



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA INFORMÁTICA, CIENCIAS
DE LA COMPUTACIÓN E INNOVACIÓN
TECNOLÓGICA**

**CARRERA DE INGENIERÍA DE SISTEMAS DE
INFORMACIÓN**

**PROPUESTA DE APLICATIVO MÓVIL PARA MEDICIÓN DEL
CONSUMO DE AGUA POTABLE, UTILIZANDO
GEOREFERENCIACIÓN PARA EMAPAT**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN**

AUTORO: BRIAN SEBASTIAN GARCIA CASTILLO

DIRECTOR: ING. LUS FERMANDO PINOS CASTILLO.

CAÑAR - ECUADOR

2024

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA INFORMÁTICA, CIENCIAS
DE LA COMPUTACIÓN E INNOVACIÓN
TECNOLÓGICA**

**CARRERA DE INGENIERÍA DE SISTEMAS DE
INFORMACIÓN**

PROPUESTA DE APLICATIVO MÓVIL PARA MEDICIÓN DEL
CONSUMO DE AGUA POTABLE, UTILIZANDO
GEOREFERENCIACIÓN PARA EMAPAT

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN**

AUTORO: BRIAN SEBASTIAN GARCIA CASTILLO

DIRECTOR: ING. LUIS FERNANDO PINOS CASTILLO.

CAÑAR - ECUADOR

2024

DIOS, PATRIA, CULTURA Y DESARROLLO

DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD

Brian Sebastian Garcia Castillo portador de la cédula de ciudadanía N.º 0302604673
Declaro ser el autor de la obra: Propuesta de aplicativo móvil para medición del consumo de agua potable, utilizando georeferenciación para Emapat, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaró que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, 22 de noviembre de 2024



Brian Sebastian Garcia Castillo

C.I. 0302604673

CERTIFICACIÓN DEL TUTOR

El trabajo de titulación denominado **Propuesta de aplicativo móvil para medición del consumo de agua potable, utilizando georeferenciación para Emapat**, elaborado por **Brian Sebastian Garcia Castillo**, previo a la obtención del título de **Ingeniera en Sistema de Información**, ha sido asesorado, revisado y supervisado durante su ejecución bajo mi tutoría, por lo que certifico que el presente documento fue desarrollado siguiendo los parámetros del método científico, se sujeta a las normas éticas de investigación, por lo que esta expedito para su presentación y sustentación ante el respectivo tribunal.

Cañar, 27 de noviembre de 2024



Ing. Luis Fernando Pinos Castillo Mgs.

CI: 0301829255

TUTOR

DEDICATORIA

En este momento de gran satisfacción y logro, quiero expresar mi más profundo agradecimiento en primer lugar a Dios por no abandonarme nunca y guiar cada uno de mis pasos y a las personas que han sido fundamentales en mi camino.

A mis padres, Carlos García y Mirian Castillo, gracias por su amor incondicional, apoyo y sacrificio. Me han dado todo lo que soy como persona, mis valores, mis principios, mi perseverancia y empeño. Su guía y orientación me ha permitido crecer y alcanzar mis metas.

"A María Victoria la persona que nunca dejó de creer en mí por su paciencia, su ayuda y su comprensión, a nuestra pequeña Eva Sophia su llegada ha coincidido con el final de la tesis, ella es lo mejor que me ha pasado y ha venido a este mundo para darme el último empujón para terminar el trabajo. Es sin duda mi referencia para el presente y el futuro, mis dos tesoros gracias por llenar mi vida de amor, alegría y propósito."

A mis hermanas, Liliana, Karla gracias por su compañía, apoyo y amor. Han sido mis cómplices y amigas en momentos difíciles y felices.

A mis sobrinos, gracias por su energía. Han sido una fuente de inspiración y motivación.

Y a mi abuelita querida, Mercedes Páramo, que está en el cielo, gracias por su legado de amor y sabiduría. Su memoria y ejemplo han sido una guía constante en mi vida.

Este logro no sería posible sin el apoyo y amor de ustedes. Les agradezco desde lo más profundo de mi corazón.

AGRADECIMIENTO

Primeramente, agradezco a Dios, por haberme dado la fortaleza, sabiduría y paciencia necesarias para llevar a cabo este proyecto. Sin Su guía, este logro no habría sido posible.

Quiero expresar mi más sincero agradecimiento a los docentes que me acompañaron durante este proceso, y en especial a mi tutor de tesis, al ing. Luis Fernando Pinos Castillo, por su valiosa orientación, dedicación y paciencia a lo largo de todo el trabajo. Su conocimiento y compromiso fueron fundamentales para el desarrollo de esta investigación, y siempre estaré agradecido por su apoyo incondicional y sus enseñanzas, las cuales me guiaron hacia la culminación exitosa de este proyecto. Gracias por ser una inspiración y un mentor excepcional.

RESUMEN

Este proyecto aborda el desarrollo y diseño de una aplicación móvil multiplataforma destinada a medir el consumo de agua potable mediante georeferenciación para EMAPAT. Se empleó la metodología Mobile-D debido a su eficacia en la entrega ágil de prototipos funcionales y su idoneidad para el desarrollo de aplicaciones móviles. Tras un análisis de frameworks multiplataforma, se concluye que Flutter es la mejor opción por su alto rendimiento, facilidad de aprendizaje y compatibilidad multiplataforma. La aplicación fue desarrollada en Dart, el lenguaje de programación de Flutter. Esta herramienta permite registrar y georeferenciar el consumo de agua en tiempo real, mejorando la precisión y la eficiencia en la gestión de recursos hídricos. El diseño de la aplicación sigue principios de UI/UX para asegurar su usabilidad, accesibilidad e interacción intuitiva. Además, la aplicación ofrece funcionalidades como autenticación de usuarios, registro de incidencias, generación de informes en tiempo real, alertas automáticas y visualización de datos en mapas interactivos, facilitando una gestión integrada, sostenible y eficiente.

Palabras Clave: aplicativo móvil, Flutter, Dart, Firebase, consumo de agua

ABSTRACT

This project addresses developing and designing a cross-platform mobile application aimed at measuring drinking water consumption through georeferencing for EMAPAT. The Mobile-D methodology has been used due to its efficiency in the agile delivery of functional prototypes and its suitability for developing mobile applications. After an analysis of cross-platform frameworks, it is concluded that Flutter is the best option due to its high performance, ease of learning, and cross-platform compatibility. The application was developed in Dart, Flutter's programming language. This tool allows recording and georeferencing water consumption in real-time, improving accuracy and efficiency in water resource management. The application design follows UI/UX principles to ensure usability, accessibility, and intuitive interaction. In addition, the application offers functionalities such as user authentication, incident logging, real-time reporting, automatic alerts, and data visualization on interactive maps, enabling integrated, sustainable, and efficient management.

Keywords: mobile application, Flutter, Dart, Firebase, water consumption

INDICE

<u>DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD</u>	¡Error! Marcador no definido.
<u>CERTIFICACIÓN</u>	iii
<u>DEDICATORIA</u>	v
<u>TABLA DE ILUSTRACIONES</u>	xiii
<u>TABLA DE TABLAS</u>	xvi
<u>INTRODUCCIÓN</u>	17
<u>CAPÍTULO I</u>	19
<u>1.1. Planteamiento del problema</u>	19
<u>1.1.1. Formulación del problema</u>	20
<u>1.2. Antecedentes de la Investigación</u>	20
<u>1.3. Justificación de la investigación</u>	22
<u>1.4. Objetivos</u>	22
<u>1.4.1. Objetivo General</u>	22
<u>1.4.2. Objetivos Específicos</u>	22
<u>1.5. Limitaciones</u>	23
<u>1.6. Delimitaciones</u>	23
<u>CAPÍTULO II</u>	24
<u>2. MARCO TEÓRICO</u>	24
<u>2.1. Gestión del Agua Potable</u>	24
<u>2.1.1. Importancia del Agua Potable</u>	24

<u>2.1.2 Sistemas de Distribución de Agua Potable</u>	25
<u>2.2. Tecnologías de Medición del Consumo de Agua</u>	25
<u>2.2.1. Medidores de Agua</u>	25
<u>2.2.2. Lectura y Registro de Consumo</u>	27
<u>2.2.2.1. Métodos tradicionales vs. métodos digitales.</u>	28
<u>2.2.2.2 Importancia de la precisión en la medición.</u>	29
<u>2.3. Aplicativos Móviles y su Implementación</u>	30
<u>2.3.1. Desarrollo de Aplicaciones Móviles</u>	31
<u>2.3.1.1 Ciclo de vida del desarrollo de software.</u>	31
<u>2.3.1.2. Principios de diseño de interfaces de usuario (UI) y experiencia de</u> <u>usuario (UX).</u>	32
<u>2.3.1.3 Lenguajes de programación más utilizados.</u>	36
<u>2.3.1.4 Framework de Desarrollo en la actualidad (Flutter, Xamarin, Ionic).</u>	i
<u>2.3.1.5 Bases de datos</u>	iii
<u>2.3.1.6. Cuadro comparativo de Base de datos</u>	v
<u>2.3.2 Interoperabilidad de Sistemas</u>	vii
<u>2.3.2.1 Concepto Interoperabilidad</u>	vii
<u>2.3.2.2 Protocolos y estándares para la integración de sistemas (API, XML,</u> <u>JSON).</u>	viii
<u>2.4. Georreferenciación y Sistemas de Información Geográfica (SIG)</u>	xii
<u>2.4.1. Conceptos Básicos de Georreferenciación</u>	xiii
<u>2.4.1.1. Definición y utilidad de la georreferenciación.</u>	xiii

<u>2.4.1.2 Métodos y tecnologías para la georreferenciación (GPS, triangulación, etc.)</u>	xv
<u>2.4.2 Tecnologías y Plataformas para SIG en Aplicativos Móviles</u>	xvii
<u>2.4.2.1 Plataformas de desarrollo para aplicativos móviles con SIG</u>	xix
<u>2.4.2.1.1. Android Studio</u>	xix
<u>2.4.2.1.2. IOS (Xcode)</u>	xx
<u>2.4.2.1.3. Cross-Platform</u>	xxi
<u>2.4.3.2. Comparativa de tecnologías y sus aplicaciones</u>	xxviii
<u>CAPÍTULO III</u>	xxx
<u>3. MARCO METODOLÓGICO</u>	xxx
<u>3.1. Enfoque de la investigación</u>	xxx
<u>3.2. Nivel de la investigación</u>	xxx
<u>3.3. Población</u>	xxx
<u>3.4. Técnicas e instrumentos de recolección</u>	xxxí
<u>3.5. Tratamiento de la información</u>	xxxí
<u>3.6. Resultados</u>	xxxí
<u>3.6.1. Análisis de los resultados</u>	xxxíi
<u>3.6.2. Análisis general de la entrevista</u>	xxxíiv
<u>CAPÍTULO IV</u>	xxxvi
<u>4. PROPUESTA</u>	xxxvi
<u>4.1. Título de la Propuesta</u>	xxxvi
<u>4.2. Presentación</u>	xxxvi

<u>4.3. Justificación</u>	xxxvi
<u>4.4. Descripción de la propuesta</u>	xxxvii
<u>4.5. Desarrollo del proyecto</u>	xxxvii
<u>4.5.1. Fase I: Exploración</u>	xxxvii
<u>4.5.1.1. Requerimientos funcionales</u>	xxxviii
<u>4.5.1.2. Requerimientos no funcionales</u>	xxxviii
<u>4.5.1.3. Establecimiento del proyecto</u>	xl
<u>4.5.2. Fase II: Inicialización</u>	xlii
<u>4.5.2.1. Esquema de solución</u>	xliii
<u>4.5.2.2. Configuraciones para el proyecto</u>	xliii
<u>4.5.3. Fase III: Producción</u>	xliv
<u>4.5.3.1. Diagrama de caso de uso</u>	xlvi
<u>4.5.3.2. Base de datos No relacional</u>	xlvi
<u>4.5.3.3. Interfaz del aplicativo móvil</u>	xlvii
<u>4.5.4. Fase IV: estabilización</u>	liii
<u>4.5.4.1. Fase de pruebas</u>	liv
<u>Conclusiones</u>	lvi
<u>Recomendaciones</u>	lviii
<u>Bibliografía</u>	lix
<u>MANUAL DE USUARIO</u>	lxxii
<u>Manual del Programador</u>	lxxxiii

1. TABLA DE ILUSTRACIONES

<u>Ilustración 1. Ciclo de desarrollo de software. Fuente: Autoría Propia.....</u>	31
<u>Ilustración 2. Logo de la BD Firebase. Fuente: (Firebase, 2024).....</u>	iii
<u>Ilustración 3. Base de datos MySQL. Fuente: (MySQL, 2024)</u>	iv
<u>Ilustración 4. Base de datos PostgreSQL. Fuente: (Kinsta, 2023)</u>	v
<u>Ilustración 5. Android Studio. Fuente: Autoría Propia</u>	xix
<u>Ilustración 6. XCODE. Fuente: (applesfera, 2024)</u>	xx
<u>Ilustración 7. Cross-Platform. Fuente: (cleancommit, 2022)</u>	xxi
<u>Ilustración 8. Arquitectura general de la infraestructura del producto. Fuente: Autoría Propia.....</u>	xliii
<u>Ilustración 9. Diagrama de Caso de uso del empleado de la EMAPAT. Fuente: Autoría Propia.....</u>	xlvi
<u>Ilustración 10. Base de Datos No relacional. Fuente: Autoría Propia</u>	xlvii
<u>Ilustración 11. Interfaz del Login. Fuente: Autoría Propia.....</u>	xlviii
<u>Ilustración 12. Interfaz del Menú. Fuente: Autoría propia.....</u>	xlix
<u>Ilustración 13. Interfaz para generar la ubicación del medidor. Fuente: Autoría Propia... </u>	l
<u>Ilustración 14. Registro del consumo de agua. Fuente: Autoría propia.....</u>	li
<u>Ilustración 16. Ubicación de incidencias. Fuente: Autoría Propia</u>	lii
<u>Ilustración 15. Visualización de las incidencias. Fuente: autoría propia.....</u>	liii
<u>Ilustración 1. Ícono de la App. Fuente: Autoría Propia</u>	lxxiii
<u>Ilustración 2. Interfaz de la carga de la app. Fuente: Autoría Propia</u>	lxxiii
<u>Ilustración 3. Interfaz de Usuario. Fuente: Autoría Propia.....</u>	lxxiv
<u>Ilustración 4. Interfaz de Olvidó Su Contraseña. Fuente: Autoría Propia.....</u>	lxxv
<u>Ilustración 5. Menú de la app. Fuente: Autoría Propia.....</u>	lxxv
<u>Ilustración 6. Interfaz de Crear Ubicación. Fuente: Autoría Propia.....</u>	lxxvi

<u>Ilustración 7. Solicitud de Activación de Ubicación. Fuente: Autoría Propia.</u>	lxxvi
<u>Ilustración 8. Ubicación. Fuente: Autoría Propia.</u>	lxxvii
<u>Ilustración 9. Identificador del Medidor. Fuente: Autoría Propia.</u>	lxxviii
<u>Ilustración 10. Notificación de datos. Fuente: Autoría Propia.</u>	lxxviii
<u>Ilustración 11. Menú Mediciones. Fuente: Autoría Propia.</u>	lxxix
<u>Ilustración 12. Gestión de las mediciones. Fuente: Autoría Propia.</u>	lxxix
<u>Ilustración 13. Visualización de las incidencias. Fuente: Autoría Propia.</u>	lxxx
<u>Ilustración 14. Registro del consumo del agua. Fuente: Autoría Propia.</u>	lxxxi
<u>Ilustración 15. Creación de un nuevo proyecto en FIREBASE. Fuente: Autoría Propia.</u>	lxxxiii
<u>Ilustración 16. Selección de Plataforma. Fuente: Autoría Propia.</u>	lxxxiv
<u>Ilustración 17. Instalación de CLI en Flutter. Fuente: Autoría Propia.</u>	lxxxiv
<u>Ilustración 18. Google-services.json. Fuente: Autoría Propia.</u>	lxxxv
<u>Ilustración 19. Autenticación de Usuarios en Firebase. Fuente: Autoría Propia.</u> ...	lxxxvi
<u>Ilustración 20. Realtime Database En FireBase. Fuente: Autoría Propia.</u>	lxxxvii
<u>Ilustración 21. Destinos. Fuente: Autoría Propia.</u>	lxxxviii
<u>Ilustración 22. Tabla Consumos. Fuente: Autoría Propia.</u>	lxxxix
<u>Ilustración 23. AndroidManifest. Fuente: Autoría Propia.</u>	xc
<u>Ilustración 24. iOS. Fuente: Autoría Propia.</u>	xc
<u>Ilustración 25. Especificación de las dependencias dentro del Proyecto. Fuente: Autoría Propia.</u>	xc
<u>Ilustración 26. Carpeta LIB. Fuente: Autoría Propia.</u>	xcii
<u>Ilustración 27. Autentification. Dart. Fuente: Autoría Propia.</u>	xciii
<u>Ilustración 28. Ícono Widget. Fuente: Autoría Propia.</u>	xciii
<u>Ilustración 29. Archivo Text Field.Dat. Fuente: Autoría Propia.</u>	xciv

<u>Ilustración 30. Main Dart. Fuente: Autoría Propia.</u>	xcv
<u>Ilustración 31. SplashScreen. Fuente: Autoría Propia.</u>	xcvi
<u>Ilustración 32. Login Dart. Fuente: Autoría Propia.</u>	xcvii
<u>Ilustración 33. AdminScreen Dart. Fuente: Autoría Propia.</u>	xcviii
<u>Ilustración 34. UbicacionesScreen Dart. Fuente: Autoría Propia.</u>	xcix
<u>Ilustración 35. DestinosScreen Dart. Fuente: Autoría Propia.</u>	c
<u>Ilustración 36. MapsScreen Dart. Fuente: Autoría Propia.</u>	ci

2.

TABLA DE TABLAS

<u>Tabla 1. Matriz comparativa de lenguajes de programación móvil. Fuente: Autoría Propia.....</u>	39
<u>Tabla 2. Matriz comparativa de las plataformas de desarrollo. Fuente: Autoría Propia... </u>	ii
<u>Tabla 3. Comparaciones de las bases de datos: Fuente: Autoría propia.....</u>	v
<u>Tabla 4. Matriz Comparativa Matriz comparativa Protocolos y estándares para la integración de sistemas. Fuente: Autoría Propia.....</u>	x
<u>Tabla 5. Características de XCODE. Fuente: Autoría Propia.....</u>	xx
<u>Tabla 6. Plataformas de Desarrollo Cross-Platform para Aplicativos Móviles con SIG. Fuente: Autoría Propia.....</u>	xxii
<u>Tabla 7. Matriz comparativa de las plataformas de desarrollo. Fuente: Autoría Propia.....</u>	xxiii
<u>Tabla 8. Comparación de tecnologías y sus aplicaciones. Fuente: Autoría Propia. ..</u>	xxviii
<u>Tabla 9. Requisitos de funcionalidad. Fuente: Autoría Propia</u>	xxxviii
<u>Tabla 10. Requisitos no funcionales. Fuente: Autoría propia</u>	xxxviii
<u>Tabla 11. Plan de iteraciones. Fuente: Autoría Propia</u>	xl
<u>Tabla 12. Características del usuario. Fuente: Autoría Propia.....</u>	xlii
<u>Tabla 13. Recuso de Hardware. Fuente; Autoría Propia.....</u>	xliii
<u>Tabla 14. Módulos de la APP Móvil. Fuente: Autoría Propia</u>	xliv
<u>Tabla 15. Prueba de aceptación del login. Fuente: Autoría Propia.....</u>	liv
<u>Tabla 16. Prueba del Mapa. Autoría Propia.....</u>	liv
<u>Tabla 17. Prueba de Registro de las incidencias. Fuente: Autoría Propia.....</u>	lv

INTRODUCCIÓN

La presente investigación aborda el desafío crucial de medir con precisión el consumo de agua potable en la EMAPAT¹. Los métodos tradicionales de medición son ineficientes y propensos a errores, lo que impide una respuesta rápida ante situaciones críticas como fugas o sobreconsumo. Esta deficiencia en la recolección de datos compromete la capacidad de EMAPAT para gestionar eficazmente los recursos hídricos, esenciales para la planificación urbana y la sostenibilidad ambiental.

El interés en este trabajo surge de la necesidad de explorar soluciones tecnológicas avanzadas que puedan mejorar la precisión y eficiencia en la medición del consumo de agua. La georreferenciación ofrece una solución potencial para agilizar el proceso de recolección de datos de consumo de agua, permitiendo una localización precisa de los medidores y un monitoreo constante. Sin embargo, la integración de esta tecnología en aplicativos móviles diseñados específicamente para mejorar la recolección de datos en la gestión del agua aún está en sus primeras etapas, especialmente en contextos urbanos de países en desarrollo.

La metodología utilizada para el desarrollo de este aplicativo es MOBILE-D, un enfoque ágil específicamente diseñado para el desarrollo de aplicaciones móviles, utilizando el framework Flutter. Este método permite una adaptación rápida y eficiente a las necesidades cambiantes del proyecto, asegurando la entrega de un producto funcional y de alta calidad en un plazo corto.

¹ Empresa Pública de Agua Potable y Alcantarillado de El Tambo (EMAPAT)

Las limitaciones de este trabajo incluyen la falta de compromiso de las partes interesadas, el tiempo limitado de 3 meses para la realización del trabajo de titulación y posibles cambios en la regulación del uso del agua o en la infraestructura tecnológica regional que podrían influir en la eficacia del aplicativo.

El objetivo general de este estudio es desarrollar un aplicativo móvil que emplea tecnología de georreferenciación para optimizar la medición y gestión del consumo de agua potable en EMAPAT, con el fin de incrementar la eficiencia operativa y promover la sostenibilidad del recurso hídrico. Los objetivos específicos incluyen desarrollar un marco teórico que respalde el uso de tecnologías de georreferenciación, analizar los requerimientos para la elaboración del aplicativo móvil y desarrollar un prototipo para la toma de consumo de agua potable utilizando georreferenciación.

La estructura de esta tesis se organiza en varios capítulos.

El primer capítulo presenta el planteamiento del problema, los antecedentes de la investigación y la justificación del estudio, así como los objetivos generales y específicos.

El segundo capítulo ofrece un marco teórico que revisa la literatura relevante sobre la gestión del agua potable, tecnologías de medición y desarrollo de aplicativos móviles.

El tercer capítulo describe el marco metodológico, detallando el enfoque de la investigación, las técnicas de recolección de datos y el análisis de resultados. Finalmente,

El cuarto capítulo expone la propuesta del aplicativo móvil.

CAPÍTULO I

1.1. Planteamiento del problema

La medición precisa del consumo de agua potable es un desafío fundamental para las entidades proveedoras de servicios, incluida la Empresa Pública de Agua Potable y Alcantarillado de El Tambo (EMAPAT). Los métodos tradicionales de medición son a menudo ineficientes, propensos a errores y no facilitan respuestas rápidas a situaciones críticas como fugas o sobreconsumo. Esta deficiencia en la recolección de datos precisos y en tiempo real compromete la capacidad de EMAPAT para gestionar eficazmente el recurso hídrico, lo cual es crucial para la planificación urbana. En este contexto, se hace imprescindible explorar soluciones tecnológicas avanzadas que permitan mejorar la exactitud y eficiencia del proceso de medición, para asegurar una gestión más efectiva y sostenible del agua.

La implementación de tecnologías innovadoras, como la georreferenciación, ofrece una solución potencial para agilizar el proceso de recolección de datos de los consumos de agua, permitiendo una localización precisa de los medidores y un monitoreo constante. Sin embargo, la integración de esta tecnología en aplicativos móviles diseñados específicamente para mejorar la recolección de datos en la gestión del agua aún está en sus primeras etapas, especialmente en contextos urbanos de países en desarrollo (Rochina Chisag , Guashpa Pasto , & Coloma Garofalo, 2022).

Esta investigación propone el desarrollo de un aplicativo móvil que utilice la georreferenciación para mejorar la medición y gestión del consumo de agua potable en EMAPAT. Este estudio buscará responder cómo un aplicativo móvil basado en georreferenciación puede contribuir a la precisión en la medición del consumo, mejorar la eficiencia en la gestión del agua y facilitar la toma de decisiones basadas en datos

precisos y en tiempo real. Asimismo, se explorará la aceptación de esta tecnología por parte de los usuarios y su impacto en la reducción del consumo innecesario y la detección de anomalías.

1.1.1. Formulación del problema

A pesar de los avances tecnológicos en la gestión de recursos hídricos, muchos sistemas de medición de consumo de agua siguen siendo ineficientes, propensos a errores y carecen de la capacidad para proporcionar datos en tiempo real. En el contexto de EMAPAT, la falta de precisión en la medición del consumo de agua potable puede llevar a pérdidas significativas de recursos, facturación inexacta y dificultades en la gestión eficiente del agua. Además, la ausencia de un sistema que integre tecnologías modernas como la georreferenciación limita la capacidad de realizar un seguimiento efectivo y de gestionar el consumo de agua de manera proactiva.

