



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA DE SISTEMAS

APLICACIÓN MÓVIL MULTIPLATAFORMA DE ASISTENCIA VIAL PARA LA
COMUNIDAD DE MOTOCICLISTAS DE LA CIUDAD DE CUENCA

Trabajo de titulación previo a la obtención del
título de Ingeniero de Sistemas

AUTORES: JUAN CARLOS BENAVIDES CABRERA
LUIS FERNANDO SÁNCHEZ SUCUZHAÑAY
TUTOR: ING. CRISTIAN FERNANDO TIMBI SISALIMA, MSc.

Cuenca - Ecuador

2023

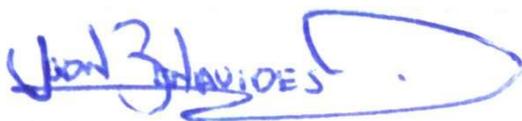
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Juan Carlos Benavides Cabrera con documento de identificación N° 0105817639 y Luis Fernando Sánchez Sucuzhañay con documento de identificación N° 0107181562; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 13 de marzo del 2023

Atentamente,



Juan Carlos Benavides Cabrera

0105817639



Luis Fernando Sánchez Sucuzhañay

0107181562

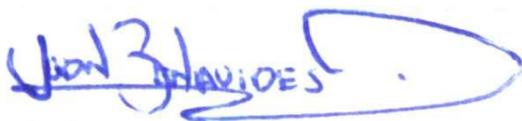
CERTIFICADO DE SESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Juan Carlos Benavides Cabrera con documento de identificación N° 0105817639 y Luis Fernando Sánchez Sucuzhañay con documento de identificación N° 0107181562, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico: “Aplicación móvil multiplataforma de asistencia vial para la comunidad de motociclistas de la ciudad de Cuenca”, el cual ha sido desarrollado para optar por el título de: Ingeniero de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 13 de marzo del 2023

Atentamente,



Juan Carlos Benavides Cabrera

0105817639



Luis Fernando Sánchez Sucuzhañay

0107181562

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Cristian Fernando Timbi Sisalima con documento de identificación N° 0103709911 docente de la Universidad Politécnica Salesiana declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: APLICACIÓN MÓVIL MULTIPLATAFORMA DE ASISTENCIA VIAL PARA LA COMUNIDAD DE MOTOCICLISTAS DE LA CIUDAD DE CUENCA, realizado por Juan Carlos Benavides Cabrera con documento de identificación N° 0105817639 y por Luís Fernando Sánchez Sucuzhañay con documento de identificación N° 0107181562, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 13 de marzo del 2023

Atentamente,



Ing. Cristian Fernando Timbi Sisalima, MsC.

0103709911

DEDICATORIA

Dedico mi tesis a mis padres Mary y Carlos, quienes, con amor, esfuerzo me han apoyado incondicionalmente en mis estudios desde pequeño y han permitido hoy cumplir mi sueño, gracias por inculcar en mí el valor del respeto, estudio, esfuerzo y dedicación. Los quiero papás.

A mis hermanos, Adrián y Verónica por haberme apoyado en este camino, por estar presentes y por apoyo moral.

A mi abuelita Judith y a mi abuelito Gonzalo que en paz descanse, por inspirarme y siempre haberse preocupado por mí.

A Camila Villacís, por su paciencia, amor, cariño, apoyo y por qué nunca faltó una palabra de aliento durante el proceso de este proyecto.

A mis amigos que conocí en mi etapa universitaria Hernán, Willy, Fernando, Oscar, Omar, Bryan, Wilson, Mayra, José queridos colegas gracias por toda su ayuda, buena voluntad, esta etapa no hubiese sido la mejor sin ustedes.

A mis amigos de Territorios Inteligentes, por haber compartido conmigo mi primera experiencia profesional ejerciendo mi carrera y siempre siendo un gran apoyo, los recuerdo a todos con mucho cariño.

Por último, va dedicada a mis amigos de toda la vida David, Fercho, Mateo, Daniel, Alejandra y Paula por formar parte de mi proceso de crecimiento.

Juan Carlos Benavides Cabrera.

El presente trabajo de titulación va dedicado a mis cercanos seres queridos, a mi abuela Zoila Lucrecia Bravo Parra, quien me educo, apoyo e instruyo con valores morales siendo así uno de mis pilares fundamentales, a mis padres Jaime Sánchez Úrgiles y Mercedes Sucuzhañay Bravo, quienes se sacrificaron por brindarme el estudio y seguir impulsándome a conseguir un mejor futuro, a mis hermanas por acompañarme y apoyarme en lo largo de toda la mi vida, por último a mis tíos por todo los consejos, ánimos, enseñanzas,

A todas las antes menciones, por el apoyo otorgado para conseguir una meta más de mi vida, aportando en esta formación profesional.

Luis Fernando Sánchez Sucuzhañay

AGRADECIMIENTO

Me faltarán páginas por nombrar a todas las personas con las que estoy enormemente agradecido, pero se merece un reconocimiento especial mis padres Mary y Carlos por el sacrificio de día a día que pusieron en mi para que mi formación académica sea la mejor.

También quiero agradecer a mis tíos, Claudio y Anita, gracias han sido como mis segundos padres.

A mi compañero de tesis Fernando, por creer en la idea de que este proyecto fuese posible, por su dedicación y sobre todo por la paciencia.

A mis profesores que han sido un pilar fundamental en mi proceso universitario en especial al Dr. Vladimir Robles, ya que aparte de ser un gran instructor en el aula de clases, ha sabido guiarme e inspirarme para nunca dejar de aprender.

Quiero agradecer a mi tutor de tesis el Ing. Cristian Timbi, ya que gracias a sus conocimientos logré conseguir mi primera experiencia laboral, y por su amplio conocimiento y desarrollo que permitieron el éxito de este proyecto.

Por último, gracias a mis profesores que dejaron en mí una gran enseñanza: Mauricio Ortiz, Álvaro Mejía, Pablo Gallegos, Diego Quinde, Walter Verdugo, María Anaís Leguizamo y Rodolfo Bojorque.

Juan Carlos Benavides Cabrera.

Agradezco a mi abuela materna, por haberme dado todo su cariño, amor como madre y padre, por acompañarme y apoyarme en todos mis años de estudio, a mis padres por el esfuerzo que han hechos durante estos años para brindarme la oportunidad de estudiar esta increíble carrera. A demás agradezco a mis amigos que conocí a lo largo de la carrera, a mis compañeros de trabajo, a los docentes competentes que impactaron en la formación de buenos profesionales con sus enseñanzas sobre las materias brindadas y experiencias en el ámbito laboral. Así también, a nuestro tutor de tesis el Ing. Cristian Fernando Timbi Sisalima quien aparte de ser un docente ejemplar en las materias impartidas, su conocimiento para culminar de la mejor manera este proyecto.

Luis Fernando Sánchez Sucuzhañay

RESUMEN

El presente estudio trata acerca de la tendencia y el aumento de usuarios motociclistas, inclinados hacia la moto aventura, y el auge de la tecnología con respecto a las aplicaciones móviles.

El objetivo principal de este proyecto es brindar asistencia vial, geolocalización y controles mecánicos mediante una aplicación. Se ha propuesto el desarrollo que permita hacer de un viaje una experiencia segura, para lo cual se realizó el análisis de requerimientos mediante encuestas a grupos de rutereros, quienes mostraron bastante interés y emoción en la propuesta y dando resultados positivos, sobre todo por la geolocalización. Posteriormente se escogió una arquitectura basada en n-capas lo que nos permitió separar el Frontend del Backend. Una vez realizada la encuesta y con los objetivos propuestos, para la implementación se escogió el framework de Flutter con su lenguaje Dart para sus interfaces, por su eficiencia, costo de desarrollo, su característica multiplataforma, programación reactiva, comportamiento nativo, se aplicó por ExpressJS(NodeJS) para el control con la base de datos por su facilidad de programar y su velocidad de respuesta, por último se optó por una base de datos no relacional MongoDB ya que es ideal por su coste bajo, su compatibilidad con JavaScript.

La aplicación servirá a que los usuarios de motocicleta tengan facilidad para viajar, sociabilizar con usuarios de interés y en caso de cualquier percance tener control en la vía. Con este proyecto se ha cumplido con el objetivo principal, de crear una aplicación funcional dirigida a la comunidad motera que les permitirá facilitar su actividad en la carretera. Un punto importante por el que hemos escogido el framework de Flutter es por el concepto de crear mediante widgets, por consiguiente, se estudió los fundamentos de programación de estas tecnologías, se realizó los diseños y se puso en marcha el proyecto.

Palabras claves: Aplicación móvil, aplicación web, servicios web, software de aplicación, flutter, base de datos no relacional.

ABSTRACT

This study deals with the trend and the increase in motorcyclist users, inclined towards adventure motorcycles, and the increase in technology regarding mobile applications.

The main objective of this project is to provide road assistance, geolocation, and mechanical controls through an application. The development that allows making a trip a safe experience has been proposed, for which the analysis of requirements was carried out through surveys of groups of routers, who showed a lot of interest and emotion in the proposal and gave positive results, especially due to geolocation. Subsequently, an architecture based on n-layers was chosen, which allowed us to separate the frontend from the backend. Once the survey was carried out and with the proposed objectives, for the implementation the Flutter framework was chosen with its Dart language for its interfaces, due to its efficiency, development cost, its multiplatform characteristic, reactive programming, native behavior, it was applied by ExpressJS (NodeJS) for the control with the database due to its ease of programming and its response speed, finally, a non-relational MongoDB database was chosen since it is ideal for its low cost, its compatibility with JavaScript.

The application will make it easy for motorcycle users to travel, socialize with users of interest and in case of any possibility of having control on the road. With this project, the main objective has been met, of creating a functional application aimed at the biker community that will allow them to facilitate their activity on the road, An important point for which we have chosen the Flutter framework is because of the concept of creating through widgets, therefore, the programming fundamentals of these technologies were studied, the designs were made, and the project was launched.

Keywords: Mobile app, web app, web services, application software, flutter, non-relational database.

ÍNDICE GENERAL

1. INTRODUCCIÓN	15
2. PROBLEMA	17
2.1. Antecedentes	17
2.2. Explicación del problema.....	17
2.3. Importancia y alcances	18
3. OBJETIVOS GENERALES Y ESPECÍFICOS	19
3.1. Objetivo General	19
3.2. Objetivo Específicos	19
4. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS	20
4.1. Plataformas en dispositivos móviles.	20
4.2. Tecnologías de desarrollo.....	30
4.3. Widgets	32
4.4. Firebase	34
4.5. Servicios web – Api Rest.....	35
4.6. Servidores alojados en la nube	35
5. MARCO METODOLÓGICO	37
5.1. Entorno de desarrollo	37
5.2. Creación de la aplicación.....	40
5.3. Levantamiento de requerimientos.....	41
5.4. Definición de la arquitectura del sistema.....	44
5.5. Creación de la aplicación.....	47
5.6. Creación de la base de datos.....	52
5.7. Creación de widgets	59
5.8. Notificaciones Push.	61
5.9. Módulos de la aplicación.....	65
6. RESULTADOS	71
6.1. Análisis de Performance.....	71
6.2. Pruebas de requerimiento funcionales.....	72
6.3. Análisis de resultados de la aplicación.	99
7. CRONOGRAMA.....	101
7.1. Cronograma de actividades.....	101
7.2. Tabla de actividades.....	102
8. PRESUPUESTO	104
9. CONCLUSIONES.....	105
10. RECOMENDACIONES.....	107

LISTA DE FIGURAS

<i>Ilustración 1.- Diagrama del funcionamiento de una aplicación móvil nativa.....</i>	<i>20</i>
<i>Ilustración 2.- Diagrama aplicaciones híbridas</i>	<i>22</i>
<i>Ilustración 3.- Diagrama de funcionamiento de una aplicación web móvil</i>	<i>23</i>
<i>Ilustración 4.- Diagrama de barras de la cuota de mercado de sistemas operativos móviles</i>	<i>27</i>
<i>Ilustración 5.- Que es Flutter</i>	<i>31</i>
<i>Ilustración 6.- Flutter project structure</i>	<i>32</i>
<i>Ilustración 7.- Ejemplo de un árbol de widgets</i>	<i>33</i>
<i>Ilustración 8.- Configuración de variable de entorno.....</i>	<i>37</i>
<i>Ilustración 9.- Flutter Doctor</i>	<i>38</i>
<i>Ilustración 10.- IDE de desarrollo, extensión Flutter</i>	<i>38</i>
<i>Ilustración 11.- IDE de desarrollo, extensión Dart.....</i>	<i>38</i>
<i>Ilustración 12.- IDE de desarrollo, extensión Awesome Flutter Snippets</i>	<i>39</i>
<i>Ilustración 13.- Diagrama de sistemas de control distribuidas.</i>	<i>39</i>
<i>Ilustración 14.- Diagrama de casos de uso interacción del usuario con la aplicación.....</i>	<i>42</i>
<i>Ilustración 15.- Casos de uso interacción de reportes y bloqueos Usuario Administrador</i>	<i>43</i>
<i>Ilustración 16.- Arquitectura del sistema.</i>	<i>44</i>
<i>Ilustración 17.- Arquitectura del sistema.</i>	<i>45</i>
<i>Ilustración 18.- Test de conexión y funcionamiento de sockets</i>	<i>46</i>
<i>Ilustración 19.- Código que permite ver si un usuario fue desconectado.....</i>	<i>46</i>
<i>Ilustración 20.- UI – Diseño del ingreso y registro de la aplicación</i>	<i>49</i>
<i>Ilustración 21.- UI – Pantallas de feed de publicaciones y chats.....</i>	<i>50</i>
<i>Ilustración 22.- UI – Agregar amigos y diseño de la pantalla de un chat.....</i>	<i>50</i>
<i>Ilustración 23.- UI – Diseño de navegación a una ruta.....</i>	<i>51</i>
<i>Ilustración 24.- Paleta de colores</i>	<i>51</i>
<i>Ilustración 25.- Pantalla de creación de organización</i>	<i>52</i>
<i>Ilustración 26.- Pantalla de creación de proyecto.</i>	<i>52</i>
<i>Ilustración 27.- Pantalla de creación de la base de datos.</i>	<i>53</i>
<i>Ilustración 28.- Arquitectura del clúster.....</i>	<i>53</i>

<i>Ilustración 29.- Colección usuario</i>	54
<i>Ilustración 30.- Colección motocicleta</i>	54
<i>Ilustración 31.- Colección manteniendo de la motocicleta</i>	55
<i>Ilustración 32.- Colección amigos</i>	55
<i>Ilustración 33.- Colección publicaciones</i>	56
<i>Ilustración 34.- Colección comentarios</i>	56
<i>Ilustración 35.- Colección rutas</i>	57
<i>Ilustración 36.- Colección coordenadas</i>	57
<i>Ilustración 37.- Colección de marcadores</i>	57
<i>Ilustración 38.- Colección mensajes</i>	58
<i>Ilustración 39.- Colección reporte</i>	58
<i>Ilustración 40.- Listado de widgets creados</i>	59
<i>Ilustración 41.- Widget Circle avatar</i>	60
<i>Ilustración 42.- Pantalla principal de configuración Firebase</i>	61
<i>Ilustración 43.- Aplicaciones creadas</i>	62
<i>Ilustración 44.- Diagrama de flujo de notificaciones</i>	62
<i>Ilustración 45.- Validación de una notificación push</i>	63
<i>Ilustración 46.- Diagrama de flujo de la notificación push</i>	64
<i>Ilustración 47.- Notificaciones en primer y segundo plano</i>	64
<i>Ilustración 48.- UI Pantalla de bienvenida o logueo y registro</i>	65
<i>Ilustración 49.- UI Pantalla de noticias o publicaciones</i>	66
<i>Ilustración 50.- UI Pantalla de historial de chats y chats</i>	67
<i>Ilustración 51.- UI Pantalla de perfil</i>	68
<i>Ilustración 52.- UI pantalla carné digital</i>	69
<i>Ilustración 53.- UI interfaces de navegación de ruta, búsqueda de destino y guardar un marcador</i>	70
<i>Ilustración 54.- Métrica de rendimiento en el dispositivo de la plataforma iOS</i>	71
<i>Ilustración 55.- Métrica de rendimiento en el simuladore de la plataforma Android</i>	72

LISTA DE TABLAS

<i>Tabla 1.- Comparación de tipos de aplicaciones móviles.....</i>	<i>24</i>
<i>Tabla 2.- Actividades del objetivo OE1.....</i>	<i>101</i>
<i>Tabla 3.- Actividades del objetivo OE2.....</i>	<i>101</i>
<i>Tabla 4.- Actividades del objetivo OE3.....</i>	<i>101</i>
<i>Tabla 5.- Actividades del objetivo OE4.....</i>	<i>102</i>
<i>Tabla 6.- Actividades del objetivo OE5.....</i>	<i>102</i>
<i>Tabla 7.- Fechas del cronograma de actividades.....</i>	<i>102</i>
<i>Tabla 8.- Presupuesto.....</i>	<i>104</i>

1. INTRODUCCIÓN

“La revolución tecnológica ya se ha producido, y lo que entendemos por disrupción digital afecta a todos los ámbitos de nuestra vida, y por supuesto a todos los sectores productivos.” (Aguilera, 2016)

Por lo tanto, el crecimiento exponencial en la industria tecnológica se extiende al ámbito móvil, que surge como consecuencia de la aceleración digital y la demanda del mercado. Los teléfonos inteligentes se han convertido en herramientas indispensables para el uso cotidiano, ya que facilitan el análisis o seguimiento de diferentes actividades. En este contexto, las aplicaciones móviles juegan un papel crucial, ya que permiten a los usuarios acceder a información en tiempo real, realizar transacciones y simplificar tareas cotidianas. Por lo tanto, el desarrollo de aplicaciones móviles se ha convertido en una prioridad para las empresas que desean adaptarse a las necesidades del mercado y mantenerse a la vanguardia de la industria tecnológica.

El proceso de desarrollo de una aplicación móvil implica varios pasos, que incluyen la planificación, el diseño, la programación, las pruebas y el lanzamiento de la aplicación en una tienda de aplicaciones. (Torres Remon, 2017)

Una aplicación móvil es un software diseñado para dispositivos móviles, que permite al usuario final cumplir con diversas funcionalidades que se obtienen gracias a los servicios específicos realizados por los desarrolladores de Back-End.

Entre las diversas aplicaciones móviles, Google Maps una aplicación móvil que ofrece información geográfica detallada y servicios de navegación en tiempo real; En la actualidad tiene más de 1000 millones de usuarios activos que pueden acceder a las fichas de más de 150 millones de empresas y lugares, lo que facilita la planeación de viajes y toma de decisiones (Earth, 2022).

Gracias a esta herramienta de ubicación efectiva que permite a los usuarios llegar de un

punto A, a un punto B se puede aprovechar para la implementación de la aplicación que ayudar a los usuarios de motocicleta.

Los motociclistas a menudo realizan viajes en los que pueden enfrentar contratiempos, como perderse, sufrir daños en su motocicleta, perder contacto con el grupo de moteros o experimentar la falta de asistencia vial. Para solucionar estos problemas, la aplicación móvil resulta muy útil. Esta aplicación utiliza la geolocalización de los usuarios para ayudarles a encontrar su camino, proporciona asistencia vial en caso de emergencia y ofrece una red social para moteros que les permite compartir su aventura y mantenerse en contacto con otros miembros del grupo.

Se ha desarrollado un sistema administrativo web a través de Flutter, que permite a los usuarios con privilegios de administrador regular el uso apropiado de la aplicación, incluyendo la capacidad de enviar notificaciones del comportamiento del usuario, bloquear usuario, la eliminación de comentarios y publicaciones.

Además, es importante mencionar que el sistema administrativo también cuenta con gráficas que muestran el historial de bloqueos por mes, lo que permite a los usuarios administradores visualizar y analizar los patrones de uso inapropiado y tomar medidas para prevenir futuros incumplimientos.

2. PROBLEMA

2.1. Antecedentes

Considerando el aumento de usuarios de motocicletas y los problemas mencionados anteriormente que enfrentan a diario, surgió la necesidad de desarrollar una aplicación que mejore su calidad de vida. La aplicación provee soluciones a problemas comunes, como la planificación de rutas, estadísticas de geolocalización de rutas, ubicación de gasolineras, talleres mecánicos y paradas específicos, asistencia vial en caso de emergencia, una sección para el control del mantenimiento del vehículo, incluyendo el cambio de aceite y el monitoreo del consumo de gasolina, creación de una red social específica para compartir rutas o imágenes de rutas. Con esta aplicación, se brinda a los motociclistas una herramienta valiosa para mejorar su experiencia de conducción y reducir los contratiempos que enfrentan en el camino, como también interactuar con miembro de la misma comunidad motociclista.

2.2. Explicación del problema

Cada año, el número de aficionados a las motocicletas sigue aumentando. Estos apasionados de la conducción suelen recorrer caminos desconocidos por entretenimiento o turismo, pero a menudo se enfrentan a percances en la ruta, como daños en su vehículo, la falta de conocimiento sobre la ubicación de la próxima gasolinera o taller mecánico, o situaciones de riesgo como un accidente de tráfico o la pérdida de uno de sus compañeros de ruta. Por estas razones, es fundamental contar con asistencia en la vía.

Además, la planificación de un recorrido puede ser un dilema para los motociclistas. La elección de la ruta más segura o la ruta con las mejores vistas y paradas puede ser una decisión difícil. Es por eso por lo que surge la necesidad de planificar el viaje o elegir una ruta recomendada por otro motociclista con experiencia.

Finalmente, es importante recordar que los conductores de motocicletas necesitan realizar un cambio de aceite periódicamente y controlar el consumo de gasolina. El olvido de alguna de estas tareas puede resultar en un inconveniente en la ruta.

2.3. Importancia y alcances

El sistema implementado será enfocado hacia los usuarios motociclistas con el objetivo de hacer que sus viajes sean más seguros y placenteros. En caso de un percance, los usuarios recibirán asistencia para evitar quedarse varados en la carretera gracias a los marcadores personalizados. También tendrán el control total de su ruta en tiempo real, ya sea en compañía de otros motociclistas o en solitario. Los usuarios podrán visualizar las rutas compartidas por otros usuarios de la aplicación y también interactuar con aquellos que consideren amigos.

3. OBJETIVOS GENERALES Y ESPECÍFICOS

3.1. Objetivo General

Diseñar e implementar un prototipo de una aplicación móvil multiplataforma que permita asistencia en la vía dirigida a la comunidad motera y además desarrollar un apartado que permita tener controles mecánicos de la motocicleta.

3.2. Objetivo Específicos

1. OE1. Estudiar los fundamentos del desarrollo de aplicaciones móviles multiplataforma y las tecnologías adecuadas.
2. OE2. Definir y documentar los requerimientos de la aplicación móvil.
3. OE3. Desarrollar la aplicación móvil multiplataforma de acuerdo con los requerimientos definidos.
4. OE4. Validar el correcto funcionamiento de la aplicación por medio de la ejecución de las diferentes pruebas de software.
5. OE5. Elaborar la documentación de la aplicación desarrollada.

4. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS

4.1. Plataformas en dispositivos móviles.

Los Aplicación móvil

Es una aplicación que es ejecutada en un dispositivo móvil, smartphone o Tablet. Una aplicación es un software que cumple ciertas funciones que proporcionan servicios a los usuarios y están diseñadas para hacer el trabajo más sencillo, hacer de nuestro dispositivo móvil más eficiente y usado en la vida cotidiana. (GCFGlobal, 2021)

Aplicaciones nativas

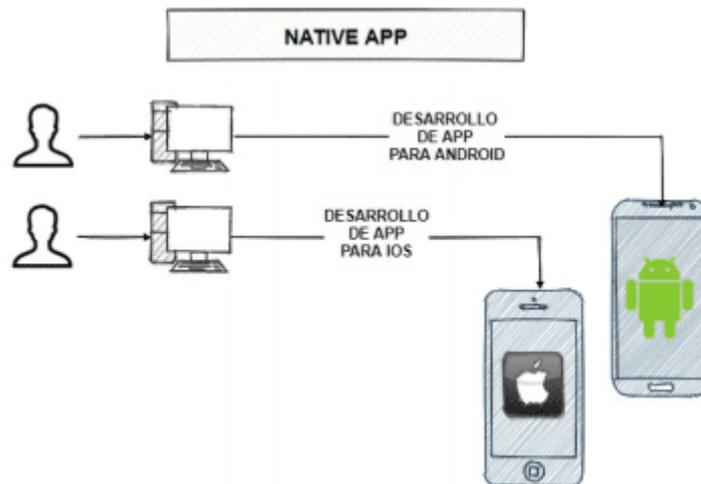


Ilustración 1.- Diagrama del funcionamiento de una aplicación móvil nativa.

Nota. Tomado de Principales tipos de aplicaciones móviles: ventajas, desventajas y ejemplos
(Bernal González, 2021)

Estas aplicaciones están diseñadas para un sistema operativo móvil en específico, Android o iOS, su principal objetivo es el alto rendimiento y excelente experiencia de usuario. Ejemplo: Kotlin. (Vázquez, 2015)

Ventajas:

- Acceso completo al dispositivo en hardware y software
- Mejor experiencia de usuario y rendimiento
- Visualización desde las tiendas de aplicaciones móviles

Desventajas:

- Requiere diferentes lenguajes de programación nativas y habilidades según el sistema operativo.
- Costos y tiempos de desarrollo altos.

Aplicaciones móviles híbridas o multiplataforma

Las plataformas multiplataforma son una solución eficaz para los desarrolladores que buscan crear aplicaciones que puedan ser utilizadas en múltiples plataformas de manera simultánea a partir de un solo conjunto de código. Ejemplo: Dart. (Velasco, 2016)

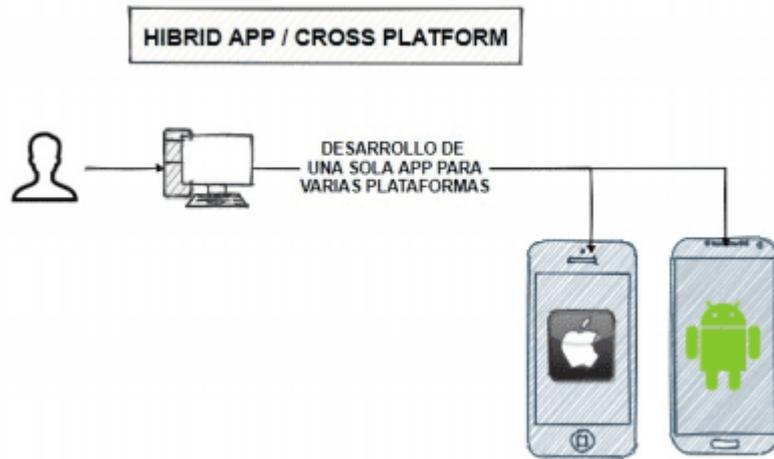


Ilustración 2.- Diagrama aplicaciones híbridas

Nota. Tomado de *Principales tipos de aplicaciones móviles: ventajas, desventajas y ejemplos* (Bernal González, 2021)

Ventajas

- Su desarrollo es menos costoso y más sencillo, ya que solo se desarrolla una vez
- Es multiplataforma y puede ser subido en más de una tienda de aplicaciones móviles.
- Permite la reutilización de código, la cual permite ahorrar tiempo a los programadores.

Desventajas

- La velocidad y el rendimiento son menores, pero no tanto, hoy en día casi imperceptibles.

Aplicaciones móviles Web

Son las aplicaciones para las que se realiza un solo desarrollo y puede ser ejecutada desde cualquier navegador móvil. Ejemplo: PHP. (Genexus, 2016)

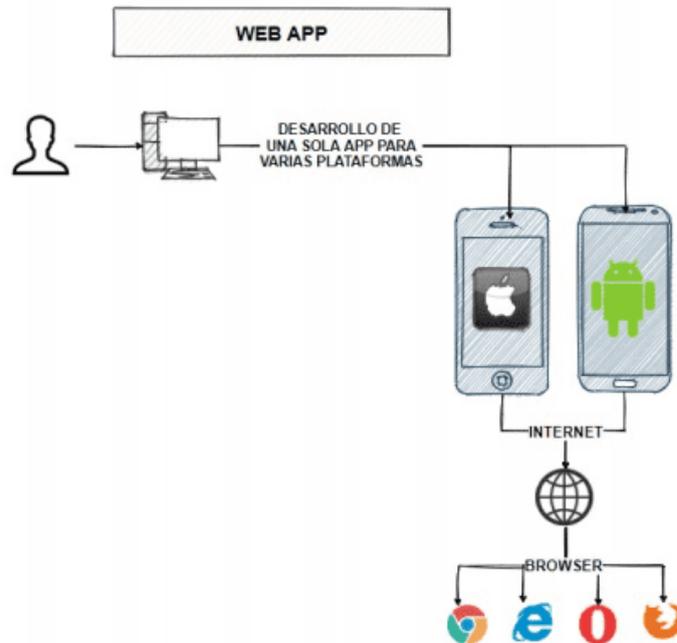


Ilustración 3.- Diagrama de funcionamiento de una aplicación web móvil

Nota. Tomado de *Principales tipos de aplicaciones móviles: ventajas, desventajas y ejemplos* (Bernal González, 2021)

Ventajas:

- Costos de desarrollo y tiempo son bajos
- Código de programación reutilizable
- No necesita instalación

Desventajas:

- Se requiere de conexión a internet
- No puede subirse a una tienda de aplicaciones móvil y pierde visibilidad
- Su acceso al hardware es parcial

Tabla 1.- Comparación de tipos de aplicaciones móviles

Tipos de aplicaciones web/ Características	Aplicaciones Nativas	Aplicaciones Híbridas	Aplicaciones Web
Coste de desarrollo	Alto	Medio	Bajo
Tiempo de desarrollo	Alto	Medio	Bajo
Multiplataforma	No	Sí	Sí
Rendimiento	Alto	Medio	Bajo
Tiendas de aplicaciones	Sí	Sí	No
Acceso al dispositivo	Completo	Alto/ Completo	Parcial
Conexión ⁸⁹ a internet	No siempre necesario	No siempre necesario	Siempre
Espacio en el dispositivo	Sí	Sí	No

Nota. Nota. Tomado de *Principales tipos de aplicaciones móviles: ventajas, desventajas y ejemplos* (Bernal González, 2021)

Tipos de aplicaciones móviles

Existen varios tipos de aplicaciones móviles, a continuación, se describen las principales (Herazo, 2020)

- **Aplicaciones de juegos:** Es la categoría más popular, algunas de estas aplicaciones requieren grandes prestaciones en los teléfonos y las empresas cada vez apuestan más a este tipo de aplicaciones.
- **Aplicaciones de productividad:** También están catalogadas como empresariales, puede realizar bastantes tareas complejas, aumentan la productividad y disminuyen los gastos.
- **Aplicaciones educativas:** Ayudan a adquirir habilidades y conocimientos. También suelen usar los profesores para mejorar su proceso de enseñanza.

- **Aplicaciones de estilo de vida:** Esta categoría comprende compras, entrenamiento, citas, dieta, etc.
- **Aplicaciones de comercio:** Estas aplicaciones ofrecen un acceso fácil de sus productos a sus clientes y fáciles métodos de pago.
- **Aplicaciones de entretenimiento:** Esta categoría permite transmitir contenido, chatear, ver contenido en línea, servicios de streaming, música, etc.
- **Aplicaciones de utilidad:** Por lo general tienen tiempos de sesión cortos, ejemplo: mapas, lector de código de barras, calculadora.
- **Aplicaciones de viaje:** Permite viajar a los usuarios de una manera más fácil y segura, recomiendan lugares, también permiten servicio de taxi.

Sistema Operativo móvil

El sistema operativo móvil es un conjunto de programas y ordenes capaces de ejecutarse sobre el hardware del dispositivo. (“Sistema operativo del móvil: qué es, cómo funciona y cuál ...”) Los teléfonos móviles por lo general están conectados a la red y por eso el sistema operativo está enfocado en la conectividad inalámbrica. Su propósito es administrar tareas y recursos del dispositivo, establece las tareas de software en base al hardware.

Capas de un sistema operativo móvil

Es importante saber cómo está formado el sistema operativo, está formado por capas que facilitan el desarrollo de aplicaciones ya que permite la comunicación con capas inferiores, y de esta manera puedan interactuar los componentes del hardware con la aplicación.

Cada capa realiza una función, se detalla a continuación. (Farooq, 2018)

- **Kernel:** Conocido también como núcleo, gestiona todos los procesos por medio del acceso a los distintos elementos de hardware del dispositivo. Es el responsable de ofrecer los servicios a los drivers o controladores de hardware. El sistema operativo de Android está basado en el kernel de Linux y iOS en un kernel heredado de Unix.
- **Middleware:** Es un grupo de módulos que permiten ejecutar servicios, que posibilitan que las aplicaciones diseñadas puedan ser ejecutadas en su plataforma. Este proceso es transparente al usuario.
- **Entorno de gestión de aplicaciones:** Contiene un gestor de aplicaciones y grupo de interfaces programables abiertas que comprende los elementos necesarios para el desarrollo de software.
- **Interfaz de usuario:** permite la comunicación del usuario con el diseño visual de la aplicación, permite ingresar las instrucciones para que el dispositivo las resuelva. La interfaz de usuario incluye los elementos gráficos que hacen el uso posible del dispositivo, sin esta capa no sería posible utilizar el teléfono.

En la actualidad, en la mayoría de los smartphones, la interfaz gráfica es mediante el uso de una pantalla táctil.

Sistemas operativos existentes actualmente

Según la página web de (Statcounter, s.f.) en todo el mundo, en base a la cuota de mercado:

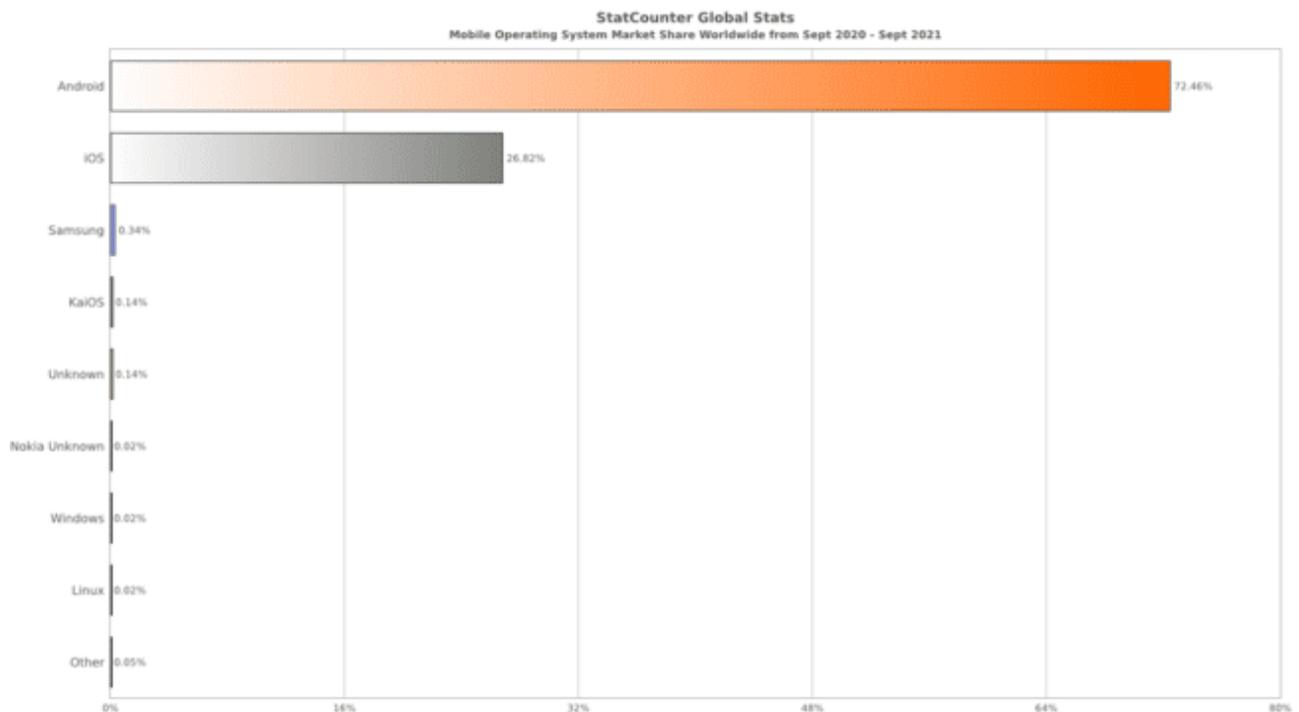


Ilustración 4.- Diagrama de barras de la cuota de mercado de sistemas operativos móviles

Nota. En Ecuador también lidera el SO Android con el 87.6% y iOS con el 11.89% según datos registrados de septiembre 2020-septiembre 2021.

Desarrollo de aplicaciones móviles

Son los procedimientos y procesos para crear software para dispositivos móviles inalámbricos como tabletas o teléfonos inteligentes.

Herramientas

- **Visual Studio Code:** es un editor de texto y código fuente que puede trabajar con algunos lenguajes de programación mediante la gestión de extensiones que permite personalizar y potenciar la herramienta. También permite administrar los atajos de teclado y refactorizar el código. VSC es de licencia Open Source. (Visual Studio Code, 2021)
- **Android Studio:** Es un IDE para el desarrollo de aplicaciones de la plataforma Android, la cual es basada en IntelliJ IDEA permite a los desarrolladores aplicaciones mediante sus herramientas y sus servicios, esto incluye código, diseño de interfaz. (android,

2022)

Android Studio cuenta con un emulador que para probar la funcionalidad de las aplicaciones creadas.

