UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:

INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:

Ingenieros de Sistemas

TEMA:

DESARROLLO DE UNA APLICACIÓN MÓVIL QUE PERMITA LA RECOPILACIÓN AUTOMÁTICA DE DATOS RELACIONADOS CON IMPERFECCIONES EN LA CALZADA PARA EL DISTRITO METROPOLITANO DE QUITO

AUTORES:

BRYAN FABIÁN GUEVARA GUAMÁN

JONATHAN RAFAEL JIMÉNEZ LEDESMA

TUTOR:

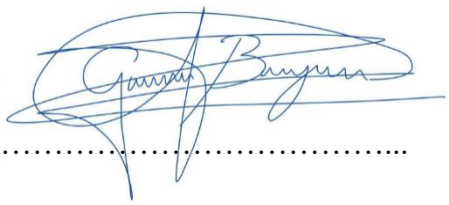
FRANKLIN EDMUNDO HURTADO LARREA

Quito, agosto del 2021

CESIÓN DE DERECHOS DE AUTOR

Nosotros Bryan Fabián Guevara Guamán, Jonathan Rafael Jiménez Ledesma, con documentos de identificación N° 1724510183 y Nº 2200363139, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: DESARROLLO DE UNA APLICACIÓN MÓVIL QUE PERMITA LA RECOPILACIÓN AUTOMÁTICA DE DATOS RELACIONADOS CON IMPERFECCIONES EN LA CALZADA PARA EL DISTRITO METROPOLITANO DE QUITO, mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



……...

Bryan Fabián Guevara Guamán
1724510183

……...

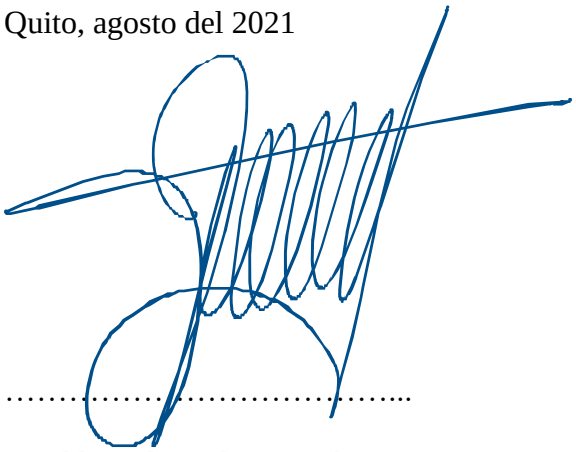
Jonathan Rafael Jiménez Ledesma
2200363139

Quito, agosto del 2021

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: DESARROLLO DE UNA APLICACIÓN MÓVIL QUE PERMITA LA RECOPILACIÓN AUTOMÁTICA DE DATOS RELACIONADOS CON IMPERFECCIONES EN LA CALZADA PARA EL DISTRITO METROPOLITANO DE QUITO, realizado por Bryan Fabián Guevara Guamán y Jonathan Rafael Jiménez Ledesma, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, agosto del 2021



……..

Franklin Edmundo Hurtado Larrea
CI: 1713382016

DEDICATORIA

Dedicado a mi madre, a mi tía y a mi padre, quienes me han brindado su apoyo incondicional para salir adelante y poder culminar mis estudios universitarios, a ellos les dedico este logro profesional.

Bryan Fabián Guevara Guamán

Este proyecto técnico lo dedico de manera muy especial a mi madre y a mis hermanas, quienes han sido los pilares principales en mi vida y durante la trayectoria de mi formación profesional, brindando su apoyo incondicional y a mi esposa que me ayudó en la etapa del desarrollo de la tesis.

Jonathan Rafael Jiménez Ledesma

AGRADECIMIENTO

Agradecemos de forma infinita a nuestros padres y familiares cercanos que depositaron toda su confianza para que finalicemos con éxito la carrera universitaria.

Queremos expresar nuestra eterna gratitud a la Universidad Politécnica Salesiana por permitirnos ser parte de su prestigiosa institución al abrir sus puertas para poder estudiar la carrera de Ingeniería de Sistemas.

Nuestro agradecimiento también a todos los docentes quienes estuvieron durante todo el trayecto de la carrera universitaria y en especial a nuestro tutor el Ing. Franklin Hurtado, por haber aceptado ser nuestro guía en el desarrollo de este proyecto técnico.

Y para culminar, también agradecemos a todos nuestros compañeros que nos apoyaron de forma incondicional y con carácter salesiano en toda la formación académica de todos los niveles en la Universidad ya que gracias al compañerismo y su amistad nada de esto sería posible para nuestra carrera profesional.

Bryan Fabián Guevara Guamán,

Jonathan Rafael Jiménez Ledesma.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

En este documento se presenta el desarrollo de un sistema informático que está conformado por una aplicación móvil y una aplicación web, en donde el apartado móvil se encargará de capturar y registrar automáticamente datos que estén relacionados con la posible existencia de desperfectos en las vías, haciendo uso de los recursos de hardware de los dispositivos móviles tales como, el acelerómetro y el GPS, ayudando a que el usuario interactúe lo menos posible con la aplicación móvil, por otro lado, el apartado web tendrá la finalidad de que los usuarios puedan tener acceso libre a los datos recolectados, para visualizar los registros y con opción a descargar los datos para su análisis.

La recolección de los datos en este proyecto se enfoca en las vías de la ciudad de Quito DM, todo esto con la finalidad de delimitar el espacio geográfico para una mejor recolección de datos al momento de entrar en la fase de pruebas del aplicativo.

La construcción de este sistema informático permite usar uno de los bienes más comunes de hoy en día por los usuarios como son los dispositivos móviles inteligentes, de manera que se pueda optimizar recursos, es decir, obtener buenos resultados con la mayor eficiencia a la hora de recolectar los datos mientras el usuario conduce su vehículo.

ABSTRACT

This document presents the development of a computer system that is made up of a mobile application and a web application, where the mobile section will be in charge of automatically capturing and recording data that are related to the possible existence of damage to the roads, making use of the hardware resources of mobile devices such as the accelerometer and GPS, helping the user to interact as little as possible with the mobile application, on the other hand, the web section will have the purpose that users can have access free to the data collected, to view the records and with the option to download the data for analysis.

The data collection in this project focuses on the roads of the city of Quito DM, all this with the purpose of delimiting the geographical space for a better data collection when entering the testing phase of the application.

The construction of this computer system allows users to use one of the most common assets today, such as smart mobile devices, so that resources can be optimized, that is, obtain good results with the greatest efficiency when it comes to collect the data while the user drives their vehicle.

INTRODUCCIÓN

Antecedentes

En la actualidad existen dos tipos de software para la recolección de datos relacionados con las imperfecciones en la calzada, estos fueron proyectos de titulación de la Universidad Politécnica Salesiana (UPS); tales son, “Solución informática de acceso ciudadano, para el registro de la ubicación de baches en la ciudad de Quito DM, utilizando comandos de voz y geolocalización” (Riera, D. Soria, J. , 2020) y “Análisis, diseño y construcción de un sistema informático basado en aplicación móvil y geolocalización para el registro de baches en las vías de la ciudad de Quito" (Castro, C. Velasco, J., 2018).

Además, cabe recalcar que las aplicaciones móviles de los proyectos de titulación anteriormente mencionadas realizan la recolección de datos de forma manual y sobre todo no existe disponibilidad de los datos recolectados para un posterior estudio investigativo.

Problemática

Ante la necesidad de recoger datos de forma automática relacionados con la posible existencia de desperfectos en las vías del Distrito Metropolitano de Quito, se requiere implementar una aplicación móvil que trabaje en la obtención de datos de forma automática usando como sensores los teléfonos inteligentes de los ciudadanos. Como resultado, se permitirá la disposición de estos datos a las personas que deseen realizar un análisis de los mismos para su respectivo estudio.

Es decir, este aplicativo tendrá como enfoque resolver el problema de la recolección de datos manuales, utilizando los recursos del dispositivo móvil, para que éste los obtenga de forma automática, con una mínima intervención del usuario durante el proceso de detección de datos sobre posibles obstáculos en las vías.

Justificación

El desarrollo de este proyecto es con la finalidad de facilitar la gestión en la obtención automática de datos sobre posibles desperfectos en las vías públicas, el uso de tecnologías puede ayudar a mejorar todos estos procesos, optimizando los recursos como tiempo y costos. En el Distrito Metropolitano de Quito existe una gran afluencia de vehículos en las vías, las mismas que en su mayoría se encuentran en mal estado ocasionando malestar al conductor provocando daños en sus vehículos o que son motivo de accidentes en las vías. Según (Secretaría de Movilidad , 2014) los datos demuestran que el modo preferencial para la movilidad de los habitantes en el distrito es el transporte público con un 61.3% y para el transporte privado con un 23.0% dando un total de 3.850.000 viajes proyectados en el año 2014. Según el INEC (Instituto Nacional de Estadísticas y Censos), el parque automotor en el Ecuador creció en más de 1,4 millones de vehículos en una década con un mayor número de vehículos matriculados en la provincia de Pichincha cuya capital es Quito con 540 827 matriculaciones de automotores hasta noviembre del 2019 (El Comercio , 2019).

La EPMMOP (Empresa Pública Metropolitana de Movilidad y Obras Públicas) durante la emergencia sanitaria, ha dado mantenimiento vial (bacheo) a 233 vías del Distrito Metropolitano de Quito, con una intervención de 3 962 baches en más de 70 barrios, con la finalidad de mejorar el tránsito vehicular (Quito Informa, 2020). Pero es necesario tener un sistema eficiente que permita obtener datos aproximados de los desperfectos en las calles para que los procesos de las actividades de bacheo no sean tan lentos. Podemos mencionar el caso del barrio de Solanda ubicado en el sur de Quito, donde un agujero de un metro de diámetro y veinte centímetros de profundidad se formó en las calles Manuel Monteros y José Abarcas (Bravo, 2019). Los moradores del sector indican que el desperfecto se agravó por las lluvias invernales, también hicieron la respectiva notificación con la EPMMOP para dar solución al problema; esto nos permite verificar que no existe un sistema como tal que notifique sobre los

baches en las calles, llevando así estos procesos de forma manual y con mucha demora ya que manifiestan que no se ha dado atención a su petición.

La necesidad de desarrollar una aplicación móvil surgió gracias a la idea de implementar nuevas tecnologías en un sector poco usado, que permitirá a las personas que deseen realizar un análisis de datos, entregar estos antecedentes obtenidos para que puedan realizar un proceso que identifique imperfecciones en la calzada del Distrito Metropolitano de Quito. Además, se debe considerar que el desarrollo de esta aplicación móvil es un prototipo que a posteriori, proporcionará un valor agregado a las empresas dedicadas en el mantenimiento vial y a la comunidad en general, permitiendo que el uso de este tipo de aplicativos móviles de como resultado un término de innovación.

Objetivos

General

Diseñar e implementar una aplicación móvil que permita la recopilación automática de datos relacionados con imperfecciones en la calzada para el Distrito Metropolitano de Quito.

Específicos

Analizar e identificar las variables necesarias que intervienen en el proceso de recolección de datos que se obtendrán de forma automática con el uso de teléfonos inteligentes.

Investigar y definir un buen uso de los recursos de hardware de los dispositivos móviles para que el sistema trabaje de forma eficiente entregando datos.

Diseñar e implementar un modelo de base de datos adecuado a las necesidades del tipo de datos que se obtendrá de forma automática mediante la aplicación móvil.

Evaluar y realizar pruebas de calidad sobre el funcionamiento de la aplicación móvil que recolectará los datos requeridos y el entorno web donde se visualizarán los mismos.

Metodología

Scrum a pequeña escala

Para el desarrollo de este trabajo, se empleó los lineamientos y las buenas prácticas de la metodología ágil Scrum a pequeña escala, debido a la naturaleza del software que es altamente iterativo. La aplicación de esta metodología permitió, a través de los Sprints, desarrollar la aplicación móvil de tal manera que se pudo culminar el proyecto con éxito y con buenos resultados.

Se seleccionó Scrum a pequeña escala debido a que el software que se desarrolló no tiene ningún antecesor como guía en cuanto a la recolección automática de datos, por lo cual, para el desarrollo del mismo, se consideró cierta cantidad de iteraciones hasta llegar a cumplir el objetivo principal, es por eso que la metodología mencionada, aportó un marco de trabajo basado en Sprints, Scrum Master, Product Owner y Development Team que se acopló a la necesidad cambiante que tuvo este proyecto.

De acuerdo con la metodología Scrum a pequeña escala, como se ha definido dentro del marco teórico, se estableció los siguientes roles a los miembros del equipo de trabajo, tal como se muestra en la tabla 1:

Tabla 1

Asignación de roles.

NOMBRE	ROL	FUNCIONES
Bryan Guevara	Scrum Master	Liderar en el cumplimento de los lineamientos de Scrum.
Tutor del trabajo de titulación	Product Owner	Verificar y priorizar los requerimientos para el proyecto.
Bryan Guevara	Development Team	Diseño y construcción de las aplicaciones móvil y web.
Rafael Jiménez	Development Team	Diseño y construcción de las aplicaciones móvil y web.

Nota. Asignación de roles según Scrum a pequeña escala. Elaborado por: Los autores.

El marco de trabajo Scrum a pequeña escala, sufre un cambio dentro de su estructura, en el cual se reemplazan el Product Backlog y el Sprint Backlog para formar el Project Backlog. A continuación, en la tabla 2 se detalla la implementación del instrumento Project Backlog, en el que contiene, las tareas de desarrollo y requisitos del proyecto.

Tabla 2

Project Backlog

ID	NOMBRE	ID SPRINT	PRIORIDAD	OBSERVACIONES
PROJECT-BL-0001	Investigación sobre la información del uso de recursos de hardware en dispositivos móviles.	Sprint 1	Alta	Investigación sobre los sensores acelerómetro y GPS.
PROJECT-BL-0002	Diseño e implementación de la interface de la aplicación móvil.	Sprint 2	Media	Sin observaciones.
PROJECT-BL-0003	Funciones para la detección de movimiento en la aplicación móvil.	Sprint 3	Alta	Se crean los métodos requeridos para el uso de los sensores.
PROJECT-BL-0004	Funciones de integración de bases de datos para almacenar registros.	Sprint 4	Alta	Se usa como gestor de base de datos Cloud Firestore y SQLite.
PROJECT-BL-0005	Diseño e implementación de la interface para la aplicación web.	Sprint 5	Media	Sin observaciones.
PROJECT-BL-0006	Adquisición de un subdominio para cargar la aplicación web.	Sprint 6	Alta	Este subdominio contratado, tiene disponibilidad por un año.
PROJECT-BL-0007	Fase de pruebas sobre los aplicativos móvil y web.	Sprint 7	Alta	Esta fase es para la detección de fallos sobre las aplicaciones.

Nota. Se toma en cuenta los valores de prioridad bajo, medio y alto, siendo alto la prioridad principal. Elaborado por: Los autores.

Y, por último, en este marco de trabajo se establece el Sprint Planning, en donde se establecen los tiempos para las actividades y se centra en el trabajo de planeación de los próximos Sprints tal como se muestra en la tabla 3.

Sprint 1. En esta iteración, se investigó sobre el uso del hardware que ayudó en la recolección de datos los cuales son los sensores acelerómetro y GPS.

Sprint 2. En este Sprint se realizó el diseño e implantación de las interfaces para el aplicativo móvil.

Sprint 3. Una vez concluido el Sprint 1 se procedió con la creación de los métodos para el uso de los sensores.

Sprint 4. En esta iteración se integró los métodos requeridos para el almacenamiento de los datos en una BDD.

Sprint 5. En este Sprint se realizó el diseño e implantación de las interfaces para el aplicativo web.

Sprint 6. En este Sprint se realizó la compra de un subdominio para subir el aplicativo web.

Sprint 7. Una vez terminado todos los Sprints anteriores se realizaron pruebas sobre los aplicativos móvil y web.