En este escenario, surge la necesidad de explorar cómo un aplicativo móvil que utilice la georreferenciación puede mejorar la precisión en la medición del consumo de agua, ofrecer datos actualizados en tiempo real y facilitar la gestión efectiva del recurso hídrico. Por lo tanto, la pregunta clave que guía esta investigación es:

¿Cómo puede un aplicativo móvil equipado con tecnología de georreferenciación mejorar la medición y gestión del consumo de agua potable en EMAPAT, y cuáles son las implicaciones de su implementación en términos de eficiencia operativa y satisfacción del usuario?

1.2. Antecedentes de la Investigación

(Torres Bastidas, 2021) en su tesis denominada “Aplicación móvil multiplataforma para la gestión de información georeferencial y servicio técnico comunitario de plomería, aplicando geolocalización Offline, en la Junta Administradora de Agua Potable de los

Barrios Occidentales de Aloasí”, describen una aplicación móvil que utiliza información georreferencial para la gestión de datos relacionados con el servicio técnico de plomería. Los autores utilizan el lenguaje de programación TypeScript, consumiendo servicios de Firebase para sincronizar los datos en tiempo real.

Este documento servirá para analizar las herramientas y frameworks utilizados para desarrollar el prototipo.

En el año (Valencia Astudillo & Delgado Tapia, 2021)proponen un prototipo de plataforma móvil para la evaluación y monitoreo de la calidad de cuerpos de agua, utilizando bioindicadores y estrategias de monitoreo colaborativo. La aplicación permite a los usuarios recopilar datos sobre la calidad del agua, georreferenciarlos y analizarlos para determinar el estado del cuerpo de agua.

Si bien el documento se enfoca en el monitoreo de la calidad de cuerpos de agua naturales, algunos de sus conceptos y tecnologías podrían ser relevantes para el desarrollo del aplicativo móvil para la medición del consumo de agua potable de EMAPAT.

(NARANJO HUERTAS , 2022)describe en su documento el desarrollo e implementación de una aplicación móvil que utiliza un servicio web para agilizar el proceso de recolección y gestión de datos catastrales, específicamente para la gestión de recursos hídricos en la ciudad de Tacna, Perú. La aplicación permite a los usuarios recopilar información de la propiedad, incluida la ubicación, las dimensiones y las características, directamente desde el campo. Luego, estos datos se sincronizan con una base de datos central a través de un servicio web, lo que facilita prácticas eficientes de gestión de los recursos hídricos.

Este documento de tesis, servirá para considerar requisitos específicos que debería tener el aplicativo móvil.

1.3. Justificación de la investigación

En un contexto global donde la sostenibilidad y la eficiencia en la gestión de recursos naturales son prioritarias, la medición precisa del consumo de agua se presenta como un desafío significativo, especialmente para las empresas encargadas de su suministro y regulación. EMAPAT, como entidad responsable del manejo del agua potable en el cantón El Tambo, enfrenta problemas comunes a muchas administraciones públicas: ineficiencias operativas, dificultades en la recopilación de datos exactos y oportunas, y la necesidad de mejorar la atención y servicio al cliente.

Es por ello que, la propuesta de desarrollar un aplicativo móvil que integre tecnologías de georreferenciación, se justifica porque permitirá una localización precisa y eficiente de la toma de mediciones de consumo de agua, optimizando la recolección de datos y la facturación. Esta precisión mejorará la eficiencia operativa, reduciendo costos y tiempo en mantenimientos y lecturas de medidores, al tiempo que facilitará una gestión más sostenible del agua mediante el análisis detallado del consumo. Además, proporcionará a los usuarios información accesible sobre su uso del agua, fomentando prácticas de consumo responsable y contribuyendo a la conservación del recurso hídrico vital.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un aplicativo móvil que emplea tecnología de georeferenciación para optimizar la medición y gestión del consumo del agua potable en EMAPAT.

1.4.2. Objetivos Específicos

- Desarrollar un marco teórico que respalde el uso de tecnologías de georreferenciación en aplicaciones móviles, para la localización de medidores de

consumo de agua potable, tecnologías y metodologías apropiadas para optimizar la gestión del agua.

- Analizar los requerimientos para la elaboración del aplicativo móvil, incluyendo características esenciales como la recolección de datos georreferenciados, interfaz de usuario, funcionalidades clave y mecanismos de seguridad para la protección de datos.
- Desarrollar un prototipo para la toma de consumo de agua potable, utilizando georreferenciación para EMAPAT.

1.5. Limitaciones

- Falta de compromiso de las partes interesadas
- Tiempo mínimo de 3 meses para la realización del trabajo de titulación
- Cambios en la regulación del uso del agua o en la infraestructura tecnológica regional, podrían influir en eficacia del aplicativo.

1.6. Delimitaciones

El prototipo a desarrollarse se elaborará en beneficio de la empresa EMAPAT del cantón El Tambo, en el que el tiempo de desarrollo será de 3 meses. El desarrollo se limitará a las características esenciales identificadas en los requerimientos.

Además, el prototipo será desarrollado utilizando un conjunto específico de tecnologías y plataformas, lo que podría limitar la portabilidad del aplicativo a otros sistemas o tecnologías no contempladas.

CAPÍTULO II

MARCO TEÓRICO

2.1. Gestión del Agua Potable

La gestión del agua potable es un área crítica de la administración pública y la ingeniería ambiental que se ocupa de garantizar la provisión segura y sostenible de agua potable a las comunidades. Este proceso involucra una variedad de actividades técnicas, políticas y administrativas destinadas a optimizar el uso y proteger los recursos hídricos.

Es fundamental para la salud pública, el bienestar económico y la sostenibilidad ambiental. Las prácticas y estrategias utilizadas deben ser integralmente diseñadas para manejar de manera efectiva tanto los recursos como la demanda, asegurando que todos los ciudadanos tengan acceso a agua potable de calidad (HatchKuri & Ribeiro, 2020).

2.1.1 Importancia del Agua Potable

De acuerdo con Araque (2022), el agua potable es crucial para la salud humana, el desarrollo económico y la estabilidad social, ya que es indispensable para la vida y constituye un recurso vital para la realización de innumerables actividades cotidianas. Su acceso seguro y confiable permite prevenir enfermedades relacionadas con el agua contaminada, promueve la higiene y el bienestar, y es esencial para sectores como la agricultura, la industria y el turismo.

Además, el agua potable juega un papel clave en la sostenibilidad ambiental, pues la gestión eficiente de este recurso ayuda a conservar los ecosistemas acuáticos y a mitigar los efectos del cambio climático, asegurando así su disponibilidad para las generaciones futuras. Por lo tanto, asegurar un suministro adecuado de agua potable es una prioridad

global que requiere inversiones continuas en infraestructura, tecnología y educación para su preservación y gestión eficaz (Nadal, 2024).

3. 2.1.2 Sistemas de Distribución de Agua Potable

Los sistemas de distribución de agua potable son redes de ingeniería diseñadas para transportar agua desde las plantas de tratamiento hasta los consumidores. Estos sistemas son fundamentales para asegurar que el agua potable llegue de manera segura y eficiente a hogares, empresas e instituciones.

Entre los componentes principales se encuentran las estaciones de bombeo que se encuentran ubicadas estratégicamente para mantener la presión necesaria en la red y asegurar que el agua alcance a las áreas destinadas. La red de tuberías; las válvulas utilizadas para controlar el flujo del agua dentro de la red; los hidratantes instalados en puntos estratégicos para proporcionar acceso al agua, y los tanques de almacenamiento que sirven para almacenar el agua tratada y ayudar a mantener la presión dentro del sistema (Tapiero & Valencia, 2021).

2.2. Tecnologías de Medición del Consumo de Agua

Con el paso del tiempo, las tecnologías de medición de consumo de agua han evolucionado significativamente, permitiendo a las empresas públicas y a los consumidores.

4. 2.2.1. Medidores de Agua

Los medidores de agua son dispositivos esenciales en la gestión del suministro de agua, utilizados para medir la cantidad de agua que consume un hogar, una empresa, o cualquier usuario final. Estos dispositivos ayudan a las compañías de agua a facturar

correctamente a los usuarios según su consumo real, y son fundamentales para la detección de fugas y la conservación del agua (GÓMEZ CADENA, 2023).

Medidores Mecánicos

Los medidores mecánicos de agua han sido la tecnología estándar durante décadas. Funcionan mediante un mecanismo de turbina que se mueve con el flujo de agua, registrando el volumen consumido. Estos dispositivos son fiables para flujos de agua moderados a altos, pero pueden ser menos precisos a bajos caudales y tienden a desgastarse con el tiempo (Cunalata Paredes, 2020).

Medidores de Chorro Múltiple

Una variación de los medidores mecánicos, los medidores de chorro múltiple distribuyen el agua de manera más uniforme sobre la turbina, lo que reduce el desgaste y mejora la precisión en aplicaciones residenciales (Tapia Naranjo, 2023).

Medidores Electromagnéticos

Estos dispositivos utilizan la ley de Faraday de la inducción electromagnética para medir la velocidad del agua que fluye a través de un campo magnético en el medidor. No tienen piezas móviles, lo que elimina el desgaste mecánico y mejora la precisión a lo largo del tiempo. Son ideales para aplicaciones industriales donde el agua puede contener partículas (Balseca Quisaguano, 2021).

Medidores Ultrasónicos

Los medidores ultrasónicos utilizan ondas sonoras para medir la velocidad del agua. Dos sensores alternan el envío y la recepción de pulsos ultrasónicos, midiendo el tiempo que tardan estos pulsos en cruzar el flujo de agua. Son extremadamente precisos

y útiles para medir flujos en que la calidad del agua o la presencia de aire pueda afectar otras tecnologías (RODRÍGUEZ YAGUAL, 2020).

Medidores Inteligentes

La última generación de tecnología de medición incluye los medidores inteligentes, que incorporan funcionalidades de comunicación para transmitir datos de consumo de manera remota y en tiempo real. Estos medidores permiten a las empresas de servicios públicos y a los consumidores monitorear el consumo con gran detalle, facilitando la gestión del agua y la identificación rápida de fugas o anomalías en el uso (Andrade Guerrero & Lozada Torres , 2020).

Sistemas de Telemetría

Los sistemas de telemetría se utilizan junto con medidores avanzados para transmitir datos de consumo a través de redes inalámbricas. Estos sistemas permiten la lectura remota y en tiempo real, lo que ayuda a reducir los costos de operación y a mejorar la respuesta ante problemas de suministro o infraestructura (Castillon Sedano, 2020).

Plataformas de Gestión de Datos

Integradas frecuentemente con medidores inteligentes, estas plataformas analizan grandes volúmenes de datos de consumo para proporcionar información detallada sobre patrones de uso, eficiencia en la distribución y optimización de recursos.

2.2.2. Lectura y Registro de Consumo

La lectura y el registro del consumo de agua son procesos esenciales para la gestión eficiente de los recursos hídricos y para garantizar la correcta facturación a los usuarios. Estos procesos involucran la medición del volumen de agua consumido por hogares, empresas, y otros establecimientos durante un período determinado (Rea Chela , 2024).

2.2.2.1. Métodos tradicionales vs. métodos digitales.

Métodos Tradicionales

1. Lectura Manual:

- Proceso: Un lector de medidores visita cada ubicación donde está instalado un medidor de agua y registra manualmente el consumo indicado por el medidor.
- Registro: Los datos son anotados en papel o ingresados en un dispositivo móvil.
- Desafíos: Este método puede ser propenso a errores humanos, es laborioso y requiere mucho tiempo, especialmente en áreas densamente pobladas (Cunalata Paredes, 2020).

Métodos Automatizados

2. Lectura Remota (AMR, Automatic Meter Reading):

- Proceso: Los medidores equipados con tecnología AMR envían datos de consumo automáticamente a la empresa de servicios a través de tecnologías como RF (radiofrecuencia) o dispositivos conectados a la red telefónica.
- Ventajas: Reduce la necesidad de visitas manuales, minimiza errores de lectura y proporciona datos en tiempo real o casi real (Lara, Samper, & Colomé, 2023).

3. Lectura Avanzada de Medidores (AMI, Advanced Metering Infrastructure):

- Proceso: Similar al AMR, pero con capacidades de comunicación bidireccional entre el medidor y el sistema central. Esto permite no solo la lectura automática del consumo sino también la gestión remota del

medidor, como activación/desactivación del servicio y diagnósticos remotos.

- Ventajas: Ofrece una gran cantidad de datos que pueden ser utilizados para análisis detallados del consumo, detección de fugas, y optimización de la distribución de agua (Lara, Samper, & Colomé, 2024).

Medidores Inteligentes:

- Proceso: Incorporan funcionalidades de AMI y proporcionan datos detallados sobre el consumo en intervalos frecuentes (por ejemplo, cada hora).
- Ventajas: Permiten a los consumidores monitorizar su propio consumo en tiempo real, promoviendo el ahorro y la eficiencia en el uso del agua (Bermeo Bermeo, 2021).

Software de Gestión y Análisis:

- Proceso: Los datos recogidos por cualquier método de lectura pueden ser analizados utilizando software especializado para detectar patrones de consumo, identificar consumos anómalos o irregulares y optimizar la gestión de la red de agua.
- Ventajas: Mejora la toma de decisiones basada en datos y facilita una respuesta rápida a problemas emergentes como fugas o fraudes (Montes, y otros, 2021).

2.2.2.2 Importancia de la precisión en la medición.

La precisión en la medición del consumo de agua es fundamental por varias razones, todas contribuyendo a una gestión más eficaz, equitativa y sostenible del recurso hídrico. La precisión en la medición del consumo de agua asegura que los usuarios paguen únicamente por la cantidad de agua que realmente consumen. Esto evita discrepancias y disputas de facturación, contribuyendo a la percepción de justicia y transparencia por parte de la empresa suministradora. La confianza en la precisión de las mediciones

fomenta una relación más positiva entre los proveedores de agua y los consumidores (Ramírez Jiménez, 2021).

Por ello, es importante contar un medidor preciso que ayude a detectar variaciones inusuales en el consumo de agua, lo que puede ayudar a indicar la presencia de fugas.

2.3. Aplicativos Móviles y su Implementación

Los aplicativos móviles son software diseñado específicamente para operar en dispositivos móviles, tales como smartphones y tabletas. Estos aplicativos permiten a los usuarios realizar una variedad de actividades, desde tareas cotidianas hasta funciones complejas, a través de interfaces que están optimizadas para pantallas táctiles y sistemas operativos móviles (Espinoza, Yacelga, & Michilena, 2022).

Características Principales

1. Portabilidad: Diseñados para ser utilizados en dispositivos móviles, los aplicativos pueden ser operados en cualquier lugar donde el usuario lleve su dispositivo.

2. Conectividad: La mayoría de los aplicativos móviles se benefician de la conectividad a internet para ampliar su funcionalidad, facilitando la entrada a datos en tiempo real, la sincronización con servicios en la nube y la comunicación en línea.

3. Interfaz de Usuario (UI): Las interfaces de los aplicativos móviles están diseñadas para ser intuitivas y fáciles de usar, aprovechando las capacidades táctiles y sensoriales de los dispositivos móviles.

4. Sistemas Operativos: Los aplicativos móviles suelen ser creados para ser funcionales en uno o más sistemas operativos móviles, como iOS de Apple o Android de Google (Instituto Federal de Telecomunicaciones, 2022).

2.3.1. Desarrollo de Aplicaciones Móviles

El aumento en la cantidad de aplicaciones disponibles en las tiendas y la amplia variedad de plataformas móviles plantea un desafío para los desarrolladores. Si una empresa decide lanzar una aplicación nativa, debe crear versiones para cada fabricante o plataforma móvil, teniendo en cuenta también las diferentes versiones y tipos de sistemas operativos (BARRERA CHUNGATA & ILLAISACA TENECOTA, 2022).

2.3.1.1 Ciclo de vida del desarrollo de software.

El ciclo de vida del desarrollo de software (SDLC, por sus siglas en inglés) es un proceso sistemático que los desarrolladores siguen para crear software de alta calidad.

Este proceso estructurado incluye varias fases que ayudan a planificar, crear, probar y desplegar un software eficazmente.

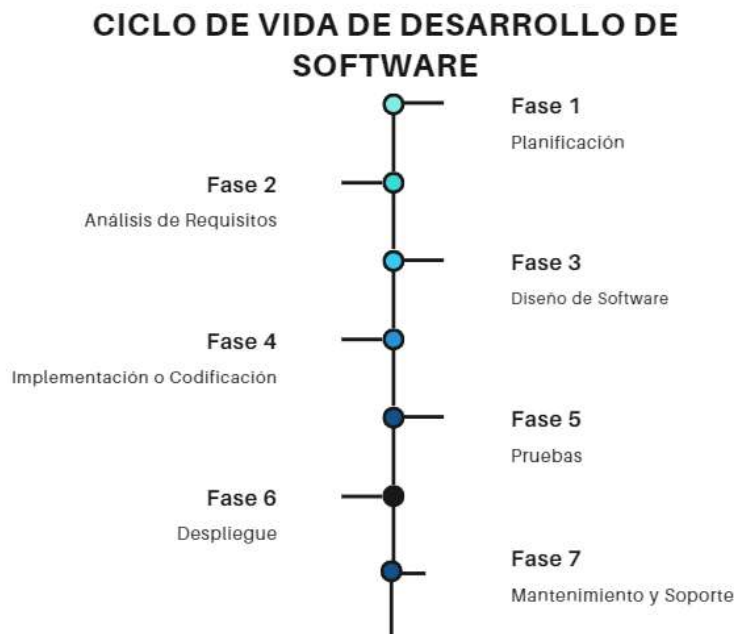


Ilustración 1. Ciclo de desarrollo de software. Fuente: Autoría Propia.

2.3.1.2. Principios de diseño de interfaces de usuario (UI) y experiencia de usuario (UX).

Principios de diseño de interfaces de usuario UI

Cuando se adentra en el entorno digital, se encuentra numerosas aplicaciones que facilitan o incluso son fundamentales en nuestras actividades diarias. Por este motivo, es crucial que, al utilizar productos o servicios digitales, se hace de manera que sea cómoda y segura. Algunos principios clave para garantizar una experiencia de usuario positiva incluyen:

Predecir el comportamiento del usuario

Anticipar el comportamiento del usuario es crucial en el diseño de experiencia de usuario (UX). Como se menciona antes, la investigación de los usuarios forma parte esencial del trabajo del diseñador UX. Esta investigación ayuda a comprender mejor los patrones de comportamiento y las expectativas del usuario. Es fundamental diseñar de manera estratégica para garantizar que el usuario pueda navegar de manera intuitiva por nuestro sitio o aplicación, obteniendo resultados de manera efectiva.

Reducir la carga cognitiva

Para ilustrar, se toma como ejemplo caminando por las calles de Ezequiel Zambrano, un lugar con abundante naturaleza y pocos habitantes. ¿Cómo se sentirá ante tantas luces, carteles para leer y movimiento? Este principio se enfoca en evitar saturar al usuario con muchas opciones, botones mal colocados o ambiguos, lo que puede causar incertidumbre al buscar información y llevar al usuario a abandonar la tarea o frustrarse en el proceso.

Simplificar la funcionalidad

Un ejemplo ilustrativo sería Amazon: al buscar un producto, basta con utilizar la barra de búsqueda para obtener resultados de manera inmediata. Es crucial que el diseño de nuestras interfaces facilite a los usuarios alcanzar sus objetivos con la menor cantidad de clics o acciones posibles.

Aclarar las llamadas en acción

Un ejemplo típico de llamada a la acción es el botón "comprar" o "añadir al carrito" en una página web o aplicación. Es crucial que estos botones sean claros y precisos para evitar confundir al usuario y dificultarle la toma de decisiones.

Administre los errores de manera efectiva

Cuando se producen errores, es fundamental mantener una comunicación transparente con el usuario. Los mensajes de error deben ser precisos y proporcionar información clara para resolverlos de inmediato. Una comunicación ineficaz puede generar frustración en el usuario, pero una comunicación efectiva puede mejorar considerablemente su experiencia (Morejón Labrada, 2020).

Qué es Diseño UX

Se discute el diseño UX (diseño de experiencia de usuario) con el objetivo de definir cómo se construye la experiencia del usuario. El término diseño se refiere principalmente a la "conceptualización" de la experiencia del usuario, más que a su creación visual. (AGUIRRE y otros, 2020)

Principios básicos (UX)

Identificar el problema

Lo principal es saber qué debemos solucionar, para qué estamos diseñando. Es importante definirlo con anticipación antes de empezar, caso contrario estaríamos realizando un trabajo sin saber para dónde apuntar. A futuro, esto puede traer problemas que se podrían haber evitado investigando a fondo para poder encontrar la solución adecuada.

Estructurar

Después de identificar el problema, se organiza el producto con un propósito claro y utilizando modelos coherentes que los usuarios ya están familiarizados y pueden reconocer fácilmente. Esto implica separar elementos distintos y agrupar aquellos que son similares entre sí, asegurando así una estructura coherente y comprensible para el usuario.

Menos, es más

Para que el diseño sea fácilmente reconocible, se debe simplificarlo mediante tareas simples y familiares. Es crucial diseñar de manera sencilla, utilizando pocos elementos para evitar sobrecargar cognitivamente a los usuarios (la carga cognitiva se refiere al esfuerzo mental necesario para completar una tarea). Es esencial que las funciones principales sean identificables rápidamente para mejorar la eficiencia de la experiencia del usuario.

Consistencia

Cuando se menciona consistencia en diseño, nos referimos a mantener la apariencia y la relación entre los elementos a lo largo de toda la experiencia para evitar confusiones. Esto implica mantener la coherencia en aspectos como la tipografía, la paleta de colores, las formas y las interacciones, entre otros. Para lograr esto, es fundamental contar con una

guía de estilos bien definida que nos permita construir cada pantalla de manera uniforme y dinámica. La consistencia facilita que los usuarios reconozcan rápidamente los elementos, mejorando así la fluidez de su interacción.

Jerarquía visual

Una buena forma de ayudar a la consistencia es crear jerarquías visuales, a través de tamaños de elementos, posiciones, colores, espacia-do. Establecer jerarquías permite que los elementos no compitan entre sí, además de permitirnos jugar con las composiciones de diseño.

Retroalimentación

Mantener al usuario bien informado durante su interacción con la interfaz es esencial para que comprenda las acciones que realiza y las consecuencias que pueden surgir, incluyendo posibles errores. Es crucial adaptar el tono y el estilo de comunicación al usuario, utilizando un enfoque claro, conciso y amigable para evitar pareceres impersonales. Además, es fundamental recordar que, como diseñadores, debemos estar receptivos a las necesidades y opiniones de los usuarios, ya que ellos son los que utilizarán la interfaz. Es esencial mejorar y ajustar la interfaz para que sea accesible y fácil de usar según sus necesidades específicas.

Siempre mejorando

Se debe tener en cuenta que el diseño es un proceso continuo; conforme avanzamos y los usuarios interactúan con la interfaz, surgen oportunidades para mejorar y evolucionar la experiencia de diseño. Después de lanzar el proyecto, es crucial evaluar constantemente su desempeño para realizar iteraciones y ajustes necesarios (Medium, 2022).

2.3.1.3 Lenguajes de programación más utilizados

Java

Java es un lenguaje de programación versátil y orientado a objetos que tiene la capacidad de funcionar en una amplia gama de dispositivos en todo el mundo. Es utilizado en diversas aplicaciones, sistemas operativos para smartphones, software empresarial y una variedad de programas reconocidos (Azure, 2023).

Ventajas

- Es un lenguaje de programación centrado en objetos.
- Posee una gestión automática de memoria altamente eficiente.
- Es capaz de operar en cualquier plataforma o dispositivo, sin importar sus requerimientos específicos.
- Su sintaxis es sencilla y fácil de aprender para los desarrolladores.
- Se encuentra entre los lenguajes de programación más seguros disponibles en el mercado laboral.
- Es un lenguaje de alto nivel utilizado para la creación de aplicaciones complejas.
- Su gran estabilidad nos proporciona una flexibilidad total (MOSQUERA CORONEL, 2022).

C#

Es un lenguaje de programación contemporáneo y seguro, orientado a objetos, creado por Microsoft como parte de su plataforma. Cuenta con una comunidad de desarrollo activa y respaldada por Microsoft. Desde su introducción, el lenguaje ha experimentado una evolución notable gracias a contribuciones continuas que lo

mantienen actualizado y relevante. Con acceso a una extensa selección de bibliotecas, frameworks y herramientas, C# sigue siendo una opción ampliamente preferida para iniciar nuevos proyectos de desarrollo en diversas plataformas (Quisaguano Collaguazo , Pallasco Venegas, Andaluz Guerrero, Martines Freire, & Corrales Beltrán, 2022).

Ventajas

- Proporciona un sólido soporte para características como lambdas y genéricos.
- Cuenta con el respaldo de Microsoft Corporation.
- Incluye una consulta integrada de lenguaje (LINQ).
- Ofrece métodos de extensión de fácil implementación.
- Realiza la recolección automática de basura.
- Permite la definición de propiedades con métodos get/set.
- Implementa una gestión automatizada de la memoria.
- Es compatible con versiones anteriores del lenguaje (hartman, 2024).

