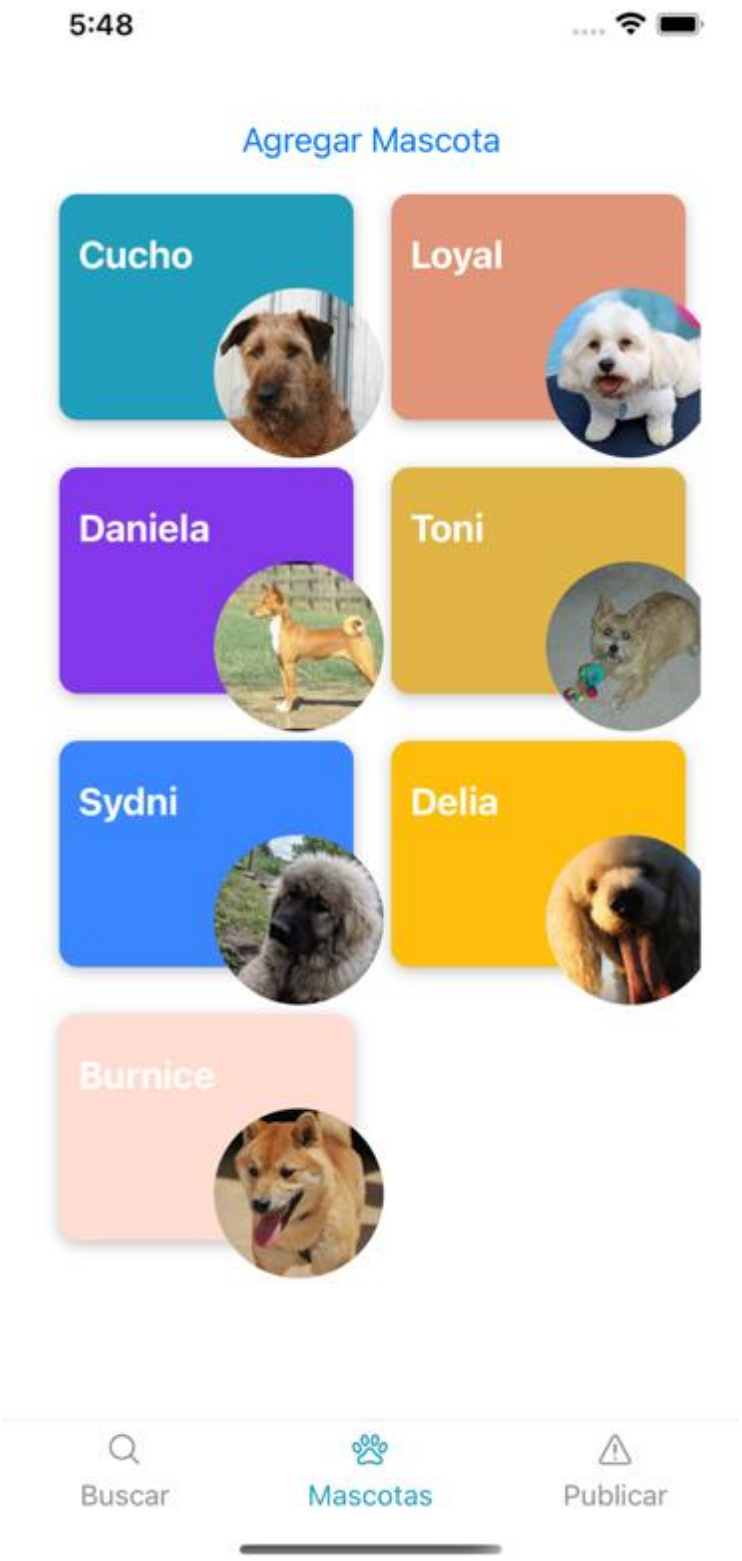
4. Listar Mascotas (API)
   Se muestra las API de login de Usuario funcionando:

Figura 4-37: Request al API para Listar Mascotas



Fuente. Elaboración Propia. Request al API existoso para Listar Mascotas

Figura 4-38: Pantalla de la aplicación móvil para listar Mascotas



Fuente. Elaboración Propia. Pantalla de la aplicación móvil para listar sus mascotas

4. Fase de Estabilización

4.1. Día de Planeamiento

4.1.1. Análisis de Requerimeintos

a. Selección de Requerimientos

1. Publicar anuncio de mascota perdida(API)
2. Publicar anuncio de mascota perdida(Móvil)
3. Listar mascotas perdidas (API)
4. Listar mascotas perdidas (Móvil)
5. Reportar avistamiento de mascota (API)
6. Reportar avistamiento de mascota (Móvil)
7. Marcar una mascota como encontrada (API)
8. Marcar una mascota como encontrada (Móvil)

b. Ánalisis de Requerimientos seleccionados

1. Publicar anuncio de mascota perdida (API)

   El dueño de una mascota debe ser capaz de publicar un anuncio de su mascota perdida, dando como detalles la fecha y hora, el lugar donde se perdió o visto por última vez, y la mascota previamente registrada . Se realizará a través del método POST.

2. Publicar anuncio de mascota perdida (Móvil)

   El dueño, previamente autenticado y con una mascota ya registrada deberá ser capaz de publicar que su mascota está perdida, registrando su actual posición como punto inicial a través del método POST.

3. Listar mascotas perdidas (API)

   Cualquier usuario autenticado, deberá de ser capaz de obtener la lista de las mascotas perdidas. Se realizará a través del método POST.

4. Listar mascotas perdidas (Móvil)

   Un usuario, previamente autenticado en la aplicación deberá ser capaz de visualizar las mascotas perdidas con las fotos para identificarlas rápidamente, a través del método POST deberá ser capaz de consumir este servicio.

5. Reportar avistamiento de mascota (API)

   Un usuario autenticado debe ser capaz de reportar un avistamiento en la aplicación, un avistamiento significa haber visto a la mascota que se encuentra perdida. Se realizará a través del método POST.

6. Reportar avistamiento de mascota (Móvil)

   Un usuario autenticado podrá marcar a través de su ubicación actual el avistamiento de una mascota, el cual quedará registrado en el mapa para

obtener todas las posiciones donde la mascota se ha visto, se consumirá la API a través del método POST.

7. Marcar una mascota como encontrada (API)

   El dueño de una mascota, previamente autenticado, deberá ser capaz de cerrar la publicación de su mascota perdida en cualquier momento. Se realizará a través del método GET.

8. Marcar una mascota como encontrada (Móvil)

   A través del aplicativo móvil, el dueño de una mascota autenticado, deberá poder visualizar un botón que indique que podrá cerrar la publicación, dando su mascota como encontrada, esto a través del método GET.

4.1.2. Planeamiento de Iteración

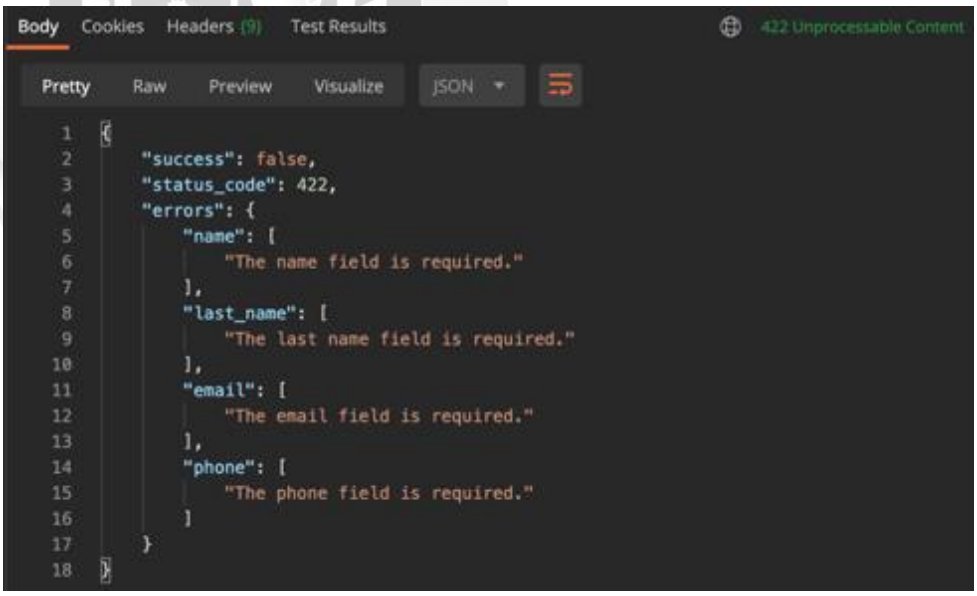
1. Feedback del proceso de testeo

   Se procedió a realizar las pruebas en los requerimientos de la fase anterior, teniendo los siguientes resultados:

- Registrar Usuarios (API)

  1. Realizar petición con ningún campo y obtener una respuesta de error requeriendo los campos (HTTP 422). **Fue exitoso**

Figura 4-39: Feedback del proceso de testeo de API para Registro de usuario sin ningún campo



Fuente. Elaboración Propia. Evidencia del registro fallido de un usuario cuando no ingresa ninguno de los campos

2. Realizar petición con un email ya existente y obtene una respuesta de error con el mensaje de que el email ya existe (HTTP 422). **Fue exitoso**

Figura 4-40: Evidencia de API para Registro de usuario con correo existente

Fuente. Elaboración Propia. Evidencia del registro fallido de un usuario cuando se ingresa un email que ya ha sido registrado por otro usuario

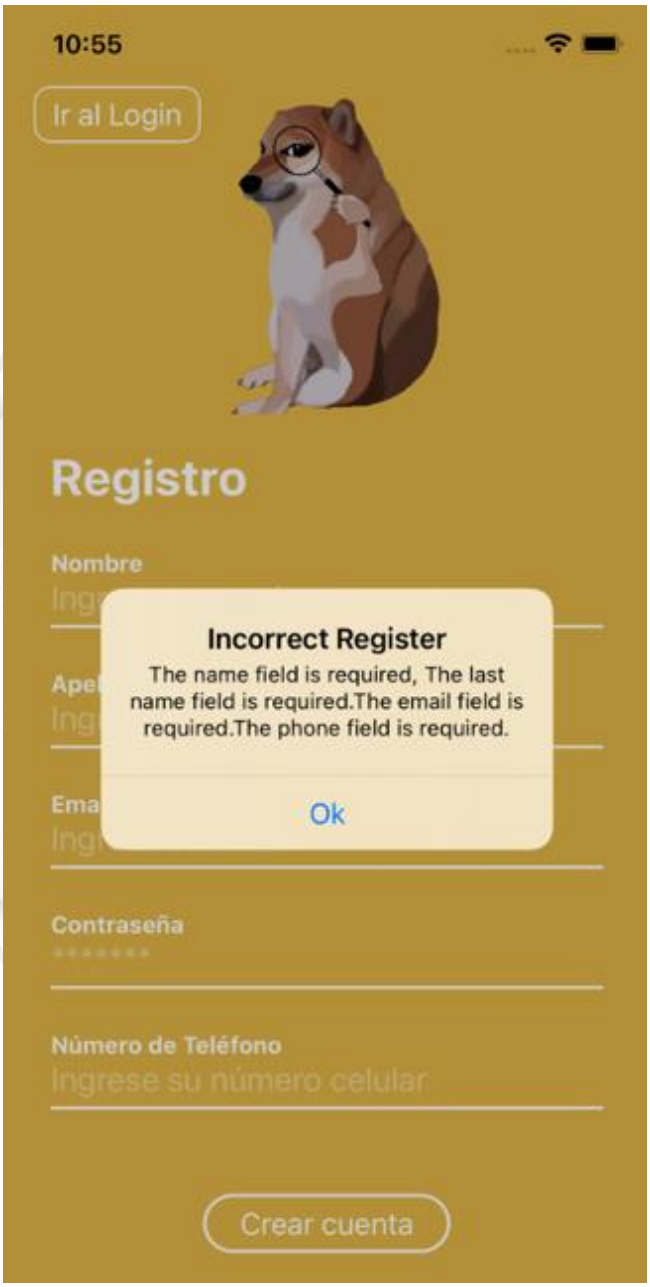
3. Realizar petición con un teléfono con menos o más de 9 digitos y obtener una respuesta de error (HTTP 422). **Fue exitoso**

Figura 4-41: Evidencia de API para Registro de usuario con teléfono erróneo

Fuente. Elaboración Propia. Evidencia del registro fallido de un usuario cuando se ingresa un teléfono de menos o más de 9 digitos.

- Registrar Usuarios (Móvil)
    1. Enviar formulario sin ningún campo y obtener una respuesta de error requeriendo los campos **Fue exitoso**

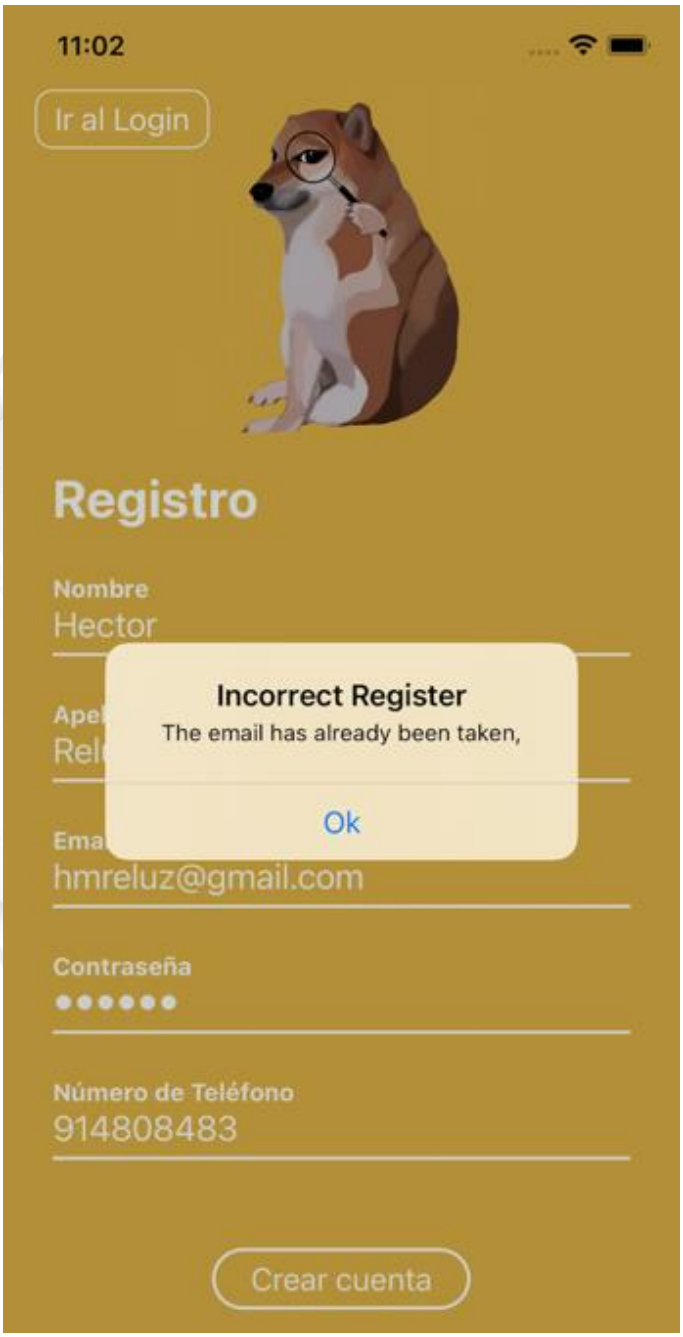
Figura 4-42: Evidencia de Móvil para Registro de usuario sin ningún campo



