

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE UNA APLICACIÓN MÓVIL PARA SERVICIOS DE PASEADOR DE PERROS (DOG WALKER) EN LA ZONA URBANA DE QUITO

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS**

JEFFERSON OMAR DÍAZ CANDO

jefferson.diaz@epn.edu.ec

DIRECTOR: ING. IVONNE FERNANDA MALDONADO SOLIZ, MSC.

ivonne.maldonadof@epn.edu.ec

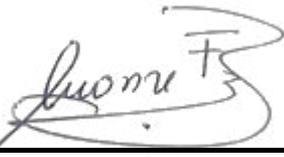
CODIRECTOR: PhD. Myriam Peñafiel

myriam.penafiel@epn.edu.ec

Quito, octubre 2021

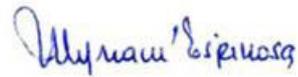
CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por el Sr. Díaz Cando Jefferson Omar como requerimiento parcial a la obtención del título de TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS, bajo nuestra supervisión:



Ing. Ivonne Maldonado, MSc

DIRECTORA DEL PROYECTO



PhD. Ing. Myriam Peñafiel

CODIRECTORA DEL PROYECTO

DECLARACIÓN

Yo, Díaz Cando Jefferson Omar con CI: 172371860-5 declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entrego toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

DEDICATORIA

El presente trabajo se lo dedico a mis familiares y amigos que me estuvieron acompañándome en mi carrera universitaria.

JEFFERSON OMAR DÍAZ CANDO

AGRADECIMIENTO

A mi mami, quien es la persona que jamás dejo de creer en mí, pese a nuestras limitaciones económicas me apoyó y no permitió que abandonara los estudios en ningún momento y en gran medida por forjarme con sus valores.

A mi hermana Marle, sus consejos me han servido en varias decisiones que he ido tomando y por haber estado en los peores momentos de mi vida.

A mis hermanos Erick, Cyndi, Ruby, han sido mi motivación de superación y permitirles enseñar que no existen imposibles.

A la Ing. Ivonne Maldonado por sus consejos a lo largo de mi carrera, sus conocimientos y experiencia lograron solventar varias dudas que iban surgiendo a lo largo del desarrollo del presente trabajo y por hacer posible la culminación de este proyecto.

Un especial agradecimiento al Ing. Juan Pablo Zaldumbide, por brindarme su amistad, su apoyo tanto dentro como fuera de mi carrera universitaria y por ser una de las personas que ha compartido varios de sus conocimientos mismos que han hecho posible la culminación de este proyecto.

JEFFERSON OMAR DÍAZ CANDO

ÍNDICE DE CONTENIDOS

1	Introducción	1
1.1	Objetivo general	2
1.2	Objetivos específicos	2
1.3	Alcance	2
2	Metodología	4
2.1	Metodología de Desarrollo	4
2.1.1	Roles	4
2.1.2	Artefactos	5
2.2	Diseño de interfaces	7
2.2.1	NinjaMock	7
2.3	Diseño de la arquitectura	7
2.3.1	Patrón arquitectónico	7
2.4	Herramientas de desarrollo	8
2.4.1	Herramientas de la Aplicación Móvil	8
2.4.2	Librerías de la Aplicación móvil	9
3	Resultados y Discusión	11
3.1	<i>Sprint</i> 0. Configuración del ambiente de desarrollo	11
3.1.1	Diseño y creación de la base de datos	11
3.1.2	Creación del proyecto en <i>Ionic</i>	11
3.1.3	Usuarios	12
3.2	<i>Sprint</i> 1. Módulos mascotas	12
3.2.1	Autenticación e Inicio de sesión	12
3.2.2	Registro y actualización de datos personales	13
3.2.3	Registro de mascotas	14
3.2.4	Actualización de mascotas	15
3.2.5	Visualización de mascotas	15
3.2.6	Eliminación de mascotas	16

3.3	<i>Sprint 2. Módulos paseo mascota</i>	17
3.3.1	Ingresar un valor para el pago y generar una solicitud.....	17
3.3.2	Ofertar un valor superior al valor inicial.....	17
3.3.3	Cancelar solicitud de paseo.....	18
3.4	<i>Sprint 3. Módulos estados de solicitudes (Dueño mascota / Dog Walker)</i>	19
3.4.1	Visualizar procesos de las solicitudes (<i>Dog Walker</i>).....	19
3.4.2	Actualizar estados (<i>Dog Walker</i>).....	19
3.4.3	Visualizar en tiempo real el cambio de estados que realice el <i>Dog Walker</i> (Dueño mascota)	20
3.4.4	Actualización estados en progreso del paseo (Dueño mascota).....	21
3.4.5	Visualización de la barra de progreso del paseo (Dueño mascota).....	22
3.5	<i>Sprint 4. Módulos Dog Walker (Dueño mascota / Dog Walker)</i>	22
3.5.1	Calificar al <i>Dog Walker</i> (dueño mascota).....	22
3.5.2	Recibir notificaciones de nuevas solicitudes (<i>Dog Walker</i>)	23
3.5.3	Recibir notificaciones ante el cambio de estados en progreso del paseo (dueño mascota).....	24
3.5.4	Visualizar valoración (<i>Dog Walker</i>).....	24
3.6	<i>Sprint 5. Pruebas y despliegue</i>	25
3.6.1	Pruebas unitarias.....	25
3.6.2	Pruebas de rendimiento.....	28
3.6.3	Pruebas de aceptación	32
3.6.4	Pruebas de compatibilidad.....	33
3.6.5	Despliegue de la Aplicación Móvil a producción	34
4	Conclusiones y Recomendaciones	35
4.1	Conclusiones	35
4.2	Recomendaciones.....	36
5	Referencias Bibliográficas	37
6	ANEXOS.....	i
6.1	Manual Técnico	i

6.2	Manual de Usuario.....	i
6.3	Manual de Instalación.....	i

ÍNDICE DE FIGURAS

Fig. 1: Diseño página solicitud – dueño mascota.....	7
Fig. 2: Patrón Arquitectónico - Aplicación Móvil.....	8
Fig. 3: Estructura de directorios para la Aplicación Móvil.....	11
Fig. 4: usuarios que interviene en la Aplicación Móvil.....	12
Fig. 5: Login	13
Fig. 6: Elección de correo.....	13
Fig. 7: Formulario "Mi perfil"	14
Fig. 8: Todos los campos son obligatorios.....	14
Fig. 9: Cédula no válida.....	14
Fig. 10: Formulario Registro de Mascotas	15
Fig. 11: Modificación de mascotas	15
Fig. 12: Listado de mascotas.....	16
Fig. 13: Advertencia para eliminar mascota.....	16
Fig. 14: Formulario de Solicitud	17
Fig. 15: Calculo de valor mínimo a pagar	17
Fig. 16: Varios Valores en los botones	18
Fig. 17: para ofertar un valor superior al valor inicial	18
Fig. 18: Mensaje confirmación para cancelar la solicitud.....	18
Fig. 19: Lista de procesos	19
Fig. 20: Pantalla con botones por defecto.....	20
Fig. 21: Pantalla con botones al cambio de estados.....	20
Fig. 22: Estado por defecto del progreso del paseo.....	21
Fig. 23: Cambio de estados y opción de botones	21
Fig. 24: Opción de botón "Finalizar el paseo"	21
Fig. 25: Calificar al <i>Dog Walker</i> cuando no ha llegado	23
Fig. 26: Calificar al <i>Dog Walker</i> cuando no ha llegado botones interactivos	23
Fig. 27: Notificaciones de Nueva Solicitud.....	23
Fig. 28: Notificación con estado "Estoy fuera"	24
Fig. 29: Notificación con estado "Paseando"	24
Fig. 30: Pantalla Valoración.....	24
Fig. 31: Módulo Perfil Usuario	25
Fig. 32: Generación de métodos para validad información.....	26
Fig. 33: Datos simulados.....	27
Fig. 34: Resultados de la prueba en el navegador.....	27

Fig. 35: Resultados de la prueba en la consola	28
Fig. 36: Creación del contenido del archivo lighthousec.js	28
Fig. 37: Ejecución y resultados de las pruebas de rendimiento	29
Fig. 38: Resultado de la página registro mascota en el navegador.....	29
Fig. 39: Resultado de la página progreso-duenio en el navegador	30
Fig. 40: Resultado de la página perfil-mascota en el navegador	30
Fig. 41: Resultado de la página home-paseador	31
Fig. 42: Resultado de la página procesos-dw	31
Fig. 43: Resultado de la página login	32
Fig. 44: Despliegue de la Aplicación Móvil a producción	34

ÍNDICE DE TABLAS

TABLA I: Asignación de roles	5
TABLA II: Historia de usuario 1 – Registrar y actualizar datos personales	6
TABLA III: Herramientas para el desarrollo de la Aplicación Móvil	8
TABLA IV: Librerías usadas en el desarrollo de la Aplicación Móvil.....	9
TABLA V: Prueba de aceptación 1 - Registrar y actualizar datos personales	32
TABLA VI: Pruebas de compatibilidad sobre diversos dispositivos móviles	33

RESUMEN

El presente proyecto surge como una idea a partir de la emergencia sanitaria que atraviesa el país, en un caso muy particular la población ecuatoriana ha tenido que acogerse a medidas de confinamiento impuestas por las administraciones gubernamentales en cada ciudad, siendo las más fuertes el caso de Quito y Guayaquil. Durante el periodo de confinamiento la población se mantuvo aislada provocando estrés por la imposibilidad de realizar actividades como: salir a caminar, ir a trabajar de forma presencial u otras que normalmente se hacían.

Teniendo en consideración el aislamiento y estrés que el confinamiento produjo en la población, se recaba información acerca de cómo repercute el encierro en otros miembros del hogar, las mascotas, obteniendo como resultados que los perros como los humanos necesitan salir de su entorno (hogar) para reducir el estrés y ansiedad en ambos casos.

En este sentido se ha desarrollado una Aplicación Móvil que permite encontrar paseadores de perros en la ciudad de Quito, así como el recibir solicitudes para brindar el servicio de paseador de perros, mejorando la salud de las mascotas.

Este proyecto está desarrollado con el uso de diferentes tecnologías, destacando el *framework Ionic* como capa de presentación dirigida al usuario final y *firebase* como sistema gestor de Base de Datos (NoSQL). Además, se hace uso de la metodología ágil *Scrum*, misma que aporta beneficios como: recopilación de requerimientos, flexibilidad, roles bien definidos y reuniones con el usuario, permitiendo que el desarrollo se concluya en los tiempos establecidos; presentando los resultados en pequeñas tareas. Finalmente, en la última sección de este informe, se mencionan las conclusiones y recomendaciones que se han obtenido a lo largo del desarrollo del presente proyecto.

PALABRAS CLAVE: *Scrum, Ionic, Framework, NoSQL, firebase, confinamiento.*

ABSTRACT

This project arises as an idea of the health emergency that the country is going through, in a very particular case the Ecuadorian population has had to take advantage of the confinement measures imposed by the government administrations in each city, the strongest being the case of Quito and Guayaquil. During the period of confinement, the population remained isolated, causing stress due to the impossibility of carrying out activities such as: walking, going to work in person or others that were carried out normally.

Considering the isolation and stress that confinement produces in the population, information is collected on how confinement affects other members of the household, pets, obtaining as results that dog like humans need to leave their environment (home) to reduce stress and anxiety in both cases.

In this sense, a Mobile Application has been developed that allows finding dog walkers in the city of Quito, as well as receiving requests to provide the dog walker service, improving the health of pets.