Dart

Dart es un lenguaje de programación moderno, orientado a objetos y de propósito general, desarrollado por Google y lanzado por primera vez en 2011. Fue diseñado para la construcción de aplicaciones web, móviles y de escritorio. Dart es especialmente conocido por ser el lenguaje detrás de Flutter, el popular framework de UI de Google para desarrollar aplicaciones de alta calidad para iOS, Android, web y escritorio desde una única base de código (MACIAS VERA, 2021).

Dart es utilizado para desarrollar aplicaciones móviles y de escritorio, incluye varias características que apoyan la programación funcional, como funciones de alto orden, funciones lambda y un extenso conjunto de funciones para operar con colecciones (Dart, 2024).

Ventajas

- Versatilidad para el desarrollo multiplataforma.
- Amplia biblioteca para el desarrollo de interfaces.
- Acceso a funciones nativas.
- Renderización y Hot Reload (Altamirano Guerra, 2021).

3.1.3.1 Matriz comparativa de lenguajes de programación

Tabla 1. Matriz comparativa de lenguajes de programación móvil. Fuente: Autoría Propia.

Criterio	Java	C# (C Sharp)	Dart
Compilación	Compilado a bytecode para la máquina virtual Java (JVM)	Compilado a lenguaje intermedio (CIL) para la plataforma .NET	Compilado a código nativo para varias plataformas (iOS, Android, web)
Paradigma	Orientado a objetos	Orientado a objetos	Basado en el framework de UI declarativo
Propósito	Desarrollo de aplicaciones empresariales, móviles y web	Desarrollo de aplicaciones empresariales, de escritorio y web	Creación de aplicaciones móviles para múltiples plataformas
Nivel	Lenguaje de alto nivel	Lenguaje de alto nivel	Marco de desarrollo de aplicaciones

Tipado

Tipado estático

Tipado estático

Tipado estático

Para el desarrollo de un aplicativo móvil de medición del consumo de agua potable con georreferenciación, Dart junto con Flutter emerge como la opción ideal. Dart ofrece una compilación directa a código nativo, optimizando el rendimiento para iOS, Android y web. Flutter, con su framework de UI declarativo, facilita la creación de interfaces modernas y responsivas, mientras que su capacidad de hot reload agiliza el desarrollo y depuración. Esto permite un desarrollo eficiente y rápido de aplicaciones multiplataforma, asegurando una experiencia de usuario fluida y un mantenimiento simplificado a través de un solo código base.

2.3.1.4 Framework de Desarrollo en la actualidad (Flutter, Xamarin, Ionic).

Flutter

La tecnología Flutter emplea el lenguaje de código abierto creado por Google, con la intención de ofrecer a los programadores la posibilidad de trabajar con un lenguaje orientado a objetos. Dart fue creado con la finalidad de simplificar y acelerar el proceso de desarrollo para los programadores (MACIAS VERA, 2021).

Xamarin

Xamarin es una plataforma de código abierto diseñada para desarrollar aplicaciones más avanzadas y eficientes para iOS, Android y Windows mediante el uso de .NET. Funciona como una capa de abstracción que facilita la comunicación entre el código compartido y el código específico de cada plataforma. Opera en un entorno administrado que ofrece ventajas como la gestión de la memoria y la recolección de objetos no utilizados. (microsoft, 2023).

Ionic

Ionic es una colección de recursos de interfaz de usuario de código abierto que permite desarrollar aplicaciones móviles de alta calidad y rendimiento utilizando tecnologías web estándar como HTML, CSS y JavaScript, con soporte integrado para populares frameworks como Angular, React y Vue (ionicframework, 2024).

3.1.4.1 Comparativa de las plataformas de desarrollo.

Tabla 2. Matriz comparativa de las plataformas de desarrollo. Fuente: Autoría Propia

Parámetro	Flutter	Xamarin	Ionic
Lenguaje	Dart	C#	JavaScript, TypeScript
Soporte Multiplataforma	Sí (iOS, Android, web, desktop)	Sí (iOS, Android, Windows)	Sí (iOS, Android, web, desktop)
Curva de Aprendizaje	Moderada	Moderada a alta (requiere conocimiento de C# y .NET)	Moderada (similar a aprender JavaScript)
Rendimiento	Alto (renderización nativa)	Alto (renderización nativa)	Moderado (dependiente de WebView)
Integración con Sistemas Externos	Buena (plugins y paquetes disponibles)	Buena (bindings para librerías nativas)	Buena (plugins y módulos disponibles)
Herramientas de Desarrollo	Flutter SDK, Android Studio, VS Code	Visual Studio, Xamarin Studio	Ionic CLI, Visual Studio Code
Comunidad y Soporte	Amplia y en crecimiento, buen soporte de Google	Amplia, soporte de Microsoft	Amplia, buen soporte de la comunidad

Ionic se destaca como la mejor opción para el desarrollo de aplicaciones debido a su versatilidad y accesibilidad. Su soporte multiplataforma abarca Android, iOS, Windows y aplicaciones web, lo que permite alcanzar una audiencia más amplia con un solo código base. Además, utiliza tecnologías web estándar como HTML, CSS y JavaScript, facilitando una

curva de aprendizaje baja y acelerando el proceso de desarrollo. La integración con Angular mejora la funcionalidad y flexibilidad de las aplicaciones, aprovechando un ecosistema bien establecido.

La amplia gama de plugins y herramientas disponibles en Ionic permite incorporar funcionalidades avanzadas sin desarrollarlas desde cero, agilizando el proceso de desarrollo y asegurando aplicaciones personalizables y escalables. Además, su comunidad activa y soporte en línea ofrecen recursos abundantes, tutoriales y soluciones a problemas comunes, facilitando el desarrollo y mantenimiento de aplicaciones modernas y eficientes. Estos factores combinados hacen de Ionic una herramienta poderosa y eficiente para el desarrollo de aplicaciones versátiles.

2.3.1.5 Bases de datos

2.3.1.5.1 Firebase

Google Firebase puede ser un software de desarrollo de aplicaciones respaldado por Google que permite a los desarrolladores desarrollar aplicaciones web, IOS y Android. Firebase proporciona herramientas para realizar un seguimiento de análisis, generar informes y solucionar fallas de aplicaciones, y crear experimentos de marketing y productos (Vaibhav , Pankaj, & Gaikwad, 2021).



Ilustración 2. Logo de la BD Firebase. Fuente: (Firebase, 2024)

2.3.1.5.2 MySQL

MySQL es un sistema de administración de bases de datos SQL de código abierto, desarrollado y respaldado por Oracle Corporation. Permite organizar y manipular datos de manera eficiente en una computadora utilizando el lenguaje SQL. Es conocido por su velocidad, confiabilidad y facilidad de uso, y ha sido utilizado exitosamente en entornos de producción exigentes durante años (Oracle Corporation, 2024).



Ilustración 3. Base de datos MySQL. Fuente: (MySQL, 2024)

2.3.1.5.3 PostgreSQL

PostgreSQL, a menudo conocido como "Post-GRES", es una base de datos de código abierto que ha ganado reconocimiento por su confiabilidad, versatilidad y adopción de estándares técnicos abiertos. A diferencia de otros sistemas de gestión de bases de datos relacionales (RDBMS), PostgreSQL ofrece soporte para una amplia gama de tipos de datos, tanto relacionales como no relacionales. Esta característica lo posiciona como una de las opciones más maduras, estables y compatibles dentro del ámbito de las bases de datos relacionales en la actualidad (IBM, 2024)



Ilustración 4. Base de datos PostgreSQL. Fuente: (Kinsta, 2023)

2.3.1.6. Cuadro comparativo de Base de datos

Tabla 3. Comparaciones de las bases de datos: Fuente: Autoría propia.

criterio	Firebase	SQL Server	PostgreSQL
Lenguaje	JSON	SQL	SQL
Tipo de base de datos	Document-Based	Relacional	Relacional
Características	<ul style="list-style-type: none"> - Amplia gama de servicios -Escalabilidad automática -Sin administración de servidores 	<ul style="list-style-type: none"> - Amplia gama de servicios - Herramientas de administración robustas - Integración con entornos Windows 	<ul style="list-style-type: none"> - Extensibilidad - Soporte para procedimientos almacenados y funciones definidas por el usuario - Compatible con múltiples sistemas operativos
Ventajas	<ul style="list-style-type: none"> - Sin necesidad de administración de servidores - Escalabilidad automática - Integración con otros servicios de Google Cloud -Datos a tiempo real 	<ul style="list-style-type: none"> - Integración con entornos Microsoft - Herramientas de desarrollo y administración robustas - Buena para aplicaciones empresariales complejas 	<ul style="list-style-type: none"> - Estándares de cumplimiento - Amplio soporte de extensiones y complementos - Buena gestión de datos geoespaciales

Desventajas

- Menos soporte para consultas complejas
- Dependencia de la conectividad con la nube
- Costos variables dependiendo del uso
- Costo de licencia
- Menos soporte para sistemas operativos no-Windows
- Requiere más recursos de hardware
- Menor rendimiento en aplicaciones que requieren muchas escrituras
- Herramientas de administración menos desarrolladas que SQL Server

En la tabla 3 se presenta la comparación de las bases de datos SQL Server y PostgreSQL son bases de datos relacionales que utilizan SQL. SQL Server se destaca por su integración con entornos Microsoft y herramientas de administración robustas, ideales para aplicaciones empresariales complejas, pero su costo de licencia y soporte limitado para sistemas no-Windows son desventajas. PostgreSQL, una base de datos de código abierto, es conocida por su extensibilidad, soporte para procedimientos almacenados y compatibilidad con múltiples sistemas operativos

Firestore es una base de datos basada en documentos que utiliza JSON para almacenar datos, ofreciendo una amplia gama de servicios integrados con Google Cloud. Su principal ventaja es la escalabilidad automática y la eliminación de la necesidad de administrar servidores, facilitando su uso en aplicaciones en tiempo real y proyectos con rápida puesta en marcha. Sin embargo, tiene limitaciones en el soporte de consultas complejas y depende de la conectividad constante a la nube, lo que puede incrementar los costos según el uso, por lo tanto, Firestore es la mejor opción para el desarrollo de la app.

2.3.2 Interoperabilidad de Sistemas

2.3.2.1 Concepto Interoperabilidad

La interoperabilidad se refiere a la capacidad de aplicaciones y sistemas para intercambiar datos de manera segura y automática, sin importar fronteras geográficas, políticas u organizativas. La coordinación en el intercambio de datos entre diferentes organizaciones y departamentos es crucial en muchos sectores, tanto para la investigación y el desarrollo como para mejorar la experiencia del usuario final. La interoperabilidad implica el uso de estándares, protocolos, tecnologías y mecanismos que facilitan el flujo de datos entre distintos sistemas con mínima intervención humana, permitiendo que estos se comuniquen y compartan información en tiempo real (AWS, 2024).

Necesidad de la Interoperabilidad de Sistemas

1. **Eficiencia Operativa:** Facilita la automatización de procesos y la integración de sistemas, reduciendo la duplicación de esfuerzos y mejorando la eficiencia general de las operaciones.
2. **Mejora en la Toma de Decisiones:** Permite acceder a información precisa y oportuna, mejorando así la calidad de las decisiones que se toman.
3. **Reducción de Costos:** Minimiza los gastos asociados con la gestión de datos, el mantenimiento de sistemas y el desarrollo de interfaces personalizadas.
4. **Flexibilidad y Adaptabilidad:** Simplifica la adopción de nuevas tecnologías y sistemas, lo que permite a las organizaciones ajustarse rápidamente a cambios en el entorno tecnológico y comercial.
5. **Experiencia del Usuario:** Mejora la accesibilidad y coherencia en el acceso a la información a través de diversas plataformas y sistemas, mejorando así la experiencia del usuario.

6. **Cumplimiento Normativo:** Facilita el cumplimiento de regulaciones y estándares que requieren la interoperabilidad de sistemas, especialmente en sectores críticos como salud, finanzas y administración pública.
7. **Colaboración y Comunicación:** Fomenta la cooperación entre diferentes entidades, departamentos y sistemas, promoviendo la comunicación efectiva y el intercambio de información.
8. **Innovación:** Estimula la innovación al permitir la integración de múltiples sistemas y tecnologías, facilitando así el desarrollo de nuevos productos y servicios.
9. **Seguridad de Datos:** Garantiza que los datos se intercambien de manera segura, protegiendo la privacidad y la integridad de la información transmitida. (Mejía Bárcen y otros, 2024)

2.3.2.2 Protocolos y estándares para la integración de sistemas (API, XML, JSON).

API

Una API, o interfaz de programación de aplicaciones, consiste en un conjunto de especificaciones y protocolos utilizados para diseñar e integrar el software de aplicaciones. Las APIs posibilitan que productos y servicios se comuniquen entre sí sin requerir conocimiento detallado de sus implementaciones subyacentes. Esta capacidad simplifica el proceso de desarrollo de aplicaciones, resultando en ahorro de tiempo y recursos financieros. Además, las APIs proporcionan flexibilidad, simplifican el diseño y la gestión de aplicaciones, y fomentan oportunidades innovadoras tanto en la creación de nuevas herramientas y productos como en la optimización de los existentes (Redhat, 2024).

Las API pueden funcionar de cuatro maneras diferentes, según el momento y el motivo de su creación:

- **API de SOAP:** Utilizan el protocolo Simple Object Access Protocol. En este tipo de API, el intercambio de mensajes entre cliente y servidor se realiza mediante XML. Aunque menos flexible, fue muy popular en el pasado.
- **API de RPC:** Conocidas como llamadas a procedimientos remotos, permiten al cliente invocar funciones o procedimientos en el servidor y recibir el resultado correspondiente.
- **API de WebSocket:** Esta es una evolución moderna de las API web, que utiliza objetos JSON para la transmisión de datos. La API de WebSocket facilita la comunicación bidireccional entre aplicaciones cliente y servidor, permitiendo al servidor enviar mensajes de devolución de llamada a los clientes conectados, lo que la hace más eficiente que las API de REST en ciertos contextos.
- **API de REST:** Son las más populares y flexibles en la web actual. En este tipo de API, el cliente envía solicitudes al servidor como datos, y el servidor utiliza esta entrada para ejecutar funciones internas y devolver los resultados al cliente (AWS, 2024).

XML

XML es el acrónimo de extensible markup language (lenguaje de marcado extensible), es decir, es un lenguaje de marcas. Un lenguaje de marcas (o de marcado) es un conjunto de reglas que sirven para codificar un documento. XML no es en sí mismo un lenguaje de programación en el que incluir condicionales, variables, bucles y otros elementos de computación, pero sí es ampliamente utilizado por programadores para acceder a los datos y transmitirlos en sus aplicaciones (García Bonacho, 2023).

JSON

JSON (JavaScript Object Notation) es un formato ligero y accesible para intercambiar datos entre sistemas. Diseñado para ser fácilmente legible y editable por humanos, JSON también se puede analizar y generar eficientemente mediante procesos automatizados. Aunque está basado en un subconjunto del lenguaje de programación JavaScript, específicamente el estándar ECMA-262 3rd Edition de diciembre de 1999, JSON es completamente independiente del lenguaje. Sus convenciones son familiares para programadores de diversos lenguajes, como C, C++, C#, Java, JavaScript, Perl, Python y otros más. Estas características hacen de JSON una opción ideal para facilitar el intercambio de datos entre diferentes plataformas y aplicaciones de manera eficaz y sin complicaciones (ibm, 2024).

3.2.1.1 Comparativa Protocolos y estándares para la integración de sistemas

Tabla 4. Matriz Comparativa Matriz comparativa Protocolos y estándares para la integración de sistemas. Fuente: Autoría Propia.

Parámetro	API	XML	JSON
Formato de Datos	Generalmente basado en texto (JSON, XML, etc.)	Basado en texto plano con estructura jerárquica mediante etiquetas	Basado en texto plano con estructura simple de pares clave-valor

Uso Principal	Facilitar la comunicación entre sistemas	Representar y transportar datos estructurados	Intercambio de datos estructurados en aplicaciones web y servicios REST
Sintaxis	Depende del tipo de API (SOAP, REST, etc.)	Utiliza etiquetas anidadas para definir estructura y metadatos	Utiliza pares de clave-valor para definir estructura de datos
Estructura			
Flexibilidad	Puede variar según el tipo de API implementada	Ofrece flexibilidad en la definición de estructuras complejas	Muy flexible y fácil de modificar según las necesidades de los datos
Interoperabilidad	Facilita la interoperabilidad entre sistemas	Favorece la interoperabilidad entre diferentes aplicaciones y plataformas	Ampliamente compatible con diversos lenguajes y sistemas
Lectura	Depende de la implementación específica	Requiere análisis y procesamiento para extraer datos estructurados	Fácilmente legible y editable por humanos y máquinas
Escritura			

Documentación	Requiere documentación específica y detallada	La estructura XML permite describir la información con metadatos	Documentación clara y concisa, generalmente fácil de entender y seguir
Ejemplo Representativo	<pre>json {"nombre": "Juan", "edad": 30}</pre>	<pre>xml <persona><nombre>Juan</nombre><edad>30</edad></persona></pre>	<pre>json {"nombre": "Juan", "edad": 30}</pre>

Los formatos API, XML y JSON son utilizados para diferentes propósitos y tienen características distintivas. API se enfoca en facilitar la comunicación entre sistemas, mientras que XML se destaca por su capacidad para representar datos estructurados mediante etiquetas jerárquicas, favoreciendo la interoperabilidad entre diferentes aplicaciones. Por otro lado, JSON es altamente flexible y utiliza pares clave-valor para intercambiar datos estructurados en servicios web y aplicaciones REST, siendo fácilmente legible y editable. Cada uno requiere diferentes niveles de documentación y posee sintaxis específicas que los hacen adecuados para diferentes contextos de desarrollo y comunicación de datos.

2.4. Georreferenciación y Sistemas de Información Geográfica (SIG)

4.1 Georreferenciación

La georreferenciación es el proceso de asociar información geográfica a datos, permitiendo que estos sean representados en un mapa utilizando sistemas de coordenadas. Esto implica asignar coordenadas geográficas específicas a diversos tipos de datos, como

imágenes, videos, medidas de sensores o cualquier otro elemento que pueda ser localizado espacialmente en la Tierra (Chapman & Wiczorek, 2020).

Dimensiones típicas de la georreferenciación:

1. **Coordenadas geográficas:** Durante el proceso de georreferenciación, se asignan coordenadas precisas a eventos o datos, abarcando dimensiones como latitud, longitud y, en algunos casos, altitud.
2. **Sistema de referencia espacial:** Este elemento se compone de un conjunto de reglas y parámetros que definen un sistema de coordenadas geográficas, incluyendo datos geodésicos, proyecciones de mapas y escala.
3. **Métodos de georreferenciación:** La georreferenciación se realiza mediante diversas tecnologías como el Sistema de Posicionamiento Global (GPS), Sistemas de Información Geográfica (SIG) e imágenes satelitales. Las dimensiones asociadas incluyen precisión, resolución espacial y disponibilidad de datos.
4. **Datos geográficos:** Durante la georreferenciación, se establece la conexión de datos con ubicaciones específicas, incluyendo la naturaleza de los datos, que pueden ser puntos, líneas o polígonos (Oliva Gamarra, 2024).

2.4.1. Conceptos Básicos de Georreferenciación

2.4.1.1. Definición y utilidad de la georreferenciación.

La georreferenciación es el proceso de asociar información geográfica a datos, permitiendo que estos sean representados en un mapa utilizando sistemas de coordenadas. Esto implica asignar coordenadas geográficas específicas a diversos tipos de datos, como imágenes, videos, medidas de sensores o cualquier otro elemento que pueda ser localizado espacialmente en la Tierra (Chapman & Wiczorek, 2020).

Dimensiones típicas de la georreferenciación:

- 1. Coordenadas geográficas:** Durante el proceso de georreferenciación, se asignan coordenadas precisas a eventos o datos, abarcando dimensiones como latitud, longitud y, en algunos casos, altitud.
- 2. Sistema de referencia espacial:** Este elemento se compone de un conjunto de reglas y parámetros que definen un sistema de coordenadas geográficas, incluyendo datos geodésicos, proyecciones de mapas y escala.
- 3. Métodos de georreferenciación:** La georreferenciación se realiza mediante diversas tecnologías como el Sistema de Posicionamiento Global (GPS), Sistemas de Información Geográfica (SIG) e imágenes satelitales. Las dimensiones asociadas incluyen precisión, resolución espacial y disponibilidad de datos.
- 4. Datos geográficos:** Durante la georreferenciación, se establece la conexión de datos con ubicaciones específicas, incluyendo la naturaleza de los datos, que pueden ser puntos, líneas o polígonos (Oliva Gamarra, 2024)
- 5. Coordenadas geográficas:** Es un sistema que utiliza la latitud y la longitud para definir la ubicación de un punto en la superficie de la Tierra. La latitud indica la posición norte o sur del ecuador, mientras que la longitud indica la posición este u oeste del meridiano de Greenwich.
- 6. Sistema de Coordenadas:** Los sistemas de coordenadas permiten que cada ubicación en la Tierra sea especificada por un conjunto de números. Los sistemas más comunes son:
- 7. Sistema de Coordenadas Geográficas (Latitud y Longitud):** Usa grados para definir posiciones en la Tierra.

- 8. Sistema de Coordenadas Proyectadas:** Transforma la superficie curva de la Tierra en un plano 2D, lo cual es útil para mapas y mediciones precisas. Ejemplos incluyen UTM (Universal Transverse Mercator) y SPCS (State Plane Coordinate System).

Utilidad de la georreferenciación

- **Visualización Espacial:** Permite representar datos en mapas interactivos, facilitando la comprensión de patrones espaciales y relaciones geográficas.
- **Análisis Geoespacial:** Facilita la realización de análisis y estudios específicos basados en la ubicación, como el estudio de patrones de migración, distribución de recursos naturales, o la planificación urbana.
- **Gestión de Recursos:** Ayuda en la gestión eficiente de recursos naturales y urbanos, al permitir la localización precisa de activos y la planificación de infraestructuras.
- **Navegación y Servicios Basados en Ubicación:** Es fundamental en aplicaciones de navegación GPS, servicios de transporte, búsqueda local y servicios de emergencia.
- **Seguimiento y Monitorización:** Permite el seguimiento en tiempo real de vehículos, personas o fenómenos naturales como huracanes o incendios forestales.
- **Estudios Ambientales:** Facilita la evaluación de impactos ambientales, monitoreo de cambios en el paisaje y gestión de áreas protegidas. (Oliva Gamarra, 2024).

2.4.1.2 Métodos y tecnologías para la georreferenciación (GPS, triangulación, etc.).

Sistemas de Información Geográfica (GIS)

Estos sistemas tienen una amplia gama de aplicaciones en distintos campos como la planificación urbana y territorial, salud pública, respuesta a emergencias, agricultura, gestión de recursos naturales, entre otros. Es por ello, que los GIS son vitales no solo por su capacidad de visualización y análisis, sino también por cómo permiten a los decisores combinar diversas

capas de información para tomar decisiones más informadas y efectivas en contextos complejos y cambiantes. Esta integración de datos espaciales y no espaciales en un solo sistema ofrece un enfoque holístico que es indispensable en muchos campos profesionales y académicos (Gary R Lock, 2022).

Los Sistemas de Navegación Global por Satélite (GNSS)

Los Sistemas de Navegación Global por Satélite (GNSS) consisten en constelaciones de satélites que emiten señales desde el espacio, las cuales contienen información sobre posicionamiento y tiempo. Los receptores GNSS captan estas señales y las usan para calcular su localización precisa (latitud, longitud y altitud) en cualquier parte del mundo, en cualquier momento y bajo cualquier condición climática (aregis, 2023).

Triangulación

La triangulación es una técnica geométrica que se utiliza para localizar un punto al calcular las distancias o los ángulos desde al menos tres puntos conocidos. Este método consiste en medir los ángulos entre el punto desconocido y los puntos de referencia, formando un triángulo, y luego emplear las propiedades de los triángulos para determinar las coordenadas del punto objetivo. La triangulación se aplica en diversas áreas, como la topografía, la navegación y la astronomía, siendo esencial para tecnologías de posicionamiento y cartografía (Lara Cordón, 2020).

Trilateración

La trilateración es un método geométrico que se utiliza para determinar la posición de un punto mediante la medición de las distancias desde al menos tres puntos conocidos. A diferencia de la triangulación, que se basa en medir ángulos, la trilateración calcula la posición exacta utilizando únicamente las distancias entre el punto desconocido y los puntos de

referencia. Conocidas estas distancias, se pueden establecer esferas (en 3D) o círculos (en 2D) alrededor de los puntos de referencia, y la intersección de estas formas geométricas revela la ubicación precisa del punto objetivo. Este método es crucial en sistemas de posicionamiento global como el GPS y otros sistemas GNSS (Gargantilla, 2020).

Sistemas de Información Geográfica (GIS)

Estos sistemas tienen una amplia gama de aplicaciones en distintos campos como la planificación urbana y territorial, salud pública, respuesta a emergencias, agricultura, gestión de recursos naturales, entre otros. Es por ello, que los GIS son vitales no solo por su capacidad de visualización y análisis, sino también por cómo permiten a los decisores combinar diversas capas de información para tomar decisiones más informadas y efectivas en contextos complejos y cambiantes. Esta integración de datos espaciales y no espaciales en un solo sistema ofrece un enfoque holístico que es indispensable en muchos campos profesionales y académicos (Gary R Lock, 2022).