Fuente. Elaboración Propia. Evidencia del registro fallido de un usuario cuando no se envia ninguno de los campos

2. Enviar formulario con un email ya existente y obtene una respuesta de error con el mensaje de que el email ya existe. **Fue exitoso**

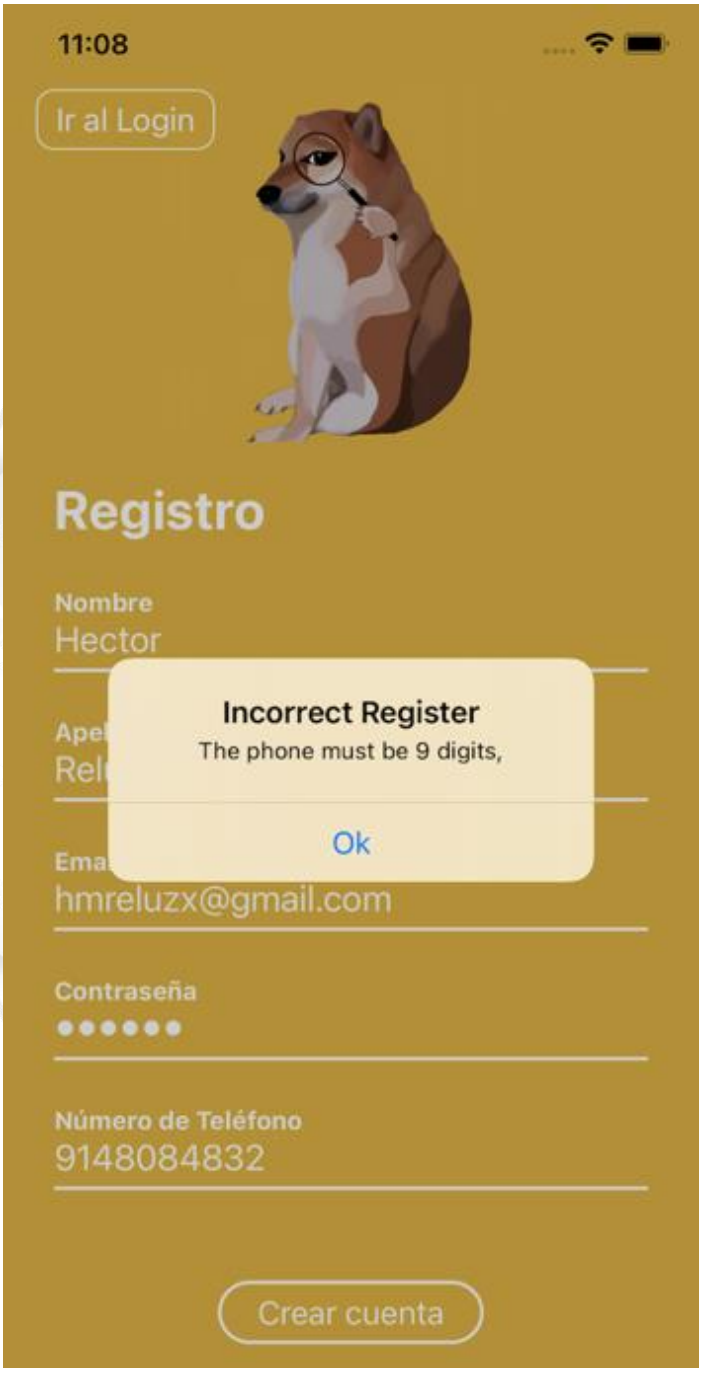
Figura 4-43: Evidencia de Móvil para Registro de usuario con email existente



Fuente. Elaboración Propia. Evidencia del registro fallido de un usuario cuando se envia un email existente

3. Enviar formulario con un teléfono con menos o más de 9 digitos y obtener una respuesta de error. **Fue exitoso**

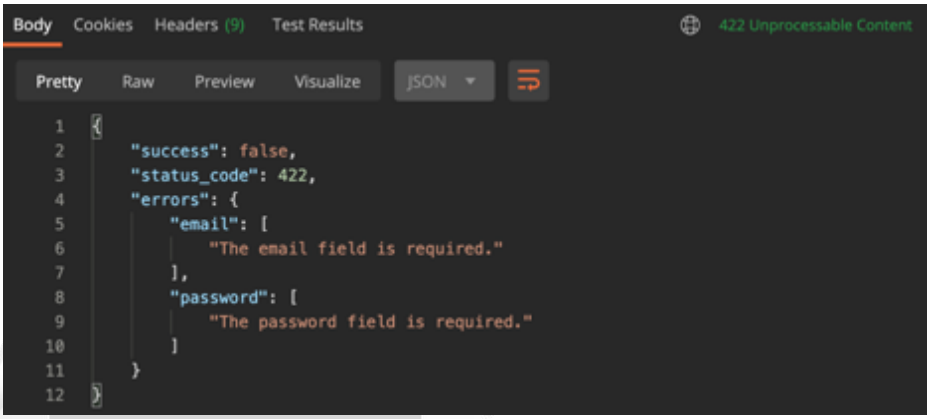
Figura 4-44: Evidencia de Móvil para Registro de usuario con teléfono erróneo



Fuente. Elaboración Propia. Evidencia del registro fallido de un usuario cuando se envia un teléfono con más o menos de 9 digitos

- Login de Usuarios (API)
    1. Realizar petición con ningún campo y obtener una respuesta de error requeriendo los campos (HTTP 422). **Fue exitoso**

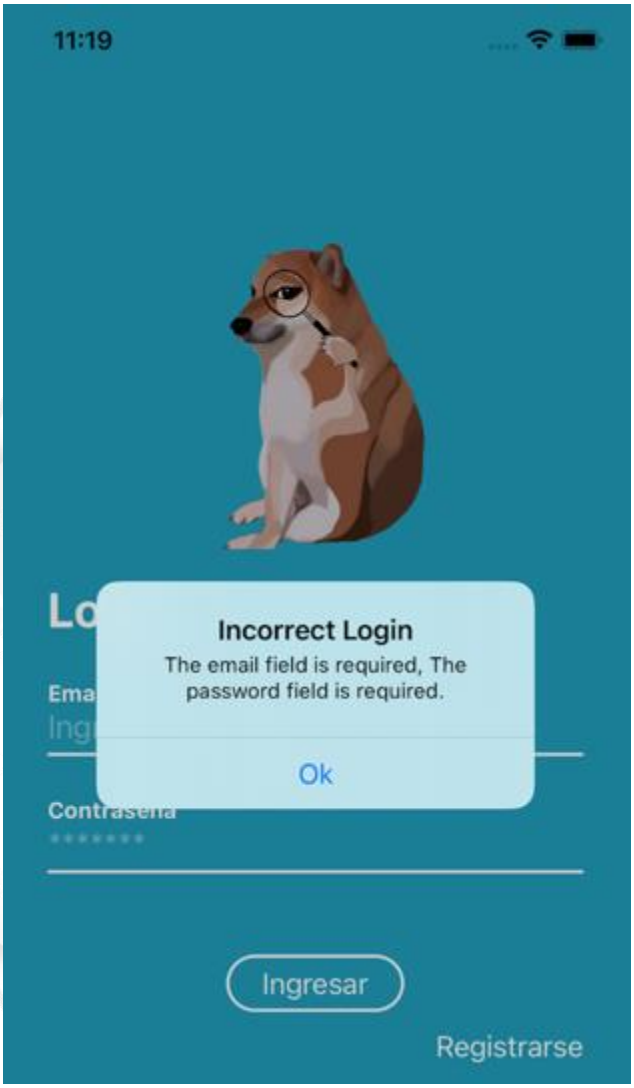
Figura 4-45: Evidencia de API para el Login sin ningún campo



Fuente. Elaboración Propia. Evidencia del registro fallido de un usuario cuando no envia el email y contraseña

- Login de Usuarios (Móvil)
    1. Enviar formulario sin ningún campo y obtener una respuesta de error requeriendo los campos. **Fue exitoso**

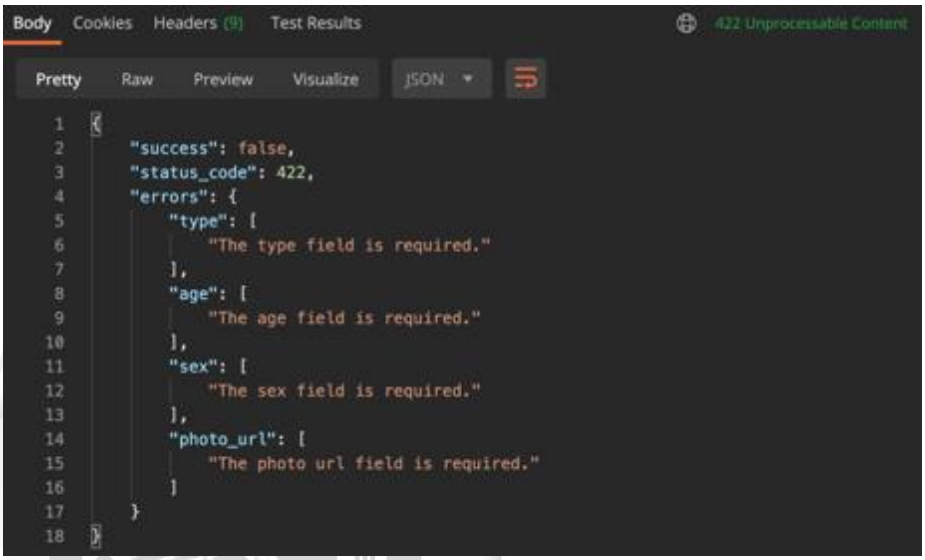
Figura 4-46: Evidencia de Móvil para el Login sin ningún campo



Fuente. Elaboración Propia. Evidencia del registro fallido de un usuario cuand no se envia el email y password.

- Registrar Mascotas (API)
    1. Realizar petición con ningún campo para registrar mascota y obtener una respuesta de error (HTTP 422). **Fue exitoso**

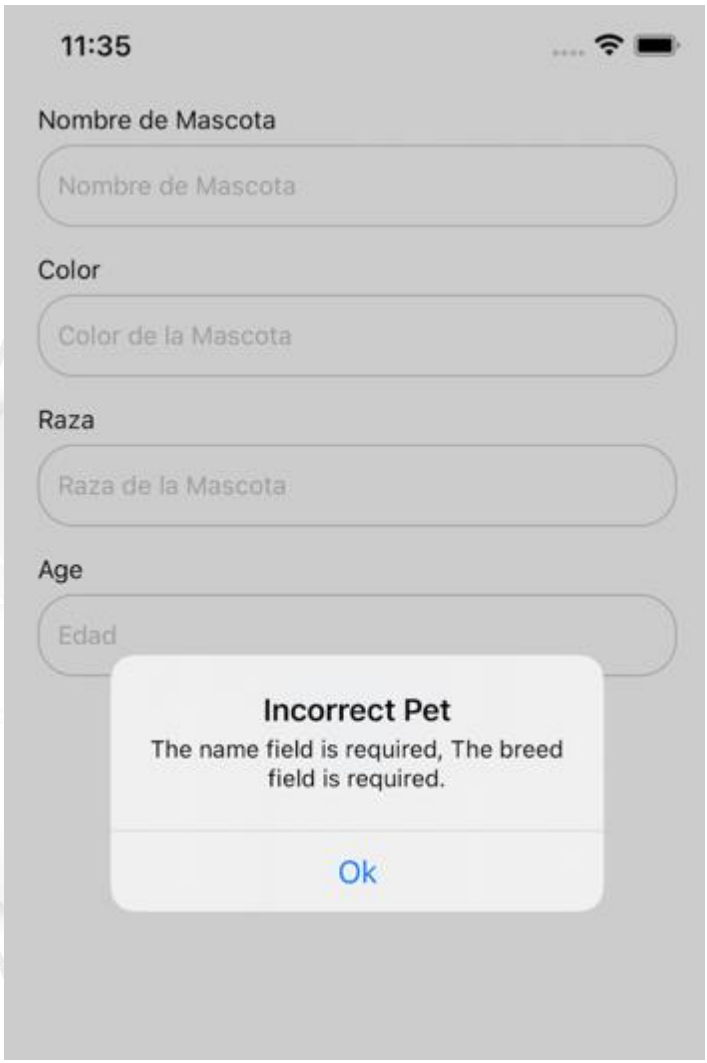
Figura 4-47: Evidencia de API para el registro de mascota sin ningún campo



Fuente. Elaboración Propia. Evidencia del registro de mascota fallido cuando no se envia ningún campo

- Registrar Mascotas (Móvil)
    1. Enviar formuarlio sin ningún campo para registrar mascota y obtener una respuesta de error (HTTP 422). **Fue exitoso**

Figura 4-48: Evidencia de Móvil para el registro de mascota sin ningún campo



Fuente. Elaboración Propia. Evidencia del registro de mascota fallido cuando no se envia ningún campo

2. Selección de nuevas características

   Debido al éxito y validación, no hay tareas implementar de los resultados del testeo.

3. Transformar los problemas del testing en nuevas tareas

   Las pruebas de testing fueron exitosas, no hay detalles en este aspecto ni tareas a ralizar.

4. Refinar tareas con diseños

No hay tareas a realizar por lo mencionado anteriormente

5. Estimar las nuevas tareas

No hay tareas a realizar por lo mencionado anteriormente

4.2. Día de Trabajo

4.2.1. Wrap-up

4.2.1.1. Seleccionar tareas

Debido a que todas las pruebas fueran exitoso, se procede a seguir con todos los requerimientos previamente analizados, que son:

- Publicar anuncio de mascota perdida(API)
- Publicar anuncio de mascota perdida(Móvil)
- Listar mascotas perdidas (API)
- Listar mascotas perdidas (Móvil)
- Reportar avistamiento de mascota (API)
- Marcar una mascota como encontrada (API)
- Marcar una mascota como encontrada (Móvil)

4.2.1.2. Discutir tareas a realizar

Se realizarán las tareas, no hubo ningún retraso respecto a requerimientos por lo que el calendario general sigue igual

4.2.2. TDD (Desarrollo Guiado de Pruebas)

4.2.2.1. Adquirir descripción de la tarea

Las pruebas guiadas se desarollarán sólo en el backend (API), ya que son las que contienen toda la lógica y el almacenamiento de la información, estas serán :

- Publicar anuncio de mascota perdida(API)
- Listar mascotas perdidas (API)
- Reportar avistamiento de mascota (API)
- Marcar una mascota como encontrada (API)

4.2.2.2. Escribir un test y correr los tests

- Publicar anuncio de mascota perdida(API)

Figura 4-49: Test fallido para Publicar anuncio de mascota perdida

Fuente. Elaboración Propia. Se escribe la prueba para publicar anuncio de mascota perdida.