This project is developed with the use of different technologies, highlighting the Ionic framework as a presentation layer aimed at the end user and firebase as a Database management system (NoSQL). In addition, the agile Scrum methodology is used, which provides benefits such as: compilation of requirements, flexibility, well-defined roles and meetings with the user, allowing to conclude the development in the established times; presenting the results in small tasks. Finally, in the last section of this report the conclusions and recommendations that have been obtained throughout the development of this project are mentioned.

KEYWORDS: *Scrum, Ionic, Framework, NoSQL, firebase.*

1 INTRODUCCIÓN

Los perros han dejado de ser únicamente animales que sirven para la guardianía, actualmente son considerados como un miembro más de la familia, por ende, requieren cuidado y cariño. Según un estudio realizado por la Secretaría de Salud del Distrito Metropolitano de Quito, 3 de cada 5 familias tiene al menos una mascota en su casa [1].

Las personas que deciden tener perros consideran que únicamente es necesario un amplio lugar donde los canes puedan caminar libremente, sin embargo, es una realidad no disponible para muchos, porque en la actualidad las personas prefieren los departamentos ya sea por su costo, vista privilegiada o ubicación, ante este escenario, los canes tienen reacciones inadecuadas a diferentes estímulos sociales [2], toman los espacios como rutinarios y no identifican nuevas sensaciones de olores o sonidos.

Para los perros el encierro es perjudicial, porque conforme pasa el tiempo aumenta su estrés, de igual forma, al no salir a pasear almacenan gran cantidad de energía, un claro ejemplo es la hiperactividad y los destrozos que ocasionan dentro del hogar [2] [3]. Cuando un perro puede salir a pasear mejora su salud tanto física como mental, debido a que libera la energía acumulada, socializa con otros perros y con otras personas [4], reduciendo el estrés y aprendiendo a no alterarse con facilidad ante situaciones que son comunes para los seres humanos como: sonidos fuertes de autos, bocinas y/o el acercamiento de personas simplemente a saludar [5], logrando que el perro este más tranquilo con seguridad y confianza en el entorno.

La búsqueda de paseadores de perros en la ciudad de Quito es una tarea compleja, por un lado, para encontrar este servicio mediante la web se aprecian varias publicaciones en plataformas como "Mercado Libre", donde los paseadores se encargan de ingresar información sobre el trabajo que realizan y en un horario determinado, en algunos casos no se adhiere a la disponibilidad de tiempo del usuario. Existen también, publicaciones en la web que mantienen el estado activo hasta la fecha de caducidad de la oferta, por último, existe una página llamada "paseaperros.com", pero al momento no se encuentra disponible para Ecuador. De ahí el hecho de que quienes desean este servicio deban hacer búsquedas demasiado tediosas en redes sociales.

Por lo ya descrito con anterioridad, se ha desarrollado de una Aplicación Móvil con el objetivo de que aquellas personas que se dedican al cuidado y atención de perros oferten dicho servicio, eliminando así la contaminación visual de postes y paredes con anuncios de este tipo. Por otro lado, las personas que deseen hacer uso del servicio

tienen la posibilidad de acceder de manera más sencilla a esta información y hacer el contacto de forma directa por medio de la Aplicación Móvil donde se encuentre centralizada la información y sin la necesidad de la búsqueda en diferentes sitios web.

1.1 Objetivo general

Desarrollar una Aplicación Móvil para servicios de paseador de perros (*Dog Walker*) en la zona urbana de Quito.

1.2 Objetivos específicos

- Determinar los requerimientos de la Aplicación Móvil.
- Diseñar el prototipo de interfaz, modelo de base de datos y la arquitectura de la Aplicación Móvil.
- Implementar los diferentes módulos de la Aplicación Móvil.
- Verificar el funcionamiento de la Aplicación Móvil.

1.3 Alcance

La conducta inadecuada como agresividad o temor a otros individuos, el estrés emocional producidos por la falta de salidas rutinarias, falta de ejercicio, aumento de peso, entre otros [2] [5], son algunas de las razones por las cuales los dueños de mascotas requieren buscar paseadores de perros. En los dos últimos años en la ciudad de Quito, a causa de la emergencia sanitaria y la pérdida de empleo se ha dificultado el acceso a escuelas de entrenamiento canino que ofrecen un servicio no solo de adiestramiento sino de paseo canino [6] [7].

Al mismo tiempo los dispositivos móviles junto a las aplicaciones móviles se han convertido en una herramienta fundamental para la realización de diversas actividades [8]. De acuerdo a un estudio de GuiaLocal.com, el uso de dispositivos móviles creció el 61% en países de América Latina, específicamente en Ecuador, un crecimiento anual del 75% en el uso de estos dispositivos [9].

Por esta razón, se ha desarrollado *Dog Walker* una Aplicación Móvil que beneficia tanto a las personas que quieren contratar el servicio de paseador de perros como a aquellas que brindan este servicio dentro de la ciudad de Quito. Funcionando como una alternativa de búsqueda de dicho servicio en un solo lugar sin la necesidad de acudir a los diferentes afiches existentes o a la web en general, encontrando la mejor opción para ayudar a su mascota y evitando que se estrese, ataque, sufra depresión, entre otros, con el simple hecho de salir a pasear.

La Aplicación Móvil está orientada a dispositivos *Android*, garantiza la consistencia, seguridad e integridad de toda la información, a través de un sistema de autenticación y de varios usuarios:

El usuario con perfil dueño de mascota en la Aplicación Móvil permite:

- Registro y autenticación mediante el correo de *Gmail* de *Google*.
- Actualización del perfil.
- Registro, actualización y eliminación de mascotas.
- Solicitar un *Dog Walker*.
- Fijar el tiempo y valor a cancelar por el servicio.
- Recibir ofertas de paseo y visualizar aspectos relevantes del *Dog Walker*.
- Calificar el servicio (*Dog Walker*).

El usuario con perfil *Dog Walker* de mascota en la Aplicación Móvil permite:

- Registro y autenticación mediante el correo de *Gmail* de *Google*.
- Modificación del perfil.
- Activar o desactivar recepción de solicitudes.
- Vista previa de una solicitud.
- Gestionar el progreso del paseo.
- Visualizar su valoración.

2 METODOLOGÍA

Las metodologías ágiles permiten que se adapte la forma de trabajar a las necesidades del proyecto, el emplearlas en el desarrollo consigue flexibilidad y trabajo en equipo [10], favoreciendo el enfoque sencillo en la documentación y aceptando cambios que puedan surgir en las diferentes etapas del ciclo de vida, en lugar de oponerse a ellos [11].

La metodología ágil *Scrum* agrupa un conjunto de actividades, que divide el proyecto en pequeños bloques o *Sprints*, con el objetivo de revisar y mejorar la fase anterior [12], por ello se considera la más apta para el desarrollo de este proyecto, permitiendo alcanzar los objetivos en los tiempos planteados.

2.1 Metodología de Desarrollo

Al emplear *Scrum* se realizan entregas parciales y regulares del producto final priorizadas por parte del dueño del proyecto. Para ello, *Scrum* está especialmente diseñada para proyectos en entornos complejos, donde se necesita obtener resultados inmediatos [13], de allí que su uso en este proyecto ha facilitado la planificación, comunicación constante con el *Product Owner*, recopilación de información, creación de listado de requerimientos, diseños, codificación, pruebas y despliegue para el desarrollo exitoso del mismo [12].

2.1.1 Roles

Scrum está basado en roles, *Product Owner*, *Scrum Master* y *Development Team*, donde la participación de cada uno es inherente a esta metodología, ya que el éxito del proyecto depende de una buena planificación de actividades de cada integrante; una comunicación clara con el *Product Owner* permite segmentar el proyecto en varios *Sprints*, ayudando a que el *Development Team* entregue un producto acorde a las necesidades [14]

Product Owner: Es responsable de garantizar una comunicación clara sobre el producto, definir las necesidades del negocio y establecer los criterios de aceptación para el producto [15]. El rol lo ejecuta un paseador de perros, quien ha proporcionado la información necesaria por medio de una serie de reuniones, en las cuales se ha logrado identificar de manera fortuita los requerimientos y herramientas para el desarrollo de la Aplicación Móvil propuesta.

Scrum Master: Actúa como líder servicial, ayuda a emplear de mejor manera la metodología *Scrum*, se enfoca en la parte del negocio, transmite la visión del proyecto

al resto del equipo [16]. El rol lo ejecuta la directora del proyecto, quien posee los conocimientos técnicos necesarios para ayudar al equipo de desarrollo a resolver dudas e inconvenientes y alcanzar de manera satisfactoria el cumplimiento del objetivo planteado.

Development Team: Es el grupo de profesionales con conocimientos y habilidades técnicas necesarias para trabajar en las historias de usuarios pendientes de cada *Sprint*, generando los entregables del proyecto [16]. El rol lo ejecuta el desarrollador del proyecto, quien cumple con las habilidades y conocimientos para ejecutar cada una de las fases de la metodología y cumplir de manera adecuada cada *Sprint*.

El equipo Scrum se encuentra conformado como se presenta en la **TABLA I**.

TABLA I: Asignación de roles

NOMBRE	ROL
Sr. Andrés Inga	<i>Product Owner</i>
Ing. Ivonne Maldonado, MSc.	<i>Scrum Master</i>
Sr. Omar Díaz	<i>Development Team</i>

2.1.2 Artefactos

Son una serie de elementos que están diseñados para garantizar la transparencia de la información, evitar tener una mala comunicación y un mal manejo de la información en la entrega del proyecto [17].

Recopilación de Requerimientos

Permite realizar la identificación y documentación de los requerimientos de la Aplicación Móvil, a partir de una comunicación efectiva con los interesados [18]. Para el cumplimiento de esta etapa se ha establecido una serie de reuniones con el *Product Owner*, obteniendo un listado inicial de manera general de los roles y funcionalidades de la Aplicación Móvil, esta lista puede ser observada en la sección “Recopilación de Requerimientos del Manual Técnico” (Pág. 1 – 4).

Historias de Usuario

Se escriben en fichas con un formato establecido, constan de descripciones cortas y simples de una característica contada desde la perspectiva de un usuario final [19]. La **TABLA II** muestra los elementos y la estructura que conforma una Historia de Usuario, de esta manera se logra clasificar adecuadamente los requerimientos del proyecto antes

de empezar con la etapa de codificación. Las demás historias de usuario pueden ser observadas en la sección Historias de usuario (Pág. 4 - 16) del Manual Técnico.

TABLA II: Historia de usuario 1 – Registrar y actualizar datos personales

HISTORIA DE USUARIO	
Identificador (ID): HU001	Usuario: Dueño mascota/ <i>Dog Walker</i>
Nombre Historia: Registrar y actualizar datos personales	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Omar Díaz	
Descripción: Los usuarios dueño mascota y <i>Dog Walker</i> están en la posibilidad de registrar y actualizar los datos solicitados en el formulario perfil.	
Observación: Si el usuario dueño mascota o <i>Dog Walker</i> desean hacer uso de la Aplicación Móvil deben tener la información del perfil de usuario completo el cual consta de: foto de perfil, nombre, apellido, edad, número de cédula (nacionalidad ecuatoriana), número de celular, ubicación domiciliaria y referencia del domicilio. Los datos solicitados en el perfil del dueño mascota o <i>Dog Walker</i> , son de suma importancia para garantizar un servicio de calidad y brindar confianza entre los usuarios. Los datos personales pueden ser actualizados cuando los usuarios lo requieran.	