- **GitHub:** Según (Bustos, 2023) y (Fernández, 2019) es la plataforma más popular para alojar código de proyectos abiertos y colaborativos de cualquier desarrollador, esta herramienta permite que los proyectos puedan ser descargados por cualquier usuario, también ayuda a mejorar el producto a partir de ramificaciones a partir de él y permite a los desarrolladores poder administrar su proyecto y tener control de sus versiones. GitHub permite subir proyectos gratis siempre y cuando sean de código abierto.

- **Udemy:** es una plataforma de aprendizaje en línea, fundada en el 2010, con el objetivo de mejorar la vida de las personas mediante cursos bajo demanda. (Udemy, 2021)

Lenguajes de programación

- **Dart:** Es un lenguaje desarrollado por Google, es de licencia Open Source con el propósito de utilizar un lenguaje orientado a objetos. Su primera versión estable fue en 2011, desde entonces el lenguaje de programación ha cambiado en lenguaje, como en objetivos principales, en 2021 llegó la versión 2.0 paso de ser opcional a estático. (Dart, 2021)

Dart también fue creado con el objetivo de corregir problemas que existían en JavaScript, Hoy en día Flutter es el principal objetivo del lenguaje.

- **Php:** Este lenguaje está destinado a la creación de páginas web, permitiendo una conexión fiable entre los servidores y la interfaz de usuario, debido a su licencia Open Source y constante actualización. (Group, 2022)

- **Git:** Es un sistema de control de versiones, el más popular actualmente de arquitectura distribuida es un proyecto de mantenimiento activo, creado en 2005 por Linus Torvalds, el creador del kernel del sistema operativo Linux. Es una herramienta para guardar la histórica revisión de versiones de un proyecto de software. (Microsoft, 2021)

Framework

- **Flutter:** Es una plataforma de tecnología híbrida, gratuita y de código abierto, que apareció en el 2018, es desarrollada por Google y basada en el lenguaje de programación orientado a objetos Dart.

Esta herramienta permite realizar aplicaciones nativas para móvil, web y escritorio desde la misma base de código. Se basa en la creación de widgets que permiten desarrollar una aplicación rápidamente para aplicaciones diferentes, reduciendo el número de pruebas.

Flutter trabaja con Material Design de Google y Cupertino de Apple, lo que permite que las interfaces de usuario sean iguales a las de las aplicaciones de la misma empresa, por ende, una experiencia de usuario óptima. (DocsFlutter, 2022)

Contiene una característica "hot-reload" o cambios en caliente, la cual permite ver las modificaciones en tiempo real.

Comparativa de aplicaciones similares dirigidas a usuarios motociclistas

Waze: es una versión de la popular aplicación de navegación Waze, específicamente diseñada para conductores de motocicletas. Ofrece rutas personalizadas para vehículos de dos ruedas y muestra información en tiempo real sobre el tráfico, así como alertas de peligros en la carretera. También permite a los usuarios reportar problemas en la ruta, como accidentes o cierres de carreteras. (Waze, s.f.)

Rever: es una aplicación que combina la navegación con una red social motera, permitiendo a los usuarios planificar rutas, compartir sus rutas y conectar con otros conductores de motocicletas. También ofrece información sobre eventos relacionados con las motocicletas y proporciona una lista de lugares recomendados para visitar en las rutas. (Rever, s.f.)

4.2. Tecnologías de desarrollo

Flutter

Es un framework de código abierto de Google se basa en el lenguaje Dart, para realizar aplicaciones compiladas nativamente para móvil y web desde un mismo código fuente, con la ventaja de que el rendimiento y la UX es igual a las aplicaciones nativas.

Se destaca por su calidad nativa, experiencia de usuario ya que incluye Material Design de Google y Cupertino de Apple, lo que permite una experiencia óptima e idéntica al del software del dispositivo móvil. (DocsFlutter, 2022)

También una de las mayores ventajas con respecto a otros framework es el hot reload o cambios en caliente los cuales permiten visualizar los cambios en menos de un segundo.

Es compatible con Mac, Windows y Linux

Flutter generalidades

Según la página oficial de (Flutter, 2022) es un SDK móvil de código abierto desarrollado por Google para la creación de aplicaciones de manera nativa, tanto para iOS como Android, mediante su motor el lenguaje Dart.

Dart se creó con el propósito de hacer que el desarrollo sea más cómodo y veloz posible, la VM de Dart y su compilación Just-In-Time permiten que los cambios se visualicen enseguida al momento de grabar los cambios.

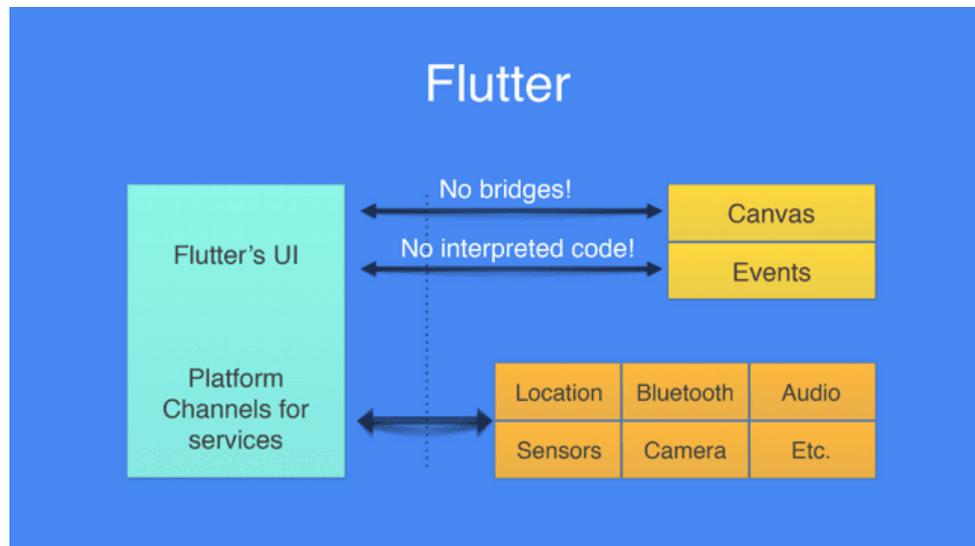


Ilustración 5.- Que es Flutter

Nota. Tomado de ¿Es Flutter el framework del futuro? (Molina, 2020)

Según (Molina, 2020) Software Development Kit es un conjunto de herramientas que permite usar como base de código, que son utilizados posteriormente compiladas.

Skia es el motor gráfico utilizado por Flutter que renderiza los objetos gráficos en 2D. Flutter está desarrollado en C++ y sus widgets en Dart.

Un Widget es un elemento gráfico que puede ser una lista, un botón, una imagen o también un widget puede estar conformado por más widgets.

Estructura de un proyecto

La estructura del proyecto consta de cuatro subcarpetas principales que contienen diferentes tipos de archivos: modelos, puntos de vista, servicios y comandos

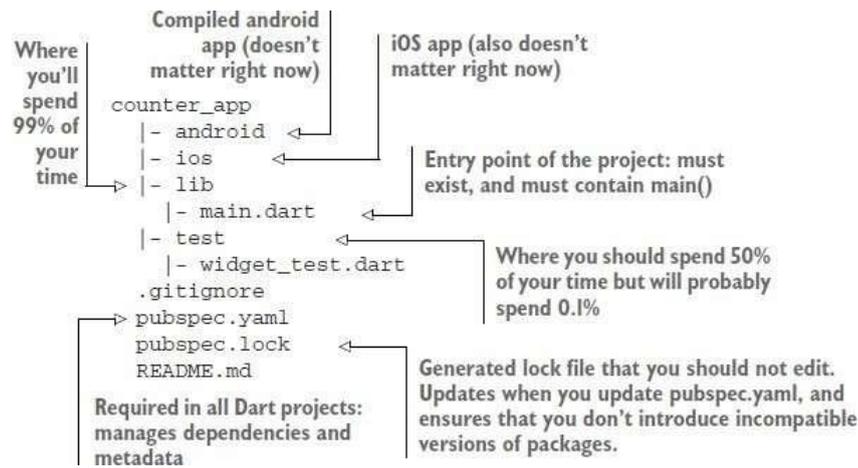


Ilustración 6.- Flutter project structure

Nota. Tomado de Flutter in Actions (Windmill, 2019)

Según (Windmill, 2019), la gran parte las carpetas de la estructura de un proyecto no son utilizadas de manera correcta, según la ilustración la carpeta “lib” es el directorio usado mayormente aquí se aloja el desarrollo lo visual como también lógica. Otro directorio importante es “test”, donde se realizan las pruebas al código para verificar que funcione bien.

En los programas de Dart, es imprescindible la función main para ejecutar el proyecto, tal como la mayoría de los lenguajes ya conocidos por ejemplo Java. Este método envuelve a otro llamado runApp lo que le permite depurar a la aplicación dependiendo de las acciones envueltas dentro de él, como son widgets. (Dart, 2021)

4.3. Widgets

Los widgets están inspirados en el moderno framework de React, se construyen utilizando un framework con el propósito de que el desarrollador construya su UI de widgets. Cuando el estado de un widget cambia, el mismo reconstruye su descripción.

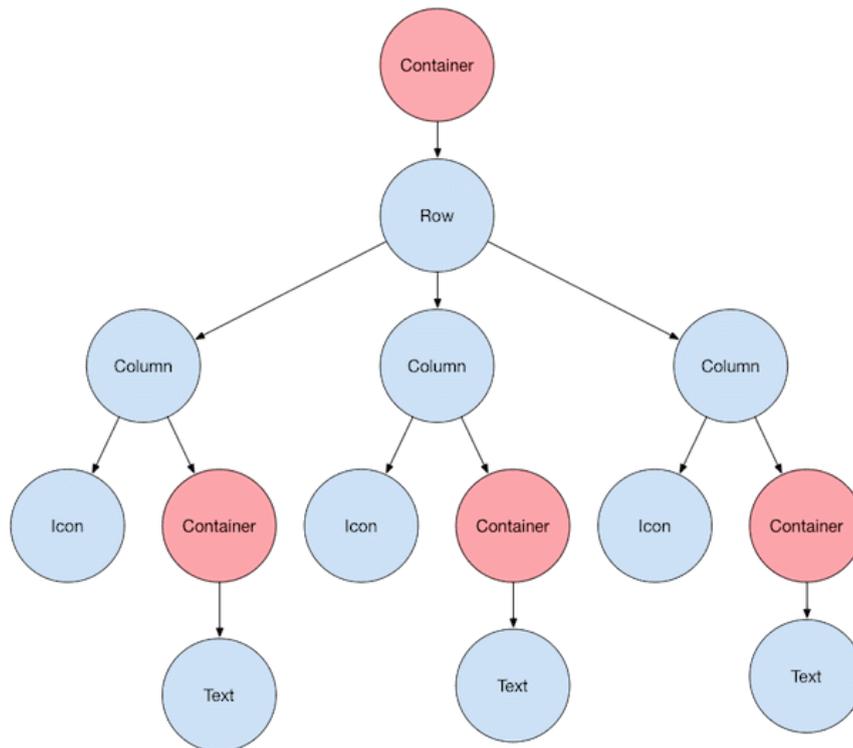


Ilustración 7.- Ejemplo de un árbol de widgets

Nota. Tomado de *Layouts en Flutter*. (DocsFlutter, 2022)

En la ilustración 7 se puede observar un ejemplo de cómo está establecida una jerarquía en un árbol de widgets, donde cada nodo se extiende de un nodo padre. Los widgets más básicos son Text, Row, Column, Stack, Container.

El widget Material Design necesita pertenecer a un MaterialApp para heredar los datos del tema, por lo tanto, se ejecuta la aplicación desde un MaterialApp que construye widgets en la raíz de la aplicación. (Flutter, 2022)

Tipos de Widgets

Stateless Widget

Según (DocsFlutter, 2022), es un widget estático que se construye con un solo estado, con una sola configuración inicial, es un widget que no cambiara, es inmutable lo que significa que no cambiara durante su configuración inicial, carece de métodos que le permitan

actualizarse por sí mismo.

Stateful Widget

Es un widget que puede cambiar dinámicamente, tiene un estado interno que lo permite administrarse a sí mismo. Es inmutable de igual manera que el Stateless Widget y tiene un objeto llamado “State”, el cual tiene la característica de ser mutable, lo cual permite que no se cambie el estado, incluso cuando Flutter redibuje.

`setState ()`: Esta función le dirá al framework que un objeto ha cambiado en el estado y se permitirá reconstruirse.

`initState ()`: Esta función es iniciada automáticamente cuando un widget se construye por tanto en este método se puede inicializar con cualquier dato necesario considerándose como el estado inicial del widget antes de dibujar en pantalla.

4.4. Firebase

Es una plataforma desarrollada por Google, basado en un contenedor en el cual se comparte los recursos y servicios aprovisionados de un Back-End para un proyecto, facilitando una mayor compatibilidad con el desarrollo de aplicaciones multiplataformas como entre Android, iOS, Web y Unity. Cuenta con una versión gratuita y pagada dependiendo del consumo de servicios pueden ser tarifados por hora. (Firebase, 2022)

Entre los recursos que comparten se encuentran la autenticación, base de datos en tiempo real, alojamiento de Firebase, almacenamiento y funciones en la nube. (Google Firebase, s.f.)

Dentro de las principales características de Firebase, para el uso de este proyecto se utilizará Cloud Messaging, mensajería multiplataforma que permite enviar mensajes de notificación o mensajes de datos y sin costo, con tres formas diferentes, dispositivos individuales, a grupos de dispositivos o a dispositivos suscritos a temas.

4.5. Servicios web – Api Rest

Un API REST son servicios web mediante una interfaz de programación que utiliza la arquitectura REST y permite interactuar con la web mediante una URL a la cual se puede acceder mediante protocolos de internet principalmente el protocolo HTTP. (RedHat, 2020)

Una API REST es una API que se ajusta a los principios de diseño de REST (Transferencia de Estado Representacional), un estilo de arquitectura que se utiliza para crear servicios web. (IBM Cloud, 2021).

Las APIs, o Interfaces de Programación de Aplicaciones, son esenciales en el diseño y desarrollo de software y aplicaciones. Estas interfaces permiten la comunicación entre diferentes aplicaciones y sistemas, proporcionando una forma estandarizada de intercambio de información.

Además de facilitar la integración de sistemas y aplicaciones, las APIs también pueden ser utilizadas para la creación de nuevas funcionalidades y servicios, lo que puede ser beneficioso para empresas y desarrolladores de software.

En general, las APIs son una herramienta valiosa para cualquier desarrollador de software, ya que les permiten integrar y aprovechar funcionalidades y servicios de terceros en sus propias aplicaciones, lo que puede ahorrar tiempo y recursos.

4.6. Servidores alojados en la nube

Un servidor alojado en la nube es un servidor que se ejecuta en una infraestructura de nube pública o privada. Los servidores alojados en la nube ofrecen una serie de ventajas sobre los servidores tradicionales, como la escalabilidad y la facilidad de uso. (Greg, s.f.)

Además de las ventajas mencionadas, los servidores alojados en la nube también ofrecen una mayor disponibilidad y redundancia, lo que significa que los datos y aplicaciones alojados en ellos están disponibles las 24 horas del día, los 7 días de la semana, y pueden ser recuperados en caso de fallos o desastres.

Digital Ocean es uno de los proveedores líderes de servicios de alojamiento en la nube, con una plataforma de alta calidad y fácil de usar que ofrece una amplia variedad de opciones de configuración y personalización. Su modelo de precios asequible y su enfoque en la simplicidad y la escalabilidad lo han convertido en una opción popular entre los desarrolladores y las empresas que buscan un alojamiento en la nube confiable y de alta calidad. (Digital Ocean, 2023)

Además de los servidores virtuales privados (VPS), Digital Ocean también ofrece una variedad de otras soluciones de alojamiento en la nube, incluyendo almacenamiento en la nube, redes privadas virtuales (VPN) y equilibradores de carga. También cuenta con una amplia comunidad de usuarios y desarrolladores que ofrecen soporte y recursos para ayudar a los usuarios a aprovechar al máximo su plataforma de alojamiento en la nube.

5. MARCO METODOLÓGICO

5.1. Entorno de desarrollo

Para la creación de proyectos Flutter, se eligió el sistema operativo según se requiera compilar el dispositivo, si se requiere compilar en un dispositivo Android, se puede desde un sistema operativo Windows como también de Linux, en caso de compilar para dispositivos iOS es necesario un sistema operativo macOS.

A continuación, se detalla la configuración:

- Android SDK

Para la instalación del SDK es necesario descargar Android Studio desde la página oficial [android.com/studio](https://developer.android.com/studio), para el desarrollo de este proyecto se instaló la versión 2021.3.1 e instalar su ejecutable.

- Flutter SDK

Para realizar la instalación del SDK de Flutter se lo descargó de su página oficial <https://docs.flutter.dev/get-started/install/windows>, para el proyecto se ha instalado la versión 3.0.5.

Una vez descargado se realizó la copiar la carpeta “Flutter” a la carpeta raíz y por último la creación de las variables de entorno apuntando a “../Flutter/bin”.

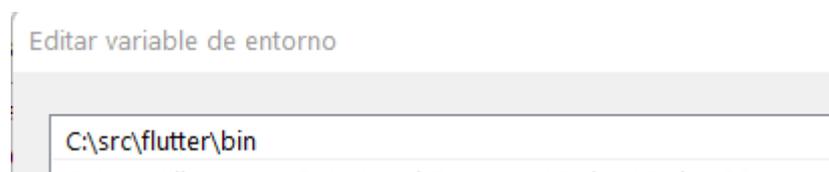


Ilustración 8.- Configuración de variable de entorno

5.1.1. Configurar Flutter

Para comprobar la instalación desde una Terminal se ejecutó el comando “flutter doctor” el cual nos dirá si existe alguna otra dependencia para completar la configuración.

```
PS C:\Users\jbena> flutter doctor

A new version of Flutter is available!

To update to the latest version, run "flutter upgrade".

Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.0.5, on Microsoft Windows [Versi#n 10.0.22000.978], locale es-US)
[✓] Android toolchain - develop for Android devices (Android SDK version 31.0.0)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop for Windows (Visual Studio Build Tools 2019 16.11.14)
[✓] Android Studio (version 2020.3)
[✓] IntelliJ IDEA Ultimate Edition (version 2019.2)
[✓] IntelliJ IDEA Ultimate Edition (version 2022.1)
[✓] VS Code (version 1.71.2)
[✓] Connected device (3 available)
[✓] HTTP Host Availability
```

Ilustración 9.- Flutter Doctor

Como se ve en la ilustración 6, se cumple con todos los requerimientos para el funcionamiento de Flutter en el equipo, sin embargo, se puede actualizar Flutter a su versión más estable y reciente.

5.1.2. IDE de desarrollo.

Entre los editores de código más populares en el lenguaje de desarrollo Flutter se encuentran Android Studio y Visual Studio Code, Se ha escogido VSC por su facilidad de navegación, los comandos y las extensiones que permiten aumentar la productividad al momento de crear código.

Las extensiones son complementos de Dart Code, Flutter y Dart son los complementos utilizados como se pueden visualizar en las siguientes ilustraciones.



Ilustración 10.- IDE de desarrollo, extensión Flutter



Ilustración 11.- IDE de desarrollo, extensión Dart

La siguiente extensión facilita la creación de código y por ello aumenta la velocidad.



Ilustración 12.- IDE de desarrollo, extensión Awesome Flutter Snippets

5.1.3. Gestor de Versiones

Gestor de versiones o control de código fuente, permite rastrear y controlar cambios en el software a lo largo del tiempo, lo que permite recuperar una versión en un futuro.

Se ha usado el sistema de control de versiones de git con GitHub, el mismo es distribuido, por su facilidad para mantener una gran cantidad de código, de programadores, por su seguridad ya que todas las estructuras están cifradas mediante SHA1 y es open Source.

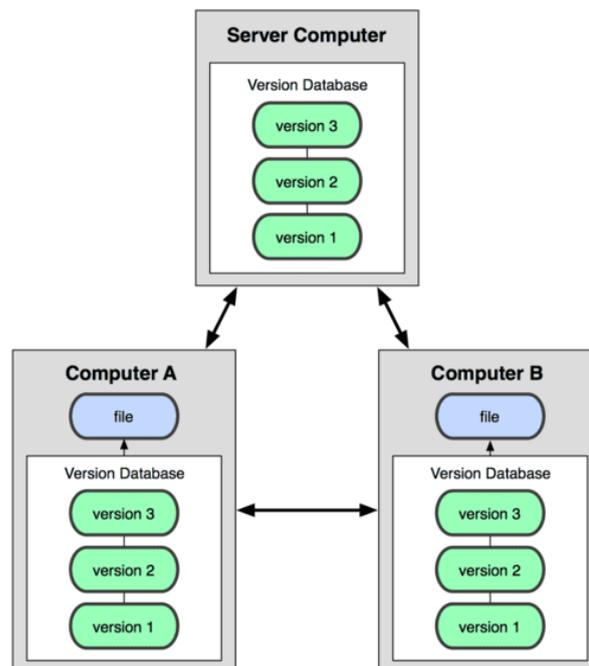


Ilustración 13.- Diagrama de sistemas de control distribuidas.

Nota. Tomado de *Tutorial de Gestión de copias de seguridad de datos* (MxAbc, 2018)

5.2. Creación de la aplicación.

El primer paso para su configuración es indicar el nombre de usuario desde la terminal con el siguiente comando

```
git config -global user.name "nombre"
```

Después se configura el email con el comando

```
git config -global user.email tuemail@mail.com
```

Para visualizar la configuración

```
git config -list
```

Una vez configurada nuestra información se descargó el proyecto mediante

```
git clone https://github.com/acens/xxxxxxx.git
```

El mismo que se podrá escribir o copiar desde la ventana del repositorio de GitHub.

Para visualizar los cambios en el dispositivo se lo puede hacer mediante el comando

```
git status
```

Para añadir cambios mediante el comando

```
git add ruta_archivo
```

Para enviar los cambios se deberá ejecutar

```
git commit -m "descripción del cambio"
```

Para subir los cambios al repositorio y pueda acceder cualquier desarrollador con el comando

```
git push
```

5.3. Levantamiento de requerimientos.

Es la primera etapa al momento de desarrollar software, en este paso se aplicó técnicas de levantamiento de requerimientos para cumplir con tiempos de desarrollo y con los requisitos presentados.

5.3.1. Identificación de stakeholders.

Los stakeholders son grupos de interés que se forman parte de las decisiones de la entidad.

Usuarios

Todas las personas que usarán la aplicación, especialmente los usuarios a los que esta dirigida, los usuarios motociclistas quienes sacarán provecho del producto.

Desarrolladores

Para la creación de esta aplicación estarán a cargo dos desarrolladores, los cuales se proponen a terminar con los objetivos planteados.

5.3.2. Requisitos Funcionales

Los siguientes requerimientos definen que es lo que hace el sistema como tal, es decir cuáles son las funcionalidades que cumple el mismo para con el usuario final. Tras el proceso de levantamiento de los requerimientos antes mencionados quedo definido que la aplicación cuenta con las siguientes características, para revisar el listado y funcionamiento de los requisitos a más detalles se incluyen en los anexos 1 y 3, [encuestas](#), manual de [usuario](#).

1. Registro de los usuarios
2. Control de sesión de los usuarios
3. Sistema de mensajería instantánea.
4. Control de agregación de amigos.
5. Creación de publicaciones.
6. Interacción de publicaciones.
7. Control de perfil.
8. Control de motocicleta.
9. Control de rutas.
10. Control de mantenimiento.

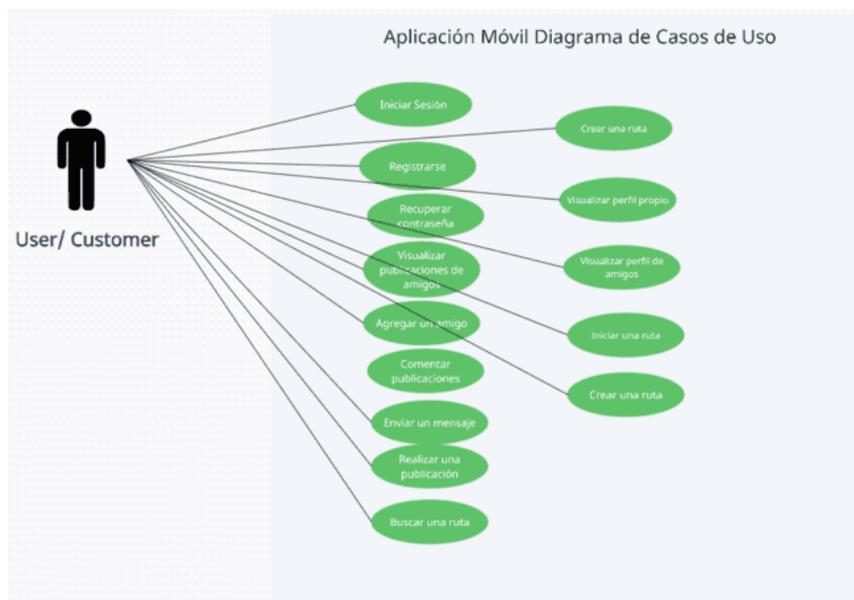


Ilustración 14.- Diagrama de casos de uso interacción del usuario con la aplicación

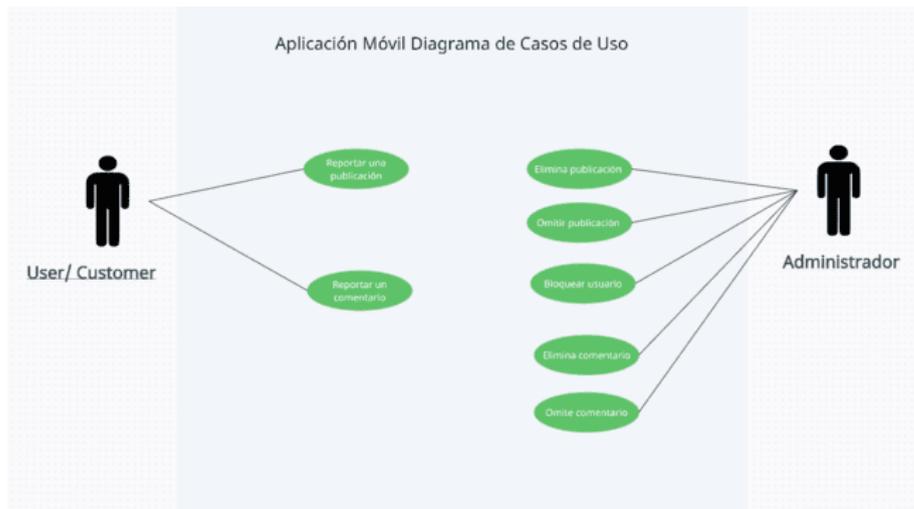


Ilustración 15.- Casos de uso interacción de reportes y bloqueos Usuario Administrador

5.3.3. Requisitos no funcionales

En los requisitos no funcionales definimos aquellas funcionalidades con respecto a la navegación del usuario con la aplicación e interfaz de usuario.

Se ha escogido el framework de Flutter ya que permite desarrollar la aplicación multiplataforma en un entorno nativo, lo cual permitió desplegar el proyecto en dos sistemas operativos (Android, iOS).

En cuanto a los colores de la aplicación se ha escogido el color dark u oscuro, que le permite al usuario motociclista ver en el día, pero más que todo ayuda para una mejor visualización para la noche.

También se ha escogido la navegación con una barra inferior, ya que es intuitiva para el usuario y permite su rápido manejo sin volver hacia atrás las ventanas.

5.4. Definición de la arquitectura del sistema.

Para la implementación de la aplicación se ha escogido una arquitectura distribuida de N capas, ya que dicha arquitectura cumple el propósito del manejo de diferentes servidores para el desarrollo, tal como se puede ver en la ilustración 14.

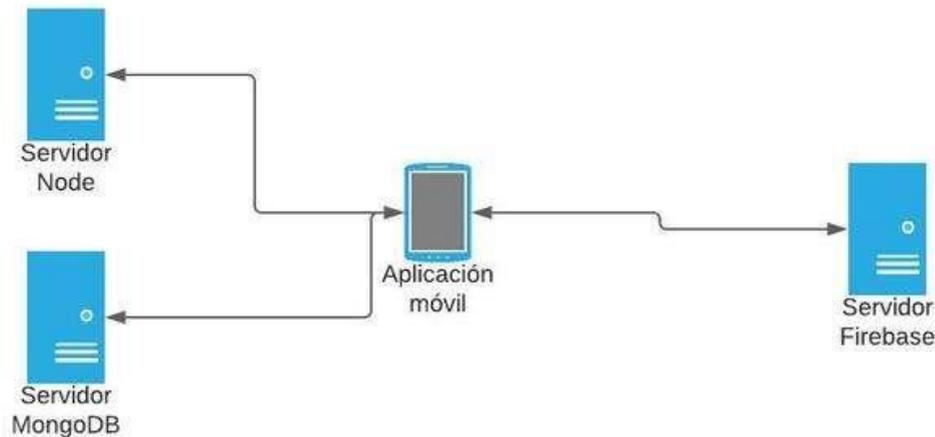


Ilustración 16.- Arquitectura del sistema.

Como se puede ver en la arquitectura de esta aplicación se han utilizado un servidor de aplicación Back-End Node almacenado en pm2 en un dominio público llamado Digital Ocean el cual cumple la función de conectarse con el servidor de base de datos no relacional MongoDB y la aplicación móvil en este caso ejecutado de manera local.

5.4.1. Servidor NodeJS

En el servidor de NodeJS esta creado mediante servicios REST, el cual envía y recibe información estática sobre los usuarios, publicaciones, perfil y mantenimiento.

Para el proyecto este servidor se ha alojado en Digital Ocean, y se ha contratado el paquete de pago de 1Gb de memoria, 25Gb de Disco Duro de estado sólido y transferencia de hasta 1Tb. Se ha escogido las características puesto que cumple con el tratamiento de datos que necesita el proyecto.

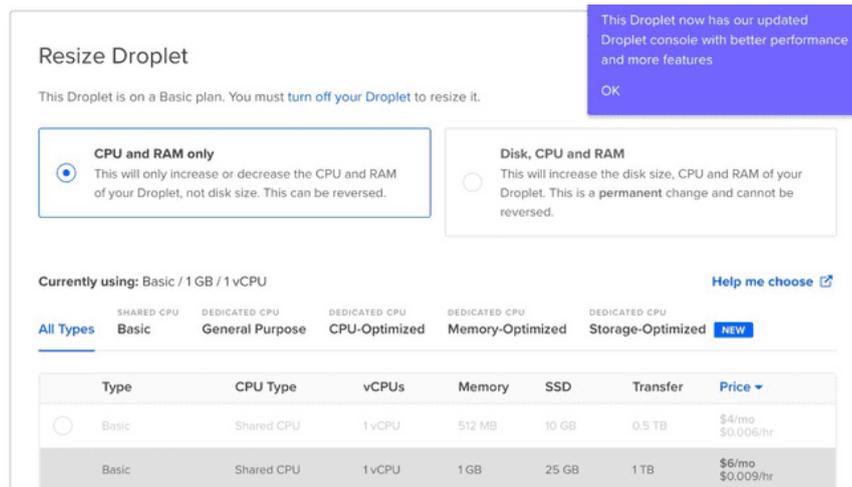


Ilustración 17.- Arquitectura del sistema.

5.4.2. Firebase

El servidor de Firebase se ha escogido puesto que se requería cumplir con el servicio de notificaciones o Firebase Cloud Messaging es un servicio gratuito, con un bajo consumo de batería, recursos entre el servidor y el celular.

5.4.3. Sockets

Se han utilizado sockets para satisfacer la necesidad del chat en tiempo real mediante comunicación bidireccional entre cliente y servidor.

Para la implementación de sockets se ha utilizado Socket.IO que permite que el servidor se integre con el servidor HTTPS y una biblioteca que se carga en la parte del frontend socket.io-client.

Para instalar el módulo se debe hacer con el siguiente comando

npm install socket.io

Para la parte del backend se realizó las primeras pruebas mediante el siguiente código para comprobar el funcionamiento del componente:

```

const express = require('express');
const app = express();
const http = require('http');
const server = http.createServer(app);
const { Server } = require("socket.io");
const io = new Server(server);

app.get('/', (req, res) => {
  res.sendFile(__dirname + '/index.html');
});

io.on('connection', (socket) => {
  console.log('a user connected');
});

server.listen(3000, () => {
  console.log('listening on *:3000');
});

```

Ilustración 18.- Test de conexión y funcionamiento de sockets

Para la parte del frontend se recibió el servicio GET /socket.io/socket.io.js y con eso se cumplió con su funcionamiento.

Cada socket también permite ver en caso de que un usuario se ha desconectado mediante las siguientes líneas

```

io.on('connection', (socket) => {
  console.log('a user connected');
  socket.on('disconnect', () => {
    console.log('user disconnected');
  });
});

```

Ilustración 19.- Código que permite ver si un usuario fue desconectado

El propósito del uso de los sockets es que se pueden enviar y recibir eventos con la información que se desee en formato JSON

5.5. Creación de la aplicación.

Después de haber analizado el por qué el framework del Flutter es el conveniente para el desarrollo del proyecto, es estableciendo como primera ventaja la ejecución en multiplataforma de forma nativa, con previa descarga y configuración, mediante el IDE de desarrollo y la extensión de Flutter se creó un proyecto en blanco con todos los archivos necesarios para su ejecución en el sistema operativo Android o iOS.

En caso de no crear mediante el IDE de desarrollo se realizó la creación mediante consola, y de la misma manera el framework creara los archivos correspondientes para su ejecución de la siguiente manera:

```
flutter create application_mobile_motorcycle
```

Con la siguiente instrucción se crea el directorio de la aplicación la cual contendrá los directorios, archivos, dependencias y configuraciones para los sistemas operativos ya mencionados, al momento de crear se puede ejecutar una simple demostración de unas clases StateFull Widget que funciona como un contador, una vez realizado esto simplemente eliminar el código inicial y empezar con su desarrollo, para revisar a más detalle la instalación se incluye en el anexo 2 manual de [instalación](#).

También se puede seleccionar el dispositivo en el que se desee ejecutar en caso de ser un emulador y deberá estar descargado e instalado mediante el IDE de Android Studio, o simplemente se podrá ejecutar la aplicación en un navegador (no recomendable). Para poder visualizar los cambios en tiempo real con ayuda de la característica Hot Reload de Flutter es necesario que sea un dispositivo físico o lógico, para lo una vez conectado se deberá ejecutar el siguiente comando:

```
flutter devices
```

Una vez elegido la lista de servicios disponibles el dispositivo o dispositivos en los que se desea ejecutar la aplicación, se ejecutará el proyecto con la siguiente instrucción:

```
flutter run
```

5.5.1. Librerías externas

Para el desarrollo de una aplicación Flutter es común usar librerías que puede ser propios del framework o realizados por la comunidad y validados por Flutter, entre ellos paquetes de Firebase, manejadores de estados, entre otros.

En la estructura del proyecto existe el archivo `pubspec.yaml`, el cual es el archivo de configuración de Flutter, donde se encuentra especificado los paquetes con sus respectivas versiones a ser tratados entre los usados en el proyecto son los siguientes:

- Administración de estados
 - bloc: ^8.0.1
 - flutter_bloc: ^8.0.0
 - provider: ^6.0.3
- Administración de Firebase
 - firebase_core: ^1.20.0
 - firebase_messaging: ^12.0.1
- Administración de mapas
 - geolocator: ^7.0.3
 - google_maps_flutter: ^2.1.1
 - google_polyline_algorithm: ^3.1.0
 - mapbox_gl: ^0.10.0
- Administración de preferencias
 - permission_handler: ^8.3.0
 - shared_preferences: ^2.0.7
- Peticiones HTTP
 - dio: ^4.0.2
 - http: ^0.13.4

- url_launcher: ^6.1.5
- Socket
 - socket_io_client: ^2.0.0
- UI
 - font_awesome_flutter: ^10.1.0
 - animate_do: ^2.1.0
 - cupertino_icons: ^1.0.2
 - material_color_utilities: ^0.1.3
 - equatable: ^2.0.3
 - flutter_dotenv: ^5.0.2
 - image_picker: ^0.8.5+3
 - intl: ^0.17.0

5.5.2. UI/UX – Tema y diseños.

El diseño se centra en mejorar la experiencia del usuario y su facilidad para navegar entre funcionalidades de manera útil, atractiva, visual y que satisfaga las necesidades del usuario.

Para empezar con el diseño se creó los mockups o vistas para tener una idea de cómo funcionará y como estará estructurada visualmente la aplicación con la ayuda de la aplicación Balsamiq, se crearon los diseños iniciales, los cuales cambiaron en base a mejorar la experiencia del usuario.