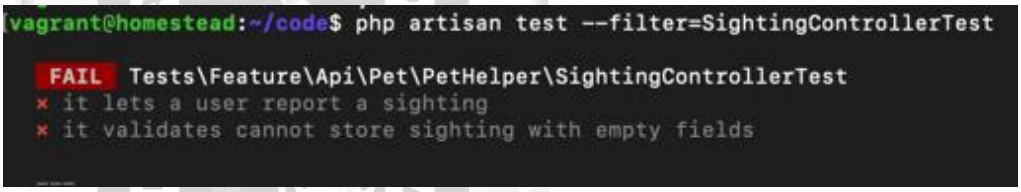
- Listar mascotas perdidas (API)

Figura 4-50: Test fallido para Listar mascotas perdidas

Fuente. Elaboración Propia. Se escribe la prueba para listar mascotas perdidas

- Reportar avistamiento de mascota (API)

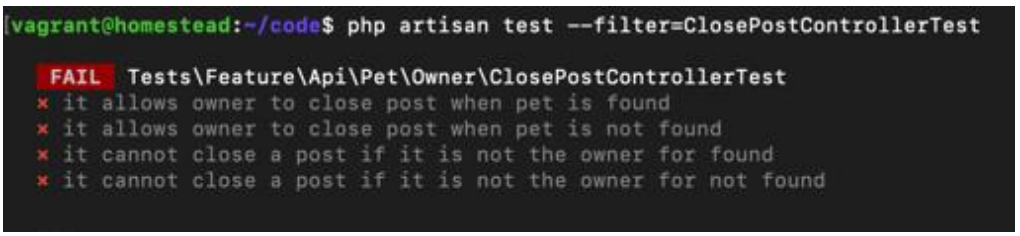
Figura 4-51: Test fallido para Reportar avistamiendo de mascota



Fuente. Elaboración Propia. Se escribe la prueba para reportar el avistamiento de mascotas

- Marcar una mascota como encontrada (API)

Figura 4-52: Test fallido para Marcar una mascota como encontrada



Fuente. Elaboración Propia. Se escribe la prueba para marcar una mascota como encontrada.

4.2.2.3. Escribir el código y correr los tests

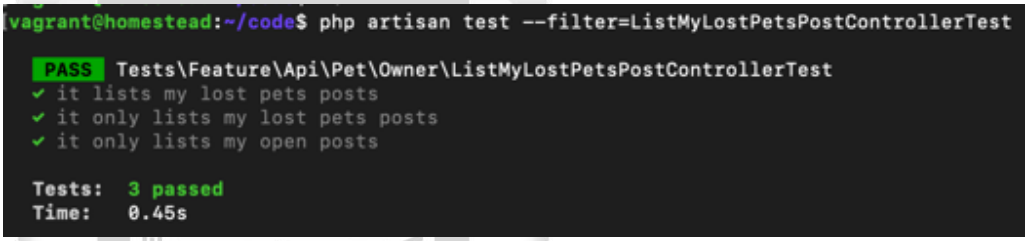
- Publicar anuncio de mascota perdida(API)

Figura 4-53: Test exitoso para publicar anuncio de mascota perdida

Fuente. Elaboración Propia. Se escribe el código para pasar la prueba de marcar una mascota como encontrada.

- Listar mascotas perdidas (API)

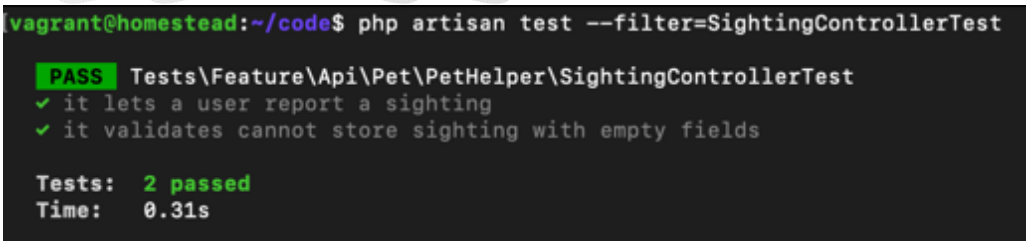
Figura 4-54: Test exitoso para listar mascotas perdidas



Fuente. Elaboración Propia. Se escribe el código para pasar la prueba de listar mascotas perdidas.

- Reportar avistamiento de mascota (API)

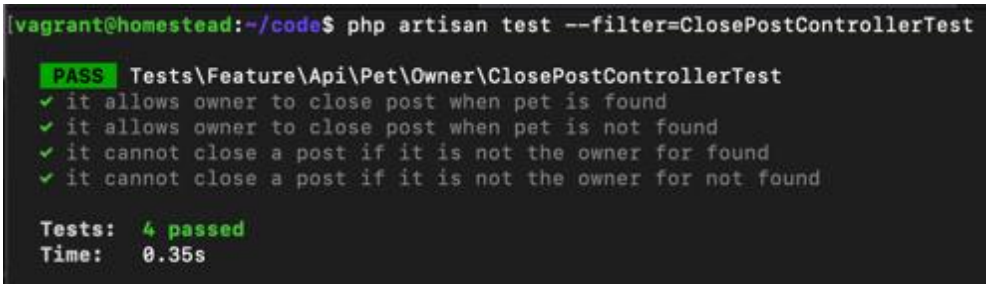
Figura 4-55: Test exitoso para reportar avistamiento de mascota



Fuente. Elaboración Propia. Se escribe el código para pasar la prueba de reportar avistamiento de mascota perdida.

- Marcar una mascota como encontrada (API)

Figura 4-56: Test exitoso para marcar mascota como encontrada



Fuente. Elaboración Propia. Se escribe el código para pasar la prueba de marcar mascota como encontrada

4.2.2.4. Refactorizar el código y correr los tests

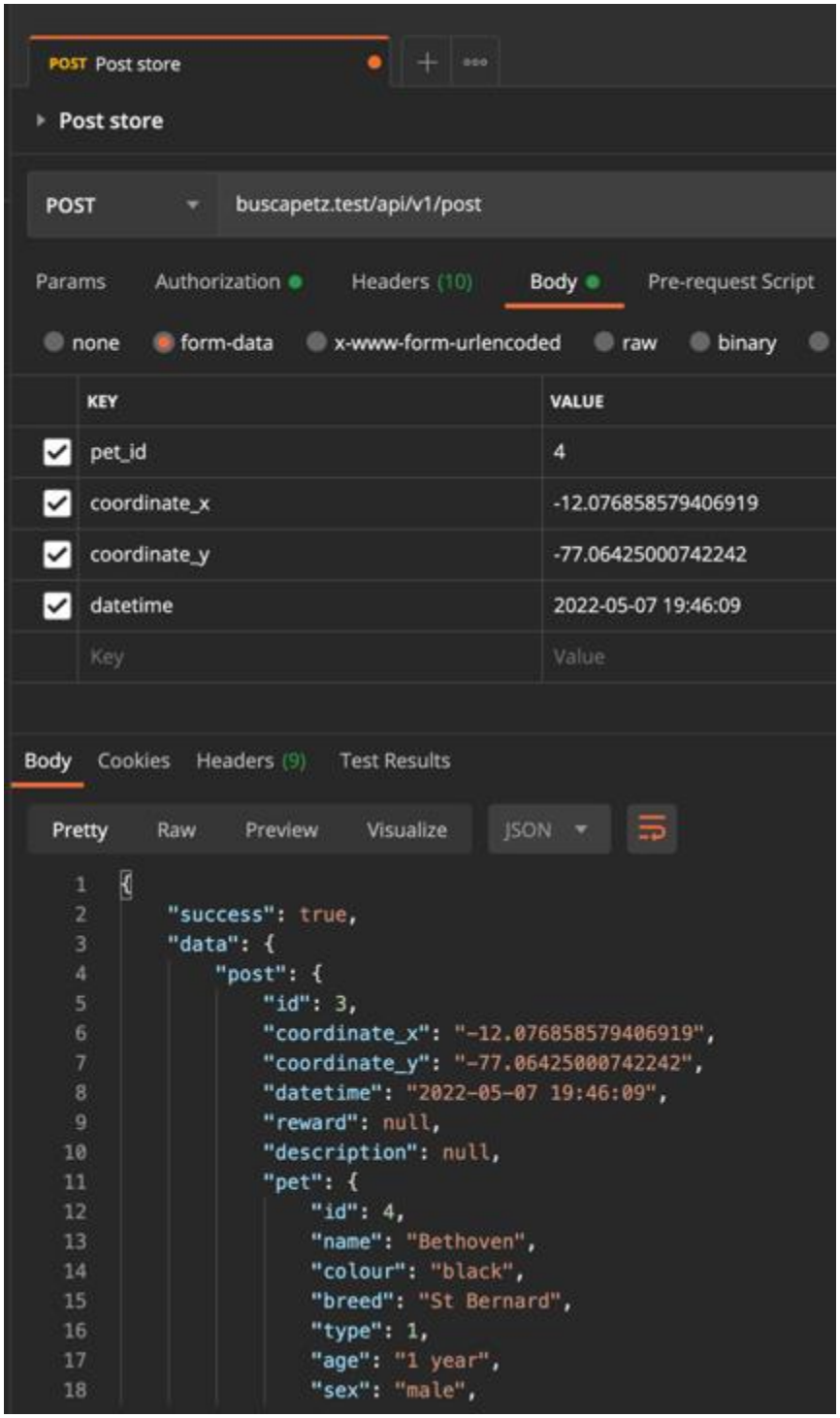
Se refactorizo métodos aplicando los conceptos SOLID y gracias a los test se pudo verificar que la integridad del sistema no se vio comprometida en lo que se refactorizó.

4.3. Día de Lanzamiento

Se procede a mostrar el funcionamiento de la API, así como la integración con el aplicatvo móvil:

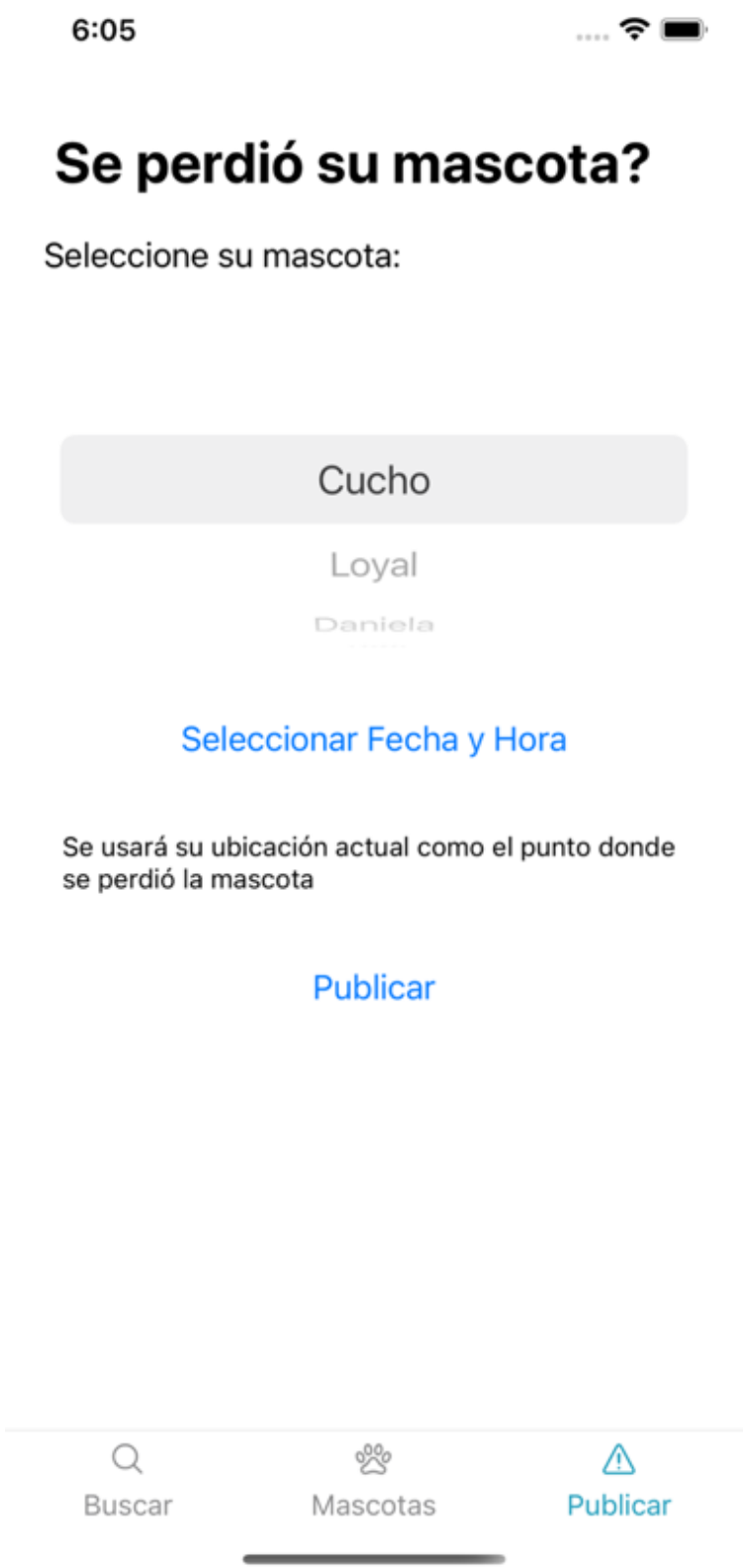
1. Publicar anuncio de mascota perdida (API)

Figura 4-57: Request al API para publicar un anuncio de mascota perdida



Fuente. Elaboración Propia. Request al API exitoso al publicar un anuncio de mascota perdida.