Product Backlog

Es una lista que contienen todos los requerimientos a implementar en el producto, se encuentra ordenado dando prioridad a los temas que brindan más valor en el momento de especificar los requerimientos [16]. El *Product Backlog* puede ser observado en la respectiva sección (Pág. 16 - 18) del Manual Técnico.

Sprint Backlog

Es una lista de tareas que se emplea para llevar a cabo las historias de usuario de cada *Sprint* [20]. Conforme a la recopilación de requerimientos, historias de usuario y *Product Backlog* se ha establecido 5 *Sprints* de desarrollo, mismos que pueden ser observados en la respectiva sección (Pág. 19 - 26) del Manual Técnico.

2.2 Diseño de interfaces

Una vez determinadas las necesidades para este proyecto y establecido los *Sprints*, a continuación, se define la herramienta que se ha empleado en el diseño de la Aplicación Móvil.

2.2.1 NinjaMock

El diseño de las interfaces para la Aplicación Móvil está determinado por prototipos (*mockups*), mismos que se han realizado con la ayuda de la herramienta NinjaMock, misma que permite diseñar prototipos mediante el uso de componentes simples [21], en este caso se ha hecho uso de los componentes de diseño orientados a las aplicaciones móviles.

En la **Fig.1** se observa un ejemplo del prototipo para la pantalla Solicitud. El resto de los prototipos pueden ser observados en la sección “Prototipo de Interfaces” (Pág. 27 - 31) del Manual Técnico.



Fig. 1: Diseño página solicitud – dueño mascota

2.3 Diseño de la arquitectura

Una vez determinados los *Sprints* para este proyecto en conjunto con el diseño de las interfaces, a continuación, se establece el patrón arquitectónico implementado en el desarrollo de la Aplicación Móvil propuesta.

2.3.1 Patrón arquitectónico

La arquitectura que posee la Aplicación Móvil está basada en el patrón arquitectónico Modelo Vista Controlador (MVC), mismo que mantiene distintas capas, las cuales se

encargan de realizar una tarea específica [22], permitiendo mantener una estructura organizada [23].

En la **Fig. 2** se observa las herramientas y el patrón arquitectónico que se ha empleado en el desarrollo, permitiendo contar con una Aplicación Móvil escalable a medida que lo requiera.

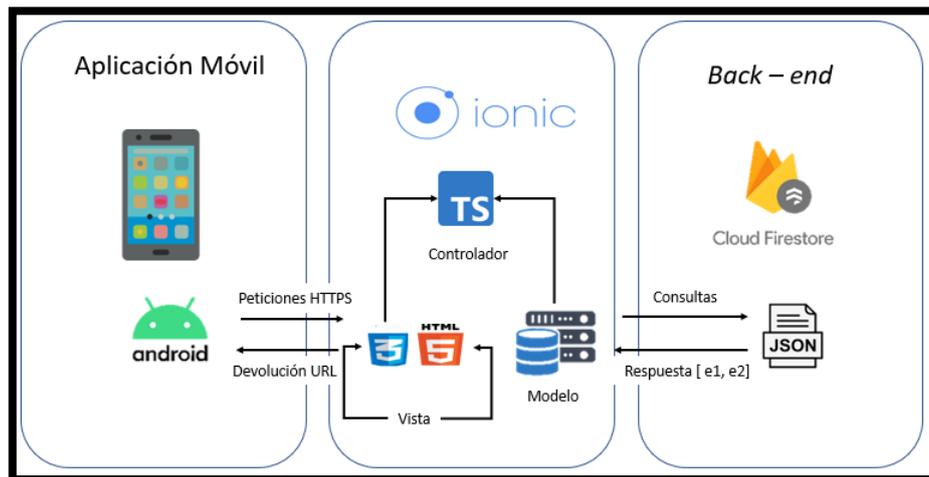


Fig. 2: Patrón Arquitectónico - Aplicación Móvil

2.4 Herramientas de desarrollo

Establecer las herramientas que se adapten a las necesidades del proyecto y optimicen el rendimiento del trabajo de desarrollo [24].

Una vez determinados los *Sprints* para este proyecto en conjunto con el diseño de las interfaces y el patrón arquitectónico. A continuación, se describe cada una de las herramientas y librerías que se han utilizado en el presente proyecto.

2.4.1 Herramientas de la Aplicación Móvil

La **TABLA III** muestra las herramientas de desarrollo que han sido utilizadas para el desarrollo de la Aplicación Móvil.

TABLA III: Herramientas para el desarrollo de la Aplicación Móvil

HERRAMIENTA	JUSTIFICACIÓN
Firestore	La implementación de esta plataforma permite desarrollar el proyecto con grandes ventajas como la selección de herramientas de acuerdo con el

	interés del programador, así como la gratuidad, el almacenamiento y la autenticación [25].
Cloud Firestore	El uso de esta herramienta en el desarrollo facilita la gestión de la información, así como las actualizaciones en tiempo real empleando un modelo de datos intuitivo [26].
Ionic	La implementación de este <i>framework</i> tiene como ventaja la compatibilidad con distintas plataformas, así como que la creación de la Aplicación Móvil sea híbrida y que este basadas en <i>HTML</i> , <i>CSS</i> y <i>JavaScript</i> [27].
Bootstrap	La aplicación de esta herramienta visual tiene el propósito de brindar una experiencia agradable al usuario, es decir, la interactividad adaptada por medio de distintos componentes [28].
CSS	Poner en funcionamiento dicha herramienta, facilita la integración de contenidos personalizados, beneficia al programador, porque puede cambiar los estilos, por ejemplo, colores, tipo de letras, sin que los contenidos se vean alterados [29].

2.4.2 Librerías de la Aplicación móvil

La **TABLA IV** muestra un conjunto de librerías utilizadas para complementar el desarrollo de la Aplicación Móvil del presente proyecto.

TABLA IV: Librerías usadas en el desarrollo de la Aplicación Móvil

LIBRERÍA	DESCRIPCIÓN
“@angular/cli@11.2.11”	Librería generar, depurar y publicar aplicaciones [30].
“@angular/fire@6.1.4”	Librería que permite hacer uso de los servicios que ofrece <i>firebase</i> [31].
“@capacitor/cli@2.4.7”	Es un <i>framework</i> el cual permite transformar un sistema web en una aplicación móvil [32].
“@codetrix-studio/capacitor-google-auth@2.1.3”	Librería que permite crear un inicio de sesión mediante <i>Google</i> [33].

3 RESULTADOS Y DISCUSIÓN

A continuación, se detalla la realización de las tareas y los resultados conseguidos en cada *Sprint* de desarrollo. Además, se describen las diversas pruebas realizadas a la Aplicación Móvil y su despliegue.

3.1 *Sprint* 0. Configuración del ambiente de desarrollo

Conforme a la planificación, los resultados del *Sprint* 0 son:

- Diseño y creación de la base de datos.
- Creación del proyecto en *Ionic*.
- Usuarios.

3.1.1 Diseño y creación de la base de datos

La base de datos NoSQL del proyecto se encuentra gestionada mediante *Cloud Firestore* el cual es uno de los servicios que proporciona la plataforma *Firebase*, en esta se almacena toda la información relacionada a las solicitudes de paseos, registros, información de los usuarios en las interacciones realizadas, entre otros. Las colecciones empleadas y definidas para la Aplicación Móvil pueden ser observadas en la sección “Diseño de la Base de Datos” (Pág. 32) del Manual Técnico.

3.1.2 Creación del proyecto en *Ionic*

Para crear el entorno de desarrollo se ha instalado las herramientas *NodeJS*, *Ionic* y el editor de código *Visual Studio Code*, lo que ha permitido crear directorios bajo el patrón arquitectónico MVC. La **Fig. 3** muestra la estructura del proyecto.

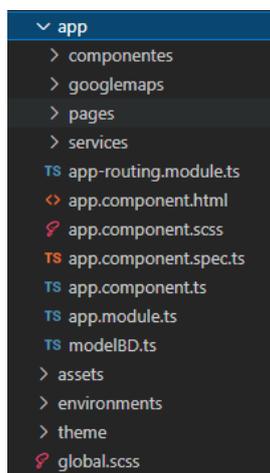


Fig. 3: Estructura de directorios para la Aplicación Móvil

3.1.3 Usuarios

En la **Fig. 4** se ilustran los usuarios que pueden interactuar con la Aplicación Móvil, cada uno según el rol que desempeñan con los permisos otorgados a los diferentes módulos asignados una vez realizada la respectiva autenticación.

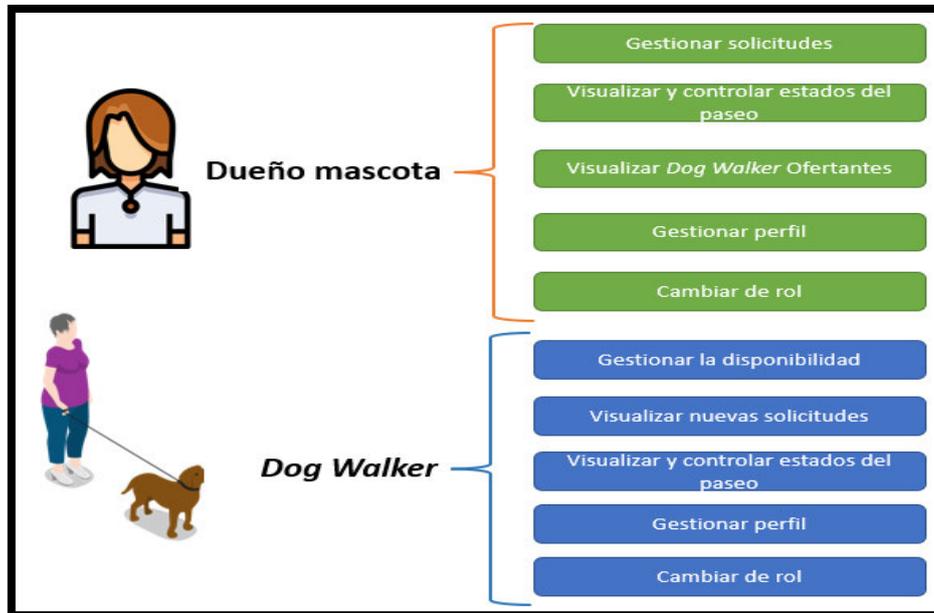


Fig. 4: usuarios que interviene en la Aplicación Móvil

3.2 *Sprint 1*. Módulos mascotas

Conforme a la planificación, los resultados del *Sprint 1* son:

- Autenticación e Inicio de sesión.
- Registro y actualización de datos personales.
- Registro de mascotas.
- Actualización de mascotas.
- Visualización de mascotas.
- Eliminación de mascotas.

3.2.1 Autenticación e Inicio de sesión.

La **Fig. 5** muestra la pantalla de *Login* de la Aplicación Móvil, cualquier usuario que desee hacer uso de esta obligatoriamente debe proporcionar su correo de *Gmail*. Si el usuario no cuenta con correo electrónico en *Gmail* la Aplicación Móvil le permite agregar una cuenta de *Gmail* haciendo uso de la librería de autenticación con *Google* como se muestra en la **Fig. 6**. La pantalla de *Login* **Fig. 5** sirve para realizar el inicio de sesión, pero, a diferencia de la autenticación ya no solicita un permiso adicional para hacer uso

de la cuenta de *Gmail*, sin embargo, muestra que se inicie sesión con un correo *Gmail* previamente registrado en el dispositivo móvil. Cabe aclarar que la Aplicación Móvil reconoce con el último perfil de usuario que se cerró la sesión.