2.4.2 Tecnologías y Plataformas para SIG en Aplicativos Móviles

Los Sistemas de Información Geográfica (SIG) en aplicativos móviles han revolucionado la forma en que se interactúa con datos geoespaciales, proporcionando herramientas poderosas para la visualización, la recopilación y el análisis de información geográfica directamente desde dispositivos móviles.

ArcGIS

Desarrollado por Esri, ArcGIS es uno de los sistemas de información geográfica más robustos y ampliamente utilizados. Ofrece una suite completa para aplicaciones móviles, incluyendo ArcGIS Online, una plataforma basada en la nube que permite a los usuarios crear, gestionar y compartir mapas y aplicaciones. ArcGIS para aplicativos móviles permite a los

usuarios acceder a mapas, realizar análisis espaciales y capturar datos en el campo (Arcgis, 2023).

QGIS

Es una opción de software libre y de código abierto que es muy popular en la comunidad GIS. Aunque es más conocido por su aplicación de escritorio, QGIS también se puede utilizar en dispositivos móviles a través de aplicaciones como QField para QGIS, que sincroniza proyectos de QGIS directamente en el campo, permitiendo la recolección y edición de datos (Gazaba, 2020)

Google Maps API

La API de Google Maps JavaScript ofrece la posibilidad de adaptar los mapas con imágenes y contenido personalizados para su visualización en páginas web y dispositivos móviles. Esta API proporciona cuatro tipos de mapas básicos (mapa de carreteras, satélite, híbrido y terreno), los cuales se pueden personalizar a través de capas y estilos, controles y eventos, y varios servicios y bibliotecas disponibles (Google Maps Platform, 2024).

GeoServer

GeoServer es un servidor que permite a los usuarios compartir y editar datos geoespaciales. Aunque no es una aplicación móvil en sí misma, puede ser utilizado en conjunto con aplicaciones móviles para proporcionar un potente back-end que maneje datos geoespaciales complejos y los sirva en formatos estándar de la industria a través de servicios web (Gaytán-Lugo M. y otros, 2020)

2.4.2.1 Plataformas de desarrollo para aplicativos móviles con SIG

2.4.2.1.1. Android Studio

Android Studio, el entorno de desarrollo integrado (IDE) oficial de Android, sucedió al Eclipse IDE en 2013. Ofrecido gratuitamente bajo la licencia Apache 2.0, se basa en el software IntelliJ IDEA. La versión estable inicial fue lanzada en diciembre de 2014. Este IDE proporciona un amplio conjunto de herramientas que facilitan la creación de aplicaciones para dispositivos Android, incluyendo funciones para diseñar interfaces de usuario y escribir código en Kotlin o Java, el lenguaje oficial (Santos González, 2023).



Ilustración 5. Android Studio. Fuente: Autoría Propia

Funcionalidades de Android Studio:

- Un sistema de compilación adaptable basado en Gradle.
- Un emulador rápido y versátil.
- Un entorno unificado que permite el desarrollo para todos los dispositivos Android.
- Función de "Apply Changes" que permite realizar cambios en el código y los recursos de la aplicación en ejecución sin necesidad de reiniciarla.
- Integración con GitHub y plantillas de código para simplificar el desarrollo de funciones comunes de aplicaciones y facilitar la importación de código de muestra.
- Diversidad de frameworks y herramientas de prueba disponibles.

- Herramientas de Lint para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros aspectos.
- Compatibilidad con C++ y NDK.
- Integración nativa con Google Cloud Platform, que simplifica la incorporación de servicios como Google Cloud Messaging y App Engine (Android Developers, 2024).

2.4.2.1.2. IOS (Xcode)



Ilustración 6. XCODE. Fuente: (applesfera, 2024)

Xcode te permite desarrollar, probar y distribuir aplicaciones para todas las plataformas de Apple. Codifica y diseña tus aplicaciones más rápido con finalización de código mejorada, vistas previas interactivas y animaciones en vivo. Usa Git staging para crear tu próxima confirmación sin salir de tu código. Explora y diagnostica los resultados de tus pruebas con informes de pruebas rediseñados con grabación de video. Y comienza a implementar sin problemas en TestFlight y la App Store desde Xcode Cloud. Crear aplicaciones increíbles nunca ha sido tan fácil (developer, 2024).

Características XCODE

Tabla 5. Características de XCODE. Fuente: Autoría Propia.

Parámetro	Descripción
Lenguaje	Swift, Objective-C
Ventajas	<ul style="list-style-type: none"> - Integración completa con el ecosistema de Apple - Herramientas nativas avanzadas - Excelentes capacidades de depuración - Soporte completo para las APIs de Apple
Soporte SIG	<ul style="list-style-type: none"> - MapKit: Anotaciones, overlays, geocodificación, rutas, integración con Apple Maps - ArcGIS Runtime SDK for iOS: Mapas 2D/3D, análisis espacial, acceso a datos SIG

2.4.2.1.3. Cross-Platform

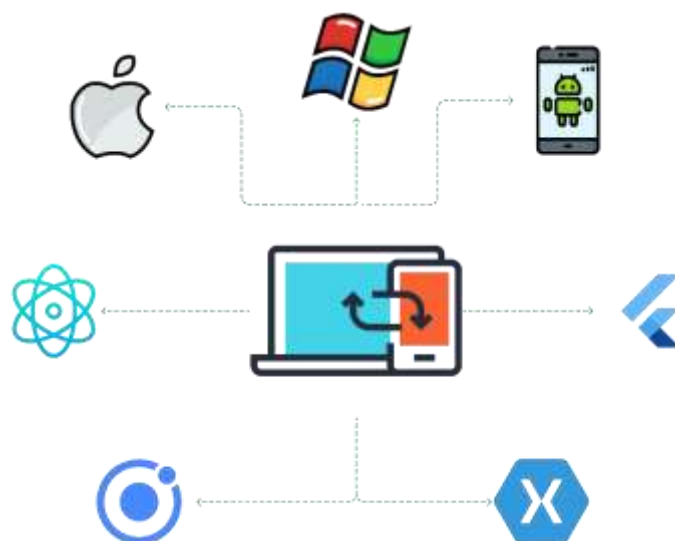


Ilustración 7. Cross-Platform. Fuente: (cleancommit, 2022)

El término "Cross-Platform" (plataforma cruzada) se refiere a la capacidad de desarrollar aplicaciones que puedan funcionar en múltiples plataformas o sistemas operativos con un solo conjunto de código base. En el contexto del desarrollo de aplicaciones móviles, esto implica crear aplicaciones que puedan ejecutarse tanto en dispositivos Android como iOS,

utilizando un lenguaje de programación y un conjunto de herramientas que permitan la portabilidad del código entre estas plataformas.

Los frameworks Cross-Platform son herramientas que facilitan este proceso al proporcionar entornos de desarrollo que permiten escribir y compilar una aplicación una vez, y luego desplegarla en diferentes plataformas con ajustes mínimos o nulos (Shevtsiv & Striuk, 2020).

Estas son las plataformas en las que se pueden desarrollar:

Tabla 6. Plataformas de Desarrollo Cross-Platform para Aplicativos Móviles con SIG. Fuente: Autoría Propia

Parámetro	Ionic	Flutter	React Native	Xamarin
Lenguaje	HTML, CSS, Dart JavaScript (con Angular)		JavaScript	C#
Ventajas	- Desarrollo rápido con tecnologías web estándar - Soporte para múltiples plataformas (Android, iOS, Web) - Amplia gama de plugins y comunidad activa	- Alta performance - Un solo código base para múltiples plataformas- Excelente soporte de UI y widgets	- Amplia comunidad de desarrolladores - Un solo código base para Android y iOS - Soporte de librerías de terceros	- Buen rendimiento y capacidad para compartir código entre plataformas - Integración con Visual Studio

Soporte SIG - Plugins de - Plugins - react-native - Xamarin.Forms.Maps
 Cordova y como -maps para - ArcGIS Runtime SDK
 Capacitor para flutter_map Google Maps y for .NET
 integraciones (para Mapbox
 con Google OpenStreet
 Maps y Map) y
 OpenStreetMa google_map
 p s_flutter

2.4.2.1.4. Matriz comparativa de plataformas de desarrollo de aplicaciones móviles

Tabla 7. Matriz comparativa de las plataformas de desarrollo. Fuente: Autoría Propia.

Característica	Android	iOS	Cross-Platform
Lenguajes de programación	Java, Kotlin	Swift, Objective-C	Dart (Flutter), JavaScript (React Native), C# (Xamarin)
Entorno de desarrollo (IDE)	Android Studio	Xcode	Varios (VS Code, Android Studio, Xcode, etc.)
Plataforma de distribución	Google Play Store	Apple App Store	Google Play Store, Apple App Store, otros
Fragmentación	Alta fragmentación debido a la variedad de dispositivos	Baja fragmentación, controlado por Apple	Depende del framework usado y las plataformas objetivo
Costo de publicación	Pago único de \$25	Suscripción anual de \$99	Varía según las plataformas a publicar
Interfaz de usuario (UI/UX)	Guidelines de Material Design	Guidelines de Human Interface	Depende del framework, puede seguir ambas guidelines

Acceso a características del dispositivo	Completo, pero requiere permisos explícitos	Completo, pero bajo control estricto de Apple	Generalmente accesible, aunque puede haber limitaciones
Rendimiento	Nativo, generalmente alto	Nativo, generalmente alto	Puede ser cercano al nativo, pero a veces inferior
Curva de aprendizaje	Media	Alta, especialmente Swift	Depende del framework, puede ser baja o media
Comunidades y soporte	Amplias comunidades, buen soporte	Amplias comunidades, buen soporte	Creciente, pero generalmente menos que nativo
Actualizaciones del sistema operativo	Rápidas, pero dependencia de fabricantes	Rápidas y controladas directamente por Apple	Depende del framework y el sistema operativo
Tiempos de desarrollo	Variable, generalmente más largo que cross-platform	Variable, generalmente más largo que cross-platform	Generalmente más corto, reutilización de código
Costo de desarrollo	Variable, puede ser alto si se desarrollan múltiples apps	Variable, puede ser alto si se desarrollan múltiples apps	Generalmente más bajo debido a la reutilización de código

La elección de Flutter como solución cross-platform se fundamenta en su capacidad para compilar código Dart directamente a código nativo, asegurando un rendimiento optimizado en Android y iOS. Flutter ofrece un entorno robusto con widgets personalizables y soporte integrado para directrices de diseño como Material Design y Cupertino, permitiendo a los desarrolladores crear interfaces de usuario coherentes y atractivas. Su flexibilidad y eficiencia en el desarrollo se complementan con una distribución simplificada a través de Google Play Store y Apple App Store, respaldada por una comunidad activa y soporte continuo de Google, lo que garantiza actualizaciones frecuentes y resolución rápida de problemas, ideal

para proyectos que requieren rapidez, calidad y compatibilidad multiplataforma sin comprometer el rendimiento nativo.

2.4.3. Herramientas y bibliotecas de software para SIG móviles (ArcGIS SDK, Google Maps API, Mapbox, Leaflet).

Para desarrollar aplicaciones móviles con capacidades de Sistemas de Información Geográfica (SIG), se pueden utilizar diversas herramientas y bibliotecas de software que facilitan la integración de funciones geoespaciales. Estas herramientas permiten desde la simple visualización de mapas hasta análisis espaciales complejos y gestión de datos geográficos.

1. Esri ArcGIS Runtime SDKs

- **Descripción:** Esri ofrece SDKs para el desarrollo de aplicaciones móviles que necesitan funcionalidades SIG robustas. Estos SDKs están disponibles para plataformas como iOS, Android, .NET, Java y Qt.
- **Funcionalidades:** Visualización de mapas, edición, análisis geoespacial, y acceso a datos geográficos en tiempo real (Arcgis, 2023).

2. Mapbox Mobile SDKs

- **Descripción:** Mapbox proporciona SDKs específicos para Android e iOS que permiten desarrollar aplicaciones móviles con mapas personalizados y potentes funciones de análisis geográfico.
- **Funcionalidades:** Creación de mapas estilizados, navegación, localización y rastreo en tiempo real, integración de capas personalizadas (mapbox, 2023).

3. Google Maps API

- **Descripción:** Las APIs de Google Maps para Android e iOS son ampliamente utilizadas para integrar mapas dentro de aplicaciones móviles, ofreciendo una rica base de datos de lugares y capacidades de visualización.
- **Funcionalidades:** Mapeo básico, geocodificación, rutas, y visualización de información detallada de lugares (Google Maps Platform, 2024).

4. Leaflet

- **Descripción:** Aunque principalmente es una biblioteca de JavaScript para aplicaciones web, Leaflet puede ser utilizado en aplicaciones móviles a través de frameworks como Cordova o Ionic.
- **Funcionalidades:** Funciones de mapeo interactivas y ligeras, fácil de extender con plugins para diversas necesidades geoespaciales (leafletjs, 2024).

5. OpenLayers

- **Descripción:** Similar a Leaflet, OpenLayers también se puede usar en aplicaciones móviles combinándolo con frameworks como Cordova. Ofrece una API más robusta para trabajar con diferentes fuentes de mapas y funciones geográficas.
- **Funcionalidades:** Soporte para múltiples fuentes de mapas, capas vectoriales y rasterizadas, interacciones avanzadas (openlayers, 2024).

6. QGIS

- **Descripción:** QField es una aplicación móvil que sincroniza con QGIS, permitiendo llevar proyectos SIG completos al campo.

- **Funcionalidades:** Edición de datos geográficos en el terreno, compatibilidad con todos los formatos y plugins de QGIS (Arcgis, 2023).

7. Carto Mobile SDK

- **Descripción:** Carto ofrece un SDK que se centra en la visualización y análisis de grandes conjuntos de datos geospaciales, ideal para aplicaciones de inteligencia de ubicación.
- **Funcionalidades:** Análisis en tiempo real, visualización de datos geográficos complejos, integración con la plataforma de análisis Carto (carto, 2024).

8. Flutter Plugins

- **Descripción:** Para desarrolladores que usan Flutter, hay varios plugins disponibles que permiten integrar funcionalidades SIG, como flutter_map para Leaflet o plugins específicos para Google Maps y Mapbox.
- **Funcionalidades:** Dependiendo del plugin, puede incluir desde visualización de mapas hasta funciones de georreferenciación y análisis espacial (Morales, 2022).

9. GeoServer

- **Descripción:** Aunque no es una herramienta de desarrollo móvil, GeoServer se puede usar para servir datos geográficos a aplicaciones móviles a través de servicios web como WMS y WFS.
- **Funcionalidades:** Servicio de mapas y datos espaciales, soporte para múltiples formatos y estándares SIG (Gaytán-Lugo M. y otros, 2020).

2.4.3.2. Comparativa de tecnologías y sus aplicaciones.

Tabla 8. Comparación de tecnologías y sus aplicaciones. Fuente: Autoría Propia.

Tecnología	Aplicaciones	Ventajas	Desventajas	Costo
Esri ArcGIS	Urbanismo, gestión de recursos naturales, defensa, emergencias	Integración completa, soporte sólido, analítica avanzada	Costo alto, curva de aprendizaje	Alto
Google Maps API	Aplicaciones móviles/web, navegación, turismo, servicios de entrega	Fácil uso, buena documentación, fuerte en visualización	Menos adecuado para análisis avanzado, dependencia de Google	Medio (basado en uso)
Mapbox	Aplicaciones móviles personalizadas, logística, transporte, juegos	Personalizable, optimizado para móviles, herramientas para desarrolladores	Costo elevado a gran escala	Medio-Alto (basado en uso)
OpenLayers	Aplicaciones web GIS, proyectos de código abierto, educación	Flexible, soporta múltiples fuentes de datos, código abierto	Complejidad de configuración, documentación menos accesible	Gratis (código abierto)
Leaflet	Aplicaciones web ligeras, visualización interactiva, proyectos educativos	Ligero, fácil de aprender, comunidad activa	Limitado en funciones avanzadas, soporte móvil limitado	Gratis (código abierto)
QGIS	Investigación, educación, gestión de recursos, planificación urbana	Código abierto, soporta muchos formatos de datos, comunidad activa	Menos integrado para aplicaciones en tiempo real, interfaz menos intuitiva	Gratis (código abierto)

Carto	Inteligencia de ubicación, marketing, análisis demográfico	Fácil de usar, fuerte en visualización y análisis, basado en la nube	Menos adecuado para datos en tiempo real, costos de suscripción	Medio (basado en suscripción)
--------------	--	--	---	-------------------------------

Entre las herramientas de sistemas de información geográfica (SIG) evaluadas, Google Maps API sobresale como la mejor elección gracias a su facilidad de uso, documentación excelente y capacidades superiores de visualización, lo que simplifica su integración en aplicaciones móviles y web. Su amplia compatibilidad y escalabilidad permiten a los desarrolladores crear soluciones robustas para una variedad de sectores, desde la navegación y el turismo hasta los servicios de entrega. Además, la confiabilidad y el soporte continuo ofrecidos por Google garantizan un rendimiento eficiente y actualizado, posicionando a Google Maps API como la opción preferida frente a otras herramientas como Esri ArcGIS, Mapbox, OpenLayers, Leaflet, QGIS y Carto.

CAPÍTULO III

3. MARCO METODOLÓGICO

3.1. Enfoque de la investigación

La presente investigación se realizará bajo la metodología MOBILE-D, utilizando un enfoque mixto. Se emplearán tanto métodos cuantitativos como cualitativos para desarrollar una propuesta de aplicativo móvil que permita la medición del consumo de agua potable utilizando georreferenciación para la EMAPAT. Este enfoque mixto permitirá no solo recolectar y analizar datos numéricos precisos del consumo, sino también comprender las percepciones y comportamientos de los usuarios y técnicos que interactúan con el aplicativo.

3.2. Nivel de la investigación

La investigación se sitúa en un nivel exploratorio y descriptivo. Inicialmente, explorará las posibilidades y desafíos de integrar tecnologías de georreferenciación en los procesos de medición del consumo de agua para identificar nuevas oportunidades de mejora. Paralelamente, describirá cómo el aplicativo influye en las prácticas actuales de gestión del agua, evaluando su impacto en la precisión de la recolección de datos y la eficiencia operativa. Este enfoque mixto no solo permitirá comprender mejor las dinámicas actuales, sino que también facilitará la implementación efectiva del aplicativo, asegurando que se maximicen los beneficios en términos de operatividad y aceptación por parte de los usuarios y técnicos de EMAPAT de EL TAMBO.

3.3. Población

La población de esta investigación se centra en el gerente de la EMAPAT.

3.4. Técnicas e instrumentos de recolección

Para la recolección de datos, se utilizará una entrevista como la principal técnica de investigación. Esta entrevista estará diseñada para capturar tanto datos cuantitativos como cualitativos. Se dirigirá al gerente de EMAPAT de El Tambo con el objetivo de obtener información detallada sobre sus experiencias, percepciones y necesidades en relación con el consumo de agua potable y la utilización de tecnologías de georreferenciación. La entrevista incluirá preguntas abiertas y cerradas que permitirán explorar a fondo los hábitos de consumo, las expectativas respecto al nuevo aplicativo y cualquier desafío percibido en su implementación y uso. La información obtenida será crucial para ajustar y mejorar el diseño del aplicativo, asegurando que responda adecuadamente a las necesidades y expectativas de sus usuarios.

3.5. Tratamiento de la información

La información recabada durante la entrevista será sistematizada y procesada meticulosamente, además de realizarse un análisis exhaustivo de los resultados obtenidos.

3.6. Resultados

Para determinar las necesidades de la EMAPAT en El Tambo, se implementará un proceso de recopilación de datos a través de la entrevista. Tras la formulación de las preguntas que conformarán, se llevará a cabo la fase de administración del mismo. Finalmente, el proceso concluirá con un análisis detallado y la interpretación de la información obtenida.

3.6.1. Análisis de los resultados

Preguntas	Respuesta
¿Utiliza actualmente algún tipo de software o aplicación móvil para la gestión de la EMAPAT?	No, actualmente no utilizamos ningún software o aplicación móvil para la gestión de la EMAPAT. Estamos buscando opciones que puedan mejorar la eficiencia y el control en nuestras operaciones.
¿Considera que un aplicativo móvil para la medición del consumo de agua potable es necesario para la EMAPAT?	Sí, considero que un aplicativo móvil para la medición del consumo de agua potable es necesario. Esto permitiría un monitoreo más eficiente y en tiempo real del consumo, mejorando la gestión y el control del recurso. También facilitaría la recopilación de datos precisos y la implementación de medidas para optimizar el uso del agua.
¿Cuáles cree que son los principales beneficios que un aplicativo móvil podría aportar a la EMAPAT?	Creo que los principales beneficios de un aplicativo móvil incluyen la capacidad de monitorear el consumo de agua en tiempo real, lo que optimizaría nuestra gestión del recurso y permitiría una respuesta rápida a cualquier anomalía. Además, la automatización en la recolección de datos reduciría los errores y el tiempo de procesamiento, mientras que las alertas y notificaciones mejorarían la eficiencia operativa y fomentarían un uso más responsable del agua.

<p>¿Qué funcionalidades considera esenciales para el aplicativo móvil?</p>	<p>Considero esenciales el registro de consumo de agua en tiempo real para un monitoreo continuo y la georreferenciación de mediciones para ubicar y analizar el consumo en diferentes áreas. Estas funcionalidades permitirán una gestión más eficaz y precisa del recurso hídrico.</p>
<p>¿Qué funciones debería incluir el aplicativo móvil para medir el consumo de agua potable?</p>	<p>Considero que el aplicativo móvil debería incluir las funciones de medición del consumo de agua y registro de ubicación mediante georreferenciación. Estas características permitirán una gestión precisa y detallada del consumo, además de facilitar la identificación de áreas específicas de interés.</p>
<p>¿Qué dispositivos móviles se utilizarán principalmente para acceder al aplicativo?</p>	<p>Principalmente utilizaremos teléfonos inteligentes y tabletas Android porque son fáciles de conseguir y de usar. Estos dispositivos son comunes entre nuestro personal, lo que facilita su adopción y uso en nuestras operaciones diarias. Pero también utilizamos iOS.</p>
<p>¿Qué precisión se requiere en la georreferenciación para medir el consumo de agua potable?</p>	<p>Se requiere una alta precisión en la georreferenciación, es decir, en metros. Esto es fundamental para obtener datos exactos sobre la ubicación y el consumo, lo que permite una gestión más eficiente y detallada del recurso hídrico.</p>

<p>¿Qué características de seguridad son imprescindibles para proteger los datos de consumo de agua potable?</p>	<p>Las características de seguridad imprescindibles son la encriptación de datos y la autenticación de usuarios. La encriptación asegura que la información esté protegida contra accesos no autorizados, mientras que la autenticación garantiza que solo los usuarios autorizados puedan acceder a los datos.</p>
<p>¿Cómo debería ser la interfaz de usuario para facilitar el acceso y la utilización del aplicativo?</p>	<p>La interfaz de usuario debería ser intuitiva y fácil de navegar. Esto facilitará que los usuarios accedan al aplicativo y lo utilicen de manera eficiente, minimizando la curva de aprendizaje y mejorando la experiencia general.</p>

3.6.2. Análisis general de la entrevista

El gerente de EMAPAT indica que actualmente no emplean ningún software o aplicación móvil para la gestión del consumo de agua. Destaca la necesidad de desarrollar un aplicativo móvil que facilite el monitoreo en tiempo real, permitiendo una mejor administración del recurso hídrico y mayor precisión en la recopilación de datos. Este aplicativo debería incluir funciones esenciales, como el registro en tiempo real del consumo y la georreferenciación de las mediciones. La aplicación deberá ser compatible con teléfonos inteligentes y tabletas, tanto en sistemas operativos Android como iOS, ofreciendo alta precisión en la georreferenciación y asegurando los datos mediante encriptación y autenticación de usuarios. Además, la interfaz debe ser intuitiva y fácil de manejar. El objetivo del proyecto es optimizar la eficiencia y el control en la gestión del agua.

Para desarrollar esta propuesta, se adoptará la metodología ágil Mobile-D, ideal para proyectos de aplicaciones móviles debido a su enfoque en entregas frecuentes de prototipos funcionales. Esto permitirá que el equipo de desarrollo se adapte rápidamente a cualquier cambio o nueva necesidad que surja durante el proceso. En cuanto a las herramientas tecnológicas, se utilizó el framework Flutter, que facilita la creación de una única aplicación compatible con Android y iOS. El código se escribió en Dart, un lenguaje de programación que asegura un rendimiento eficiente cuando se usa con Flutter. Asimismo, se integró Firebase como sistema de almacenamiento en tiempo real, permitiendo la sincronización de los datos de consumo y la protección de la información mediante encriptación. Para la georreferenciación, se empleará la API de Google Maps, lo cual garantizó una ubicación precisa de los medidores. Estas tecnologías combinadas permitieron construir una aplicación sólida, eficaz y fácil de usar, optimizando la gestión del agua potable.