Tabla 3

Sprint Planning

ID SPRINT	DURACIÓN	OBSERVACIÓN
Sprint 1	4 semanas	Sin observación.
Sprint 2	1 semana	Se debe cumplir con el Sprint 1.
Sprint 3	4 semanas	Se debe cumplir con el Sprint 1.
Sprint 4	2 semanas	Se debe cumplir con el Sprint 2 y 3.
Sprint 5	1 semana	Se debe cumplir con el Sprint 4.
Sprint 6	1 semana	Se debe cumplir con el Sprint 5.
Sprint 7	2 semanas	Se debe cumplir con todos los anteriores Sprints.

Nota. El tiempo total de la ejecución de todos los Sprints es de 15 semanas. Elaborado por: Los autores.

CAPÍTULO I

1.1 MARCO TEÓRICO

1.1.1 Software libre y datos abiertos

1.1.1.1 Software libre. El software libre posee como característica principal el código abierto, respetando las decisiones de los usuarios. “El software libre es aquel que les da a sus usuarios la libertad de ejecutar, copiar, estudiar, modificar y distribuir el software. En otras palabras, da la posibilidad de controlar el programa y lo que hace” (Souza, rockcontent , 2019). Esto quiere decir que el software libre no es sinónimo de gratuidad, pero sí de disponibilidad a los usuarios para estudiarlo y modificarlo.

1.1.1.2 Datos Abiertos. El concepto de datos abiertos refiere a un tipo de datos que están disponibles para todo tipo de usuario sin restricciones. Los datos abiertos se caracterizan porque cualquier persona los puede utilizar o redistribuir, tomando en cuenta las medidas para preservar el origen y la condición de Datos Abiertos (The World Banck , 2019). En este sentido, los datos abiertos son una fuente de estudio de un determinado tema que los usuarios desean utilizar.

1.2.1 Metodología Scrum

1.2.1.1 Scrum. La familia de métodos de desarrollo ágiles evolucionó a partir de los conocidos ciclos de vida incremental e iterativo. Nacieron de la creencia que un acercamiento más en contacto con la realidad humana y la realidad del desarrollo de productos basados en el aprendizaje, innovación y cambio daría mejores resultados (Deemer, P. Benefield, G. Larman, C. Vodde, B., 2009).

Scrum en la actualidad es uno de los marcos de trabajo más usados para la gestión y el desarrollo de proyectos. “Scrum es una de las metodologías ágiles más conocidas para la gestión de proyectos, consiste en un conjunto de prácticas y roles que permiten el trabajo de entregas

incrementales de un producto" (Rebeca, 2021). Es decir, Scrum fomenta los valores y principios dentro de un equipo de trabajo para poder llevar a cabo el desarrollo de un producto con éxito.

Scrum es un marco de trabajo incremental e iterativo, que se usa para el desarrollo de proyectos de software. Esta metodología está formada por estructuras o ciclos llamados "Sprints", que son iteraciones de un tiempo mínimo de una semana y máximo cuatro semanas.

En este marco de trabajo se cuenta con cuatro roles para la gestión de los proyectos los cuales son:

1.1.1.2.1 Scrum Master. Verifica el cumplimiento de los lineamientos en los que se basa Scrum, por parte de todo el equipo de desarrollo del proyecto.

1.1.1.2.2 Product Owner. Es el encargado de que se cumplan los objetivos y requerimientos planteados para un determinado proyecto.

1.1.1.2.3 Development Team. Es el grupo de desarrolladores que se encargará de trabajar en la construcción del software en base a los requerimientos planteados.

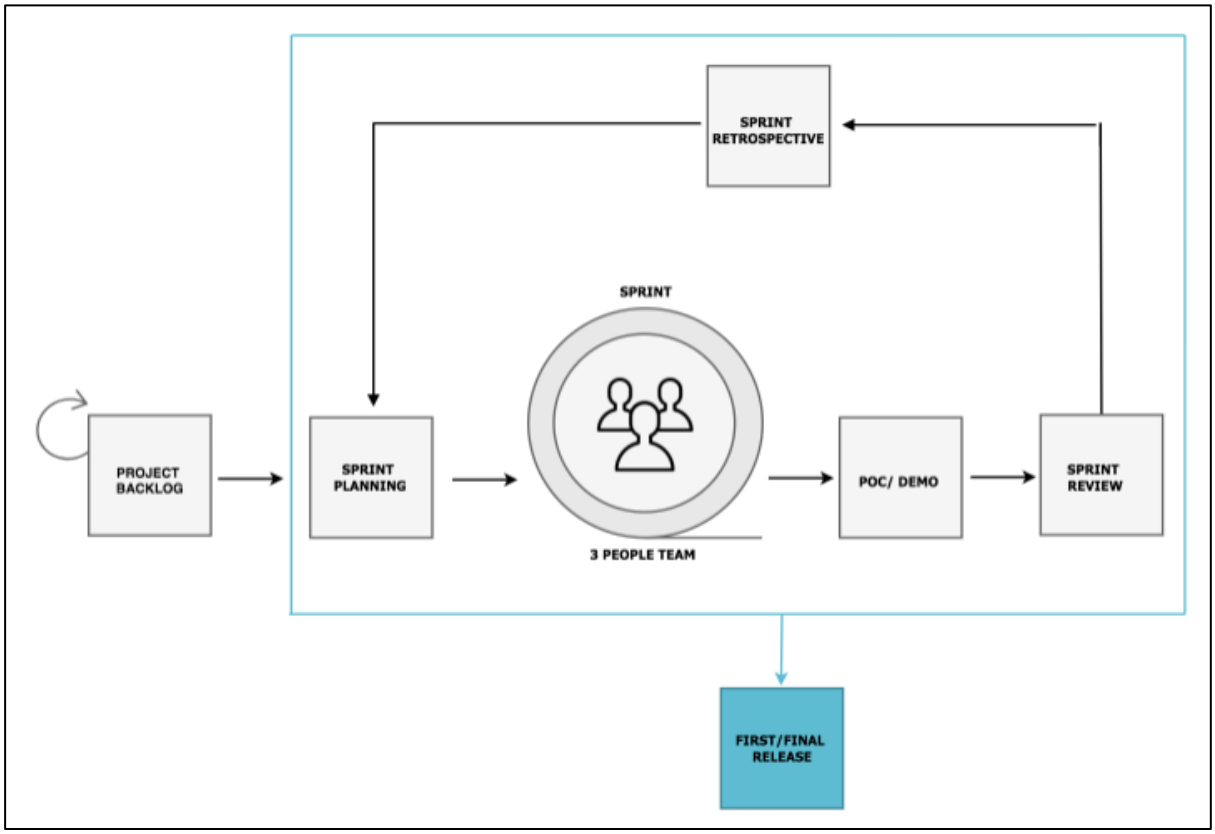
1.2.1.2 Scrum a pequeña escala. Este concepto es prácticamente nuevo dentro de las metodologías ágiles, es la variante del marco de trabajo Scrum clásico. Según Gancarczyk y Griffin (2019):

> El Scrum a pequeña escala lo describe como un mejor marco de trabajo de personas, definido por y para equipos pequeños de un máximo de tres personas y que respalda la planificación, el desarrollo y la entrega de soluciones de software de calidad de producción. Además, este marco de trabajo se centra en el concepto de miembros del equipo que ocupan múltiples roles en el proyecto.

Por consiguiente, este marco de trabajo es adecuado para proyectos conformados por equipos más pequeños tal como se muestra en la figura 1, que, pese a su reducido grupo de trabajo, desean adoptar Scrum y sus principios para el desarrollo de su proyecto.

Figura 1

Marco de trabajo de Scrum a pequeña escala



Nota. Marco de trabajo de Scrum a pequeña escala. Fuente: (Agnieszka Gancarczyk y Leigh Griffin, 2019).

1.2.1.3 Principios de Scrum a pequeña escala. Scrum a pequeña escala cuenta con principios tales como, la comunicación basada en valores de los miembros del proyecto, este principio refiere a la comunicación que debe tener el equipo de desarrollo de forma interna y externa. Luego se tiene como siguiente principio el desarrollo de calidad, que se enfoca en hacer uso de buenas prácticas y lineamientos de calidad durante el desarrollo del software. Otro principio fundamental es la propiedad de la entrega, que permite al equipo tomar la iniciativa y hacerse responsable de entregar la solución del software de un Sprint a otro. Y, por último, el principio del cierre iterativo que se orienta en identificar las brechas en los requisitos a través de un enfoque de firma iterativa (Agnieszka Gancarczyk, 2019).

1.2.2 Herramientas

1.2.2.1 Visual Studio Code. Visual Studio Code es un entorno de desarrollo integrado, que posee portabilidad para los diferentes Sistemas Operativos, entre sus características principales está presente la usabilidad ya que gracias a su soporte nativo se puede programar en diferentes lenguajes como JS, TypeScript y Node.js, además tiene múltiples extensiones como C ++, C #, Java, Python, PHP, Go (Código de Visual Studio , 2021). Por tanto, esta herramienta es ideal para el desarrollo de cualquier proyecto de software.

1.2.2.2 NetBeans. NetBeans es un entorno de desarrollo, que se creó específicamente para Java, es libre, por lo tanto, está disponible para todos los usuarios. Es un entorno de desarrollo que permite programar tanto en Java, o en otros lenguajes como C, C++, PHP, etc. (PC Resumen , 2019). Esto indica que NetBeans, es un IDE multifuncional, que permite trabajar con diferentes lenguajes de programación, haciendo de esta herramienta un entorno robusto para el desarrollo de proyectos de software.

1.2.2.3 GitHub. GitHub es actualmente la plataforma de desarrollo colaborativo más usada por los programadores, la cual permite alojar proyectos y hacer controles de versiones del software. Fernández (2019) plantea que Github es un entorno creado para almacenar el código de los proyectos de software de cualquier desarrollador, además esta plataforma permite descargar el código de los proyectos y también entrar a otros perfiles para colaborar con el desarrollo del código. Por consiguiente, GitHub es una herramienta colaborativa potente basada en la computación en la nube que brinda servicios colaborativos a los desarrolladores.

1.2.2.4 Android Studio. Android Studio es un IDE que cuenta con un editor y simulador virtual, diseñado en específico para el sistema operativo Android. Este entorno de desarrollo integrado está orientado para el desarrollo de apps para Android. Además del potente editor de códigos Android Studio ofrece funciones que aumentan la productividad cuando se desarrolla apps para Android (Developers, 2021). En conclusión, Android Studio es una herramienta comúnmente usada para crear el ambiente de virtualización mejor conocido como AVD (Dispositivo virtual de Android).

1.2.2.5 Microsoft Project. Microsoft Project es una herramienta muy usada por los administradores de proyectos, la cual permite llevar a cabo una buena gestión sobre los proyectos. Según Softtrader (2020), "Microsoft Project (MSP) es un software de gestión de proyectos creado para que los directores de proyectos gestionen proyectos. Con Microsoft Project puede planificar proyectos, asignar tareas, administrar recursos, crear informes y más". Dicho de otra manera, esta herramienta permite evaluar las actividades y estimar tiempos de duración de un proyecto.

1.2.2.6 LucidChart. LucidChart es una herramienta colaborativa que permite a los usuarios trabajar remotamente sobre un documento en la nube. "Lucidchart es un espacio de trabajo gráfico que combina diagramas, visualización de datos y colaboración para acelerar la comprensión e impulsar la innovación" (Lucidchart , 2021). En otras palabras, esta herramienta permite trabajar con elementos visuales para diseñar diferentes tipos de diagramas, según el interés del usuario.

1.2.2.7 MockFlow. MockFlow es una herramienta UX colaborativa gracias a la implementación de la nube, permitiendo el análisis y participación de los colaboradores en tiempo real. Esta aplicación web es una considerable solución para diseñadores web y desarrolladores que requieran una herramienta para realizar esquemas de páginas en su flujo de trabajo de diseño. Permite crear estructuras visuales en la nube y es excelente para integrar rápidamente las herramientas como por ejemplo con Trello (MockFlow, 2019). Por tanto, Mockflow es la herramienta adecuada para modelar estructuras visuales en el diseño gráfico de un proyecto.

1.2.3 Lenguajes de programación

1.2.3.1 Java. Java es una plataforma informática y un lenguaje de programación, cuya característica principal es la creación de objetos permitiendo una mejor abstracción de los problemas que los usuarios deseen resolver mediante este lenguaje. Está orientado a objetos, independiente de la plataforma hardware donde se desarrolla y utiliza una sintaxis similar a la de C++. Este lenguaje tiene una curva de aprendizaje baja, además se dispone de una gran cantidad de código de terceros existente. Java, ofrece un código robusto, que permite un manejo automático de la memoria, lo que minimiza el número de errores (SEAS, 2019). En otras palabras, Java es un lenguaje de programación muy fácil de aprender y debido a eso es muy utilizado por los usuarios en distintos campos.

1.2.3.2 Dart. Dart es un lenguaje de programación orientado a objetos de código abierto, desarrollado por Google, es relativamente joven ya que fue revelado en el 2011. Este lenguaje de programación se ha vuelto común en su uso debido a la optimización para clientes sobre las aplicaciones que son rápidas y multiplataforma, además Dart genera una buena experiencia de usuario con la menor cantidad posible de sintaxis y, al mismo tiempo, prestar atención a la memoria disponible del dispositivo (OINOS, 2020). Por tanto, Dart posee varias características como la programación estructurada y un lenguaje familiar con una fácil curva de aprendizaje.

1.2.3.3 PHP. PHP (Hypertext Prepocessor), es uno de los lenguajes de programación más populares de código abierto dedicado para la creación de aplicaciones web. Este lenguaje de programación que permite desarrollar aplicaciones y crear sitios web, es fácil de usar y está en constante perfeccionamiento es una opción segura para los desarrolladores que desean trabajar en proyectos de software (Souza, 2020). En definitiva, PHP es un lenguaje de programación de código abierto y multiplataforma, teniendo como característica principal la separación de estructuras para no contaminar código a esto se le conoce como Modelo Vista Controlador (MVC), permitiendo de esta manera tener un código ordenado y limpio.

1.2.4 Framework

1.2.4.1 Flutter. Flutter es el Framework de Dart, esta herramienta de Google permite realizar aplicaciones nativas para móvil, web y escritorio desde una única base de código, por lo que el rendimiento alcanzado es superior a aplicaciones basadas en web-views. Flutter utiliza sus propios componentes, llamados widgets, por lo que la misma aplicación se verá igual en cualquier dispositivo, independientemente de su sistema operativo o la versión. Gracias a ello, el desarrollador no tiene que preocuparse por que el diseño de su aplicación se vea mal en dispositivos antiguos (Diví, 2020). Dicho de otra manera, Flutter es un marco de trabajo de código abierto enfocado para el desarrollo de aplicaciones móviles.

1.2.5 Gestor de bases de datos

1.2.5.1 Cloud Firestore. Cloud Firestore es una base de datos de tipo NoSQL orientada al desarrollo de dispositivos móviles, cuya característica principal es la sincronización de datos en tiempo real. Esta base de datos es flexible y escalable para los desarrolladores que trabajan con servidores, dispositivos móviles, la Web desde Firebase y Google Cloud. Cloud Firestore además ofrece una integración sin interrupciones con otros productos de Firebase y Google Cloud, incluido Cloud Functions (Firebase , 2021). En otras palabras, el uso de Cloud Firestore es ideal para el desarrollo de aplicaciones móviles debido a su alojamiento en la nube y facilidad de acceso a los datos.

1.2.5.2 SQLite. Es una biblioteca escrita en lenguaje C que usa un motor de base de datos SQL pequeño, rápido, autónomo, de alta confiabilidad y con todas las funciones. Además, es el gestor de base de datos más utilizado para entornos móviles ya que SQLite está integrado en todos los teléfonos móviles, debido a que posee un formato de archivo estable, multiplataforma y compatibilidad con versiones anteriores (SQLite , 2021).