Ilustración 20.- UI – Diseño del ingreso y registro de la aplicación



Ilustración 21.- UI – Pantallas de feed de publicaciones y chats

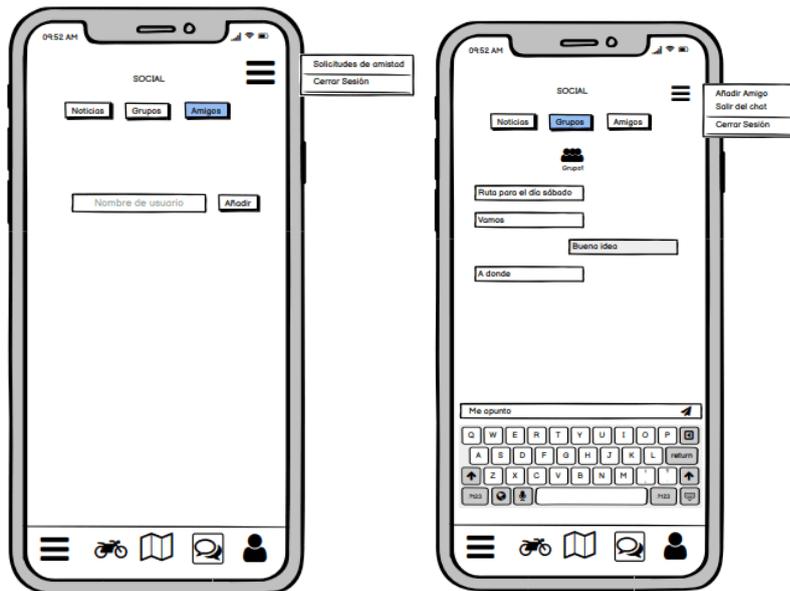


Ilustración 22.- UI – Agregar amigos y diseño de la pantalla de un chat

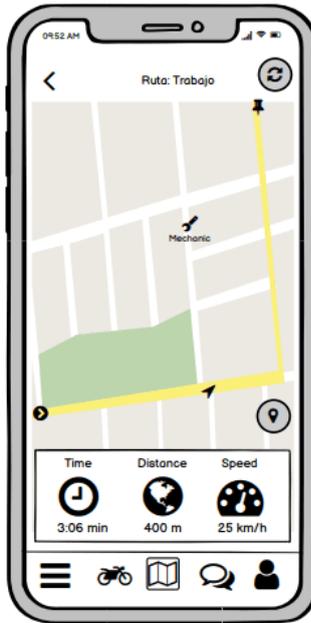


Ilustración 23.- UI – Diseño de navegación a una ruta

Después de haber realizado el prototipo, se definió la línea gráfica correspondientes a la aplicación con el fin de mejorar el manejo la experiencia del usuario se acoge al siguiente diseño.

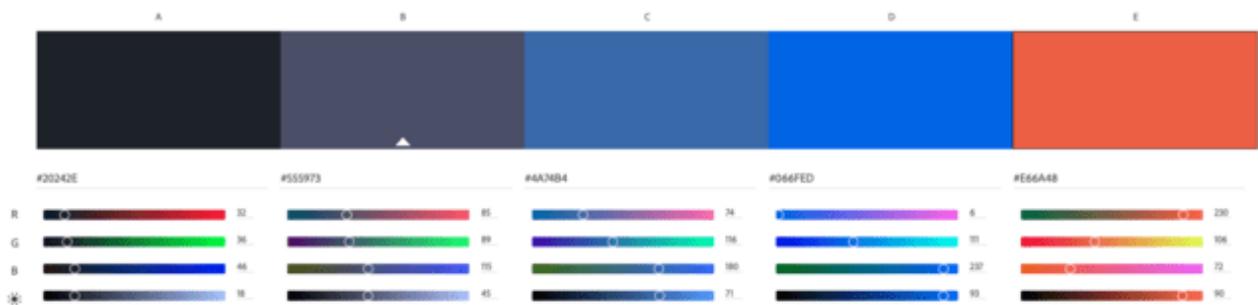


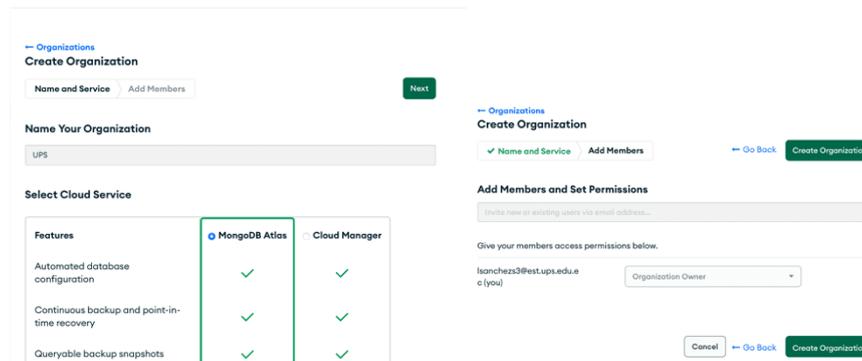
Ilustración 24.- Paleta de colores

5.6. Creación de la base de datos.

5.6.1. Almacenamiento remoto en MongoDB Atlas Cloud

Para la creación de la aplicación se optó por el almacenamiento en la nube de MongoDB Atlas Cloud que mediante un sistema de base de datos no relacional facilita el trabajo y la carga de datos a través de una API que permite ampliar el Back-End de la nube a la aplicación móvil.

Para empezar, se creó una organización, al cual va seguido de un proyecto, además en cada uno se define el acceso y permisos de cada miembro, como se visualiza en las siguientes ilustraciones.

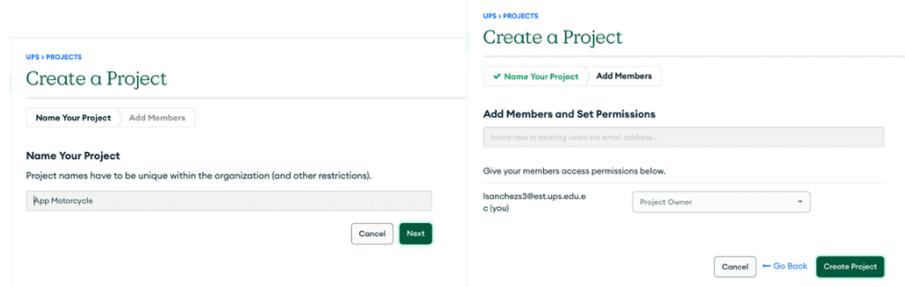


The screenshot shows the 'Create Organization' interface in MongoDB Atlas. It is divided into two main sections. The left section, titled 'Name Your Organization', includes a text input field containing 'UPS' and a 'Next' button. Below this is a 'Select Cloud Service' section with a table of features:

Features	MongoDB Atlas	Cloud Manager
Automated database configuration	✓	✓
Continuous backup and point-in-time recovery	✓	✓
Queryable backup snapshots	✓	✓

The right section, titled 'Add Members and Set Permissions', includes a text input field for an email address, a dropdown menu for the role (set to 'Organization Owner'), and a 'Create Organization' button.

Ilustración 25.- Pantalla de creación de organización



The screenshot shows the 'Create a Project' interface in MongoDB Atlas. It is divided into two main sections. The left section, titled 'Name Your Project', includes a text input field containing 'Ppp Motorcycle' and a 'Next' button. The right section, titled 'Add Members and Set Permissions', includes a text input field for an email address, a dropdown menu for the role (set to 'Project Owner'), and a 'Create Project' button.

Ilustración 26.- Pantalla de creación de proyecto.

En las siguientes ilustraciones podemos visualizar, como se creó la base de datos en el entorno de Cloud, con opciones básicas de configuración para obtener la opción gratuita.

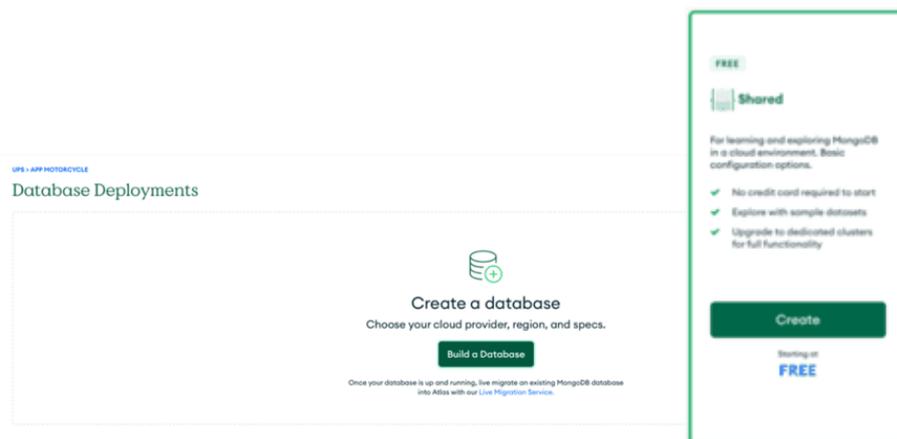


Ilustración 27.- Pantalla de creación de la base de datos.

Dentro de las opciones de configuración, se seleccionó el nivel de clúster donde se almacenará la base de datos, para este proyecto se optó por la opción de un clúster con recursos compartidos de memoria RAM y CPU, además con un tamaño de almacenamiento de 512 MB.

Cluster Tier M0 Sandbox (Shared RAM, 512 MB Storage)
 Encrypted

Hourly price is for a MongoDB replica set with 3 data bearing servers.

Shared Clusters for development environments and low-traffic applications

Tier	RAM	Storage	vCPU	Price
✓ M0 Sandbox	Shared	512 MB	Shared	Free forever

M0 clusters are best for getting started, and are not suitable for production environments.

500 max connections	Low network performance	100 max databases	500 max collections
---------------------	-------------------------	-------------------	---------------------

Ilustración 28.- Arquitectura del clúster

A continuación, se explicará el uso y estructura de cada módulo con su respectiva colección de datos creada para este proyecto.

Usuarios

En esta colección se almacenó los datos necesarios de cada usuario como nombre, email, password, identificación, teléfono, listado de token FCM de los dispositivos en el logeo para el envío de notificaciones push, opciones por default como en la foto de avatar y el estado de conexión.

```

1 {
2   "fcm": [
3     "eN0t1xxZT9SGxizkE6v1_M:APA91bFGL_jerMbH6Sfkl8LkZSGWk7Z7T1vdzw0RxyQfI
4   ],
5   "identity_id": "",
6   "name": "",
7   "username": "",
8   "email": "",
9   "mobile": "",
10  "password": "",
11  "avatar_user": "cbc5b93a-82ac-4903-9475-9503aa990070.png",
12  "state": false,
13  "__v": 0
14 }

```

Ilustración 29.- Colección usuario

Motocicleta

En esta colección se almacenó la información correspondiente a una motocicleta como la marca, modelo, capacidad de cilindraje, año, la foto de avatar por default, además el identificador del usuario.

```

1 {
2   "mark": "",
3   "avatar_motorcycle": "c1a98909-5e11-4161-b6bd-88e2e887c341.png",
4   "model": "",
5   "cylinder_capacity": ,
6   "year": "",
7   "created_by": {
8     "$oid": ""
9   },
10  "__v": 0
11 }

```

Ilustración 30.- Colección motocicleta

Mantenimiento

En esta colección se almacenó la información correspondiente al mantenimiento de una motocicleta como la distancia recorrida, distancia del próximo mantenimiento, la fecha del mantenimiento, además el identificador del usuario y el identificador de la motocicleta.

```

1 {
2   "motorcycle": {
3     "$oid": "631948325e7d61287db8b107"
4   },
5   "kmcurent": 13500,
6   "kmnext": 15000,
7   "date": {
8     "$date": {
9       "$numberLong": "1664055038582"
10    }
11  },
12  "created_by": {
13    "$oid": "6292aa888c1c64c57ffabfbf9"
14  },
15  "__v": 0
16 }

```

Ilustración 31.- Colección manteniendo de la motocicleta

Amigos

En esta colección se almacenó las solicitudes de amistad, el identificador del usuario para aquel que solicita o recibe la solicitud y por default el estado de la solicitud.

```

1 {
2   "requester": {
3     "$oid": "62afef7420f94356bee577d4"
4   },
5   "recipient": {
6     "$oid": "6293c01291d78aa6096b5063"
7   },
8   "status": 0,
9   "__v": 0

```

Ilustración 32.- Colección amigos

Publicación

En esta colección se almacenó la información correspondiente a una publicación como la descripción, el identificador del usuario que crea la publicación, la fecha de creación de la publicación, un listado de reacciones el cual contiene el identificador de cada usuario, además cuenta con dos tipos de contenido, una imagen o una ruta, la cual almacena el identificador de la ruta.

```

1 {
2   "description": "",
3   "created_by": {
4     "$oid": "6319318cc12716d8565a676d"
5   },
6   "image": "80745ffc-ff30-4492-afef-78b4d5d2da9c.jpg",
7   "likes": [
8     {
9       "$oid": "6292aa888c1c64c57ffabbf9"
10    }
11  ],
12  "created_At": {
13    "$date": {
14      "$numberLong": "1662605920446"
15    }
16  },
17  "__v": 0
18 }

```

Ilustración 33.- Colección publicaciones

Comentarios

En esta colección se almacenó la información correspondiente a un comentario, como la descripción del comentario, el identificador de la publicación, el identificador del usuario y la fecha de creación.

```

1 {
2   "comment": "",
3   "publication": {
4     "$oid": "6309d833dae6921f254f7aca"
5   },
6   "created_by": {
7     "$oid": "628ed677afacf80a80186594"
8   },
9   "created_At": {
10    "$date": {
11      "$numberLong": "1661599413518"
12    }
13  },
14  "__v": 0
15 }

```

Ilustración 34.- Colección comentarios

Rutas

En esta colección se almacenó la información correspondiente a una ruta como la descripción, el identificador del usuario y la fecha de creación.

```

1 {
2   "description": "",
3   "created_by": {
4     "$oid": "6292aa888c1c64c57ffabbf9"
5   },
6   "created_At": {
7     "$date": {
8       "$numberLong": "1664421054517"
9     }
10  },
11  "__v": 0
12 }

```

Ilustración 35.- Colección rutas

Coordenadas

En esta colección se almacenó la información correspondiente a las coordenadas de una ruta como la latitud y longitud, el identificador de la ruta y la fecha de creación.

```

1 {
2   "lat": "-2.865131",
3   "lng": "-78.990723",
4   "route": {
5     "$oid": "6328f1354d27c654b3d9af3d"
6   },
7   "created_At": {
8     "$date": {
9       "$numberLong": "1663627574744"
10  }
11  },
12  "__v": 0
13 }

```

Ilustración 36.- Colección coordenadas

Marcador

En esta colección se almacenó la información correspondiente a un marcador, como el tipo de marcador, la descripción, la latitud y longitud, el identificador del usuario y la fecha de creación.

```

1 {
2   "type": "0",
3   "description": "",
4   "created_by": {
5     "$oid": "6292aa888c1c64c57ffabbf9"
6   },
7   "lat": -2.8516401,
8   "lng": -78.9792763,
9   "createdAt": {
10    "$date": {
11      "$numberLong": "1660431815796"
12    }
13  },
14  "__v": 0
15 }

```

Ilustración 37.- Colección de marcadores

Mensaje

En esta colección se almacenó la información correspondiente de un mensaje, como la descripción del mensaje, el identificador del usuario que envía y recibe el mensaje, la fecha de creación.

```
1 {
2   "from": {
3     "$oid": "6292d70e07ae1fc1e8238893"
4   },
5   "to": {
6     "$oid": "628ed677afacf80a80186594"
7   },
8   "message": "",
9   "createdAt": {
10    "$date": {
11      "$numberLong": "1656215262094"
12    }
13  },
14  "__v": 0
15 }
```

Ilustración 38.- Colección mensajes

Reporte

En esta colección se almacenó la información correspondiente a un reporte de una publicación o un comentario, el usuario que es reportado, el usuario que envía el reporte, el estado del reporte, el estado de la lectura del reporte.

```
1 {
2   "user": {
3     "$oid": "6396c2b27b86c16eb422f0bd"
4   },
5   "created_by": {
6     "$oid": "6292aa888c1c64c57ffabfb9"
7   },
8   "publication": {
9     "$oid": "63976b447b86c16eb422f659"
10  },
11  "reason": "Contenido inapropiado",
12  "comentary": "",
13  "deletedAt": null,
14  "isRead": true,
15  "isReadAdmin": false,
16  "status": "lock",

```

Ilustración 39.- Colección reporte

5.7. Creación de widgets

Para el proyecto, se han creado widgets propios con el objetivo de reutilizarlos, hace el código más legible, de menos líneas y fácil de administrar. Al usarlos permiten cumplir con el estándar de código más limpios y evitar la redundancia de código, para lo cual se han desarrollado los siguientes widgets:

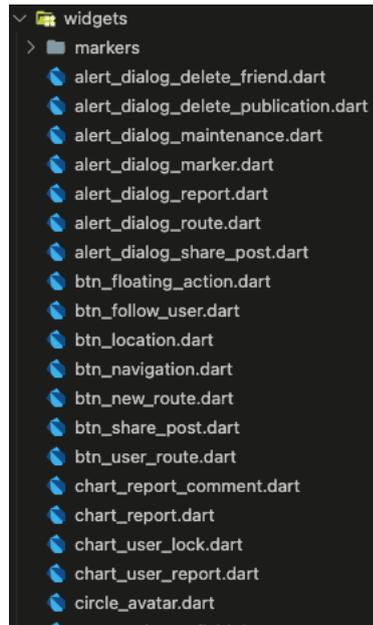


Ilustración 40.- Listado de widgets creados

Por ejemplo, el widget de Circle Avatar permite obtener un avatar circular, el cual consta con definición de dos parámetros, el radio de circunferencia y la imagen a visualizar. para visualizar el widget a más detalle se incluye el anexo 3 manual de [usuario](#).

Es recomendable que el widget sea declarado como una clase, ya que esto permite usar el context es decir el enlace a la ubicación del widget en la estructura de su árbol.

```

import 'package:application_mobile_motorcycle/export.dart';
import 'package:flutter/material.dart';

class CircleAvatarUser extends StatelessWidget {
  final double? radius;
  final String? image;

  const CircleAvatarUser({
    Key? key,
    this.radius,
    required this.image,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return CircleAvatar(
      backgroundColor: thirdColor().withOpacity(0.2),
      radius: radius,
      backgroundImage: NetworkImage(
        '$URL_USER_IMAGE/$image',
      ), // NetworkImage
    ); // CircleAvatar
  }
}

```

Ilustración 41.- Widget Circle avatar

5.8. Notificaciones Push.

5.8.1. Configuración FCM

Para la implementación de las notificaciones se optó por la mensajería en la nube de Firebase, por ventajas en la implementación y compatibilidad durante el uso del proyecto, para enviar mensajes desde el servidor y recibir datos en la aplicación mediante el servicio de transporte correspondiente a la plataforma.

Dentro de la implantación se creó un proyecto en la consola de Firebase, en la sección del panel principal del proyecto, se ingresó a la configuración del proyecto como se muestra en la siguiente ilustración.

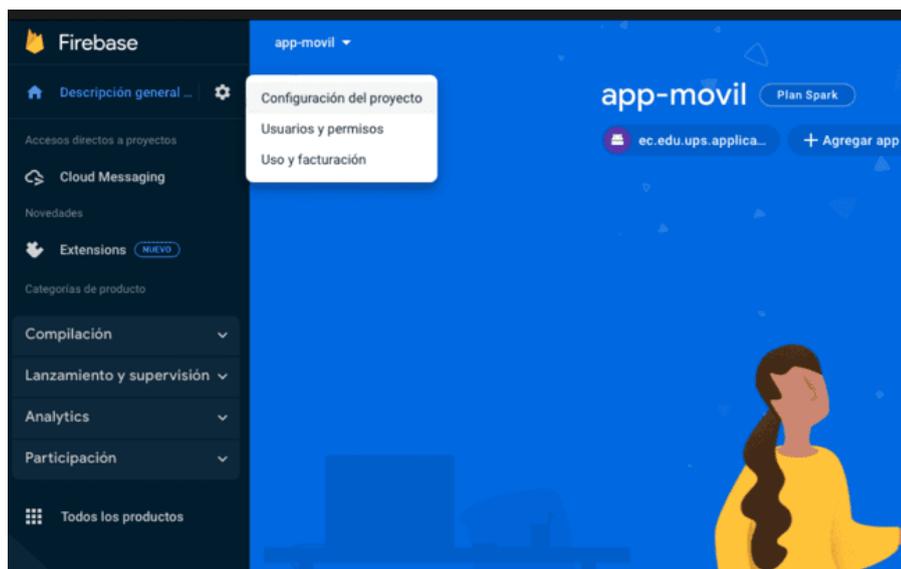


Ilustración 42.- Pantalla principal de configuración Firebase

En la pantalla de configuración general del proyecto, se dirigió hacia el final para agregar la aplicaciones respectivas a cada plataforma, iOS y Android, dentro de la creación se registró el ID del paquete de cada plataforma, se prosiguió a descargar los archivo google-services.json y GoogleService-Info.plist, en el contenido de los archivos se encuentra la configuración necesaria ingresada en el primer paso.

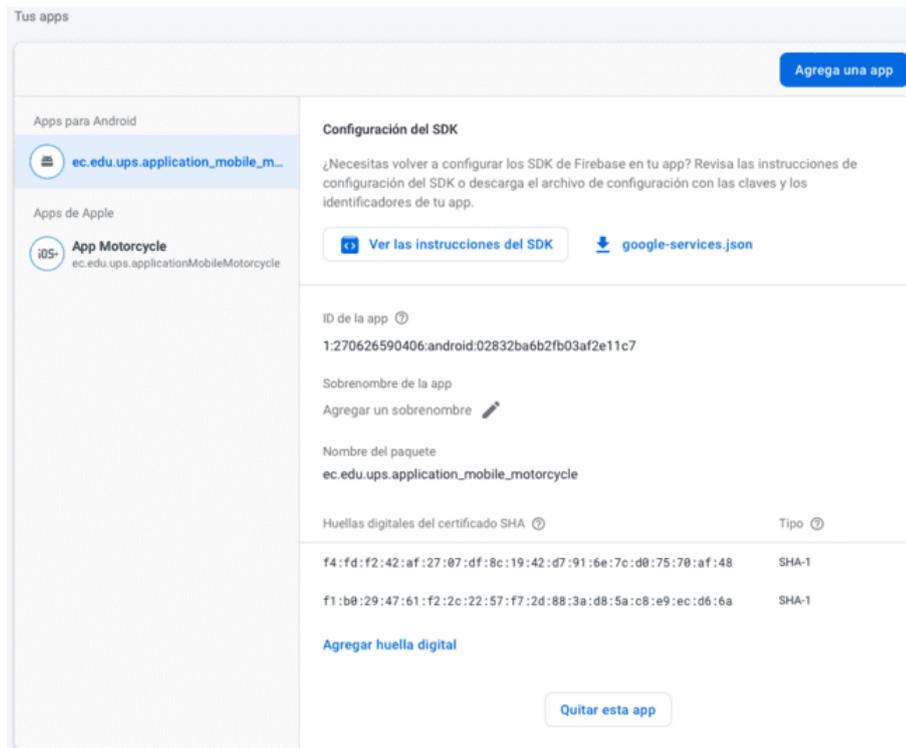


Ilustración 43.- Aplicaciones creadas

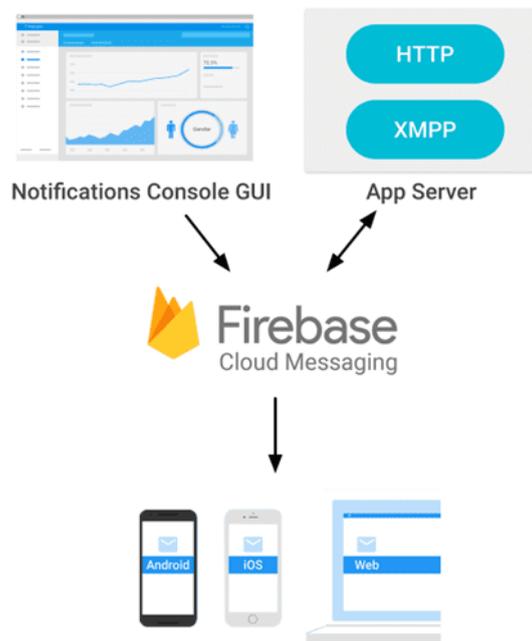


Ilustración 44.- Diagrama de flujo de notificaciones

Dentro de la implementación en el Back-End, se optó por una arquitectura para el manejo de los datos, donde se requiere un listado de Token de ID a donde se envía las notificaciones, el mensaje de la notificación, nombre del usuario remitente, la acción a realizar, el tipo y estado de la notificación.

```
const user = await User.findById(readMotorcycleMaintenance.created_by);
if (user != null) {
  if (user.token != null) {
    fcm.customerNotification(user.fcm, 'Realizar mantemiento', req.user.name, {
      click_action: 'FLUTTER_NOTIFICATION_CLICK',
      type: 'request',
      statesMessage: 'maintenance',
    });
  } else {
    res.status(407).json({
      ok: false,
      msg: 'Token not found'
    });
  }
} else {
  res.status(404).json({
    ok: false,
    msg: 'User not found'
  });
}
```

Ilustración 45.- Validación de una notificación push

En la siguiente imagen se describe el diagrama de flujo de la notificación push, donde se valida la existencia del usuario destinatario, procediendo a validar el listado de Token, terminado con el proceso de enviar la notificación.

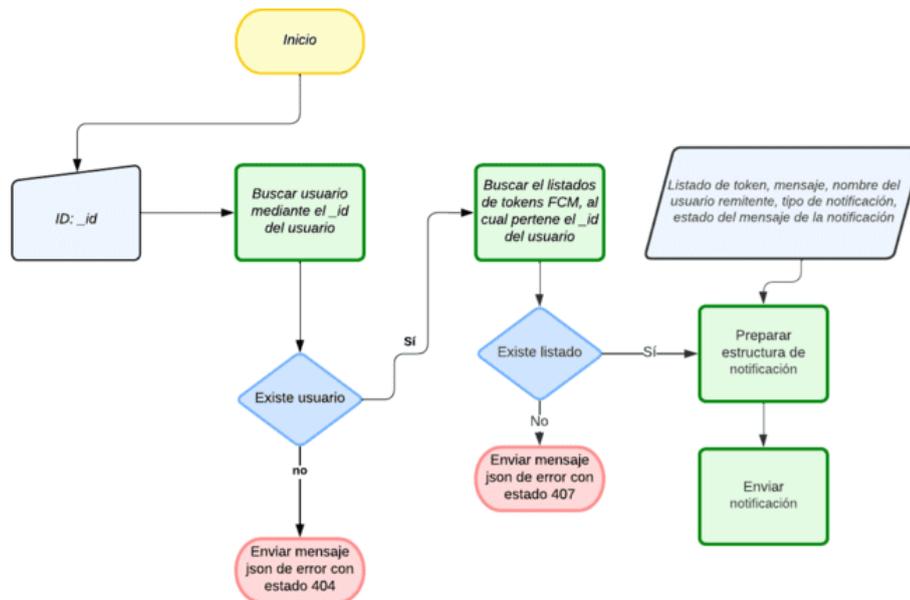


Ilustración 46.- Diagrama de flujo de la notificación push

En la siguiente imagen se visualiza como se recibe las notificaciones en primer y segundo plano, con el ejemplo de realizar mantenimiento, para revisar el funcionamiento de las demás notificaciones a más detalle se incluye el anexo 3 manual de [usuario](#).

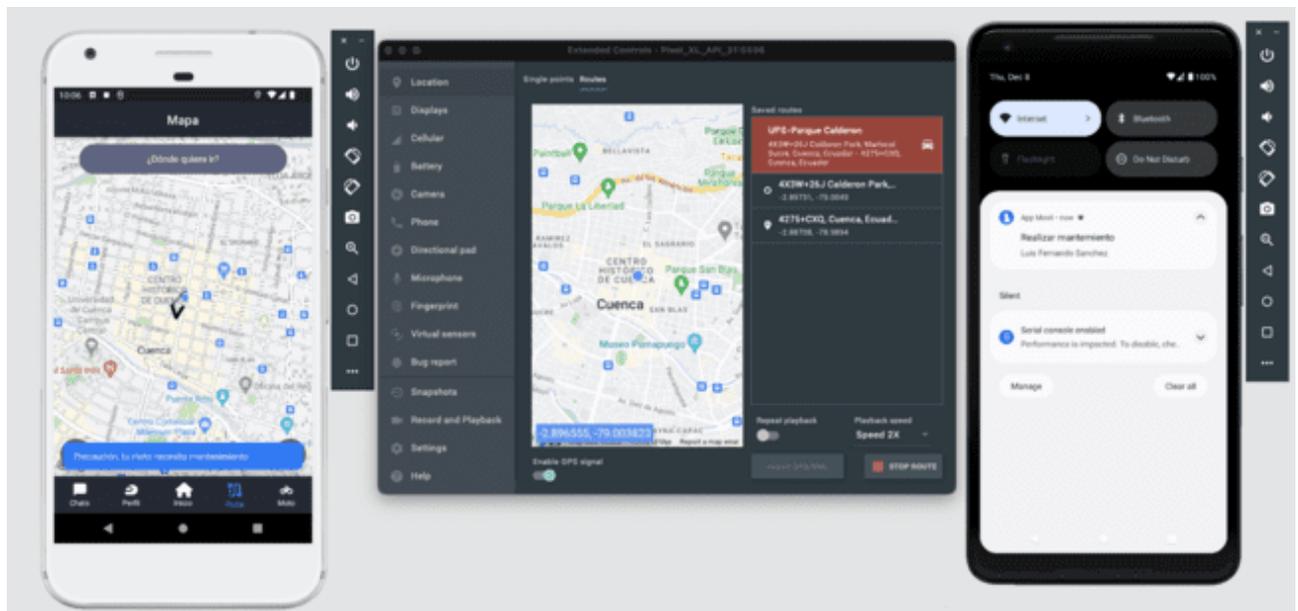


Ilustración 47.- Notificaciones en primer y segundo plano

5.9. Módulos de la aplicación

5.9.1. Logueo y registro de usuarios

Al iniciar la aplicación se hizo el registro o inicio de sesión donde se podrá crear un usuario con un único email y cédula valida, esto es ideal ya que la toda la información que se encuentra en la base de datos se puede filtrar a partir del usuario.

Una vez ya creada la cuenta se puede iniciar sesión con las credenciales correctas, caso contrario saldrá un mensaje de error.

Cabe recalcar que al iniciar sesión se crea un token de inicio de sesión.

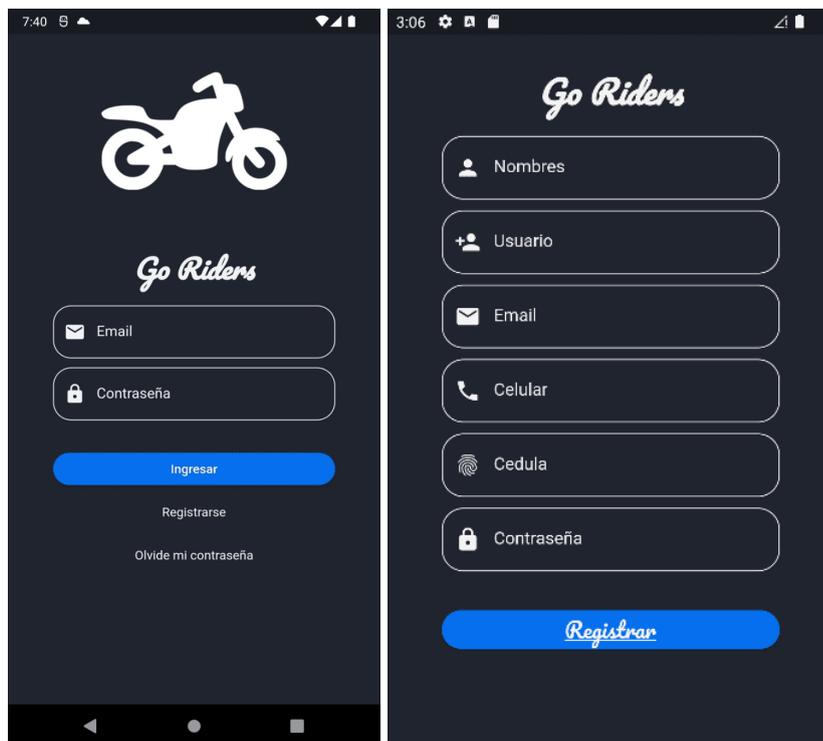


Ilustración 48.- UI Pantalla de bienvenida o logueo y registro

5.9.2. Feed o noticias

En este apartado de red social se puede agregar o eliminar amigos mediante una búsqueda, para después visualizar las solicitudes enviadas. Cuando un usuario acepte la solicitud de amistad en el contenedor se visualizarán todas las publicaciones realizadas por ese usuario y por todos los usuarios que tenga agregados.

El usuario logueado también podrá realizar una publicación mediante la carga de un contenido multimedia, toma de una foto mediante la cámara del dispositivo y un texto que puede ser opcional, de igual manera al publicar, el post estará visible para todos los amigos del usuario.

También el usuario podrá realizar comentarios o agregar “likes” las publicaciones, y el dueño de la publicación será notificado.

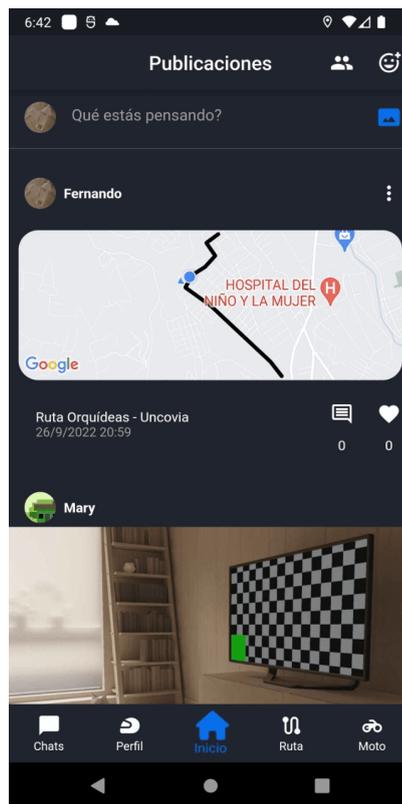


Ilustración 49.- UI Pantalla de noticias o publicaciones

5.9.3. Chat

En esta pantalla se podrán iniciar conversaciones con un usuario agregado, los usuarios conectados se visualizarán con un punto de color verde y los desconectados con un punto de color rojo. Al iniciar una conversación el servicio de mensajería funciona en tiempo real y mediante notificaciones al momento de que se ha recibido un mensaje de texto.

También se visualiza el historial de chats con toda la cronología del usuario que son almacenados en el servidor.

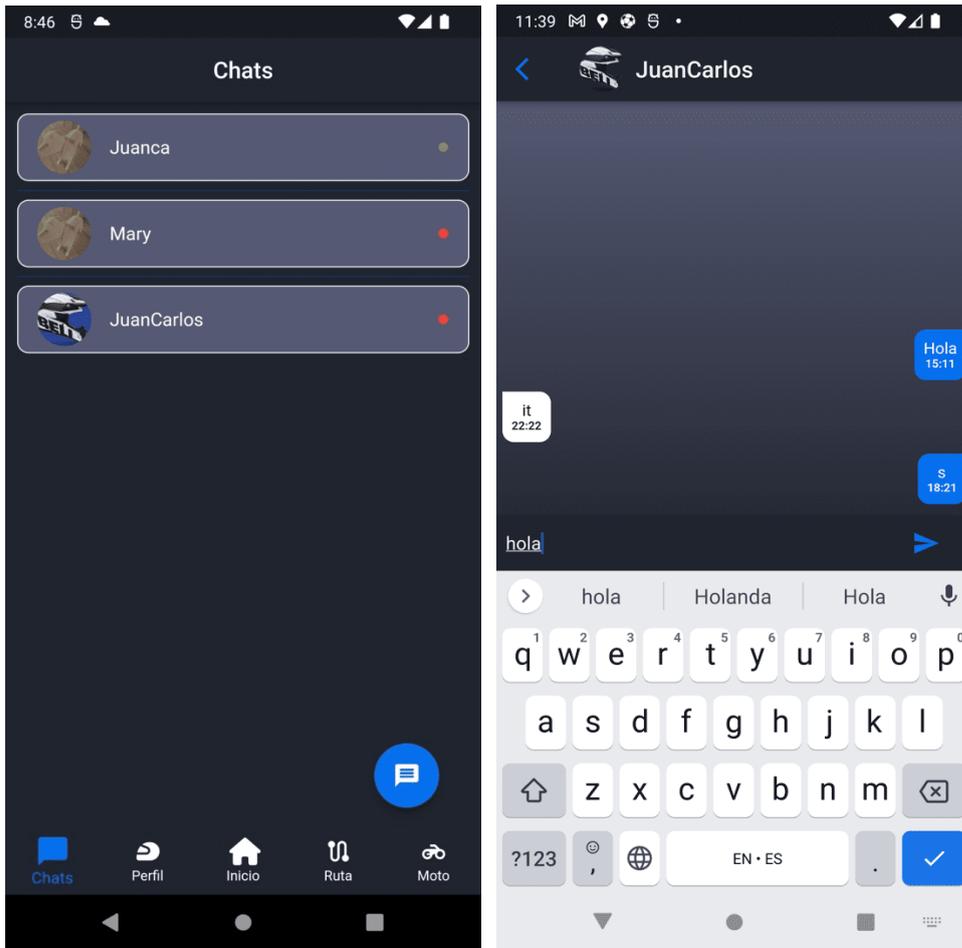


Ilustración 50.- UI Pantalla de historial de chats y chats

5.9.4. Perfil

En el apartado de Perfil, se podrá visualizar toda la información del usuario logueado con el que se realizó el registro, los mismos datos pueden ser editados y guardados, de igual manera en esta ventana se permite la carga de una foto de perfil o toma de una foto con el dispositivo móvil.

