



UNIVERSIDAD  
CATÓLICA  
DE CUENCA

**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

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

**CARRERA DE INGENIERÍA EN SISTEMAS DE  
INFORMACIÓN**

**DISEÑO DE UN SISTEMA DE INFORMACIÓN SEGURO  
PARA EL RASTREO DE LOS BUSES DE TRANSPORTE  
PÚBLICO DE LA PARROQUIA GUAPÁN MEDIANTE UN  
DISPOSITIVO GPS**

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

**AUTOR: JONNATHAN ISMAEL CRESPO GUILLÉN**

**JACOBO STIVE MINCHALA GONZÁLEZ**

**DIRECTOR: ING. ANDRÉS SEBASTIÁN QUEVEDO SACOTO**

**AZOGUES - ECUADOR**

**2023**

**DIOS, PATRIA, CULTURA Y DESARROLLO**



**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

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

**CARRERA DE INGENIERÍA EN SISTEMAS DE  
INFORMACIÓN**

**DISEÑO DE UN SISTEMA DE INFORMACIÓN SEGURO PARA EL  
RASTREO DE LOS BUSES DE TRANSPORTE PÚBLICO DE LA  
PARROQUIA GUAPÁN MEDIANTE UN DISPOSITIVO GPS  
TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN**

**AUTOR: JONNATHAN ISMAEL CRESPO GUILLÉN**

**JACOBO STIVE MINCHALA GONZÁLEZ**

**DIRECTOR: ING. ANDRÉS SEBASTIÁN QUEVEDO SACOTO**

**AZOGUES - ECUADOR**

**2023**

**DIOS, PATRIA, CULTURA Y DESARROLLO**

**Declaratoria de Autoría y Responsabilidad**

**Jonnathan Ismael Crespo Guillén** portador de la cédula de ciudadanía N° **0302633581**. Declaro ser el autor de la obra: **“Diseño de un sistema de información seguro para el rastreo de los buses de transporte público de la parroquia Guapán mediante un dispositivo GPS”**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Azogues, **08 de marzo de 2023**

F: 

**Jonnathan Ismael Crespo Guillén**

**C.I. 0302633581**

**Declaratoria de Autoría y Responsabilidad**

**Jacobo Stive Minchala González** portador de la cédula de ciudadanía N° **0350007571**. Declaro ser el autor de la obra: **“Diseño de un sistema de información seguro para el rastreo de los buses de transporte público de la parroquia Guapán mediante un dispositivo GPS”**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Azogues, **8 de marzo de 2023**

F: .....



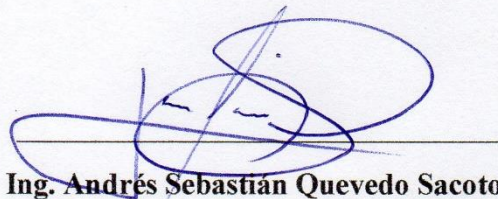
**Jacobo Stive Minchala González**

**C.I. 0350007571**



## **CERTIFICACIÓN DEL TUTOR**

Certifico que el presente trabajo fue desarrollado por los Estudiantes Jonnathan Ismael Crespo Guillén y Jacobo Stive Minchala González, bajo la supervisión del tutor designado (Ing. Andrés Sebastián Quevedo Sacoto); la investigación propuesta sirve como requisito previo a la obtención del título de Ingeniero en Sistemas de Información, el tema "DISEÑO DE UN SISTEMA DE INFORMACIÓN SEGURO PARA EL RASTREO DE LOS BUSES DE TRANSPORTE PÚBLICO DE LA PARROQUIA GUAPAN MEDIANTE UN DISPOSITIVO GPS" cumple con todas las observaciones realizadas por el tribunal evaluador, por lo que las ideas, opiniones vertidas en el presente, son de exclusiva responsabilidad de los autores.



**Ing. Andrés Sebastián Quevedo Sacoto.**

**DIRECTOR**

## DEDICATORIA

Mi tesis va dedicada con mucho amor y cariño a mis abuelos que, aunque ya no están aquí para compartir este momento conmigo, sé que siempre estarán en mi corazón y en mi mente.

Recuerdo con cariño los momentos en que hablaban con orgullo sobre mi futuro y me animaban a perseguir mis sueños. Su amor y apoyo siempre fueron un faro en mi vida, guiándome hacia un camino de éxito y realización personal. Aunque no pudieron verme graduarme y comenzar mi carrera, sé que estarían orgullosos de lo que he logrado.

Cada vez que me sentía desanimado o inseguro, recordaba sus palabras de aliento y sus enseñanzas. Me enseñaron la importancia del trabajo duro, la perseverancia y la dedicación, valores que he llevado conmigo en cada etapa de mi vida académica y profesional. Agradezco por cada uno de los consejos que me dieron y por ser siempre un modelo a seguir.

Me siento afortunado de haber tenido unos abuelos tan amorosos y dedicados. Vuestra presencia en mi vida siempre me inspiró a seguir adelante y a alcanzar mis metas. Sé que me están observando desde el cielo con orgullo y amor, y eso me llena de alegría y de esperanza.

Esta tesis es una manera de honrar su legado y de agradecerles por todo lo que han hecho por mí. Sé que, si no fuera por su amor y apoyo, no habría llegado hasta aquí. Sus enseñanzas y valores me han guiado en cada paso del camino, y espero que este sea el comienzo de muchos éxitos más en mi carrera como ingeniero.

Les agradezco por ser una parte tan importante de mi vida y por haberme inculcado los valores y la pasión por la ingeniería. Siempre estarán en mi corazón y en mi mente, y prometo seguir honrando nuestro legado en todo lo que haga.

Jonnathan Ismael Crespo Guillén

## AGRADECIMIENTO:

En primer lugar, quiero agradecer a Dios quien me guio con un corazón lleno de gratitud y alegría y me acompañado en el camino hacia la culminación de mi tesis.

Quiero agradecer a Guillermo Crespo y Luisa Guillén, mis padres amados quienes, sin su presencia y ayuda, este logro no habría sido posible, de igual forma quiero agradecer a mis hermanos por siempre estar ahí para ayudarme en todo momento.

Hoy, mientras finalizo mi tesis, no puedo evitar sentir un profundo agradecimiento hacia todos ustedes. No sería posible haber llegado hasta aquí sin su amor, apoyo y dedicación.

Quiero reconocer la ayuda inigualable de mis padres y de su amor incondicional, paciencia y sacrificio. Gracias por haber trabajado duro para darme una educación de calidad, y por haberme inculcado valores importantes que me han guiado en mi vida. Sé que no siempre fue fácil, pero nunca dejaron de apoyarme en cada etapa de mi vida académica, incluso cuando tuvieron que hacer sacrificios importantes para ayudarme a alcanzar mis metas. Nunca me cansaré de agradecerles por todo lo que han hecho por mí.

A mis hermanos, gracias por estar a mi lado en cada momento de mi vida. Sus palabras de aliento, su amor y su apoyo han sido invaluable para mí. Saber que tenía su amor y su apoyo me dio la fuerza y la motivación necesarias para seguir adelante incluso en los momentos más difíciles. Agradezco su paciencia y comprensión en los momentos en que tuve que dedicar todo mi tiempo y energía a mi investigación. Gracias por siempre haber estado allí para mí.

Por último, pero no menos importante, quiero agradecer a mis familiares por su amor incondicional y su guía en mi vida. Gracias por darme fuerza en momentos de duda y por haberme guiado por el camino correcto. A través de las muchas pruebas y desafíos que encontré en mi camino, siempre sentí su amor y presencia. Agradezco su guía en cada momento de mi vida, y especialmente en mi camino académico.

Espero que esta tesis sea el comienzo de muchos éxitos y logros en mi vida, y les agradezco a todos por haber sido parte de este importante hito en mi vida académica. Agradezco a Dios por haberme dado la oportunidad de tener una familia tan amorosa y apasionada, y por haberme guiado hacia esta etapa de mi vida. Con todo mi amor y agradecimiento.

Jonnathan Ismael Crespo Guillén



## DEDICATORIA

A mi querida madre Marcela que siempre me apoyo desde que era pequeño, te agradezco por alentarme a seguir adelante inclusive en los momentos mas difíciles. Tu fe en mi siempre me ha mantenido avanzando en este camino académico, este logro no es solo mío, si no tuyo y espero que te haga sentir orgullosa de mí como yo lo estoy de ser tu hijo.