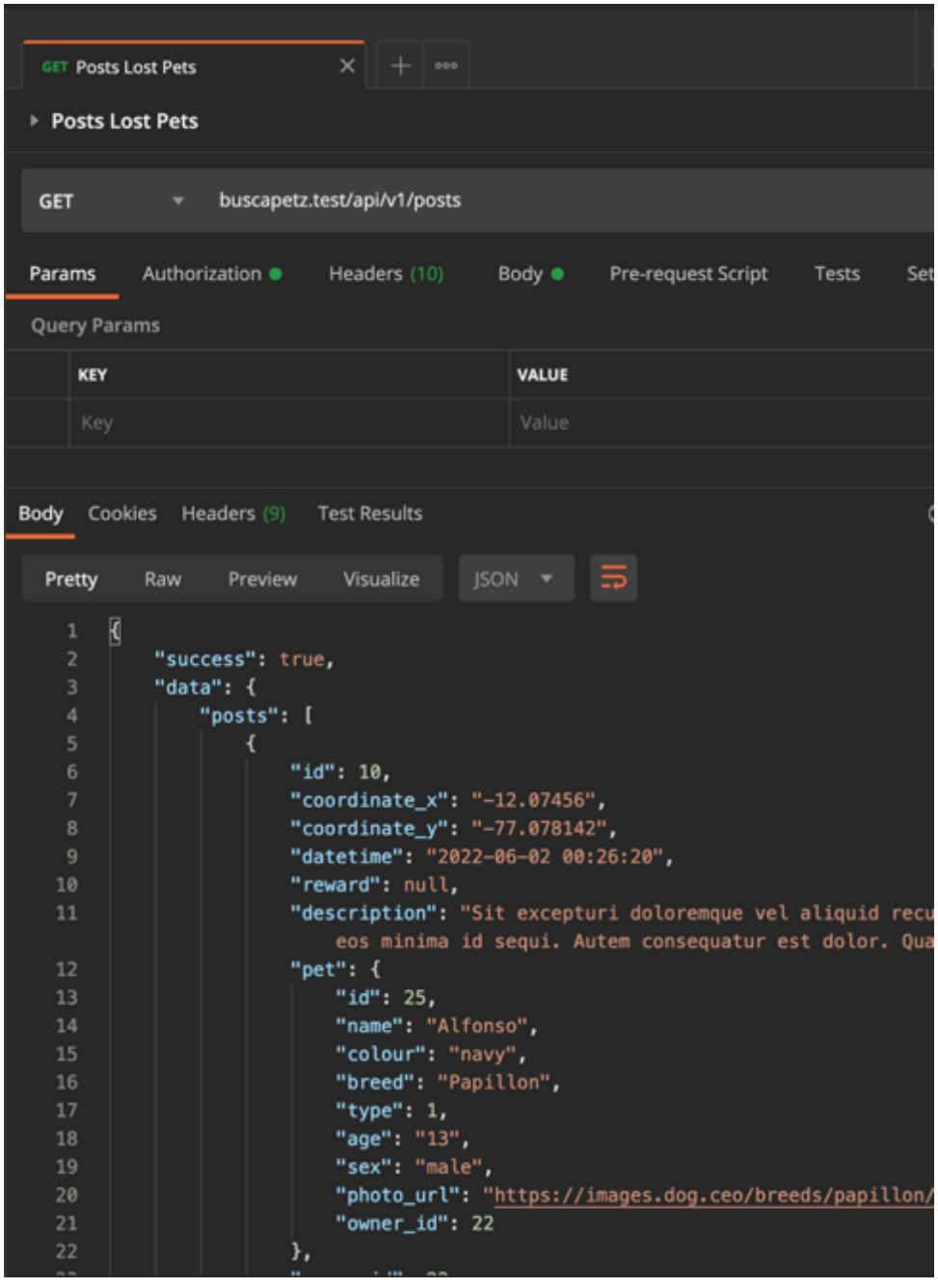
Figura 4-58: Pantalla de la aplicación móvil para Reportar una masccota como perdida



Fuente. Elaboración Propia. Pantalla de la aplicación móvil para reportar una mascota como perdida

1. Listar Mascotas Perdidas (API)

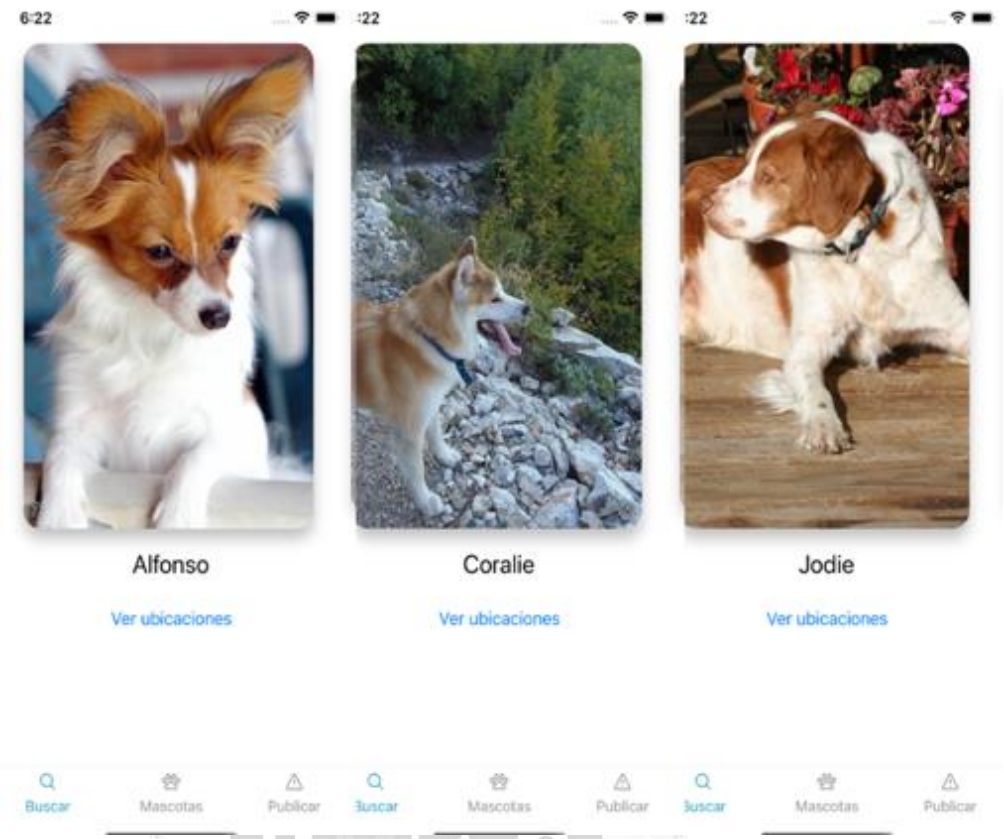
Figura 4-59: Request al API para obtener la lsita de las mascotas perdidas



Fuente. Elaboración Propia. Request al API exitoso para obtener la lista de los anuncions de mascotas perdida.

2. Listar Mascotas Perdidas (Móvil)

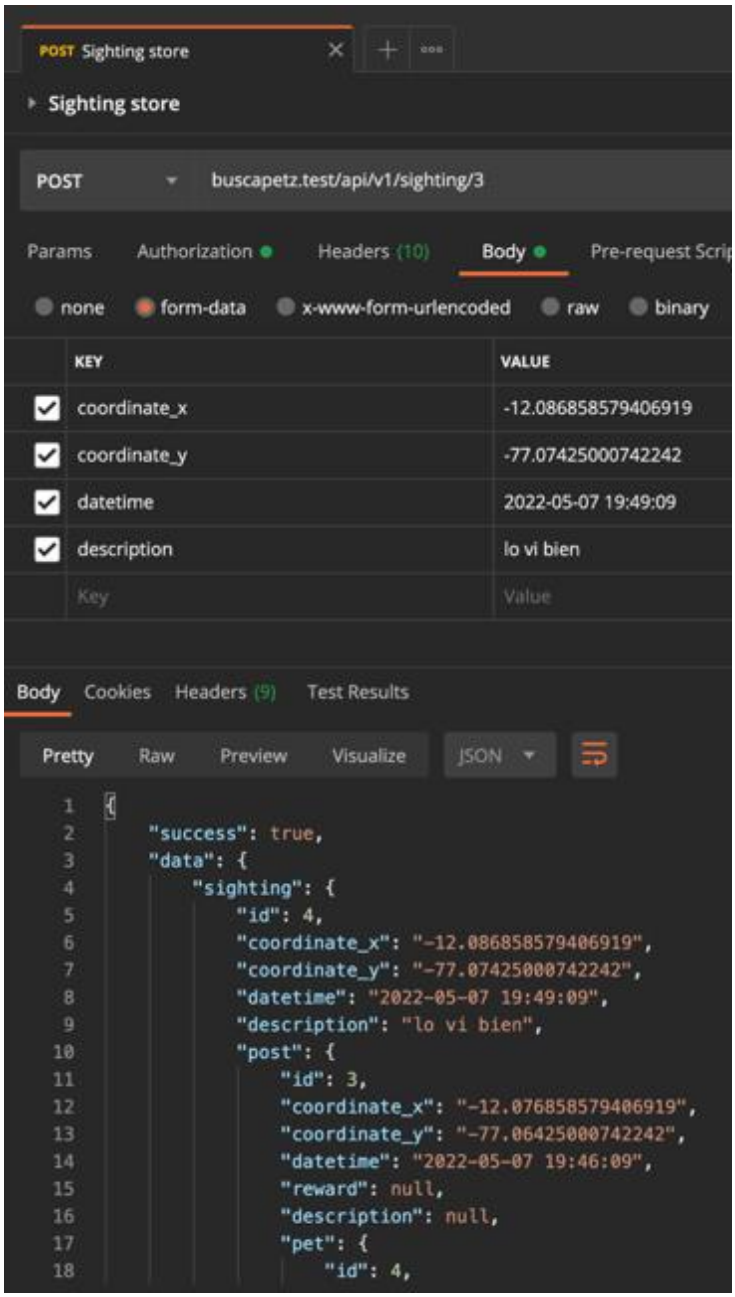
Figura 4-60: Pantalla de la aplicación móvil que muestra la lsita de mascotas perdidas



Fuente. Elaboración Propia. Pantalla de la aplicación móvil que muestra a través de un carrusel de fotos, las mascotas perdidas.

3. Reportar avistamiento de mascota (API)

Figura 4-61: Request al API para reportar el avistamiento de una mascota



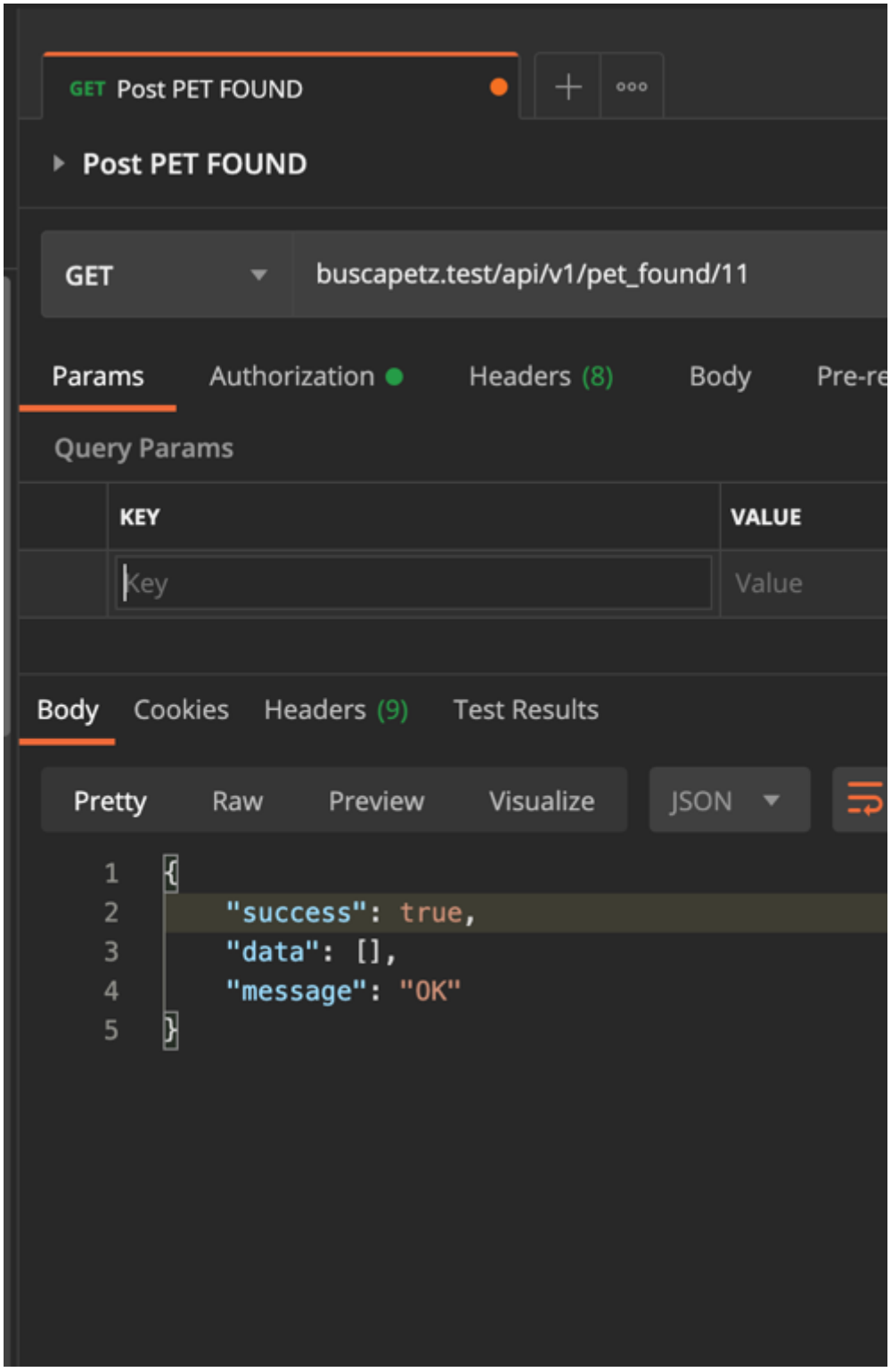
Fuente. Elaboración Propia. Request al API para reportar el avistamiento de una mascota perdida.