1.2.6 Componentes para la recopilación de datos

1.2.6.1 Acelerómetro. El acelerómetro es un componente tecnológico útil para la medición de aceleraciones. Según Fernández (2019):

> Es un componente mecánico de tamaño reducido gracias a su nanotecnología, y fabricado en silicio. El acelerómetro sirve para que el móvil sepa en qué orientación está colocado, de manera que el dispositivo pueda saber cuándo está en posición horizontal, o en vertical, e incluso cuando está boca abajo. El acelerómetro del móvil consta de una parte móvil que se mueve dependiendo de la aceleración que se le aplique, y de otra fija que interpreta el voltaje resultante de este movimiento para determinar la velocidad a la que lo hace y su orientación. En los móviles suelen estar compuestos de tres ejes para medir el movimiento en un espacio tridimensional.

Esto indica que el uso de este sensor permitirá recoger datos sobre los ejes x, y, z, para medir diferentes tipos de aceleraciones.

1.2.6.2 Sensor GPS. El GPS es un componente tecnológico útil para la medición de posicionamiento mediante coordenadas. Según Fernández (2019):

> Los sensores GPS del móvil están diseñados para estar continuamente leyendo la señal que le transmiten los satélites de GPS, utilizando por lo general la señal de varios de ellos para triangular la posición del dispositivo móvil. Esta señal no consume datos, pero puede ser más débil en interiores, haciendo que la posición sea más difícil de medir cuando se está dentro de un lugar cerrado.
>
> Por lo general, suele utilizar los ángulos de intersección de al menos tres satélites respecto a nuestro móvil para triangular dónde se encuentra. Cada vez más, a estos datos también se les une la distancia que hay del móvil a las torres de telefonía móvil para obtener una ubicación más precisa.

Por tanto, el sensor GPS permitirá obtener datos sobre geolocalización tales como la longitud y latitud.

CAPÍTULO II

2.1 ANÁLISIS Y DISEÑO

2.1.1 Análisis

En esta sección, primero se identificó las variables esenciales que intervienen en la recolección de datos las cuales son: el tipo de vehículo a conducir y la conectividad a internet del dispositivo móvil. Además, gracias a la ingeniería de software se examinó los requerimientos funcionales y no funcionales de las aplicaciones móvil y web, permitiendo estructurar el marco de referencia para modelar de forma detallada el sistema.

2.1.1.1 Requerimientos del software. A continuación, se planteó los requerimientos funcionales y no funcionales de la aplicación, permitiendo que el equipo de desarrollo se haya guiado al momento de la elaboración del proyecto técnico.

2.1.1.1.1 Requerimientos funcionales. Tanto la aplicación móvil, como la aplicación web realizan las funciones indicadas en la tabla 4, en donde se realizó una estimación en porcentaje de lo que cada módulo representa en el proyecto y con sus respectivas especificaciones en la tabla 5.

Tabla 4

Requerimientos funcionales

CÓDIGO	MÓDULO	PORCENTAJE
RF-01	Ingreso con alias.	5%
RF-02	Elegir el tipo de vehículo de transporte.	5%
RF-03	Captura de datos de un evento.	80%
RF-04	Registro en la BDD.	5%
RF-05	Visualización de datos registrados.	5%

Nota. Se asocia a cada requerimiento funcional un código identificador, además se asignó un porcentaje a cada módulo. Elaborado por: Los autores.

Tabla 5

Especificación de requerimientos funcionales - aplicativos móvil-web

CÓDIGO	DESCRIPCIÓN	ENTRADAS	SALIDAS	PROCESO	PRIORIDAD
RF-01	La aplicación permitirá acceder al sistema, ingresando un alias y un correo electrónico válido.	- Alias - Correo electrónico.	- Confirmación de acceso - Acceso a seleccionar tipo de vehículo.	Al abrir la aplicación por primera vez, se mostrará un formulario para ingresar un nombre o un alias y el correo electrónico. Luego de ingresar los datos solicitados correctamente, se deberá presionar el botón ¨Ingresar¨ que llevará a la ventana para seleccionar el tipo de vehículo con el que se va a desplazar. Caso contrario, si el alias no es el adecuado o el correo no es válido, se mostrará un mensaje de error y no se permitirá acceder a la siguiente pantalla.	Media
RF-02	La aplicación permitirá escoger entre tres tipos de medio de transporte (auto, bicicleta y motocicleta), para la recolección de datos.	- Bicicleta - Motocicleta - Carro	- Empezar con el registro de datos mientras se conduce.	Luego de ingresar a la aplicación con un alias y correo electrónico, se procede a seleccionar el tipo de vehículo que va a usar (auto, bicicleta y motocicleta), según sea el caso, para luego acceder a la pantalla de recolección de datos.	Alta
RF-03	La aplicación móvil recoge los datos capturados como posible desperfecto en la vía mientras se está conduciendo el vehículo seleccionado.	- Fecha - Eje X - Eje Y - Eje Z - Correo electrónico - Hora - Latitud - Longitud - Usuario - Vehículo	- Captura de datos mediante la aplicación móvil a través de los sensores del Smartphone.	Posterior a la selección del tipo de vehículo, la aplicación registrará los movimientos que puedan asociarse a un desperfecto en la calzada y los capturará.	Alta
RF-04	La aplicación debe validar si existe o no conexión a internet, para posteriormente insertar en Firebase o en SQFLite.	- Fecha - Eje X - Eje Y - Eje Z - Correo electrónico - Hora - Latitud - Longitud - Usuario - Vehículo	- Registro de datos recolectados en la base de datos Firebase si hay conexión a internet. - Registro de datos recolectados en la base de datos SQFLite si no hay conexión a internet.	La aplicación móvil será capaz de verificar si hay conexión a internet o no. En el caso de que haya conexión, los datos obtenidos, se registrarán en Firebase, por otro lado, si el usuario no cuenta con acceso a internet, los datos se guardarán en la base interna del dispositivo SQFLite. Si posteriormente, el usuario accede a internet, los datos almacenados en SQFLite, se borrarán y se guardarán en Firebase.	Alta
RF-05	La aplicación web permitirá tener disponibles los datos almacenados en Firebase tanto visual como descarga.	- Fecha - Eje X - Eje Y - Eje Z - Correo electrónico - Hora - Latitud - Longitud - Usuario - Vehículo	- Datos almacenados en Firebase descargables en formato de texto plano.	La aplicación web permitirá visualizar los datos almacenados en Firebase, mediante una tabla, además, se deberá colocar un botón que permita descargar todos los datos en formato de texto plano (Excel).	Alta

Nota. Se toma en cuenta los valores de prioridad bajo, medio y alto, siendo alto la prioridad principal. Elaborado por: Los autores.

2.1.1.1.2 Requerimientos no funcionales. Para los requerimientos no funcionales, tanto la aplicación móvil como la aplicación web, se detallan en la tabla 6 y con sus respectivas especificaciones en la tabla 7.

Tabla 6

Requerimientos no funcionales

CÓDIGO	MÓDULO
RNF-01	Disponibilidad de la aplicación móvil y web.
RNF-02	Disponibilidad de datos en la aplicación web.
RNF-03	Usabilidad en las aplicaciones web y móvil.

Nota. Se asocia a cada requerimiento no funcional un código identificador. Elaborado por: Los autores

Tabla 7

Especificaciones de requerimientos no funcionales - aplicativos móvil-web

CÓDIGO	DESCRIPCION	ENTRADAS	SALIDAS	PROCESO	PRIORIDAD
RNF-01	La aplicación móvil y web deben brindar disponibilidad completa para su uso.	- Arquitectura del sistema	-	La aplicación móvil y web deben estar siempre disponibles para su uso continuo y su funcionamiento no debe ser interrumpido a menos que sea afectado por factores externos que estén fuera del alcance.	Alta
RNF-02	La aplicación web debe brindar disponibilidad completa para su uso.	- Arquitectura del sistema web.	Datos disponibles visualmente y descargables.	La aplicación web debe dar disponibilidad completa para su uso, brindando a los usuarios una fuente de datos que esté constantemente a su disposición sin interrupciones a menos que sea afectado por factores externos que estén fuera del alcance.	Alta
RNF-03	El diseño de las interfaces gráficas tanto de la aplicación web y móvil, deben contar con un diseño minimalista.	- Botones - Formularios - Iconos	Ventanas que componen interfaz gráfica móvil y web.	El sistema informático debe estar desarrollado con interfaces minimalistas y amigables con el usuario, para que sea fácil de utilizar, haciendo que la experiencia con las aplicaciones sea fluida y cómoda.	Alta

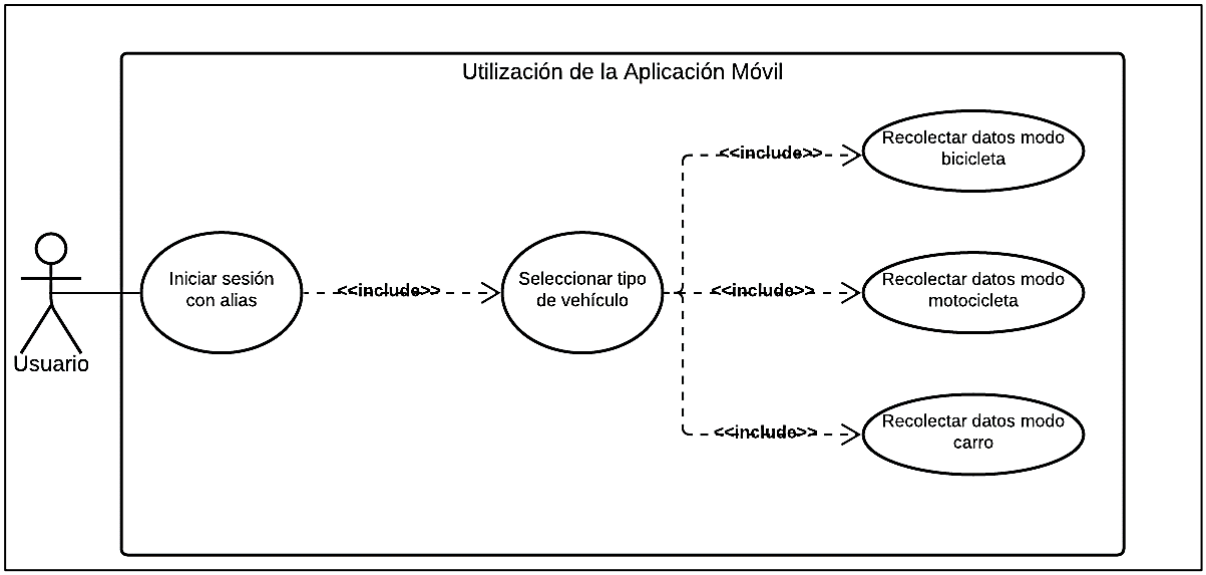
Nota. Se toma en cuenta los valores de prioridad bajo, medio y alto, siendo alto la prioridad principal. Elaborado por: Los autores.

2.1.1.2 Casos de uso

2.1.1.2.1 Caso de uso para la aplicación móvil. Tal como se muestra en la Figura 2**,** este módulo otorga al usuario ingresar con un alias a la aplicación móvil, donde puede seleccionar un modo de conducción para posteriormente recolectar los datos de forma automática.

Figura 2

Diagrama de caso de uso para la aplicación móvil

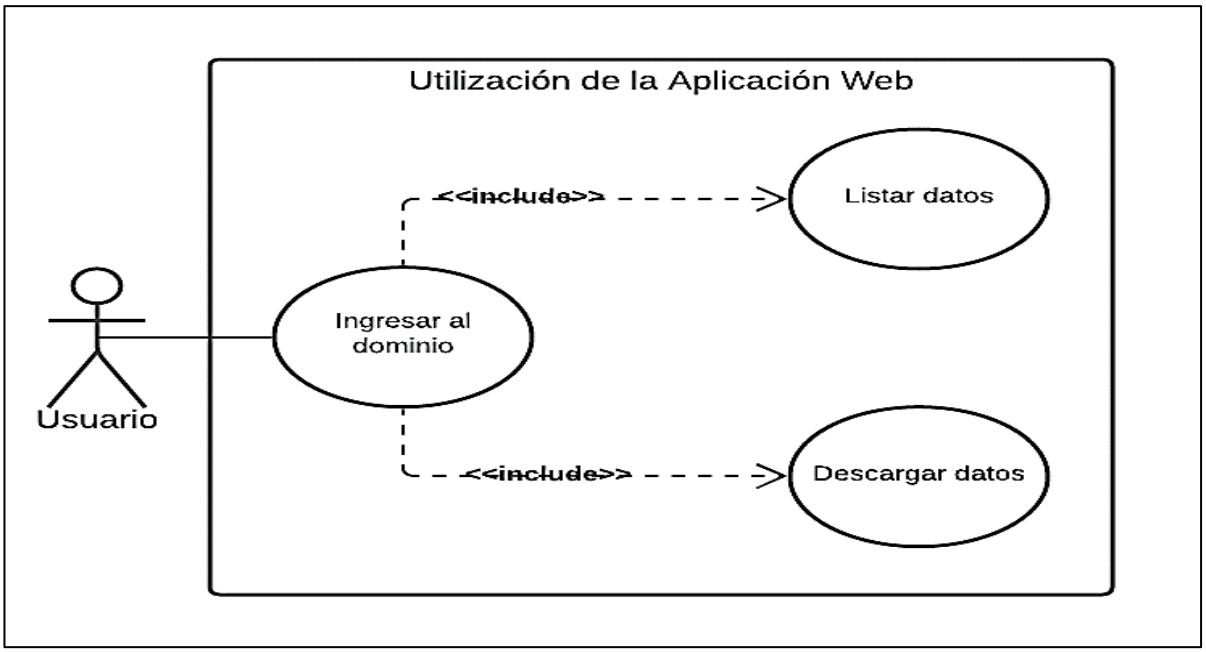


Nota. Diagrama de caso de uso que describe a la aplicación móvil. Elaborado por: Los autores.

2.1.1.2.2 Caso de uso para la aplicación web. Este módulo tiene la finalidad de permitir la visualización y descarga de datos en formato texto plano, como se presenta en la figura 3.

Figura 3

Diagrama de caso de uso para la aplicación web



Nota. Diagrama de caso de uso que describe a la aplicación web. Elaborado por: Los autores.

2.1.2 Diseño

Este apartado, se centra en la trasformación de los requerimientos a través de mecanismos esquemáticos que detallan las funciones estructurales del proyecto técnico.

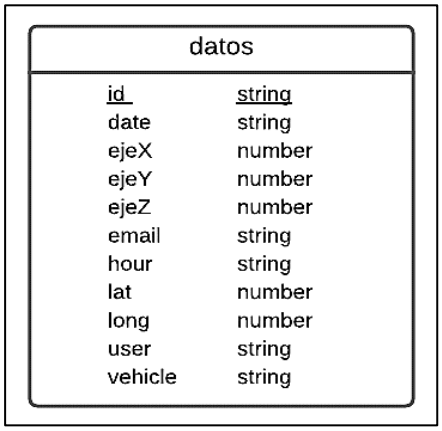
2.1.2.1 Diagrama de base de datos. De acuerdo con las necesidades del proyecto se implementaron dos bases de datos, la primera es Cloud Firestore que es una base de datos NoSQL, siendo de utilidad para el almacenamiento y acceso de los datos desde la aplicación móvil y web.

Además, se ha implementado la BDD SQLite, para el apartado móvil que permite recolectar datos sólo en caso de que el usuario no tenga conectividad a internet para luego enviar los registros a Cloud Firestore, una vez que el usuario se conecte a internet.

2.1.2.1.1 Diagrama de base de datos Cloud Firestore. En el esquema de la figura 4, se muestra la estructura de la BDD que se encarga de almacenar los registros recolectados a través de la aplicación móvil siempre y cuando el usuario esté conectado a internet. Además, desde el apartado web se puede visualizar y descargar los mismos.

Figura 4

Diagrama de base de datos Cloud Firestore

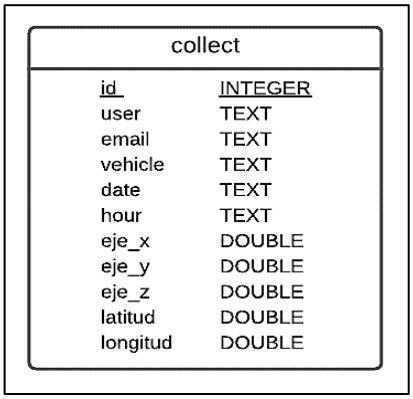


Nota. Estructura de tabla de Cloud Firestore. Elaborado por: Los autores.