Jacobo Stive Minchala González

## AGRADECIMIENTO:

Agradezco de todo corazón a todas las personas que creyeron en mí desde el principio y siempre tuvieron fe en mí.

A mis hermanos menores Dennis y Uriel que siempre están ahí para motivarme a seguir adelante para poder apoyarles a futuro.

Agradezco a mi compañero de tesis Jonnathan C. y amigos Diego L, Marlon B y Alex M, entre otros más, que siempre estuvieron a mi lado en todo momento.

Jacobo Stive Minchala González

## Resumen

La falta de un aplicativo móvil eficiente para la localización en tiempo real de los buses de la parroquia Guapán causa incertidumbre y dificultad para los usuarios, lo que puede llevar a retrasos y aumento en el tiempo de espera en las paradas de autobús, por lo que se ha propuesto el desarrollo de un sistema de información seguro que proporcione la ubicación de los buses en la parroquia Guapán mediante un dispositivo de rastreo, la aplicación permitiría mejorar la experiencia del usuario en el uso del transporte público, facilitar la planificación de sus viajes y disminuir el tiempo de espera en las paradas de autobús. El primer capítulo detalla el análisis, planteamiento del problema y los objetivos. El segundo capítulo corresponde al marco teórico, realizando una comparación de proyectos similares para seleccionar la mejor herramienta que ayude a resolver el problema planteado. El tercer capítulo detalla sobre las investigaciones que fueron aplicadas y las actividades realizadas para cumplir con los objetivos. En el cuarto capítulo se desarrolló el diseño del sistema de información de los buses basado en la tecnología GPS. Finalmente, en el quinto capítulo se especifican las conclusiones y recomendaciones del trabajo de titulación.

*Palabras clave:* aplicación móvil, modulo gps, modulo esp8266, geolocalización, rastreo

## **Abstract**

The lack of an efficient mobile application for the real-time location of buses in Guapán parish causes uncertainty and difficulty for users. It can lead to delays and increased waiting time at bus stops, so it has been proposed to develop a secure information system that provides the location of buses in Guapán parish through a tracking device. This application would improve the user experience in public transport, facilitate the planning of their trips, and reduce the waiting time at bus stops. The first chapter analyzes the problem statement and objectives. The second chapter is about the theoretical framework, comparing similar projects to select the best tool to solve the problem. The third chapter details the research that was applied and the activities conducted to accomplish the objectives. In the fourth chapter, the design of the bus information system based on GPS technology was developed. Finally, the fifth chapter refers to the conclusions and recommendations of this degree work.

*Keywords:* mobile application, gps module, esp8266 module, geolocation, tracking

# INDICE DE CONTENIDO

DEDICATORIA.....	VI
AGRADECIMIENTO:.....	VII
DEDICATORIA.....	IX
AGRADECIMIENTO:.....	X
INDICE DE CONTENIDO .....	13
Índice de Figuras.....	15
Índice de Tablas .....	18
Capítulo I .....	19
Introducción.....	19
1.1 El problema.....	21
1.2 Planteamiento del problema.....	21
1.3 Objetivos.....	22
1.3.1 Objetivo general.....	22
1.3.2 Objetivos específicos .....	22
1.4 Justificación .....	23
1.5 Estado del Arte .....	24
1.6 Conclusiones .....	26
1.7 Alcance .....	27
Capítulo 2 .....	28
2.1 Marco Teórico .....	28
2.2 Marco Conceptual .....	28
2.2.1 Proyectos de rastreo Vehicular .....	28
2.2.2 Requerimiento del dispositivo de Rastreo.....	30
2.2.3 Modulo GPS Neo 6M .....	33
2.2.4 Frameworks para desarrollo en Android. ....	35
2.2.5 Android Studio .....	36
2.2.6 Spring Boot.....	38
2.2.7 Servicios en la Web .....	38
2.2.8 Base de Datos .....	42
2.2.9 PostgreSQL .....	45
2.3 Metodología .....	46
2.3.1 Metodologías para el desarrollo de Software .....	46
2.3.2 Fases de un Scrum .....	52
Capítulo 3 .....	55



3.1.	Diseño del Sistema .....	55
3.2.	Descripción general .....	55
3.2.1.	Especificaciones de los requerimientos del sistema.....	56
3.3.	Funcionalidad del sistema de geolocalización de autobuses .....	60
Capítulo 4	.....	65
4.1.	Propuesta.....	65
4.2.1.	Desarrollo del dispositivo de Rastreo .....	66
4.2.2.	Creación del Dispositivo .....	67
4.3.	Implementación de Scrum .....	68
4.3.1.	Sprint 1 .....	71
4.3.2.	Sprint 2.....	76
4.3.3.	Sprint 3.....	85
4.3.4.	Sprint 4.....	95
4.3.5.	Sprint 5.....	107
Capítulo 5	.....	119
5.1.	Conclusiones .....	119
5.2.	Recomendaciones .....	119
Bibliografías.....		120
Glosario .....		124

## Índice de Figuras

Figura 1.	App Movilízate UI.....	29
Figura 2.	Moovit.....	29
Figura 3.	Modulo GPS Neo 6M.....	33
Figura 4.	Proceso para adquirir un token de autenticación válido del Administrador de cuentas de Android.....	41
Figura 5.	Triangulo de Hierro.....	47
Figura 6.	Modelo Cascada.....	47
Figura 7.	Metodología Ágil.....	48
Figura 8.	Pilares Scrum.....	52
Figura 9.	Arquitectura del Sistema.....	56
Figura 10.	Vista del diagrama general.....	60
Figura 11.	Mapa de procesos del administrador.....	61
Figura 12.	Mapa de Procesos del Conductor.....	62
Figura 13.	Visualización de la actividad del conductor.....	63
Figura 14.	Mapa de Procesos del Usuario.....	64
Figura 15.	Vista de las actividades que puede hacer el usuario.....	64
Figura 16.	Dispositivo de Rastreo.....	66
Figura 17.	Esquema de conexión de módulos.....	68
Figura 18.	Proceso Scrum.....	70
Figura 19.	Diagrama d Gantt Sprint 1.....	73
Figura 20.	Librerías y Controladores.....	74
Figura 21.	Asignación de usuario y contraseña.....	74
Figura 22.	Método para habilitar Wifi.....	75
Figura 23.	URL del servidor.....	75
Figura 24.	Transformación de datos.....	75
Figura 25.	Coordenadas obtenidas del dispositivo.....	76
Figura 26.	Diagrama de Gantt Sprint 2.....	79
Figura 27.	Clase WebConfig.....	80
Figura 28.	Clase BusControlador.....	81
Figura 29.	Clase DatosBus.....	82
Figura 30.	Clase SecurityConfiguration.....	82
Figura 31.	Método Security.....	83
Figura 32.	Interfaz BusRepositorio.....	83
Figura 33.	Conexión a la base de datos.....	83

Figura 34.	Seguridad de datos.....	84
Figura 35.	Datos enviados del dispositivo.....	84
Figura 36.	Base de datos en Heroku PostgreSQL.....	84
Figura 37.	Diagrama de Gantt Sprint 3.....	88
Figura 38.	Clase MainActivity.....	89
Figura 39.	Clase MainActivity.....	89
Figura 40.	Layout MainActivity.....	90
Figura 41.	Clase Login.....	90
Figura 42.	Clase Login.....	91
Figura 43.	Clase Login.....	91
Figura 44.	Layout Login.....	92
Figura 45.	Clase Cerrar sesión.....	92
Figura 46.	Layout Cerrar sesión.....	93
Figura 47.	Inicio de sesión de la Aplicación.....	94
Figura 48.	Cerrar sesión de la Aplicación.....	94
Figura 49.	Menú para cambio de pantalla.....	95
Figura 50.	Diagrama de Gantt Sprint 4.....	98
Figura 51.	Clase Ira.....	98
Figura 52.	Clase Ira.....	99
Figura 53.	Clase Ira.....	100
Figura 54.	Clase Ira.....	100
Figura 55.	Clase Ira.....	101
Figura 56.	Clase Ira.....	101
Figura 57.	Método Parada más cercana.....	102
Figura 58.	Método rata más cercana.....	102
Figura 59.	Método rata más cercana.....	103
Figura 60.	Método Trazo de línea.....	103
Figura 61.	Método Trazar Ruta.....	103
Figura 62.	Método check GPS.....	104
Figura 63.	Obtener localización del bus.....	104
Figura 64.	Layout de la clase Ira.....	105
Figura 65.	Lista de buses.....	106
Figura 66.	Vista del bus en Movimiento en tiempo real.....	106
Figura 67.	Buscador de sitios.....	107
Figura 68.	Diagrama de Gantt Sprint 5.....	110
Figura 69.	Clase Parada.....	111