También en la parte inferior se cargarán todas las publicaciones que ha realizado el usuario en modo de historial cronológico.

En la parte superior derecha se podrá encontrar el icono de cerrar la sesión del usuario.

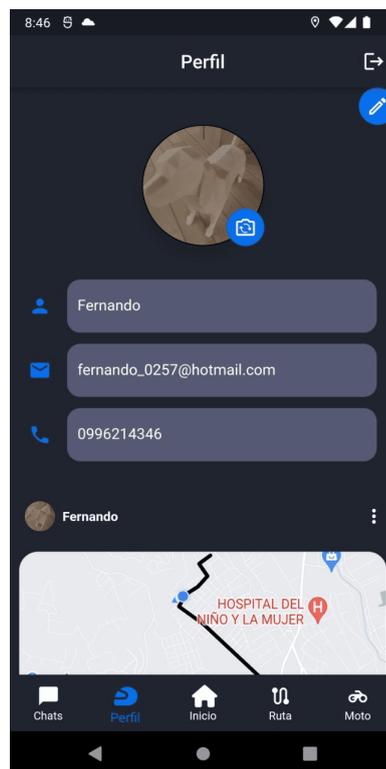


Ilustración 51.- UI Pantalla de perfil

5.9.5. Moto

Este módulo se permite realizar la carga de información de la motocicleta con datos como marca, modelo, cilindraje, y año de creación del vehículo, una vez completado el registro se puede realizar la carga de la foto de la moto mediante un archivo o mediante la toma de una fotografía.

También en este apartado se permite realizar el control del último cambio de aceite de la motocicleta, y visualizar cuando será el próximo cambio, este módulo es útil para el usuario ya que puede evitar daños en su motocicleta si el cambio es realizado en su tiempo correcto.

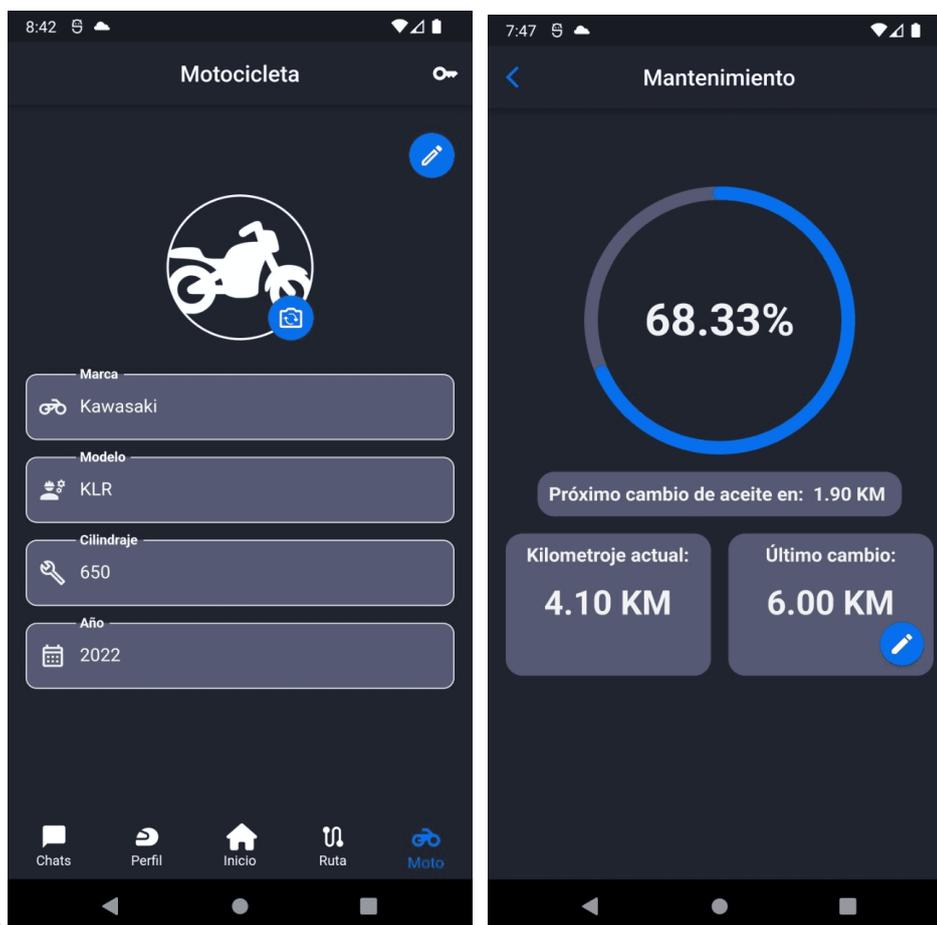


Ilustración 52.- UI pantalla carné digital

5.9.6. Rutas

Este módulo permite buscar y agregar marcadores personalizados con datos específicos de latitud y longitud, además también de crear y realizar rutas, antes de comenzar a realizar las rutas es necesario comenzar con buscar un lugar de destino, una vez defino el lugar destino ya sea mediante un servicio de búsqueda con los servicios de MapBox o colocar el marcador central manualmente, continua con dos opciones la primera de agregar un marcador personalizado, la segunda de confirmar el lugar de destino.

El usuario visualizará una ruta creada la cual podrá realizar con indicaciones específicas de Google Maps, en caso de otra opción podrá realizar la ruta con las polilíneas dibujadas en el mapa.

También el usuario podrá crear rutas personalizadas en un entorno que se encuentre más cómodo, las cuales puede guardar la ruta o compartir dichas rutas para los demás usuarios.

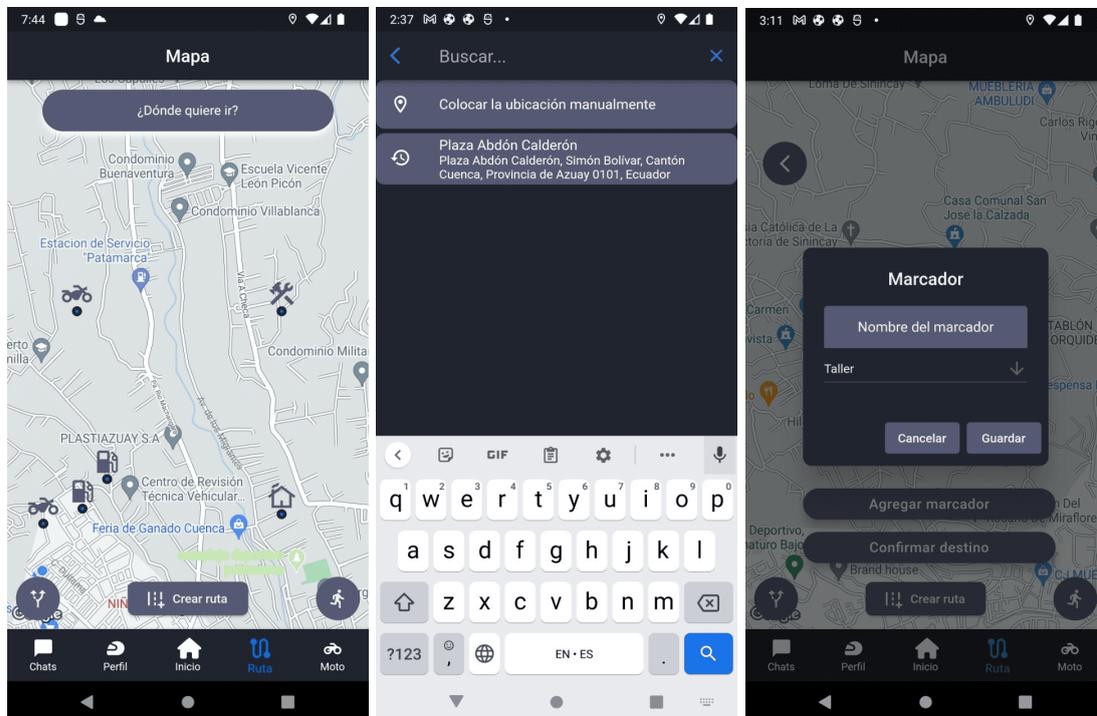


Ilustración 53.- UI interfaces de navegación de ruta, búsqueda de destino y guardar un marcador

6. RESULTADOS

En este capítulo actual se puede visualizar los resultados obtenidos del diseño e implementación, de los requerimientos funcionales definidos para el desarrollo del proyecto propuesto, con el consumo de API Services desde Node JS hacia el framework de Flutter.

6.1. Análisis de Performance.

En el desarrollo de proyecto se realizó diferentes pruebas para recolectar diferentes análisis, por tal motivo se hicieron en dispositivos físicos o emulados tanto para iOS como Android, en gran parte de la creación de rutas se optó por emular en dispositivos Android debido a la compatibilidad y la facilidad del uso de geolocalización. Los siguientes análisis corresponden a la ejecución física en un dispositivo físico iPhone XR, con la versión iOS 16.0.2 y un emulador Pixel 2 XL con la versión de Android 12 API 3.

Las siguientes capturas muestran la información general de registros y análisis de ejecución en distintos dispositivos, con el “conjunto de herramientas de rendimiento y depuración DevTools” del lenguaje Dart. Brindando información del rendimiento de la aplicación. (Flutter, 2022)

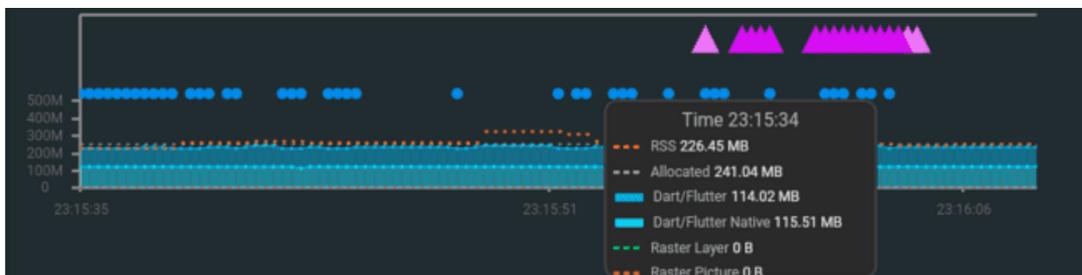


Ilustración 54.- Métrica de rendimiento en el dispositivo de la plataforma iOS

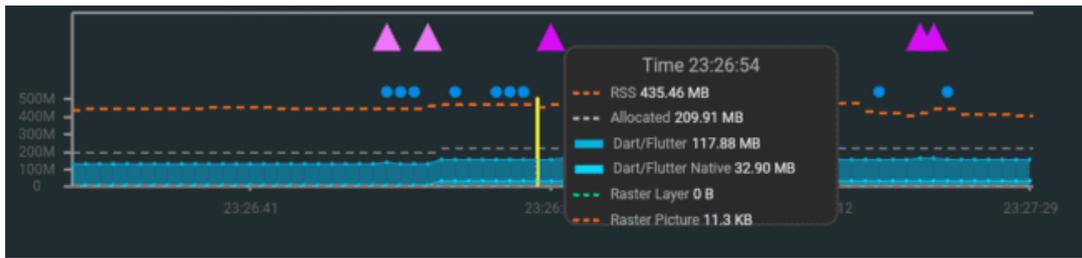


Ilustración 55.- Métrica de rendimiento en el simulador de la plataforma Android

En las ilustraciones se visualiza un incremento del 209.02 MB en el tamaño de conjuntos residentes, en la ejecución del simulador de Android, la información es obtenida de los picos más altos del uso de memoria de Flutter.

6.2. Pruebas de requerimiento funcionales.

El objetivo de estas pruebas de requerimiento funcional es demostrar que coincidían los requerimientos funcionales con el comportamiento de la aplicación después de terminar con su desarrollo, evitando que se encuentren problemas.

6.2.1. Pruebas funcionales unitarias.

- Servicio - Registro de cuenta.
- Servicio – Iniciar sesión.
- Servicio – Recuperar contraseña.
- Aplicación móvil.

CASO DE PRUEBA PF-01	
Responsable	Luis Fernando Sánchez Sucuzhañay
Fecha de Ejecución	11-11-2022
Requerimiento	Implementación de registro de cuenta
¿Prueba Aprobada?	Si

Resultado Esperado

En el inicio de la aplicación, se obtiene un menú de opciones para ingresar a la aplicación, al seleccionar “Registrarse”, es necesario ingresar información del usuario, para verificar si la información ingresada ya se encuentra registrada o se procede a crear un nuevo usuario. Otra opción es seleccionar “Ingreso”, la cual procede desde el registro de la cuenta. Por último la opción de “Olvide mi contraseña”, es necesario contar con el email de cuenta registrada para solicitar una nueva contraseña temporal, la cual se enviará al correo electrónico.

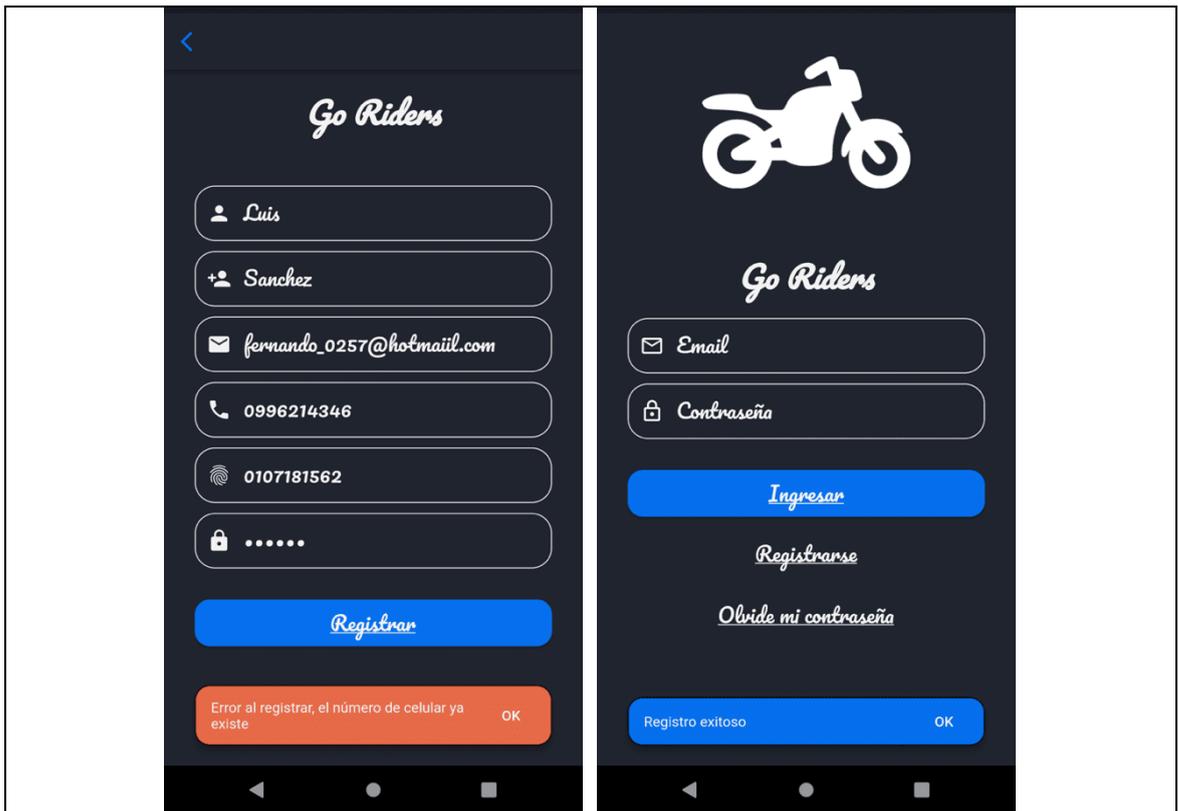
Resultados esperados:

- Visualizar pantalla principal y de registro.
- Selección entre pantallas “Registrarse” y “Registrar” cuenta.
- Validar ingreso a la aplicación.
- Solicitar contraseña temporal.
- Mensajes alertas.

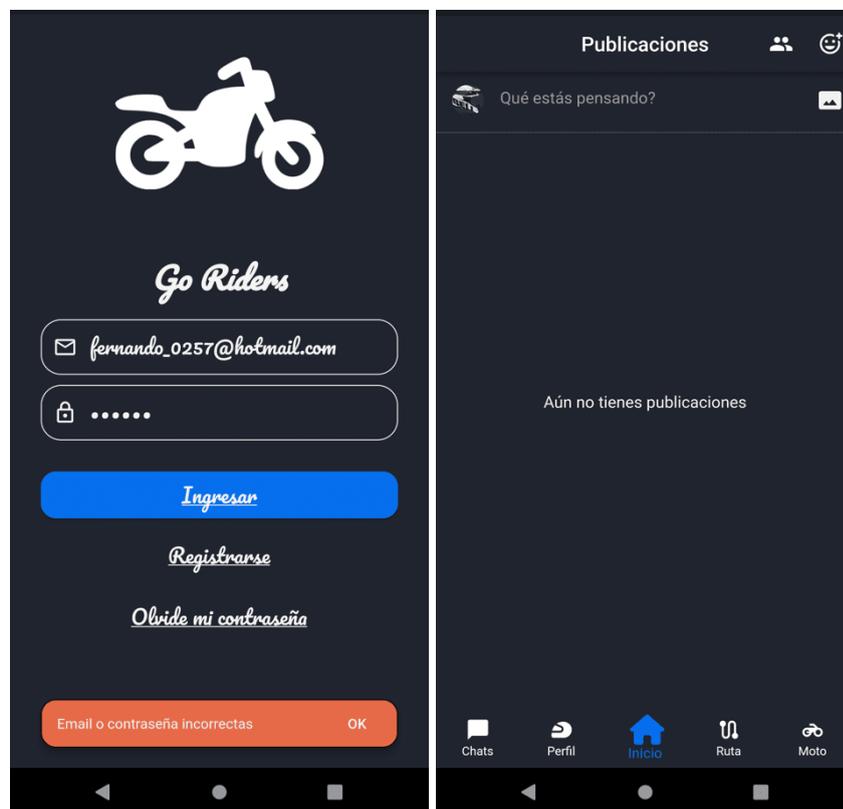
Resultado Real

Resultados obtenidos:

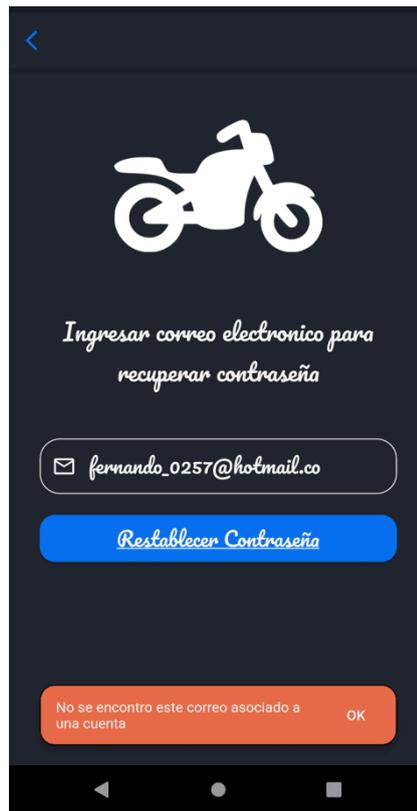
- La aplicación verifica la navegación entre pantallas, la visualización de mensajes erróneos, y creación de cuenta.



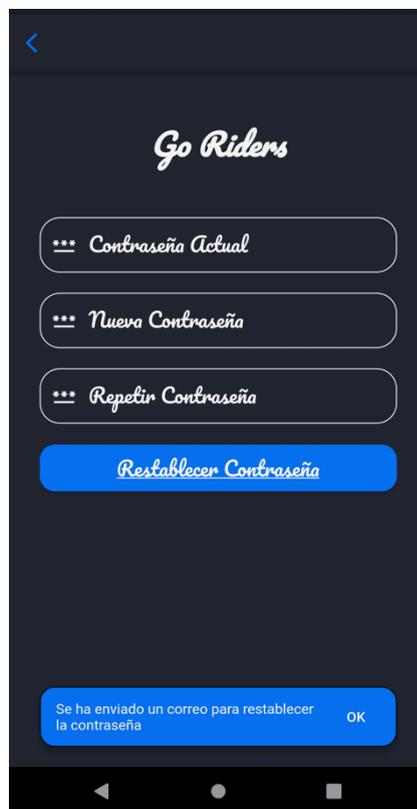
- Verifica el inicio de sesión con la nueva cuenta creada, navegación a la pantalla principal de la aplicación.



- Verifica si la cuenta se encuentra registrada.



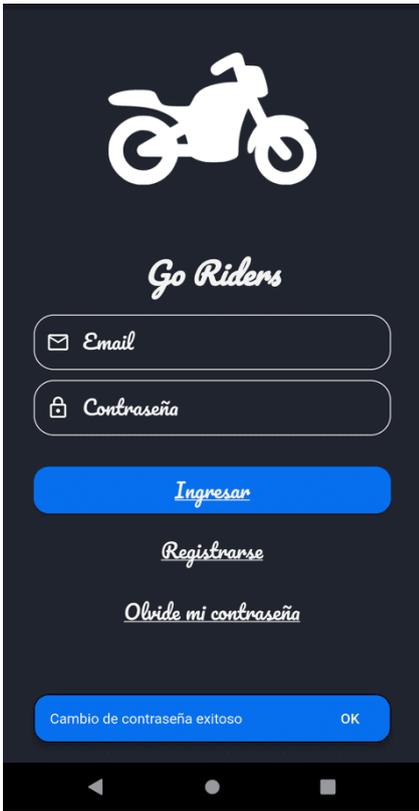
- Recibir correo electrónico con la contraseña temporal.



Recuperar contraseña
Para: Fernando Sanchez

Se ha solicitado recuperar la contraseña de la cuenta asociada a este correo electrónico. Su contraseña temporal es: d9eaxath

- Verificar cambio de contraseña.



- Servicio - Registro de motocicleta.
- Servicio – Registro de mantenimiento de motocicleta.
- Aplicación móvil.

CASO DE PRUEBA PF-02	
Responsable	Luis Fernando Sánchez Sucuzhañay
Fecha de Ejecución	11-11-2022
Requerimiento	Implementación de registro de motocicleta
¿Prueba Aprobada?	Si

Resultado Esperado

En la pantalla principal, se visualiza una opción de motocicleta, al seleccionar dicha opción es necesario ingresar información de la motocicleta del usuario, se proseguí a crear la motocicleta, además se tiene verificar los datos para brindar el mantenimiento.

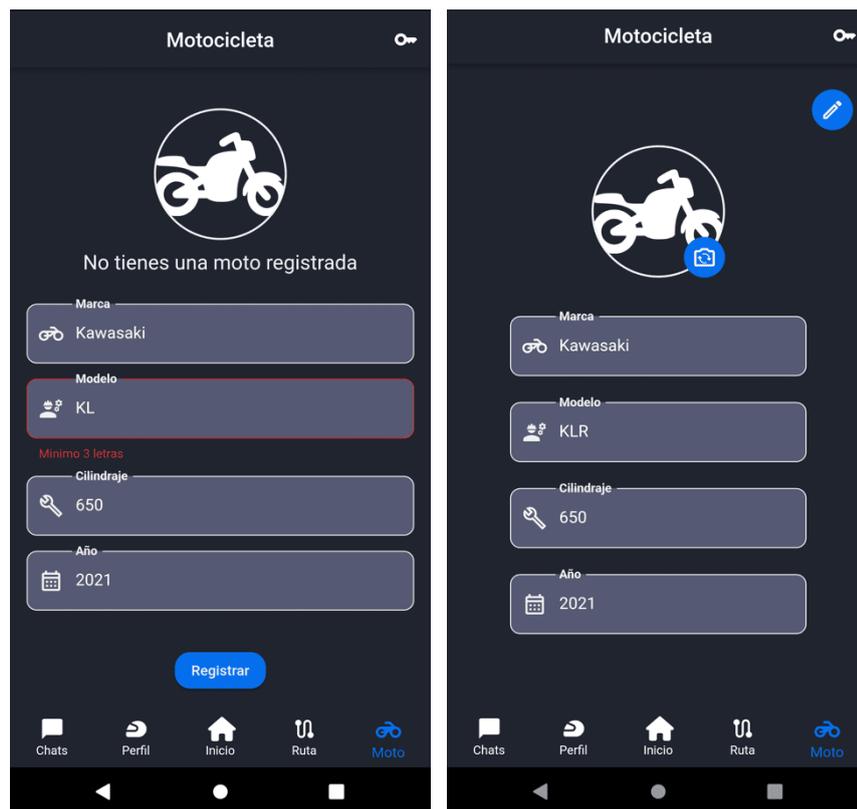
Resultados esperados:

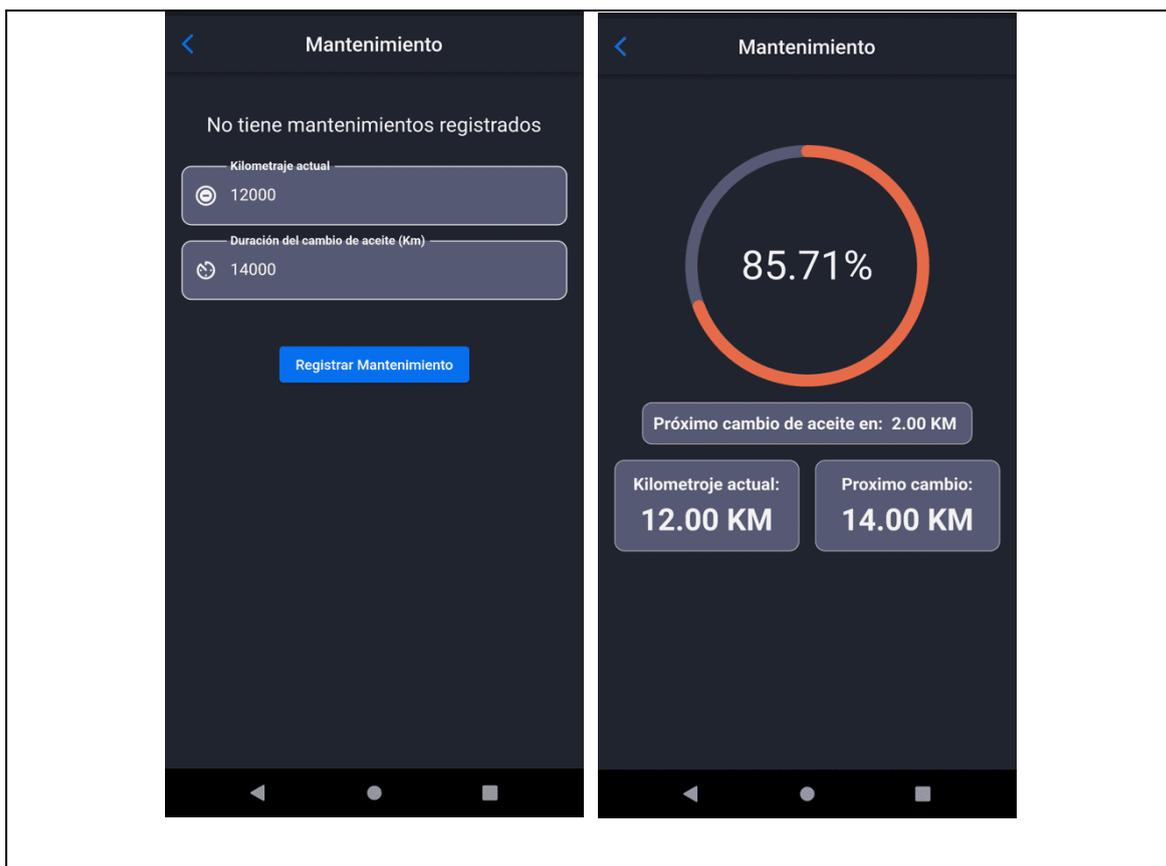
- Visualizar pantalla de registro de motocicleta.
- Validar el mantenimiento de la motocicleta.
- Mensajes alertas.

Resultado Real

Resultados obtenidos:

- Verificar la visualización de mensajes erróneos, creación de la motocicleta con su respectivo mantenimiento.





- Servicio - Registro de rutas y marcadores personalizados.
- Aplicación móvil.

CASO DE PRUEBA PF-03	
Responsable	Luis Fernando Sánchez Sucuzhañay
Fecha de Ejecución	11-11-2022
Requerimiento	Implementación de registro de rutas y marcadores
¿Prueba Aprobada?	Si
Resultado Esperado	
<p>En la pantalla principal, se visualiza una opción de rutas, al seleccionar dicha opción es necesario brindar permisos de geolocalización para ingresar a visualizar el mapa, dentro del mapa se visualiza opciones de crear una ruta con ingresando de una dirección exacta o colocando el marcador en el punto exacto, además se colocar el marcador de forma manual también se visualiza la opción de crear un marcador personalizado. Otra opción para crea una ruta es en tiempo real.</p>	

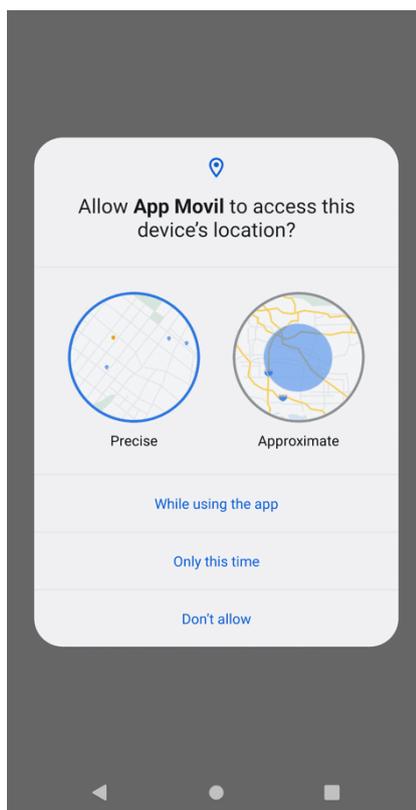
Resultados esperados:

- Visualizar los permisos de geolocalización.
- Validar la creación de rutas de forma manual o en tiempo real.
- Validar la creación de marcadores personalizados.

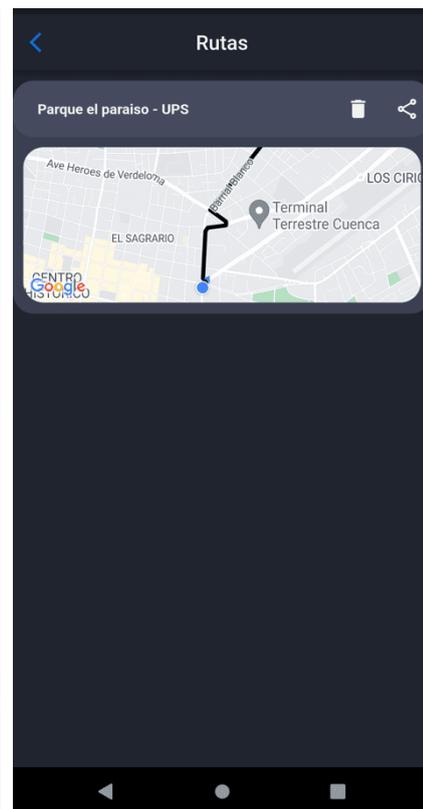
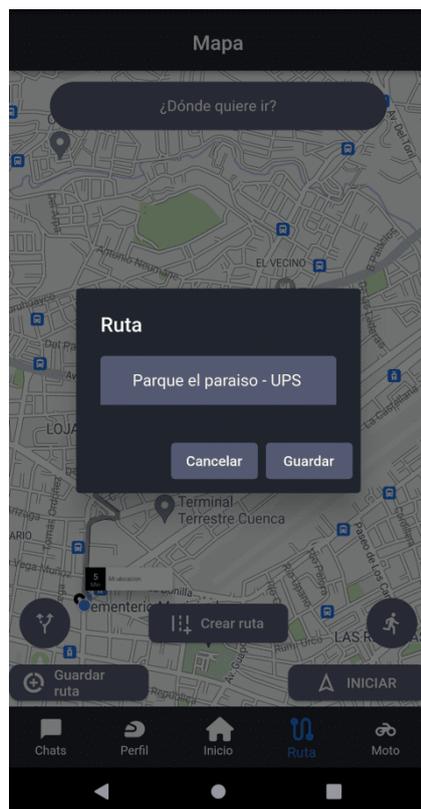
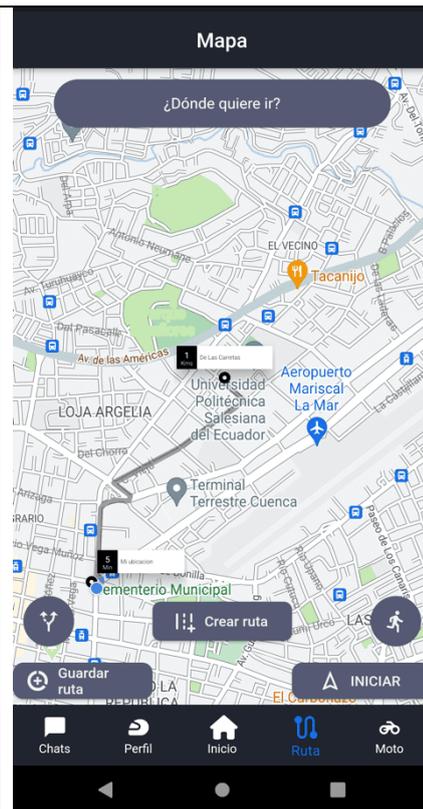
Resultado Real

Resultados obtenidos:

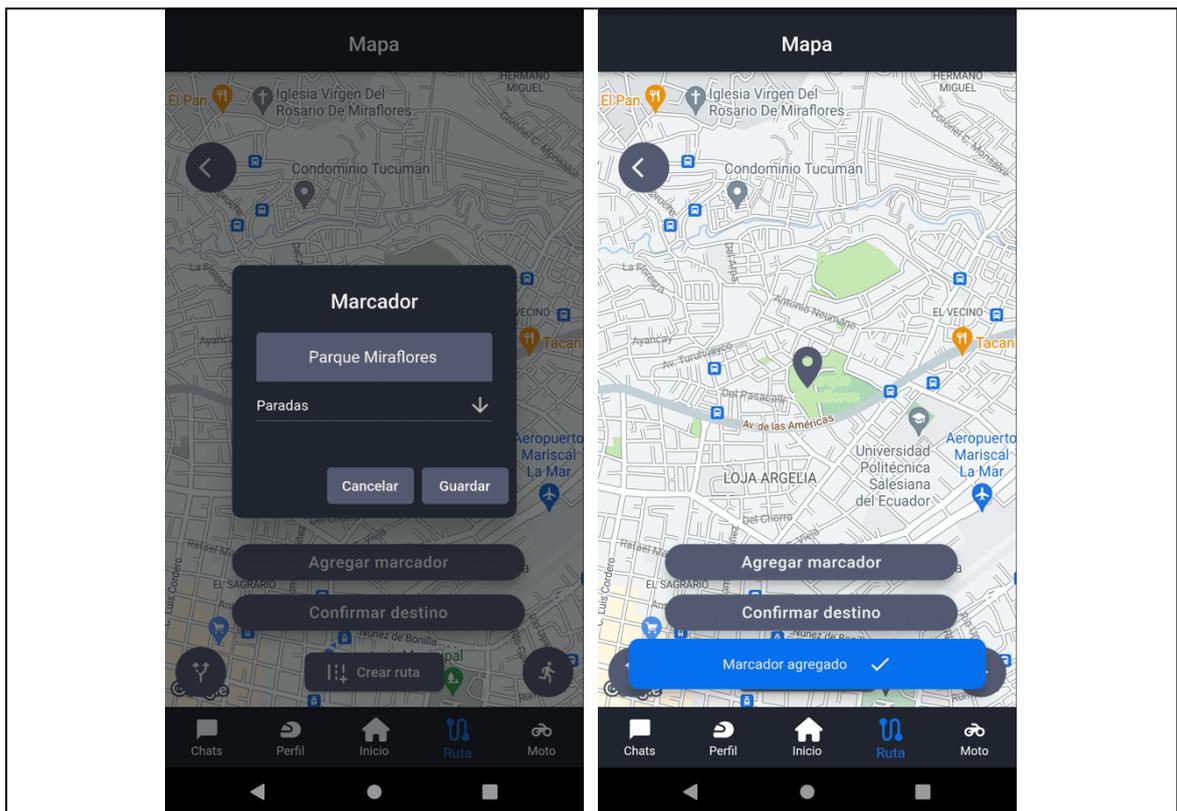
- Solicitar permiso de geolocalización, para ingresar a visualizar el **mapa**.