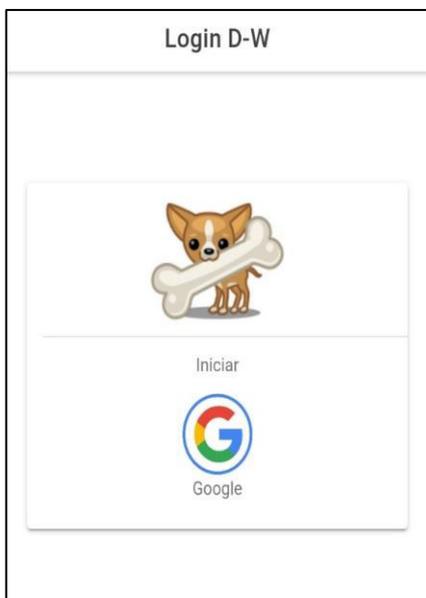


Fig. 5: Login

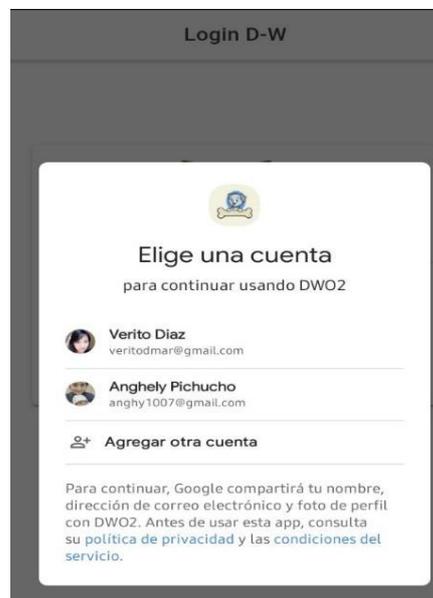


Fig. 6: Elección de correo

3.2.2 Registro y actualización de datos personales

La **Fig. 7** muestra el formulario de registro y actualización de datos personales para el usuario que previamente ha iniciado sesión. El formulario solicita: foto, nombre, apellido, número de cédula de identidad (ecuatoriana), número de teléfono celular, edad, dirección domiciliaria y referencia de la dirección. Existiendo las validaciones necesarias para cada uno de los campos del formulario, si un campo se deja en blanco o es completado de manera incorrecta la Aplicación Móvil presenta un mensaje “Todos los campos son obligatorios” o “Cédula no válida” dependiendo el caso (**Fig. 8** y **Fig. 9**). En el Manual de Usuario se puede apreciar en detalle todo el procedimiento para el inicio de sesión. Cabe mencionar que esta funcionalidad es la misma para el usuario dueño mascota y el usuario *Dog Walker*.

Mi Perfil
 Foto de Perfil*
 ainga201828@gmail.com
 Nombre*
 Apellido*

Fig. 7: Formulario "Mi perfil"

Apellido*
 Díaz
 Cédula de Identidad*
 Celular*
 0961838029
 Edad*
 26
 Ubicación*
 Paccha 152
 Referencia*
 Dos Pisos
 Todos los datos son obligatorios (*)

Fig. 8: Todos los campos son obligatorios

Cédula de Identidad*
 1227373780
 Celular*
 0961838029
 Edad*
 26
 Ubicación*
 Paccha 152
 Referencia*
 Dos Pisos
 Cédula no Valida!!

Fig. 9: Cédula no válida

3.2.3 Registro de mascotas

La **Fig. 10** muestra el formulario para registrar mascotas. El formulario solicita de forma obligatoria: foto, nombre, edad, sexo, tamaño y agresividad, esto debido a que son aspectos indispensables para el interesado en aceptar la solicitud. En el Manual de Usuario se puede apreciar de forma detallada el proceso para registrar una mascota.

Fig. 10: Formulario Registro de Mascotas

3.2.4 Actualización de mascotas

Una vez que se ha registrado una mascota, se puede modificar los datos de esta, para realizar la acción se hace uso del formulario de registro mascotas (**Fig. 11**). De igual forma cuando se recurre a la actualización de datos los mismos son validados antes de guardar la información de la mascota. En el Manual de Usuario se puede apreciar de forma detallada el proceso para actualizar la información de las mascotas.

Fig. 11: Modificación de mascotas

3.2.5 Visualización de mascotas

Cuando ya se ha realizado el registro de una o varias mascotas, estas son presentadas en un listado como muestra la **Fig. 12**. De esta forma el usuario dueño mascota visualiza

cada una de sus mascotas registradas en donde se presenta foto y nombre de la mascota.

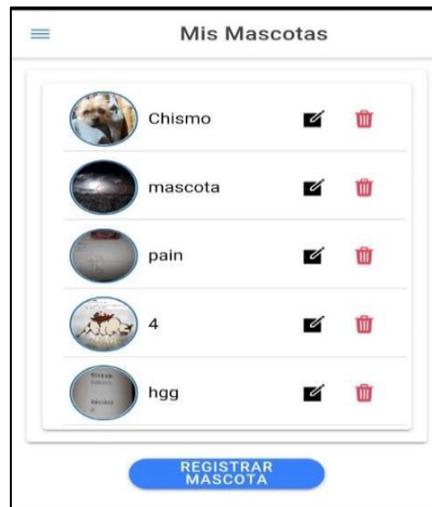


Fig. 12: Listado de mascotas

3.2.6 Eliminación de mascotas

La Fig. 12 muestra el listado de mascotas que se tiene registrado, junto a cada mascota existen dos íconos uno en forma de lápiz, el cual hace alusión a la edición de mascotas y el otro ícono en forma de un basurero, el cual hace mención a la eliminación de mascotas. Cuando el usuario dueño mascota pulsa sobre el ícono en forma de basurero se muestra un mensaje de advertencia, el cual solicita la confirmación como se muestra en la Fig. 13 para posteriormente eliminar o no eliminar la mascota. En el Manual de Usuario se puede apreciar de forma detallada el proceso para la eliminación de mascotas.

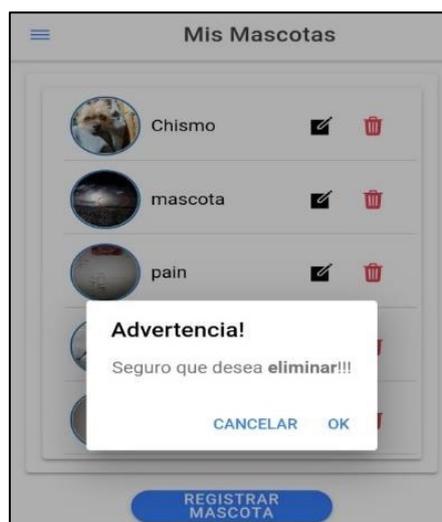


Fig. 13: Advertencia para eliminar mascota

3.3 *Sprint 2*. Módulos paseo mascota

Conforme a la planificación, los resultados del *Sprint 2* son:

- Ingresar un valor para el pago y generar una solicitud.
- Ofertar un valor superior al valor inicial.
- Cancelar solicitud de paseo.

3.3.1 Ingresar un valor para el pago y generar una solicitud

Una vez que el usuario dueño mascota ha realizado el registro de sus mascotas y se dirija a la sección “Quiero un paseo” puede realizar el procedimiento para generar una solicitud, donde debe completar un formulario **Fig. 14** donde se solicita: seleccionar duración del paseo , valor mínimo y una sección opcional de recomendaciones para el paseo. Cabe mencionar, que el formulario verifica si la información ingresada es consistente. Además, el valor mínimo a pagar se calcula de manera interna **Fig. 15**. En el Manual de Usuario se puede apreciar de forma detallada el proceso para generar una solicitud.

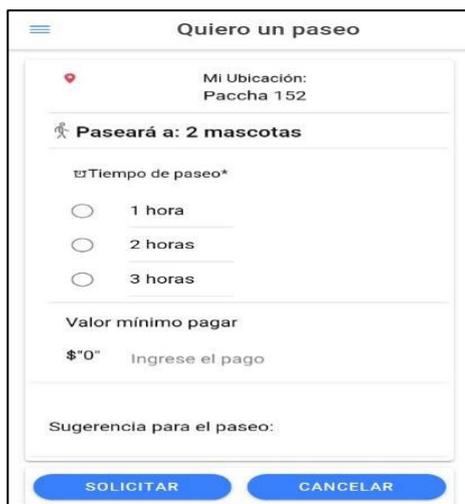


Fig. 14: Formulario de Solicitud

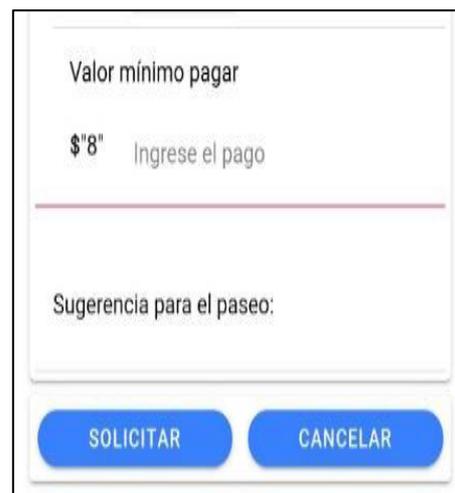


Fig. 15: Calculo de valor mínimo a pagar

3.3.2 Ofertar un valor superior al valor inicial

Una vez que se ha generado la solicitud automáticamente se redirige a la sección “Solicitud”, en la sección “Solicitud” el usuario dueño mascota visualiza una serie de botones con diferentes valores como se muestra en la **Fig. 16**, al pulsar sobre uno de estos botones la Aplicación Móvil muestra un mensaje de confirmación como se muestra en la **Fig. 17** para ofertar un valor superior al valor inicial. En el Manual de Usuario se

puede apreciar de forma detallada el proceso para ofertar un valor superior al valor inicial.



Fig. 16: Varios Valores en los botones

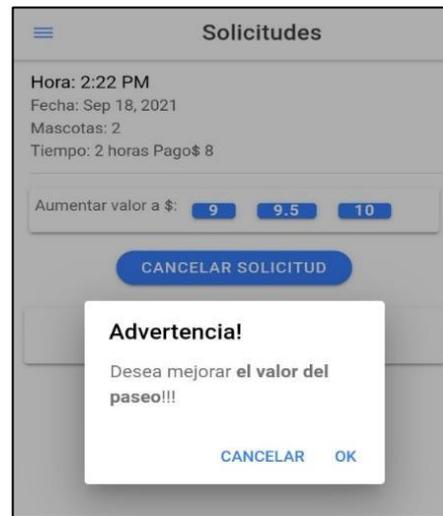


Fig. 17: para ofertar un valor superior al valor inicial

3.3.3 Cancelar solicitud de paseo.

El usuario dueño mascota puede sentir la necesidad de cancelar la solicitud previamente creada por diversos motivos. Para cancelar la solicitud, se redirige a la sección "Solicitud", una vez en la sección "Solicitud" pulsa el botón cancelar solicitud. De forma simultánea se presenta un mensaje como se muestra en la Fig. 18 que solicita la confirmación para cancelar la solicitud. Cabe mencionar que, la solicitud no se elimina del registro de la Base de Datos, simplemente cambia de estado para que deje de ser visible para el usuario.



Fig. 18: Mensaje confirmación para cancelar la solicitud

3.4 *Sprint 3. Módulos estados de solicitudes (Dueño mascota / Dog Walker)*

Conforme a la planificación, los resultados del *Sprint 3* son:

- Visualizar procesos de las solicitudes (*Dog Walker*).
- Actualizar estados (*Dog Walker*).
- Visualizar cambio de estados en tiempo real realizado por el usuario *Dog Walker*.
- Actualizar estados en progreso del paseo (Dueño mascota).
- Visualizar barra de progreso del paseo (Dueño mascota).