Figura 70.	Clase Parada .....	112
Figura 71.	Configuración de la vista del usuario.....	112
Figura 72.	Check GPS .....	113
Figura 73.	Coordenada de las paradas .....	113
Figura 74.	Método de las rutas de autobuses.....	114
Figura 75.	Coordenadas de las Rutas.....	114
Figura 76.	Clase Coordenadas.....	115
Figura 77.	Obtener Coordenadas de rutas .....	115
Figura 78.	Señales guía de rutas .....	116
Figura 79.	Interfaz de obtencion de coordenadas .....	116
Figura 80.	Parada de Autobuses .....	117
Figura 81.	Guía de rutas.....	117
Figura 82.	Lista de rutas a escoger .....	118
Figura 83.	Vista de una de las rutas.....	118

## Índice de Tablas

Tabla 1.	Comparativa de microcontroladores .....	31
Tabla 2.	Comparativa de Módulos .....	32
Tabla 3.	Comparativa de Frameworks para desarrollo móvil .....	36
Tabla 4.	Protocolo de Autenticación .....	40
Tabla 5.	Propiedades de Base de Datos .....	43
Tabla 6.	Comparativa de Base de Datos Relacionales .....	44
Tabla 7.	Cuadro comparativo de Metodologías .....	49
Tabla 8.	Metodologías Agiles .....	50
Tabla 9.	Pinout del módulo NEO 6M.....	67
Tabla 10.	Precio de los componentes .....	68
Tabla 11.	Product Backlog. ....	69
Tabla 12.	Historia de Usuario 1 .....	71
Tabla 13.	Sprint Backlog 1 .....	72
Tabla 14.	Tareas del Primer Sprint.....	72
Tabla 15.	Tareas Culminadas del primer Sprint.....	73
Tabla 16.	Historia de usuario 2.....	76
Tabla 17.	Product Backlog 2.....	77
Tabla 18.	Tareas del segundo Sprint. ....	78
Tabla 19.	Tareas culminadas del Segundo Sprint. ....	78
Tabla 20.	Historia de Usuario 3 .....	85
Tabla 21.	Product Backlog 3.....	85
Tabla 22.	Tareas del tercer Sprint .....	86
Tabla 23.	Tareas culminadas del tercer Sprint.....	87
Tabla 24.	Cuarta historia de Usuario.....	95
Tabla 25.	Cuarto Product Backlog.....	96
Tabla 26.	Tareas del Cuarto Sprint.....	96
Tabla 27.	Tareas terminadas del cuarto Sprint.....	96
Tabla 28.	Historia de Usuario del quinto Sprint .....	107
Tabla 29.	Quinto Product Backlog.....	108
Tabla 30.	Tareas del quinto Sprint .....	109
Tabla 31.	Tareas culminadas del quinto Sprint.....	109



# Capítulo I

## Introducción

Con las mejoras tecnológicas que se presentan día a día, la calidad de vida de las personas se va perfeccionando cada vez más, ofreciendo soluciones a los problemas que se tenían antes, permitiendo así tener más accesibilidad a dispositivos más pequeños y de gran utilidad, dichos dispositivos son computadoras móviles denominados como smartphone que han proporcionado diversas oportunidades para resolver las dificultades que se presentan durante el día mediante las aplicaciones que estos poseen.

Hoy en día, diferentes compañías van a la par en cuanto a tecnología, por lo que en el mercado se encuentran un sin número de nuevas aplicaciones, métodos o herramientas electrónicas que cambian su forma de trabajo para aumentar su producción y ganancias, las compañías optan por crear nuevas tendencias en el ámbito actual ya sea creando robots que ayuden a realizar alguna tarea o en algunos casos solo por simple entretenimiento. Existen dispositivos que son de gran ayuda en el sector del transporte público de pasajeros y deben tener una calidad de comunicación excelente con el usuario, que muestre la ubicación del bus, de las paradas, información de las rutas y demás componentes que generen beneficios a dichas compañías como a sus usuarios.

En el Ecuador, los autobuses inter parroquiales cumplen y garantizan la función de movilización de una parada determinada hacia otra a los usuarios que lo usan frecuentemente; pero, durante ese trayecto se dan casos ajenos al servicio que provocan retrasos y/o en ocasiones interrupciones de los vehículos por lo que no se puede determinar o aproximar el tiempo en el que este llegara hacia su próxima parada, todo esto junto a los problemas que se dan por el desconocimiento que existe sobre la ubicación del bus o situación que se haya dado, las personas lo ven como negativo.

Dicho lo más importante, el objetivo principal de esta investigación es comparar, analizar y probar distintas tecnologías con las que se pudo desarrollar una aplicación móvil para dispositivos Android, misma que permite a los usuarios que usan el transporte público, tener la posibilidad de conocer la ubicación y el recorrido de los vehículos durante su viaje, estimando su tiempo de llegada hacia las distintas paradas, mostrando información necesaria sobre el autobús y ayudando a que los choferes puedan agilizar su servicio.

En el presente trabajo se desarrolló un sistema de información seguro para poder visualizar la localización exacta del transporte Inter parroquial de la parroquia Guapán utilizando la tecnología GPS (Global Positioning System) y red móvil, capaz de ofrecer la localización del vehículo en tiempo real, misma que permite controlar el horario de salida y recorrido de los autobuses.

Este trabajo de investigación se desarrolló bajo la siguiente estructura:

- 1) Primer Capítulo: Corresponde al problema de investigación, donde se detalla las razones que llevaron a la realización del trabajo de investigación, con el análisis y planteamiento del mismo, proponiendo así los objetivos para el desarrollo del trabajo de investigación.
- 2) Segundo Capítulo: Este capítulo corresponde al desarrollo del marco teórico, y se muestran los antecedentes investigativos que tienen relación con la aplicación móvil, es decir se realizó un análisis de artículos similares a los del proyecto y algunos conceptos básicos que son de ayuda para el desarrollo del mismo.
- 3) Tercer Capítulo: Se detallan los tipos de investigación que se aplicaron, que medios se utilizó para la recolección de información, y las actividades que se realizaron para cumplir con los objetivos.

- 4) Cuarto Capítulo: Corresponde al desarrollo y diseño de la propuesta del sistema de información de los buses basado en la tecnología GPS, con el fin de cumplir los objetivos planteados en el Capítulo I.
- 5) Quinto Capítulo: Una vez con los objetivos cumplidos este capítulo corresponde a las conclusiones y recomendaciones que se obtuvieron al finalizar el trabajo de investigación.

## 1.1 El problema

Se define y plantea el problema que se presenta para los usuarios del transporte público de la cooperativa San Francisco. describiendo más adelante la justificación que muestra las alternativas para solucionar el problema con la implementación de apps y herramientas tecnológicas, los objetivos que se deberán cumplir y el alcance del proyecto para llegar a una solución.

El problema se presenta por falta de conocimiento de las rutas u horarios por parte de los usuarios del transporte público, mismos que al esperar desperdician tiempo para llegar a sus trabajos, estudios u otro tipo de ocupación, además de incluir personas que llegan de lugares lejanos y no sepan que unidad de bus tomar para llegar a su destino, además en la mayoría de unidades de transporte no cuentan con un letrero que les muestre a donde se dirigen, siendo tedioso preguntar a gente que desconocen de su destino, dejándose llevar por instintos propios para tomar un autobús.

## 1.2 Planteamiento del problema

Se busca resolver un gran problema que se presenta para los usuarios del transporte público en la Parroquia Guapán, quienes actualmente no cuentan con una plataforma o aplicación que les informe en directo la ubicación del bus y el tiempo estimado de llegada hacia las paradas, debido a que la compañía Inter-parroquial San Francisco

presenta poco interés para interactuar con sus usuarios, sobre el itinerario exacto de salida y llegada de sus unidades de transporte, es por ello que se requiere de un sistema que brinde información clara y precisa al público para evitar problemas de comunicación entre la empresa y los usuarios, creando un sistema que sea amigable para todo público.