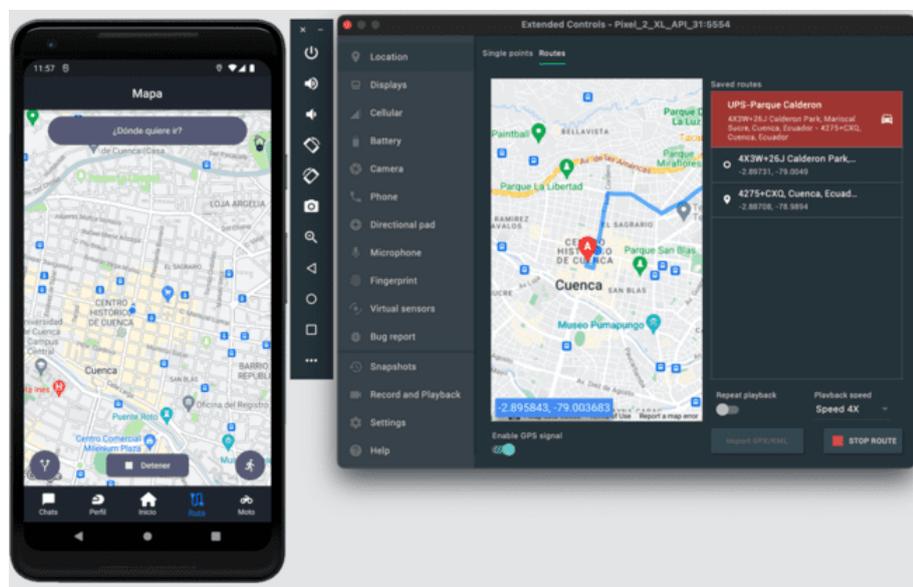
- Verificar la creación de ruta de forma manual.

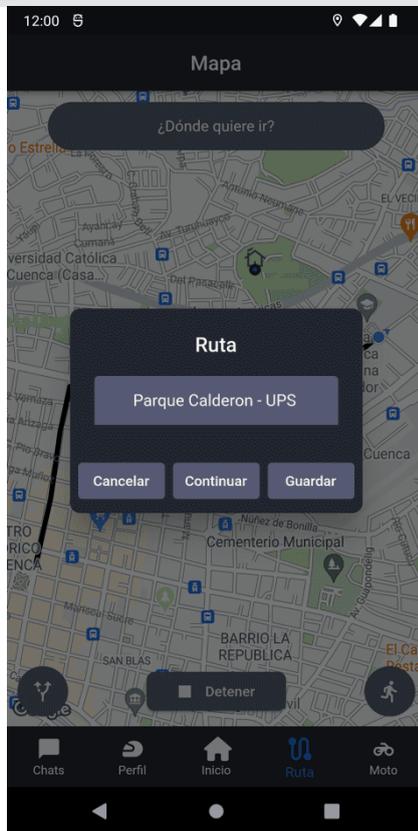
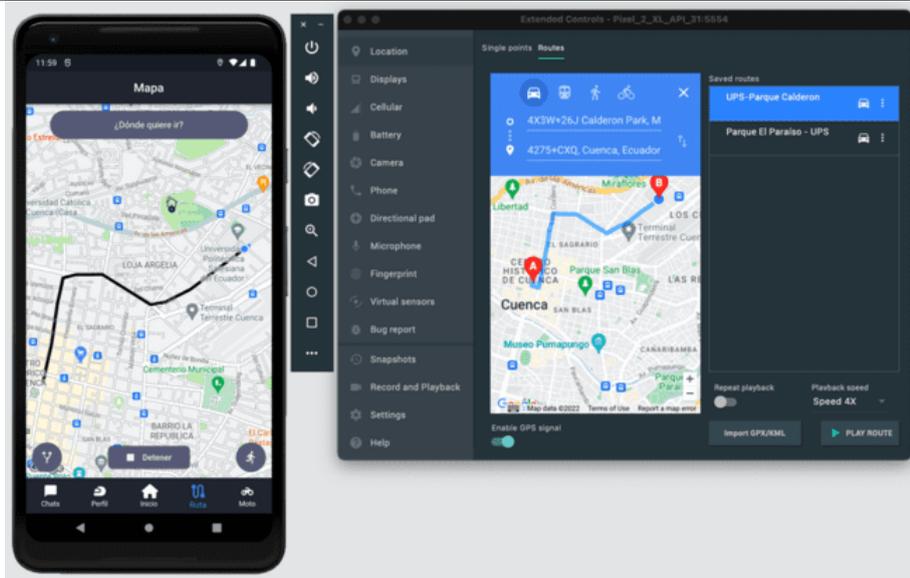


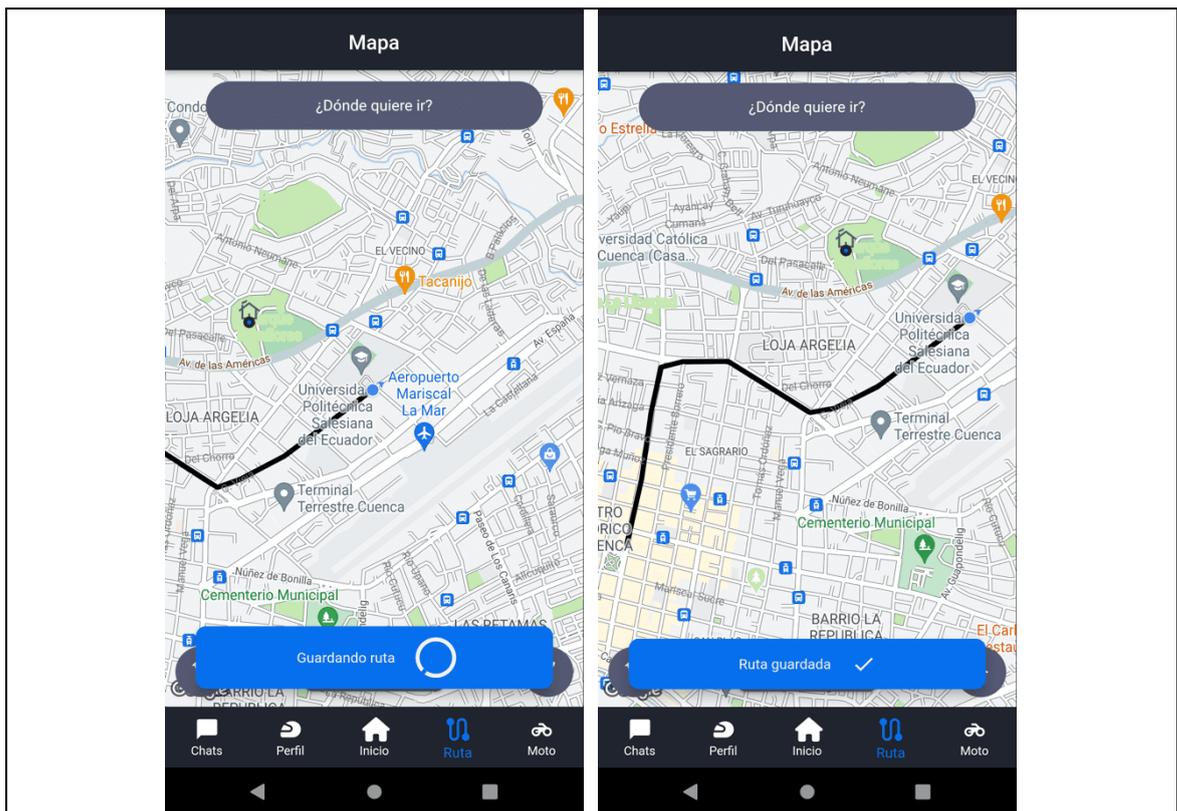
- Verificar la creación del marcador personalizado de forma manual.



- Verificar la creación de ruta en tiempo real, mediante el simulador de Android con la opción de simular una ruta.







- Servicio – Publicación de imagen y rutas.
- Aplicación móvil.

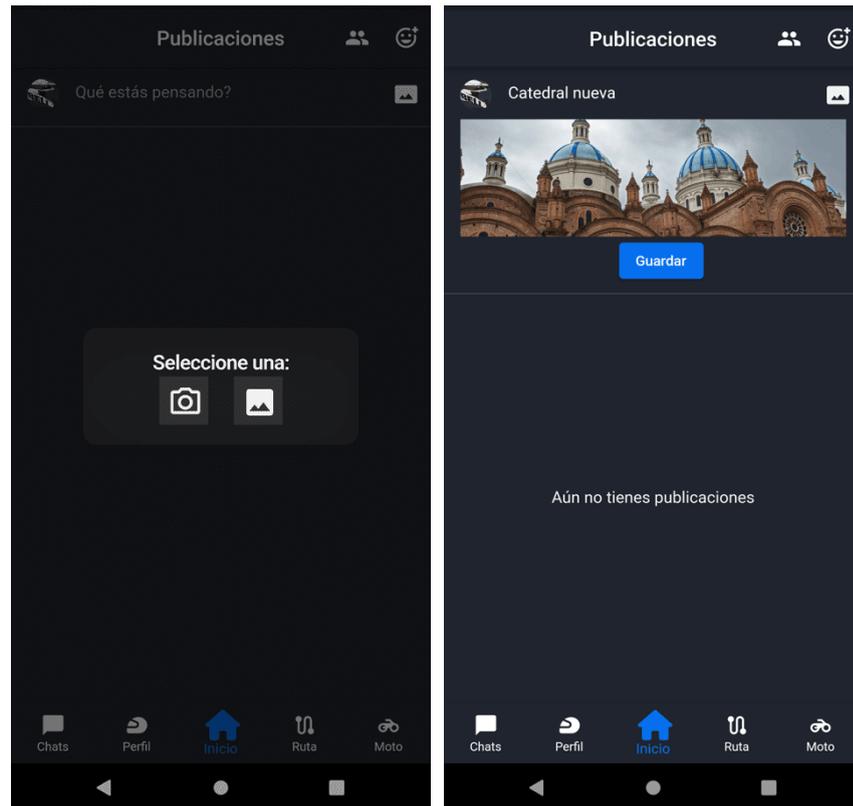
CASO DE PRUEBA PF-04	
Responsable	Luis Fernando Sánchez Sucuzhañay
Fecha de Ejecución	11-11-2022
Requerimiento	Implementación de registro de publicaciones
¿Prueba Aprobada?	Si
Resultado Esperado	
<p>En la pantalla principal, se visualizan las publicaciones registradas por el usuario o amigos del usuario, dentro de las publicaciones de tiene de dos opciones, la primera son publicaciones con un contenido de imágenes, la segunda con un contenido de una ruta registrada, además de publicación puede brindarse una reacción y agregar comentarios a las publicaciones.</p> <p>Resultados esperados:</p> <ul style="list-style-type: none"> • Registrar publicaciones. 	

- Visualizar publicaciones.
- Validar reacciones y comentarios de una publicación.

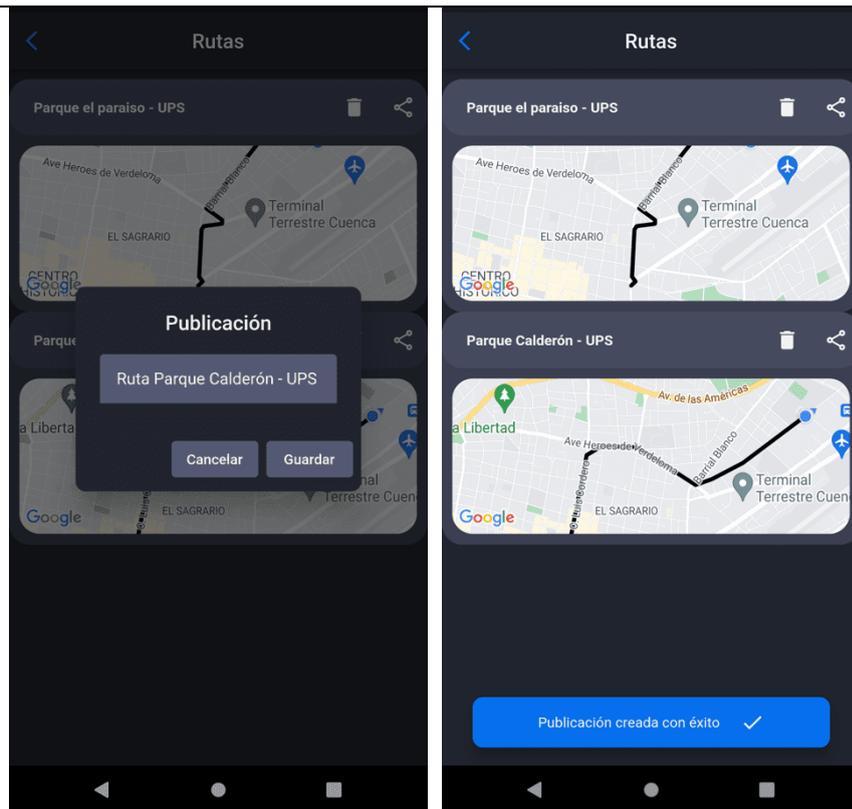
Resultado Real

Resultados obtenidos:

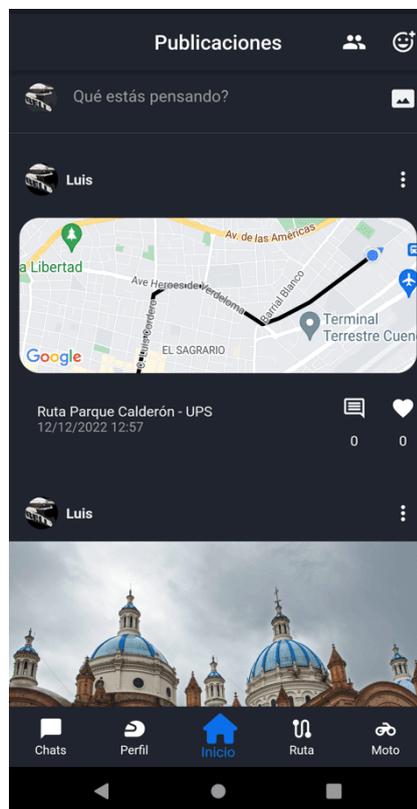
- Verificar la creación del tipo de publicación con el contenido de imagen.



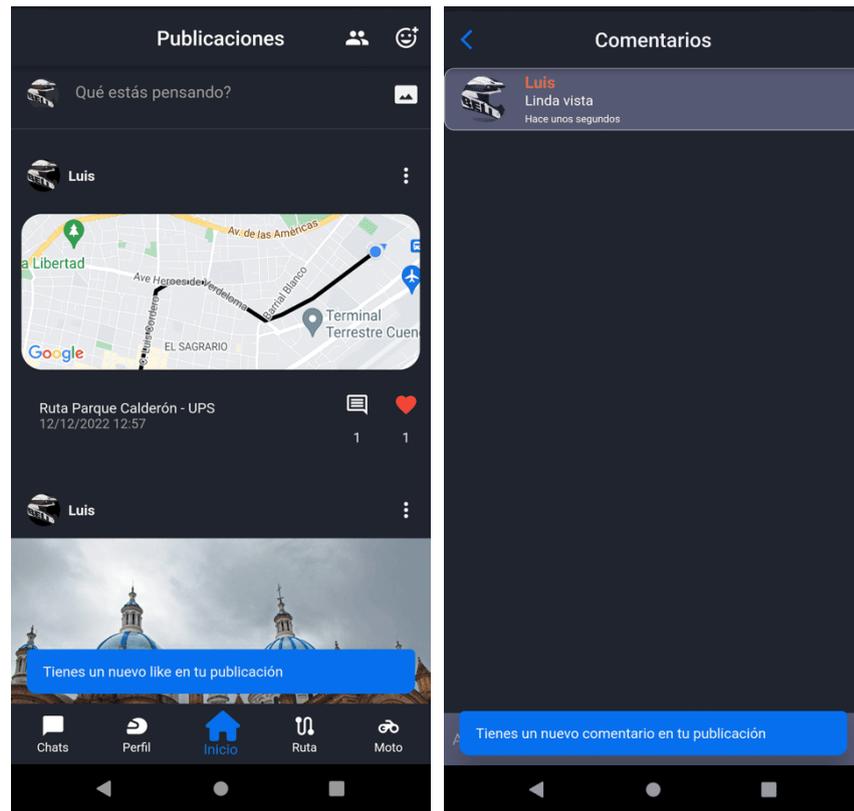
- Verificar la creación del tipo de publicación con el contenido de una ruta registrada.



- Verificar la creación de las publicaciones.



- Verificar la reacción y comentarios en una publicación.



- Servicio – Registro de solicitudes de amistad.
- Aplicación móvil.

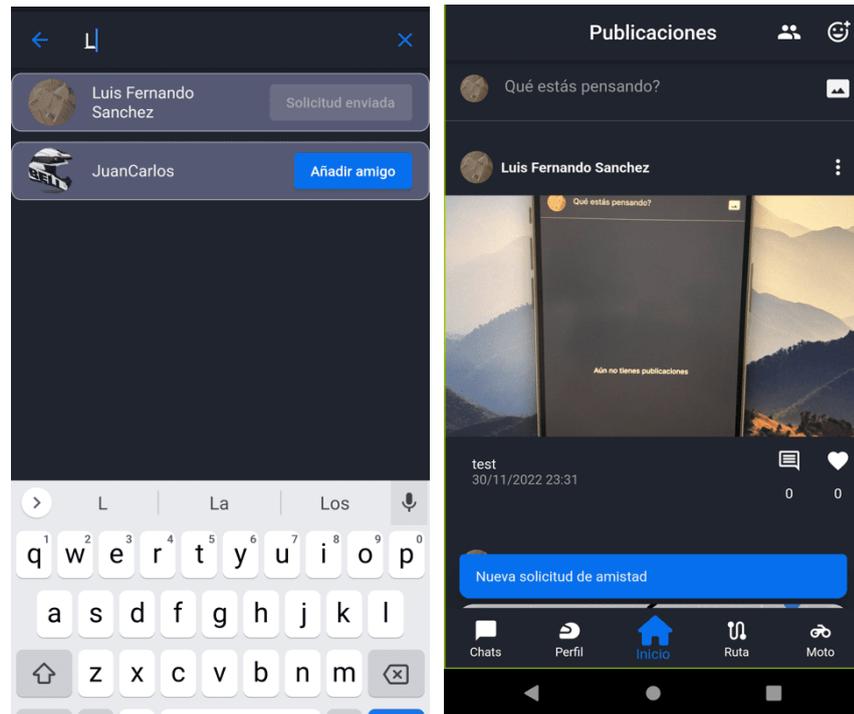
CASO DE PRUEBA PF-05	
Responsable	Juan Carlos Benavides Cabrera
Fecha de Ejecución	12-11-2022
Requerimiento	Implementación de registro de solicitudes de amistad.
¿Prueba Aprobada?	Si
Resultado Esperado	
En la pantalla principal, en la parte superior se visualizan dos opciones de acciones, en la primera opción se visualizan las solicitudes de amistad que tiene el usuario, los amigos que tiene el usuario.	
Resultados esperados:	

- Registrar nuevas solicitudes de amistad.
- Visualizar las solicitudes de amistad.

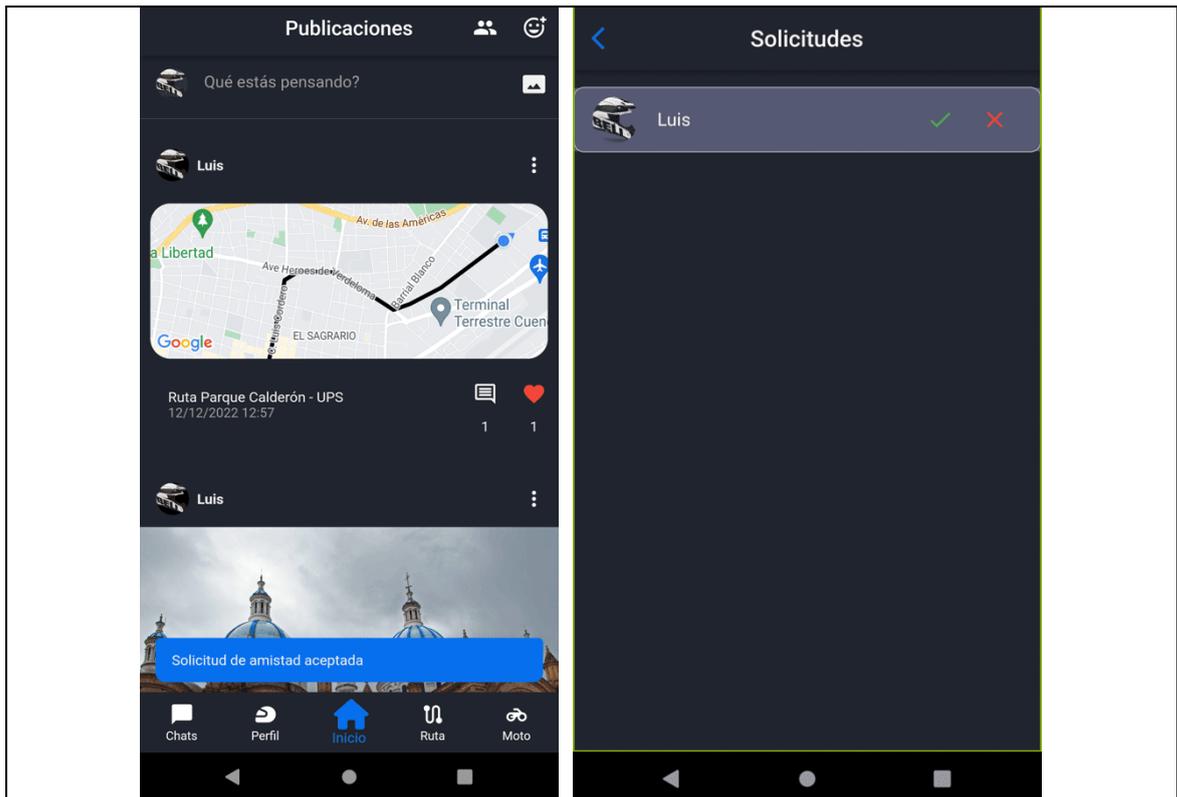
Resultado Real

Resultados obtenidos:

- Verificar las solicitudes de amistad.



- Verificar la reacción de una solicitud de amistad.



- Servicio – Chat.
- Aplicación móvil.

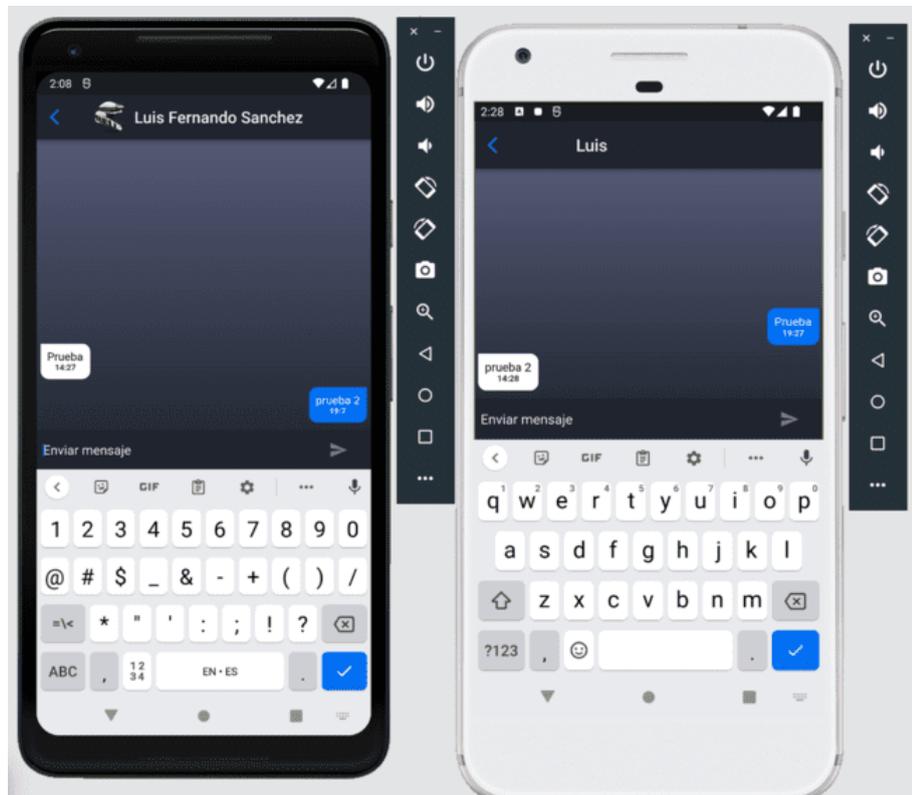
CASO DE PRUEBA PF-06	
Responsable	Juan Carlos Benavides Cabrera
Fecha de Ejecución	12-11-2022
Requerimiento	Implementación de un chat.
¿Prueba Aprobada?	Si
Resultado Esperado	
<p>En la pantalla principal, se visualiza una opción de mensajes, al seleccionar dicha opción se visualiza un listado de chat con los usuarios tiene amistad y están en línea para chatear. Dentro de cada chat se visualiza los mensajes registrados y los que llegan en tiempo real.</p> <p>Resultados esperados:</p> <ul style="list-style-type: none"> • Visualizar mensaje entre usuarios en tiempo real. 	
Resultado Real	

Resultados obtenidos:

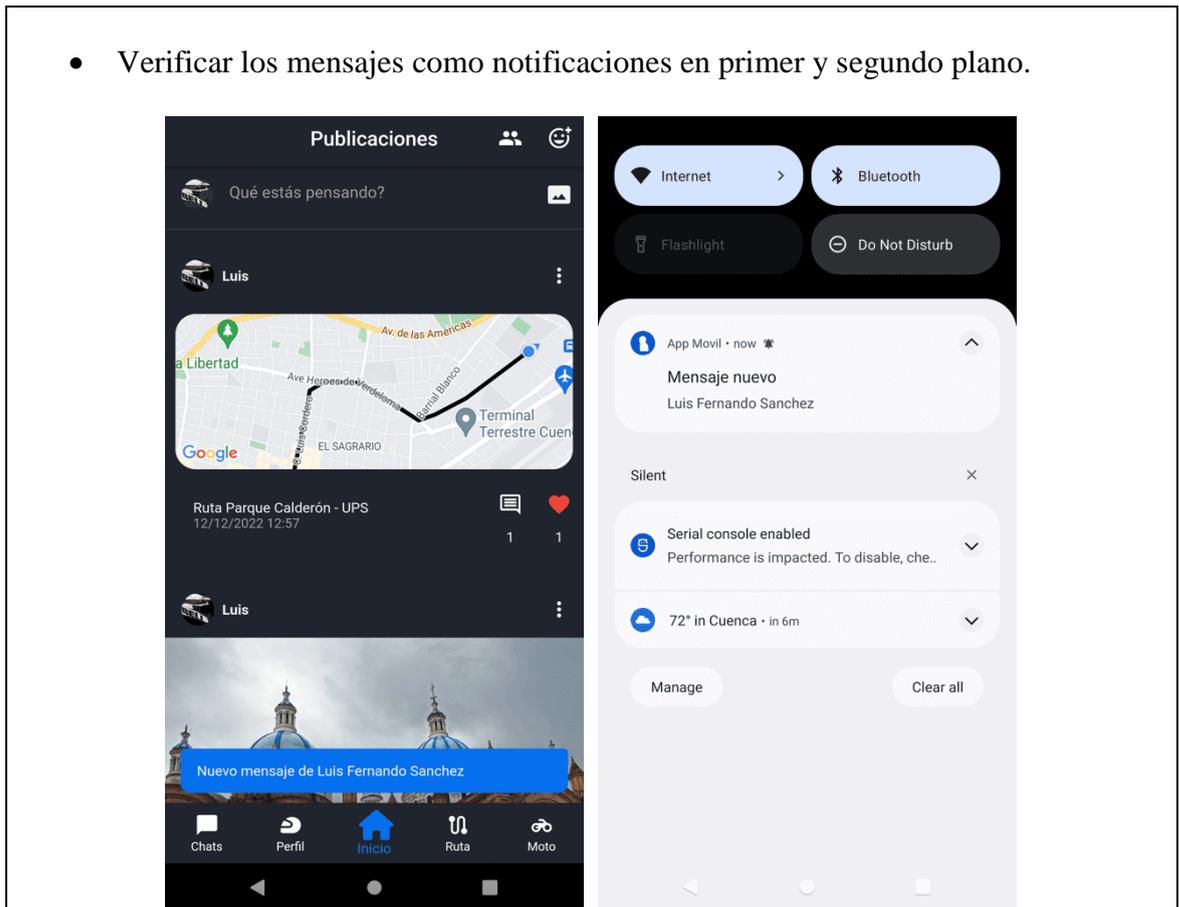
- Verificar el listado de chat del usuario.



- Verificar el chat el tiempo real.



- Verificar los mensajes como notificaciones en primer y segundo plano.



- Servicio – Reportar publicación y comentario.
- Aplicación móvil.
- Aplicación web.

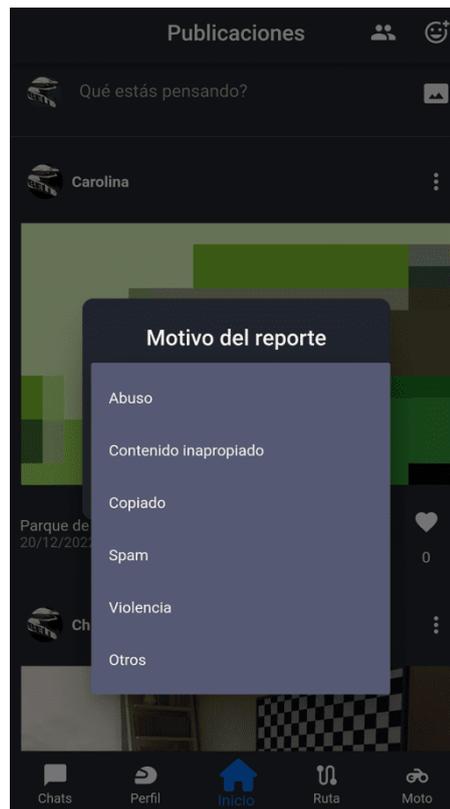
CASO DE PRUEBA PF-07	
Responsable	Luis Fernando Sánchez Sucuzhañay
Fecha de Ejecución	18-02-2023
Requerimiento	Implementación de crear reportes.
¿Prueba Aprobada?	Si
Resultado Esperado	
En la parte superior, a la izquierda de una publicación o un comentario, se visualiza una opción de reportar, al seleccionar dicha opción, se muestra un listado de motivos por el cual se va a reportar.	
Resultados esperados:	

- Visualizar listado de motivos del reporte.
- Realizar reporte.

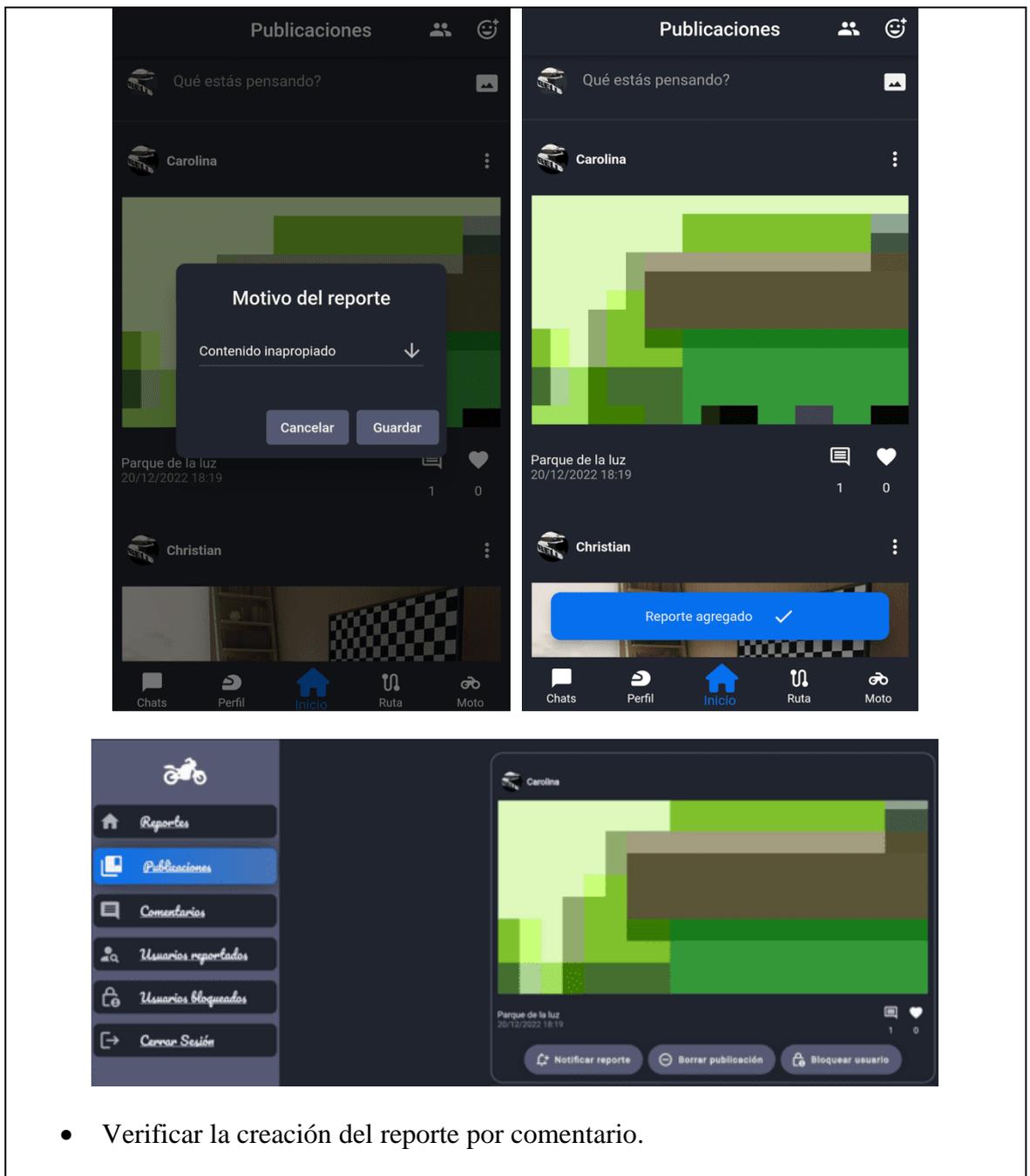
Resultado Real

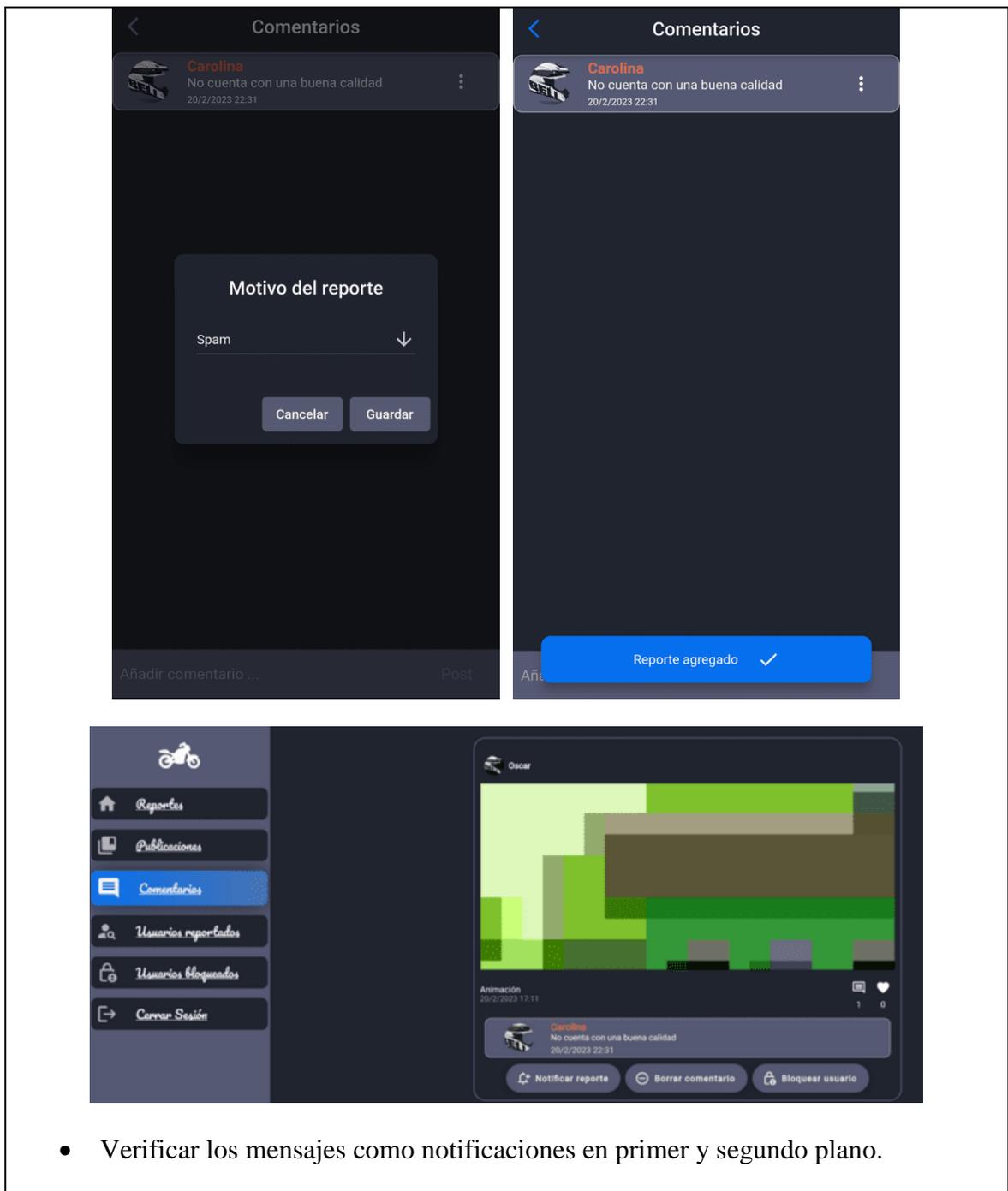
Resultados obtenidos:

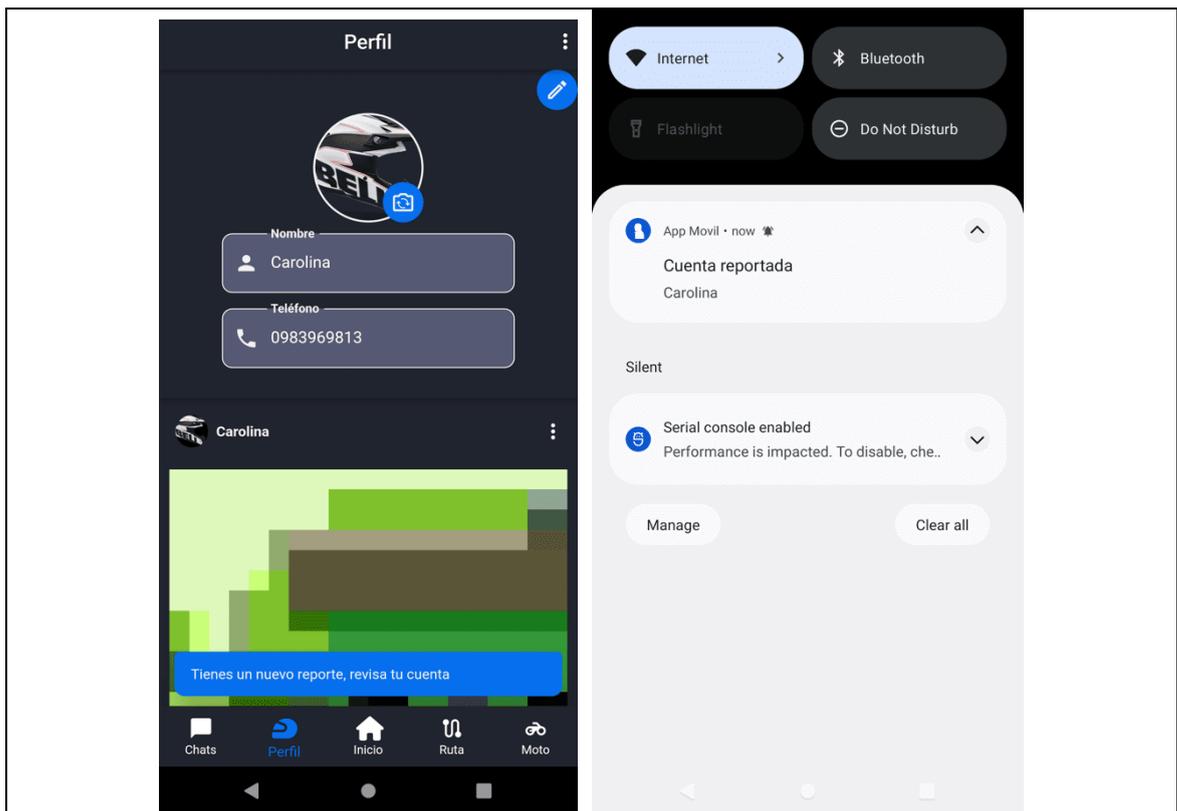
- Listado de motivos del reporte.