3.4.1 Visualizar procesos de las solicitudes (*Dog Walker*)

Una vez que el usuario *Dog Walker* ha sido aceptado para que brinde un paseo, este puede visualizar los procesos que está llevando a cabo. En la **Fig. 19** se muestra la pantalla de los procesos que está ejecutando, en donde se destaca la foto y el nombre de la persona que ha solicitado el servicio, tiempo de paseo, valor a recibir por el paseo y el estado en el que se encuentra el proceso. En el Manual de Usuario se puede apreciar de forma detallada los procesos que se están llevando a cabo.



Fig. 19: Lista de procesos

3.4.2 Actualizar estados (*Dog Walker*)

Una vez que el usuario *Dog Walker* ha sido aceptado para que brinde un paseo y haya seleccionado un proceso en la sección "Procesos" puede visualizar una pantalla con la información necesaria de la persona a quien debe brindar el servicio, además de varios botones como se muestra en la **Fig. 20** con diversas funciones como: cómo llegar, ver mascotas, ocultar mascotas, llego en 5 minutos, llego en 10 minutos, estoy fuera, paseando, paseo finalizado y ya no estoy activo.

La **Fig. 21** muestra una pantalla que en la cual existen botones adicionales en los cuales el usuario *Dog Walker* pulsa y de forma simultánea se procede a cambiar de estado.

Entre los estados se tiene: “ya no estoy disponible, llego en 10 minutos, llego en 5 minutos, estoy fuera, inicia paseo, paseo finalizado”. Cabe destacar que, los botones para los cambios de estado son dinámicos por lo que, a medida que interactúa los mismos son o no visibles. En el Manual de Usuario se puede apreciar de forma detallada el cambio de estados.



Fig. 20: Pantalla con botones por defecto



Fig. 21: Pantalla con botones al cambio de estados

3.4.3 Visualizar en tiempo real el cambio de estados que realice el *Dog Walker* (Dueño mascota)

Cuando el usuario dueño mascota ha aceptado que el usuario *Dog Walker* brinde el servicio, la Aplicación Móvil redirige a la sección “Progreso”, en esta sección se puede visualizar el cambio de estados que ha realizado el usuario *Dog Walker*. La Fig. 22 muestra el estado por defecto que observa el usuario dueño mascota. Como aspecto importante, los cambios de estados son visualizados a medida que el usuario *Dog Walker* pulse sobre los botones que generan el cambio de estados (Fig. 23).



Fig. 22: Estado por defecto del progreso del paseo

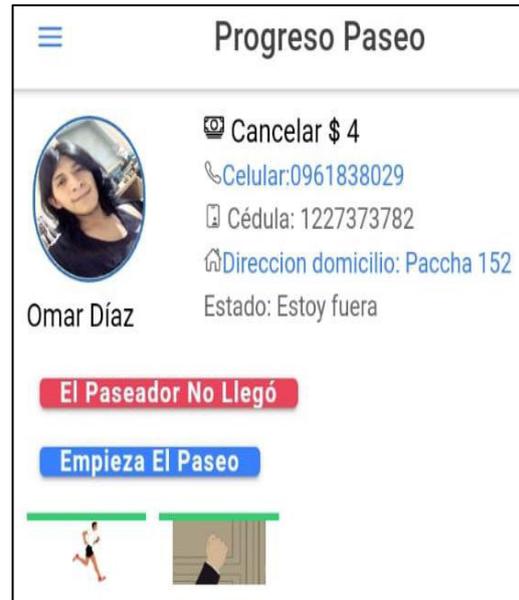


Fig. 23: Cambio de estados y opción de botones

3.4.4 Actualización estados en progreso del paseo (Dueño mascota)

Después que el usuario *Dog Walker* haya pulsado el “botón estoy fuera”, se activa de manera automática el botón “empieza el paseo”, los botones son dinámicos y estos reaccionan al cambio de estados que se vayan produciendo. Cuando físicamente empieza el paseo, el usuario *Dog Walker* puede olvidar pulsar el botón “empieza el paseo” o “paseo finalizado” que generan el cambio de estado, para ello, el usuario dueño mascota puede pulsar dichos botones si lo considera pertinente. La **Fig. 23** y **Fig. 24** muestran los botones que aparecen a medida que se producen los cambios de estado.

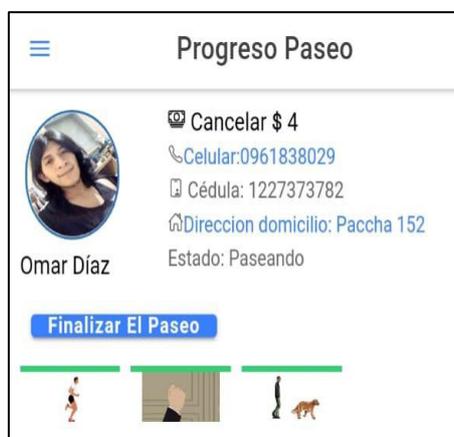


Fig. 24: Opción de botón "Finalizar el paseo"

Es importante mencionar que, el botón “el paseador no llegó” también genera un cambio de estado. Ante este cambio de estado se elimina automáticamente el proceso de paseo en la sección “Procesos” del usuario *Dog Walker*. En el Manual de Usuario se puede apreciar en forma detallada la actualización de estados en progreso.

3.4.5 Visualización de la barra de progreso del paseo (Dueño mascota)

La barra de progreso permite tener una mejor experiencia de usuario mientras usa la Aplicación Móvil. La **Fig. 23** y la **Fig. 24** muestran la barra de progreso, esta barra de interactúa con los siguientes cambios de estado: llego en 10 minutos, llego en 5 minutos, estoy fuera, empieza el paseo y paseo finalizado.

3.5 Sprint 4. Módulos *Dog Walker* (Dueño mascota / *Dog Walker*)

Conforme a la planificación, los resultados del *Sprint 4* son:

- Calificar al *Dog Walker* (dueño mascota)
- Recibir notificaciones de nuevas solicitudes (*Dog Walker*)
- Recibir notificaciones ante el cambio de estados en progreso del paseo (dueño mascota)
- Visualizar valoración (*Dog Walker*)

3.5.1 Calificar al *Dog Walker* (dueño mascota)

Una vez finalizado el proceso del paseo o si se pulsa sobre “el paseador no llegó”, en la sección “Progreso” se visualiza un cuadro para proceder a calificar al *Dog Walker*. Para ello se presenta figuras que hacen referencia a la calificación del 1 al 5, siendo 1 la calificación más baja y 5 las más alta. La **Fig. 25** y **Fig. 26** muestran el cuadro para calificar al *Dog Walker*. Cabe mencionar que, si el usuario dueño mascota pulsa en “Ahora No”, por defecto la calificación es de cinco. Además de que, sin realizar calificación, el usuario dueño mascota no puede generar una nueva solicitud por lo que, necesariamente debe calificar al *Dog Walker*.



Fig. 25: Calificar al *Dog Walker* cuando no ha llegado



Fig. 26: Calificar al *Dog Walker* cuando ha llegado botones interactivos

3.5.2 Recibir notificaciones de nuevas solicitudes (*Dog Walker*)

Una vez que el usuario *Dog Walker* se ha registrado de manera correcta y tras haber indicado la disponibilidad como activa se reciben notificaciones de las nuevas solicitudes que los usuarios dueño mascota generan. El objetivo de recibir notificaciones tras existir una nueva solicitud es alertar al usuario *Dog Walker* que en efecto existe una nueva solicitud, la **Fig. 27** muestra las solicitudes de nuevos paseos existentes. En el Manual de Usuario se puede apreciar en forma detallada el funcionamiento de las notificaciones ante el surgimiento de nuevas solicitudes.

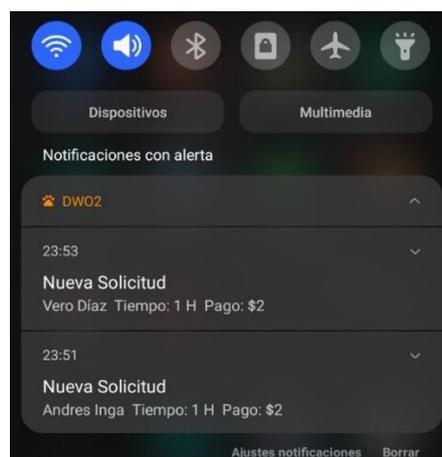


Fig. 27: Notificaciones de Nueva Solicitud

3.5.3 Recibir notificaciones ante el cambio de estados en progreso del paseo (dueño mascota)

La **Fig. 28** y **Fig. 29** muestran las solicitudes que recibe el usuario dueño mascota, esto se produce para ir alertando sobre los avances que tenga el usuario *Dog Walker* y de esta forma realizar actividades como: entregar a las mascotas para el paseo, recibir a las mascotas cuando el paseo haya finalizado u otro. En el Manual de Usuario se puede apreciar en forma detallada el funcionamiento de las notificaciones.

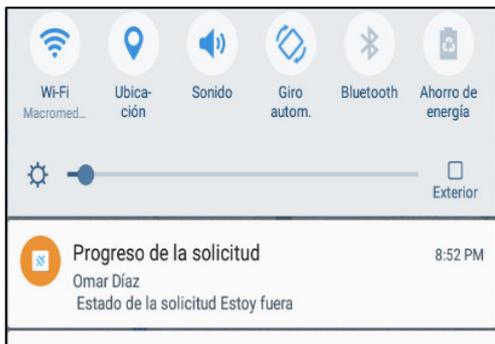


Fig. 28: Notificación con estado "Estoy fuera"

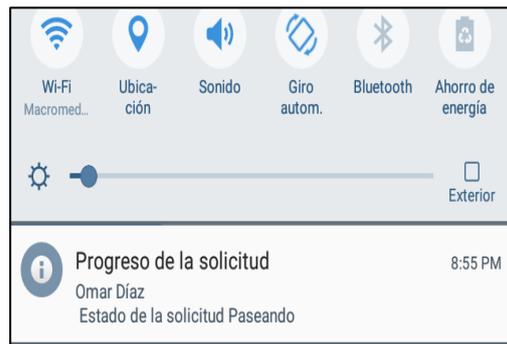


Fig. 29: Notificación con estado "Paseando"

3.5.4 Visualizar valoración (*Dog Walker*)

La **Fig. 30** muestra la pantalla de valoración del usuario *Dog Walker*, en esta únicamente se presenta la calificación que el usuario *Dog Walker* ha obtenido al brindar el servicio de paseador. La calificación es un promedio total del número de paseos que ha realizado a lo largo del tiempo. Además, la pantalla muestra gráficas y mensajes acorde a la calificación que tiene hasta ese momento, con el objetivo de motivar al paseador a mejorar su servicio.



Fig. 30: Pantalla Valoración

3.6 *Sprint* 5. Pruebas y despliegue

Conforme a la planificación, los resultados del *Sprint* 5 son:

- Pruebas unitarias.
- Pruebas de rendimiento.
- Pruebas de aceptación.
- Pruebas de compatibilidad.
- Despliegue de la Aplicación Móvil a producción.

3.6.1 Pruebas unitarias

El *Unit Testing*, consiste en probar pequeñas porciones de código de los componentes del software, para corroborar el buen funcionamiento de este [34], por ello de su importancia, ya sea por requerimientos de la empresa o para que otro desarrollador continúe y sepa que es lo que debe incluir el código.