Figura 4-62: Pantalla de aplicación móvil para marcar el avistamiento de una mascota

Fuente. Elaboración Propia. Pantalla de la aplicación móvil para marcar el avistamiento de una mascota perdida

4. Marcar una mascota como encontrada (API)

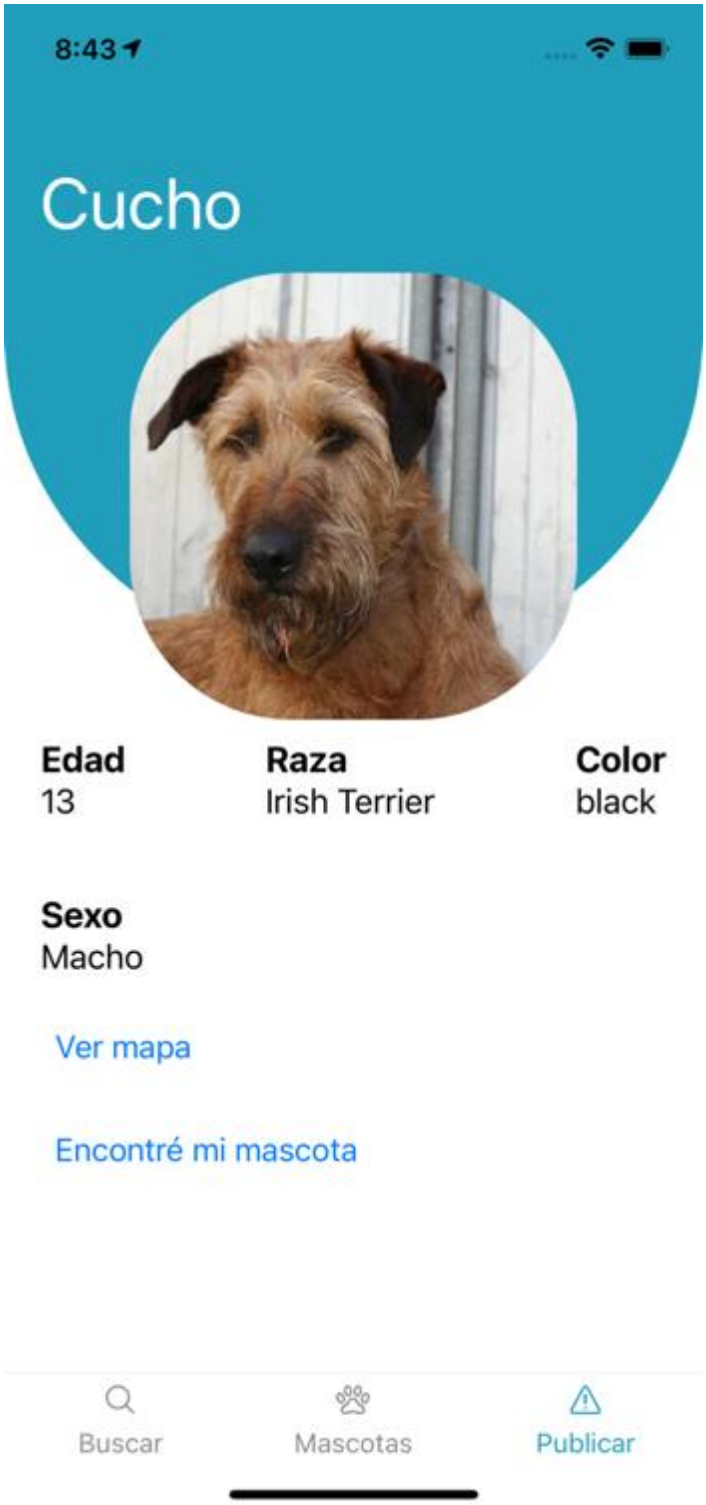
Figura 4-63: Request al API para listar las mascotas perdidas



Fuente. Elaboración Propia. Request al API para cerrar la publicación de la mascota perdida porque fue encontrada

5. Marcar una mascota como encontrada (Móvil)

Figura 4-64: Pantalla de la aplicación móvil para marcar una mascota como encontrada



Fuente. Elaboración Propia. Pantalla de la aplicación móvil para cerrar la publicación de la mascota perdida porque fue encontrada

5. Fase de Pruebas y Arreglos

5.1. Pruebas del Sistema

Previas pruebas

5.2. Día de Planeamiento

5.2.1. Análisis de Requerimientos

- Enviar notificaciones SMS (API)

  Este requerimiento permitirá notificar al dueño de la mascota de que hubo un avistamiento vía SMS, justo después de que alguien realice un avistamiento.

- Enviar notificaciones Email (API)

  Este requerimiento permitirá notificar al dueño de la mascota de que hubo un avistamiento vía email, justo después de que alguien realice un avistamiento.

5.2.2. Planeamiento de Iteración

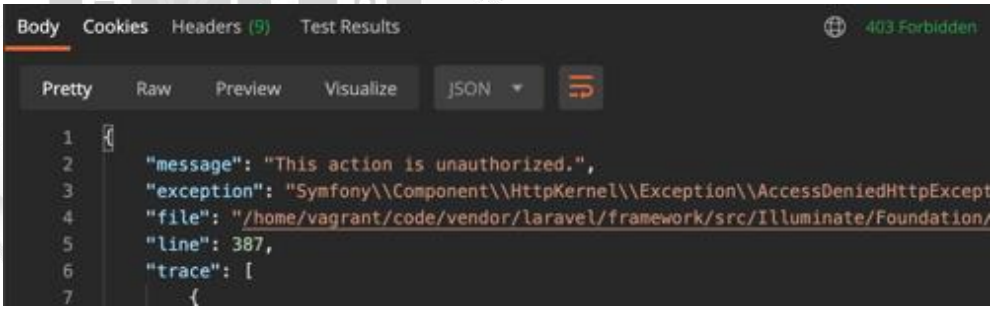
1. Feedback de testeo

   Se procedió a realizar las pruebas en los requerimientos de la fase anterior, teniendo los siguientes resultados:

- Publicar anuncio de mascota perdida(API)

  1. Realizar petición sin mascota y obtener una respuesta de error (HTTP 403). **Fue exitoso**

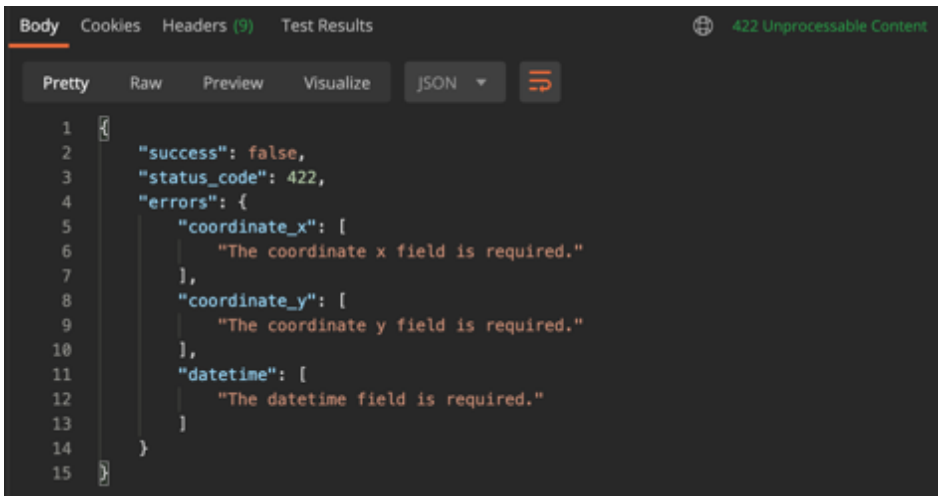
Figura 4-65: Feedback del proceso de testeo del API para publicar anuncio sin campos



Fuente. Elaboración Propia. Evidencia del registro fallido registrar un anuncio sin ningún campo

  2. Realizar petición con sólo la mascota y obtener una respuesta de error requeriendo los otros campos (HTTP 422). **Fue exitoso**

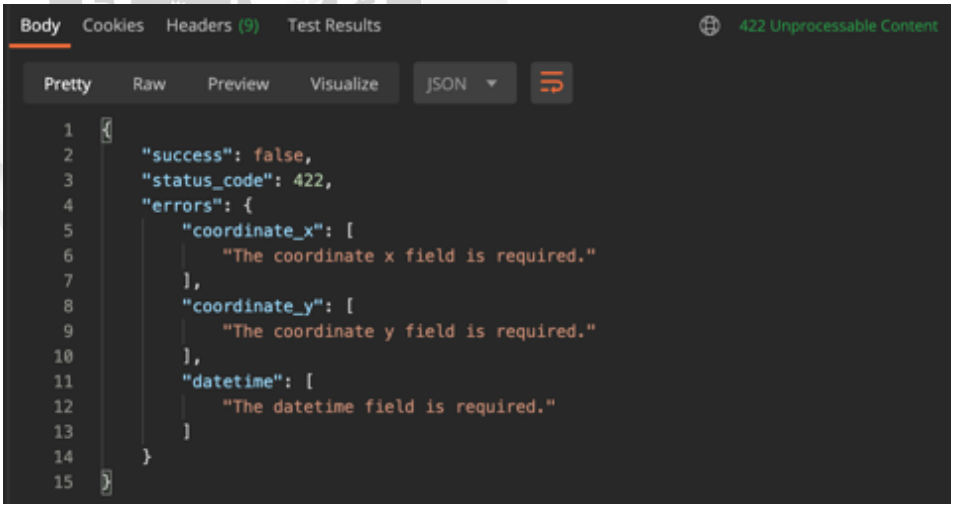
Figura 4-66: Feedback del proceso de testeo del API para publicar anuncio enviando sólo mascota



Fuente. Elaboración Propia. Evidencia del registro fallido de un usuario cuando no ingresa ninguno de los campos

- Reportar avistamiento de mascota (API)
  1. Realizar petición sin ningún mascota y obtener una respuesta de error (HTTP 22). **Fue exitoso**

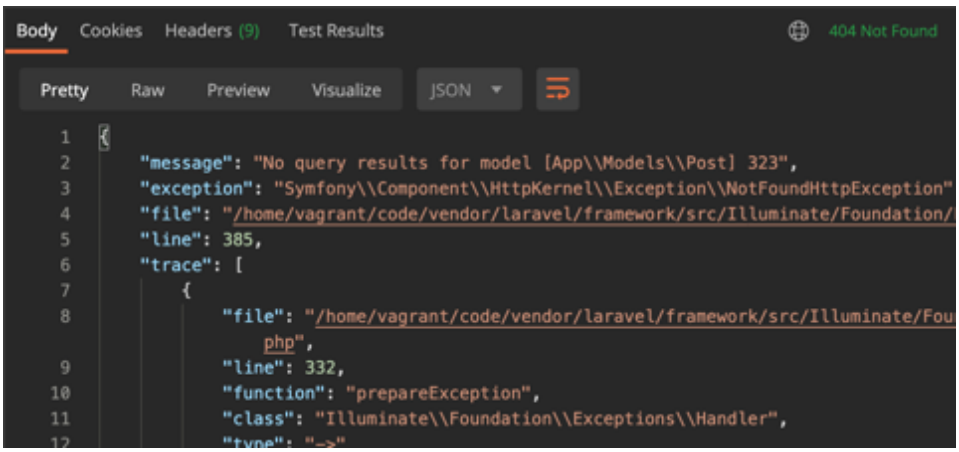
Figura 4-67: Feedback del proceso de testeo del API para reportar avistamiento



Fuente. Elaboración Propia. Evidencia del registro fallido de reportar cuando no envia ningún campo

- Marcar una mascota como encontrada (API)
  1. Realizar petición mandado una mascota que no existe y obtener una respuesta de error (HTTP 403). **Fue exitoso**

Figura 4-68: Feedback del proceso de testeo de marcar una mascota como encontrada



Fuente. Elaboración Propia. Evidencia del registro fallido de marcar una mascota como encontrada al mandar una mascota que no existe

2. Selección de nuevas características

   Debido al éxito y validación, no hay tareas implementar de los resultados del testeo.

3. Transformar los problemas del testing en nuevas tareas

   Las pruebas de testing fueron exitosas, no hay detalles en este aspecto ni tareas a ralizar.

4. Refinar tareas con diseños

   No hay tareas a realizar por lo mencionado anteriormente

5. Estimar las nuevas tareas

   No hay tareas a realizar por lo mencionado anteriormente

5.3. Día de Trabajo

5.3.1. Wrap-up

Debido a que todas las pruebas fueran exitoso, se procede a seguir con todos los requerimientos previamente analizados, que son:

- Enviar notificación por SMS cuando hay un avistamiento
- Enviar notificación por email cuando hay un avistamiento

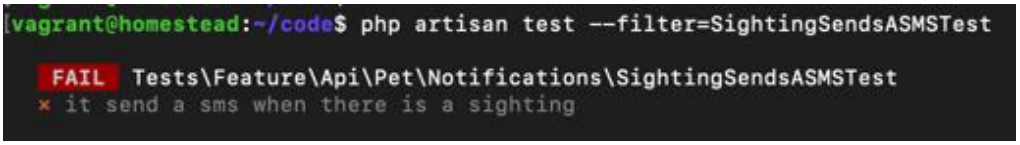
5.3.2. TDD (Desarrollo Guíado de Pruebas)

5.3.2.1. Adquirir descripción de la tarea

Las pruebas guiadas se desarollarán sólo en el backend (API), ya que son las que contienen toda la lógica y el almacenamiento de la información, estas serán :

- Enviar notificación por SMS cuando hay un avistamiento

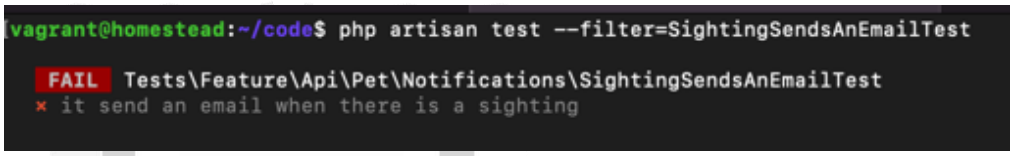
Figura 4-69: Test fallido para enviar notificaciones SMS



Fuente. Elaboración Propia. Se escribe la prueba para enviar notificaciones SMS cuando hubo un avistamiento

- Enviar notificación por email cuando hay un avistamiento

Figura 4-70: Test fallido para enviar notificaciones email

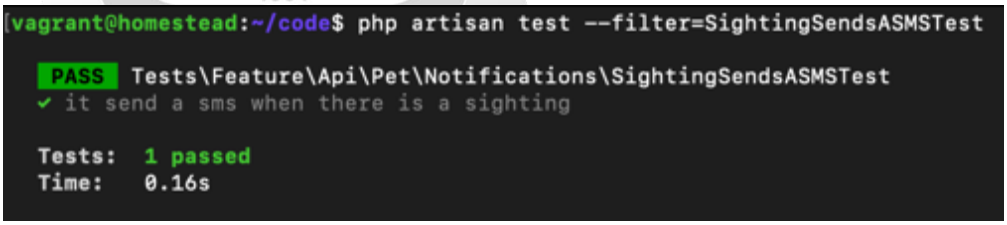


Fuente. Elaboración Propia. Se escribe la prueba para enviar notificaciones email cuando hubo un avistamiento

5.3.2.2. Escribir el código y correr los tests

- Enviar notificación por SMS cuando hay un avistamiento

Figura 4-71: Test exitoso para enviar notificaciones SMS



Fuente. Elaboración Propia. Se escribe el código para pasar la prueba de enviar una notificación SMS cuando hay un avistamiento

- Enviar notificación por email cuando hay un avistamiento

Figura 4-72: Test exitoso para enviar notificaciones email

Fuente. Elaboración Propia. Se escribe el código para pasar la prueba de enviar una notificación email cuando hay un avistamiento

5.3.3. Refactorización

Se refactorizo métodos aplicando los conceptos SOLID y gracias a los test se pudo verificar que la integridad del sistema no se vio comprometida en lo que se refactorizó.