## 1.3 Objetivos

### 1.3.1 Objetivo general

Desarrollar un prototipo de un sistema de información seguro en Android que utilice tecnologías IoT para mostrar la ubicación en tiempo real de los autobuses de la compañía Inter-parroquial San Francisco de la parroquia Guapán, para poder mejorar la eficiencia del servicio y la satisfacción de los usuarios del transporte público.

### 1.3.2 Objetivos específicos

- 1º) Realizar un estudio exhaustivo de la problemática planteada, mediante el análisis de conceptos y tecnologías ya implementadas en el mercado, para identificar las mejores prácticas y herramientas para el desarrollo del prototipo de la aplicación móvil en Android.
- 2º) Obtener la información necesaria para el desarrollo del prototipo, mediante el uso de la técnica de recolección de datos, como la observación, con el objetivo de lograr una comprensión detallada de las necesidades y expectativas de los usuarios y los involucrados en la compañía de transporte.
- 3º) Desarrollar el aplicativo móvil en Android utilizando la metodología scrum para asegurar una implementación eficiente y efectiva, cumpliendo con los objetivos.

4º) Evaluar y asegurar la correcta funcionalidad del sistema mediante pruebas exhaustivas y un enfoque en la mejora continua que garantice un rendimiento óptimo de la aplicación.

## 1.4 Justificación

Se desarrolló una aplicación móvil segura para los dispositivos Android que permite la localización y monitorización de los autobuses en la compañía Inter parroquial de San Francisco que operan en la parroquia Guapán mediante un dispositivo de rastreo GPS y tecnologías IoT, que permite la visualización de cada vehículo, siendo de gran ayuda para las personas que viven en las distintas comunidades del sector, dando a los usuarios tiempo para salir con anticipación a la parada e incluso saber por dónde se encuentra el vehículo.

Se visualiza las rutas de los distintos buses de manera gráfica en un mapa, como también las distintas paradas, además de contar con un panel que nos indica la ubicación exacta en directo del vehículo con un mensaje que muestra la velocidad a la que va, teniendo así una aplicación atractiva e intuitiva para la comodidad del usuario, los choferes contarán con la posibilidad de manipular el dispositivo de rastreo para encenderlo en el inicio y apagarlo en el fin de la ruta, también se implementó métodos de seguridad que está conformado por la autenticación de Google, tanto para los clientes como para los conductores.

Esto se desarrolló utilizando un módulo wifi ESP8266 y un módulo GPS Neo 6m, también el software libre de Android Studio, Arduino, así como de igual manera la base de datos PostgreSQL y Firebase, se usó un servidor gratuito Heroku configurado con Spring Boot para enviar y recibir la información mediante internet, siendo servicios que serán consumidos por todos los usuarios que posean la aplicación.



## 1.5 Estado del Arte

En enero de 2010 en Guayaquil-Ecuador los estudiantes Pedro Fausto Alzamora Ramírez y Alex Javier Bautista Ramírez, dieron a conocer su tesis denominada “Control y monitorización del recorrido de los buses de transporte público mediante tecnología GPS y GSM”, realizando marcaciones virtuales en el recorrido de los buses, evitando el exceso de velocidad para llegar a una parada o controlador de tiempo a marcar.

Haciendo uso de las tecnologías GPS y GSM, se puede obtener datos requeridos tales como la hora actual, la fecha, latitud, longitud (localización) y velocidad del vehículo monitoreado, lo cual dará información exacta en el momento requerido.

Este sistema de seguimiento vehicular se aplicó para el transporte público de la ciudad de Guayaquil, realizando la verificación de la información obtenida durante el trayecto del mismo, dando el servicio de envió de mensajes de alerta a la estación y al propietario del carro. Teniendo la seguridad que será una forma idóneos para contrarrestar el problema de accidentes de los buses de transporte urbano, además de saber la posición precisa al momento del traslado de los mismos [1].

En otra investigación, Villa Verónica Alexandra, declaro que se trata de resolver el manejo de los recorridos de la Cooperativa de transporte Atahualpa, usando un equipo GPS para cada transporte de bus respectivamente, contando la cooperativa con un promedio de 30 buses aproximadamente y así medir el tiempo de salida como el de llegada, además de poder ver las multas emitidas y sus respectivas sanciones por exceso de velocidad que marca cada GPS.

Este sistema trata de disminuir las competencias que existe entre estos mismos ya que al ser un servicio de transporte público, pone en peligro a muchas vidas, es por ello que tratan

de prevenir los accidentes de tránsito, multas por exceso de velocidad, daño de los vehículos y quejas por parte de sus usuarios [2].

En abril del 2016 en Quito -Ecuador el estudiante Juan Pablo Rojas Hernández, dio a conocer su trabajo de titulación denominado “Desarrollo de un Prototipo Funcional para la Aplicación Móvil Q-Bus, plataforma que brinda Información de las rutas de Transporte Público en la Ciudad de Quito utilizando Bluetooth Low Energy, Códigos QR y Geo Posicionamiento”. En el mencionado trabajo de titulación expone el análisis, diseño e implementación de un prototipo funcional para la aplicación móvil Q-Bus en la plataforma IOS que brinda información de las rutas de transporte público en la ciudad de Quito haciendo uso de las tecnologías Bluetooth Low Energy, Códigos QR y Geo Posicionamiento. Para el desarrollo de la investigación utilizaron la metodología de desarrollo basada en prototipos, de esta manera se pudo obtener un producto de alta fidelidad que se asemeje en la mayor medida posible a una aplicación móvil final [3].

En el año de 2018, Betancourt Rodríguez Luis Eduardo realiza un estudio basado en los servicios públicos para la geolocalización de transportes públicos con el fin de optimizar su tiempo de llegada a sus respectivos sitios, además de obtener sitios de referencia, ver el vehículo en tiempo real y mostrar información de la ruta que recorrerá, optando a resolver dicho inconveniente para que la ciudadanía sepa en donde se encuentra su respectivo transporte publico [4].

En un estudio realizado en 2018 por Álvaro Yélamos San Andrés en el país de España, propuso una aplicación para la mejora de los servicios de buses urbanos para la ciudad de Guadalajara, que brinda información de sus respectivas rutas a usuario y únicamente estaría enfocada al transporte urbano, además de integrar una información predictiva para el tiempo de llegada de los vehículos a su respectiva parada [5].

Por otro lado, Rommel Ufredo Hurtado Suárez realizó una investigación en el año 2018, que tuvo el interés de enfocarse en un trabajo de calidad para los servicios de transporte público, por ello propuso implementar un sistema de geolocalización para los buses de transporte público, lo cual permitiría ver sus respectivas rutas, así como el tiempo de llegada hacia una parada o estación, todo eso lo hará haciendo uso de módulos GPS, GSM, Arduino, entre otros, con el fin de armar un sistema de seguimiento en tiempo real y con ello dar la información necesaria al usuario para poder conocer su paradera o del vehículo en si [6].

Luis Javier Quinde Pozo, en el año 2019 de la ciudad de Santa Elena en Ecuador, relata que existe una falta de información por parte de la Agencia Nacional de Tránsito (ANT), al no contar con una aplicación que brinde información sobre las distintas rutas del transporte público, siendo un problema para su ciudad y para la empresa en si, por ello da la solución de crear “Un sistema de seguimiento en tiempo real del transporte público”, que sea de gran ayuda para aquellas personas que no conozcan un lugar e incluso para turistas [7].

## 1.6. Conclusiones

En conclusión, el estado del arte ha mostrado una gran cantidad de investigaciones enfocadas en mejorar la eficiencia y precisión del seguimiento, haciendo uso de tecnologías modernas e intuitivas con el fin de resolver el problema de información que existe por parte de las compañías de transporte. A pesar de que varias tesis han abordado el tema del rastreo de buses, la mayoría carece de medidas de seguridad adecuadas para proteger la privacidad y los datos de los usuarios. Esto es una preocupación importante, ya que los datos de rastreo de buses pueden incluir información personal y confidencial, como la ubicación de un usuario en tiempo real.