Se ha realizado la verificación de variables inicializadas y obtención de clientes esperados en el método crear un usuario mediante un servicio propio de *firestore*. El servicio en mención realiza la creación, lectura, actualización y eliminación (*CRUD*).

La información que es de vital importancia, al crear un usuario, se lo realiza mediante el *createDoc* (servicio propio de *firebase*) que se encuentra en el archivo *firestore.service.ts*. Los campos que deben estar definidos son el nombre, apellido, edad, número de cédula, entre otros. Cuando no se completa los campos requeridos, el usuario no puede crear su perfil de manera correcta. El módulo debe tener todos sus campos completos con la información requerida como se muestra en la **Fig. 31**.

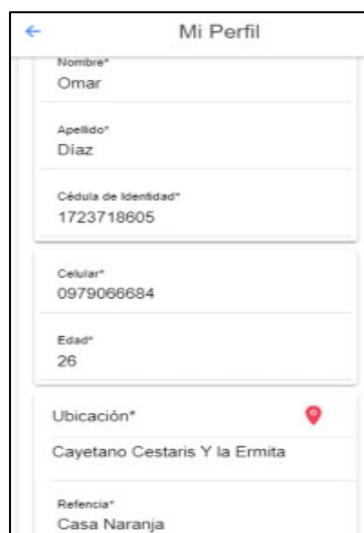


Fig. 31: Módulo Perfil Usuario

La **Fig. 32** hace referencia al método *createDoc* que se encuentra en el archivo *firestore.ts*, generando un método que verifica que las variables estén definidas, y otro método que comprueba que los atributos necesarios se hayan completado al momento de crear un usuario, finalmente un tercer método que autentica las variables que tienen información con los atributos necesarios de un usuario.

```
it('Test - Verificar si las variables del cliente estan definidas', () => {
  const fixture = TestBed.createComponent(TestComponent);
  const app = fixture.componentInstance;
  expect(app.cliente).toBeDefined();
});

it('Test - Validar carga de datos del cliente', () =>{
  const fixture = TestBed.createComponent(TestComponent);
  const app = fixture.componentInstance;
  expect(app.cliente).toBeDefined();
  expect(app.inicioClientes().length).toBeGreaterThan(0);
});

it('Test - registro de clientes', () =>{
  const fixture = TestBed.createComponent(TestComponent);
  const app = fixture.componentInstance;
  expect(app.dataCliente([app.cliente])).toBeDefined();
  expect(app.dataCliente([app.cliente]).length).toBeGreaterThan(0);
});
```

Fig. 32: Generación de métodos para validad información

Al ejecutarse la prueba unitaria, se revisa si las variables están inicializadas para posteriormente tomar datos simulados que se crean con antelación como se muestra en la **Fig. 33**, después de verificar que los campos se encuentran inicializados y comprobar si está o no llegando información a la *data* para la creación del cliente, esperando obtener un resultado exitoso si las variables están definidas y si el arreglo de clientes es superior a uno.

```
this.arregloClientes[0] = {
  uid: '1',
  email: 'omar@gmail.com',
  celular: '0987654321',
  foto: '',
  referencia: 'Casa de dos pisos',
  ubicacion: {
    lat: null,
    lng: null,
    direccion: 'Calle Principal'
  },
  edad: 24,
  nombre: 'Omar',
  apellido: 'Diaz',
  cedula: '1723718605',
  mascotas: [],
  role: 'duenio',
  estadoPaseador: 'inActivo',
};

this.arregloClientes[1] = {
  uid: '2',
  email: 'estaf@gmail.com',
  celular: '0987654321',
  foto: '',
  referencia: 'Casa color azul',
  ubicacion: {
    lat: null,
    lng: null,
    direccion: 'calle la j'
  },
  edad: 24,
  nombre: 'Estafania',
  apellido: 'Aguilar',
  cedula: '1723718600',
  mascotas: [],
  role: 'duenio',
  estadoPaseador: 'inActivo',
};
```

Fig. 33: Datos simulados

Se procede a ejecutar el comando *ng test*, mismo que inicia el ejecutor de tareas *Karma* a través del puerto 9876 del navegador *Google Chrome* de forma automática, en este se indica que las pruebas han sido exitosas como se muestra en la **Fig. 34**.

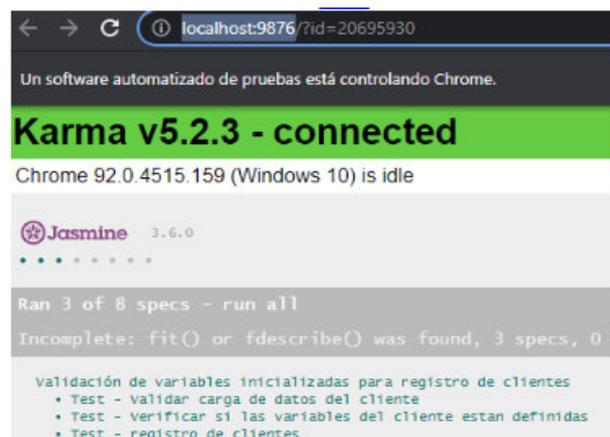


Fig. 34: Resultados de la prueba en el navegador

Además, a través de la pantalla de consola *cmd* de *visual studio code* también se puede observar el mensaje de las pruebas realizadas con éxito como ilustra la **Fig. 35**.

```
WARN: 'Ionic Angular was already initialized. Make sure IonicModule.forRoot() is just called once.'
Chrome 92.0.4515.159 (Windows 10): Executed 1 of 8 SUCCESS (0 secs / 0.125 secs)
WARN: 'Ionic Angular was already initialized. Make sure IonicModule.forRoot() is just called once.'
Chrome 92.0.4515.159 (Windows 10): Executed 2 of 8 SUCCESS (0 secs / 0.185 secs)
Chrome 92.0.4515.159 (Windows 10): Executed 3 of 8 (skipped 5) SUCCESS (0.288 secs / 0.241 secs)
TOTAL: 3 SUCCESS
```

Fig. 35: Resultados de la prueba en la consola

Las restantes pruebas unitarias que se han realizado pueden ser observadas en la sección correspondiente (Pág. 33 - 37) del Manual Técnico.

3.6.2 Pruebas de rendimiento

Las pruebas de rendimiento tienen el objetivo de someter a un sistema con la finalidad de medir su estabilidad, velocidad en esas condiciones de trabajo [35].

Con el objetivo de obtener un resultado de rendimiento en módulos específicos de la Aplicación Móvil se procede a realizar una serie de pruebas dentro del entorno de desarrollo *Ionic*. El valor de rendimiento esperado debe ser igual o superior al 80 %.

La Fig. 39 muestra la creación del archivo *lighthouse.js* necesario para realizar las pruebas en las páginas consideradas de alto tráfico para la Aplicación Móvil. Cabe destacar que en cada página se ha realizado tres pruebas obteniendo el promedio de cada una.

```
module.exports = {
  ci: {
    collect: [
      startServerCommand: 'ionic serve',
      numberOfRuns: 3, // número de análisis
      url: [
        // Dueño
        'http://localhost:8100/registro-mascota',
        'http://localhost:8100/progreso-duenio',
        'http://localhost:8100/perfil-mascota',
        // Paseador
        'http://localhost:8100/home-paseador',
        'http://localhost:8100/procesos-dw',
        // Dueño - paseador
        'http://localhost:8100/',
      ],
      // chromePath: '/usr/bin/google-chrome',
    ],
    upload: {
      target: "temporary-public-storage",
    },
  },
};
```

Fig. 36: Creación del contenido del archivo *lighthouse.js*

La Fig. 40 muestra la cantidad de las pruebas concluidas en cada una de las páginas elegidas mediante *cmd* de *visual studio code*.

```
C:\Users\hit9\Desktop\TestAD-W-Piky>hcl autorun
  ✔ lighthouseci/ directory writable
  ✔ Configuration file found
  ✔ Chrome installation found
  ⚠ GitHub token not set
Healthcheck passed!

Started a web server with "ionic serve"...
WARNING: Timed out waiting for the server to start listening.
Ensure the server prints a pattern that matches /listen|ready/i when it is ready.
Running Lighthouse 3 time(s) on http://localhost:8100/registro-mascota
Run #1...done.
Run #2...done.
Run #3...done.
Running Lighthouse 3 time(s) on http://localhost:8100/progreso-duenio
Run #1...done.
Run #2...done.
Run #3...done.
Running Lighthouse 3 time(s) on http://localhost:8100/perfil-mascota
Run #1...done.
Run #2...done.
Run #3...done.
Running Lighthouse 3 time(s) on http://localhost:8100/home-paseador
Run #1...done.
Run #2...done.
Run #3...done.
Running Lighthouse 3 time(s) on http://localhost:8100/procesos-dw
Run #1...done.
Run #2...done.
Run #3...done.
Running Lighthouse 3 time(s) on http://localhost:8100/
```

Fig. 37: Ejecución y resultados de las pruebas de rendimiento

Desde la Fig. 41 hasta la Fig. 46 se muestran los resultados de rendimiento en las páginas registro-mascota, progreso-duenio, perfil-mascota, home-paseador, progresos-dw, login.