2.1.2.1.2 Diagrama de base de datos SQLite. En el esquema de la figura 5, se muestra la estructura que permite almacenar los registros en la base interna del dispositivo móvil, mientras el usuario no tenga conexión a internet.

Figura 5

Diagrama de base de datos de SQLite



Nota. Estructura de tabla de SQLite. Elaborado por: Los autores.

2.1.2.2 Diagrama de clases. En este punto, los diagramas describen cada una de las clases que conforman la aplicación móvil y web, además, se detalla cómo se relacionan cada una de ellas.

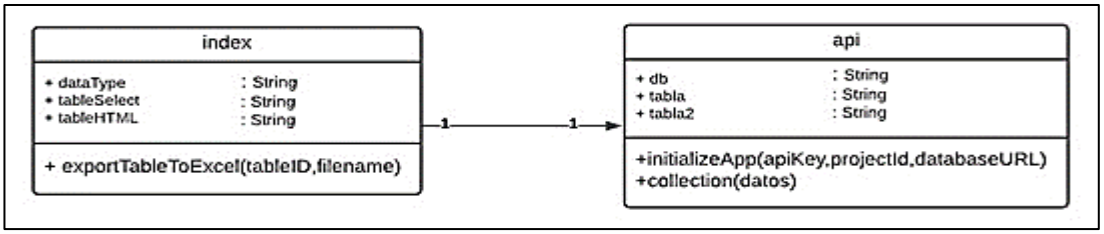
2.1.2.2.1 Diagrama de clases de aplicación web. A continuación, en la figura 6, se presenta el diagrama de clases de la aplicación web.

2.1.2.2.1.1 Clase index. En este apartado se presenta la tabla con los datos registrados en la BDD, además permite descargarlos en formato xls.

2.1.2.2.1.2 Clase api. Aquí se consume los servicios de Cloud Firestore para obtener todos los registros almacenados en la BDD.

Figura 6

Diagrama de clases de aplicación web



Nota. Clases de la aplicación web. Elaborado por: Los autores.

2.1.2.2.2 Diagrama de clases de aplicación móvil. A continuación, en la figura 7, se presenta el diagrama de clases de la aplicación móvil.

2.1.2.2.2.1 Clase login_page. Permite acceder a la aplicación con un alias.

2.1.2.2.2.2 Clase driving_mode. Permite seleccionar el vehículo a conducir.

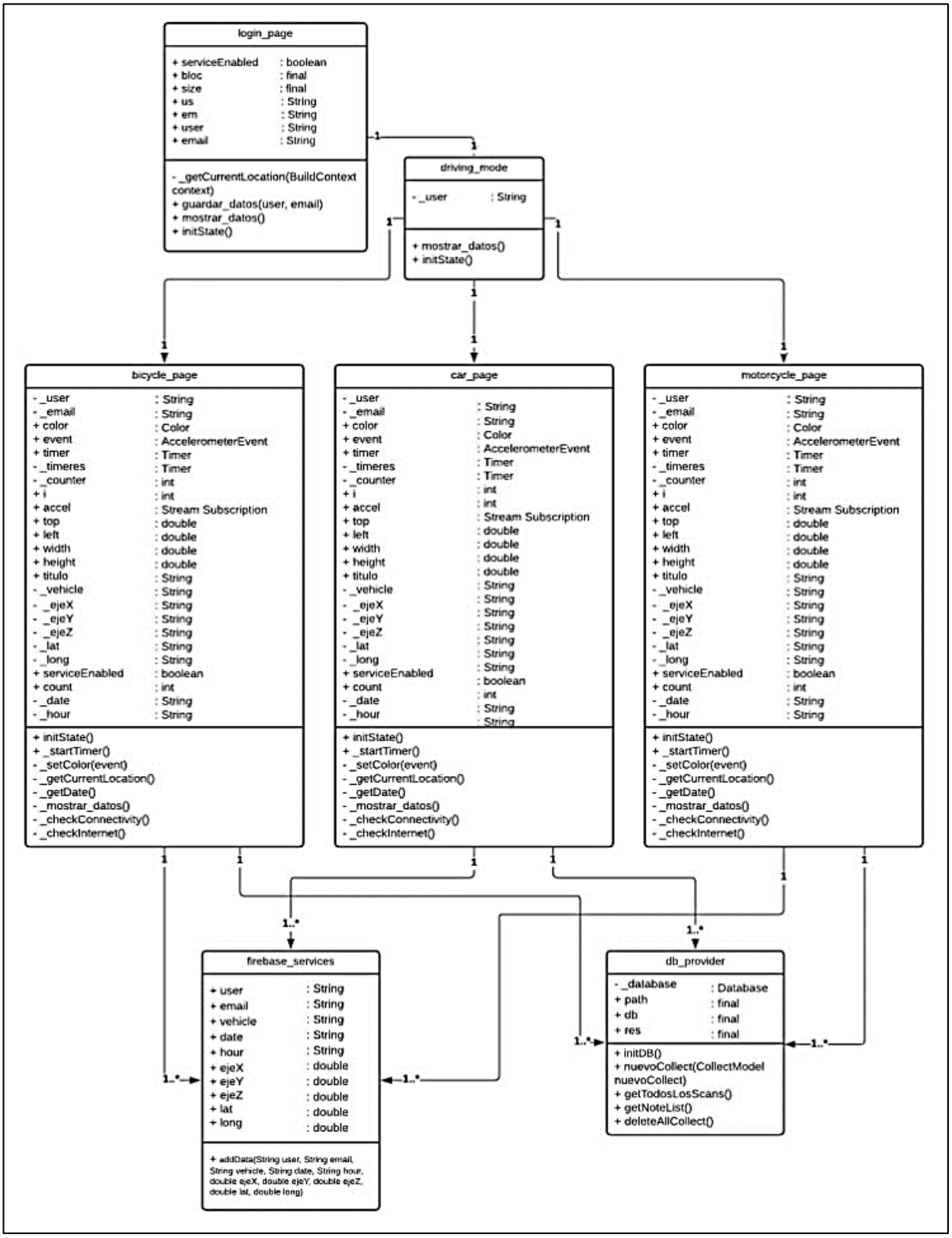
2.1.2.2.2.3 Clases bicycle_page, car_page y motorcycle_page. Permite recolectar los datos relacionados con los posibles desperfectos en las vías de acuerdo con el modo de conducción seleccionado.

2.1.2.2.2.4 Clase firebase_services. Permite guardar los datos en la BDD externa.

2.1.2.2.2.5 Clase db_provider. Permite almacenar los datos en la BDD interna.

Figura 7

Diagrama de clases de aplicación móvil



Nota. Diagrama que describe las clases de la aplicación móvil. Elaborado por: Los autores.

2.1.2.3 Diagrama de componentes. Este apartado consta de tres entornos principales, detallados en la Figura 8.

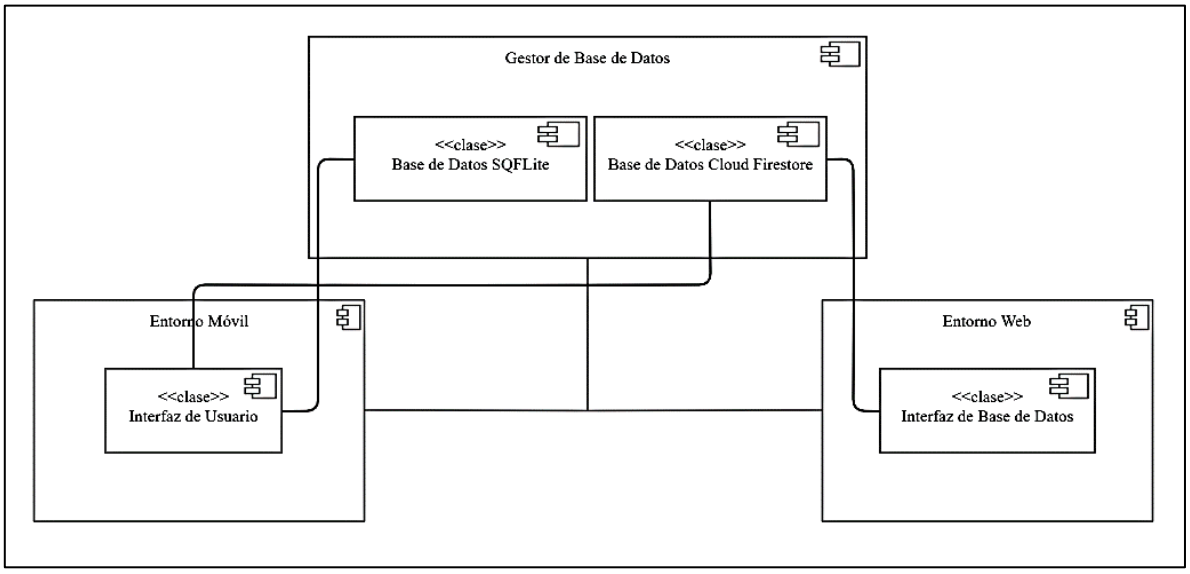
2.1.2.3.1 Entorno móvil. Este entorno es el principal y está desarrollado en Flutter, que permite recoger los datos.

2.1.2.3.2 Entorno web. Este apartado es de gran utilidad para los usuarios que desean acceder a los datos recolectados por la aplicación móvil para un posterior estudio de investigación.

2.1.2.3.3 Gestor de base de datos. Se utilizó dos BDDs (SQLite y Cloud Firestore) donde se almacenan los datos geoespaciales recolectados por la aplicación móvil.

Figura 8

Diagrama de componentes



Nota. Diagrama de componentes de las aplicaciones. Elaborado por: Los autores.

2.1.2.4 Interfaces gráficas. Para esta división, se muestra los prototipos de las GUI (Interfaces graficas de usuarios) del aplicativo móvil y del aplicativo web, los mismos que se diseñaron en Mockflow que es una herramienta de marketing digital.

2.1.2.4.1 Aplicación móvil. Para la aplicación móvil se ha dividido en tres interfaces.

2.1.2.4.1.1 Interface ingreso de alias. Esta pantalla es el inicio de la aplicación móvil, donde al usuario se le solicita permisos para el uso de los sensores, además, debe ingresar un alias y correo electrónico, tal como se muestra en la figura 9.

Figura 9

Interface ingreso de alias

Nota. Diseño de la pantalla Ingresar Alias. Elaborado por: Los autores.

2.1.2.4.1.2 Interface selección de vehículo. La interface gráfica que se muestra en la figura 10, cuenta con tres modos de conducción, en el cual el usuario, debe seleccionar un tipo de vehículo.

Figura 10

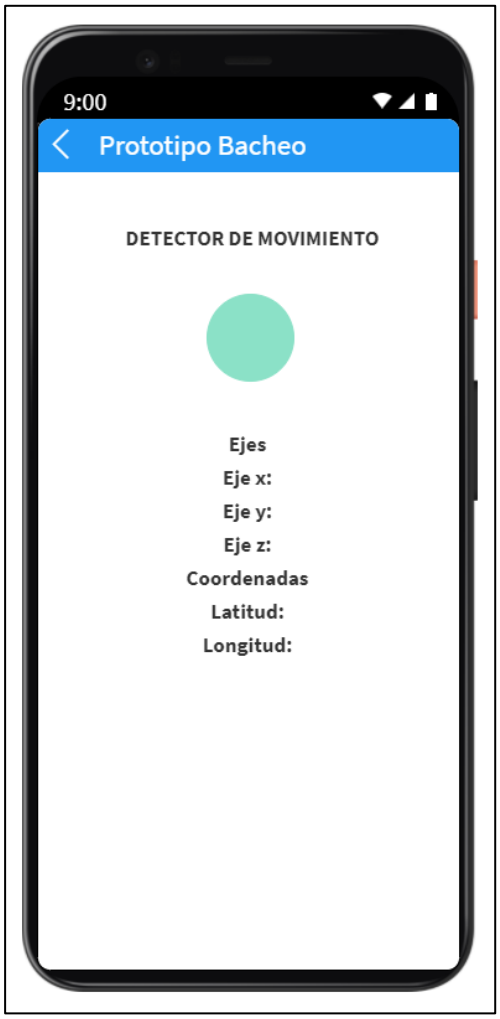
Interface selección de vehículo

Nota. Diseño de pantalla Selección Vehículo. Elaborado por: Los autores.

2.1.2.4.1.3 Interface de recolección de datos. Una vez que el usuario haya seleccionado un tipo de vehículo, la aplicación pasa a trabajar de forma automática para la recolección de los datos, en el cual solo se muestran los datos de georreferencia tal como se expone en la figura 11.

Figura 11

Interface de recolección de datos

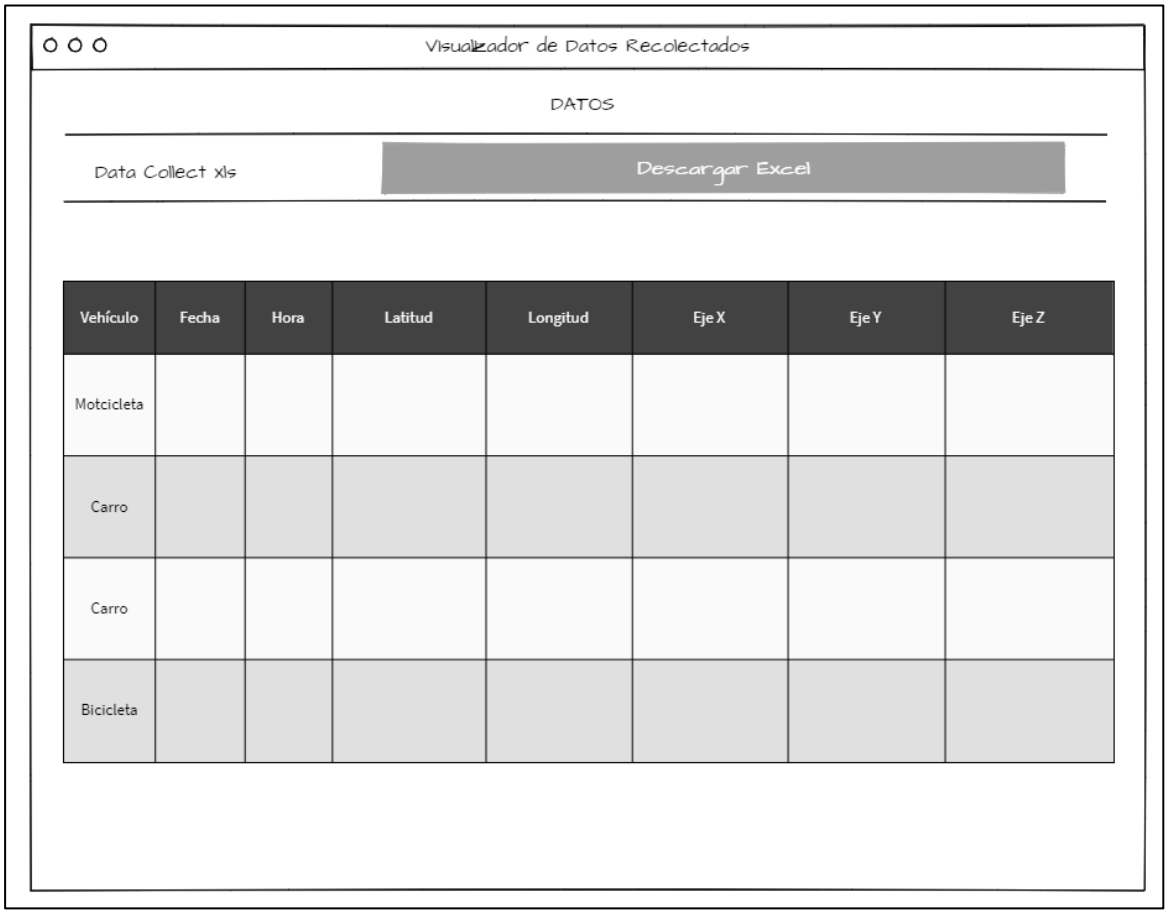


Nota. Diseño de pantalla Detector de Movimiento. Elaborado por: Los autores.