- Verificar la creación del reporte por publicación.







- Servicio – Notificar reporte, eliminar publicación y comentario, bloquear usuario.
- Aplicación móvil.
- Aplicación web.

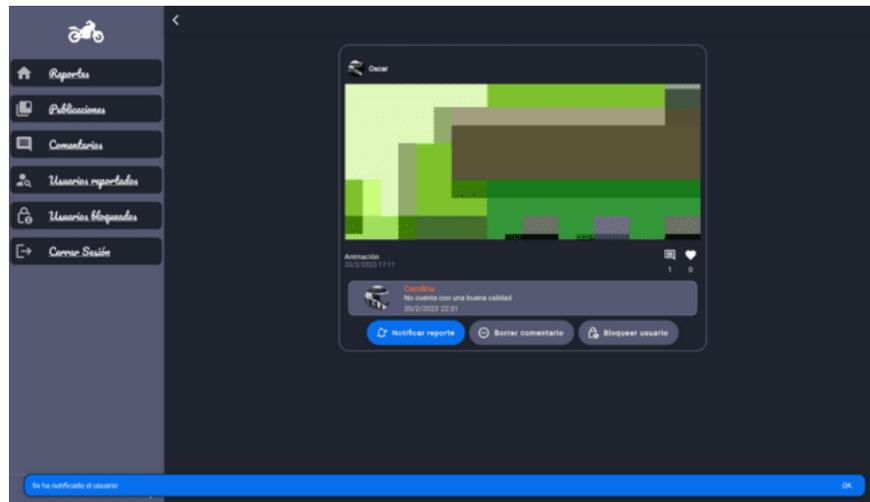
CASO DE PRUEBA PF-08	
Responsable	Luis Fernando Sánchez Sucuzhañay
Fecha de Ejecución	18-02-2023
Requerimiento	Implementación de notificar reporte, eliminar publicación y comentario, bloquear usuario.
¿Prueba Aprobada?	Si
Resultado Esperado	
<p>En el panel principal del administrador, se visualizan 3 acciones, notificar reporte, eliminar publicación y comentario, bloquear cuenta.</p> <p>Resultados esperados:</p> <ul style="list-style-type: none"> • Notificar reportes. • Eliminar publicaciones o comentarios. 	

- Bloquear usuarios.

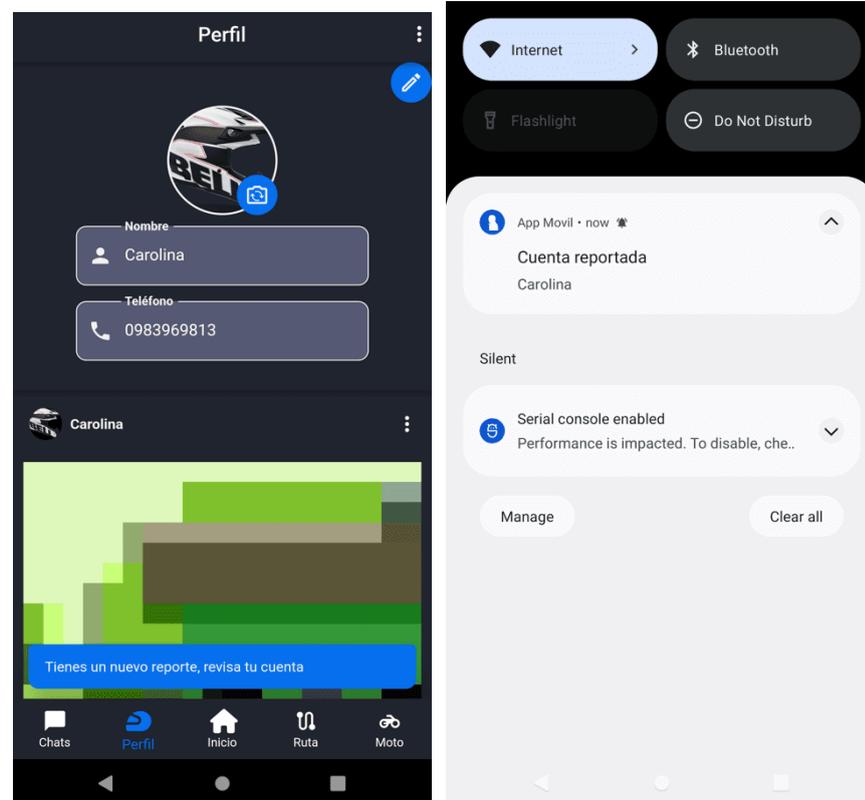
Resultado Real

Resultados obtenidos:

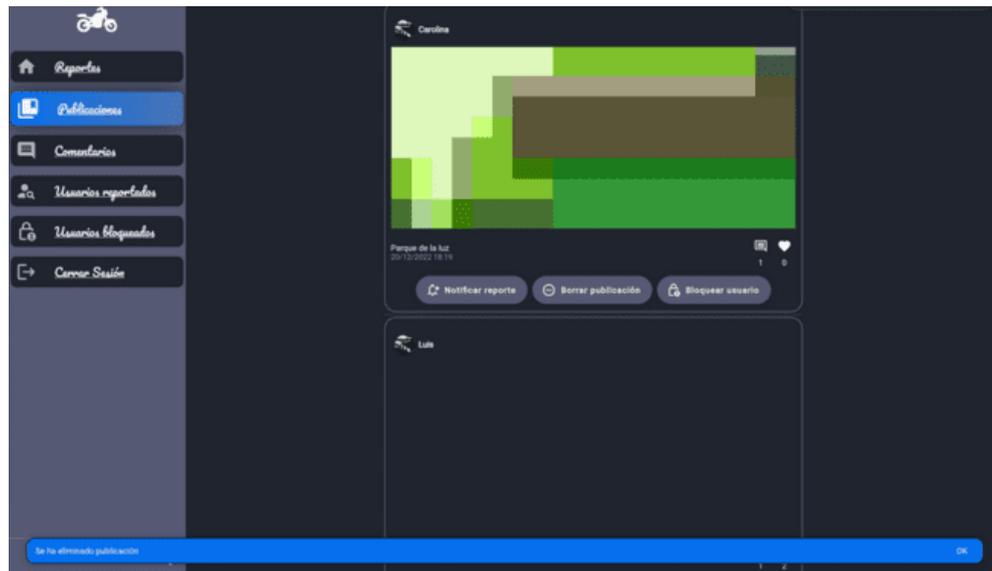
- Realizar notificación de reporte.



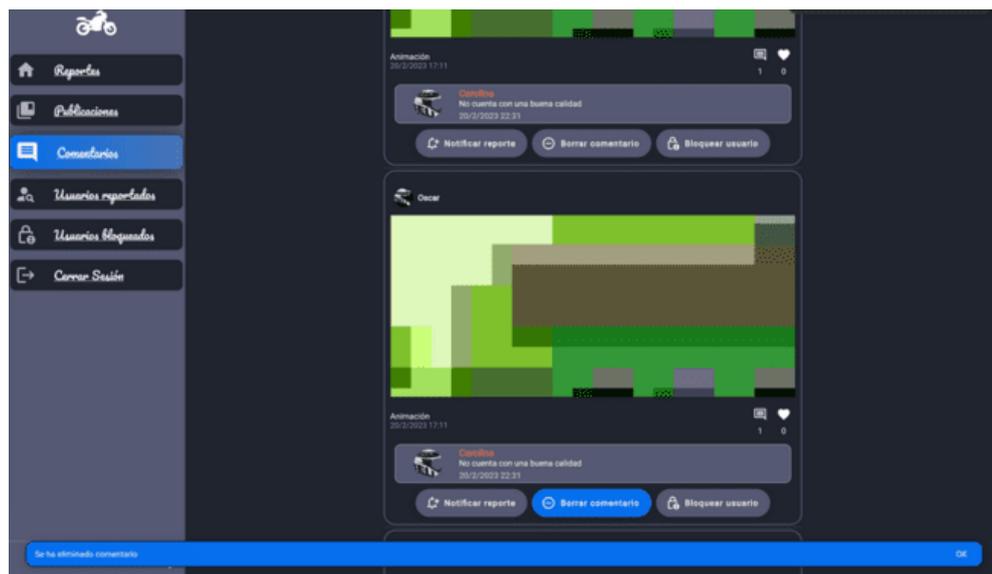
- Verificar los reportes como notificaciones en primer y segundo plano.



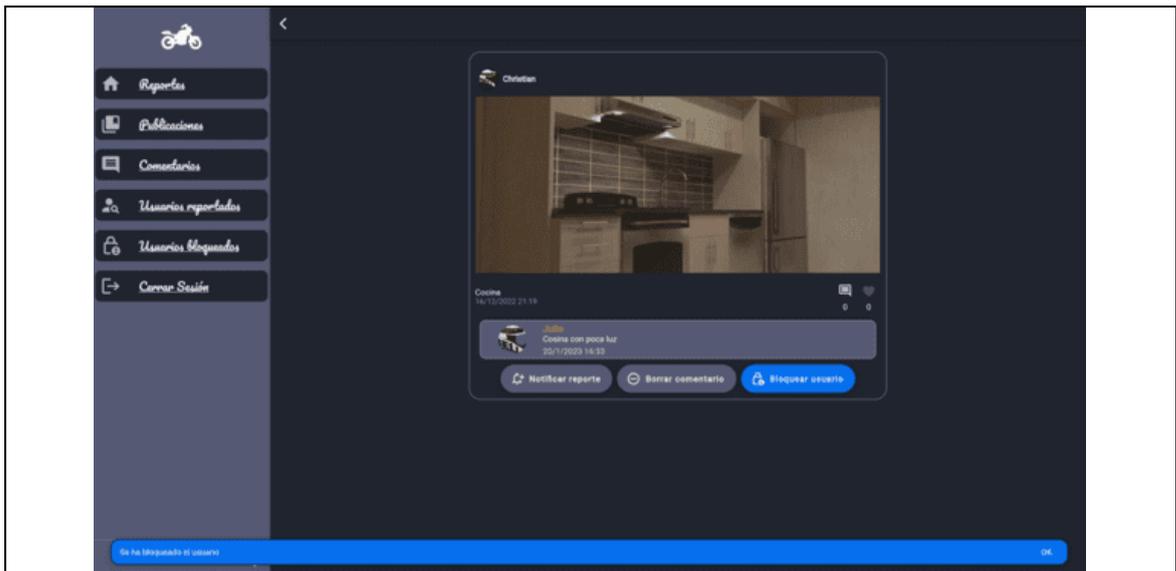
- Realizar eliminación de publicación.



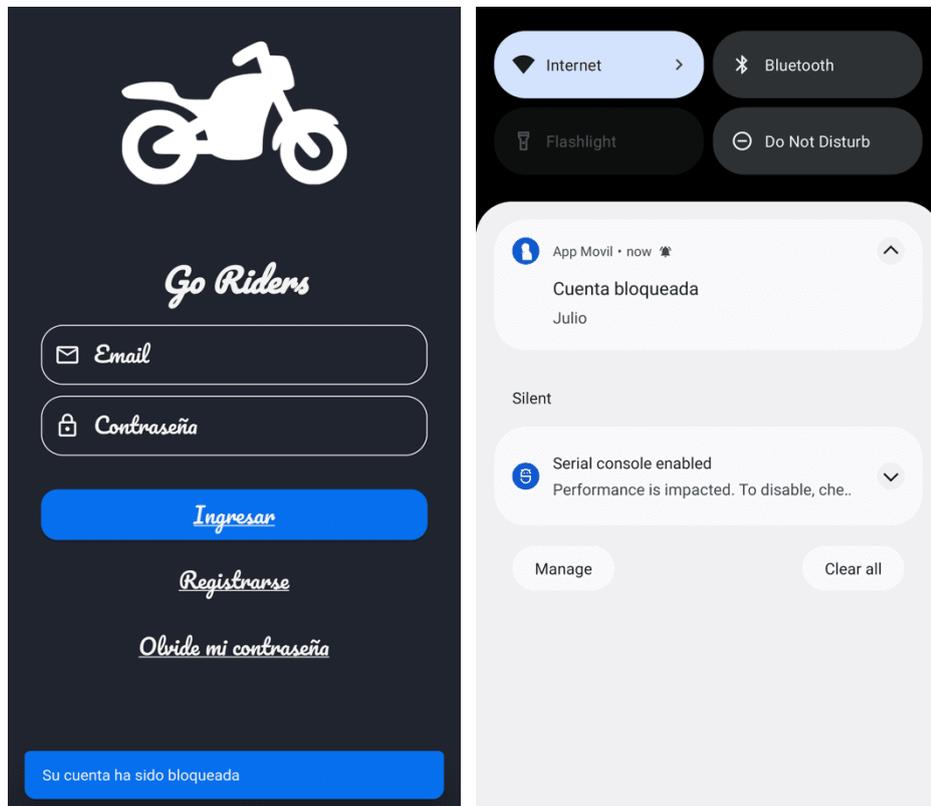
- Realizar eliminación de publicación.



- Realizar bloqueo de usuario.



- Verificar los mensajes como notificaciones en primer y segundo plano.



- Verificar el bloqueo de cuenta por parte del usuario.



Go Riders

✉ *julio@email.com*

🔒 •••••

Ingresar

Registrarse

Olvide mi contraseña

Su cuenta se encuentra bloqueada

OK

6.3. Análisis de resultados de la aplicación.

- Se ha realizado una aplicación para los usuarios motociclistas, con el cual les permita estar más conectados mediante noticias o publicaciones y el servicio de mensajería.
- Se ha logrado realizar la carga y control del perfil de usuario y publicaciones lo que permite tener un control de historial de las actualizaciones de perfil
- El apartado de rutas le permite al usuario compartir e iniciar viajes con la ayuda de GPS, también encontrar lugares de interés y ayuda.
- El módulo de perfil de motocicleta y mantenimiento le ayuda al motociclista a tener un control periódico acerca del cambio de aceite de su vehículo y así evitar complicaciones mecánicas futuras.
- Se ha realizado una aplicación con distribución para web, para los usuarios con roles de administrador.

Se ha cumplido con el objetivo general de crear aplicación dirigida a la comunidad motera que va a permitir un mejor desenvolvimiento en la ruta, también permite registrar controles mecánicos para evitar posibles fallas en el vehículo, además se implementó una red social específica para esta comunidad.

Se ha estudiado las diferentes tecnologías que permitieron el desarrollo y despliegue de la aplicación como lo son Dart y su framework Flutter, ExpressJs, Firebase, MongoDB y tecnologías relacionadas con el software nativo de Android y iOS que permitieron realizar configuraciones para el sistema operativo móvil que permitió cumplir con el primer objetivo específico.

Con respecto al objetivo número dos se ha realizado el análisis de requerimientos idóneos, el alcance de la aplicación y se ha realizado la investigación acerca de la

documentación de las tecnologías usadas desde la web del autor y la practica con ejemplos de este lo que permitió un avance a nivel de conocimiento para después un más simple desenvolvimiento al momento de desarrollar

En el objetivo tres se ha desarrollado la aplicación de acuerdo con los requerimientos planteados con un modelo de arquitectura de N-capas, lo que permitió conectar la aplicación con distintos servidores según las necesidades. Se ha utilizado el servicio gratuito de mapas de Google y MapBox que permite el manejo de mapas y rutas. También se ha desplegado el servidor de backend en el dominio de Digital Ocean, en modalidad de pago ya que solventaba la producción de solicitudes a comparación de modo gratuito.

Se ha realizado la validación mediante pruebas de resistencia mediante la conexión de dispositivos físicos y lógicos, tomando en cuenta dos aspectos, la carga a nivel de dispositivo físico según las características del teléfono y la carga en los servidores de solicitudes, lo que permitió conocer la cantidad de carga que pueden manejar los sistemas y se ha identificado los posibles problemas de rendimiento en caso de que la aplicación llegase a tener un gran alcance dentro de los dispositivos lógicos. Para finalizar con la documentación se ha realizado manuales de [instalación](#), [usuario](#) y [administrador](#).

7. CRONOGRAMA

7.1. Cronograma de actividades.

Descripción de actividades.

OE1. Estudiar los fundamentos del desarrollo de aplicaciones móviles multiplataforma y las tecnologías adecuadas.

Tabla 2.- Actividades del objetivo OE1.

No	Actividades.
1	Recolectar información de bibliotecas educativas.
2	Revisar la bibliografía obtenida.
3	Identificar la información relevante al tema de titulación.
4	Definir las herramientas para el desarrollo de la aplicación móvil.

OE2. Definir y documentar los requerimientos de la aplicación.

Tabla 3.- Actividades del objetivo OE2.

No	Actividades.
1	Definir los requerimientos con lluvia de ideas.
2	Definir las historias de usuarios con énfasis a los requerimientos.
3	Analizar y verificar los requerimientos funcionales y no funcionales de la aplicación móvil.
4	Documentar los requerimientos de la aplicación móvil

OE3. Desarrollar la aplicación móvil de acuerdo con los requerimientos definidos.

Tabla 4.- Actividades del objetivo OE3.

No	Actividades.
1	Desarrollar diagramas de estados.
2	Desarrollar diseño de bocetos de la aplicación móvil.
3	Diseñar la línea grafica de la aplicación móvil.
4	Codificar la aplicación móvil acorde al documento de requerimientos.

OE4. Validar el correcto funcionamiento de la aplicación por medio de ejecución con diferentes pruebas de software.

Tabla 5.- Actividades del objetivo OE4.

No	Actividades
1	Analizar la calidad de código de software
2	Establecer los casos de pruebas de la aplicación móvil.
3	Validar los casos de prueba de la aplicación móvil.
4	Monitorear y documentar el comportamiento de la aplicación móvil final.

OE5. Elaborar la documentación técnica de la aplicación móvil desarrollada.

Tabla 6.- Actividades del objetivo OE5.

No	Actividades
1	Documentar los procesos, métodos y servicios de la aplicación móvil.
2	Manual de instalación de la aplicación móvil.
3	Manual de administración de la aplicación móvil.
4	Manual de usuario de la aplicación móvil.

7.2. Tabla de actividades.

Tabla 7.- Fechas del cronograma de actividades.

Objetivo	Actividad	Horas	Duración días	Fecha Inicio	Fecha Fin
Obj. 1	Actividad 1	94	11	19/10/21	2/11/21
	Act 1,1	24	3	19/10/21	21/10/21
	Act 1,2	27	3	22/10/21	26/10/21
	Act 1,3	16	2	27/10/21	28/10/21
	Act 1,4	27	3	29/10/21	2/11/21
Obj. 2	Actividad 2	90	10	3/11/21	16/11/21
	Act 2,1	27	3	3/11/21	5/11/21
	Act 2,2	27	3	8/11/21	10/11/21
	Act 2,3	18	2	11/11/21	12/11/21
	Act 2,4	18	2	15/11/21	16/11/21
Obj. 3	Actividad 3	450	50	17/11/21	27/01/22
	Act 3,1	45	5	17/11/21	23/11/21
	Act 3,2	81	9	24/11/21	6/12/21
	Act 3,3	27	3	7/12/21	9/12/21
	Act 3,4	297	33	10/12/21	27/01/22
Obj. 4	Actividad 4	88	11	28/01/22	11/06/22

	Act 4,1	24	3	28/01/22	10/02/22
	Act 4,2	16	2	11/02/22	03/04/22
	Act 4,3	24	3	04/04/22	28/04/22
	Act 4,4	24	3	29/04/22	11/06/22
Obj. 5	Actividad 5	78	10	11/06/22	25/02/23
	Act 5,1	27	3	11/06/22	16/07/22
	Act 5,2	27	3	17/07/22	21/08/22
	Act 5,3	12	2	22/08/22	03/11/22
	Act 5,5	12	2	04/11/22	25/02/23

8. PRESUPUESTO

Las tecnologías que se utilizaran pertenecen a la licencia Open Source. Dentro del proyecto los sistemas operativos que se utilizaran están basados en Windows y Mac OS, que permitirá el desarrollo del proyecto. Así también se desarrollará la aplicación móvil con simuladores de Android y iOS con el fin de reducir coste, para el desarrollar la aplicación web se utiliza la configuración del navegador Chrome, la herramienta para desarrollo es Visual Studio Code.

Tabla 8.- Presupuesto

Descripción	Cantidad	Costo Unitario	Costo Total
Bienes Tecnológicos			
Cursos Académicos Online	4	\$100	\$400
MacBook Pro-2018	1	\$1500	\$1500
Acer Predator 2019	1	\$1000	\$1000
Samsung A32	1	\$ 210	\$ 210
Servidor Digital Ocean	5	\$6	\$30
Personal			
Asesoría Especializada	1	\$200	\$200
Otros			
Imprevistos	1	\$100.00	\$100.00
total			\$3440.00

9. CONCLUSIONES

La aplicación desarrollada fue creada utilizando el framework de Flutter, lo que permitió su desarrollo multiplataforma para su uso en dispositivos móviles con diferentes sistemas operativos y también la administración mediante una página web realizada con el mismo framework. Además, se utilizó un servidor en la nube para proporcionar servicios y almacenamiento de datos, lo que mejoró la escalabilidad y el rendimiento de la aplicación.

El proyecto ofrece una solución efectiva para mejorar la planificación y el seguimiento de rutas de motocicleta, lo que puede mejorar significativamente la seguridad y la eficiencia de la conducción. Además, la aplicación permite una gestión más eficaz del mantenimiento de las motocicletas al programar recordatorios para inspecciones y reparaciones, lo que puede prolongar la vida útil de las motocicletas y reducir los costos asociados. La inclusión de estas funciones adicionales puede mejorar aún más la experiencia del usuario y la seguridad en la conducción. También se ofrece una red social dirigida a comunidades motociclistas, con funciones basadas a brindar comunicación, sugerir rutas o contenido de una ruta por medio de publicaciones y encontrar miembros de la comunidad, como último punto de brinda una página administrativa, con el objetivo de controlar el contenido y comentarios que se son realizados en una publicación.

Los resultados del estudio indican que la aplicación es fácil de usar y ofrece una amplia gama de funciones que satisfacen las necesidades de diversos usuarios, desde conductores individuales hasta empresas de servicios de transporte. Además, el desarrollo de la aplicación ha permitido identificar áreas de mejora en la gestión de rutas de motocicleta, mantenimiento, creación de nuevos usuarios con diferentes roles como ejemplos mecánicos, la creación de una moto tienda y en la red social brindar más contenido, lo que puede guiar futuras investigaciones y desarrollos en este campo.

En conclusión, este trabajo demuestra cómo la tecnología puede ser utilizada para mejorar la seguridad y eficiencia en la conducción de motocicletas. La utilización de un framework multiplataforma y la implementación de un servidor en la nube permiten la creación de soluciones escalables y adaptables a diferentes necesidades. Estas conclusiones pueden ser valiosas para futuras investigaciones y desarrollos en este campo y en otros relacionados con la movilidad.

10. RECOMENDACIONES

- La arquitectura tecnológica de aplicaciones distribuidas multicapa permite la escalabilidad del proyecto a futuro de manera sencilla, además de reducir los costos económicos. Se recomienda desplegar el servidor API REST con todas las funcionalidades del proyecto y gracias a los servicios en la nube se permite una mejor manera de utilizar los recursos tecnológicos.
- Al desarrollar la aplicación móvil se debe que tomar en cuenta que los componentes utilizados puedan ser compatibles con la mayoría de las versiones del sistema operativo, y así la aplicación tenga un mayor alcance con los usuarios.
- Al realizar el diseño de la aplicación, se deberá analizar aspectos sobre la navegabilidad y usabilidad ya que ayuda a mejorar la experiencia del usuario.
- Es importante diseñar un prototipo estético donde se establezcan los elementos a usarse en el proyecto lo que permitirá ayudar a materializar las ideas.
- Después de visualizar los resultados obtenidos, se recomienda los siguientes requerimientos para futuros trabajos dentro del proyecto como por ejemplo, realizar y visualizar paradas dentro de las rutas, calificaciones de rutas, creación de profesionales (mecánicos) para el control del mantenimiento, etc.

REFERENCIAS BIBLIOGRÁFICAS

- Earth, G. (2022). Obtenido de <https://earth.google.com/>
- Bernal González, D. (Abril de 2021). *Profile*. Obtenido de Principales tipos de aplicaciones móviles: ventajas, desventajas y ejemplos: <https://profile.es/blog/tipos-aplicaciones-moviles-ventajas-ejemplos/>
- Genexus. (Enero de 2016). *Genexus*.
- Visual Studio Code. (Octubre de 2021). *Visual Studio Code*. Obtenido de <https://code.visualstudio.com>
- Fernández, Y. (Octubre de 2019). *Xakata*. Obtenido de <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>
- Udemy. (Octubre de 2021). *Udemy*. Obtenido de <https://www.udemy.com>
- Dart. (Octubre de 2021). Obtenido de <https://dart.dev/overview>
- Microsoft. (Mayo de 2021). *Azure DevOps*. Obtenido de <https://docs.microsoft.com/es-es/devops/develop/git/what-is-git>
- Google Firebase. (s.f.). *Firebase*. Obtenido de <https://firebase.google.com/docs/cloud-messaging?hl=es-419>
- RedHat. (2020). Obtenido de <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- flutter. (2022). *Flutter*. Obtenido de <https://docs.flutter.dev/get-started/install>
- Flutter. (2022). *flutter*. Obtenido de <https://docs.flutter.dev/development/tools/devtools/overview>
- Firebase. (2022). *Firebase*. Obtenido de https://firebase.google.com/docs/cloud-messaging/?authuser=0#how_does_it_work
- android. (2022). *Developer android*. Obtenido de <https://developer.android.com/studio/intro?hl=es-419>
- Group, T. P. (2022). *PHP*. Obtenido de <https://www.php.net/manual/es/intro-what-is.php>
- Statcounter. (s.f.). Obtenido de <https://statcounter.com/>
- Molina, D. (04 de 2020). *BBVA Next Technologies*.
- DocsFlutter. (2022). *Flutter Docs*. Obtenido de <https://flutter-es.firebaseio.com/docs/development/ui/layout>
- Windmill, E. (2019). *Flutter in Action*. Manning.
- MxAbc. (2018). *GitHub Gestion de copias de seguridad*. Obtenido de <https://github.com/MxABC/git-sourcetree>
- Bustos, G. (2023). *Hostinger*. Obtenido de <https://www.hostinger.es/tutoriales/que-es-github#:~:text=Esto%20se%20debe%20a%20que,control%20de%20versiones%20de%20Git.>
- Aguilera, I. (2016). *Lo que estaba por llegar yaa está aquí*. La esfera de los libros.
- Torres Remon, M. (2017). *DESARROLLO DE APLICACIONES MÓVILES CON ANDROID*. Marcombo.
- Waze. (s.f.). *Waze*. Obtenido de <https://riderchris.com/es/waze-for-motorcycle/>

Rever. (s.f.). *BMW Motorrad*. Obtenido de <https://www.bmw-motorrad.bo/es/experience/stories/tour/rever/rever-app.html>

IBM Cloud. (Abril de 2021). Obtenido de <https://www.ibm.com/es-es/cloud/learn/rest-apis>

Greg, S. (s.f.). *Hosting Victory*. Obtenido de <https://hostingvictory.com/>

Digital Ocean. (2023). *Digital Ocean Docs*. Obtenido de <https://docs.digitalocean.com/products/getting-started/>

ANEXOS

ANEXO 1 – ENCUESTAS REALIZADAS

Antes de realizar el proyecto, era importante tomar en cuenta la opinión por parte de los usuarios motociclistas, para a realizar el levantamiento requerimientos funcionales o no funcionales de la aplicación, para lo cual se optó realizar una encuesta, las cuales fueron enviadas a grupos de motociclistas, con un total de 42 respuestas, 12 más de las esperadas.

Dentro de la encuesta, se formularon preguntas de opciones y respuesta, las cuales se detalla con las siguientes ilustraciones:

- 1.1.** En la mayoría de los usuarios encuestados, se visualiza que desconocen sobre información relaciona a una aplicación dirigía a los motociclistas.

¿Conoce o usa alguna aplicación que esté dirigida a usuarios de motocicletas?

42 respuestas

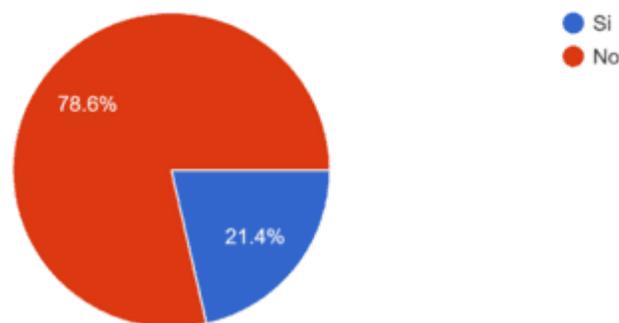


Ilustración A1 1.- Porcentaje del conocimiento de aplicaciones para motociclistas

- 1.2.** Se encontraron 9 respuestas con el nombre de algunas aplicaciones que nos permitió investigar los requerimientos no funcionales

En caso de haber respondido que "Sí" en la anterior pregunta, ¿podría nombrar la/las aplicación/es?

9 respuestas

Motocali
drivvo, calimoto
locatoweb, weRide, google maps, Drivvo, maps.me wikiloc
Calimoto, Wikiloc
Maps
Braappmotos
Relieve
Calimoto,

Ilustración A1 2.- Listado de aplicaciones dirigidas a motociclistas

1.3. La siguiente pregunta se realizó, para saber a qué usuarios nos estamos dirigiendo, tomando en cuenta que por lo general las motos de mayor cilindrada son ocupadas para viajar, más no tanto para la ciudad

¿Cuál es la cilindrada de su motocicleta?

42 respuestas

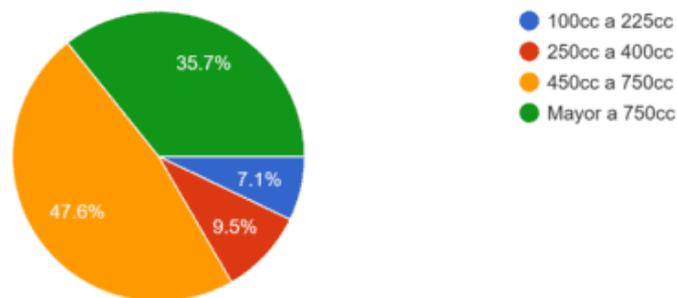


Ilustración A1 3.- Porcentaje de capacidad de cilindraje

1.4. La siguiente pregunta, se obtuvo una mejor definición de los requerimientos funcionales a realizar en la ejecución de esta aplicación, como se observa en la información de los usuarios que requieren asistencia vial, como por ejemplo servicios mecánicos, médicos, localización de gasolineras y servicios de auxilio.

¿Qué servicios de emergencia requiere usted en una ruta?

42 respuestas

Grúa
Ambulancias
Asistencia Mecanica y/o médica
GPS ambulancia / mecánico
fallas mecánicas
Accidentes daños mecánicos
mecánico, salud
Gasolina, algún repuesto
Remolque o mecánica rápida

Ilustración A1 4.- Listado de asistencia vial

- 1.5.** La siguiente pregunta, define los servicios necesarios que debe contener esta aplicación, los que más destacaron fueron control de ubicaciones, control de rutas, recomendaciones de rutas.

¿Qué servicio le gustaría dentro de una aplicación que permita facilitar sus rutas?

42 respuestas

Gps
Gasolineras, restaurantes, hoteles, hospitales
Me gustaría algo como un botón de emergencia con diferentes sentidos, en casa de accidente, mecánico o algo por el estilo
Hospedaje
Tipos de caminos, on road, off road
ver rutas con caminos alternos a una via principal ej. offroad
Conectividad sin servicio de telefonía
GPS, rutas guardadas,
Que me Guie

Ilustración A1 5.- Listado de servicios en rutas

- 1.6.** Dentro de los requerimientos planteados, la aplicación debería mostrar rutas y sugerir destinos, y se realizó la siguiente pregunta, para tener conocimiento si a más usuarios les

gustaría dando una respuesta favorable.

¿Le gustaría que una aplicación sugiera destinos?

42 respuestas

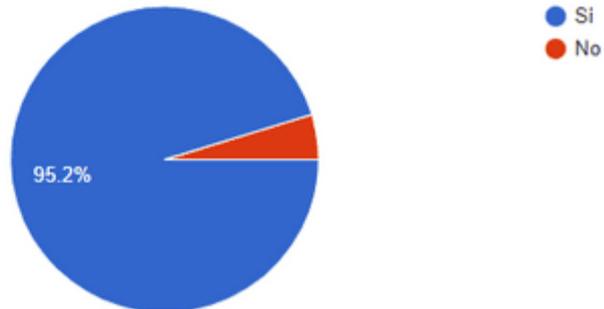


Ilustración A1 6.- Porcentaje de sugerencia de destinos

- 1.7. La siguiente pregunta, se realizó conocer la necesidad de los usuarios de recordar o tener presente cuando realizar el cambio de aceite y así prevenir daños futuros mecánicos dando una respuesta favorable.

¿Le gustaría contar con una aplicación que le ayude a llevar el control para el mantenimiento de su motocicleta? Ejemplo: Aproximación de fecha para cambio de aceite.

42 respuestas

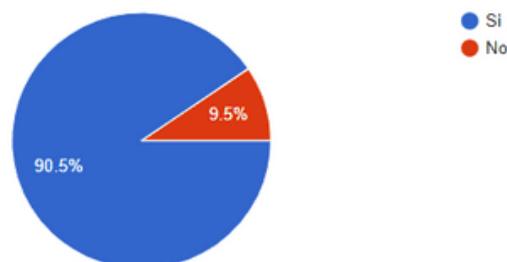


Ilustración A1 7.- Porcentaje de control de mantenimiento de una motocicleta

En base a la información obtenido por las preguntas realizadas, se definió los requerimientos a desarrollar en el presente proyecto.

Lista de Ilustraciones Anexo 1

Ilustración A1 1.- Porcentaje del conocimiento de aplicaciones para motociclistas	110
Ilustración A1 2.- Listado de aplicaciones dirigidas a motociclistas	111
Ilustración A1 3.- Porcentaje de capacidad de cilindraje	111
Ilustración A1 4.- Listado de asistencia vial	112
Ilustración A1 5.- Listado de servicios en rutas	112
Ilustración A1 6.- Porcentaje de sugerencia de destinos	113
Ilustración A1 7.- Porcentaje de control de mantenimiento de una motocicleta	113

ANEXO 2 – MANUAL DE INSTALACIÓN

Manual de instalación del framework Flutter, con las características del lenguaje Dart para el desarrollar y ejecutar correcta de la aplicación realizada.