5.4. Día de Lanzamiento

Se procede a probar las funcionalidades en la integración del backend (API) con el móvil para los siguientes requerimientos:

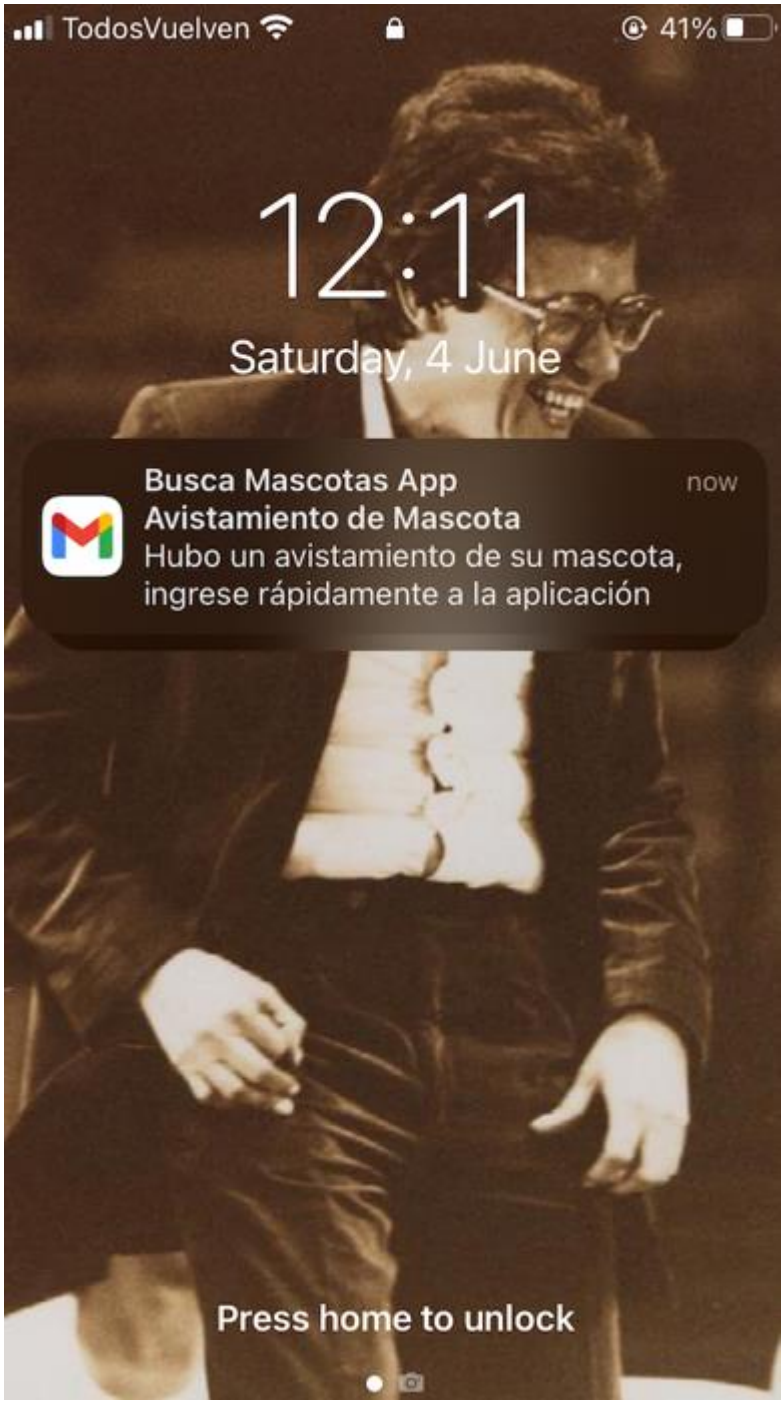
- Enviar notificación por SMS cuando hay un avistamiento

Figura 4-73: Notificación SMS recibida tras avistamiento

Fuente. Elaboración Propia. El dueño de la mascota recibirá una notificación SMS indicando que hubo un avistamiento.

- Enviar notificación por email cuando hay un avistamiento

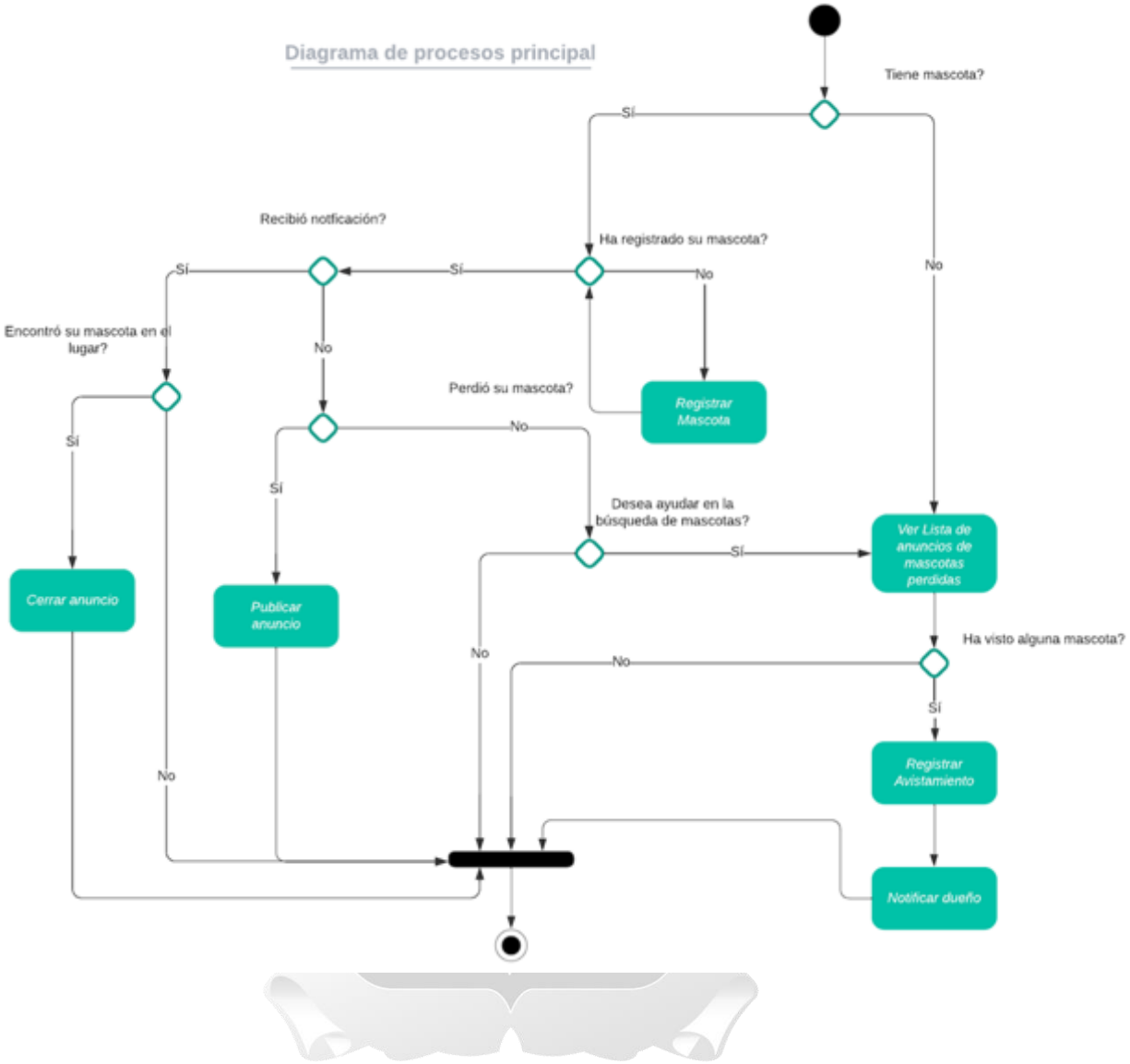
Figura 4-74: Notificación email recibida tras avistamiento



Fuente. Elaboración Propia. El dueño de la mascota recibirá una notificación email indicando que hubo un avistamiento.

4.2 Descripción de los artefactos elaborados

Diagrama de Procesos Principal



4.3 Descripción de la solución tecnológica

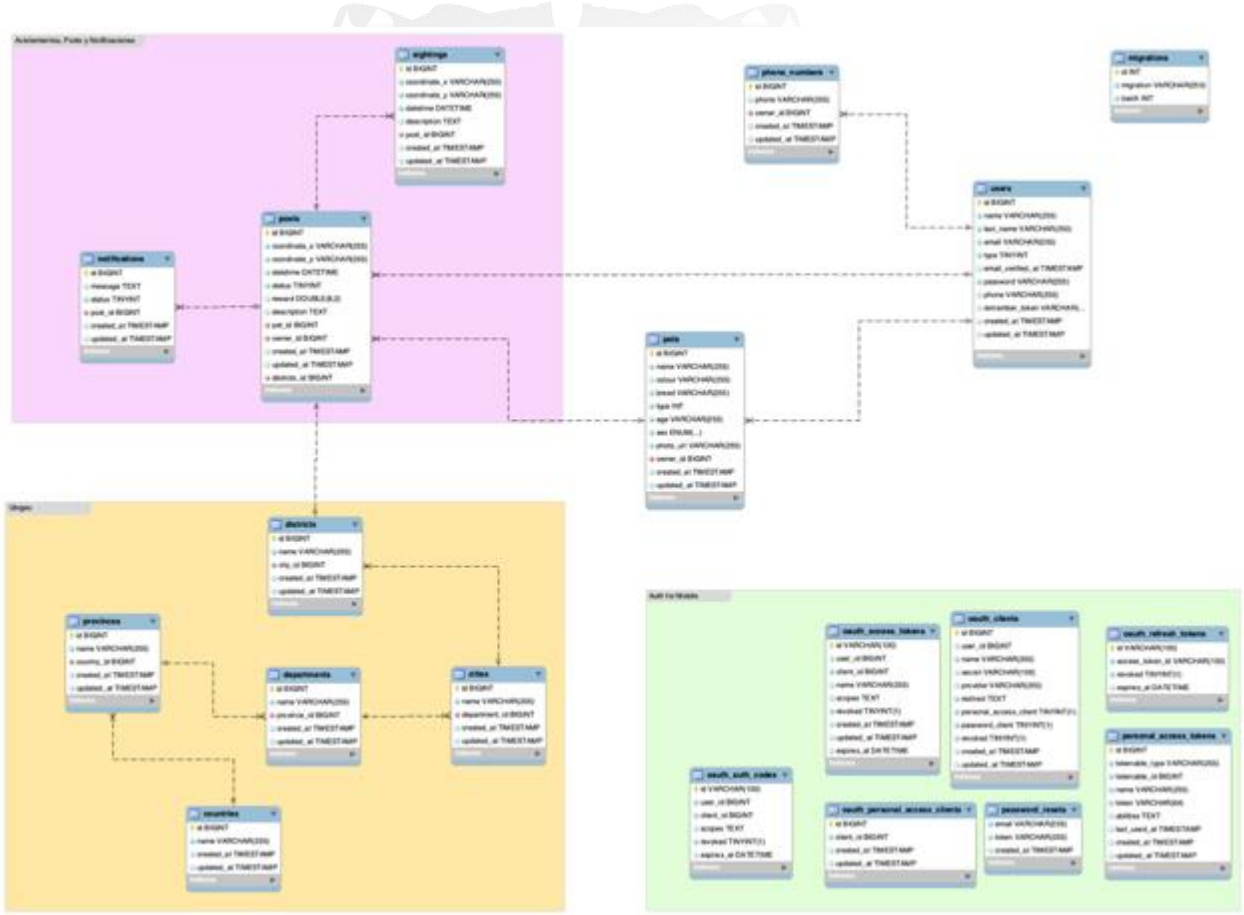
La aplicación móvil se desarrollo en base a una arquitectura de cliente-servidor, donde todas los clientes serán los dispositivos móviles.

El backend se realizo con PHP siguiendo los principios SOLID y con las prácticas de PSR-2 y PSR-4, todo fue realizado con TDD (Test Driven Development) para asegurar un software de calidad y resistente a cambios.

Diseño de Base datos

Se muestra el diseño de la base de datos que maneja el API para brindar los servicios a la aplicación, usando las prácticas de normalización.