2.1.2.4.2 Interface de aplicación web. Para el entorno web se diseñó una página, como se indica en la figura 12, la cual se subió a un hosting con el subdominio http://registrodatosbache.sbgconsultancy.com/, en donde se le permite al usuario ver y descargar los datos que se han recolectado a través del dispositivo móvil y se encuentran almacenados en Cloud Firestore.

Figura 12

Interface de aplicación web



Nota. Diseño de pantalla de la aplicación web. Elaborado por: Los autores.

CAPÍTULO III

3.1 CONSTRUCCIÓN Y PRUEBAS

3.1.1 Construcción.

En esta sección, se presenta el código fuente de la aplicación móvil y la aplicación web haciendo énfasis en las partes más importantes de la construcción del sistema.

3.1.1.1 Estándares y buenas prácticas de programación. En esta sección se detalla cuáles fueron los estándares y buenas prácticas usados durante el proceso de desarrollo como el tipo de escritura para métodos y nombres de clases, además del uso de herramientas de software como parte de buenas prácticas de programación.

3.1.1.1.1 Creación de clases. La asignación del nombre de las clases fue de acuerdo a la creación de cada página, esto fue de gran ayuda al momento de crear el enrutamiento para una mejor navegación en la aplicación móvil.

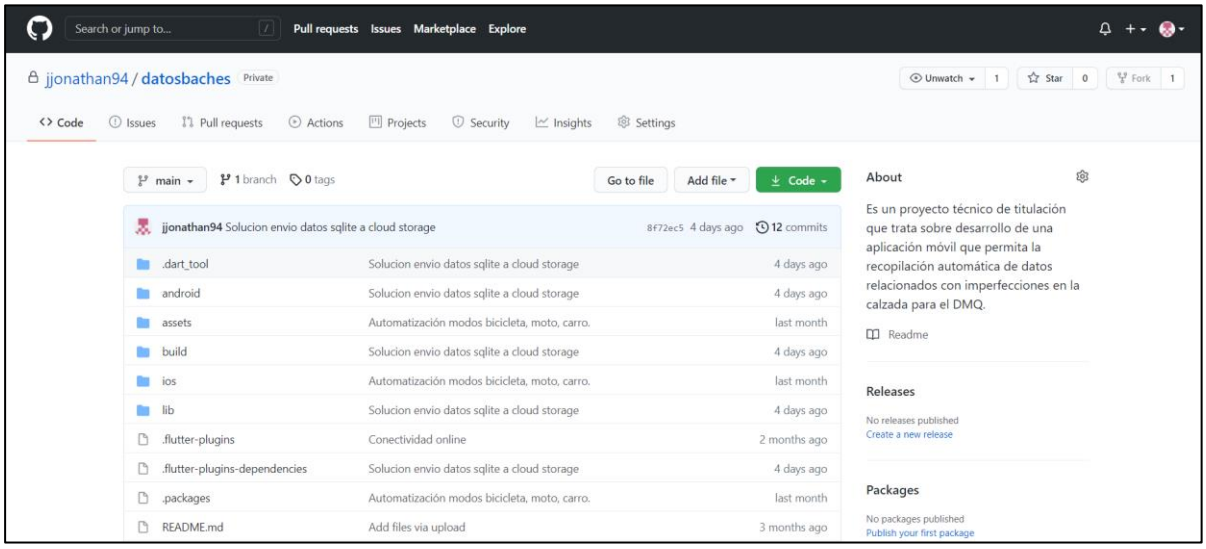
3.1.1.1.2 Creación de métodos. Para la creación de los métodos, se estableció que todos sean privados ya que se usaron solo dentro de esa clase, las definiciones de nombres de los métodos están estandarizados de tal manera que van en inglés y minúscula el verbo de acción, seguido de una palabra con mayúscula en la primera letra.

3.1.1.1.3 Tabulación del código. Una buena práctica dentro del mundo de la programación es tabular las líneas de código, de esta manera para cualquier desarrollador es fácil entender cada estructura que se ha programado.

3.1.1.1.4 Uso de GitHub. El uso de esta herramienta colaborativa fue de gran soporte para el equipo de desarrollo ya que permitió almacenar el proyecto en la nube, como se muestra en la figura 13, que brindó seguridad y respaldo durante la construcción del aplicativo móvil.

Figura 13

Repositorio del proyecto datosbaches en GitHub

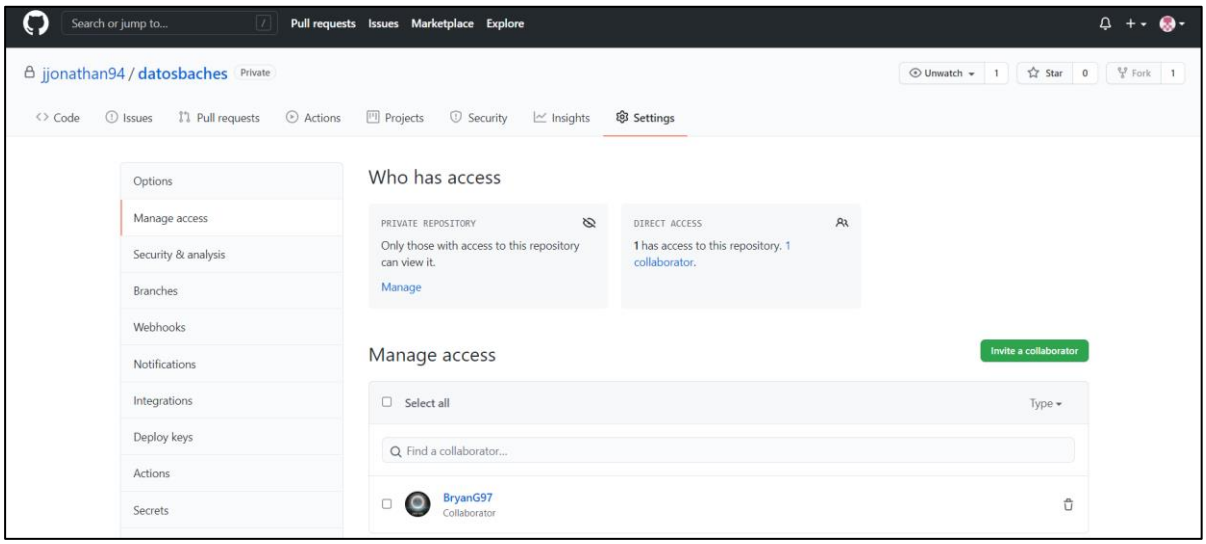


Nota. Archivos del proyecto subidos en GitHub. Fuente: (GitHub, 2021).

Este proyecto fue conformado por dos estudiantes, desde este punto surgió la necesidad de usar una herramienta colaborativa para desarrolladores y la mejor opción fue el uso de GitHub, a continuación, en la figura 14, se presenta la administración de acceso al proyecto, donde se pudo agregar los colaboradores que trabajaron conjuntamente en el desarrollo del proyecto.

Figura 14

Permisos de acceso como colaborador al proyecto datosbaches



Nota. Permisos de acceso como colaborador al proyecto en GitHub. Fuente: (GitHub, 2021).

3.1.1.2 Código. En este apartado se describe cada componente desarrollado durante la etapa de la programación de las aplicaciones móvil y web.

3.1.1.2.1 Aplicación móvil. En esta sección, se expondrá solo el código implementado para la aplicación móvil.

3.1.1.2.1.1 Permisos. Al crear el proyecto en Flutter se genera de forma automática el archivo AndroidManifest.xml como se muestra en la figura 15, que es el archivo de configuración de la aplicación móvil.

Figura 15

Código del archivo androidmanifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.datosbaches">
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
    <application
        android:name="io.flutter.app.FlutterApplication"
        android:label="Recolector de Datos"
        android:icon="@mipmap/ic_launcher">
        <activity
            android:name=".MainActivity"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|
            smallestScreenSize|locale|layoutDirection|fontScale|screenLayout|density|uiMode"
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">
            <meta-data
              android:name="io.flutter.embedding.android.NormalTheme"
              android:resource="@style/NormalTheme"
              />
            <meta-data
              android:name="io.flutter.embedding.android.SplashScreenDrawable"
              android:resource="@drawable/launch_background"
              />
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <meta-data
            android:name="flutterEmbedding"
            android:value="2" />
    </application>
</manifest>
```

Nota. Archivo androidmanifest.xml. Elaborado por: Los autores.

3.1.1.2.1.2 ***Dependencias.*** El uso de dependencias tuvo como finalidad, permitir el uso de librerías. Las dependencias se ingresaron dentro del archivo de configuración llamado pubspec.yaml, dentro de este archivo es necesario respetar los lineamientos para el ingreso de código, tal como se muestra en la figura 16, para evitar errores de sintaxis al momento de la ejecución del proyecto.

Se debe hacer una actualización del pubspec.yaml después de insertar una nueva dependencia, de esta manera se evita errores de compilación a futuro, asegurando que la dependencia se agregó con éxito. A continuación, se describen cada una de las dependencias usadas en la aplicación móvil:

Cupertino Icons. Esta dependencia, como se muestra en la figura 16 de la línea 4, permitió dar una mejor apariencia a la interfaz de usuario, agregando íconos que identificaron ciertas zonas de la aplicación móvil.

Sensors. Esta dependencia, como se muestra en la figura 16 de la línea 9, se usó para activar un complemento de Flutter orientado a el uso de sensores, tales como el giroscopio y acelerómetro.

Geolocator. Esta dependencia, como se muestra en la figura 16 de la línea 10, es un complemento de Flutter que ofrece servicios de geolocalización.

Shared Preferences. Esta dependencia, como se muestra en la figura 16 de la línea 11, tuvo como finalidad almacenar datos simples de forma asíncrona para usarlos posteriormente.

SQFLite. Esta dependencia, como se muestra en la figura 16 de la línea 12, es un complemento orientado a el uso de la base de datos interna del dispositivo móvil.

Path Provider. Esta dependencia, como se muestra en la figura 16 de la línea 13, permitió identificar la ubicación en donde se creó la base de datos interna.

Cloud Firestore. Esta dependencia, como se muestra en la figura 16 de la línea 14, permitió hacer uso de la API de Firebase para acceder al servicio de almacenamiento en la nube que ofrece Google.

Flutter Offline. Esta dependencia, como se muestra en la figura 16 de la línea 16, permitió a la aplicación detectar el estado de conectividad fuera de línea o en línea, que fue de utilidad para saber en qué base de datos se almacenaran los registros.

Flutter Launcher Icons. Esta dependencia, como se muestra en la figura 16 de la línea 18, permitió cambiar el ícono por defecto de la aplicación a un ícono personalizado.

Figura 16

Código del archivo pubspec.yaml

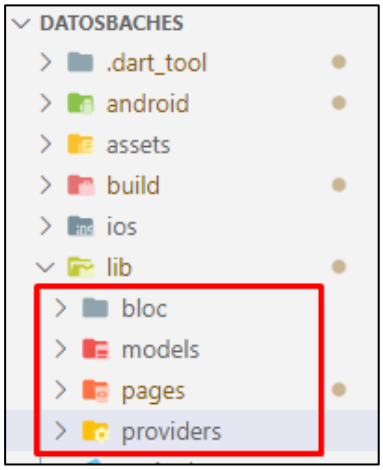
```
1  dependencies:
2    flutter:
3      sdk: flutter
4    cupertino_icons: ^1.0.0
5    rxdart: ^0.25.0
6    google_sign_in: ^4.5.6
7    flutter_signin_button: ^1.1.0
8    get: ^3.24.0
9    sensors: ^0.4.0+1
10   geolocator: ^5.1.5
11   shared_preferences: ^0.5.11
12   sqflite: ^1.3.2+1
13   path_provider: ^1.6.24
14   cloud_firestore: ^0.14.4
15   firebase_core: ^0.5.3
16   flutter_offline: ^2.0.0
17   intl: ^0.17.0
18   vflutter_launcher_icons: ^0.8.1
```

Nota. Dependencias usadas en el proyecto. Elaborado por: Los autores.

3.1.1.2.1.3 Paquetes. Es un contenedor que almacena clases, permitiendo a los desarrolladores llevar de forma ordenada el código de un proyecto en desarrollo. En el proyecto datosbaches se usó los siguientes paquetes tal como se muestra en la Figura 17.

Figura 17

Paquetes del proyecto datosbaches



Nota. Paquetes del proyecto datosbaches. Elaborado por: Los autores.

3.1.1.2.1.4 Clases. En la programación orientada a objetos, una clase es una plantilla donde se implementan todas las variables, contenedores, métodos o funciones que se utilizan en el desarrollo de una aplicación. En la tabla 8 se describe las funciones asignadas a cada clase, además en la figura 18 se muestran todas las clases que se crearon para el proyecto datosbaches.

Tabla 8

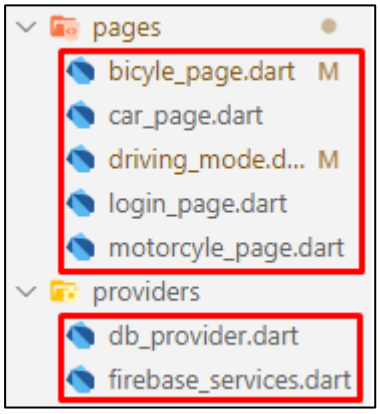
Descripción de clases

NOMBRE DE CLASES	DESCRIPCIÓN
bicycle_page	Permite recolectar los datos relacionados con los posibles desperfectos en las vías en modo bicicleta.
car_page	Permite recolectar los datos relacionados con los posibles desperfectos en las vías en modo carro.
driving_mode	Permite seleccionar el vehículo a conducir.
login_page	Permite acceder a la aplicación con un alias.
motorcycle_page	Permite recolectar los datos relacionados con los posibles desperfectos en las vías en modo motocicleta.
db_provider	Permite almacenar los datos en la BDD interna.
firebase_services	Permite guardar los datos en la BDD externa.

Nota. Se describe la función de cada clase que conforma la aplicación móvil. Elaborado por: Los autores.

Figura 18

Clases del proyecto datosbaches



Nota. Figura que indica las clases de la aplicación móvil. Elaborado por: Los autores.

3.1.1.2.1.5 Contenedores. Son estructuras que permiten llevar a cabo una mejor construcción a la hora de programar, ya que gracias a su construcción en el mundo de Flutter es muy habitual el uso de widgets facilitando así la programación al desarrollador. A continuación, se presentan los widgets creados para la aplicación móvil.

Contenedor de rutas. Este widget se encuentra dentro de la clase main.dart, y contiene el direccionamiento de todas las rutas en las que el usuario podrá navegar a través de la aplicación móvil, tal como se muestra en la figura 19.

Figura 19

Contenedor de rutas de la aplicación móvil

```
Widget build(BuildContext context) {
  return Provider(
    child: MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Sitema de Colección de Datos',
      initialRoute: 'loginpage',
      routes: {
        'loginpage': (BuildContext context) => LoginPage(),
        'drivingpage': (BuildContext context) => DrivePage(),
        'bicyclepage': (BuildContext context) => BicylcePage(),
        'motorcyclepage': (BuildContext context) => MotorcylcePage(),
        'carpage': (BuildContext context) => CarPage(),
        'selectDatosDb': (BuildContext context) => Select_DatosDb(),
      },
      theme: ThemeData(
        primaryColor: Colors.blueAccent,
      ), // ThemeData
    ), // MaterialApp
  ); // Provider
}
```

Nota. Contiene el direccionamiento de las páginas de la aplicación móvil. Elaborado por: Los autores.

Contenedor mensaje alerta. Este componente, tal como se muestra en la figura 20, notifica por medio de una ventana flotante, en el caso que un usuario no haya activado los servicios de ubicación.

Figura 20

Contenedor de notificación de activación de servicios de ubicación