## 1.7 Alcance

Se desarrolló e implementó un prototipo de una aplicación móvil para todos los dispositivos Android, en donde se puede visualizar la ubicación exacta de los vehículos de transporte público de la compañía San Francisco en tiempo real, con mapas que muestran el recorrido que realiza el bus, el tiempo estimado de llegada del vehículo hacia la parada más cercana del usuario, la velocidad actual a la que se está circulando.

Como primer punto, se tomó como caso de estudio a las personas que usan el bus diariamente y presentan inquietudes con el servicio de transporte, esto nos ayudó a diseñar e implementar el prototipo de hardware que cuenta con un módulo GPS de rastreo en un vehículo de prueba, de la misma manera se programó el envío de las coordenadas hacia una base de datos que funciona dentro de un servidor para poder obtener la ubicación del vehículo en tiempo real mientras se moviliza.

En segundo, se programó la aplicación móvil y web con un entorno de desarrollo gratuito, se realizaron pruebas para el óptimo funcionamiento del sistema que cuenta con el respectivo desarrollo de la base de datos, posee un sistema de registro para los respectivos usuarios, los mismos tienen un id y una contraseña para el ingreso seguro a la plataforma y con todo esto se puede visualizar toda la información de manera sencilla mediante un dispositivo móvil con sistema operativo Android.

## Capítulo 2

### 2.1 Marco Teórico

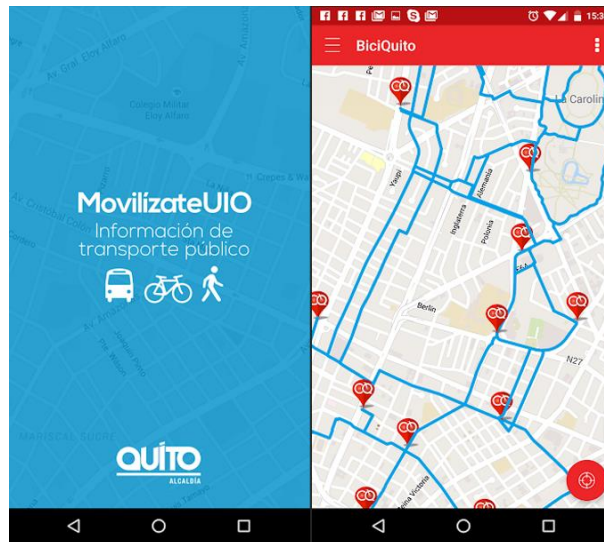
Una vez generado el estado del arte, se obtiene una visión a donde se quiere llegar con el desarrollo del aplicativo móvil y el dispositivo de rastreo satelital para el seguimiento de autobuses en tiempo real, teniendo en cuenta los Módulos a usar y la tecnología ESP8266 que tendrá constante comunicación con el servidor, haciendo uso de un gestor de base de datos, ya que este deberá enviar información a través de la aplicación que soporte Android, por otro lado, se integrara el servicio de Google Maps que se usa para graficar las respectivas rutas de cada línea de autobús, además de mostrar el punto exacto en donde este se encuentre en el mapa dentro de la parroquia mencionada.

### 2.2 Marco Conceptual

#### 2.2.1 Proyectos de rastreo Vehicular

En la actualidad existen pocas aplicaciones que brinden un servicio de información sobre el paradero de un vehículo, sin embargo, la información de algunas aplicaciones solo cuenta con datos de ciudades grandes, como Movilízate UIO y Moovit. A continuación, veremos la descripción de cada aplicación, donde se tendrá una idea de que desarrollar.

Movilízate UIO



*Figura 1. App Movilízate UI*

Creada por el municipio del Distrito Metropolitano de Quito con el apoyo de Google Maps, esta aplicación se pone a disposición de la ciudad al brindar información sobre sus rutas ya sea de buses o de sus estaciones de BiciQuito como se ve en la figura 1, dando un servicio de seguridad con la posibilidad de notificar algún tipo de emergencia, incidencias o denuncias que puedan ocurrir en el transcurso del viaje [8].

## Moovit



*Figura 2. Moovit*

Moovit es una aplicación gratuita que brinda información sobre el servicio de transporte público disponible para Android, IOS y plataforma webs como se observa en la figura 2,

fue lanzado en el año 2012 y actualmente está disponible en 112 países constando con 3,500 ciudades del mundo con sus respectivos idiomas, la aplicación brinda el servicio de información de bicicletas, transporte como Uber, Scooter, entre otros [9]. En 2017 se integran soluciones como Mobility as a Service (MaaS), con herramientas de inteligencia artificial que integra planificaciones y optimizaciones con el propósito de mejorar la gestión y eficiencia operativa [9].

### 2.2.2 Requerimiento del dispositivo de Rastreo

#### **Microcontroladores**

Los microcontroladores son pequeños circuitos que van integrados en una CPU, se caracterizan por ser una mini computadora que realiza procesos operaciones y complejos, al ser unos dispositivos diminutos, estos requieren ser programados para llevar a cabo una función, dependiendo del proyecto que se realice con cada dispositivo que hay.[10]

Existe una gran variedad de compañías que brindan al público unas placas programables, que entre ellas puede tener mejores apartados que otros, es por ello que se realizó una comparativa entre los tipos de microcontroladores que hay en el mercado como se observa en la tabla 1.

Tabla 1. Comparativa de microcontroladores

Microcontroladores			
Características	Raspberry PI	Arduino	System on Chip (SoC)
Puede funcionar con una batería		x	x
Puede estar en constante funcionamiento			X
Contiene Librerías para interactuar con el dispositivo	X	X	X
Cuenta con almacenamiento integrado		X	X
Programable	X	X	X
Entrada de puertos USB	X	X	X
Fácil Uso	X	X	X
Precio accesible		X	X

Adaptado por: Minchala J, Crespo J.

Al haber hecho un análisis comparativo de los tres microcontroladores más usados hoy en día, se obtuvo dos candidatos para su uso, debido a que estas mini computadoras se acoplan al trabajo propuesto, si bien es cierto que raspberry cuenta con otros apartados que lo aventaja, no cuenta con las características necesarias para su uso, al tener claro lo dicho anteriormente, procedemos a comparar los módulos de estos fabricantes para su uso [10].

### 2.2.2.1 Módulos

Tanto Arduino como un SoC. son prototipos eléctricos Open Source, con una arquitectura de Software y Hardware flexible para el uso. Principalmente diseñado para la creación de un entorno productivo de diferentes tipos de microordenadores con una gran infinidad de funciones, logrando esto por medio de una placa[11]. Gracias a su aplicación, se puede



programar desde su IDE, con solo conectar un cable USB se implementarán los cambios realizados.

Ambos módulos cuentan con su propia librería de proyectos y de dispositivos, dando la facilidad de interactuar y crear proyectos propios, además, incluyen con una gran base de información, sensores, servomotores, mini focos led, relés, entre otros apartados más [12]. Para saber qué módulos cumplen con las características necesarias, se planteó a realizar un cuadro comparativo como se detalla en la Tabla 2, con el fin de identificar las propiedades requeridas.

Tabla 2. Comparativa de Módulos

Módulos		
Características	ESP8266	Arduino Uno
Microcontrolador	X	X
Mas de 1 Mega de Memoria	X	
Tensión de funcionamiento de 5V		X
Wifi	X	
Bluetooth		
32 bits	X	
Precio accesible	X	

Adaptado por: Minchala J, Crespo J.

Analizado las propiedades de cada Módulo, se obtuvo una clara obtención de los resultados para la aplicación del dispositivo, ESP8266 obtiene mayores ventajas que el Arduino Uno, ya que este como ventaja tiene parámetros más altos como; la integración de Wifi y más espacio de memoria, bajo precio de coste, actualización del firmware por Over The Air (OTA) por potencia y fácil uso, sacando una gran ventaja en realizar tareas en conjunto, sin embargo, el Arduino Uno podría realizar las mismas tareas pero se necesitaría la incorporación de otros componentes para igualar las características que se desea realizar con el [13].

Teniendo en cuenta las características de los módulos Arduino que se necesita para armar el prototipo, se debe tener en presente otros aspectos importantes para la implementación del mismo.