Figura 4-75: Diseño de la Base de Datos

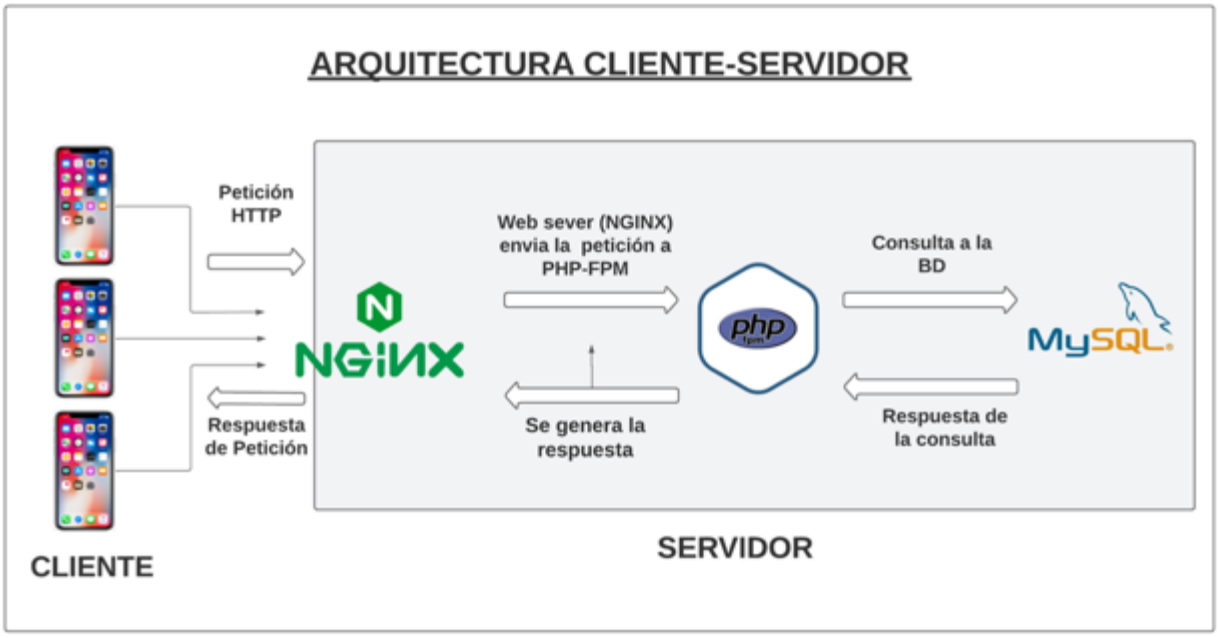


Fuente: Elaboración Propia

Arquitectura

Se describe la arquitectura que se ha usado a lo largo del proyecto

Figura 4-76: Arquitectura Cliente-Servidor

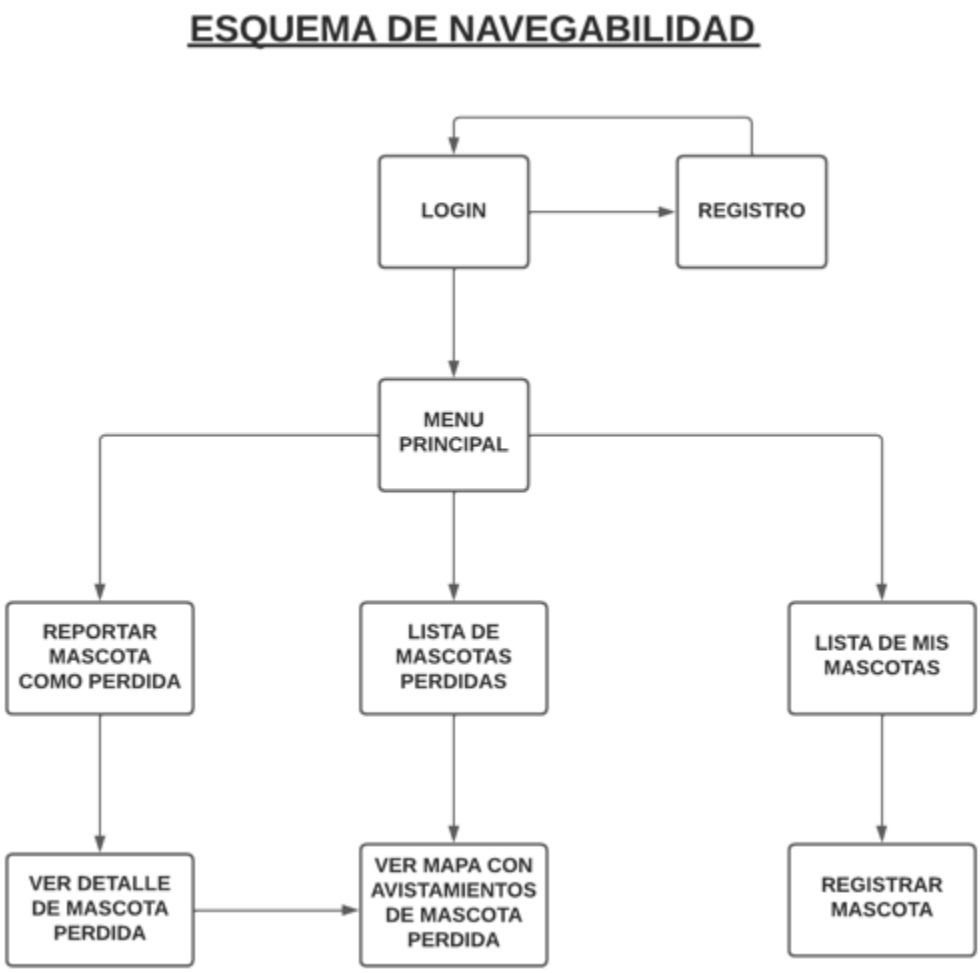


Fuente: Elaboración Propia

Esquema de Navegabilidad

Se brinda el esqueam de navegación general de la aplicación móvil.

Figura 4-77: Esquema de Navegabilidad



Fuente: Elaboración Propia

CAPÍTULO 5: VALIDACIÓN DE LA SOLUCIÓN TECNOLÓGICA

En el siguiente capítulo se presenta com objetivo la validación de la solución tecnológica de los objetivos general y específicos planteados en el primer capítulo.

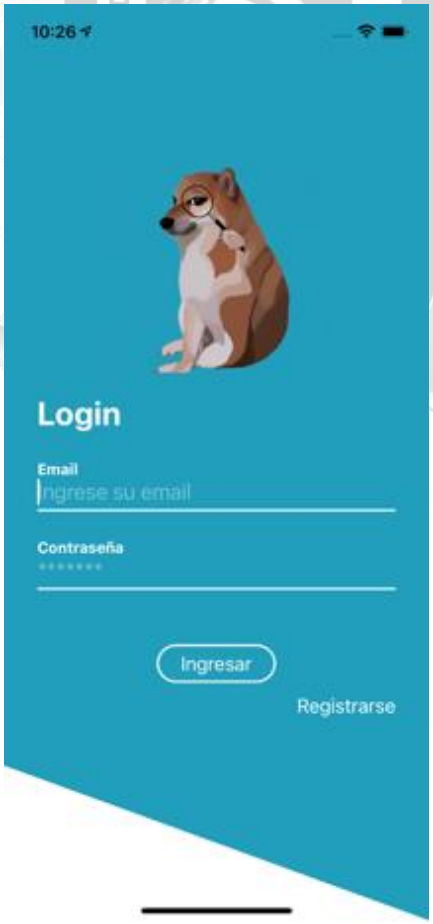
1. Desarrollar una aplicación móvil para optimizar la búsqueda de mascotas perdidas en Pueblo Libre a través de puntos GPS para el rastreo de avistamientos

   La validación del objetivo general anterior mencionado es demostrar que a través de una solución tecnológica, como es una aplicación móvil puede generar un gran beneficio para dueños de mascotas que pasan por el lamentable episodio que es el extravío de sus queridos animalitos.

   Es por medio de esta aplicación móvil que el dueño podrá hacer la publicación inmediata de su mascota, así como recibir inmediatamente los avistamientos marcados por otros usuarios donde se ha visto la mascota en un mapa que permita reorientar la dirección de la búsqueda.

   El aplicativo móvil contiene una pantalla de Login, desde la que se permitirá el ingreso al sistema al usuario, además desde la misma el usuario se podrá registrar si es que no cuenta con credenciales. [Ver figura 5-1]

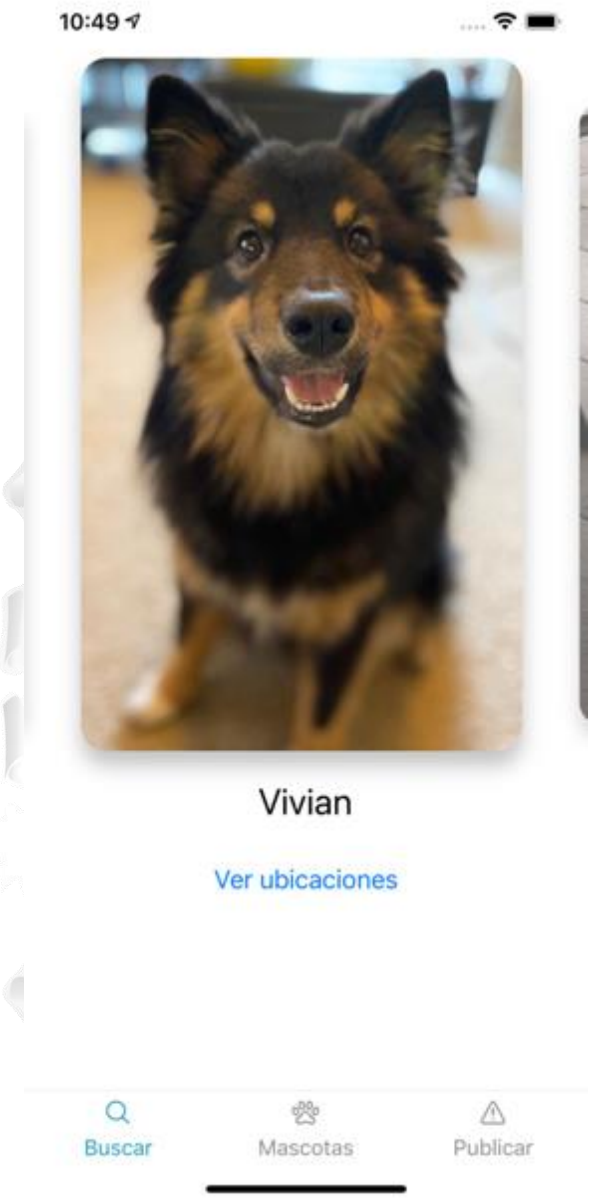
Figura 5-1: Pantalla para iniciar sesión en la aplicación móvil



Fuente. Elaboración Propia. Es la pantalla inicial de la aplicación

Después de la validación de las credenciales respectivas la aplicación mostrará un menú que permitirá el fácil acceso a distintos menús para las diferentes acciones que provee. [Ver figura 5-2]

Figura 5-2: Pantalla después de iniciar sesión

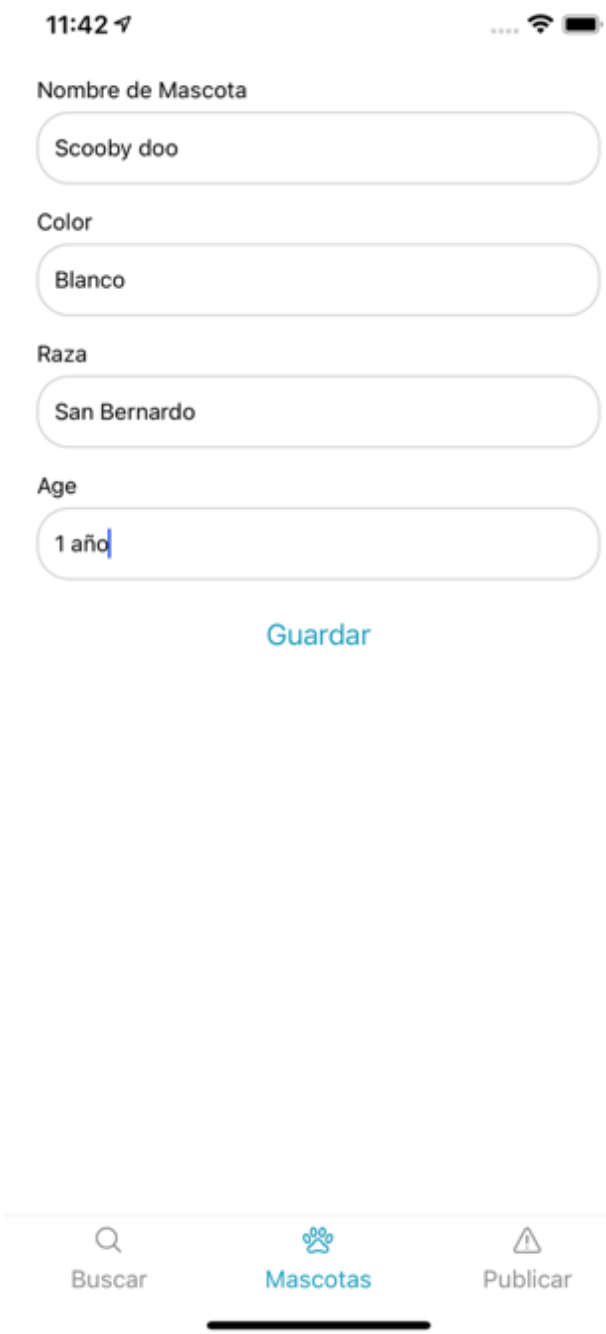


Fuente. Elaboración Propia. Pantalla después de un logueo exitoso

2. Permitir crear anuncios de una mascota perdida que sean visible por todos los usuarios de la aplicación.

   Respecto a este objetivo, se dio la facilidad al usuario de registrar sus mascotas [Ver Figura 5-3, 5-4] previamente, sin necesidad de esperar el extravío de alguno .

Figura 5-3: Registro de una mascota



Fuente. Elaboración Propia. Formulario con los campos necesarios para realizar el registro de una mascota

Figura 5-4: Visualización de mis mascotas

Fuente. Elaboración Propia. Pantalla que muestra las mascotas pertenecientes por un usuario

Ya que se cuenta con el previo de registro de las mascotas, la aplicación permite realizar la publicación del anuncio de perdida en menos de 4 segundos [Ver Figura 5-5] , ya que para publicar el anuncio sólo es necesario seleccionar la mascota fecha, hora y la aplicación tomará la ubicación del usuario en ese momento.

Figura 5-5: Registro para publicar el anuncio de mascota perdida

Fuente. Elaboración Propia. Pantalla que muestra las mascotas pertenecientes por un usuario

Para verificar que el proceso se realizó con éxito debería poder observar que su mascota se encuentra en la lista de anuncios de mascotas pérdidas. [Ver Figura 5-6]

Dada la velocidad con la que se puede publicar un anuncio (4 segundos) con los datos esenciales para la identificación de la mascota se puede dar este objetivo como validado.

Figura 5-6: Pantalla de los anuncios de mascotas perdidas

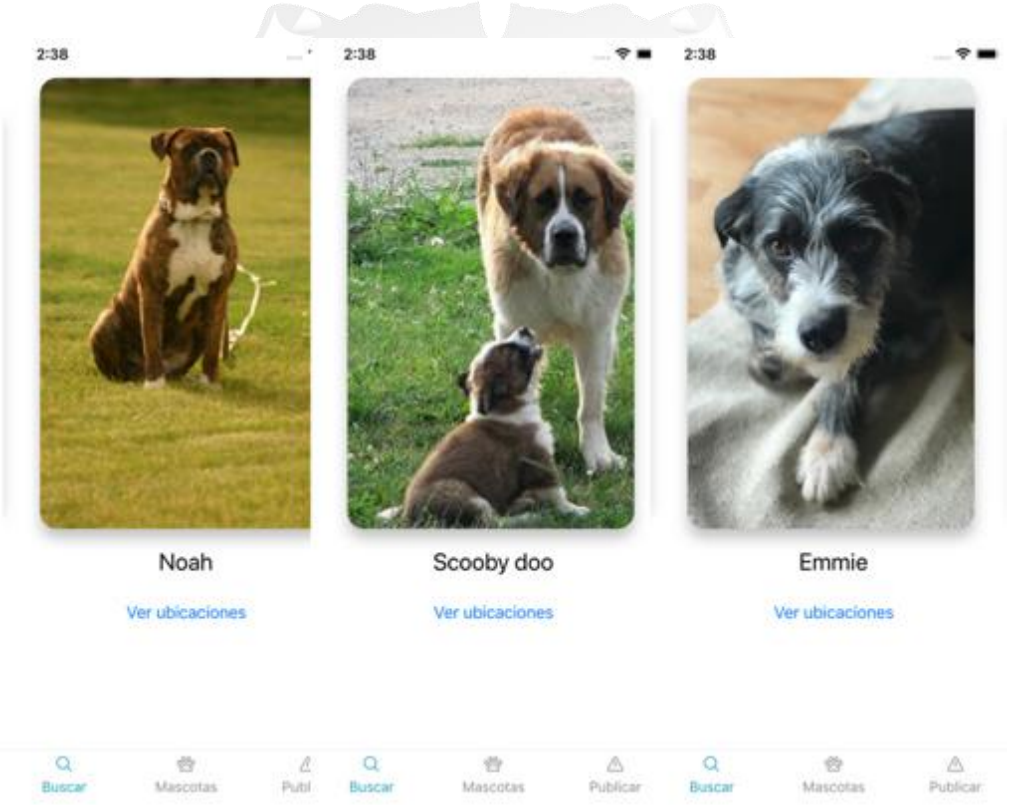
Fuente. Elaboración Propia. Pantalla que muestra la foto de las mascotas perdidas

3. Permitir registrar el avistamiento de una mascota para permitir trazar un recorrido de donde ha sido visto y cambiar el rango de búsqueda.