```
Widget _mensajeAlerta(BuildContext context) {
  return AlertDialog(
    title: Text('Notificaciones'),
    content: Text("Active los Servicios de Ubicación"),
    actions: [
      // ignore: deprecated_member_use
      FlatButton(
          child: Text("Aceptar"),
          textColor: Colors.blue,
          onPressed: () {
            Navigator.of context .pop 'aceptar' ;
          }), // FlatButton
      // ignore: deprecated_member_use
      FlatButton(
          child: Text("Cancelar"),
          textColor: Colors.red,
          onPressed: () {
            Navigator.of context .pop 'cancelar' ;
          }), // FlatButton
    ],
  ); // AlertDialog
}
```

Nota. Mensaje de alerta para activar los servicios de ubicación. Elaborado por: Los autores.

3.1.1.2.1.6 Métodos. En la construcción de la aplicación móvil se usaron muchas subrutinas, las mismas que están definidas dentro de cada clase, a continuación, se describe los métodos que se crearon dentro del proyecto datosbaches.

Método obtención de ubicación. Esta subrutina permite habilitar el servicio de geolocalización cada vez que se lo requiera.

Figura 21

Método de obtención de ubicación

```
void _getCurrentLocation(BuildContext context) async {
  serviceEnabled = await Geolocator().isLocationServiceEnabled();
  if (!serviceEnabled) {
    return showDialog(
      context: context,
      builder: (_) => _mensajeAlerta(context),
    );
  } else {
    final position = await Geolocator().getCurrentPosition(
        desiredAccuracy: LocationAccuracy.bestForNavigation);
    print(position);
  }
}
```

Nota. Método que obtiene las coordenadas de ubicación. Elaborado por: Los autores.

Método SQLite a Cloud Firestore. Esta subrutina que se muestra en la figura 22, se ha creado para migrar los registros que están en la base interna del dispositivo móvil a la base externa.

Figura 22

Método inserción a base de datos externa desde base de datos interna

```
FirebaseServices.dbf.addData(
  this.noteList g .user.trim  ,
  this.noteList g .email.trim  ,
  this.noteList g .vehicle.trim  ,
  this.noteList g .date.trim  ,
  this.noteList g .hour.trim  ,
  double.parse this.noteList[g].ejeX.toString() ,
  double.parse this.noteList[g].ejeY.toString() ,
  double.parse this.noteList[g].ejeZ.toString() ,
  double.parse this.noteList[g].lat.toString() ,
  double.parse this.noteList[g].long.toString() );
```

Nota. Inserta en Cloud Firestore los registros almacenados temporalmente el SQFLite. Elaborado por: Los autores.

Método iniciar acelerómetro. Esta subrutina como se muestra en la figura 23, se ha creado con la finalidad de permitir iniciar el uso del acelerómetro en el dispositivo móvil.

Figura 23

Método iniciar acelerómetro

```
void _startTimer() {
  if (accel == null) {
    accel = accelerometerEvents.listen((AccelerometerEvent eve) {
      setState(() {
        event = eve;
      });
    });
  } else {
    accel.resume();
  }
  if (timer == null || !timer.isActive) {
    timer = Timer.periodic(Duration(milliseconds: 200), (_) {
      _setColor(event);
    }); // Timer.periodic
  }
}
```

Nota. Método que activa el acelerómetro. Elaborado por: Los autores.

Método verificar internet. Esta subrutina que se muestra en la figura 24, se ha creado para poder identificar cuando el usuario tiene acceso o no a internet mientras use la aplicación, para enviar los registros a la base interna del dispositivo móvil o a la nube.

Figura 24

Método verificar internet

```
void _checkInternet() async {
  try {
    final result = await InternetAddress.lookup('google.com');
    if (result.isNotEmpty && result[0].rawAddress.isNotEmpty) {
      setState(() {
        //INSERT FIREBASE
        FirebaseServices.dbf.addData ···
      });
    }
  } on SocketException catch (_) {
    setState(() {
      ///Insert en la DB interna del dispositivo
      final tempCollect = new CollectModel(···
      DBProvider.db.nuevoCollect(tempCollect);
    });
  }
}
```

Nota. Método que verifica la conectividad a internet. Elaborado por: Los autores.

Método iniciar recolección. Esta subrutina tal como se muestra en la figura 25, es la más importante dentro del aplicativo móvil ya que se puede considerar como el núcleo de la aplicación, en este apartado es donde se ejecuta la recolección de los datos gracias al uso de los recursos del hardware del dispositivo móvil tales como el acelerómetro y el GPS.

Con el ajuste de la sensibilidad para el acelerómetro se definió los rangos para los ejes “X, Y, Z” tal como se muestra en el Anexo 1, lo cual permite capturar y registrar los datos de un evento considerado como un posible desperfecto en la vía mientras el usuario conduce un determinado tipo de vehículo.

Figura 25

Método iniciar recolección

```
void _setColor(AccelerometerEvent event) {
  String xs = (event?.x ?? 0).toStringAsFixed(3);
  String ys = (event?.y ?? 0).toStringAsFixed(3);
  String zs = (event?.z ?? 0).toStringAsFixed(3);

  double x = double.parse(xs);
  double y = double.parse(ys);
  double z = double.parse(zs);

  //DEFINICION DE RANGOS PARA BICICLETA
  if (x >= 3.8 || y >= 11.7 || z >= 8.3) {
    setState(() {…

    //PARAR
    timer.cancel();
    accel.pause();

    //OBTENER UBICACION GPS
    _getCurrentLocation();
    //SETEO VARIABLES A CERO

    //REANUDO ACELEROMETRO
    _counter = 3;
    _timeres = Timer.periodic(Duration(seconds: 1), (timer) {
      setState(    …
    }); // Timer.periodic
  } else {
    setState(() {…
    i = 0;
  }
}
```

Nota. Método que inicia la recolección de datos. Elaborado por: Los autores.

3.1.1.2.1.7 Interfaces. La aplicación móvil cuenta con tres interfaces de usuario, a continuación, se presenta cada una de las interfaces.

Interface de ingreso de alias. En este apartado se presenta la GUI de ingreso de alias, como se muestra en la figura 26.

Figura 26

Interface de ingreso de alias

Nota. Pantalla de inicio. Elaborado por: Los autores.

Interface de selección de vehículo. Esta GUI permite que se pueda seleccionar el vehículo que el conductor va a usar, tal como se muestra en la figura 27.

Figura 27

Interface de selección de vehículo

Nota. Pantalla Selección de Vehículo. Elaborado por: Los autores.

Interface de detector de movimiento. En esta GUI que se muestra en la figura 28, el usuario no interactúa con la aplicación, debido a que esta interface es solo para la visualización del detector de movimientos.

Figura 28

Interface de detector de movimiento

Nota. Pantalla de recolección de datos. Elaborado por: Los autores.

3.1.1.2.2 Aplicación web. En esta sección, se presenta únicamente, el código que conforma la aplicación web.

3.1.1.2.2.1 Clases. A continuación, en la tabla 9 se describe las funciones asignadas a cada clase, además en la figura 29, se presentan las clases del apartado web de este proyecto.

Tabla 9

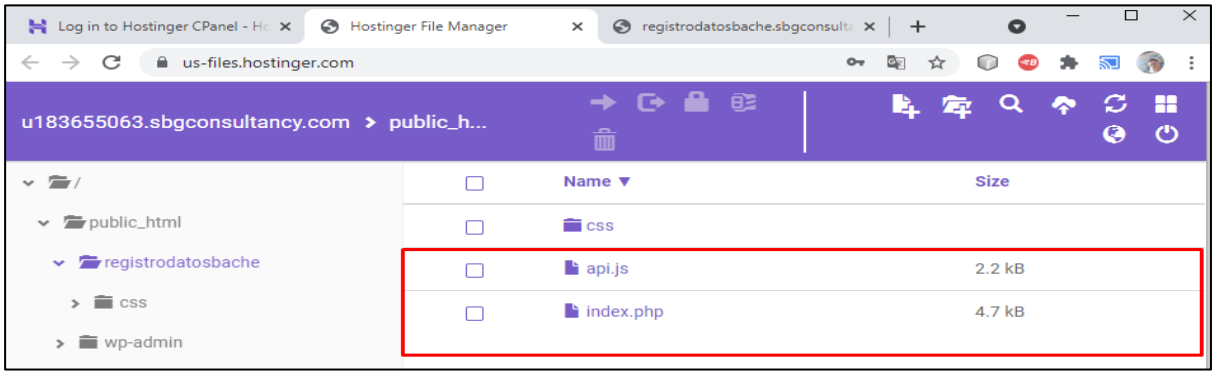
Descripción de clases

NOMBRE DE CLASE	DESCRICIÓN
index	En este apartado se presenta la tabla con los datos registrados en la BDD, además permite descargarlos en formato xls.
api	Aquí se consume los servicios de Cloud Firestore para obtener todos los registros almacenados en la BDD.

Nota. Se describe la función de cada clase de la aplicación web. Elaborado por: Los autores.

Figura 29

Clases de la aplicación web



Nota. Figura que muestra las clases usadas para la aplicación web. Elaborado por: Los autores.

3.1.1.2.2.2 Métodos. Los métodos, son los encargados de proveer funciones a las aplicaciones, además, contienen determinadas instrucciones a ejecutar para poder lograr un objetivo.

Método inicializar conexión a Firebase. Este método, tal como se muestra en la figura 30, realiza la conexión con Firebase, mediante una APIKey proporcionada por Cloud Firestore, además se especificó el Id del proyecto y su URL de identificación.

Figura 30

Método de conexión con Firebase

```
firebase.initializeApp({
    apiKey: 'AAAAymZmafg:APA91bFKcVWsPCJorEt7x8OcSOJrGhGQhGmrOakBZPtSYYcvJG5Ow_qnKpsOLFaurnQ4KP0e
        -pe3_ZlYp2jBSzRJpP_YfYBUD3V2esnUQpg65Lc9r2k06vrJgNdvlQNFeyofte0iNWOi',
    projectId: 'datacollect-42251',
    databaseURL: "https://datacollect-42251.firebaseio.com",
});
```

Nota. Método que conecta con la base de datos en Cloud Firestore. Elaborado por: Los autores.

Método consultar datos. La función de este método, como se muestra en la figura 31, es traer todos los registros almacenados en Firebase, para posteriormente, listaros en una tabla de forma ordenada.

Figura 31

Método de consulta de datos

```
var db = firebase.firestore();
let tabla = document.querySelector('#mitabla tbody');
db.collection("datos")
    .get()
    .then(function(querySnapshot) {
        querySnapshot.forEach(function(doc) {
            let dato = (doc.id, "=>" ,doc.data());
                tabla.innerHTML += `<tr>
                    <td><h2>${dato.vehicle}</h2></td>
                    <td><h2>${dato.date}</h2></td>
                    <td><h2>${dato.hour}</h2></td>
                    <td><h2>${dato.lat}</h2></td>
                    <td><h2>${dato.long}</h2></td>
                    <td><h2>${dato.ejeX}</h2></td>
                    <td><h2>${dato.ejeY}</h2></td>
                    <td><h2>${dato.ejeZ}</h2></td>
                </tr>`;