### 2.2.3 Modulo GPS Neo 6M



*Figura 3. Modulo GPS Neo 6M*

El módulo GPS Neo 6M, es un dispositivo GPS fabricado por U-Blox, diseñado principalmente para el uso en aplicaciones de geolocalización, este módulo es compatible con Raspberry o Módulos Arduino, tiene dimensiones de (12 x 16 x 2mm), cuenta con una antena de gran amplitud, un indicador LED, su propia batería de respaldo y una memoria EEPROM para almacenar la información que recolecta al moverse, este tipo de dispositivo son usados en diferentes tipos de vehículos para la recolección de datos, observándose en la figura 3 el diseño del módulo [14].

La precisión del dispositivo es de 2.5 metros a 5 metros con función de calibración, su orientación es de 0.5°, la obtención de datos son Latitud, Longitud, Velocidad, Fecha y Hora, útil para el desarrollo del dispositivo de rastreo vehicular.

#### 2.2.3.1 *The Global Positioning System (GPS).*

The Global Positioning System (GPS), es un sistema de navegación y posicionamiento global, la cual fue implementado por el departamento de Defensa de los Estados Unidos,

compuesta de una red de 24 satélites y fue principalmente diseñado para brindar información a los militares en la década de 1980, tiempo después el gobierno declaró que el sistema de posicionamiento estuviera disponible para todo el mundo.

Su alcance está disponible a nivel mundial y funciona en cualquier condición climática las 24 horas del día sin hacer uso de una membresía o cargos adicionales, siendo usado por muchas personas o empresas para el ámbito profesional o ya sea personal [15].

### 2.2.3.2 *Funcionamiento de un sistema GPS*

El objetivo de un sistema GPS es obtener las coordenadas de cualquier punto geográfico de la tierra, conformado por 3 números reales: (X, Y, Z), son los encargados de transmitir el posicionamiento del objeto, “La distancia que tiene un usuario con un satélite, se calcula multiplicando el tiempo de vuelo que emite la señal del satélite por la velocidad de propagación”[16]. Para la obtención de los cálculos, los relojes de los satélites deben estar sincronizado con los receptores que generaran un mismo código que medirá el tiempo de vuelo.

### 2.2.3.3 *Funciones GPS*

Las tareas o actividades que puede realizar un dispositivo GPS son muchas gracias al avance tecnológico, entre las actividades que puede realizar esta la navegación, lecturas atmosféricas, entre otras, esto requiere de una gran precisión en tiempo real donde la información sea exacta y no exista fallos, entre las actividades que puede realizar un sistema GPS están:

- **Lectura de fenómenos atmosféricos:** Las lecturas de fenómenos atmosféricos han sido predichas por artefactos que emiten señales GPS, brindando información de la velocidad a la que se propaga.

- **Navegación y geolocalización:** Los sistemas GPS usan sistemas guiados de sitios explorados que serán vistos en mapas, ofreciendo información del paradero del objeto que cuente con un sistema GPS[16].
- **Topografías:** Los topógrafos hacen uso de un sistema GPS para el levantamiento de un terreno, inventarios agrarios, entre otros.
- **Navegación aérea y marítima:** Los aviones y vehículos submarinos hacen uso de un sistema GPS para navegar ya sea por agua o tierra, siendo implementado en 1946 y siendo un sistema de navegación estándar a nivel mundial [17].
- **Navegación desasistida de vehículos:** Se incorporan en los vehículos un sistema Differential Global Positioning System (DGPS) con el fin de monitorear el tráfico presente, además se usa en tareas de vigilancia en zonas hostiles e incluso el sistema se encuentra presente en camiones de carga, para monitoreo y seguridad [16].

#### 2.2.4 Frameworks para desarrollo en Android.

Los frameworks han sido de gran utilidad para la creación de una aplicación, debido a que presentan una gran demanda para el desarrollo de aplicaciones en móviles en Android e IOS, incluyendo distintos lenguajes de programación como: C#, Hyper Text Markup Language (HTML), Hojas de estilo en Cascada (CSS), TypeScript, Java, entre otros.

Un framework permite facilitar el trabajo de los desarrolladores con la integración de herramientas y componentes que agilicen la programación, un framework es considerado una biblioteca por la forma que tiene de disminuir los errores que se vayan presentando en el camino. Gracias a la existencia de diferentes frameworks, se puede desarrollar aplicaciones Web, Móviles, Inteligencia Artificial (IA), entre otros [18].

Para saber qué tipo de frameworks se acopla al trabajo, se realizó una comparativa de algunos frameworks para el desarrollo móvil como se observa en la Tabla 3, con el fin de saber sus ventajas y desventajas que presente estos sistemas.

Tabla 3. Comparativa de Frameworks para desarrollo móvil

<i>Frameworks</i>				
	<b>Xamarin</b>	<b>React Native</b>	<b>Android SDK</b>	<b>Flutter</b>
<b>Compatibilidad con Java</b>	X	X	X	X
<b>IDE Único de desarrollo</b>	X		X	
<b>Android development</b>	X	X	X	X
<b>Arquitectura MVVM</b>	X		X	
<b>Librerías y complementos</b>	X	X	X	X

Adaptado por: Minchala J, Crespo J.

Al comparar los distintos frameworks de desarrollo en Android, se pudo obtener una amplia variedad de las características que posee cada uno de los framework, ya sea en sus librerías, complementos, lenguaje de programación, entre otros apartados más. Si bien es cierto que el desarrollo de aplicaciones en multiplataforma tiene sus ventajas por tener su flexibilidad al poseer un lenguaje universal como es JavaScript y al tener la reutilización de código para evitar la fragmentación del mismo, sin embargo, Android SDK es un perfecto ganador por su rendimiento que posee para los dispositivos que van dirigido este proyecto [19], [20].

Una vez aclarado los complementos de los frameworks, decidimos escoger Android SDK para el desarrollo del aplicativo móvil para el rastreo de los autobuses.

### 2.2.5 Android Studio

Es una Plataforma de desarrollo integrado exclusiva para la creación de aplicaciones Android y está basado en IntelliJ. Incluye una amplia red de funciones que mejoran la

interacción con el desarrollador, Android Studio usa principalmente el lenguaje Java y Kotlin, donde sus API integran el mismo lenguaje.

Android Studio integra como componente a Android SDK, que incluyen sus propias herramientas para interacción con la propia plataforma de Android, incluyendo su propio sistema para cada versión de su respectiva plataforma con niveles de API como: Celulares, Android Wear y Android TV. Además, incluye los servicios de Google Play la cual se etiqueta con la API de Google.

El proceso de compilación de una app creada en este programa incluye un espacio de trabajo, la escritura de la app ayuda a la agilidad del trabajo de desarrollo, incluye la compilación y ejecución del mismo donde se creará un paquete APK para probarlo en dispositivos Android, por lo que puede generar perfiles y hacer pruebas para solucionar los errores que se vayan presentando, para luego ser publicado a los usuarios [21].

Antes de comenzar a desarrollar un programa en Android Studio, este nos lanzara un mensaje la cual nos dice en que lenguaje deseamos programar, donde nos aparecerá entre Java y Kotlin como sugerencias.

#### *2.2.5.1. Lenguaje Java*

Java es un lenguaje de programación que está orientado a objetos, usado para la creación de aplicaciones en multiplataforma de cualquier tipo de dispositivo, ya sea para celulares como para TV, Java es principalmente usado por empresas para el desarrollo de aplicaciones Web y IoT, cuenta con un Kit para desarrollo en Java, disponible para Windows y Mac, siendo versátil para el sistema operativo en Android con una facilidad de aprendizaje sencilla y ágil de comprender [22].

## 2.2.6 Spring Boot

Spring es un Framework enfocada en java para desarrollo de aplicaciones empresariales, Spring esta principalmente orientada en la “fontanería” para las aplicaciones de las empresas. Las funciones que ofrece Spring es la inyección de las dependencias y evento enfocadas para tecnologías fundamentales, admite las pruebas MVC, acceso a los datos como las transacciones, Marshalling XML y lenguajes como Kotlin y lenguajes que sean considerados dinámicos. Spring es de fácil uso al ser un lenguaje de programación basado en Java, integrando la seguridad, microservicios, entre otros, con el fin de desarrollar una productividad buena en el trabajo [23].

## 2.2.7 Servicios en la Web

Los servicios Web son softwares creados para el intercambio de datos por medio de una función de red, apoyándose en distintos estándares de los servicios, los servicios Web usan el lenguaje XML que se adaptan a cualquier tipo de negocio conociéndosele como el intercambio electrónico de datos. Esto permite que los sistemas de software sean más autónomas y fáciles de manipulas, dando como resultado un valor agregado para los servicios Web [24].