   Respecto a este objetivo específico, el cual es fundamental para realizar el objetivo general, se implementó a través de la tecnología GPS el poder obtener las coordenadas exactas del usuario en el momento en que disponga marcar un avistamiento de la mascota perdida, lo cual dispone obtener una lista de todos los avistamientos hechos por usuarios por la hora y fecha, a través de esos datos se podrá marcar en un mapa la dirección o rumbo en el que la mascota se dirige. El dueño de la mascota a través de la aplicación podrá orientar la búsqueda o cambiar el rango según los avistamientos, lo cual reducirá considerablemente el tiempo de búsqueda y realizar búsqueda en lugares erróneos.

   Para adjuntar evidencia del proceso, el usuario tiene que estar previamente autenticado en la aplicación, y seleccionar el anuncio de la mascota que fue reconocida, ingresar a Ver Ubicaciones [Ver Figura 5.7]

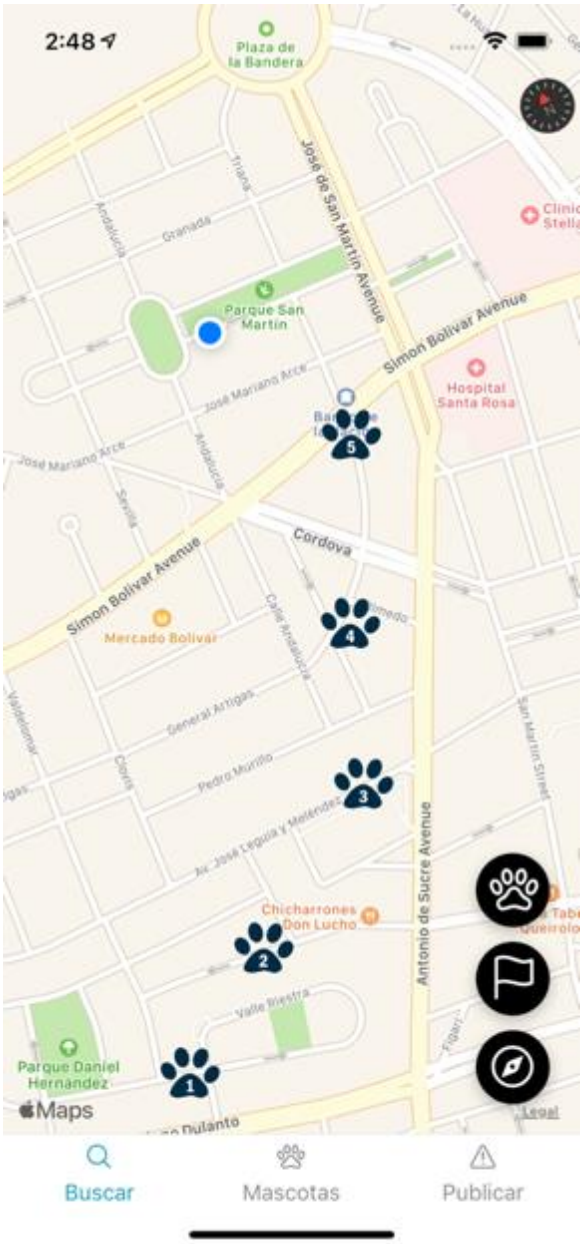
Figura 5-7: Tres Pantalla de mascotas perdidas



Fuente. Elaboración Propia. Pantalla que muestra laas fotos de tres mascotas perdidas

Después de ello, sólo necesitaría seleccionar la huella del animal en el mapa, el cual tomará la localización actual del usuario y quedará registrada [Ver Figura 5.8 ,y 5.9]

Figura 5-8: Pantalla que muestra los avistamientos de la mascota



Fuente. Elaboración Propia. Pantalla que muestra los avistamientos marcados por otros usuarios y que son visibles para todos

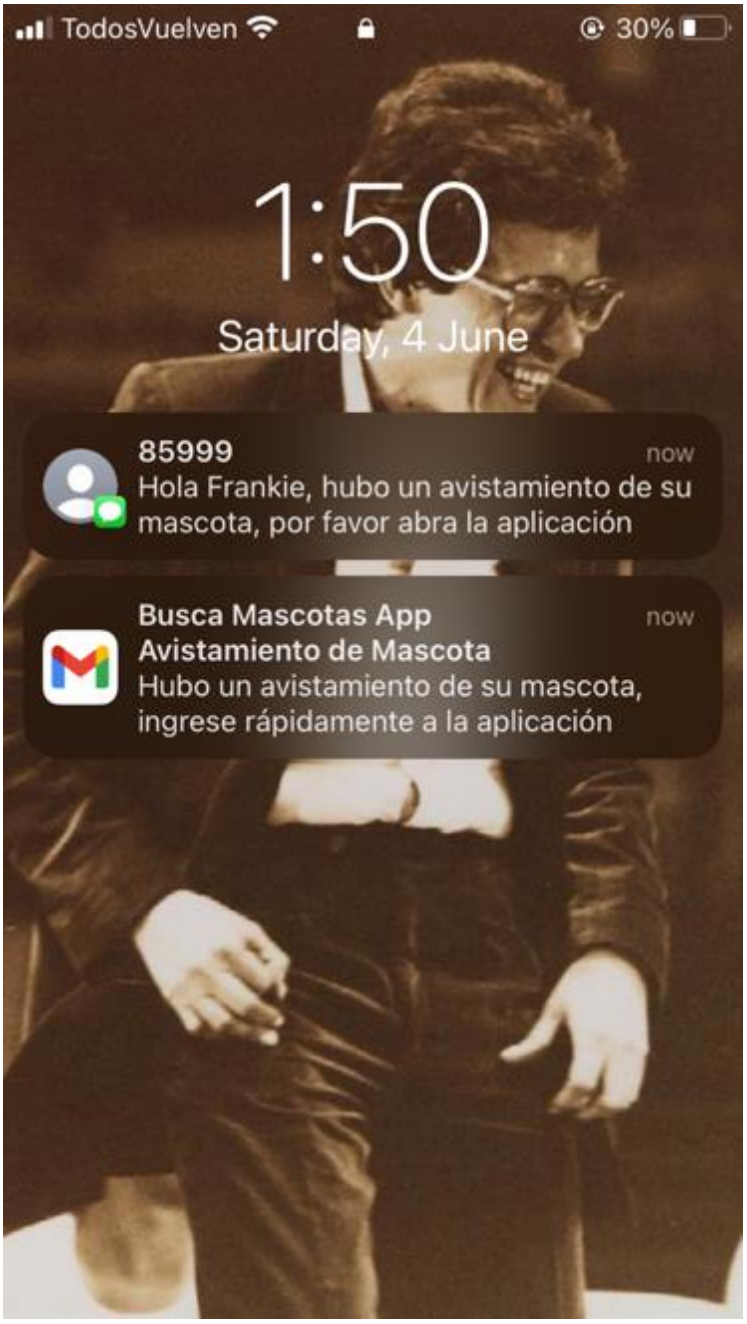
Figura 5-9: Pantalla que muestra el avistamiento marcado

Fuente. Elaboración Propia. Pantalla que muestra la pocisión actual del usuario y que es marcada y es reflejada en la aplicación

4. Notificar por email y SMS al dueño de la mascota en caso de un avistamiento.

   Este objetivo permite alertar al dueño de la mascota de que hubo un avistamiento, el cual ha sido almacenada en unas coordenadas GPS, las cuales gracias a la aplicación puede ser visualizado fácilmente en un mapa. El dueño de la mascota podrá acercarse rápidamente a la ubicación para confirmarlo y cerrar el anuncio.

Figura 5-10: Pantalla que muestra las notificaciones de email y SMS



Fuente. Elaboración Propia. Pantalla que muestra las notificaciones email y SMS que recibe el dueño de la mascota cuando hubo un avistamiento

5. Implementar la aplicación con la metodología MOBILE-D.

   Como se pudo observar en el capítulo III y IV , se implemento la metodología MOBILE-D, definiendo las fases y subfases que se aplicarían al proyecto, una de las ventajas que ha proporciona MOBILE-D es la adaptabilidad, es una metodología que define muchas fases pero en la que no todas necesitan ser implementadas, cada una varía de acuerdo al proyecto, además de la subfases.

   En una opinión personal, la primera fase de MOBILE-D es algo tediosa y puede desanimar a muchos de seguir implementandola, pero una vez definido todo las demás fases se hacen cada vez más fácil, debido a que fase siguiente repite algunas subfases de la anterior, que ya son familiares y más fáciles de implementar, recomiendo totalmente el uso de esta metodología para proyectos donde se requiera una documentación y planeamiento riguroso pero no se cuenten con todos los requerimientos definidos, ya que es en cada fase donde se va definiendo los siguientes requerimientos, como se realizarían en una metodologia ágil.

CONCLUSIONES

La aplicación móvil colaborativa para la búsqueda de mascotas caninas perdidas en Pueblo Libre mediante GPS y notificaciones por email y SMS beneficia inmesurablemente a los dueños que han perdido a su mascota, por el ahorro del tiempo y la eficiencia que lleva el uso de la herramienta.

Basado en la premisa anterior, se puede extraer las siguientes conclusiones del presente trabajo de investigación:

- La implementación del aplicativo para la búsqueda de mascotas caninas permitió a los usuarios publicar el anuncio de una mascota perdida en la aplicación con tan sólo 3 interacciones previo registros de su mascota, lo que se logró haciendo una aplicación rápida, estable y intuitiva al usuario.
- La implementación del aplicativo para la búsqueda de mascotas caninas permitió a los usuarios registrar el avistamiento de un perfil en búsqueda, desde cualquier ubicación, con tan sólo buscar el perfil de la mascota perdida, ubicarse en el mapa y marcar el avistamiento, este gran logro fue posible gracias al uso y acceso del GPS, además de los marcadores de los avistamientos ubicados en el mapa, lo que permite orientar o cambiar el rumbo de la búsqueda
- La implementación del aplicativo para la búsqueda de mascotas caninas permitió notificar vía email y sms al dueño de la mascota. Gracias a una correcta implementación de los servicios de mensajería de email.y texto, tan pronto como hay un avistamiento de una mascota se procede a notificar al dueño, permitiendo mantener al dueño muy al tanto de donde orientar su búsqueda o acercarse para verificar si es su mascota. Es este objetivo específico el que marca la diferencia, al brindar al usuario una herramienta fiable que le comunique mediantes dos medios, asegurando una alta disponibilidad.
- La implementación del aplicativo para la búsqueda de mascotas caninas no hubiese sido posible sin la metodología MOBILE-D, permitiendo un ciclo de desarrollo de software iterativo marcando los hitos y objetivos en cada fase y sub fase de desarrollo, beneficiando enormemente el flujo de planeación y desarrollo. Gran parte del tiempo de este trabajo fue destinado a la lectura de los documentos de Mobile-D, debido a no encontrar referentes para su implementación, muchos de los cuales implementaron sólo los títulos de las fases pero no ahondaron las subfases y objetivos de cada una de ellas que en mi experiencia es lo más importante.

RECOMENDACIONES

Sobre las recomendaciones del presente trabajo, se procede a detallar lo siguiente:

- Se recomienda hacer de más uso de los datos del registro de mascota, de forma que permita hacer más eficiente la aplicación, como agregando filtros.
- Se recomienda poder permitir el usuario establecer un avistamiento desfasado, es decir permitir al usuario realizar un avistamientono no necesariamente en su ubicación actual, debido a que pudo haber visto una mascota perdida, pero no estar enterado hasta mucho después.
- Se recomienda permitir al usuario registrar más de un correo y teléfono para las notificaciones, así como implementar las notificaciones a nivel de aplicación, en caso de contar con un presupuesto alto implementar notificaciones por Whatsapp y llamada.
- Se recomienda en caso de querer hacer uso de Mobile-D leer toda la documentación brindada por la web oficial, es muy extensa, pero a su vez muchas subfases se repiten por lo que al pasar la segunda fase se hace más fácil de implementar.

REFERENCIAS BIBLIOGRÁFICAS

Almario, M. y Rubiano, K. (2017) . GeoPetFinder: Aplicación para dispositivos móviles para la búsqueda de perros extraviados en la ciudad de Bogotá. [Tesis de Pregrado, Universidad Distrital Francisco José de Caldas]. https://repository.udistrital.edu.co/handle/11349/6510

Amaya, Y (2013). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles

Andina (2018) Unas 420,000 mascotas se pierden cada año en Perú por falta de identificación https://andina.pe/agencia/noticia-unas-420000-mascotas-se-pierden-cada-ano-peru-falta-identificacion-732435.aspx

Andrade, K. (2018). Desarrollo de un sistema web y móvil de registro y control de mascotas del gobierno autónomo descentralizado municipal de Otavalo, para las plataformas Ios Y Android. [Tesis de Pregrado, Universidad Técnica del Norte]. http://repositorio.utn.edu.ec/handle/123456789/8190

AVMA (2021). On Check the Chip Day, a lifesaving reminder to ensure pets are microchipped with current contact info https://www.avma.org/news/press-releases/check-chip-day-lifesaving-reminder-ensure-pets-are-microchipped-current-contact

Bitbucket (2021). Breve presentación de BitBucket https://bitbucket.org/product/es/guides/getting-started/overview

Blanes, J. (2020). ¿Qué es React Native?. Deloitte https://www2.deloitte.com/es/es/pages/technology/articles/que-es-react-native.html

Kantar (2018). Casi la Mitad de hogares peruanos tienen una mascota https://www.kantar.com/latin-america/inspiracion/consumo-masivo/hogares-con-mascotas

Kat Albrecht (2018). Lost Dog Behavior https://www.missinganimalresponse.com/lost-dog-behavior/

INEI (2017). 6. Características de los Hogares https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1539

IBM (2020). Application Programming Interface (API) https://www.ibm.com/cloud/learn/api

Mendez, A., Villafañe, A., Martinez, J. y Criollo, J. (2019). Aplicación móvil para adopción de mascotas abandonadas “peluditos.com”. [Tesis de Postgrado, Universidad Nacional Abierta y a Distancia]. https://repository.unad.edu.co/handle/10596/27805

Outsystems (2020). What is a mobile application? https://www.outsystems.com/glossary/what-is-mobile-application/

Peña, J., (2020). Sistema web basado en la gestión de mascotas y su geolocalización en caso de extravío en la Municipalidad Distrital de Puente Piedra

PHP (2020). ¿Qué es PHP? https://www.php.net/manual/es/intro-whatis.php

Santander (24 de Agosto de 2021). La economía colaborativa: ¿qué es y qué nos puede aportar?. https://www.santander.com/es/stories/la-economia-colaborativa-que-es-y-que-nos-puede-aportar

VTT (2004). Mobile -D.
http://virtual.vtt.fi/virtual/agile/mobiled.html