CAPÍTULO IV

4. PROPUESTA

4.1. Título de la Propuesta

PROPUESTA DE APLICATIVO MÓVIL PARA MEDICIÓN DEL CONSUMO DE AGUA POTABLE UTILIZANDO GEORREFERENCIACIÓN PARA EMAPAT

4.2. Presentación

El presente capítulo describe la propuesta de desarrollo de un aplicativo móvil para la EMAPAT. La finalidad del aplicativo es mejorar la precisión y eficiencia en la medición y gestión del consumo de agua potable mediante el uso de tecnología de georreferenciación. Este proyecto responde a la necesidad identificada de modernizar los métodos tradicionales de medición, que son ineficientes y propensos a errores, y busca optimizar el uso del recurso hídrico en el cantón El Tambo, Cañar, Ecuador.

4.3. Justificación

La gestión eficiente del agua potable es crucial para la sostenibilidad ambiental y la planificación urbana. EMAPAT enfrenta desafíos significativos debido a la utilización de métodos tradicionales de medición del consumo de agua, que no proporcionan datos en tiempo real y son propensos a errores. La implementación de un aplicativo móvil que utilice georreferenciación permitirá:

- Monitoreo en tiempo real del consumo de agua.
- Reducción de errores en la recolección de datos.
- Mejora en la gestión y optimización del uso del agua.
- Facilitación en la toma de decisiones basadas en datos precisos.

- Aumento en la satisfacción del usuario al proporcionar información accesible y precisa sobre su consumo.

4.4. Descripción de la propuesta

En el presente trabajo, se propone el desarrollo de un aplicativo móvil para la medición del consumo de agua potable en la EMAPAT, utilizando georreferenciación. El aplicativo se desarrollará con la tecnología Flutter y siguiendo la metodología de desarrollo en Mobile-D. Su objetivo es medir el consumo de agua potable y facilitar el trabajo de los empleados de EMAPAT, mejorando la eficiencia en la gestión del recurso hídrico.

4.5. Desarrollo del proyecto

La estructura del desarrollo del aplicativo móvil para la medición del consumo de agua potable se basa en la metodología en MOBILE-D, un enfoque sistemático y secuencial para el diseño de sistemas móviles. Este proceso se desglosa en una serie de fases ordenadas y bien definidas, que guían cada etapa del proyecto desde el inicio hasta la finalización.

Las fases son las siguientes:

- I. Fase de exploración
- II. Fase de inicialización.
- III. Fase de producción.
- IV. Fase de estabilización
- V. Fase de pruebas.

4.5.1. Fase I: Exploración

Para el desarrollo de la app móvil destinada a la medición del consumo de agua potable, es esencial definir claramente los requerimientos necesarios. Estos requerimientos se especificarán a continuación:

4.5.1.1. Requerimientos funcionales.

Tabla 9. Requisitos de funcionalidad. Fuente: Autoría Propia

ID	Requerimiento	Descripción
RF001	Login	El gerente de EMAPAT debe poder iniciar sesión en el aplicativo móvil utilizando sus credenciales específicas. Esta función debe incluir autenticación segura y una interfaz para ingresar el nombre de usuario y la contraseña.
RF002	Cargar el mapa de Google	La aplicación debe cargar y mostrar un mapa de Google en la pantalla principal, utilizando la API de Google Maps para proporcionar una visualización precisa de la ubicación del usuario y otros puntos relevantes.
RF003	Maps en el aplicativo	La app debe integrar la funcionalidad de Google Maps dentro del aplicativo, permitiendo a los usuarios interactuar con el mapa para ver rutas, ubicaciones y otra información relevante en tiempo real.
RF004	Registro de incidentes	Los usuarios deben poder registrar incidentes relacionados con el consumo de agua o problemas técnicos a través de un formulario dentro de la app. El registro debe incluir detalles del incidente.

4.5.1.2. Requerimientos no funcionales

Tabla 10. Requisitos no funcionales. Fuente: Autoría propia

ID	Requerimiento	Descripción
----	---------------	-------------

RNF1	Interfaz de usuario	La aplicación debe tener una interfaz de usuario intuitiva y fácil de usar, con un diseño que facilite la navegación y la interacción de los usuarios con todas las funcionalidades del aplicativo.
RNF2	Framework	La aplicación debe ser desarrollada utilizando el framework Flutter, que permite crear aplicaciones móviles de alta calidad con una sola base de código para múltiples plataformas.
RNF3	Lenguaje de programación(dart)	El desarrollo de la aplicación debe realizarse en el lenguaje de programación Dart, que es compatible con el framework Flutter y facilita el desarrollo eficiente y mantenible.
RNF3	Base de datos	La app debe utilizar Firebase como sistema de gestión de base de datos, proporcionando almacenamiento en tiempo real, autenticación y otras funcionalidades de backend necesarias para el funcionamiento de la app.
RNF4	Idioma	La aplicación debe ofrecer soporte para múltiples idiomas, permitiendo a los usuarios seleccionar el idioma de su preferencia y garantizando que todos los textos estén correctamente traducidos.

RNF5	Disponibilidad	La aplicación debe estar disponible para los sistemas operativos móviles más comunes (iOS y Android) y debe funcionar de manera confiable en las versiones más recientes de estos sistemas.
-------------	----------------	---

4.5.1.3. Establecimiento del proyecto

La aplicación móvil estará diseñada para que el gerente de EMAPAT pueda monitorear el consumo de agua potable. El gerente podrá acceder a información detallada sobre el consumo en tiempo real y recibir alertas importantes, como notificaciones sobre un uso excesivo o problemas en el suministro. Además, la aplicación utilizará Google Maps para mostrar la ubicación exacta de los medidores de agua y las áreas de servicio, permitiendo al gerente ver las ubicaciones mediante coordenadas X, Y. Esta herramienta también permitirá el reporte y seguimiento de incidencias relacionadas con el servicio de agua, facilitando una gestión más eficiente y rápida de los problemas.

Tabla 11. Plan de iteraciones. Fuente: Autoría Propia

N°	ITERACION	ACTIVIDADES	SEMANA	CULMINACIÓN
1	Autenticación de usuario y configuración inicial	<ul style="list-style-type: none"> • Implementación del sistema de login. • Configuración de Firebase. • Validación de credenciales. 	1	El aplicativo permite al gerente iniciar sesión de manera segura

2	Visualización del consumo de agua	<ul style="list-style-type: none"> • Conexión a la base de datos para obtener datos de incidentes. • Implementación de gráficos y tablas para visualizar los incidentes reportados. 	2	El gerente puede ver el consumo de los incidentes en gráficos y tablas detalladas.
3	Integración de Google Maps	<ul style="list-style-type: none"> • Configuración de la API de Google Maps • Implementación de la funcionalidad de mapas 	1	El mapa de Google muestra las ubicaciones de los medidores de agua.
4	Registro de incidencias	<ul style="list-style-type: none"> • Desarrollo del formulario para reportar incidencias • Almacenamiento de datos de incidencias 	2	El gerente puede registrar y gestionar incidencias relacionadas con el servicio.
5	Notificaciones y alertas	<ul style="list-style-type: none"> • Implementación del sistema de notificaciones • Configuración de alertas para eventos importantes 	2	Se completan las pruebas finales y se corrigen todos los errores identificados.

6	Pruebas finales y ajustes	<ul style="list-style-type: none"> • Pruebas de usuario • Corrección de errores • Ajustes de la interfaz de usuario 	2	Se completan las pruebas finales y se corrigen todos los errores identificados.
7	Despliegue y capacitación	<ul style="list-style-type: none"> • Capacitación para el uso de la aplicación 	1	el gerente ha sido capacitado en su uso.

Tabla 12. Características del usuario. Fuente: Autoría Propia.

Usuario	Gerente de la EMAPAT
Habilidades	Conocimiento básico en aplicaciones móviles y gestión de recursos hídricos
Tareas	Monitorear el consumo de agua, visualizar datos en tiempo real, recibir notificaciones sobre consumo excesivo, y gestionar incidencias relacionadas con el servicio de agua

4.5.2. Fase II: Inicialización

En esta etapa, se efectúan las configuraciones finales y se verifican los diversos componentes del proyecto. Esta etapa es esencial para asegurar que todos los elementos del sistema estén completamente funcionales antes de su lanzamiento.

4.5.2.1. Esquema de solución

A continuación, se muestra la estructura general de la solución del producto destaca la capacidad de identificar problemas, proporcionando un marco para detectar y abordar posibles desafíos.

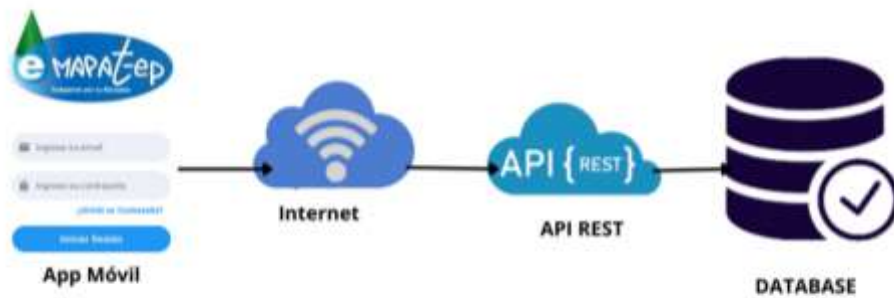


Ilustración 8..Arquitectura general de la infraestructura del producto. Fuente: Autoría Propia

4.5.2.2. Configuraciones para el proyecto

Recurso de Hardware

Tabla 13. Recuso de Hardware. Fuente; Autoría Propia.

Recurso	Descripción
Portatil	Computadora portátil utilizada para el desarrollo y prueba del aplicativo móvil, así como para la gestión del sistema.
Celular	Sistema Android, Sistema IOS

Recurso de Software

- **Entorno de Desarrollo (IDE):** Se eligió Visual Studio Code como IDE principal debido a su compatibilidad con Flutter y su robusto conjunto de herramientas para el desarrollo móvil.
- **Framework y Lenguaje:** Se utilizó Flutter junto con el lenguaje Dart, permitiendo la creación de una aplicación multiplataforma con una única base de código.
- **Base de Datos:** Se empleó Firebase para el almacenamiento de datos en tiempo real y la gestión de la autenticación de usuarios.

4.5.3. Fase III: Producción

En esta fase, se desarrollan los módulos esenciales del aplicativo móvil para la medición del consumo de agua potable, cumpliendo con los requerimientos funcionales establecidos en la fase de planificación. Cada módulo es diseñado e implementado para asegurar que el sistema funcione de manera integral y eficiente, proporcionando una experiencia de usuario completa y satisfactoria

Tabla 14. Módulos de la APP Móvil. Fuente: Autoría Propia

Módulo	ID	Requerimiento	Descripción
Módulo de Autenticación	M001	RF001	• Acceso y Autenticación
Módulo de Georreferenciación y Mapas	M002	RF002, RF003	• Carga y Visualización de Mapas • Interacción con el Mapa
Módulo de Registro de Incidencias	M003	RF004	• Reporte de Incidencias

- Gestión de Incidencias
-

4.5.3.1. Diagrama de caso de uso

Esta etapa incluye la definición de la usabilidad de la aplicación, detallando cómo el empleado interactuará con el sistema. Asimismo, se determinan las tareas específicas y se asignan responsabilidades a la persona encargada de llevarlas a cabo, garantizando una implementación clara y eficiente.

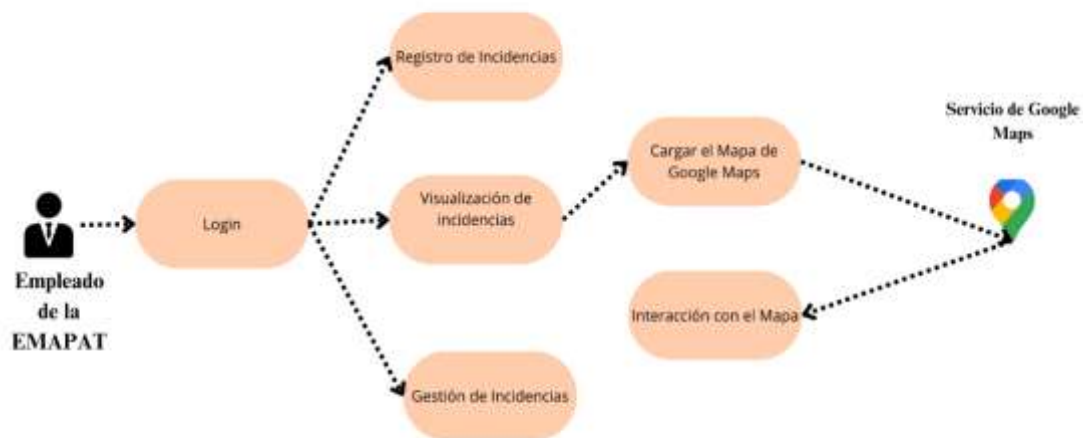


Ilustración 9. Diagrama de Caso de uso del empleado de la EMAPAT. Fuente: Autoría Propia.

4.5.3.2. Base de datos No relacional

El diagrama de la base de datos fue creado utilizando Firebase, la cual proporciona una solución de base de datos no relacional en tiempo real. El diagrama define la estructura de los datos, las relaciones entre las diferentes colecciones y documentos almacenados, así como los mecanismos de autenticación y control de acceso. Esto permite gestionar eficientemente la información relacionada con los usuarios, el consumo de agua, los incidentes reportados y cualquier otra funcionalidad relevante de la aplicación.

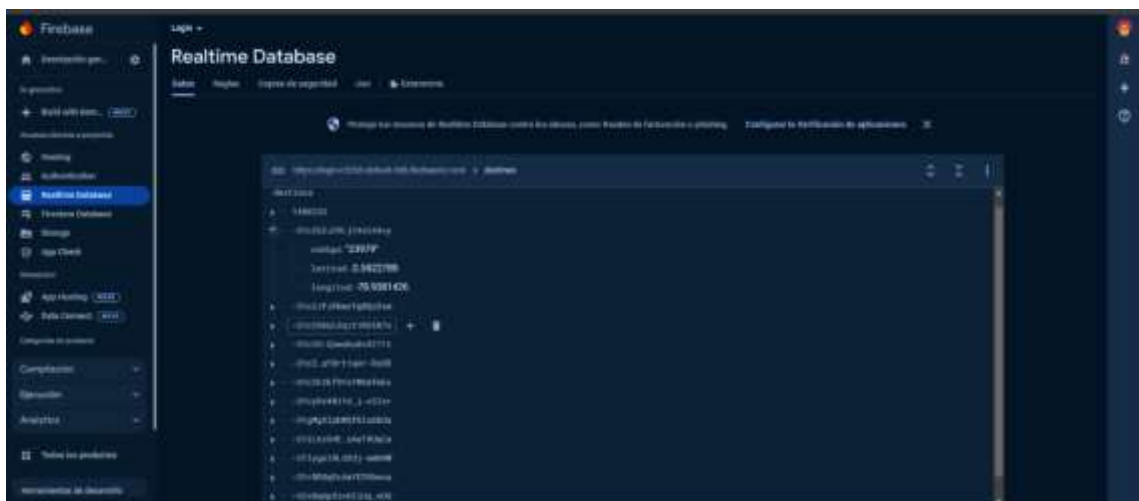


Ilustración 10. Base de Datos No relacional. Fuente: Autoría Propia

4.5.3.3. Interfaz del aplicativo móvil

1. Usuario

A continuación, la interfaz de inicio de sesión presenta el logotipo de la EMAPAT, donde el usuario puede ingresar utilizando su correo electrónico y contraseña.



Ilustración 11. Interfaz del Login. Fuente: Autoría Propia.

2. Interfaz del menú

Se encuentran las opciones a las cuales puede acceder para crear la ubicación, las mediciones y ver la ubicación

BIENVENIDO

EMAPAT-EP DASHBOARD

Opciones:



Ilustración 12. Interfaz del Menú. Fuente: Autoría propia.

3. Registro de la ubicación de la incidencia

En la imagen se muestra la interfaz utilizada para crear la ubicación del medidor.

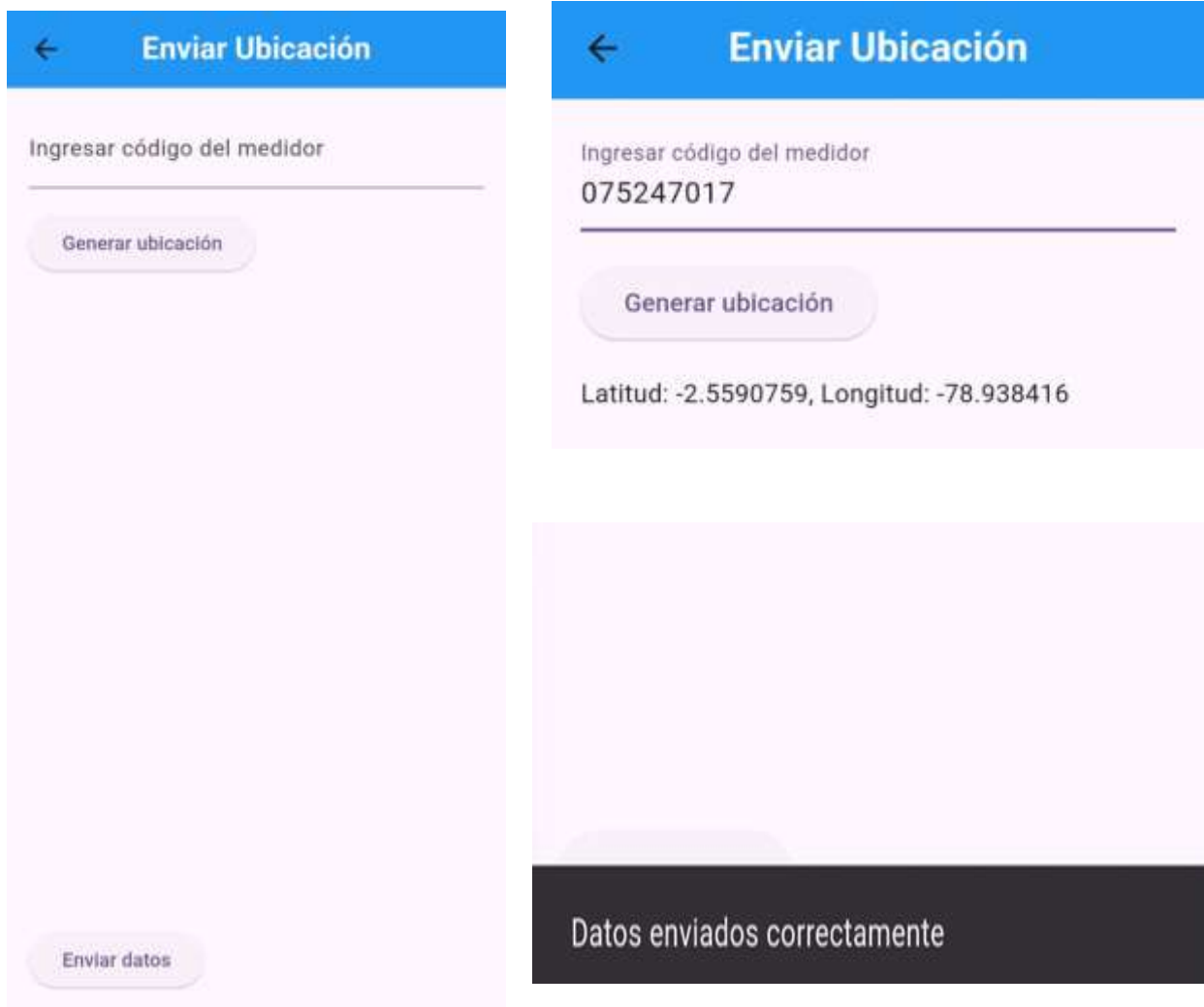


Ilustración 13. Interfaz para generar la ubicación del medidor. Fuente: Autoría Propia

4. Registro del consumo de agua

Interfaz en la cual se registra el consumo del agua del medidor 5657.

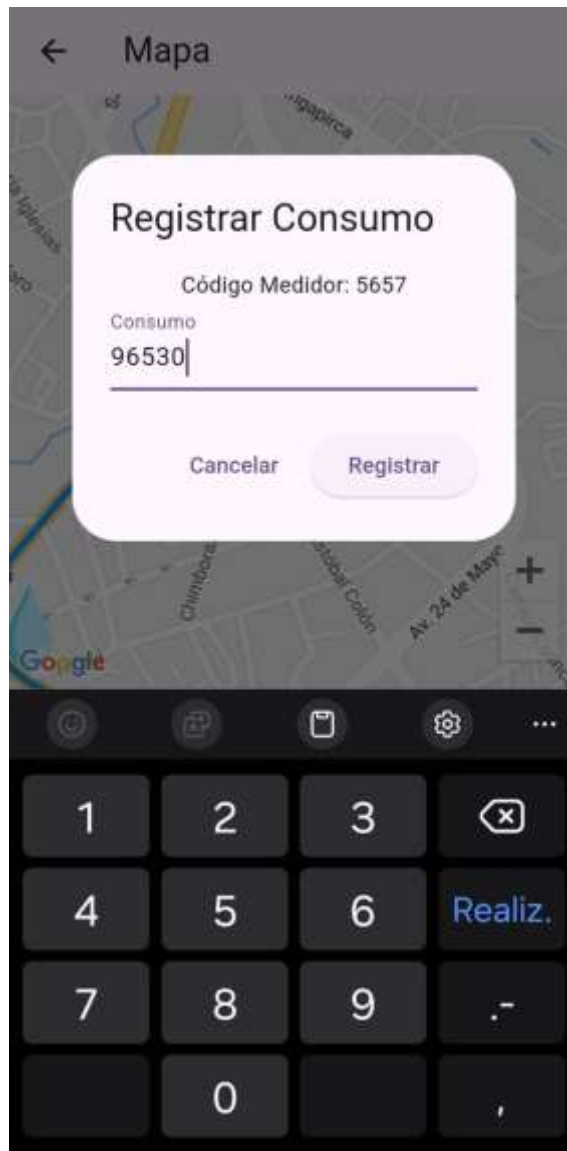


Ilustración 14. Registro del consumo de agua. Fuente: Autoría propia.

5. Visualización de la ubicación de las incidencias

Se muestra la ubicación de las incidencias registradas en Google Maps, permitiendo al gerente visualizarlas.

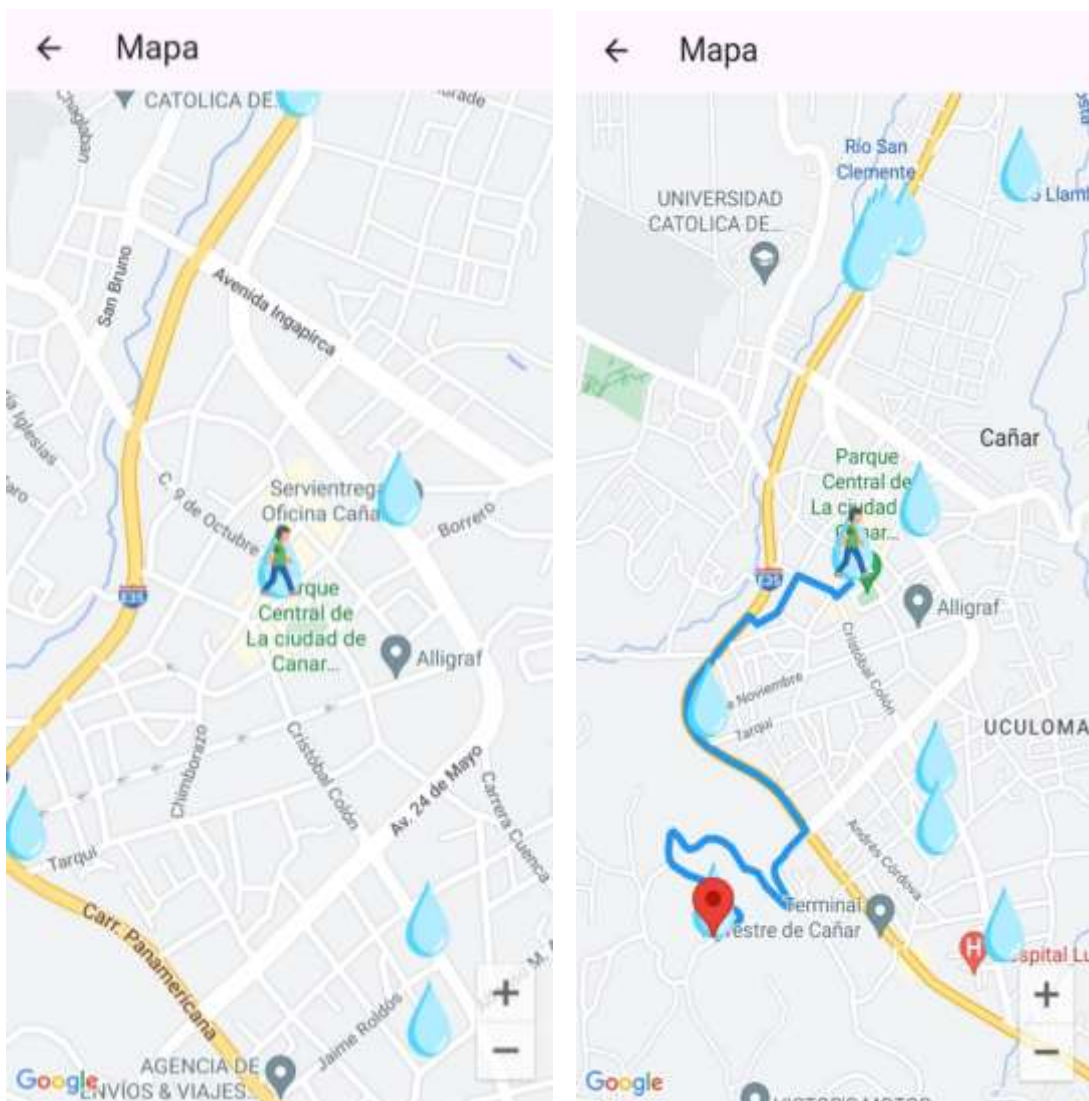


Ilustración 15. Ubicación de incidencias. Fuente: Autoría Propia

6. Gestión de las incidencias

En la siguiente imagen se muestra la interfaz en la cual se visualiza los destinos de los medidores.