Flutter

Para desarrollar y ejecutar Flutter es necesario que su entorno de desarrollo cuente con los siguientes requerimientos mínimos, de los sistemas operativos.

<i>Sistemas Operativos</i>	<i>Windows</i>	<i>MacOS</i>	<i>Linux</i>	<i>Chrome OS</i>
<i>Versiones</i>	Windows 10 o posteriores de 64 bits	MacOS	Linux de 64 bits	Chrome OS de 64 bits
<i>Espacio de disco</i>	1.64 GB	1.8 GB	600 MB	600 MB
<i>Herramientas</i>	<ul style="list-style-type: none"> • Windows • PowerShell 5 • Git 	<ul style="list-style-type: none"> • XCode • Git 	<ul style="list-style-type: none"> • Bash • Curl • File • Git • Mkdir • Rm • Unzip • Which • Xz-utils • zip 	<ul style="list-style-type: none"> • Bash • Curl • Git • Mkdir • Rm • Unzip • Which

Tabla A2 1.- Requerimiento mínimos de los sistemas operativos para Flutter (flutter, 2022)

Para el desarrollo de este proyecto de Flutter, se instalado en los siguientes sistemas operativos, Windows 10 y macOS Monterey, con el objetivo de compilar en los dispositivos de Android y iOS, para descargar el instalador de cada sistema operativo se recomienda el siguiente link flutter.dev.

A continuación, se especifica las herramientas necesarias para el funcionamiento correcto del entorno de desarrollo.

Flutter SDK

Windows

Agregar Flutter a una variable de entorno PATH, ingresar a Editar variables de entorno, desde la barra de búsquedas ingresando “env”, una vez dentro de entorno de variables, seleccionar editar la variable “Path”, agregar una nueva variable de entorno, donde debe contar la dirección completa y exacta a flutter\bin posteriormente instalada.

```
C:\flutter\bin
```

macOS

Agregar la configuración de Flutter en “Path” desde el terminal ya sea bash o zsh, ingresar a la carpeta del usuario.

```
cd /User/ “Nombre usuario”
```

Agregar en el archivo .zshrc o .bashrc la siguiente línea de comando, con la dirección completa y exacta a flutter/bin.

```
export PATH = “$PATH: flutter/bin”
```

```
~ flutter --version
Flutter 3.0.1 • channel stable • https://github.com/flutter/flutter.git
Framework • revision fb57da5f94 (5 months ago) • 2022-05-19 15:50:29 -0700
Engine • revision caaafc5604
Tools • Dart 2.17.1 • DevTools 2.12.2
```

Ilustración A2 1.- Verificar la versión de Flutter

Android Studio

Descargar el instalador de la fuente oficial del siguiente link [android.com](https://developer.android.com/studio), una vez instalador en los dos sistemas operativos, es necesario agregar y verificar la configuración de Android SDK.

Ingresar a la administración de SDK.

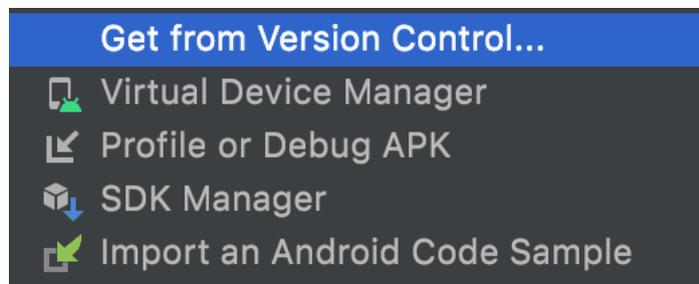


Ilustración A2 2.- Opciones de Android Studio

Seleccionar en el menú la opción, configuración del SDK, a continuación revisar que conste seleccionados todas las opciones de Android SDK.

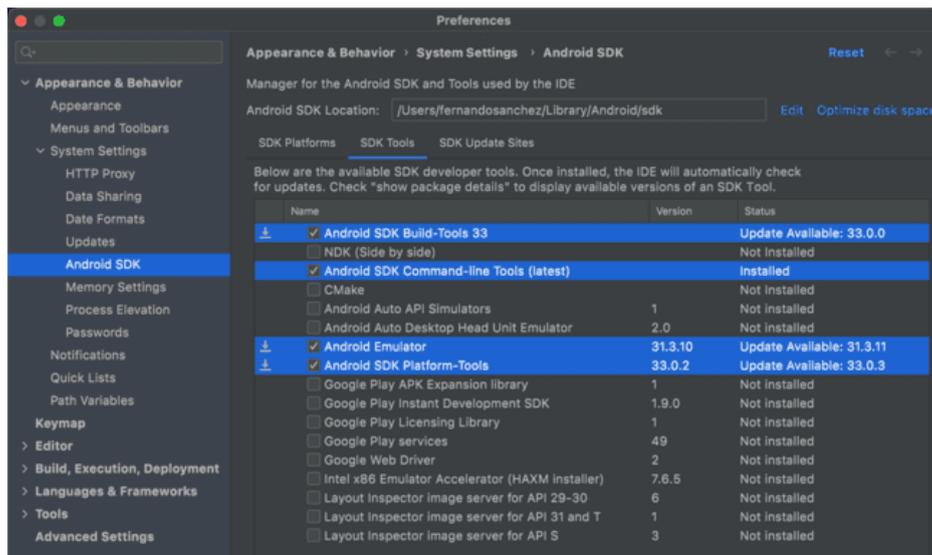


Ilustración A2 3.- Preferencias de Android SDK

Aceptar los términos de la licencia de Android con Flutter, con el siguiente comando:

```
flutter doctor --android-licenses
```

XCode

Descargar el programa de la App Store, una vez instalado con sus respectivas librerías configurar las herramientas de línea de comandos, ejecutando los siguientes comandos:

```
sudo xcode-select --switch
/Applications/Xcode.app/Contents/Developer
sudo xcodebuild -runFirstLaunch
```

Aceptar los términos de la licencia de XCode, confirmando o ejecutando los siguientes comandos:

```
sudo xcodebuild -license
```

IDE de desarrollo VS Code

En este proyecto se eligió IDE de desarrollo VS Code, por brindar una mejor

adaptación y ejecución de la aplicación a realizar, cada recalcar que también se puede utilizar Android Studio como IDE de desarrollo. Ambos necesitan instalar la extensión de Dart para ejecutar Flutter. La instalación de VS Code, se recomienda descargar el instalador de la fuente oficial del siguiente enlace [visualstudio](https://visualstudio.com),

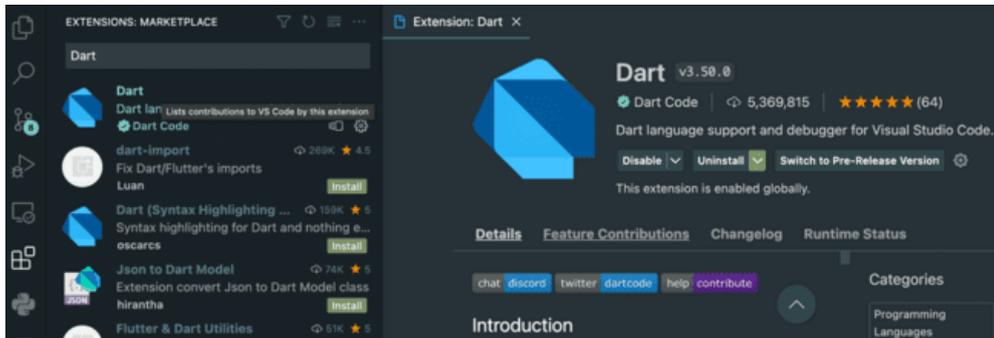


Ilustración A2 4.- Instalación de la extensión Dart en VS Code

Antes de continuar comprobar el correcto funcionamiento de Android SDK, XCode y VS Code.

```
flutter doctor
```

```
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.0.1, on macOS 12.6 21G115 darwin-x64, locale es-EC)
[✓] Android toolchain - develop for Android devices (Android SDK version 32.1.0-rc1)
[✓] Xcode - develop for iOS and macOS (Xcode 14.0.1)
[✓] Chrome - develop for the web
[✓] Android Studio (version 2021.2)
[✓] VS Code (version 1.71.2)
[✓] Connected device (2 available)
[✓] HTTP Host Availability
```

Ilustración A2 5.- Comprobar instalación de Android SDK y XCode con Flutter Doctor

Lista de Ilustraciones Anexo 2

Ilustración A2 1.- Verificar la versión de Flutter	117
Ilustración A2 2.- Opciones de Android Studio	117
Ilustración A2 3.- Preferencias de Android SDK.....	118
Ilustración A2 4.- Instalación de la extensión Dart en VS Code.....	119
Ilustración A2 5.- Comprobar instalación de Android SDK y Xcode con Flutter Doctor	119

Lista de Tablas Anexo 2

Tabla A2 1.- Requerimiento mínimos de los sistemas operativos para Flutter (flutter, 2022).....	116
---	-----

ANEXO 3 – MANUAL DE USUARIO

Ventana Principal

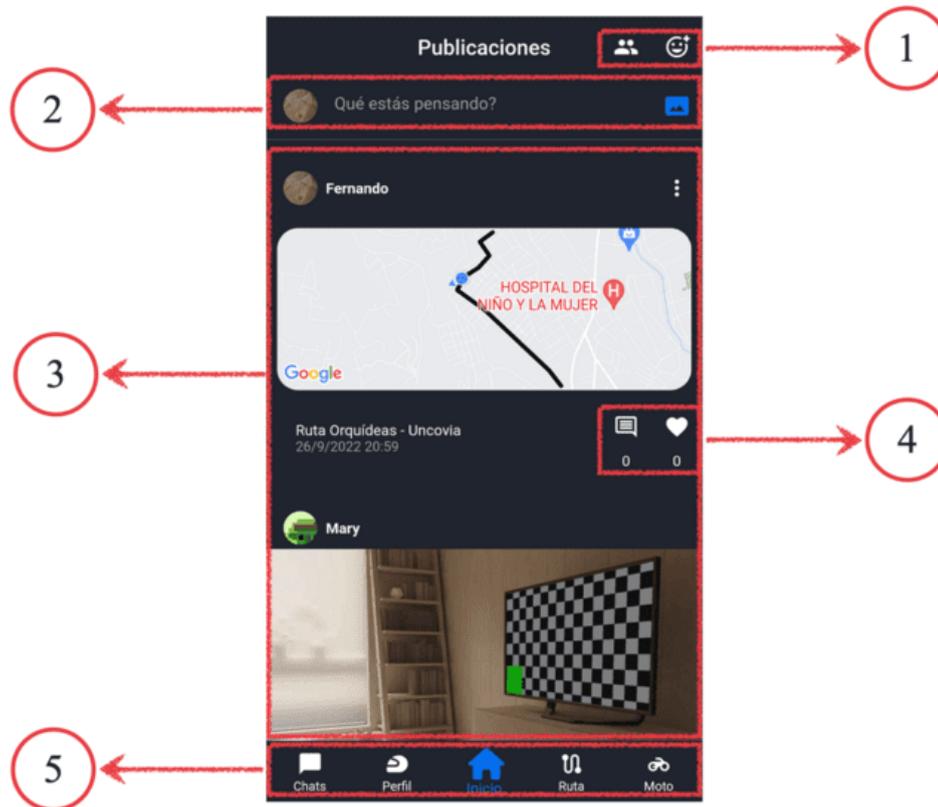


Ilustración A3 1.- UI Pantalla principal

- 1) Botones de opciones para buscar amigos o visualizar solicitudes de amistad.
- 2) Zona para crear publicaciones.
- 3) Zona de publicaciones.
- 4) Botones de opciones para comentar o reaccionar a la publicación.
- 5) Barra de menú con botones opcionales.

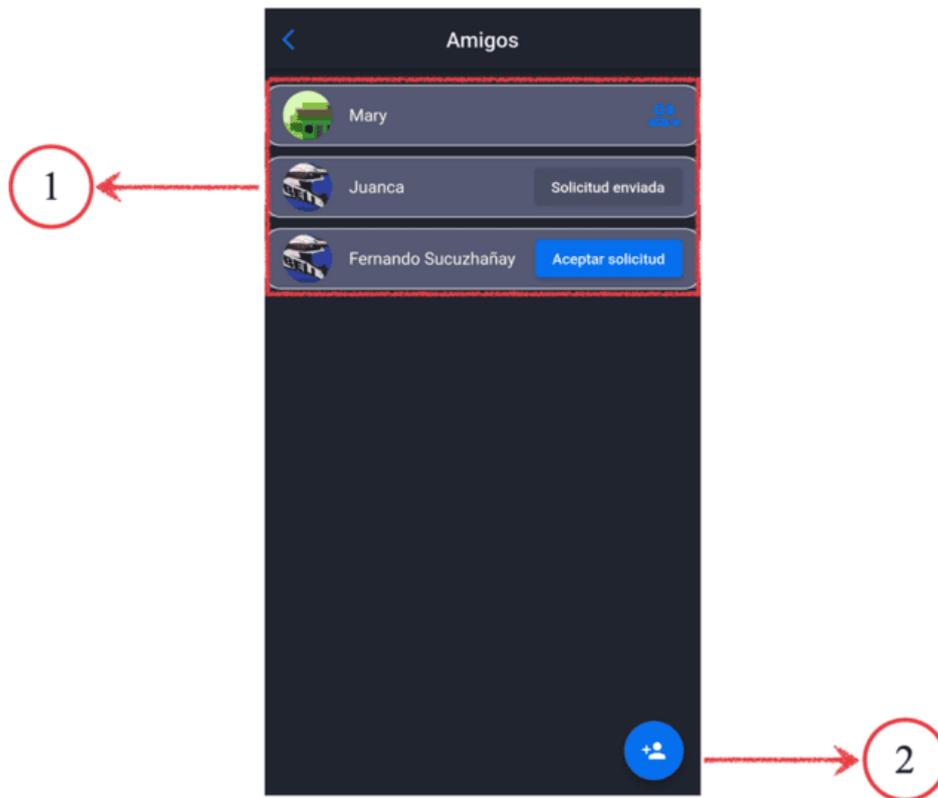


Ilustración A3 2.- UI Listado de usuarios

- 1) Zona de listado de amigos, solicitudes de amistades enviadas o recibidas.
- 2) Botón de añadir un nuevo amigo.

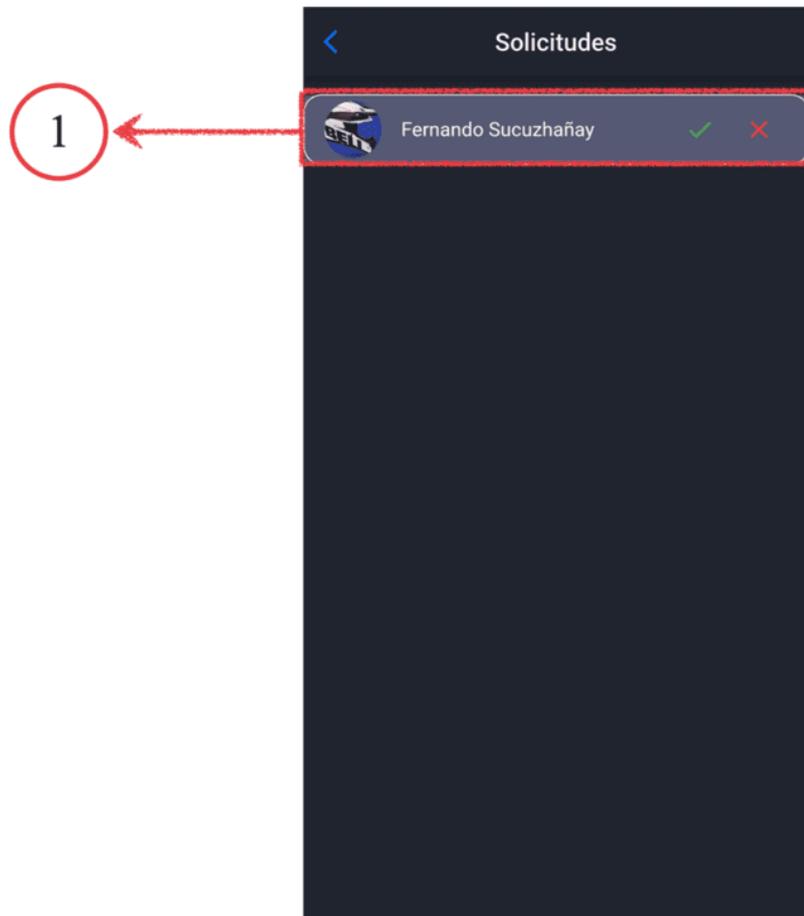


Ilustración A3 3.- UI Listado de solicitudes de amistad

- 1) Botones opcionales de aceptar o rechazar solicitud de amistad.

Ventana Principal – Rutas

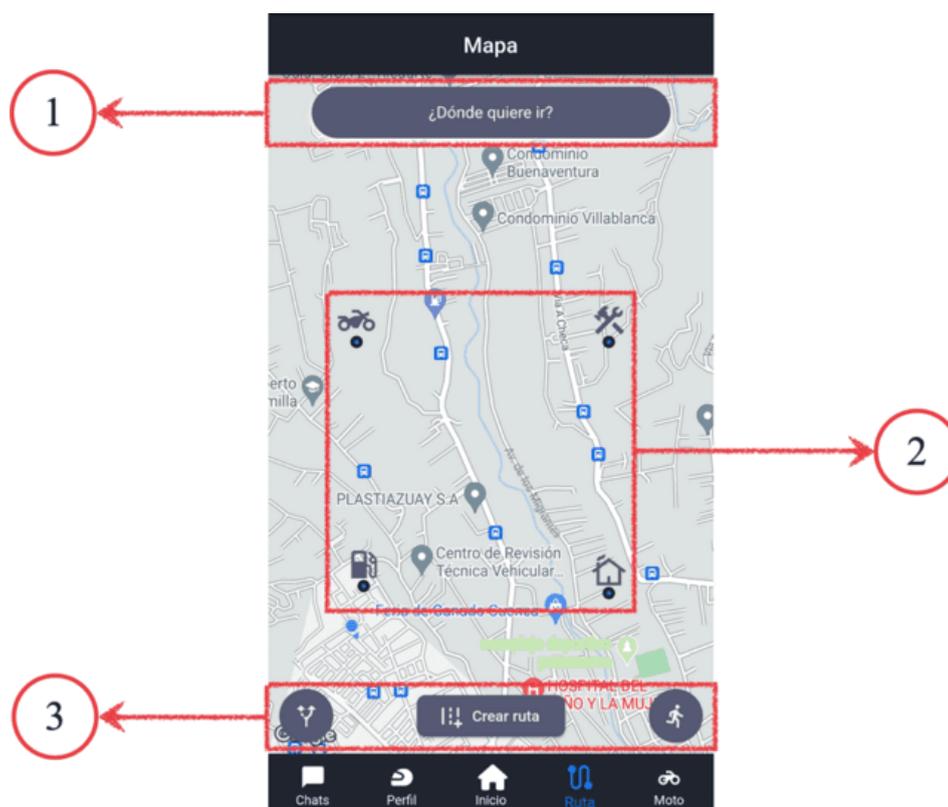


Ilustración A3 4.- UI Rutas

- 1) Botón de búsqueda a una ubicación mediante dirección o manualmente.
- 2) Zona de marcadores personalizados y privados.
- 3) Barra de opciones, visualizar, crear rutas y volver a la posición inicial.

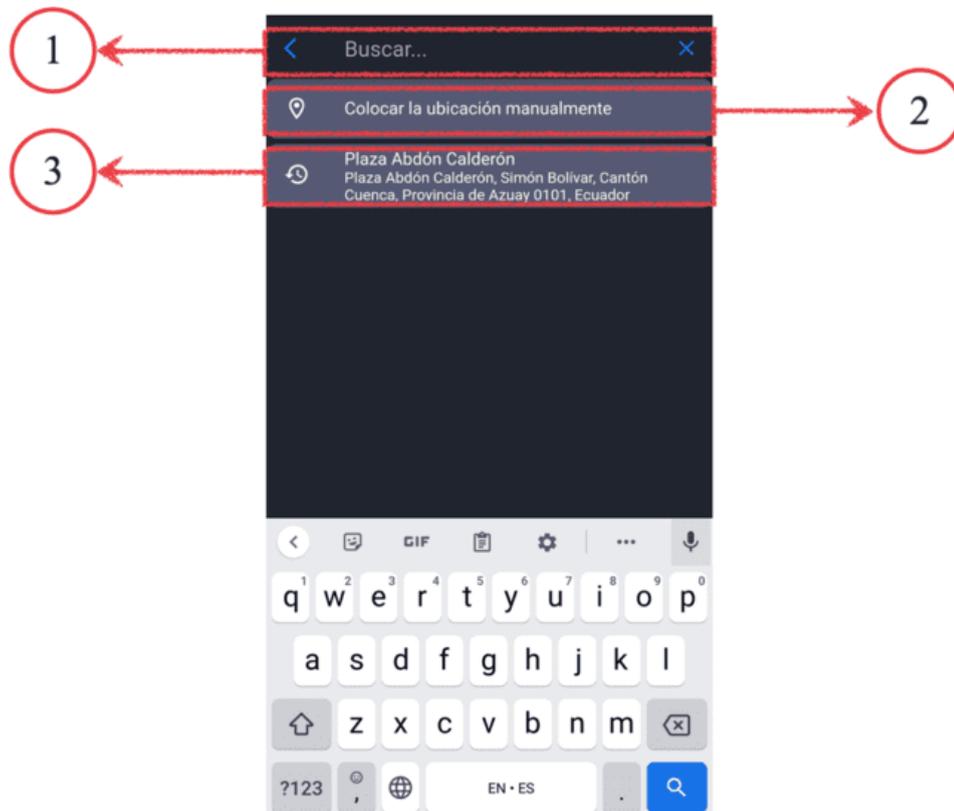


Ilustración A3 5.- UI Listado de historial de rutas

- 1) Barra de búsqueda.
- 2) Botón de ubicación manualmente.
- 3) Zona de historial de búsquedas realizadas.

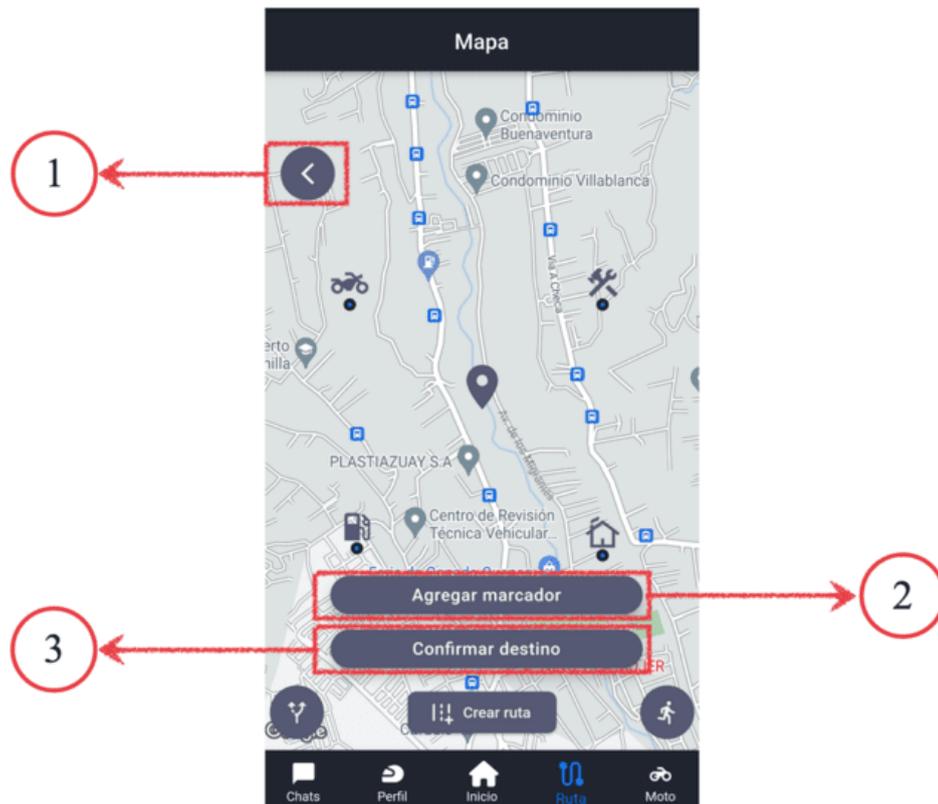


Ilustración A3 6.- UI Opciones de ubicación manualmente

- 1) Botón de cancelar ubicación manualmente.
- 2) Botón de agregar marcador personalizado.
- 3) Botón de confirmar la ubicación de destino.

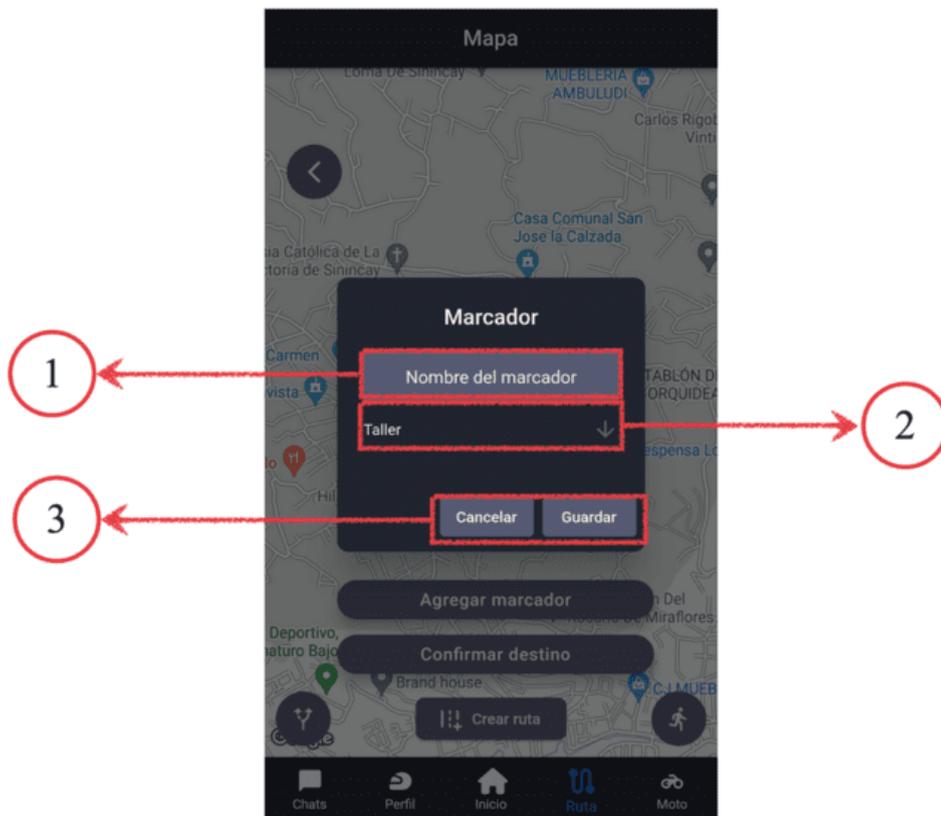


Ilustración A3 7.- UI Alerta de dialogo para guardar marcador

- 1) Caja de texto para definir el nombre del marcador.
- 2) Botón opcional, tipo de marcador.
- 3) Botón de cancelar o guardar marcador.

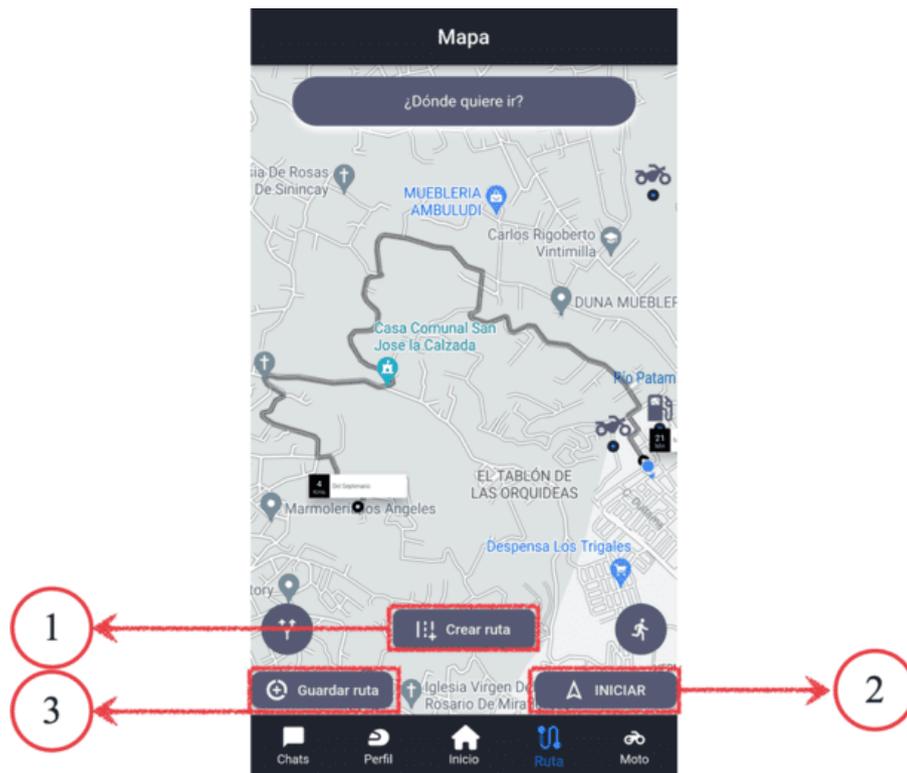


Ilustración A3 8.- UI Opciones crear, visualizar e iniciar ruta

- 1) Botón de crear ruta.
- 2) Botón de realizar ruta con Google Maps.
- 3) Botón de guardar ruta.

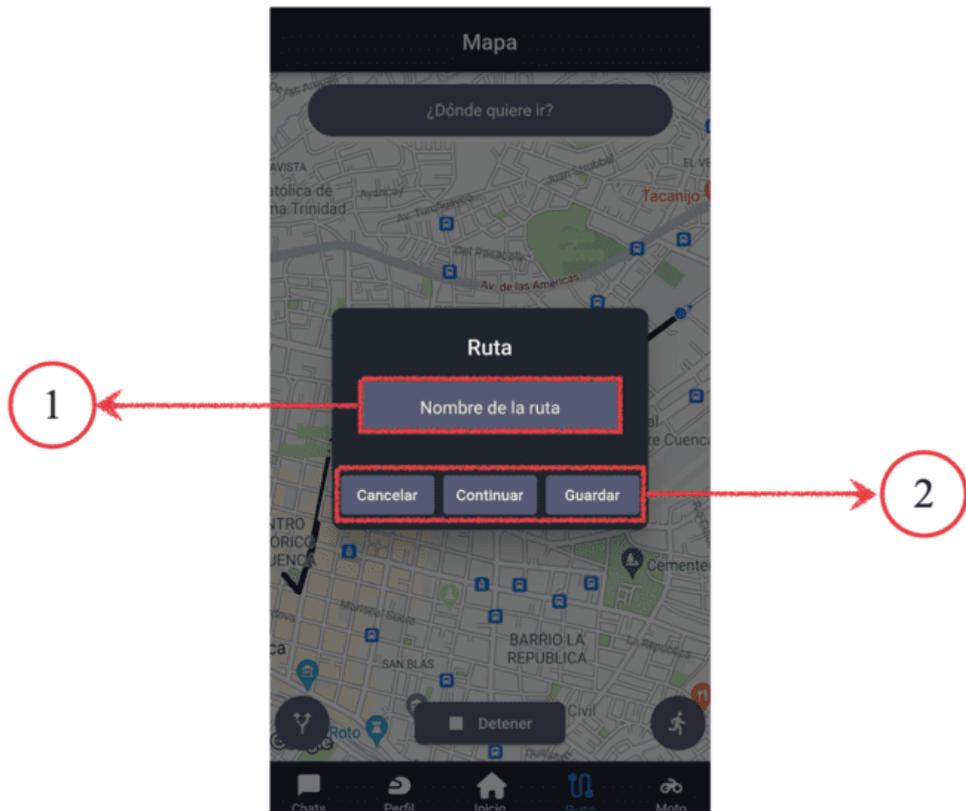


Ilustración A3 9.- UI Alerta de dialogo para guardar ruta

- 1) Caja de texto para definir el nombre de la ruta.
- 2) Botones opcionales de cancelar, continuar o guardar la ruta.



Ilustración A3 10.- UI Listado de rutas

- 1) Zona de ruta.
- 2) Botón opcional de eliminar o publicar ruta.

Ventana Principal – Motocicleta

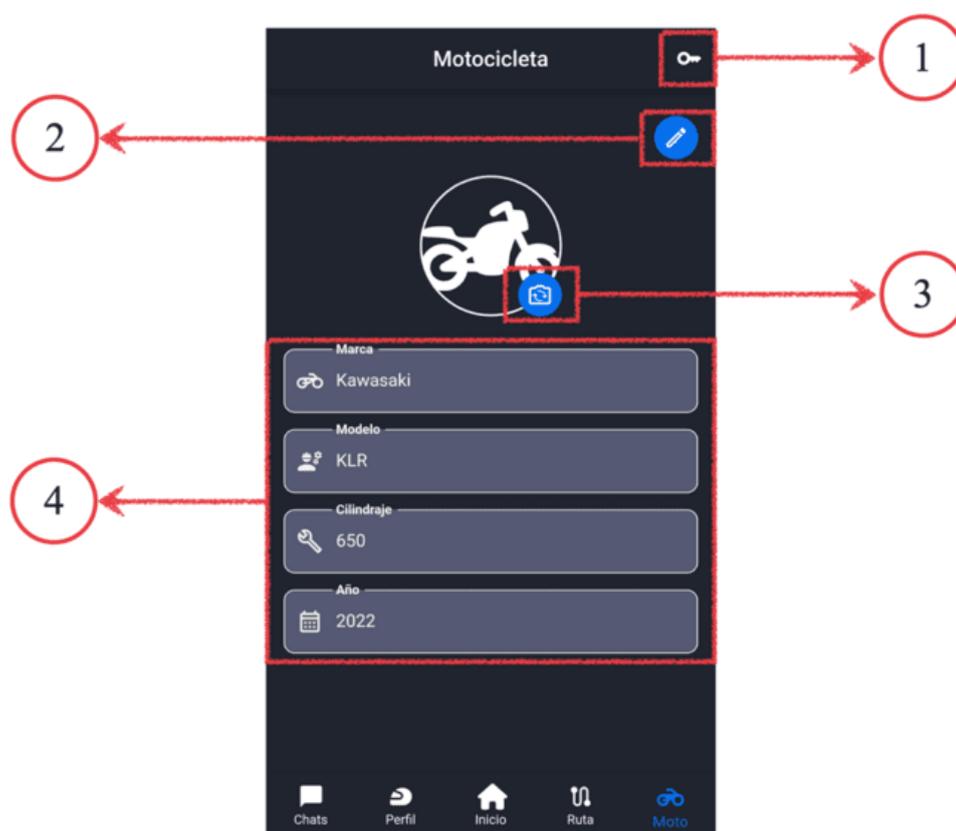


Ilustración A3 11.- UI Motocicleta

- 1) Botón de visualizar mantenimiento.
- 2) Botón de editar datos de la motocicleta.
- 3) Botón de editar foto de la motocicleta.
- 4) Zona de datos de la motocicleta.

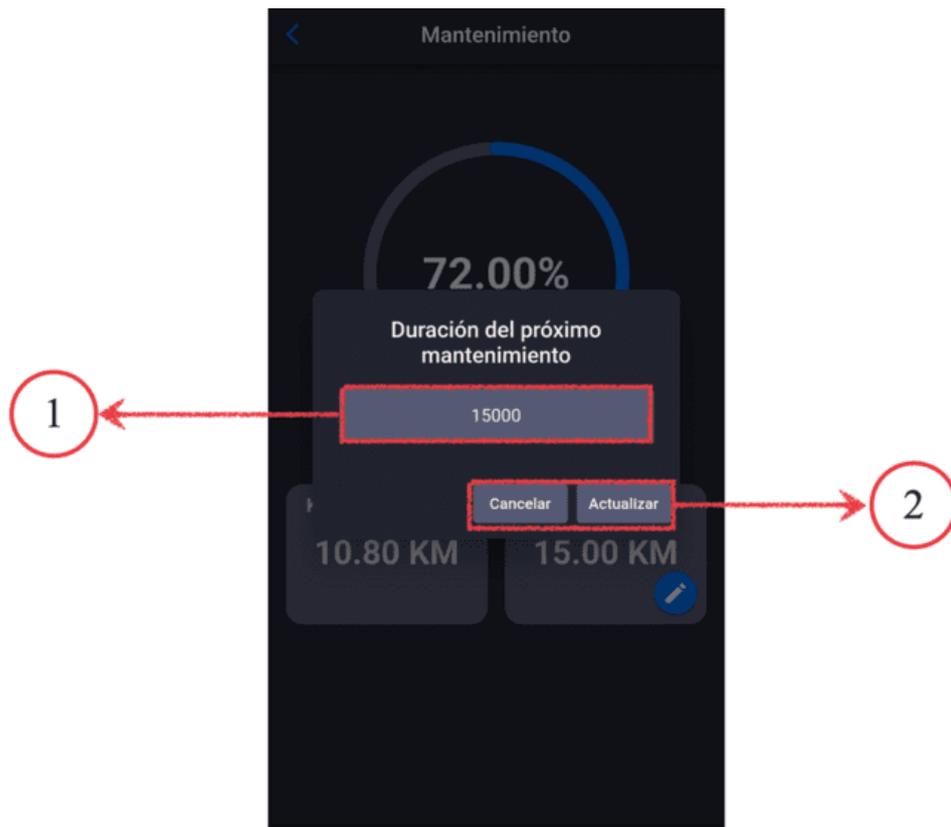


Ilustración A3 12.- UI Alerta de dialogo para actualizar kilometraje de mantenimiento

- 1) Caja de texto para ingresar la duración del cambio de aceite en kilómetros.
- 2) Botones opcionales de cancelar o actualizar el cambio de aceite.