### 2.2.7.1 *API*

Las API “Application Programming Interface” son un grupo de bibliotecas y protocolos que brindan sus servicios para la comunicación entre cliente y servidor. La app desarrollada hace una solicitud al servidor, teniendo como resultado una respuesta por parte de este. Las API pueden brindar distintos servicios como funciones según requiera el usuario para uso de su aplicación, estos deben contar con la seguridad necesaria para evitar cualquier tipo de ataque y su rendimiento debe soportar el tráfico de datos para evitar los cuellos de botella.

La arquitectura de las API se clasifica por el uso que se le da. Entre estas hay; API Privadas; API públicas; API de Socios y API Compuestas, cada una con su rol respectivo [25].

#### 2.2.7.2 *Postman*

Postman es una plataforma que nos permite usar las peticiones GET, POST, DELETE y UPTATE, permitiéndonos obtener una respuesta en los formatos XML, JSON y en Texto, su manejo es fácil e intuitivo para cualquier persona con de crear API de manera rápida [26].

#### 2.2.7.3 *Firebase*

Creada por Google, Firebase es un Kit de desarrollo de software con muchas funciones para la programación de aplicaciones Web o móvil, Firebase permite el desarrollo en multiplataforma con integración de un conjunto de interfases de programación ya incluidas, no solo cuenta con opciones de desarrollador, al contrario, incluye el monitoreo y medición del preformas del proyecto, cuenta con sincronización con base de datos a través de la nube, permitiendo un ágil uso de Firebase [27].

### **Protocolos de Autenticación**

Los protocolos de seguridad son necesarios para la seguridad de la información de una aplicación, ya sea Web o móvil. Las autenticaciones son registros criptográficos que tienen como objetivos el intercambio de datos de entre el cliente y servidor, permitiendo que la información del usuario no sea robada por aplicaciones de terceros [28].



Debido a esto se procedió a realizar un cuadro comparativo de los protocolos de autenticación de datos para la seguridad del usuario, mostrando las características más importantes para su implementación como se observa en la tabla 4.

Tabla 4. Protocolo de Autenticación

Protocolo de Autenticación				
	LDPA	Oauth 2.0	RADIUS	Secure Shell (SSH)
Autenticación	X	X	X	X
Autorización		X		
Autenticación Multifactor		X	X	X
Acceso Condicional		X	X	X

Adaptado por: Minchala J, Crespo J.

Aun vez obtenida la comparativa de estos protocolos de autenticación, observamos que OAuth cuenta con una seguridad más estable para su agregación en el proyecto, si bien es cierto que existen más protocolos de autenticación, los de la tabla 4 son los más usados por los desarrolladores, es por ello que se hará uso de OAuth 2.0.

#### 2.2.7.4 *OAuth 2.0*

Es un protocolo para la autorización enfocado principalmente para la seguridad de una aplicación, ya sea en web, móvil o de escritorio. OAuth cuenta con gestores de token que permiten el acceso de manera estructurada, también cuenta con autorización de metadatos de los servidores, entre otras funciones más [29].

- **Autenticación con los servicios OAuth 2.0**

Google para sus autenticaciones o autorizaciones usa OAuth 2.0 como protocolo en sus API, principalmente usada para sitios web o servidores de clientes y web.

En una aplicación lo más importante es la seguridad ya que hoy en día existe muchos ataques por parte de aplicaciones de terceros, esto hace que sea necesario la autenticación a la hora de acceder a un servicio, además de incluir a la app, siendo esto primordial para que pueda ser interactuado por el usuario [30].

Oauth 2.0 ofrece un token de acceso para añadirlo en las aplicaciones, es muy común que muchos desarrolladores implementen la seguridad por autenticación por el bien de sus usuarios, para saber cómo funciona este proceso, Developers lo demuestra en la siguiente figura 4.

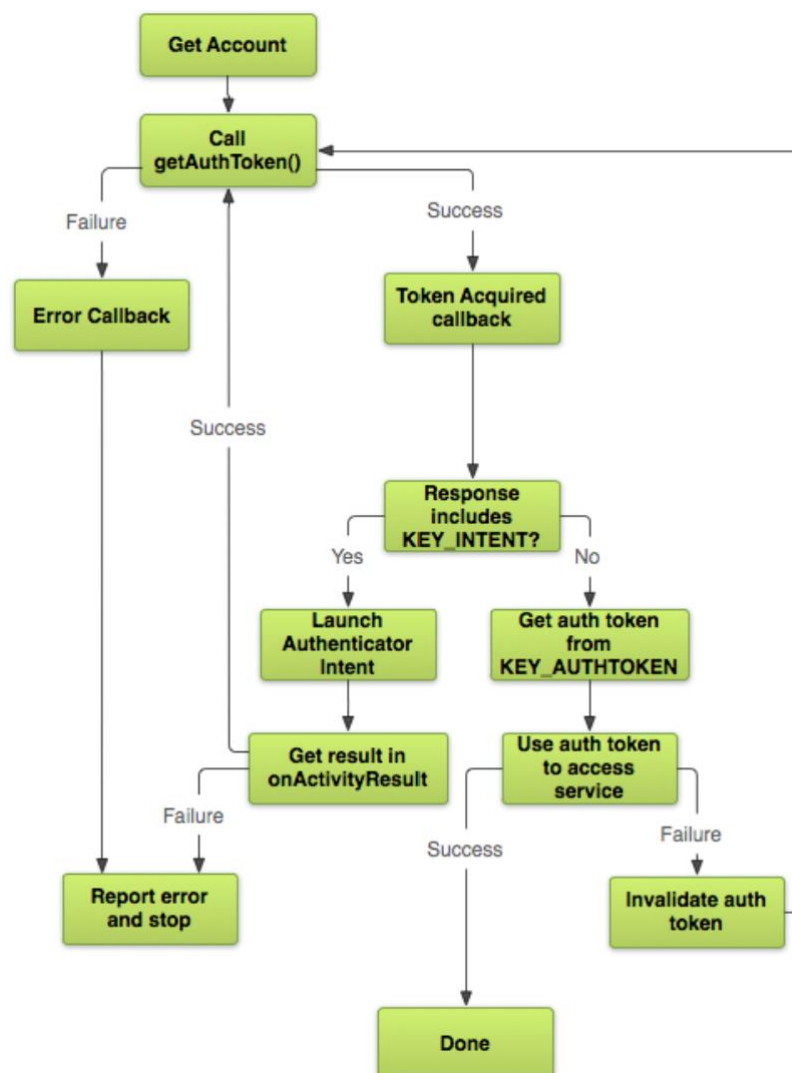


Figura 4. Proceso para adquirir un token de autenticación válido del Administrador de cuentas de Android

## 2.2.8 Base de Datos

Una base de datos (BD), es un centro de almacenamiento y medición de datos que las organizaciones usan para sus aplicaciones, por lo que la información que genera las empresas son grandes cantidades de datos, es por ello que las bases de datos permiten realizar búsqueda por comandos, consultas e ingreso de datos, con el fin de obtener la información guardada en tablas, siendo estas lo que compone a una base de datos [31].

### 2.2.8.1 *Gestor de base de Datos*

Un Gestor de base de datos (SGBD) es considerado como complementos de software que deben realizar funciones muy específicas y complejas, ya que deben estar preparadas para la gestión y una buena administración de datos, dado a que manejan información muy importante para una compañía o empresa. Los procesos que realiza un SGBD es la transformación de las consultas en una agrupación de nivel bajo que van dirigido al gestor, obteniendo una vista muy clara de la información administrada. Un SGBD puede realizar un llamamiento al gestor de los ficheros que llevara a cabo unas consultas donde examinara el esquema conceptual y externo determinando que tipo de registro se requerirán [31].

Gracias a los SGBD, se han generado muchos tipos de base de datos, ya sean Base de datos relacionales (SQL) o no relacionales (No-SQL), por ello se procedió a realizar una comparativa como se ve en la Tabla 5, con el fin de obtener una mayor autonomía y escalabilidad en los datos.