Ilustración 16. Visualización de las incidencias. Fuente: autoría propia.

4.5.4. Fase IV: estabilización

En la fase de estabilización, se realizaron pruebas detalladas para asegurar que el prototipo del aplicativo móvil cumpliera con los requisitos iniciales definidos. Estas pruebas se fundamentaron en los requisitos obtenidos durante las fases anteriores de análisis y desarrollo, y los resultados indicaron que se habían satisfecho de forma efectiva.

4.5.4.1.Fase de pruebas

En esta fase se ejecutan las pruebas, asegurando que el sistema funcione correctamente y permitiendo controlar los requisitos.

I. Login

Prueba de aceptación

Tabla 15. Prueba de aceptación del login. Fuente: Autoría Propia.

Requerimiento	Proceso	Cumplido	Finalizado
RF001	El usuario debe ingresar su correo y contraseña para iniciar sesión.	Aprueba satisfactoriamente	✓

II. Mapa

Prueba de aceptación

Tabla 16. Prueba del Mapa. Autoría Propia.

Requerimiento	Proceso	Cumplido	Finalizado
RF002	Cargar y mostrar el mapa para que el usuario pueda visualizar y ubicar el registro del consumo de agua.	Aprueba satisfactoriamente	✓
RF003	Google Maps permite interactuar con el mapa para ver las ubicaciones del medidor a tiempo real.	Aprueba satisfactoriamente	✓

III. Registro de las incidencias

Prueba de aceptación

Tabla 17. Prueba de Registro de las incidencias. Fuente: Autoría Propia.

Requerimiento	Proceso	Cumplido	Finalizado
RF003	El usuario debe de ingresar los datos: la ubicación de donde se encuentra para registrar el medidor	Aprueba satisfactoriamente	✓

Conclusiones

El presente trabajo ha demostrado que la implementación de un aplicativo móvil basado en georreferenciación es una solución eficaz para mejorar la precisión en la medición del consumo de agua potable en EMAPAT. La aplicación permite localizar con exactitud los medidores de agua y recopilar datos en tiempo real, mejorando la gestión operativa y contribuyendo significativamente a la sostenibilidad del recurso hídrico.

Se desarrolló un marco teórico que respalda el uso de la tecnología de georreferenciación como una herramienta clave para optimizar la gestión del agua potable. A partir de este marco, se identificaron y analizaron los requerimientos funcionales y no funcionales necesarios para el diseño del aplicativo móvil, asegurando que cumpla con los estándares de seguridad de datos y ofrezca una interfaz de usuario eficiente.

El desarrollo del prototipo utilizando el framework Flutter fue exitoso, ya que se comprobó que es una plataforma idónea para garantizar la compatibilidad multiplataforma, lo que permite su implementación en dispositivos Android e iOS. Esto facilita el acceso al aplicativo tanto para los técnicos de EMAPAT como para los usuarios finales, mejorando la eficiencia en la detección de fugas y consumos anómalos.

Los principales desafíos identificados incluyen la resistencia al cambio por parte del personal y las limitaciones tecnológicas de la infraestructura actual. Sin embargo, estos obstáculos pueden ser mitigados mediante estrategias adecuadas de capacitación y mejoras en la infraestructura.

Este proyecto sienta las bases para futuras investigaciones y desarrollos tecnológicos en la gestión de recursos hídricos, y plantea la posibilidad de expandir el uso de georreferenciación a otros sectores que requieran una recolección precisa de datos espaciales.

Recomendaciones

- Se recomienda proceder con la implementación del aplicativo móvil desarrollado, ya que proporciona una solución eficaz para medir y gestionar el consumo de agua potable, aumentando la precisión de los datos y mejorando la capacidad de respuesta ante situaciones como fugas o consumos excesivos.
- Realizar talleres de capacitación para el personal administrativo que utilizará el aplicativo. Estos talleres deben incluir la instrucción sobre cómo interpretar los datos generados por la aplicación, así como la operación básica del sistema, para garantizar una transición sin problemas al nuevo método de trabajo.
- Desarrollar un plan para el mantenimiento regular y la actualización del aplicativo, asegurando que la tecnología permanezca funcional y relevante a lo largo del tiempo. Se recomienda establecer un contrato con una empresa especializada para manejar estas tareas.
- Utilizar los datos obtenidos a través del aplicativo para promover la transparencia en la facturación del servicio de agua potable. Esto fortalecerá la confianza de los usuarios en la gestión de EMAPAT y fomentará un uso más consciente y responsable del agua por parte de la comunidad.

Bibliografía

Rochina Chisag , A. G., Guashpa Pasto , E. W., & Coloma Garofalo, J. A. (2022). Gestión de información de lecturas del consumo de agua potable mediante una aplicación móvil. *Innovación y Software*, 6-25.

Aguirre Barrera, J., & Aguirre Barrera, S. (01 de 01 de 2020). *repository.unicatolica.edu.co*.
Obtenido de repository.unicatolica.edu.co:
[https://repository.unicatolica.edu.co/bitstream/handle/20.500.12237/2037/ART%
8dCULO_METODOLOG%c3%8dAS_PARA_DESARROLLO_PROYECTOS.pdf?s
equence=1&isAllowed=y](https://repository.unicatolica.edu.co/bitstream/handle/20.500.12237/2037/ART%c3%8dCULO_METODOLOG%c3%8dAS_PARA_DESARROLLO_PROYECTOS.pdf?sequence=1&isAllowed=y)

AGUIRRE, E. R., FERRER, M., BUSTOS, B., & MÉNDEZ, R. (2020). UX Design: a Methodology for the design of efficient digital projects focused on users. *Espacios*, 41, 9.

Altamirano Guerra, M. A. (01 de 01 de 2021). *repositorio.uta.edu.ec*. Obtenido de repositorio.uta.edu.ec:
<https://repositorio.uta.edu.ec/bitstream/123456789/33831/1/t1889si.pdf>

Andrade Guerrero, G. A., & Lozada Torres , L. P. (01 de 03 de 2020). *repositorio.utc.edu.ec*.
Obtenido de repositorio.utc.edu.ec:
[http://repositorio.utc.edu.ec/bitstream/27000/8469/1/UTC-PIM-
000428.pdf#page=66.43](http://repositorio.utc.edu.ec/bitstream/27000/8469/1/UTC-PIM-000428.pdf#page=66.43)

Android Developers. (06 de 05 de 2024). *developer.android.com*. Obtenido de developer.android.com: <https://developer.android.com/studio/intro?hl=es-419>

applesfera. (24 de 01 de 2024). *www.applesfera.com*. Obtenido de *www.applesfera.com*:

<https://www.applesfera.com/nuevo/xcode-que-sirve-que-funciones-para-desarrolladores-incluye-este-ide-apple>

Araque Arellano, M. (2022). *Diseño hidráulico de plantas de tratamiento de agua potable*. Editorial Abya-Yala.

arcgis. (28 de 09 de 2023). *pro.arcgis.com*. Obtenido de *pro.arcgis.com*:

<https://pro.arcgis.com/es/pro-app/latest/help/mapping/device-location/gnss-and-location-devices.htm#:~:text=GNSS%3A%20sistema%20global%20de%20navegaci%C3%B3n,se%20utiliza%20habitualmente%20en%20Norteam%C3%A9rica.>

Arcgis. (28 de 09 de 2023). *resources.arcgis.com*. Obtenido de *resources.arcgis.com*:

<https://resources.arcgis.com/es/help/getting-started/articles/026n00000014000000.htm>

AWS. (27 de 05 de 2024). *aws.amazon.com*. Obtenido de *aws.amazon.com*:

<https://aws.amazon.com/es/what-is/interoperability/#:~:text=La%20interoperabilidad%20se%20refiere%20a,compartan%20informaci%C3%B3n%20en%20tiempo%20real.>

Azizjon o'g'li, N. A. (2024). The SQL server language and its structure. *American Journal of Open University Education*, 12-15.

Azure. (08 de 09 de 2023). *azure.microsoft.com*. Obtenido de *azure.microsoft.com*:

<https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-java-programming-language#:~:text=Java%20es%20un%20lenguaje%20de,empresarial%20y%20muchos%20programas%20conocidos.>

Balseca Quisaguano, J. W. (01 de 07 de 2021). *repositorio.espe.edu.ec*. Obtenido de [repositorio.espe.edu.ec: https://repositorio.espe.edu.ec/bitstream/21000/25208/1/T-ESPE-044590.pdf](https://repositorio.espe.edu.ec/bitstream/21000/25208/1/T-ESPE-044590.pdf)

BARRERA CHUNGATA, B. J., & ILLAISACA TENECOTA, P. X. (08 de 03 de 2022). *dspace.ups.edu.ec*. Obtenido de dspace.ups.edu.ec: DESARROLLO DE UNA APLICACIÓN MÓVIL Y WEB PARA LA INSPECCIÓN DE JUNTAS DE AGUA POTABLE USANDO SERVICIOS DE GEOLOCALIZACIÓN Y ALMACENAMIENTO EN LA NUBE

Calderón Mena, D. C., & Moreno Cabrera, G. C. (01 de 03 de 2020). *bibdigital.epn.edu.ec/bitstream*. Obtenido de [bibdigital.epn.edu.ec/bitstream: https://bibdigital.epn.edu.ec/bitstream/15000/20808/1/CD%2010330.pdf](https://bibdigital.epn.edu.ec/bitstream/15000/20808/1/CD%2010330.pdf)

carto. (2024). *carto.com*. Obtenido de [carto.com: https://carto.com/blog/getting-started-with-cartodb-native-mobile-sdk](https://carto.com/blog/getting-started-with-cartodb-native-mobile-sdk)

Castillon Sedano, V. C. (01 de 01 de 2020). *repositorio.continental.edu.pe*. Obtenido de repositorio.continental.edu.pe: file:///C:/Users/usuario/Downloads/IV_FIN_109_TE_Castillon_Sedano_2020.pdf

Cercado Basurto , D. A., & Jaramillo Chonillo , F. D. (01 de 01 de 2021). *dspace.utc.edu.ec*. Obtenido de dspace.utc.edu.ec: <https://dspace.utc.edu.ec/bitstream/27000/9094/2/UTC-PIM-000544.pdf>

Certideal. (26 de 07 de 2023). *www.certideal.es*. Obtenido de www.certideal.es: <https://www.certideal.es/content/208-ios>

Chapman, A. D., & Wieczorek, J. R. (30 de 12 de 2020). *repository.oceanbestpractices.org*. Obtenido de repository.oceanbestpractices.org:

<https://repository.oceanbestpractices.org/bitstream/handle/11329/1486/georeferencing-best-practices.en.pdf?sequence=1&isAllowed=y>

cleancommit. (15 de 05 de 2022). *cleancommit.io*. Obtenido de cleancommit.io:

<https://cleancommit.io/blog/cross-platform-app-development-developing-an-application-for-maximum-exposure/>

Cunalata Paredes, D. Y. (01 de 01 de 2020). *repositorio.uta.edu.ec*. Obtenido de repositorio.uta.edu.ec:

<https://repositorio.uta.edu.ec/bitstream/123456789/31643/1/t1736ec.pdf#page=14.19>

Dart. (19 de 05 de 2024). *dart.dev*. Obtenido de dart.dev: <https://dart.dev/>

developer. (06 de 05 de 2024). *developer.android.com*. Obtenido de developer.android.com:

<https://developer.android.com/studio?hl=es-419>

developer. (29 de 05 de 2024). *developer.apple.com*. Obtenido de developer.apple.com:

<https://developer.apple.com/xcode/>

edutic. (36 de 10 de 2023). *edutic.up.edu.pe*. Obtenido de edutic.up.edu.pe:

<https://edutic.up.edu.pe/catalogo-software/ios/>

Escobar Bonilla, G. S., & Campaña Guzmán, A. I. (01 de 01 de 2022).

repositorio.espe.edu.ec. Obtenido de repositorio.espe.edu.ec:

<https://repositorio.espe.edu.ec/bitstream/21000/8188/1/AC-ESPEL-SOF-0002.pdf#page=1.29>

Espinoza, J. L., Yacelga, A. R., & Michilena, W. G. (2022). Las aplicaciones móviles y su impacto en la sociedad. *Revista Universidad y Sociedad*, 237-243.

Firebase. (01 de 01 de 2024). *firebase.google.com*. Obtenido de firebase.google.com:

<https://firebase.google.com/products/realtime-database?hl=es-419>

García Bonacho, D. (06 de 2023). *www.esic.edu*. Obtenido de *www.esic.edu*:

<https://www.esic.edu/rethink/tecnologia/lenguaje-xml-que-es-para-que-sirve-c>

GARCÍA JÉMEZ, C. (24 de 06 de 2021). *oa.upm.es*. Obtenido de *oa.upm.es*:

https://oa.upm.es/70603/1/TFG_CRISTINA_GARCIA_JEMEZ.pdf#page=8.08

Gargantilla, P. (30 de 12 de 2020). Como funciona el GPS. *ABC*.

Gaytán-Lugo, M., Farías-Mendoza, N., Chávez-Valdez, R., & Cervantes Zambrano, F.

(2020). Diseño e Implementación de un Geoportal Catastral para Visualización de Cartografía e Integración de Servicios Geoespaciales. *Revista Internacional de Investigación e Innovación*, 8(45), 20-39.

Gaytán-Lugo, M., Farías-Mendoza, N., Chávez-Valdez, R., & Cervantes-Zambrano, F.

(2020). Diseño e Implementación de un Geoportal Catastral para Visualización de Cartografía e Integración de Servicios Geoespaciales. *Revista internacional de investigación e innovación tecnológica*, 8(45), 20-39.

Gazaba, F. J. (17 de 03 de 2020). *pergamino.ar*. Obtenido de *pergamino.ar*:

<https://pergamino.ar/descargas/introduccion-a-los-sig-con-qgis.pdf>

GÓMEZ CADENA, P. A. (01 de 01 de 2023). *repositorio.utn.edu.ec*. Obtenido de

repositorio.utn.edu.ec:

<https://repositorio.utn.edu.ec/bitstream/123456789/13840/3/04%20MEC%20464%20TESIS%20GRADO.pdf#page=19.14>

Google Maps Platform. (06 de 05 de 2024). *developers.google.com*. Obtenido de

developers.google.com:

<https://developers.google.com/maps/documentation/javascript/overview?hl=es-419#:~:text=La%20API%20de%20Maps%20JavaScript%20te%20permite%20person>

alizar%20los%20mapas,p%C3%A1ginas%20web%20y%20dispositivos%20m%C3%B3viles.

Hadida, S., & Troilo, F. (01 de 01 de 2020). *www.econstor.eu*. Obtenido de *www.econstor.eu*:
<https://www.econstor.eu/bitstream/10419/238381/1/756.pdf#page=6.64>

hartman, j. (16 de 03 de 2024). *www.guru99.com*. Obtenido de *www.guru99.com*:
<https://www.guru99.com/es/java-vs-c-sharp-key-difference.html#:~:text=Diferencia%20clave%20entre%20Java%20y%20C%23&text=Java%20es%20un%20lenguaje%20orientado,de%20tor%20para%20m%C3%BAtiples%20operatores>.

HatchKuri, G., & Ribeiro, W. C. (2020). Gestión del agua y relaciones de poder en América Latina. *Agua y Territorio*, 11-12.

Hussein Toman, S. (24 de 11 de 2020). *jqcsm.qu.edu.iq*. Obtenido de *jqcsm.qu.edu.iq*:
<https://jqcsm.qu.edu.iq/index.php/journalcm/article/view/715/533>

ibm. (29 de 02 de 2024). *www.ibm.com*. Obtenido de *www.ibm.com*:
<https://www.ibm.com/docs/es/baw/23.x?topic=formats-javascript-object-notation-json-format>

IBM. (01 de 05 de 2024). *www.ibm.com*. Obtenido de *www.ibm.com*:
<https://www.ibm.com/mx-es/topics/postgresql>

ionicframework. (28 de 04 de 2024). *ionicframework.com*. Obtenido de *ionicframework.com*:
<https://ionicframework.com/docs>

Lara Cerdón, S. (01 de 07 de 2020). *oa.upm.es*. Obtenido de *oa.upm.es*:
https://oa.upm.es/67025/1/TFM_SONIA_LARA_CORDON.pdf

Lara, J., Samper, M., & Colomé, D. (2023). Predicción a corto plazo de sistemas de medición inteligentes mediante arquitecturas de aprendizaje profundo multivariable y multipaso. *CENACE*, 153-164.

leafletjs. (01 de 01 de 2024). *leafletjs.com*. Obtenido de leafletjs.com:
<https://leafletjs.com/reference.html>

learn microsoft. (20 de 01 de 2024). *learn.microsoft.com*. Obtenido de learn.microsoft.com:
<https://learn.microsoft.com/es-es/windows/dev-environment/javascript/nodejs-overview>

Lizárraga Osuna, K. A. (9 de 12 de 2020). *repositorio.upsin.edu.mx*. Obtenido de repositorio.upsin.edu.mx:
<http://repositorio.upsin.edu.mx/Fragmentos/tesinas/A031LIZARRAGAOSUNAKEVINANTONIO8101.pdf#page=20.08>

MACIAS VERA, E. V. (01 de 11 de 2021). *dspace.utb.edu.ec*. Obtenido de dspace.utb.edu.ec: <http://dspace.utb.edu.ec/bitstream/handle/49000/10516/E-UTB-FAFI-SIST-000242.pdf?sequence=1&isAllowed=y>

MADRIGAL CHAVES, W. (01 de 01 de 2020). *repositorio.usam.ac.cr*. Obtenido de repositorio.usam.ac.cr:
<https://repositorio.usam.ac.cr/xmlui/bitstream/handle/11506/2075/LEC%20ING%20SIST%200039%202020.pdf?sequence=1&isAllowed=y#page=3.08>

mapbox. (6 de 11 de 2023). *docs.mapbox.com*. Obtenido de docs.mapbox.com:
<https://docs.mapbox.com/help/getting-started/mobile-apps/>

Medium. (01 de 09 de 2022). *medium.com*. Obtenido de *medium.com*:

<https://medium.com/@nerdas.estudio/principios-de-dise%C3%B1o-ux-ui-18f04ec0e9ae>

Mejía Bárcen, G. R., Cruz Mejía, O., & Meléndez Ruiz, E. I. (2024). Nivel de interoperabilidad en el sistema de información de la universidad pública en México. *Revista de Ciencias Sociales (Ve)*, 28, 56-73.

microsoft. (13 de 07 de 2023). *learn.microsoft.com*. Obtenido de *learn.microsoft.com*:

<https://learn.microsoft.com/es-es/previous-versions/xamarin/get-started/what-is-xamarin>

Molina Ríos, J. R., Honores Tapia, J. A., Nieves, P. S., & Pardo León, H. P. (2021). ESTADO DEL ARTE: METODOLOGÍAS DE DESARROLLO DE APLICACIONES MÓVILES. *3C Tecnología*, 17-15.

Morales, A. (14 de 06 de 2022). *mappinggis.com*. Obtenido de *mappinggis.com*:

<https://mappinggis.com/2015/03/las-mejores-apis-javascript-para-webmapping/>

Morejón Labrada, S. (2020). Principios del proceso de Diseño de Interfaz de Usuario. *revista cubana de transformación digital*, 1(3), 143-155.

Morocho Pichizaca, L. M. (01 de 01 de 2022). *dspace.ucacue.edu.ec*. Obtenido de

dspace.ucacue.edu.ec:

<https://dspace.ucacue.edu.ec/server/api/core/bitstreams/16669c18-3478-4047-b47d-f435dd11cc5b/content#page=19.08>

MOSQUERA CORONEL, K. J. (01 de 01 de 2022). *dspace.utb.edu.ec*. Obtenido de

dspace.utb.edu.ec: <http://dspace.utb.edu.ec/bitstream/handle/49000/13056/E-UTB-FAFI-SIST-000391.pdf?sequence=1&isAllowed=y#page=6.26>

MySQL. (13 de 03 de 2024). *www.mysql.com*. Obtenido de *www.mysql.com*:

<https://www.mysql.com/>

Nadal, L. G. (18 de 01 de 2024). *rua.ua.es*. Obtenido de *rua.ua.es*:

[https://rua.ua.es/dspace/bitstream/10045/141082/1/EL_DERECHO_FUNDAMENTA
L_AL_AGUA_ANALISIS_COMPARADO_ENTRE_Gonzalez_Nadal_Laura.pdf](https://rua.ua.es/dspace/bitstream/10045/141082/1/EL_DERECHO_FUNDAMENTA
L_AL_AGUA_ANALISIS_COMPARADO_ENTRE_Gonzalez_Nadal_Laura.pdf)

NARANJO HUERTAS , J. (2022). *dspace.uniandes.edu.ec*. Obtenido de

dspace.uniandes.edu.ec:

[https://dspace.uniandes.edu.ec/bitstream/123456789/14714/1/UT-SIS-PDI-005-
2022.pdf](https://dspace.uniandes.edu.ec/bitstream/123456789/14714/1/UT-SIS-PDI-005-
2022.pdf)

Oliva Gamarra, J. J. (01 de 01 de 2024). *repositorio.ucv.edu.pe*. Obtenido de

repositorio.ucv.edu.pe:

[https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/133413/Oliva_GJJ-
SD.pdf?sequence=1&isAllowed=y](https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/133413/Oliva_GJJ-
SD.pdf?sequence=1&isAllowed=y)

openlayers. (01 de 01 de 2024). *openlayers.org*. Obtenido de *openlayers.org*:

<https://openlayers.org/>

Oracle Corporation. (13 de 03 de 2024). *dev.mysql.com*. Obtenido de *dev.mysql.com*:

<https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>

Quisaguano Collaguazo , L. R., Pallasco Venegas, M. S., Andaluz Guerrero, A. A., Martines

Freire, M. N., & Corrales Beltrán, S. H. (20 de 08 de 2022). Desarrollo híbrido con
flutter. *Ciencia Latina Revista Científica Multidisciplinar*, 4594-4609.

Rea Chela , J. A. (01 de 01 de 2024). *dspace.unach.edu.ec*. Obtenido de *dspace.unach.edu.ec*:

<https://dspace.unach.edu.ec/bitstream/51000/12758/1/Rea%20Ch.%20Johan%20A.%>

282024%29_determinaci%C3%B3n%20del%20comportamiento%20de%20consumo.
pdf

Redhat. (18 de 04 de 2024). *www.redhat.com*. Obtenido de *www.redhat.com*:

<https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>

Renfro, B. A., Stein, M., Reed, E. B., Morales, J., & Villalba, E. J. (2020). *An Analysis of Global Positioning System (GPS) Standard Positioning Service Performance for 2019*. La Universidad de Texas en Austin.

RODRÍGUEZ YAGUAL, H. I. (01 de 01 de 2020). *repositorio.upse.edu.ec*. Obtenido de *repositorio.upse.edu.ec*:

<https://repositorio.upse.edu.ec/bitstream/46000/5406/1/UPSE-TIP-2020-0010.pdf#page=1.00>

ROJAS BARRAZA, C. J. (01 de 01 de 2022). *cybertesis.unmsm.edu.pe/*. Obtenido de *cybertesis.unmsm.edu.pe/*:

https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/18818/Rojas_bc.pdf?sequence=1&isAllowed=y#page=18.10

Rojas, E. M. (2020). Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo. *Revista Ibérica de Sistemas e Tecnologías de Informação*, 586-599.