        });
    })
    .catch(function(error) {
        console.log("Error getting documents: ", error);
    });
```

Nota. Lee los registros de Cloud Firestore. Elaborado por: Los autores.

Método descargar Excel. Este método, tal como se muestra en la figura 32, tiene como finalidad, permitir al usuario descargar todos los datos listados en formato Excel.

Figura 32

Método de descarga de archivo Excel

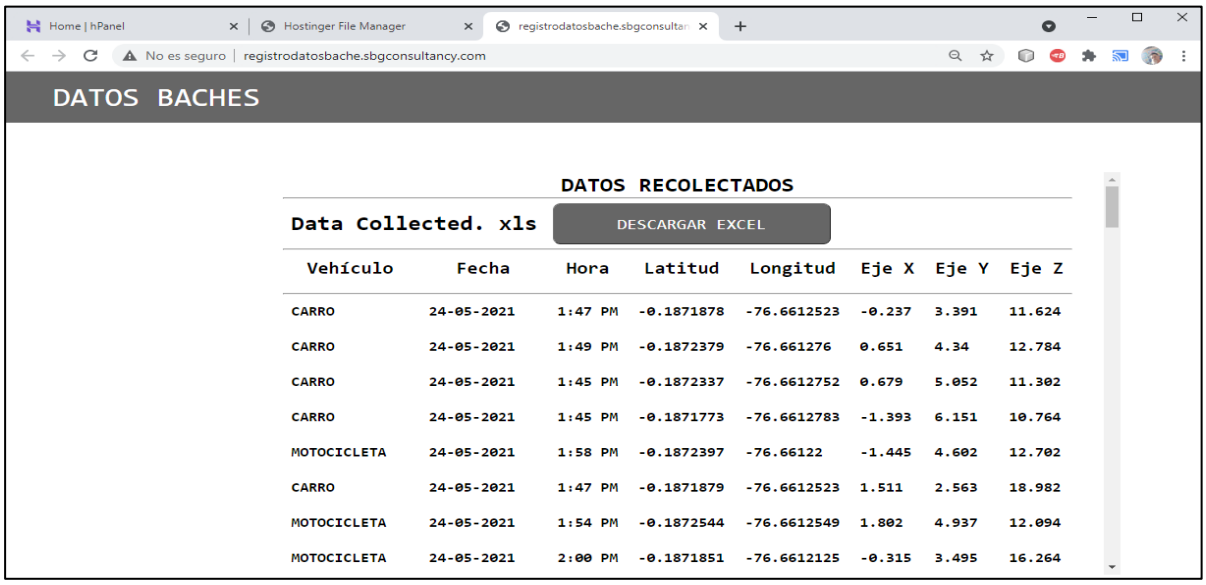
```
<script >
    function exportTableToExcel(tableID, filename = ''){
        var downloadLink;
        var dataType = 'application/vnd.ms-excel';
        var tableSelect = document.getElementById(tableID);
        var tableHTML = tableSelect.outerHTML.replace(/ /g, '%20');
        filename = filename?filename+'.xls':'DataCollected.xls';
        downloadLink = document.createElement("a");
        document.body.appendChild(downloadLink);
        if(navigator.msSaveOrOpenBlob){
            var blob = new Blob(['ufeff', tableHTML], {
                type: dataType
            });
            navigator.msSaveOrOpenBlob( blob, filename);
        }else{
            downloadLink.href = 'data:' + dataType + ', ' + tableHTML;
            downloadLink.download = filename;
            downloadLink.click();
        }
    }
</script>
```

Nota. Descarga los datos en Excel. Elaborado por: Los autores.

3.1.1.2.2.3 Interface. Aquí se visualiza los datos recolectados, también se puede descargar los datos en un archivo Excel, tal como se muestra en la figura 33.

Figura 33

Interface de la aplicación web



Nota. Aplicación Web. Elaborado por: Los autores.

Para interpretar los datos que se muestran a través del entorno web se describe una breve definición de cada campo.

Vehículo: Este campo permite diferenciar el tipo de vehículo que ha seleccionado el conductor.

Fecha: Es el registro de la fecha en la que se registró el evento mientras el usuario conducía.

Hora: Es el registro de la hora en la que se registró el evento mientras el usuario conducía.

Latitud y Longitud: Estos pertenecen a los puntos de geolocalización que señalan donde se identificó un posible desperfecto en la calzada.

Eje X: Este campo permite registrar eventos relacionados con la evasión de obstáculos en las vías.

Eje Y: Este campo permite registrar eventos relacionados con obstáculos cuando el conductor los sobrepasa.

Eje Z: En este campo se registra datos cuando el conductor frena de manera repentina.

3.1.2 Pruebas

En esta sección, se presenta las pruebas realizadas a la aplicación móvil y la aplicación web haciendo énfasis en la funcionalidad, fiabilidad, eficiencia y usabilidad que se tomó como referencia los atributos de la norma ISO 9126, en la cual se ejecutó un plan de pruebas que se encuentra en el Anexo 2, con su respectiva matriz de pruebas ubicada en el Anexo 3.

3.1.2.1 Evaluación de funcionalidad. Con la ejecución de las pruebas de funcionalidad, se establece el nivel de cumplimiento que tiene la aplicación en cuanto a los requerimientos funcionales.

3.1.2.1.1 Aplicación móvil. En la tabla 10, se detalla el proceso de evaluación de las pruebas de funcionalidad a través de acciones para el entorno de la aplicación móvil, pudiendo así, verificar mediante el estado si el requerimiento se ha completado o no.

Tabla 10

Pruebas de funcionalidad

PRUEBA	ACCIONES	ESTADO	OBERVACIONES
Ingreso de alias	- Ingresar a la aplicación móvil - Completar el formulario ingresando el alias y el correo electrónico - Pulsar en el botón "Ingresar"	Completado	- Para el ingreso del alias se debe al menos ingresar 4 caracteres, excepto los caracteres especiales, donde el primer carácter debes ser una letra mayúscula y que contiene un tamaño de 4 a 10 caracteres. - Se deben completar los campos del formulario para habilitar el botón de ingreso
Comprobación de sensores	- El usuario deberá permitir el acceso al uso de los sensores tanto del acelerómetro como el GPS. - Seleccionar tipo de vehículo.	Completado	- Si el usuario no concede los permisos al ingresar por primera vez a la aplicación la misma volverá a solicitar los permisos al seleccionar el tipo de vehículo.
Inserción en bases de datos	- Después de seleccionar el vehículo a conducir, el conductor ya no interactuará con la aplicación para que pueda conducir mientras se recogen los datos de forma automática.	Completado	- En esta sección los datos son recolectados de forma automática. - Si el usuario tiene acceso a internet, los datos se subirán directo al Cloud Firestore. - Si el usuario no está conectado al internet, los datos se almacenarán en la base interna del celular SQLite y posteriormente cuando tenga acceso al internet se enviarán a la nube y se borraran del dispositivo.

Nota. Prueba de funcionalidad en la aplicación móvil. Elaborado por: Los autores.

3.1.2.1.2 Aplicación web. En la tabla 11, se detalla el proceso de evaluación de los requerimientos de funcionalidad a través de acciones para el entorno de la aplicación web, determinando de esta manera si el requerimiento se ha completado o no.

Tabla 11

Evaluación de funcionalidad, aplicación web

PRUEBA	ACCIONES	ESTADO	OBERVACIONES
Verificación de reportes	- Acceder a la aplicación web para visualizar los datos recogidos.	Completado	- El usuario debe ingresar al siguiente subdominio: http://registrodatosbache.sbgconsultancy.com/, para visualizar los datos.
Manejo de reportes	- Descargar reporte al seleccionar el botón "Descargar Excel".	Completado	- El usuario debe ingresar al siguiente subdominio: http://registrodatosbache.sbgconsultancy.com/, para descargar los datos.

Nota. Prueba de funcionalidad, parte web. Elaborado por: Los autores.

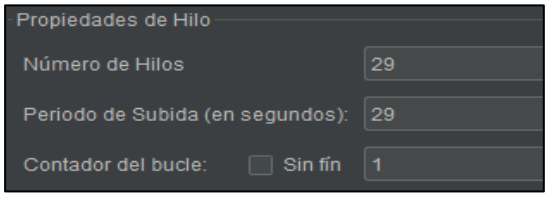
3.1.2.2 Evaluación de fiabilidad. Para medir la fiabilidad fue necesario tomar en cuenta varias subcaracterísticas entre las más importantes están la tolerancia a fallos, la recuperabilidad, y la conformidad de fiabilidad.

3.1.2.2.1 Prueba de carga en la aplicación web. En esta sección se midió la fiabilidad a través de pruebas de carga mediante la extensión BlazeMeter, para determinar la probabilidad de un correcto funcionamiento del entorno web.

3.1.2.2.1.1 Escenario 1. En este tipo de prueba se realizó una simulación a través de JMeter donde accedieron 29 usuarios consecutivos a la aplicación web, tal como se muestra en la figura 34.

Figura 34

Configuración de 29 usuarios para pruebas de carga consecutiva



Nota. Configuración, prueba de carga. Fuente: (Software JMeter, 2021).

3.1.2.2.1.2 Resultado de las pruebas de carga. En la tabla 12, se muestra que el tiempo mínimo de respuesta es de 0.556 segundos, mientras que el tiempo máximo es de 2.726 segundos y en promedio el tiempo que se demora el apartado web en presentar los datos al usuario es de 0.898 segundos. Se puede apreciar, que se transmiten 15.73 Kb por segundo y la media de Bytes es de 6253.1 por segundo.

Tabla 12

Resultados de las pruebas de carga

# MUESTRAS	MEDIA ms	MIN ms	MAX ms	Kb/seg	MEDIA DE BYTES
29	898	556	2726	15.73	6253.1

Nota. Tabla de resultados de la simulación del escenario 1. Elaborado por: Los autores.

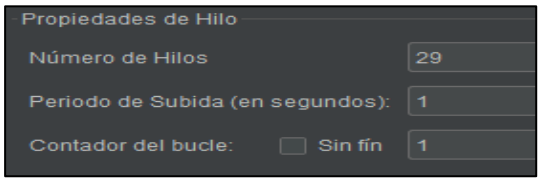
3.1.2.3 Evaluación de eficiencia. En esta sección se midió la eficiencia a través de pruebas de estrés con BlazeMeter para el entorno web, al mismo tiempo para el entorno móvil se realizó pruebas de rendimiento.

3.1.2.3.1 Prueba de estrés en la aplicación web. En este apartado se midió la eficiencia a través de pruebas de estrés mediante la extensión BlazeMeter, para determinar la probabilidad de un correcto funcionamiento del entorno web.

3.1.2.3.1.1 Escenario 2. En este tipo de prueba se realizó una simulación a través de JMeter donde accedieron 29 usuarios en un segundo a la aplicación web, tal como se muestra en la figura 35.

Figura 35

Configuración de 29 usuarios por segundo para pruebas de estrés



Nota. Configuración, prueba de estrés. Fuente: (Software JMeter, 2021).

3.1.2.3.1.2 Resultado de las pruebas de estrés. En la tabla 13, se muestra que el tiempo mínimo de respuesta es de 0.3034 segundos, mientras que el tiempo máximo es de 6.871 segundos, en promedio el tiempo que se demora el apartado web en presentar los datos al usuario es de 5.315 segundos. Por consiguiente, se pudo apreciar que se transmiten 14.27 Kb por segundo y la media de Bytes es de 2876.2 por segundo.

3.1.2.3.2 Análisis de pruebas de carga y estrés. En la ingeniería de software es muy habitual realizar este tipo de pruebas a los sistemas informáticos, de manera que, para la prueba de carga se midió el rendimiento del aplicativo web con una carga de 29 usuarios en 29 segundos, tal como se muestra en la figura 34, a diferencia de la prueba de estrés, que se midió el esfuerzo de sobrecarga del aplicativo web con un número de 29 usuarios por segundo tal como se muestra en la figura 35.

Tabla 13

Resultados de las pruebas de estrés

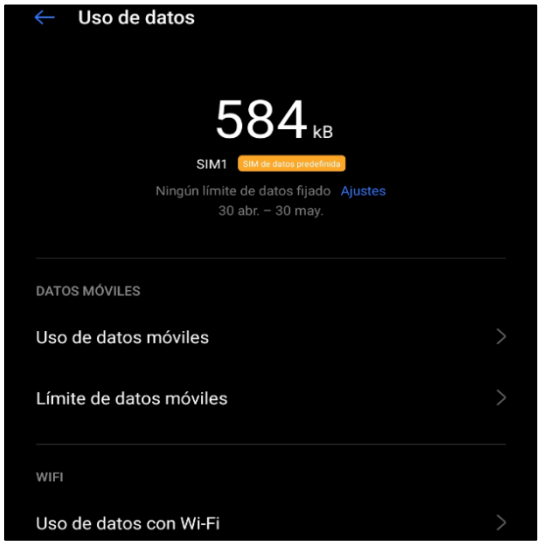
# MUESTRAS	MEDIA ms	MIN ms	MAX ms	Kb/seg	MEDIA DE BYTES
29	5315	3034	6871	14.27	2876.2

Nota. Tabla de resultados de la simulación del escenario 2. Elaborado por: Los autores.

3.1.2.4 Evaluación del rendimiento en la aplicación móvil. Para esta evaluación dentro del apartado móvil se realizó pruebas de rendimiento en donde se verificó el consumo de datos tanto en el uso de Wi-Fi como el de datos móviles, tal como se muestra en la figura 36.

Figura 36

Herramienta para medir el uso de datos de una aplicación móvil



Nota. Medir uso de datos. Elaborado por: Los autores.

3.1.2.4.1 ***Escenario 3.*** En este escenario se simuló el registro de 1 evento y se verificó el consumo de datos tal como se muestra en la figura 37, para esto se hizo uso del Wi-Fi y a través de la administración de aplicaciones se pudo obtener los resultados del uso de datos mientras la aplicación estuvo en ejecución.

Figura 37

Registro de 1 evento con la aplicación Recolector de Datos

Nota. Uso de datos con un evento. Elaborado por: Los autores.

3.1.2.4.2 ***Escenario 4.*** En este escenario se simuló el registro de 10 eventos y se verificó el consumo de datos tal como se muestra en la figura 38.

Figura 38

Registro de 10 eventos con la aplicación Recolector de Datos

Nota. Uso de datos con 10 eventos. Elaborado por: Los autores.

3.1.2.4.3 Escenario 5. En este escenario se simuló el registro de 50 eventos y se verificó el consumo de datos tal como se muestra en la figura 39.

Figura 39

Registro de 50 eventos con la aplicación Recolector de Datos

Nota. Uso de datos con 50 eventos. Elaborado por: Los autores.

3.1.2.4.4 Escenario 6. En este escenario se simuló el registro de 100 eventos y se verificó el consumo de datos tal como se muestra en la figura 40.

Figura 40

Registro de 100 eventos con la aplicación Recolector de Datos

Nota. Uso de datos con 100 eventos. Elaborado por: Los autores.

3.1.2.4.5 Resultados del uso de datos en el dispositivo móvil a través de Wi-Fi. La recolección de datos para esta prueba arrojó resultados favorables tal como se muestra en la tabla 14.

Tabla 14

Resultados del uso de datos en el dispositivo móvil a través de Wi-Fi

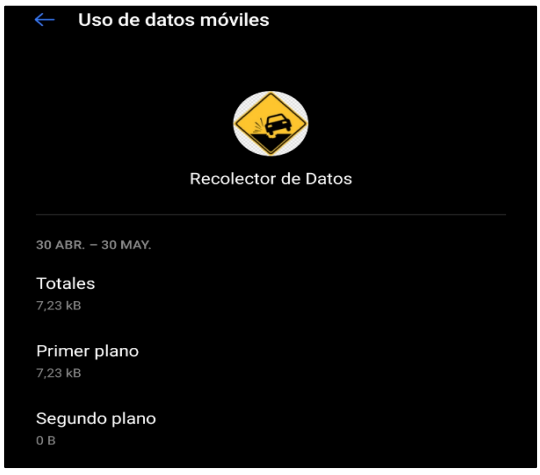
Nº PRUEBA	Nº REGISTROS	PRIMER PLANO	SEGUNDO PLANO	TOTAL
1	1	22.74 kB	380 B	23.11 kB
2	10	30.68 kB	1.74 kB	32.42 kB
3	50	53.37 kB	3.18 kB	56.55 kB
4	100	99.08 kB	4.53 kB	104 kB

Nota. Resultados consumo de datos. Elaborado por: Los autores.

3.1.2.4.6 Escenario 7. En este escenario se simuló el registro de 1 evento, verificando así el consumo de datos tal como se muestra en la figura 41, para esto se hizo uso de los datos móviles y a través de la administración de aplicaciones se pudo obtener los resultados mientras la aplicación estuvo en ejecución.

Figura 41

Registro de 1 evento con la aplicación Recolector de Datos

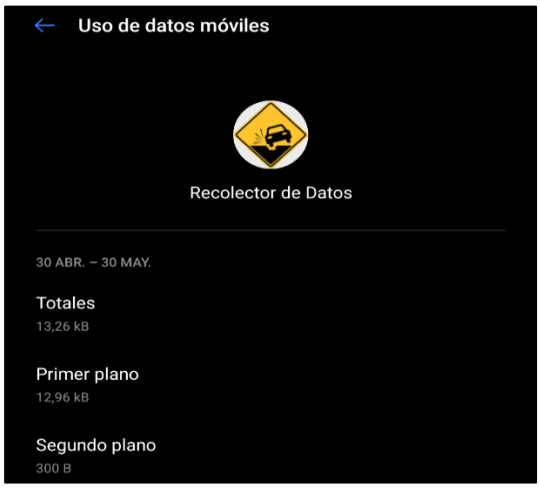


Nota. Uso de datos móviles con un evento. Elaborado por: Los autores.

3.1.2.4.7 Escenario 8. En este escenario se simuló el registro de 10 eventos y se verificó el consumo de datos tal como se muestra en la figura 42.

Figura 42

Registro de 10 eventos con la aplicación Recolector de Datos

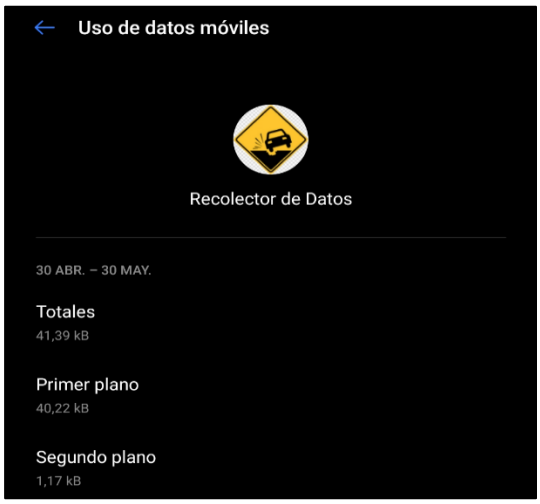


Nota. Uso de datos móviles con 10 eventos. Elaborado por: Los autores.

3.1.2.4.8 Escenario 9. En este escenario se simuló el registro de 50 eventos y se verificó el consumo de datos tal como se muestran en la figura 43.

Figura 43

Registro de 50 eventos con la aplicación Recolector de Datos

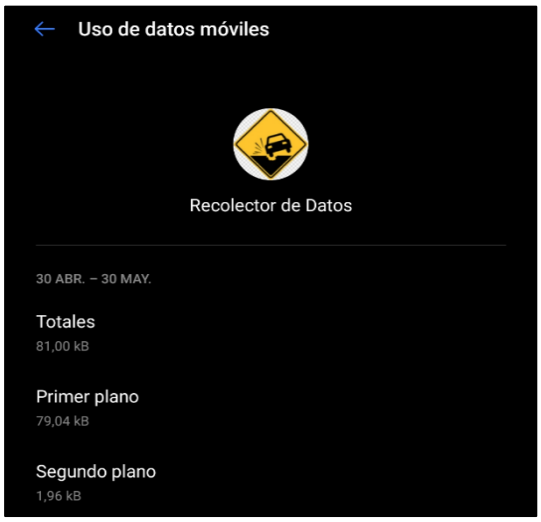


Nota. Uso de datos móviles con 50 eventos. Elaborado por: Los autores.

3.1.2.4.9 Escenario 10. En este escenario se simuló el registro de 100 eventos y se verificó el consumo de datos tal como se muestran en la figura 44.

Figura 44

Registro de 100 eventos con la aplicación Recolector de Datos



Nota. Uso de datos móviles con 100 eventos. Elaborado por: Los autores.

3.1.2.4.10 Resultados del uso de datos en el dispositivo móvil a través de datos móviles. La recolección de datos para esta prueba entregó resultados favorables tal como se muestra en la tabla 15.

Tabla 15

Resultados del uso de datos en el dispositivo móvil a través de datos móviles

Nº PRUEBA	Nº REGISTROS	PRIMER PLANO	SEGUNDO PLANO	TOTAL
1	1	7.23 kB	0 B	7.23 kB
2	10	12.96 kB	300 B	13.26 kB
3	50	40.22 kB	1.17 kB	41.39 kB
4	100	79.04 kB	1.96 kB	81.00 kB

Nota. Resultados, consumo de datos móviles. Elaborado por: Los autores.