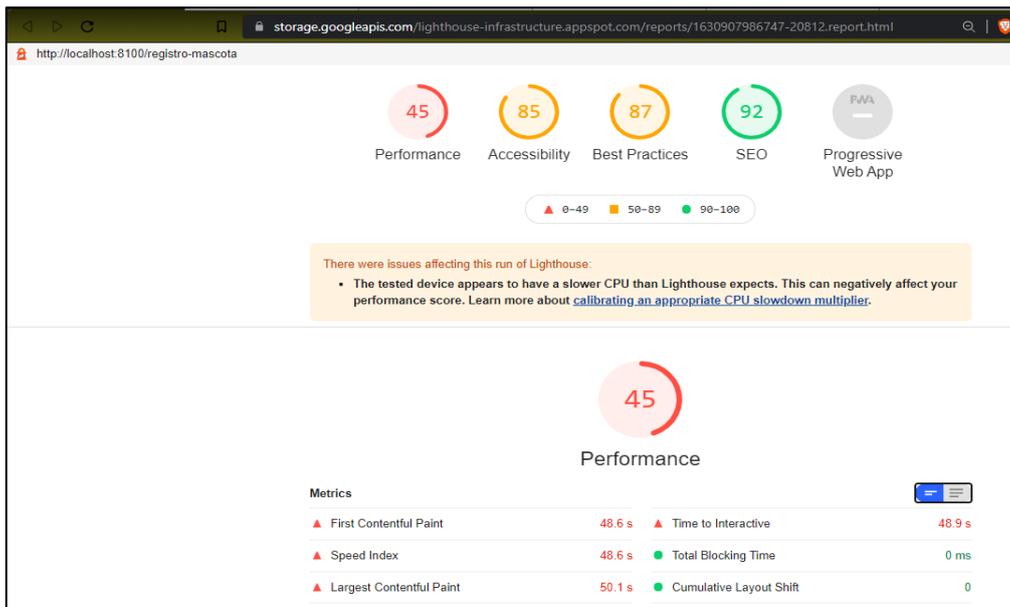


Fig. 38: Resultado de la página registro mascota en el navegador

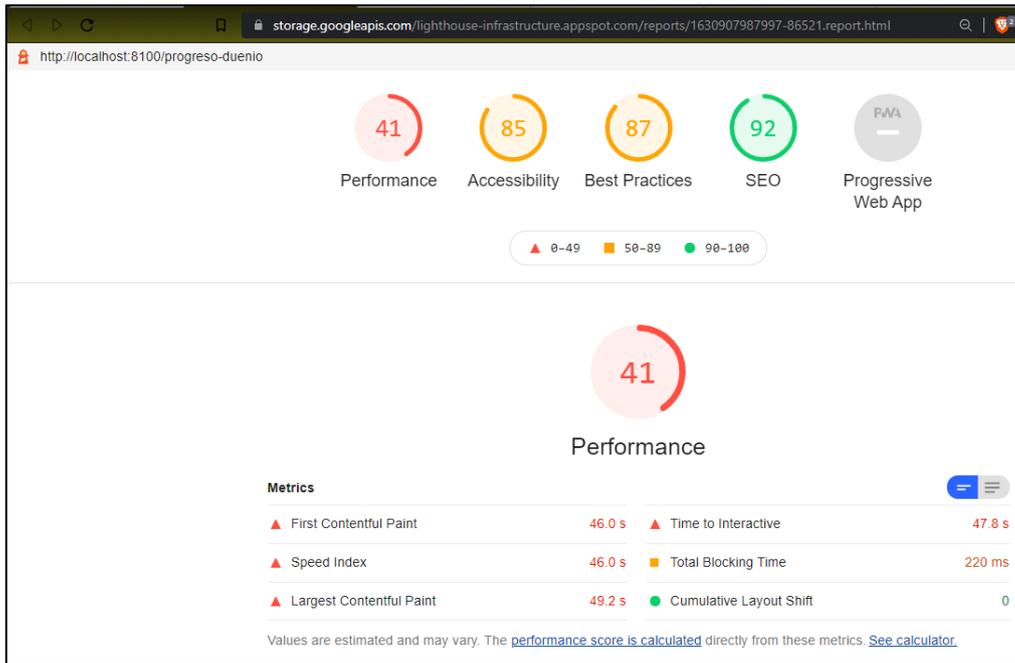


Fig. 39: Resultado de la página progreso-duenio en el navegador

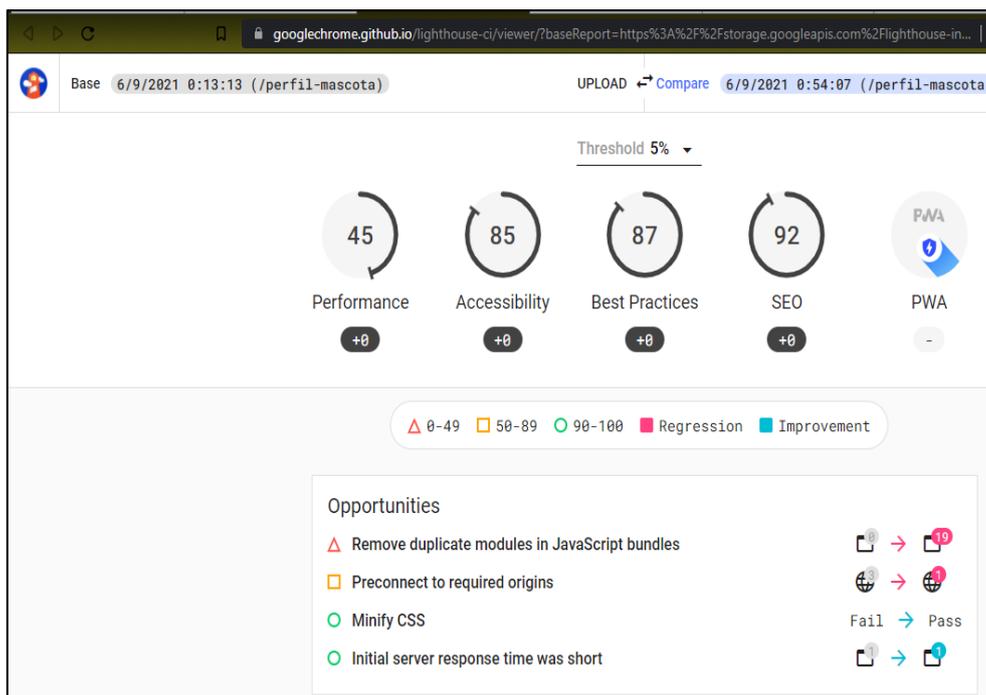


Fig. 40: Resultado de la página perfil-mascota en el navegador

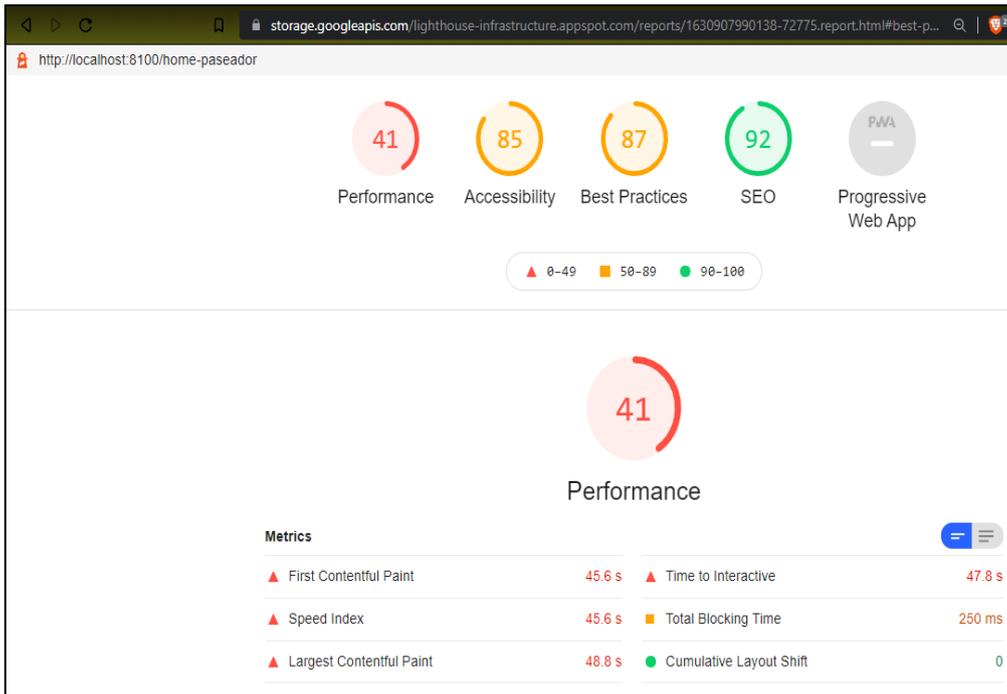


Fig. 41: Resultado de la página home-paseador

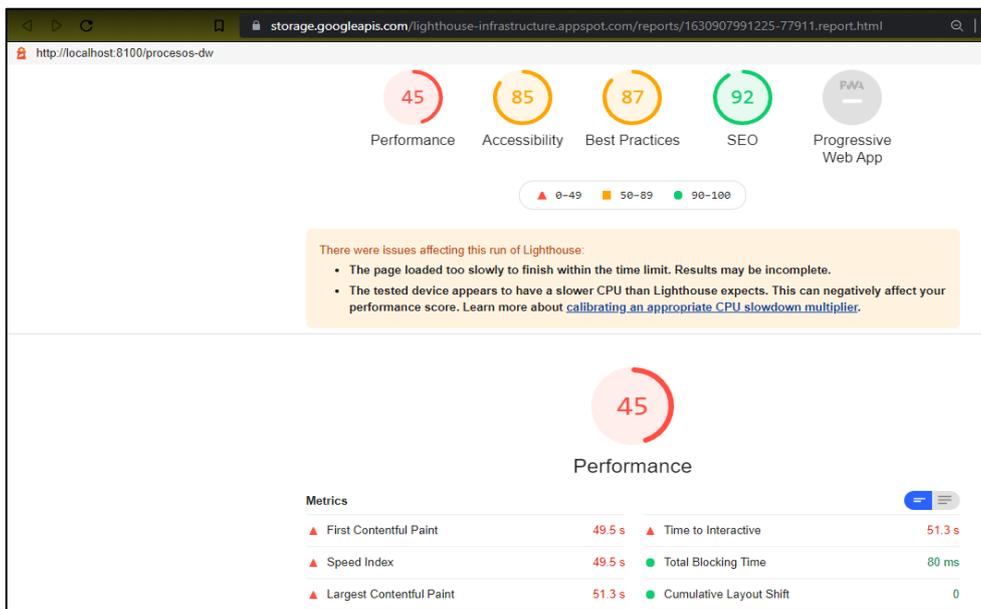


Fig. 42: Resultado de la página procesos-dw

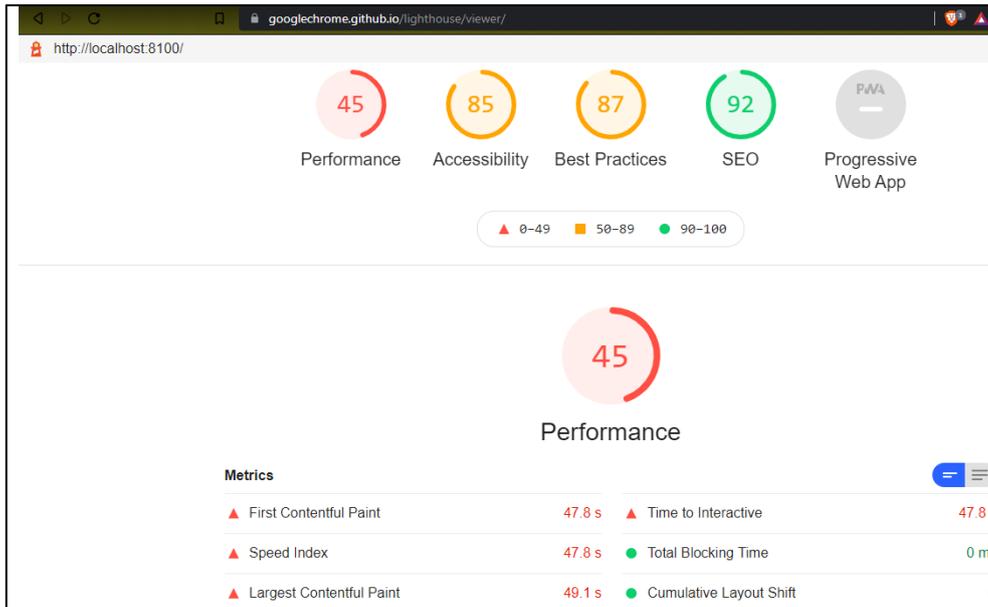


Fig. 43: Resultado de la página login

Otra prueba de rendimiento realizada a la Aplicación Móvil puede ser observada en la sección correspondiente (Pág. 37 - 38) del Manual Técnico.

3.6.3 Pruebas de aceptación

El objetivo de realizar pruebas de aceptación es verificar que la Aplicación Móvil cumple con las expectativas desde el punto de vista de los usuarios finales. Cumpliendo así los requerimientos y procesos del negocio establecido en los requerimientos iniciales [36].

La **TABLA V** muestra un ejemplo de las pruebas de aceptación que se ha realizado en el proyecto, las pruebas restantes pueden ser observadas en la sección correspondiente (Pág. 27 - 31) del Manual Técnico. Las pruebas de aceptación realizadas se encuentran en el Manual Técnico – Sección “Pruebas de Aceptación” (Pág. 38 – 54).