Ventana Principal – Chat

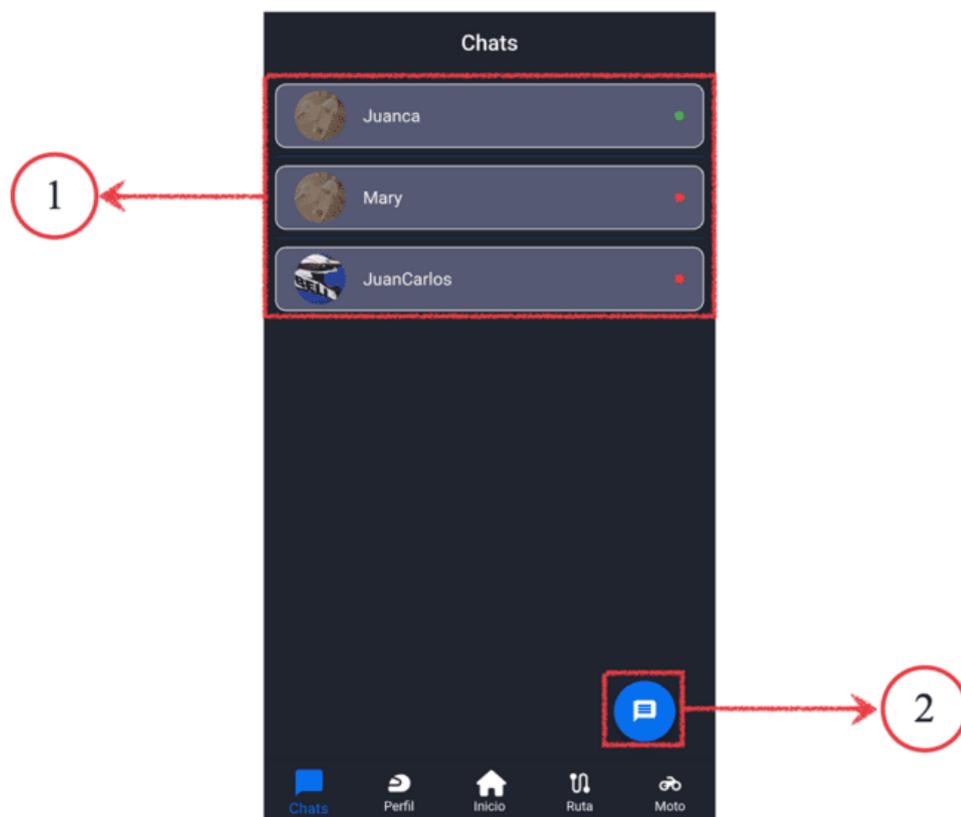


Ilustración A3 13.- UI Chat

- 1) Zona de listado de chats.
- 2) Botón de agregar nuevo chat.

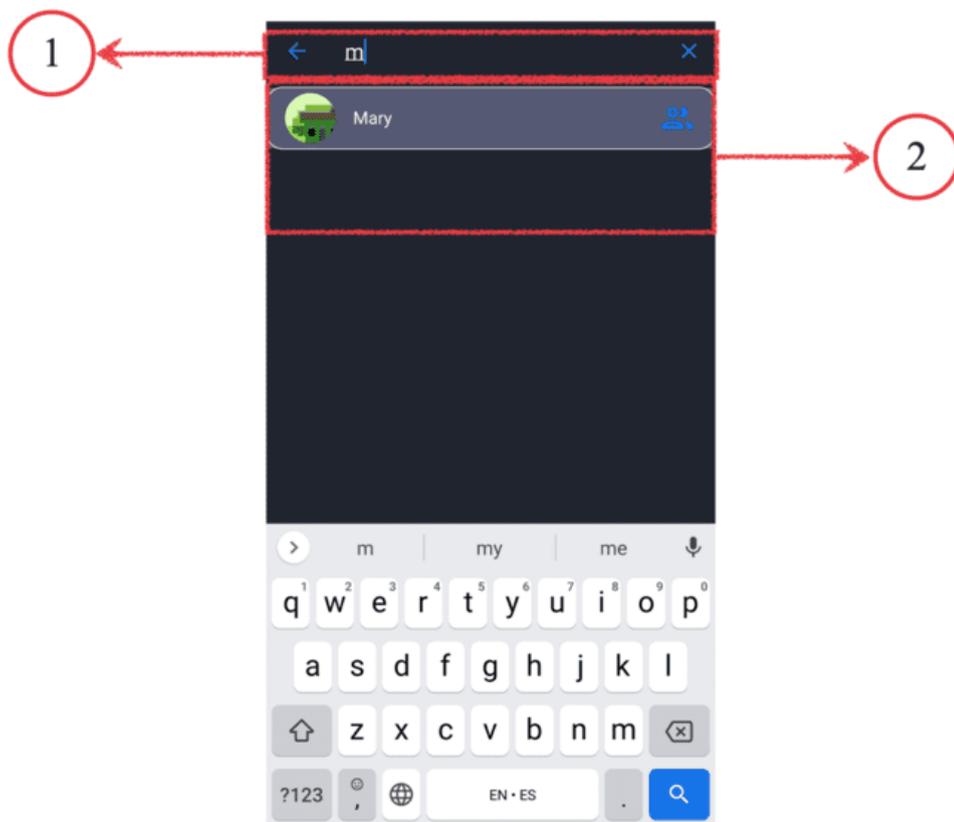


Ilustración A3 14.- UI Buscar y listar amigos para iniciar chat

- 1) Barra de búsqueda de amigos.
- 2) Zona de resultados de búsqueda realizada.

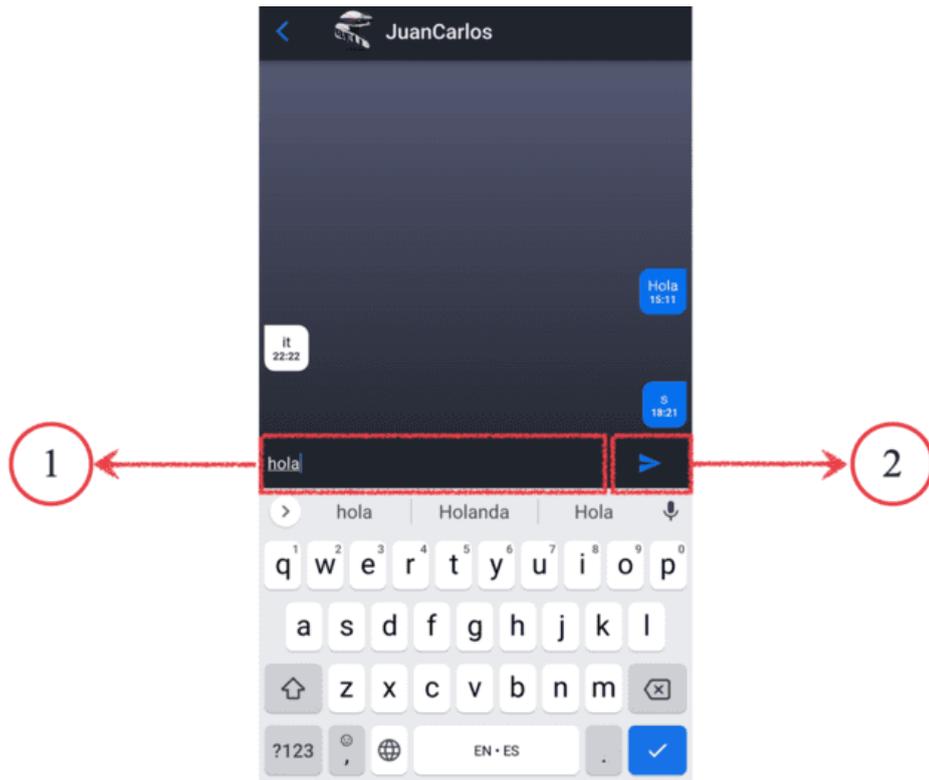


Ilustración A3 15.- UI Pantalla de chat entre amigos

- 1) Caja de texto para ingresar nuevo mensaje.
- 2) Botón de enviar mensaje.

Ventana del perfil de usuario

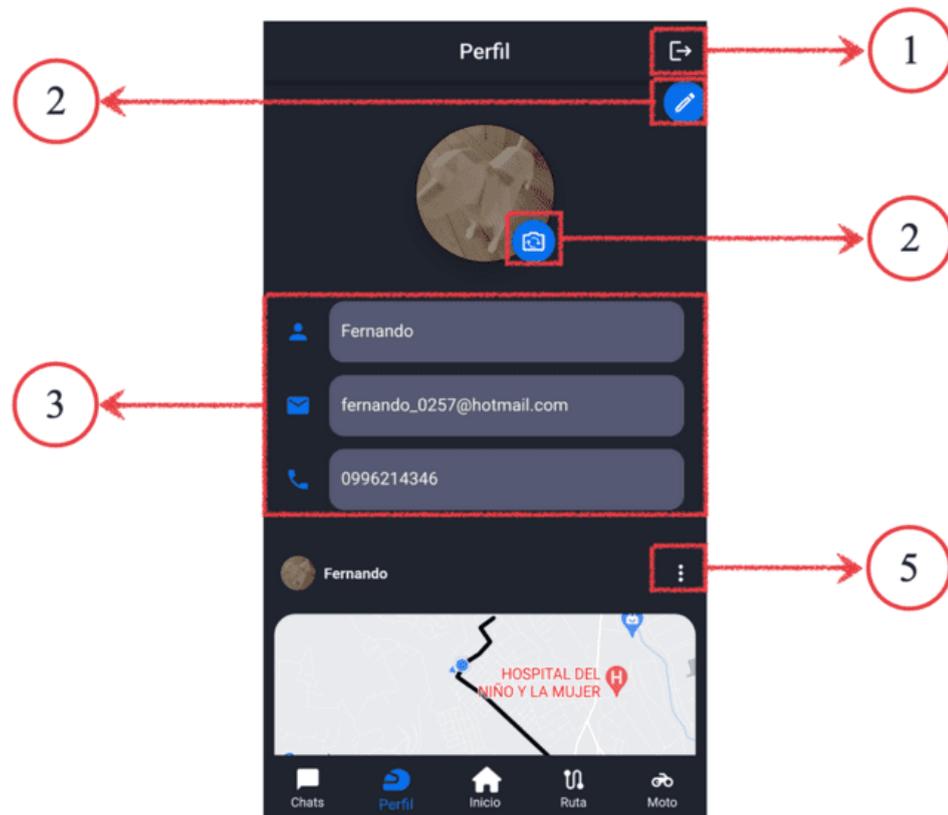


Ilustración A3 16.- UI Perfil del usuario

- 1) Botón de cerrar sesión.
- 2) Botón editar datos del usuario.
- 3) Botón de editar foto del usuario.
- 4) Zona de datos del usuario.
- 5) Botón de eliminar publicación.

Ventana del perfil de amigos

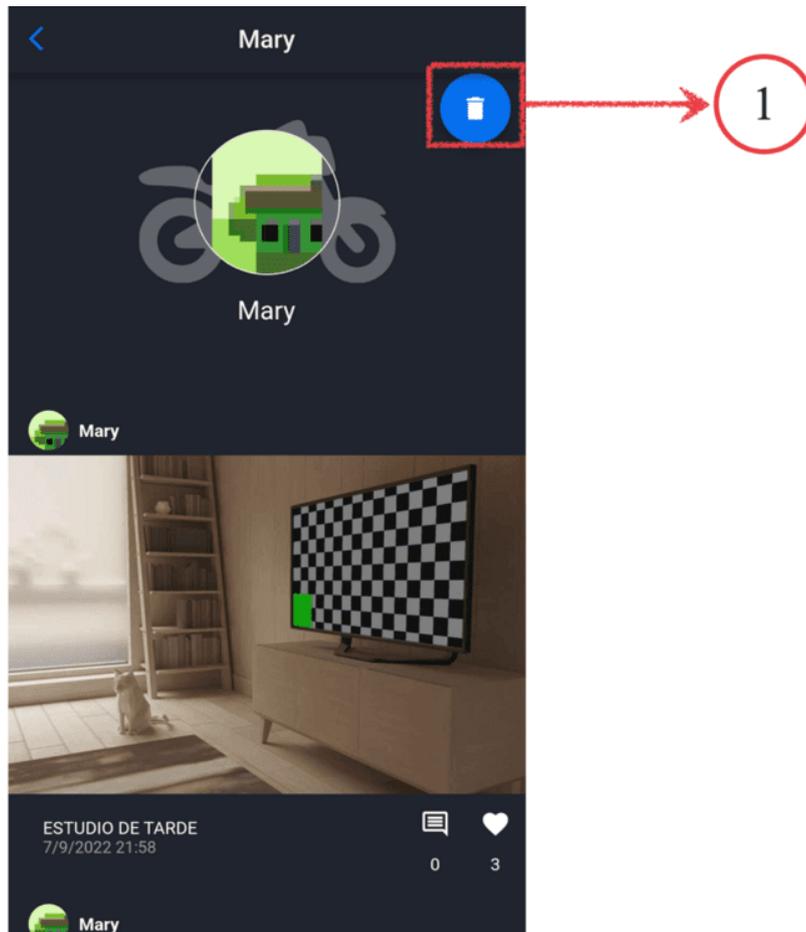


Ilustración A3 17.- UI Perfil de amigos

- 1) Botón de eliminar amigo.
- 2) Botones opcionales de comentar o reaccionar a la publicación.

Ventana de publicaciones

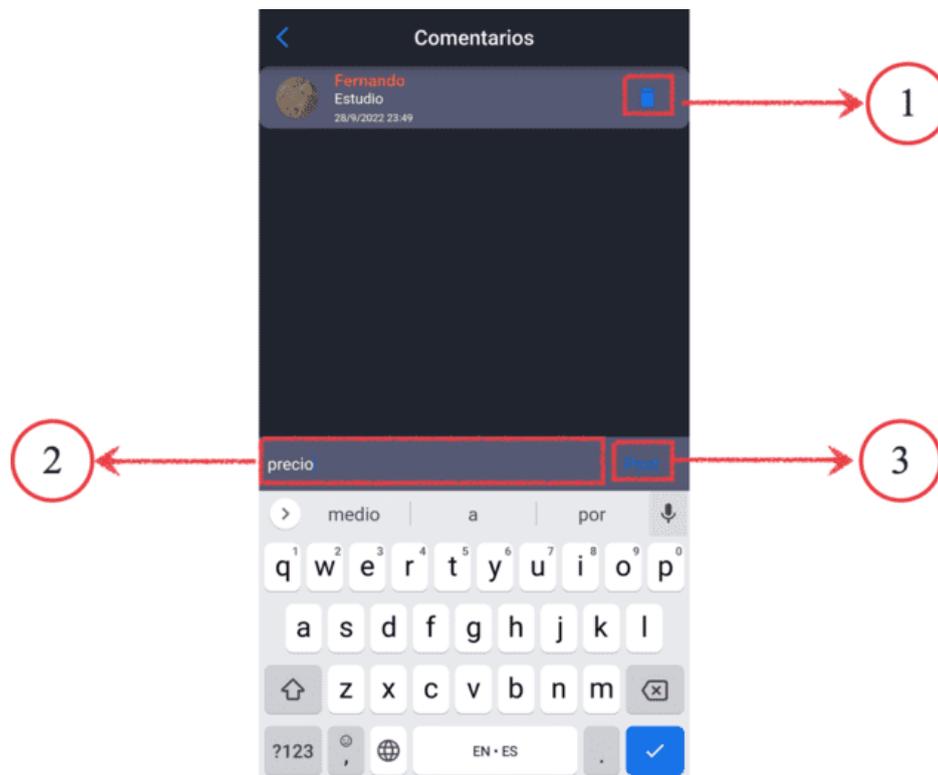


Ilustración A3 18.- UI Publicaciones

- 1) Botón de eliminar comentario de una publicación.
- 2) Caja de texto para ingresar nuevo comentario.
- 3) Botón de publicar comentario.

Lista de Ilustraciones Anexo 3

Ilustración A3 1.- UI Pantalla principal.....	121
Ilustración A3 2.- UI Listado de usuarios.....	122
Ilustración A3 3.- UI Listado de solicitudes de amistad.....	123
Ilustración A3 4.- UI Rutas.....	124
Ilustración A3 5.- UI Listado de historial de rutas	125
Ilustración A3 6.- UI Opciones de ubicación manualmente	126
Ilustración A3 7.- UI Alerta de dialogo para guardar marcador	127
Ilustración A3 8.- UI Opciones crear, visualizar e iniciar ruta.....	128
Ilustración A3 9.- UI Alerta de dialogo para guardar ruta	129
Ilustración A3 10.- UI Listado de rutas	130
Ilustración A3 11.- UI Motocicleta	131
Ilustración A3 12.- UI Alerta de dialogo para actualizar kilometraje de mantenimiento	132
Ilustración A3 13.- UI Chat	133
Ilustración A3 14.- UI Buscar y listar amigos para iniciar chat	134
Ilustración A3 15.- UI Pantalla de chat entre amigos	135
Ilustración A3 16.- UI Perfil del usuario.....	136
Ilustración A3 17.- UI Perfil de amigos	137
Ilustración A3 18.- UI Publicaciones.....	138

ANEXO 4 – MANUAL DE ADMINISTRADOR

Ventana Principal

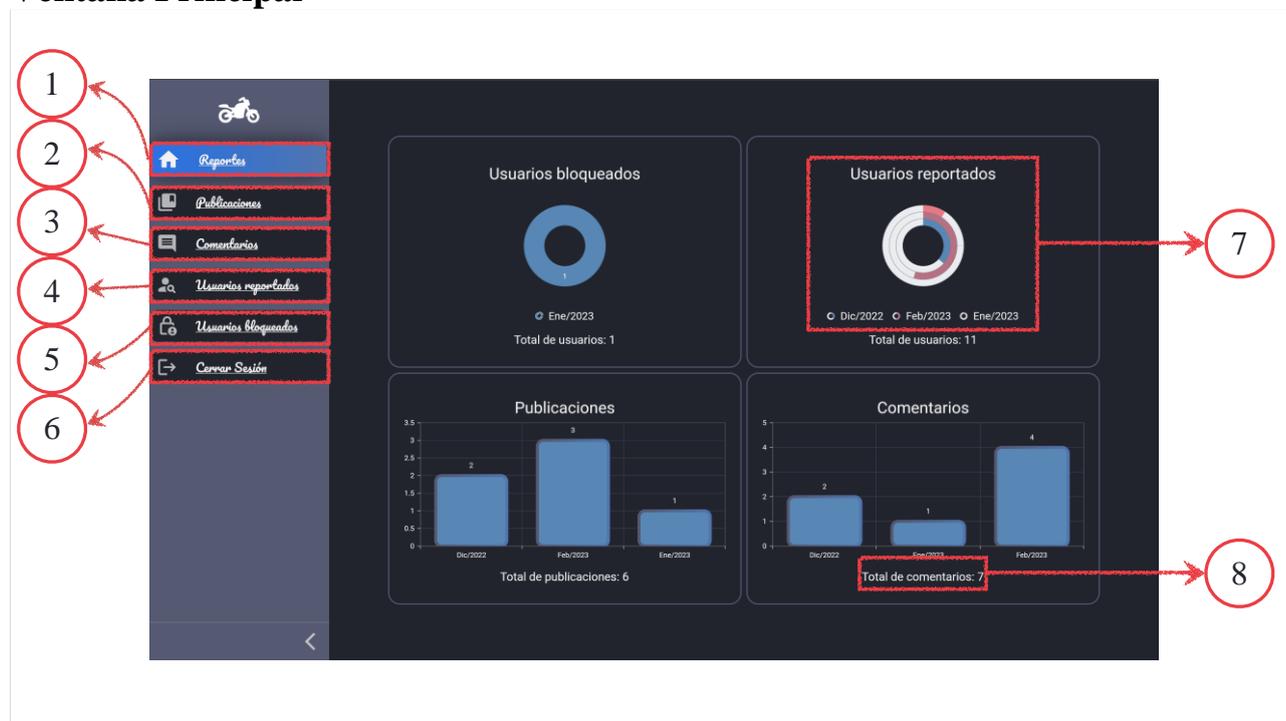


Ilustración A4 1.- UI Panel principal

- 1) Botón de ventana principal.
- 2) Botón de listado de publicaciones.
- 3) Botón de listado de comentarios.
- 4) Botón de listado de usuarios reportados.
- 5) Botón de listado de usuarios bloqueados.
- 6) Botón de cerrar sesión.
- 7) Zona de graficas informáticas de usuarios reportados y bloqueados, publicaciones y comentarios reportados.
- 8) Zona de datos de la cantidad de información establecida en el punto 7.

Ventana de publicaciones

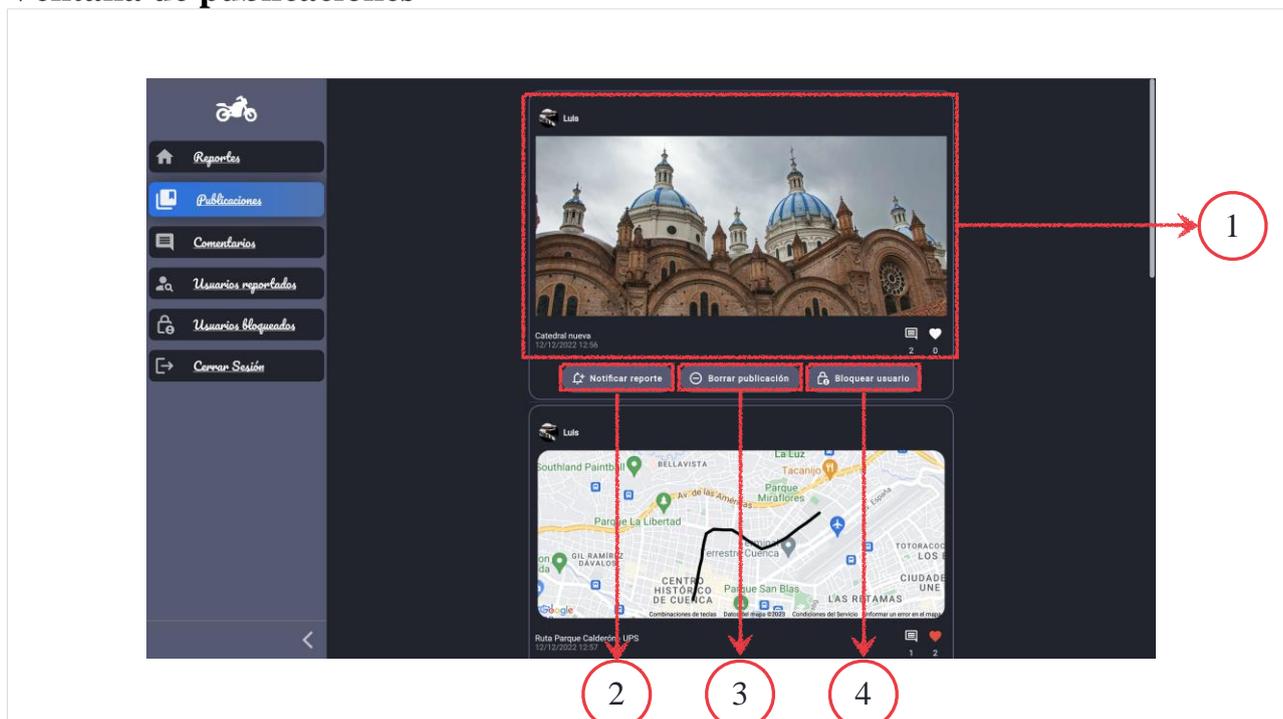


Ilustración A4 2.- UI Panel de publicaciones reportadas

- 1) Zona de la publicación.
- 2) Botón de notificar reporte.
- 3) Botón de borrar publicación.
- 4) Botón de bloquear usuario.

Ventana de comentarios

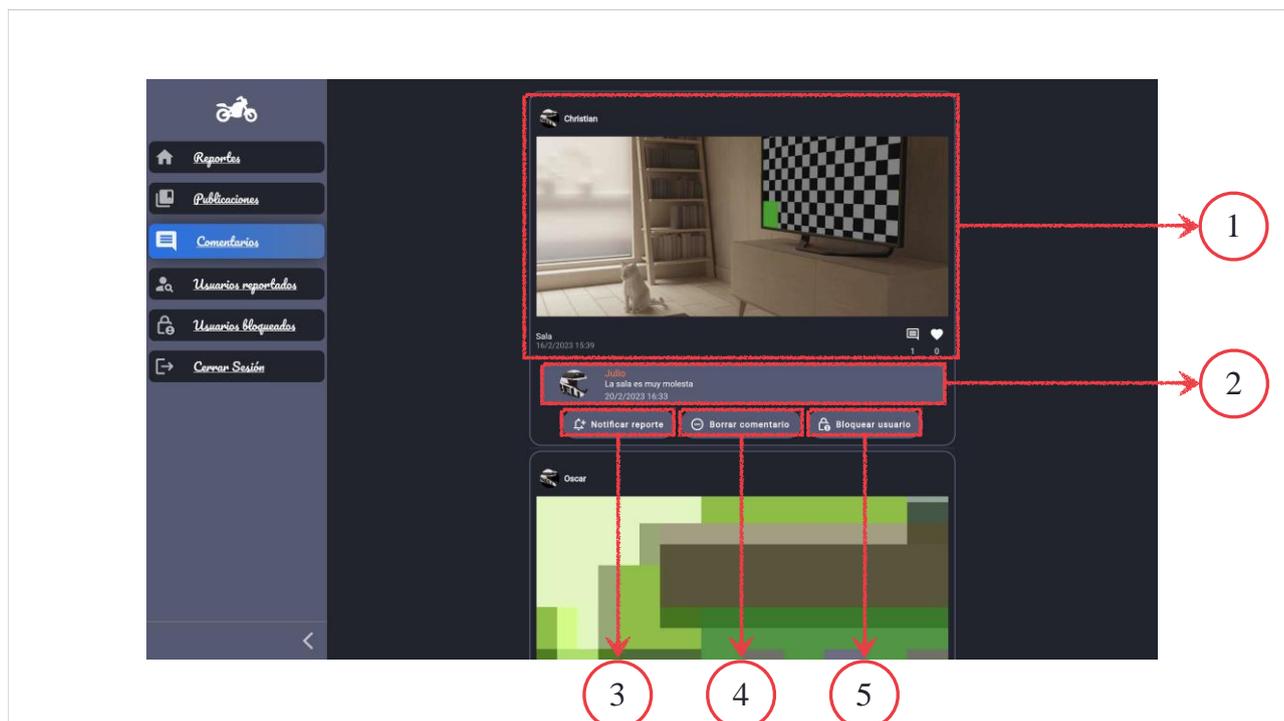


Ilustración A4 3.- Panel de comentarios reportados

- 1) Zona de la publicación.
- 2) Zona del comentario.
- 3) Botón de notificar reporte.
- 4) Botón de borrar comentario.
- 5) Botón de bloquear usuario.

Ventana de usuarios reportados

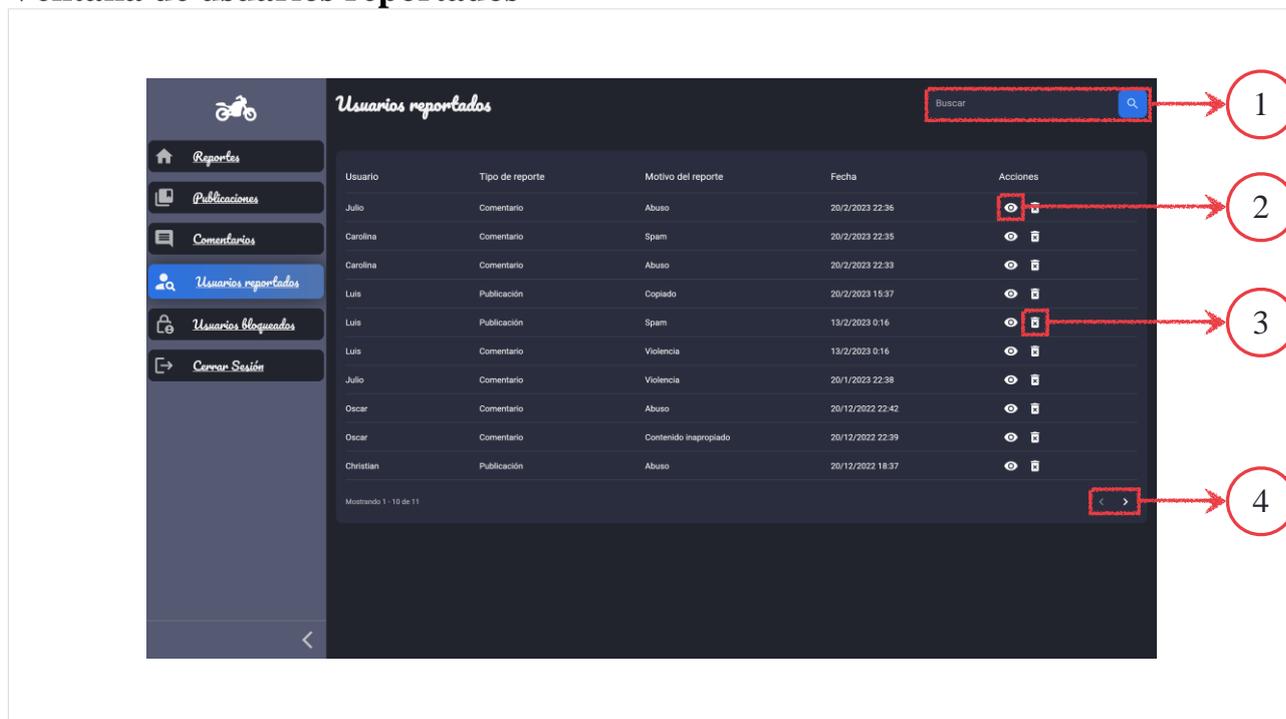


Ilustración A4 4.- Panel de listado de usuarios reportados

- 1) Botón y zona de buscar usuarios reportado, con filtros de nombre del usuario o motivo del reporte.
- 2) Botón de visualizar reporte.
- 3) Botón de eliminar reporte.
- 4) Botón de cambiar página.

Ventana de usuarios reportados

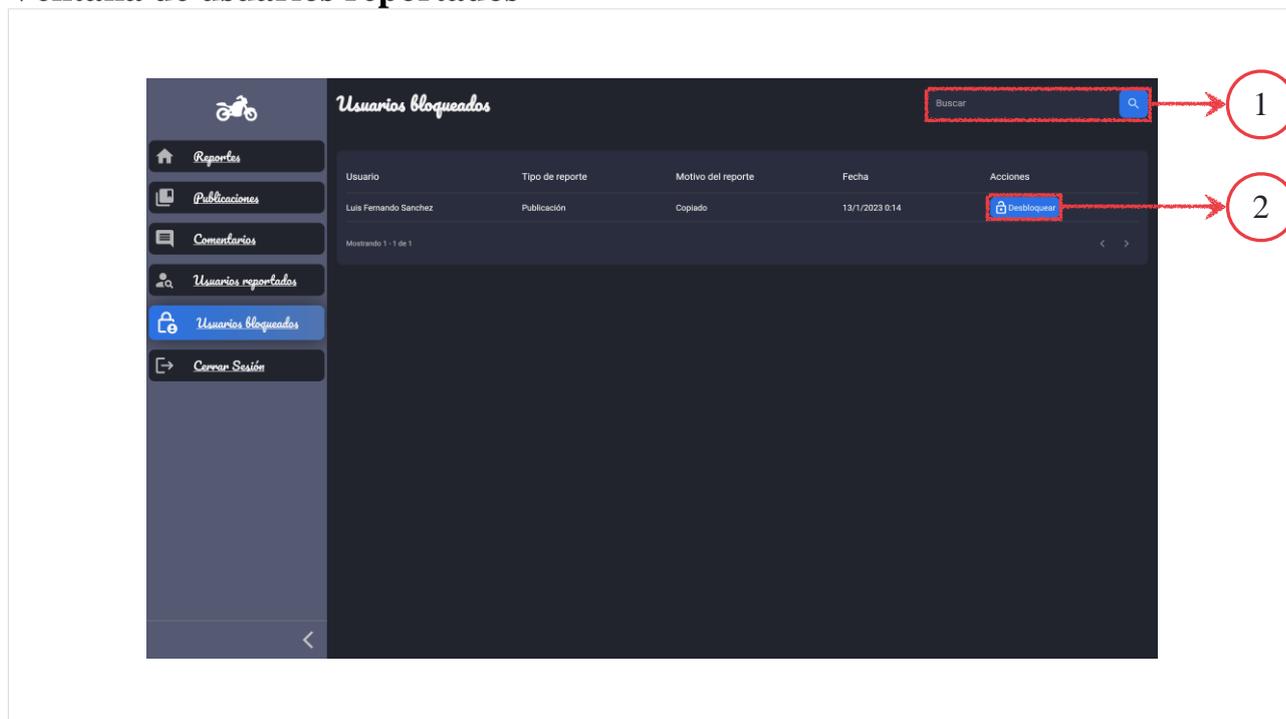


Ilustración A4 5.- UI Panel de listado de usuarios bloqueados

- 1) Botón y zona de buscar usuarios reportado, con filtros de nombre del usuario o motivo del reporte.
- 2) Botón de desbloquear usuario.

Ventana de cerrar sesión

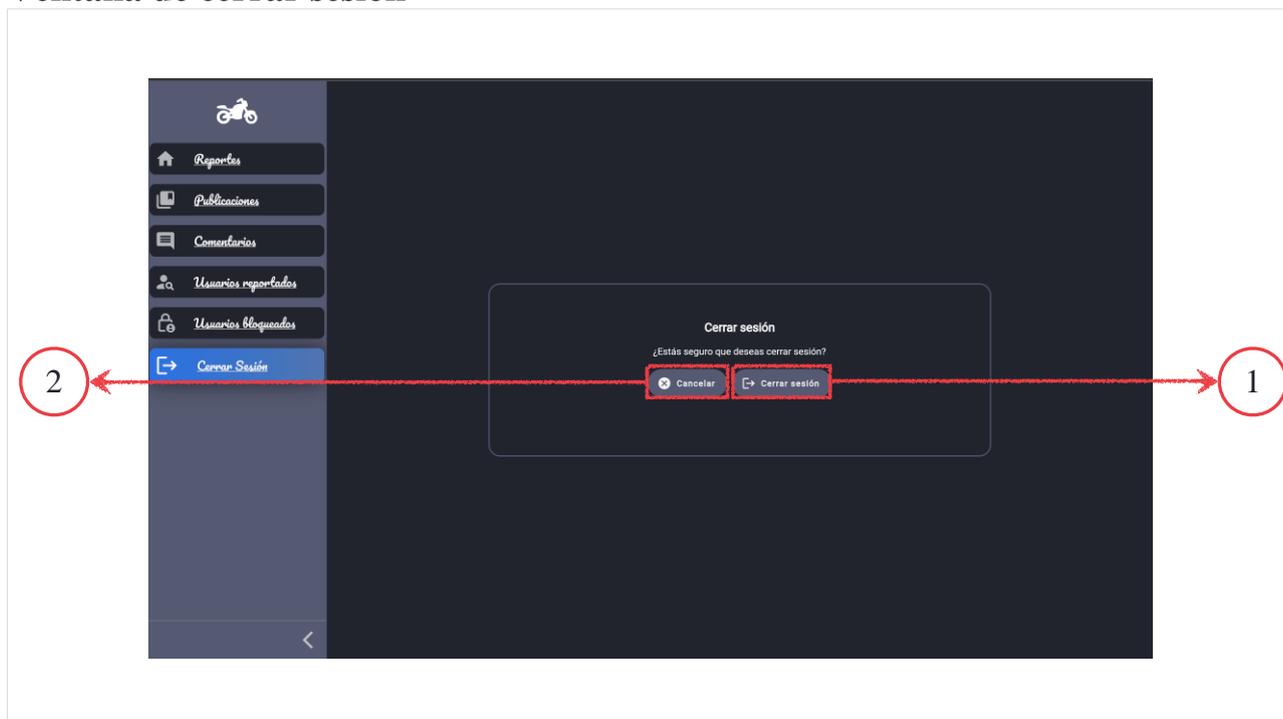


Ilustración A4 6.- UI Panel de cerrar sesión

- 1) Botón de cerrar sesión.
- 2) Botón de cancelar cerrar sesión

Lista de Ilustraciones Anexo 4

Ilustración A4 1.- UI Panel principal	140
Ilustración A4 2.- UI Panel de publicaciones reportadas	141
Ilustración A4 3.- Panel de comentarios reportados	142
Ilustración A4 4.- Panel de listado de usuarios reportados	143
Ilustración A4 5.- UI Panel de listado de usuarios bloqueados	144
Ilustración A4 6.- UI Panel de cerrar sesión	145