3.1.2.4.11 Análisis de resultados. Las pruebas que se realizaron en el aplicativo móvil indican un rendimiento aceptable en cuanto al consumo de datos, permitiendo así a la aplicación, capturar y enviar registros sin inconvenientes debido a la flexibilidad que entrega el entorno donde se desarrolló la aplicación móvil, ya que gracias a los manejadores de estados se logró optimizar los recursos. Cabe mencionar que la aplicación usa algunos componentes sin la necesidad de tener conectividad a internet para así ahorrar consumo de datos. En las tablas 14 y 15, se hace una comparativa entre el consumo de datos con Wi-Fi y datos móviles, donde se puede ver un ligero mayor consumo de datos en las pruebas con Wi-Fi, esto se debe porque al registrar los datos vía Wi-fi, se hizo por primera vez la petición con la base de Cloud Firestore para insertar los datos y una vez registrado el primer dato, se pudo percibir el Real Time de Firebase al estar conectados a internet registrando eventos con la aplicación móvil.

3.1.2.5 Evaluación de usabilidad. En esta sección, primero se estableció el tamaño de la muestra mediante la fórmula de la muestra finita, permitiendo de esta manera seleccionar a un determinado número de personas, con las que se trabajó en toda la fase de evaluación de usabilidad.

3.1.2.5.1 Cálculo de la muestra. Para obtener el dato numérico sobre la población que se trabajó, se verificó en el registro de las matrículas vehiculares. Según el INEC en el año 2019 se matricularon 2.311.960 vehículos, este valor se tomó para el cálculo de la muestra que se indica en la figura 45.

Figura 45

Cálculo de la muestra

Parametro	Valor
N	2.311.960
Z	1,960
P	90,00%
Q	10,00%
e	12,00%

Tamaño de muestra

n= ***24,01***

$$n = \frac{N * Z_\alpha^2 * p * q}{e^2 * (N-1) + Z_\alpha^2 * p * q}$$

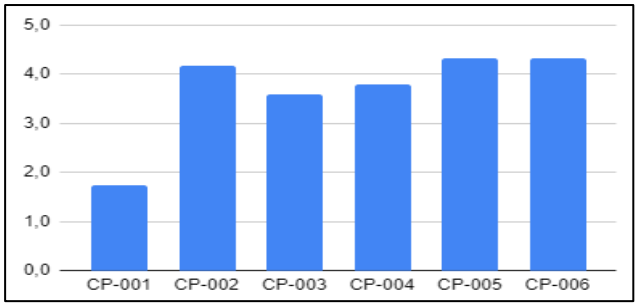
n = Tamaño de muestra buscado
N = Tamaño de la Población o Universo
Z = Parámetro estadístico que depende el Nivel de Confianza
e = Erro de estimación máximo aceptado
p = Probabilidad de que ocurra el evento estudiado (éxito)
q = $(1 - p)$ = Probabilidad de que no ocurra el evento estudiado

Nota. Cálculo de muestra para las encuestas. Elaborado por: Los autores.

3.1.2.5.2 Resultados de la encuesta. Para obtener estos resultados, los usuarios del muestreo hicieron una evaluación de las aplicaciones móvil y web a través de una encuesta que se les proporcionó, permitiendo así cuantificar la usabilidad de las aplicaciones tal como se muestra en la figura 46, en dónde la escala va del 1 al 5, siendo 1 el valor más bajo y 5 el valor más alto en cuanto a usabilidad.

Figura 46

Resultados de la encuesta



Nota. Las etiquetas horizontales son los códigos de cada pregunta en la encuesta que se encuentra en Anexo 4 Encuesta de usabilidad. Elaborado por: Los autores.

3.1.2.5.3 Análisis de resultados. Las pruebas que se realizaron a los usuarios permitieron obtener un promedio en general para determinar la usabilidad de la aplicación, dando como resultado positivo de 3,7 sobre 5 tal como se muestra en la tabla 16, superando la meta propuesta de 2,5 en la matriz de pruebas, ubicada en el Anexo 3, esto significa que la integración de diseños minimalistas en las aplicaciones permitió simplificar las interacciones y dar una mejor experiencia al usuario.

Tabla 16

Resultados de la encuesta de usabilidad

CP-001	CP-002	CP-003	CP-004	CP-005	CP-006	PROMEDIO
1,8	4,2	3,6	3,8	4,3	4,3	3,7

Nota. Los códigos usados en esta tabla pertenecen a cada pregunta de la encuesta que se encuentra en Anexo 5 Resultados estadísticos de encuesta por cada pregunta. Elaborado por: Los autores.

CONCLUSIONES

- Con la construcción de las aplicaciones, se determinó que la automatización en la recolección de los datos es alta, es decir que el usuario interactúa de forma mínima gracias a la adecuada gestión de los recursos del hardware de los dispositivos móviles inteligentes.
- La comprensión de las variables que intervienen en el proceso de recolección de datos permitió definir funciones específicas para determinadas tareas tales como: la selección del vehículo a conducir y la conectividad del dispositivo móvil que identifica la inserción de los registros en la BDD.
- El uso del marco de trabajo Scrum a pequeña escala, es una variante de la metodología ágil original, por consiguiente, permitió la asignación de múltiples roles a los miembros que integraron este proyecto técnico, ya que este modelo se enfoca para grupos de trabajo de máximo tres personas.
- Para el desarrollo de la aplicación móvil, se utilizó el Framework Flutter, debido a que es un entorno nuevo de desarrollo incluido por Google, con un enfoque en la programación orientada a objetos, entregando como resultado interfaces amigables para el usuario, como se pudo evidenciar en las pruebas de usabilidad realizadas a la aplicación.

RECOMENDACIONES

- En una nueva versión de la aplicación móvil se recomienda la revisión y actualización de las dependencias en la documentación de Flutter, con la finalidad de evitar errores durante la compilación del proyecto.
- Para futuras versiones de la aplicación móvil se sugiere hacer uso de la integración multiplataforma para diferentes entornos como Windows Phone e iOS, con el objetivo de que el aplicativo no solo se limite para usuarios con dispositivos móviles Android.
- Se recomienda realizar un estudio sobre Machine Learning para implementar un algoritmo que ayude a mejorar la precisión en la recolección de los datos relacionados con las imperfecciones en las vías.
- Por último, se recomienda integrar un módulo de geolocalización dentro de la aplicación móvil, en la cual se le notifique al conductor la presencia de posibles desperfectos en la ruta de conducción.

GLOSARIO DE TÉRMINOS

GPS: Sistema de Posicionamiento Global

DM: Distrito Metropolitano

JS: JavaScript

PHP: Procesador de Hipertexto

IDE: Entorno de Desarrollo Integrado

AVD: Dispositivo Virtual de Android

MSP: Microsoft Project

UX: Experiencia de Usuario

MVC: Modelo Vista Controlador

SQL: Lenguaje Estructurado de Consultas

BDD: Base de Datos

GUI: Interfaz Gráfica de Usuario

LISTA DE REFERENCIAS

Páginas Web

Agnieszka Gancarczyk. (2019). *opensource.com*. Obtenido de https://opensource.com/article/19/2/small-scale-scrum-principles

Bravo, D. (2019). *El Comercio*. Obtenido de https://www.elcomercio.com/actualidad/quito-solanda-bache-transito-barrios.html

Código de Visual Studio . (2021). *Código de Visual Studio* . Obtenido de https://code.visualstudio.com/docs

Deemer, P. Benefield, G. Larman, C. Vodde, B. . (2009). *Información Básica de Scrum Versión 1.1*. Obtenido de http://www.goodagile.com/scrumprimer/scrumprimer_es.pdf

Developers . (2021). *developers* . Obtenido de https://developer.android.com/studio/intro?hl=es-419

Diví, V. (2020). *inLab FIB* . Obtenido de https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart

El Comercio . (2019). *Parque automotor de Ecuador creció en 1,4 millones de vehículos en una décadav*. Obtenido de https://www.elcomercio.com/actualidad/parque-automotor-ecuador-crecimiento-decada.html

Fernández, Y. (2019). *XATAKA Basics*. Obtenido de https://www.xataka.com/basics/sensores-que-encontraras-tu-movil-como-funcionan-sirven

Fernández, Y. (2019). *XATAKA Basics* . Obtenido de https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores

Firebase . (2021). *Firebase* . Obtenido de https://firebase.google.com/docs/firestore?hl=es#:~:text=Cloud%20Firestore%20es%20una%20base,desde%20Firebase%20y%20Google%20Cloud.&text=Cloud%20Firestore%20tambi%C3%A9n%20ofrece%20una,Google%20Cloud%2C%20incluido%20Cloud%20Functions.

INEC. (2019). *Instituto nacional de estadistica y censos*. Obtenido de https://www.ecuadorencifras.gob.ec/transporte/

Lucidchart . (2021). *Lucidchart* . Obtenido de https://www.lucidchart.com/pages/es/ejemplos/herramienta-de-diseno-de-bases-de-datos

MockFlow. (2019). *MarTech FORUM*. Obtenido de https://www.martechforum.com/herramienta/mockflow/

OINOS. (2020). *Digital Guide OINOS*. Obtenido de https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/lenguaje-de-programacion-dart-de-google/

PC Resumen . (2019). *PC Resumen* . Obtenido de https://www.pcresumen.com/menu-software/25-entornos-de-desarrollo/21-netbeans-ide

Quito Informa. (2020). *Quito Informa* . Obtenido de http://www.quitoinforma.gob.ec/2020/06/17/en-233-vias-se-ha-realizado-mantenimiento-vial/

Rebeca. (2021). *next_u*. Obtenido de https://www.nextu.com/blog/que-es-scrum/

SEAS. (2019). *Blog SEAS*. Obtenido de https://www.seas.es/blog/informatica/conoce-el-lenguaje-de-programacion-java/v

Secretaría de Movilidad . (2014). *Municipio del Distrito Metropolitano de Quito* . Obtenido de http://gobiernoabierto.quito.gob.ec/wp-content/uploads/documentos/pdf/diagnosticomovilidad.pdf

Softtrader . (2020). *Softtrader* . Obtenido de https://softtrader.es/blog-microsoft/que-hay-de-nuevo-en-microsoft-project-2019/

Souza, I. (2019). *rockcontent* . Obtenido de https://rockcontent.com/es/blog/software-libre/

Souza, I. (2020). *rockcontent*. Obtenido de https://rockcontent.com/es/blog/php/

SQLite . (2021). *SQLite* . Obtenido de https://www.sqlite.org/index.html

The World Banck . (2019). *The World Banck* . Obtenido de http://opendatatoolkit.worldbank.org/es/essentials.html

Tesis

Castro, C. Velasco, J. (2018). *Solución informática de acceso ciudadano, para el registro de la ubicación de baches en la ciudad de quito dm, utilizando comandos de voz y geolocalización* . Obtenido de Repositorio Universidad Politecnica Selesiana Sede Quito: https://dspace.ups.edu.ec/bitstream/123456789/19088/1/UPS%20-%20TTS085.pdf

Riera, D. Soria, J. . (2020). *Solución informática de acceso ciudadano, para el registro de la ubicación de baches en la ciudad de Quito DM, utilizando comandos de voz y geolocalización.* Obtenido de Repositorio Universidad Politécnica Selesiana Sede Quito: https://dspace.ups.edu.ec/bitstream/123456789/19088/1/UPS%20-%20TTS085.pdf

Agnieszka Gancarczyk. (2019). *opensource.com*. Obtenido de https://opensource.com/article/19/2/small-scale-scrum-principles

Bravo, D. (2019). *El Comercio*. Obtenido de https://www.elcomercio.com/actualidad/quito-solanda-bache-transito-barrios.html

Castro, C. Velasco, J. (2018). *Solución informática de acceso ciudadano, para el registro de la ubicación de baches en la ciudad de quito dm, utilizando comandos de voz y geolocalización* . Obtenido de Repositorio Universidad Politecnica Selesiana Sede Quito: https://dspace.ups.edu.ec/bitstream/123456789/19088/1/UPS%20-%20TTS085.pdf

Código de Visual Studio . (2021). *Código de Visual Studio* . Obtenido de https://code.visualstudio.com/docs

Deemer, P. Benefield, G. Larman, C. Vodde, B. (2009). *Información Básica de Scrum Versión 1.1.* Obtenido de http://www.goodagile.com/scrumprimer/scrumprimer_es.pdf

Developers. (2021). *developers*. Obtenido de https://developer.android.com/studio/intro?hl=es-419

Diví, V. (2020). *inLab FIB* . Obtenido de https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart

El Comercio . (2019). *Parque automotor de Ecuador creció en 1,4 millones de vehículos en una décadav*. Obtenido de https://www.elcomercio.com/actualidad/parque-automotor-ecuador-crecimiento-decada.html

Fernández, Y. (2019). *XATAKA Basics*. Obtenido de https://www.xataka.com/basics/sensores-que-encontraras-tu-movil-como-funcionan-sirven

Fernández, Y. (2019). *XATAKA Basics* . Obtenido de https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores

Firebase . (2021). *Firebase* . Obtenido de https://firebase.google.com/docs/firestore?hl=es#:~:text=Cloud%20Firestore%20es%20una%20base,desde%20Firebase%20y%20Google%20Cloud.&text=Cloud%20Firestore%20tambi%C3%A9n%20ofrece%20una,Google%20Cloud%2C%20incluido%20Cloud%20Functions.

INEC. (2019). *Instituto nacional de estadistica y censos*. Obtenido de https://www.ecuadorencifras.gob.ec/transporte/

Lucidchart . (2021). *Lucidchart* . Obtenido de https://www.lucidchart.com/pages/es/ejemplos/herramienta-de-diseno-de-bases-de-datos

MockFlow. (2019). *MarTech FORUM*. Obtenido de https://www.martechforum.com/herramienta/mockflow/

OINOS. (2020). *Digital Guide OINOS*. Obtenido de https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/lenguaje-de-programacion-dart-de-google/

PC Resumen . (2019). *PC Resumen* . Obtenido de https://www.pcresumen.com/menu-software/25-entornos-de-desarrollo/21-netbeans-ide

Quito Informa. (2020). *Quito Informa* . Obtenido de http://www.quitoinforma.gob.ec/2020/06/17/en-233-vias-se-ha-realizado-mantenimiento-vial/

Rebeca. (2021). *next_u*. Obtenido de https://www.nextu.com/blog/que-es-scrum/

Riera, D. Soria, J. . (2020). *Solución informática de acceso ciudadano, para el registro de la ubicación de baches en la ciudad de Quito DM, utilizando comandos de voz y geolocalización.* Obtenido de Repositorio Universidad Politécnica Selesiana Sede

Quito: https://dspace.ups.edu.ec/bitstream/123456789/19088/1/UPS%20-%20TTS085.pdf

SEAS. (2019). *Blog SEAS*. Obtenido de https://www.seas.es/blog/informatica/conoce-el-lenguaje-de-programacion-java/v

Secretaría de Movilidad . (2014). *Municipio del Distrito Metropolitano de Quito* . Obtenido de http://gobiernoabierto.quito.gob.ec/wp-content/uploads/documentos/pdf/diagnosticomovilidad.pdf

Softtrader . (2020). *Softtrader* . Obtenido de https://softtrader.es/blog-microsoft/que-hay-de-nuevo-en-microsoft-project-2019/

Souza, I. (2019). *rockcontent* . Obtenido de https://rockcontent.com/es/blog/software-libre/

Souza, I. (2020). *rockcontent*. Obtenido de https://rockcontent.com/es/blog/php/

SQLite . (2021). *SQLite* . Obtenido de https://www.sqlite.org/index.html

The World Banck . (2019). *The World Banck* . Obtenido de http://opendatatoolkit.worldbank.org/es/essentials.html