TABLA V: Prueba de aceptación 1 - Registrar y actualizar datos personales

PRUEBA DE ACEPTACIÓN	
Identificador de prueba (ID): PA001	Identificador historia de usuario: HU001
Nombre prueba de aceptación: Registrar y actualizar datos personales	
Descripción: Los usuarios dueño mascota y <i>Dog Walker</i> , tiene la posibilidad actualizar sus datos personales	
Condiciones de prueba: <ul style="list-style-type: none"> Tener conexión a internet 	

<ul style="list-style-type: none"> • Activar el <i>GPS</i> • Iniciar sesión
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Iniciar sesión • Dirigirse al módulo mi perfil • Registrar y actualizar datos personales
<p>Resultado deseado:</p> <p>La Aplicación Móvil mostró un mensaje de éxito cuando se completó el formulario de manera correcta.</p> <p>La Aplicación Móvil mostró un mensaje de error cuando algún dato estuvo mal ingresado o no se ingresó.</p>
<p>Evaluación de la prueba:</p> <p>Se ha comprobado el resultado esperado.</p> <p>Aprobación del cliente 100%</p>

3.6.4 Pruebas de compatibilidad

Las pruebas de compatibilidad se usan para determinar si la Aplicación Móvil es lo suficientemente competente para ejecutarse en diferentes dispositivos móviles [37] con diferentes versiones del sistema operativo y resolución de pantalla.

A continuación, se presenta en la **TABLA VI** los dispositivos en los cuales se han realizado las diferentes pruebas de compatibilidad. Los respectivos resultados se encuentran en la sección correspondiente (Pág. 54 – 56) del Manual Técnico.

TABLA VI: Pruebas de compatibilidad sobre diversos dispositivos móviles

Marca	Modelo	Versión Sistema Operativo	Aceptación
<i>Samsung</i>	A20	10	Aprobado
<i>Samsung</i>	A10	11	Aprobado
<i>Nokia</i>	<i>Nokia 6</i>	7.1.1	Aprobado
<i>Samsung (Tablet)</i>	<i>Galaxy Tab E8</i>	6.01	Aprobado

3.6.5 Despliegue de la Aplicación Móvil a producción

El proceso para el despliegue de la Aplicación Móvil en la tienda de aplicaciones *Google Play* se detalla en el Manual de Instalación – Sección “despliegue de la Aplicación Móvil en *Google Play Store*” (Pág. 5 - 17).

La **Fig. 44** muestra de forma general la información de la Aplicación Móvil tras haber sido aprobada y posteriormente puesta en producción por parte de la tienda de aplicaciones *Google Play*

Versiones más recientes ⓘ					
Versión	Versión más reciente	Segmento	Estado de la versión	Última actualización	Países/regiones
1(1.0)	2	Producción	✓ Disponible en Google Play Lanzamiento completo	21 de septiembre de 2021 19:31	1 de 176

Fig. 44: Despliegue de la Aplicación Móvil a producción

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El producto obtenido cumple con los objetivos establecidos, permitiendo que los ciudadanos dispongan de un medio tecnológico que les permite solicitar un servicio de paseo para sus mascotas, y prevenir que las mismas padecen de alguna enfermedad física, emocional u otras.
- El producto permite que, las personas que desean generar un ingreso extra puedan hacer uso de este sin la necesidad de recurrir a publicaciones en redes sociales u otros canales.
- El uso de la metodología ágil *Scrum* a lo largo del proyecto ha permitió segmentar el proyecto, logrando de esta manera presentar avances funcionales que dieron lugar a la culminación exitosa del mismo.
- Las herramientas empleadas para el desarrollo como *Cloud Firestore*, *Ionic*, *Capacitor*, entre otros han permitido obtener una aplicación que cumple con la arquitectura MVC
- El uso de *Cloud Firestore* ha permitido almacenar y manipular los datos de forma eficiente entre los roles creados para la Aplicación Móvil, posibilitando que toda la transaccionalidad de la información se realice en tiempo real.
- El uso del *Framework Ionic* junto a *Capacitor* han permitido el desarrollo de la Aplicación Móvil sin la instalación manual de varios *plugin* durante la etapa de desarrollo, porque *Capacitor* ya viene con varios *plugin* instalados. También la estructura de archivos que este *Framework* presenta ayudó a desarrollar de mejor manera el patrón arquitectónico MVC
- Con cada una de las pruebas de compatibilidad realizadas en la Aplicación Móvil, se puede demostrar que la misma funciona en diversos dispositivos móviles que usan el sistema operativo *Android* con versiones que varían desde *Android 6.01* hasta *Android 11*.

4.2 Recomendaciones

- Se recomienda, a aquellas personas que deseen hacer uso de la Aplicación Móvil, tener un smartphone con una versión superior a la 5.1 de *Android*, con la finalidad de evitar fallos o errores en las funcionalidades de la Aplicación Móvil.
- Se recomienda, mantener activo el *GPS* porque, este brinda una experiencia optima al momento de visualizar los mapas.
- Los usuarios dueño mascota y *Dog Walker* deben contar con accesos a internet debido a que la Aplicación Móvil está diseñada para trabajar únicamente de manera online.
- Para futuras actualizaciones se recomienda mejorar el envío y recepción de notificaciones porque, actualmente no está optimizada para que funcione en rangos de distancia específicos.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] El Telégrafo, «eltelégrafo,» www.eltelegrafo.com.ec, 30 agosto 2015. [En línea]. Available: <https://bit.ly/3tVSWlv>. [Último acceso: 19 septiembre 2021].
- [2] J. Luque, «Experto Animal,» expertoanimal.com, 21 abril 2020. [En línea]. Available: <https://www.expertoanimal.com/consecuencias-de-no-sacar-a-pasear-al-perro-24881.html>. [Último acceso: 09 octubre 2020].
- [3] P. Carolina, «Consumer,» www.consumer.es, 30 enero 2012. [En línea]. Available: <https://bit.ly/3konxLz>. [Último acceso: 24 febrero 2021].
- [4] ali, «ali,» diariamenteali.com, 03 septiembre 2020. [En línea]. Available: https://diariamenteali.com/articulo/beneficios_de_sacar_a_pasear_a_tu_perro. [Último acceso: 24 febrero 2021].
- [5] dog hero, «dog hero,» heroe.doghero.com.ar, [En línea]. Available: <https://heroe.doghero.com.ar/perro-asustado-que-hacer/#:~:text=Una%20de%20las%20causas%20m%C3%A1s,m%C3%A1s%20profundos%20que%20los%20nuestros..> [Último acceso: 24 febrero 2021].
- [6] GK, «GK,» gk.city, [En línea]. Available: <https://bit.ly/3hP6aBU>. [Último acceso: 01 septiembre 2021].
- [7] Lord Guau, «Lord Guau,» www.lordguau.com, [En línea]. Available: <https://www.lordguau.com/adiestramiento/>. [Último acceso: 01 septiembre 2021].
- [8] illusion Studio, «www.illusionstudio.es,» illusionstudio, 12 febrero 2020. [En línea]. Available: <https://www.illusionstudio.es/apps-moviles-importancia-actualidad#:~:text=A%20d%C3%ADa%20de%20hoy%20las,Comunicaci%C3%B3n..> [Último acceso: 25 agosto 2021].
- [9] Ciencia Digital, «Ciencia Digital,» cienciadigital.org, 2019. [En línea]. Available: <https://cienciadigital.org/revistacienciadigital2/index.php/CienciaDigital/article/view/83>

- [20] SOFTENG, www.softeng.es, [En línea]. Available: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>. [Último acceso: 16 septiembre 2021].
- [21] Capterra, «Capterra,» www.capterra.ec, [En línea]. Available: <https://www.capterra.ec/software/153402/ninjamock>. [Último acceso: 16 septiembre 2021].
- [22] desarrolloweb.com, «desarrolloweb.com,» desarrolloweb.com, 28 julio 2020. [En línea]. Available: <https://desarrolloweb.com/articulos/que-es-mvc.html>. [Último acceso: 16 septiembre 2021].
- [23] J. Aguilar, «campusmvp MVP,» www.campusmvp.es, [En línea]. Available: <https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>. [Último acceso: 16 septiembre 2021].
- [24] zendesk, «zendesk,» www.zendesk.com.mx, 12 enero 2021. [En línea]. Available: <https://bit.ly/3AIDs2G>. [Último acceso: 16 septiembre 2021].
- [25] S. López, «DIGITAL 55,» www.digital55.com, 2020 mayo 17. [En línea]. Available: <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/>. [Último acceso: 16 septiembre 2021].
- [26] Firebase, «Firebase,» firebase.google.com, [En línea]. Available: <https://firebase.google.com/docs/firestore/rtdb-vs-firestore?hl=es>. [Último acceso: 13 septiembre 2021].
- [27] profile, «profile,» profile.es, 22 febrero 2021. [En línea]. Available: <https://profile.es/blog/que-es-ionic/>. [Último acceso: 13 septiembre 2021].
- [28] rockcontent, «rockcontent,» rockcontent.com, 12 abril 2020. [En línea]. Available: <https://rockcontent.com/es/blog/bootstrap/>. [Último acceso: 13 septiembre 2021].
- [29] M. Robledano, «OpenWebinars,» openwebinars.net, 26 junio 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-css/>. [Último acceso: 13 septiembre 2021].
- [30] bssoftware, «bssoftware,» bsw.es, [En línea]. Available: <https://n9.cl/ozm1x>. [Último acceso: 16 septiembre 2021].

- [31] M. García, «DIGITAL55,» www.digital55.com, 25 noviembre 2019. [En línea]. Available: <https://bit.ly/3kmlp5F>. [Último acceso: 16 septiembre 2021].
- [32] I. Buioli, «www.moldeointeractive.com.ar,» Moldeo Interactive, 14 noviembre 2019. [En línea]. Available: <https://bit.ly/3tTdzVz>. [Último acceso: 16 septiembre 2021].
- [33] ICHI.PRO, «[I CHI.PRO](http://ichi.pro),» ichi.pro, [En línea]. Available: <https://ichi.pro/es/inicio-de-sesion-de-google-en-la-aplicacion-ionic-capacitor-con-angular-205671215375904>. [Último acceso: 16 septiembre 2021].
- [34] M. Pulido, «[slashmobility](http://slashmobility.com),» slashmobility.com, 27 agosto 2019. [En línea]. Available: <https://slashmobility.com/blog/2019/08/tutorial-unit-testing-con-karma-y-jasmine/>. [Último acceso: 31 agosto 2021].
- [35] C. Lucena, «[OpenWebinars](http://openwebinars.net),» openwebinars.net, 14 junio 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-prueba-de-rendimiento-software/>. [Último acceso: 25 septiembre 2021].
- [36] digité, «[digité](http://digite.com),» www.digite.com, [En línea]. Available: <https://www.digite.com/es/agile/pruebas-de-acceptacion/>. [Último acceso: 17 septiembre 2021].
- [37] myservername.com, «myservername.com,» es.myservername.com, [En línea]. Available: <https://bit.ly/3Cutehe>. [Último acceso: 17 septiembre 2021].

6 ANEXOS

- A continuación, se listan los documentos que se generaron a partir de la Aplicación Móvil.

6.1 Manual Técnico

1. Recopilación de requerimientos
2. Historias de usuario
3. *Product Backlog*
4. *Sprint Backlog*
5. Prototipo de interfaces
6. Diseño de la Base de Datos
7. Pruebas unitarias
8. Pruebas de rendimiento
9. Pruebas de aceptación
10. Pruebas de compatibilidad

6.2 Manual de Usuario

- <https://www.youtube.com/watch?v=MptnzWwgM7Q>

6.3 Manual de Instalación

- Despliegue de la Aplicación Móvil en *Google Play Store*