Santos González, S. (01 de 06 de 2023). *oa.upm.es*. Obtenido de *oa.upm.es*:

https://oa.upm.es/75453/1/TFG_SERGIO_SANTOS_GONZALEZ.pdf

Shevtsiv, N. A., & Striuk, A. M. (27 de 11 de 2020). <http://ds.knu.edu.ua/>. Obtenido de <http://ds.knu.edu.ua/>:

<http://ds.knu.edu.ua/jspui/bitstream/123456789/3550/1/Shevtsiv%20N.%20A.%20CROSS%20platform%20development.pdf>

Tamayo Lozada, J., & Ruiz Miyares, L. (2021). El Diccionario básico escolar en móviles y tabletas con sistema operativo Android. *Serie Científica de la Universidad de las Ciencias Informáticas*, 54-66. Obtenido de dialnet.unirioja.es:
<file:///C:/Users/usuario/Downloads/Dialnet-ElDiccionarioBasicoEscolarEnMovilesYTabletasConSis-8590654.pdf>

Tapia Naranjo, B. D. (01 de 01 de 2023). *repositorio.uta.edu.ec*. Obtenido de *repositorio.uta.edu.ec*:
<file:///C:/Users/usuario/Downloads/Tesis%20I.C.%201713%20-%20Tapia%20Naranjo%20Bryan%20David.pdf>

Tapiero, D. I., & Valencia, M. M. (2021). SIG aplicado a la optimización del tiempo de diseño en redes de distribución de agua potable. *Ingeniería Hidráulica y ambiental*, 68-80.

Torres Bastidas, J. (2021). *repositorio.uta.edu.ec*. Obtenido de *repositorio.uta.edu.ec*:
<https://repositorio.uta.edu.ec/bitstream/123456789/32046/1/t1754si.pdf>

Vaibhav , Y., Pankaj, C., & Gaikwad, D. (2021). FIREBASE: DESCRIPCIÓN GENERAL Y USO. *Revista internacional de investigación sobre modernización en tecnología y ciencia de ingeniería*, 1178-1183.

Valencia Astudillo, W. G., & Delgado Tapia, C. E. (2021). *dspace.ups.edu.ec*. Obtenido de *dspace.ups.edu.ec*: <https://dspace.ups.edu.ec/bitstream/123456789/21427/5/UPS-GT003532.pdf>

Zheng, W. (01 de 06 de 2022). *oa.upm.es*. Obtenido de *oa.upm.es*:

https://oa.upm.es/71341/1/TFG_WEI_ZHENG.pdf

Anexos

1.1. MANUAL DE USUARIO

Propuesta de aplicativo móvil
para medición del consumo de
agua potable, utilizando
georeferenciación para
EMAPAT

Sebastián G.

Propuesta de aplicativo móvil para medición del consumo de agua potable, utilizando georeferenciación para EMAPAT

Manejo del Prototipo de Aplicativo Móvil EMAPAT de El Tambo

En caso de que el prototipo se implemente en el futuro, se deberá subir a la tienda de aplicaciones Google Play Store para que el usuario de la EMAPAT pueda descargarlo e instalarlo en los dispositivos móviles.

Interfaz inicial

Una vez que la aplicación esté instalada en el dispositivo móvil, se podrá reconocer por el siguiente icono en el menú de aplicaciones.



Ilustración 17. Ícono de la App. Fuente: Autoría Propia

Al iniciar el aplicativo, aparece una pantalla de carga de 2 segundos que muestra la imagen de la EMAPAT.



Ilustración 18. Interfaz de la carga de la app. Fuente: Autoría Propia

Pantalla de Inicio de Sesión

Después de la pantalla de carga, la aplicación muestra una opción para iniciar sesión utilizando una cuenta de Gmail. Al seleccionar esta opción, los usuarios pueden autenticar su identidad de manera segura y rápida, lo que les permite acceder a todos los servicios y funcionalidades de la aplicación sin restricciones. Esta integración con Gmail facilita una experiencia de usuario fluida y eficiente, eliminando la necesidad de crear y recordar nuevas credenciales de acceso.



Ilustración 19. Interfaz de Usuario. Fuente: Autoría Propia

En caso de que el usuario olvide su contraseña, puede recuperarla fácilmente haciendo clic en el enlace "Olvidó su contraseña". Este enlace dirige al usuario a un proceso de recuperación de contraseña, asegurando que puedan restablecer su acceso a la aplicación de manera sencilla y segura.

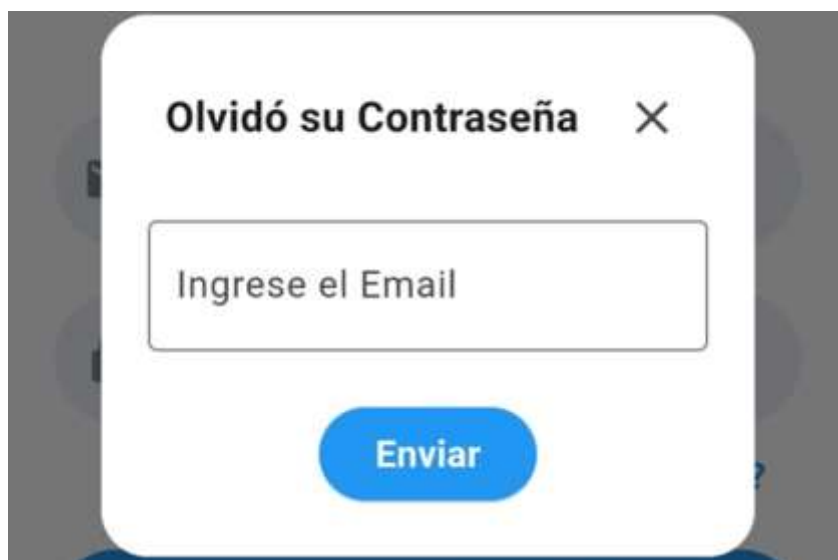


Ilustración 20. Interfaz de Olvidó Su Contraseña. Fuente: Autoría Propia.

Pantalla Menú

Después de iniciar sesión con la cuenta de Google, la aplicación ofrece el siguiente menú en el cual se encuentran los siguientes servicios: CREAR UBICACIÓN, MEDICIONES, VER UBICACIÓN Y SALIR.

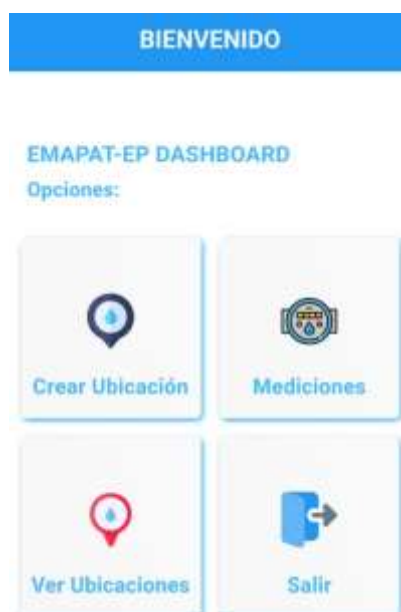


Ilustración 21. Menú de la app. Fuente: Autoría Propia.

Pantalla Crear Ubicación

Al ingresar a “Crear Ubicación” se muestra la siguiente pantalla:



Ilustración 22. Interfaz de Crear Ubicación. Fuente: Autoría Propia.

En la interfaz, se genera la ubicación. Si el GPS no está activado, aparecerá el siguiente mensaje:

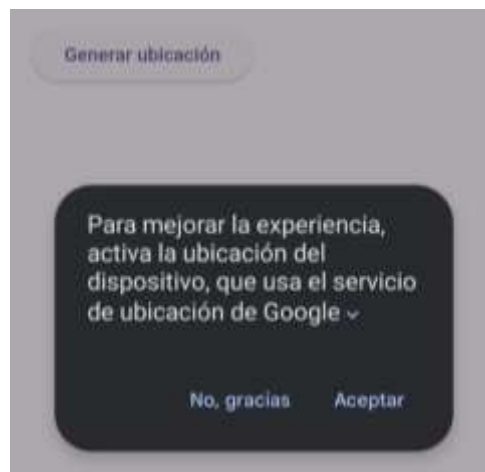


Ilustración 23. Solicitud de Activación de Ubicación. Fuente: Autoría Propia.

Una vez activado el GPS, se visualiza la ubicación:

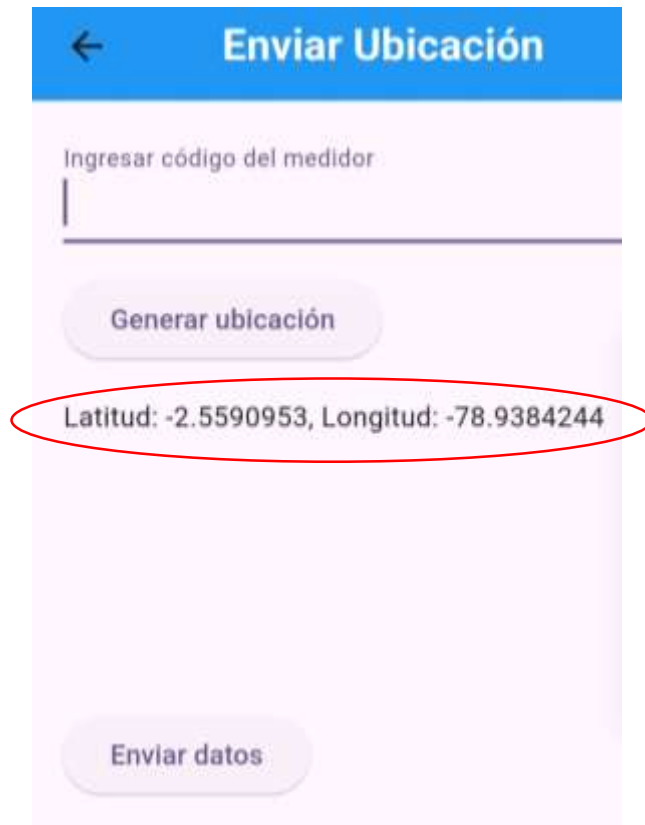


Ilustración 24. Ubicación. Fuente: Autoría Propia.

A continuación, se introduce el identificador del medidor para transmitir la ubicación

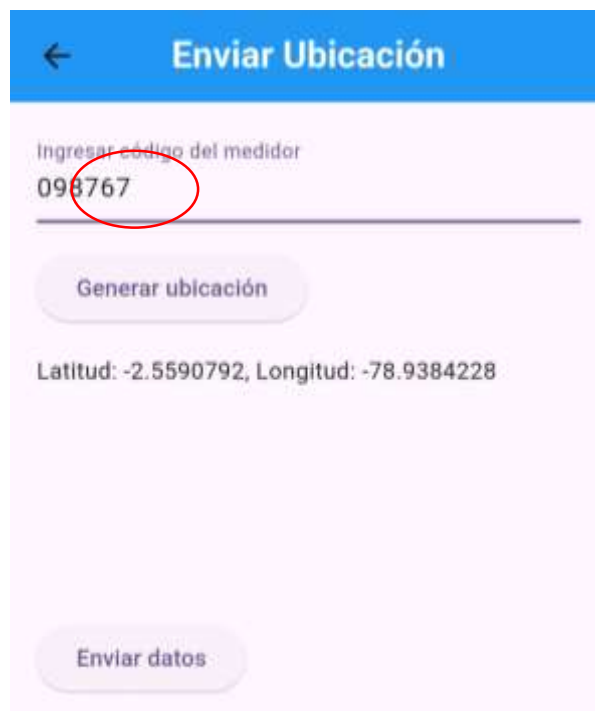


Ilustración 25. Identificador del Medidor. Fuente: Autoría Propia.

Al enviar los datos aparecerá una notificación.

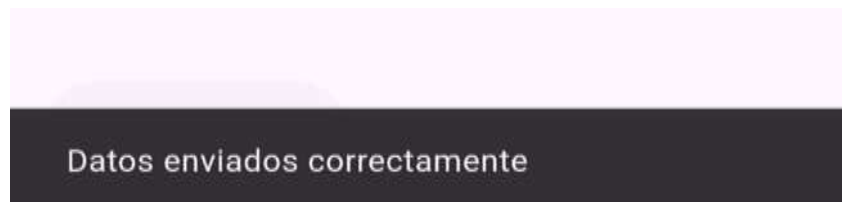


Ilustración 26. Notificación de datos. Fuente: Autoría Propia.

Pantalla de las mediciones

Regrese a la pantalla principal y accedemos a la sección de mediciones para verificar si el medidor ha sido registrado correctamente. En esta sección, podemos visualizar una lista de todos los medidores registrados hasta el momento. Buscamos el medidor específico que acabamos de registrar, asegurándonos de que toda la información asociada, como el número de serie, la ubicación y los datos de configuración, se haya guardado de manera correcta. Si encontramos algún error o inconsistencia, podemos editar los detalles del medidor o repetir el proceso de registro según sea necesario.



Ilustración 27. Menú Mediciones. Fuente: Autoría Propia.

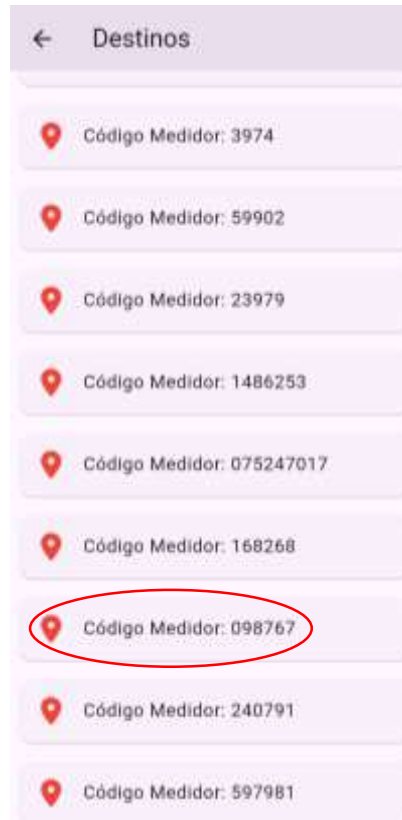


Ilustración 28. Gestión de las mediciones. Fuente: Autoría Propia.

Luego de visualizar, se hace clic en el código del medidor para acceder a su ubicación. Al hacer esto, se abrirá una nueva pantalla o sección que mostrará detalles más específicos sobre la ubicación del medidor, incluyendo un mapa o una descripción de la ubicación física. Esto permitirá confirmar que el medidor está correctamente asignado a la ubicación esperada y que todos los datos relacionados son precisos. Si encontramos algún error o inconsistencia, podemos editar los detalles del medidor o repetir el proceso de registro según sea necesario.

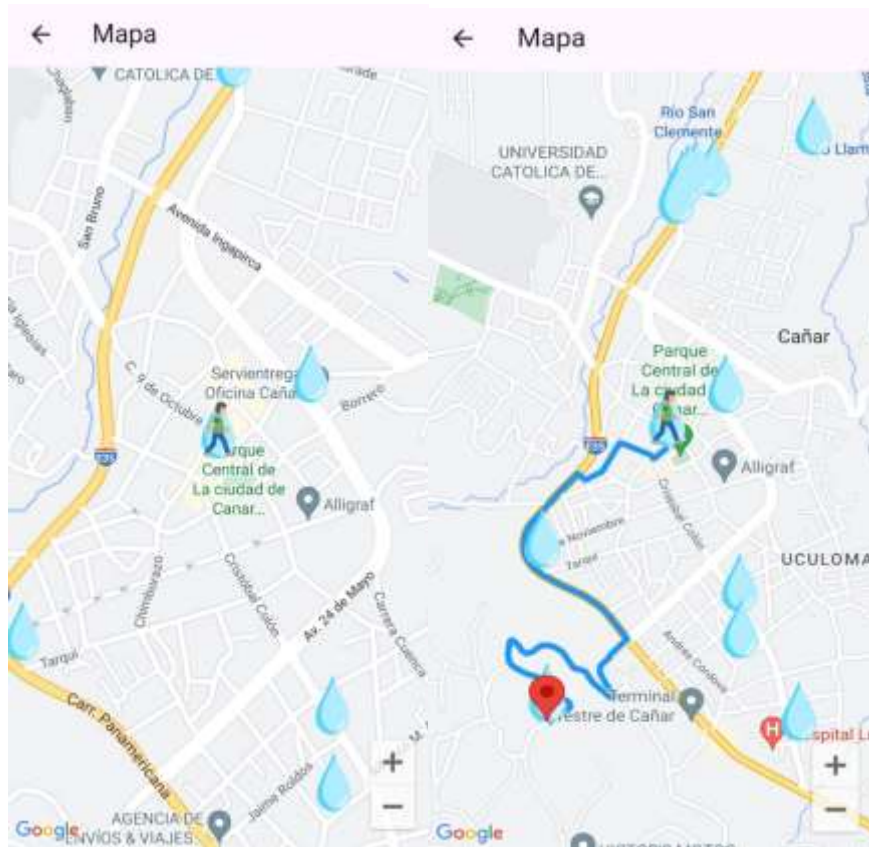


Ilustración 29. Visualización de las incidencias. Fuente: Autoría Propia.

Por último, se registra el consumo del agua del medidor

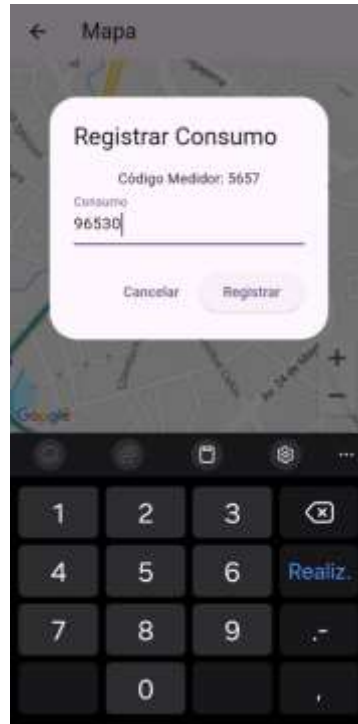


Ilustración 30. Registro del consumo del agua. Fuente: Autoría Propia.

MANUAL DEL PROGRAMADOR

Propuesta de aplicativo móvil
para medición del consumo de
agua potable, utilizando
georeferenciación para
EMAPAT

Sebastián G.

Manual del Programador

El manual del programador tiene como propósito explicar la estructura y el desarrollo del prototipo, detallando las herramientas utilizadas en su creación. Incluye información técnica redactada de manera que cualquier persona con el conocimiento necesario en esta área pueda comprender e interpretar el contenido descrito.

Diseño e Implementación de la Base de Datos

Diseño e Implementación de la Base de Datos de la EMAPAT

Para comenzar a utilizar Firebase, lo primero que debemos hacer es crear un nuevo proyecto en la plataforma. Este proceso es sencillo y se puede completar en unos pocos pasos. Primero, se accede a la consola de Firebase en el sitio web oficial y se autentica con nuestra cuenta de Google. Una vez dentro de la consola, se selecciona la opción "Crear proyecto" para iniciar.

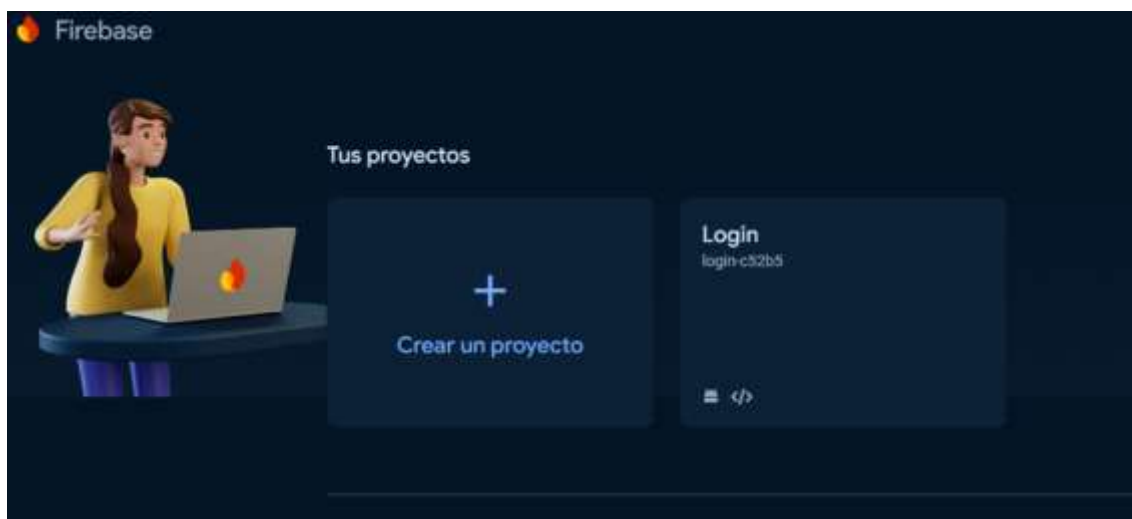


Ilustración 31. Creación de un nuevo proyecto en FIREBASE. Fuente: Autoría Propia.

A continuación, se selecciona la plataforma en este caso FLUTTER:

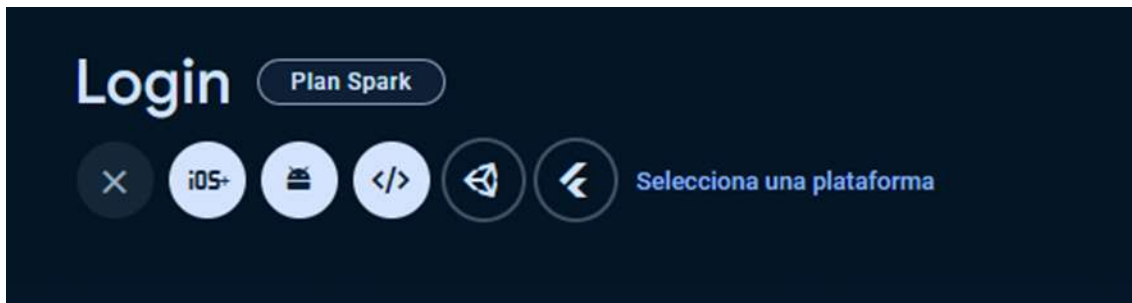


Ilustración 32. Selección de Plataforma. Fuente: Autoría Propia.

Para instalar la interfaz de línea de comandos (CLI) de Flutter, primero descarga el SDK de Flutter desde el sitio oficial. Luego, descomprime el archivo descargado y agrega la ruta flutter/bin a las variables de entorno de tu sistema. Una vez hecho esto, abre una terminal y ejecuta el comando flutter doctor para verificar que la instalación se haya realizado correctamente y resolver cualquier dependencia faltante.



Ilustración 33. Instalación de CLI en Flutter. Fuente: Autoría Propia.

El archivo google-services.json permite configurar la conexión a los servicios de Firebase en una aplicación Flutter. Este archivo contiene información esencial del proyecto, como el número de proyecto, la URL de Firebase, el ID del proyecto y el bucket de almacenamiento. Además, incluye detalles específicos del cliente, como el ID de la aplicación móvil y el nombre del paquete. Esta configuración asegura que la aplicación pueda interactuar correctamente con los servicios de Firebase, permitiendo funcionalidades como autenticación, bases de datos en tiempo real, análisis y más.

A continuación, este archivo debe colocarse en el directorio adecuado del proyecto, generalmente dentro de la carpeta android/app, para que las configuraciones sean reconocidas y aplicadas correctamente por la aplicación.

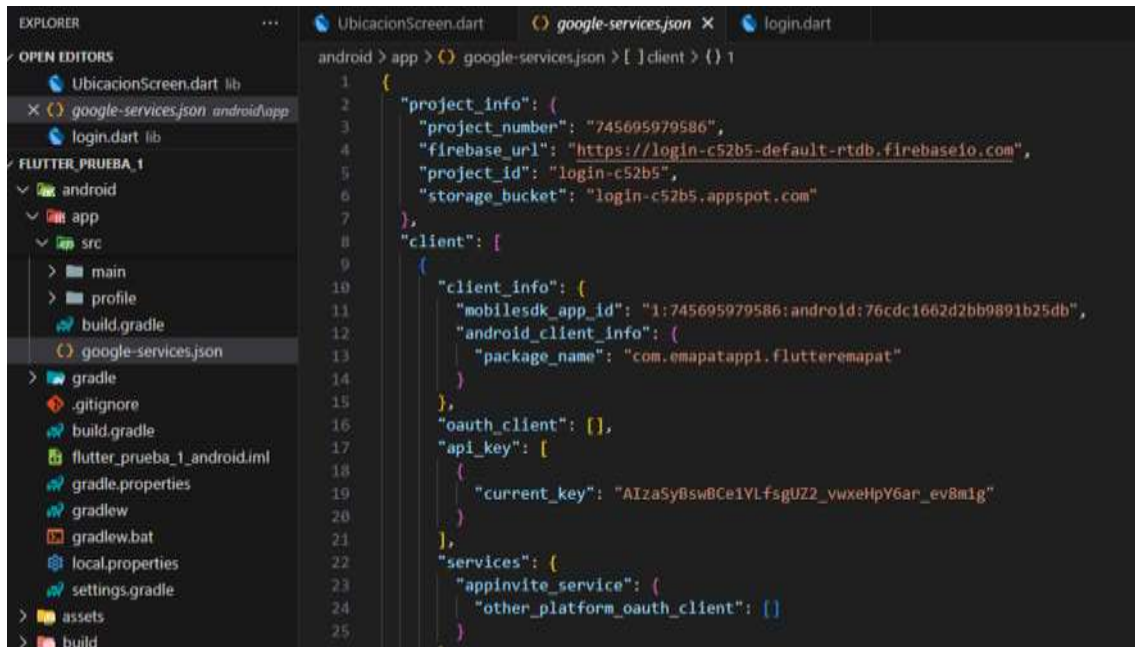


Ilustración 34. Google-services.json. Fuente: Autoría Propia.

Autenticación de Usuarios en Firebase

En la pestaña de "Authentication" del proyecto creado en Firebase, se puede gestionar a los usuarios que van a tener acceso a la aplicación. Esta sección permite configurar diversos métodos de autenticación, como correo electrónico y contraseña, proveedores de identidad federada (Google, Facebook, Twitter, etc.), y otros métodos personalizados. A través de esta interfaz, se pueden crear, editar y eliminar cuentas de usuario, así como gestionar las sesiones de los usuarios y sus permisos. Esto facilita el control de acceso y la seguridad de la aplicación, asegurando que solo los usuarios autorizados puedan acceder a sus funcionalidades y datos.

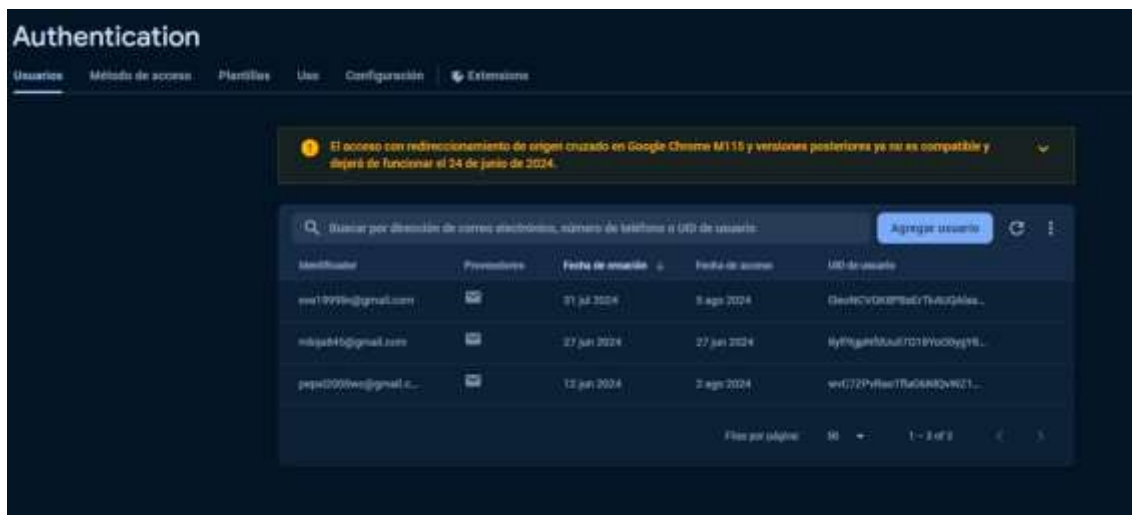


Ilustración 35. Autenticación de Usuarios en Firebase. Fuente: Autoría Propia.

Realtime Database en Firebase

En la pestaña de "Realtime Database" de Firebase, se pueden crear y gestionar bases de datos en tiempo real que permiten almacenar y sincronizar datos entre los usuarios de la aplicación. En este caso, se crean dos tablas llamadas "destinos" y "consumos". La tabla "destinos" puede almacenar información relacionada con diferentes destinos, como ubicaciones y detalles específicos. Por otro lado, la tabla "consumos" puede registrar datos sobre el consumo, como el uso de recursos o servicios. La Realtime Database asegura que cualquier cambio en los datos se refleje instantáneamente en todos los dispositivos conectados, proporcionando una experiencia de usuario fluida y coherente.



Ilustración 36. Realtime Database En FireBase. Fuente: Autoría Propia.

Destinos

En esta tabla se almacenan los datos de las ubicaciones de los medidores. Cada entrada en la tabla "destinos" incluye información detallada sobre la ubicación específica de cada medidor, como la dirección, coordenadas geográficas, y cualquier otra información relevante para identificar y localizar el medidor. Esto permite un seguimiento preciso y una gestión eficiente de los medidores distribuidos en diferentes ubicaciones.



Ilustración 37. Destinos. Fuente: Autoría Propia.

Consumos

En esta tabla se almacenan los datos de los consumos de cada medidor. Cada entrada en la tabla "consumos" incluye información detallada sobre el uso registrado por cada medidor, como la cantidad de recursos consumidos, las fechas y horas de las mediciones, y cualquier otra información relevante para el seguimiento y análisis del consumo. Esto permite monitorear y gestionar eficientemente los patrones de consumo de los diferentes medidores.

consumos			
▶	5634		
▶	5657		
▶	12658		
▶	48519		
▶	876912		
▶	897967		
▶	123633		
▶	628729		
▼	1358699	codigo: "1358699"	consumo: "486569"
		latitud: -2.559079	longitud: -78.9384175

Ilustración 38. Tabla Consumos. Fuente: Autoría Propia.

Implementación de la API de Google Maps

Android

Dentro del archivo AndroidManifest.xml, se agrega el código proporcionado por Flutter donde se coloca la KEY de Google Maps. Este archivo de manifiesto es esencial para la configuración de permisos y servicios utilizados por la aplicación Android. Al incluir la clave de API de Google Maps en este archivo, se habilita el acceso a los servicios de mapas dentro de la aplicación, permitiendo la integración de funcionalidades como la visualización de mapas y la ubicación geográfica.

```
<manifest ...
  <application ...
    <meta-data android:name="com.google.android.geo.API_KEY"
              android:value="YOUR KEY HERE"/>
```

Ilustración 39. AndroidManifest. Fuente: Autoría Propia.

En caso de que sea para un sistema operativo **iOS**, dentro del archivo AppDelegate.swift se agrega el código proporcionado por Flutter donde se coloca la KEY de Google Maps. Esto permite habilitar los servicios de mapas en la aplicación iOS, facilitando la integración de funcionalidades de Google Maps como la visualización de mapas y la ubicación geográfica. Se muestra a continuación en la ilustración 24

```
#include "AppDelegate.h"
#include "GeneratedPluginRegistrant.h"
#import "GoogleMaps/GoogleMaps.h"

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application
  didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
  [GMSServices provideAPIKey:@"YOUR KEY HERE"];
  [GeneratedPluginRegistrant registerWithRegistry:self];
  return [super application:application didFinishLaunchingWithOptions:launchOptions];
}
@end
```

Ilustración 40. iOS. Fuente: Autoría Propia.

Programación del Aplicativo Móvil EMAPAT-EP

El aplicativo móvil está desarrollado en el lenguaje de programación Dart para dispositivos móviles con sistemas operativos Android e iOS, utilizando el software de desarrollo Flutter.

Especificación de las Dependencias dentro del Proyecto

Las dependencias utilizadas para el desarrollo de este aplicativo se especifican dentro del archivo `pubspec.yaml`. Este archivo define todas las bibliotecas y paquetes necesarios para que la aplicación funcione correctamente, asegurando que todos los componentes externos sean gestionados y mantenidos de manera eficiente durante el desarrollo, como se muestra en la siguiente imagen.

```

dependencies:
  flutter:
    sdk: flutter
  flutter_native_splash: ^2.4.1
  lottie: ^3.1.4
  cupertino_icons: ^1.0.6
  firebase_core: ^3.1.1
  cloud_firestore: ^5.1.0
  firebase_auth: ^5.1.1
  google_maps_flutter: ^2.7.0
  firebase_database: ^11.0.3
  permission_handler: ^11.3.1
  location: ^5.0.0
  font_awesome_flutter: ^10.7.0
  flutter_polyline_points: ^2.1.0
  http: ^1.2.1
  dio: ^5.5.0

dev_dependencies:
  flutter_test:
    sdk: flutter

  flutter_launcher_icons: ^0.13.1

# The "flutter_lints" package below contains a set of recommended lints to
# encourage good coding practices. The lint set provided by the package is
# activated in the `analysis_options.yaml` file located at the root of your
# package. See that file for information about deactivating specific lint
# rules and activating additional ones.
  flutter_lints: ^3.0.0

```

Ilustración 41. Especificación de las dependencias dentro del Proyecto. Fuente: Autoría Propia.

Estructura del Proyecto y Programación

El proyecto del aplicativo se organiza en la carpeta principal “lib”, que contiene las diferentes pantallas (Screens) de la aplicación. Cada pantalla está separada en archivos individuales para mantener una estructura ordenada y facilitar el mantenimiento.

Carpeta Login

Dentro de “lib”, se ha creado una carpeta específica llamada “Login”, la cual incluye:

- **Services:** Subcarpeta que contiene los servicios para la autenticación de usuarios, como verificación de credenciales y gestión de sesiones.

- **Widget:** Subcarpeta con componentes visuales reutilizables para la pantalla de inicio de sesión, como botones y campos de entrada.

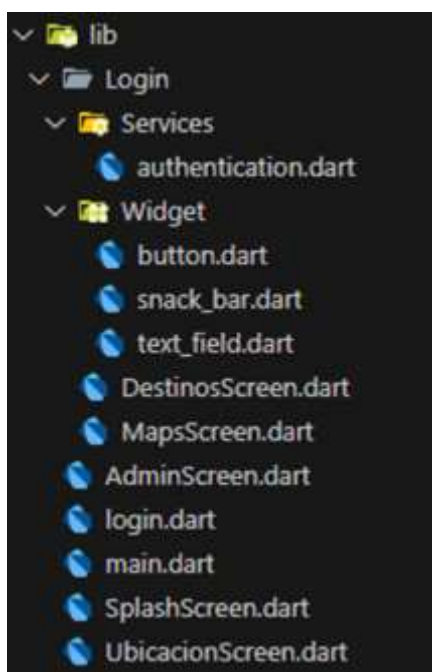


Ilustración 42. Carpeta LIB. Fuente: Autoría Propia.

Authentication.dart

El archivo **Authentication.dart** se encarga de la comunicación con el servicio de autenticación de usuarios de Firebase. Este archivo incluye las funciones necesarias para manejar el inicio de sesión, registro y gestión de sesiones de los usuarios, utilizando los servicios proporcionados por Firebase. Al centralizar la lógica de autenticación en este archivo, se facilita la integración y mantenimiento del sistema de autenticación en la aplicación, asegurando una gestión segura y eficiente de los usuarios.

```
import 'package:firebase_auth/firebase_auth.dart';

class AuthServices{
  final FirebaseAuth _auth = FirebaseAuth.instance;

  Future<String> loginUser({
    required String email,
    required String password,
  }) async{
    String res = "Ha ocurrido un error";
    try{
      if (email.isNotEmpty || password.isNotEmpty) {
        await _auth.signInWithEmailAndPassword(
          email: email,
          password: password,
        );
        res = "Éxito";
      }else{
        res = "Por favor llene todos los campos";
      }
    }catch (e) {
      return e.toString();
    }
    return res;
  }

  Future<void> singOut() async{
    await _auth.signOut();
  }
}
```

Ilustración 43. Autentification. Dart. Fuente: Autoría Propia.

Carpeta Widget

Esta carpeta incluye los archivos “button.dart”, “snack_bar.dart” y “text_field.dart” los cuales permiten dar un estilo a los widget que se utilizan dentro de la aplicación.

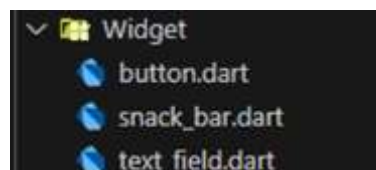


Ilustración 44. Ícono Widget. Fuente: Autoría Propia.

```

import 'package:flutter/material.dart';

class TextFieldInputs extends StatelessWidget {
  final TextEditingController textEditingController;
  final bool isPass;
  final String hintText;
  final IconData icon;

  const TextFieldInputs({
    super.key,
    required this.textEditingController,
    this.isPass = false,
    required this.hintText,
    required this.icon,
  });

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 10),
      child: TextField(
        obscureText: isPass,
        controller: textEditingController,
        decoration: InputDecoration(
          hintText: hintText,
          hintStyle: const TextStyle(
            color: Colors.black45,
            fontSize: 18,
          ), // TextStyle
          prefixIcon: Icon(
            icon,
            color: Colors.black45,
          ), // Icon
          contentPadding: const EdgeInsets.symmetric(vertical: 15, horizontal: 30),
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(30),
            borderSide: BorderSide.none,
            filled: true,
          ), // OutlineInputBorder
          fillColor: const Color(0xFFD9F8),
          enabledBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(30),
            borderSide: const BorderSide(
              width: 2,
              color: Colors.blue,
            ), // BorderSide
          ), // OutlineInputBorder
          focusedBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(30),
            borderSide: const BorderSide(
              width: 2,
              color: Colors.blue,
            ), // BorderSide
          ), // OutlineInputBorder
        ), // InputDecoration
      ), // TextField
    ); // Padding
  }
}

```

Ilustración 45. Archivo `Text_Field.Dat`. Fuente: Autoría Propia.

Main.dart

El archivo `Main.dart` es la pantalla principal de la aplicación, encargada de dirigir a los usuarios a una pantalla de bienvenida personalizada, conocida como `SplashScreen`. Esta pantalla de inicio se muestra al abrir la aplicación, proporcionando una introducción visual funcional antes de redirigir a las pantallas principales de la aplicación.

```

import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:flutter_prueba_1/AdminScreen.dart';
import 'package:flutter_prueba_1/SplashScreen.dart';

Run | Debug | Profile
void main() async{
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ), // ThemeData
      home: StreamBuilder(
        stream: FirebaseAuth.instance.authStateChanges(),
        builder: (context, snapshot) {
          if (snapshot.hasData) {
            return const AdminScreen();
          } else {
            return const SplashScreen();
          }
        }, // StreamBuilder
      ); // MaterialApp
    }
  }
}

```

Ilustración 46. Main Dart. Fuente: Autoría Propia.

SplashScreen.dart

El archivo SplashScreen.dart se encarga de mostrar una pantalla de bienvenida al usuario al iniciar la aplicación. Esta pantalla suele incluir el logotipo de la aplicación, animaciones o mensajes de bienvenida, y se muestra durante unos segundos antes de redirigir al usuario a la pantalla principal de la aplicación. La función principal del SplashScreen.dart es mejorar la experiencia del usuario proporcionando una transición visual agradable al abrir la aplicación.

```

import 'dart:async';
import 'package:flutter/material.dart';
import 'package:flutter_prueba_1/login.dart';
import 'package:lottie/lottie.dart';

class SplashScreen extends StatefulWidget {
  const SplashScreen({super.key});

  @override
  State<SplashScreen> createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
  @override
  void initState() {
    super.initState();
    Timer(Duration(seconds: 3), () => Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => Login()));)
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Lottie.asset('assets/animations/gota.json', fit: BoxFit.fill),
      ), // Center
    ); // Scaffold
  }
}

```

Ilustración 47. SplashScreen. Fuente: Autoría Propia.

Login.dart

El archivo Login.dart se encarga de gestionar la pantalla de inicio de sesión de la aplicación. En esta pantalla, los usuarios pueden ingresar sus credenciales para acceder a sus cuentas. Login.dart incluye los campos de entrada para el nombre de usuario y la contraseña, así como botones para enviar la información y realizar el proceso de autenticación. Este archivo interactúa con los servicios de autenticación definidos en Autenticacion.dart para verificar las credenciales del usuario y manejar la lógica de inicio de sesión. También puede incluir enlaces a funciones adicionales como la recuperación de contraseña y el registro de nuevos usuarios.

```

import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter_prueba_1/AdminScreen.dart';
import 'package:flutter_prueba_1/Login/Services/authentication.dart';
import 'package:flutter_prueba_1/Login/widget/button.dart';
import 'package:flutter_prueba_1/Login/widget/snack_bar.dart';
import 'package:flutter_prueba_1/Login/widget/text_field.dart';

class Login extends StatefulWidget {
  const Login({super.key});

  @override
  State<Login> createState() => _LoginState();
}

class _LoginState extends State<Login> {
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  bool isLoading = false;
  final auth = FirebaseAuth.instance;

  void dispose(){
    super.dispose();
    emailController.dispose();
    passwordController.dispose();
  }

  void loginUsers() async{
    String res = await AuthServices().loginUser(
      email: emailController.text,
      password: passwordController.text,
    );
    if (res == "Éxito"){
      setState(() {
        isLoading = true;
      });
      Navigator.of(context).pushReplacement(
        MaterialPageRoute(builder: (context) => const AdminScreen(),
        ), // MaterialPageRoute
      );
    }else{
      setState(() {
        isLoading = false;
      });
      showSnackBar(context, res);
    }
  }
}

```

Ilustración 48. Login Dart. Fuente: Autoría Propia.

AdminScreen.dart

El archivo AdminScreen.dart se encarga de gestionar la pantalla de administración de la aplicación. Esta pantalla está destinada a los usuarios con permisos de administrador y proporciona acceso a diversas funciones administrativas, como la gestión de usuarios, la configuración de parámetros del sistema, y la supervisión de actividades dentro de la aplicación. AdminScreen.dart incluye una interfaz de usuario con menú y opciones específicas para administrar diferentes aspectos de la aplicación, asegurando que los administradores puedan realizar sus tareas de manera eficiente y segura.

```

import 'package:flutter/material.dart';
import 'package:flutter_prueba_1/Login/DestinosScreen.dart';
import 'package:flutter_prueba_1/Login/MapsScreen.dart';
import 'package:flutter_prueba_1/Login/Services/authentication.dart';
import 'package:flutter_prueba_1/UbicacionScreen.dart';
import 'package:flutter_prueba_1/login.dart';

class AdminScreen extends StatefulWidget {
  const AdminScreen({super.key});

  @override
  State<AdminScreen> createState() => _AdminScreenState();
}

class _AdminScreenState extends State<AdminScreen> {
  Card makeDashboardItem(String title, String img, int index){
    return Card(
      elevation: 2,
      margin: const EdgeInsets.all(8),
      child: Container(
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(5),
          gradient: const LinearGradient(
            begin: FractionalOffset(0.0, 0.0),
            end: FractionalOffset(3.0, -1.0),
            colors: [
              color.fromARGB(255, 248, 248, 248),
              color(0xFFFFFFFF)
            ],
          ), // LinearGradient
          boxShadow: const [
            boxShadow(
              color: color.fromARGB(255, 68, 188, 255),
              blurRadius: 3,
              offset: Offset(2, 2),
            ) // BoxShadow
          ],
        ), // BoxDecoration
        child: InkWell(
          onTap: () async{
            if(index==0){
              //1.item
              Navigator.push(context, MaterialPageRoute(builder: ((context) => UbicacionesScreen())));
            }
            if(index==1){
              //2.item
              Navigator.push(context, MaterialPageRoute(builder: ((context) => DestinosScreen())));
            }
            if(index==2){

```

Ilustración 49. AdminScreen Dart. Fuente: Autoría Propia.

UbicacionesScreen.dart

El archivo UbicacionesScreen.dart gestiona la pantalla de ubicaciones en la aplicación, permitiendo al usuario obtener y enviar su ubicación actual a una base de datos en Firebase. La clase UbicacionesScreen, que extiende StatefulWidget, maneja la lógica a través de _UbicacionesScreenState y utiliza TextEditingController para la entrada de un código, además de variables para almacenar la latitud y longitud. La función _getCurrentLocation solicita permisos de ubicación, obtiene las coordenadas actuales y las actualiza, mientras que _sendData envía el código, la latitud y la longitud a Firebase Database, creando una nueva entrada en el nodo destinos.

```

import 'package:flutter/material.dart';
import 'package:firebase_database/firebase_database.dart';
import 'package:permission_handler/permission_handler.dart';
import 'package:location/location.dart';

class UbicacionesScreen extends StatefulWidget {
  const UbicacionesScreen({super.key});

  @override
  State<UbicacionesScreen> createState() => _UbicacionesScreenState();
}

class _UbicacionesScreenState extends State<UbicacionesScreen> {
  final TextEditingController _codigoController = TextEditingController();
  double? _latitude;
  double? _longitude;
  final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref().child('destinos');

  Location location = new Location();

  void _getCurrentLocation() async {
    var status = await Permission.location.status;
    if (!status.isGranted) {
      await Permission.location.request();
    }

    try {
      LocationData currentLocation = await location.getLocation();
      setState(() {
        _latitude = currentLocation.latitude;
        _longitude = currentLocation.longitude;
      });
    } catch (e) {
      print("Error: $e");
    }
  }

  Future<void> _sendData() async {
    String codigo = _codigoController.text;

    if (codigo.isNotEmpty && _latitude != null && _longitude != null) {
      String id = _databaseReference.push().key ?? '';
      await _databaseReference.child(id).set({
        'codigo': codigo,
        'latitud': _latitude,
        'longitud': _longitude,
      });
    }
  }
}

```

Ilustración 50. UbicacionesScreen Dart. Fuente: Autoría Propia.

DestinosScreen.dart

El archivo DestinosScreen.dart gestiona la pantalla de destinos en la aplicación, mostrando una lista de ubicaciones almacenadas en la base de datos Firebase. La clase DestinosScreen extiende StatefulWidget y utiliza _DestinosScreenState para manejar la lógica de la pantalla. En esta clase, se recuperan y muestran las ubicaciones almacenadas en Firebase, permitiendo al usuario visualizar las entradas del nodo destinos con detalles como el código, la latitud y la longitud. La pantalla también incluye funcionalidades para actualizar y eliminar registros, facilitando la gestión de las ubicaciones guardadas.

```

import 'package:flutter/material.dart';
import 'package:firebase_database/firebase_database.dart';
import 'package:flutter_prueba_1/Login/MapsScreen.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';

class DestinosScreen extends StatefulWidget {
  const DestinosScreen({super.key});

  @override
  State<DestinosScreen> createState() => _DestinosScreenState();
}

class _DestinosScreenState extends State<DestinosScreen> {
  final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref().child('destinos');
  List<Map<dynamic, dynamic>> _destinosList = [];

  @override
  void initState() {
    super.initState();
    _fetchDestinos();
  }

  void _fetchDestinos() {
    _databaseReference.onValue.listen((event) {
      final List<Map<dynamic, dynamic>> loadedDestinos = [];
      final data = event.snapshot.value as Map<dynamic, dynamic>;

      data.forEach((key, value) {
        loadedDestinos.add(value as Map<dynamic, dynamic>);
      });

      setState(() {
        _destinosList = loadedDestinos;
      });
    });
  }

  void _navigateToMap(double lat, double lng) {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => MapPage(destinationLat: lat, destinationLng: lng),
      ), // MaterialPageRoute
    );
  }
}

```

Ilustración 51. DestinosScreen Dart. Fuente: Autoría Propia.

MapsScreen.dart

El archivo MapsScreen.dart gestiona la pantalla de mapas en la aplicación, mostrando una interfaz de usuario que integra un mapa interactivo mediante el uso de la API de Google Maps. La clase MapsScreen extiende StatefulWidget y utiliza _MapsScreenState para manejar la lógica de la pantalla. En esta clase, se configura y muestra un mapa con una posición inicial predeterminada, permitiendo al usuario visualizar la ubicación actual y centrar el mapa en esta ubicación. La pantalla incluye funcionalidades para activar la visualización de la ubicación del usuario y opciones para mover el mapa a la posición deseada mediante un botón flotante. La pantalla facilita la interacción del usuario con el mapa, mejorando la experiencia de navegación y localización dentro de la aplicación.

```

import 'package:firebase_database/firebase_database.dart';
import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'package:permission_handler/permission_handler.dart';
import 'package:location/location.dart';
import 'package:flutter_polyline_points/flutter_polyline_points.dart';

class MapPage extends StatefulWidget {
  final double? destinationLat;
  final double? destinationLng;

  const MapPage({super.key, this.destinationLat, this.destinationLng});

  @override
  State<MapPage> createState() => _MapPageState();
}

class _MapPageState extends State<MapPage> {
  BitmapDescriptor? customIcon1, customIcon2;
  GoogleMapController? mapController;
  static const LatLng _pGooglePlex = LatLng(-2.510209809663967, -78.92806805944743);
  Set<Marker> markers = {};
  Location location = Location();
  List<LatLng> polylineCoordinates = [];
  late PolylinePoints polylinePoints;

  @override
  void initState() {
    super.initState();
    polylinePoints = PolylinePoints();
    _checkLocationPermission();
    _addDestinationMarker();
    _listenToLocationChanges();
    _loadMarkers();
    _loadCustomIcons();
  }

  Future<void> _loadCustomIcons() async {
    customIcon1 = await BitmapDescriptor.fromAssetImage(
      ImageConfiguration(size: Size(10, 10)),
      'assets/images/soltar.png',
    );
    customIcon2 = await BitmapDescriptor.fromAssetImage(
      ImageConfiguration(size: Size(10, 10)),
      'assets/images/caminar.png',
    );
  }
}

```

Ilustración 52. MapsScreen Dart. Fuente: Autoría Propia.

ANEXO 2 AUTORIZACION DE PUBLICACION EN EL REPOSITORIO INSTITUCIONAL



Universidad
Católica
de Cuenca

AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL

Brian Sebastian Garcia Castillo portador(a) de la cédula de ciudadanía N° 0302604673 En calidad de autor/a y titular de los derechos patrimoniales del trabajo de titulación **Propuesta de aplicativo móvil para medición del consumo de agua potable, utilizando georeferenciación para Emapat**, de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de este trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior. Cañar, **27 de noviembre de 2024**

F. 

Brian Sebastian Garcia Castillo

C.I. 0302604673

www.ucacue.edu.ec