Tabla 5. Propiedades de Base de Datos

Base de Datos Relacional (SQL)	Base de Datos No-Relacional (No-SQL)
Hace manejo de esquemas	No hace falta realizar esquemas
Requiere en muchos de los casos normalizar	No hace falta normalizar
Manipula tablas con registros	Manipula arreglos a los documentos
Es primordial crear una llave primaria	No hace falta una llave primaria, es decir, el sistema lo identifica con una llave
Es útil para manejar transacciones	No realiza transacciones
Al tener campos vacíos se puede tener un desperdicio de memoria	Al tener campos repetidos se puede tener un desperdicio de memoria

Adaptado por: Minchala J, Crespo J.

Al obtener una clara referencia de las diferencias que presenta las bases de datos relacionales y no relacionales, hay que tomar en cuenta que tipo de base de datos será utilizada para realizar una buena gestión, flexibilidad y rendimiento. Por lo que, las bases de datos relacionales cuentan con tablas bien estructuradas, proporcionando acceso a los datos de manera ordenada, mientras que las bases de datos no relacionales carecen de un sistema normalizado.

Es así que se hizo una comparativa de los sistemas de base de datos como se ve a continuación en la Tabla 6.

Tabla 6. Comparativa de Base de Datos Relacionales

Sistema de Base de Datos			
	MySQL	Oracle	PostgreSQL
DBMS Relacional	X	X	X
Esquema de Datos	X	X	X
Compatible con XML	X	X	X
Métodos de Partición	Particionamiento horizontal, Particionamiento con MySQL	Particionamiento horizontal, Sharding	Particionamiento mediante rango, por Hash a través de PostgreSQL 11
Licencia		X	
Lenguajes de programación compatibles	C, C#, C++, D, Delphi, Java, JavaScript (Node.js), Objective-C, PHP, Python, Ruby	C, C#, C++, Clojure, Delphi, Eiffel, Groovy, Haskell, Java, JavaScript, Objective-C, OCaml, Perl, PHP, Python, Ruby, Scala, Visual basic	.Net, C, C++, Delphi, Java (JDBC), JavaScript (Node.js), Perl, PHP, Python. Tcl

Adaptado por: Minchala J, Crespo J.

Al realizar una comparativa de los sistemas de base de datos relacionales, hemos optado por hacer uso de PostgreSQL, ya que, en la creación de un proyecto, este depende de qué tipo de bases de datos se hará uso, tomando en cuenta la necesidad del rendimiento y de las consultas.

### 2.2.9 PostgreSQL

Es una base de datos relacional que usa el lenguaje SQL que permite la creación de aplicaciones y de entornos ante las fallas, la administración de datos, entre otros, Postgres al ser una protoforma gratis permite realizar muchas funciones para facilitar el gestionamiento de datos grandes. Entre sus principales funciones que PostgreSQL posee son:

- Consultas Complejas.
- Tipos de datos.
- Rendimiento.
- Seguridad.
- Vistas.
- Extensibilidad.
- Transacciones de datos.

Estas son unas de las principales características que posee PostgreSQL, sin embargo, cuenta con enlaces de lenguajes de programación como: C, C++, Java, entre otros como se mencionó anteriormente en la Tabla 4 [32].

#### 2.2.9.1 *Arquitectura PostgreSQL*

La arquitectura de PostgreSQL hace uso de un modelo Cliente – Servidor, que consiste de la siguiente manera:

Un Servidor: gestiona los archivos recibidos de la base de datos para realizar las conexiones con la base de la aplicación de los clientes, realizando las acciones pertinentes hacia la base de datos en petición a los clientes [31].

Un Usuario: permite llevar a cabo las peticiones de la base de datos, dependiendo de la petición este puede ser de naturaleza diversa, es decir que puede estar orientada a un texto, a un servidor que este programado para mostrar una página web o una herramienta que esté relacionada con una base de datos [31].

## 2.3 Metodología

### 2.3.1 Metodologías para el desarrollo de Software

Las metodologías para el desarrollo de software, es un grupo de métodos que se usan para la creación de un software, el propósito de las metodologías de desarrollo es agilizar el trabajo en equipo y organizar de forma correcta sus funciones para crear un software de calidad.

Las metodologías de desarrollo de software se dividen en Clásicas y Agiles, la metodología ágil es una alternativa a la metodología tradicional, más cuando se trata de equipos de desarrollo, antes de utilizar una metodología para la gestión, es importante tener en cuenta los enfoques del proyecto a seguir, ya que cada una de las metodologías que tienen sus propias ventajas y desventajas [33]. En cada proyecto siempre existen 3 variables a tomar en cuenta, esto se viene a llamar el “Triangulo de Hierro” figura 5, que está conformado de la siguiente manera:

- Alcance: Determina las actividades para lograr un buen producto deseado.
- Tiempo: El periodo de duración de la entrega del producto.
- Costo: Valor que se determina al tomar en cuenta los recursos utilizados.

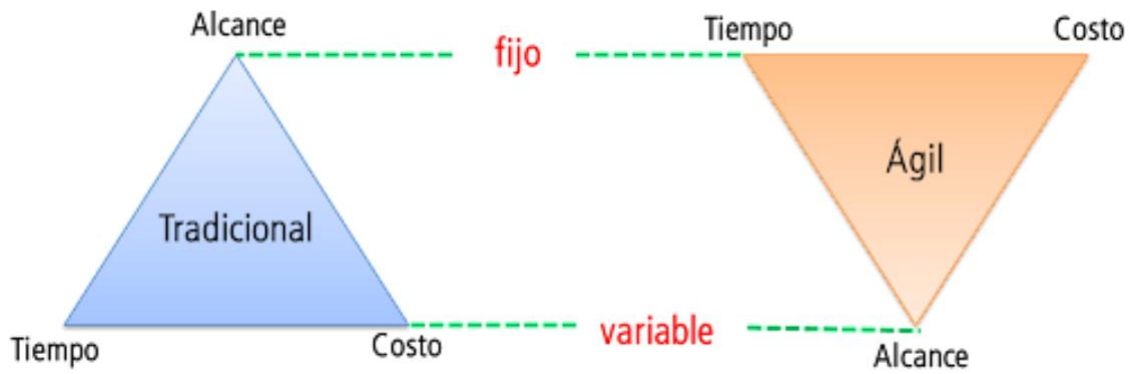


Figura 5. Triángulo de Hierro

### 2.3.1.1 Metodologías Clásicas

Las metodologías clásicas o tradicionales se enfocan en la gestión de los proyectos y se caracterizan por su rigidez, su poca flexibilidad y la abstención de realizar cambios, esta metodología surge del Departamento de Defensa de los Estados Unidos, caracterizándose por sus fases de secuencia siendo inutilizada la gestión de proyectos tradicional para el desarrollo de trabajos difíciles, dado que no se puedan generar cambios en el producto [33].

Las metodologías clásicas usan una secuencia en cascada para el análisis de la metodología, siendo un solo camino sin regreso a modificaciones de las actividades, figura 6.

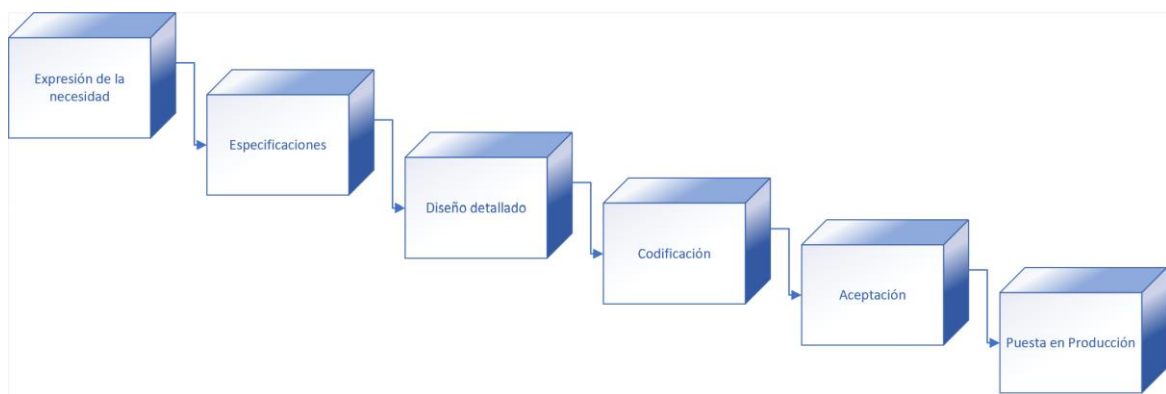


Figura 6. Modelo Cascada



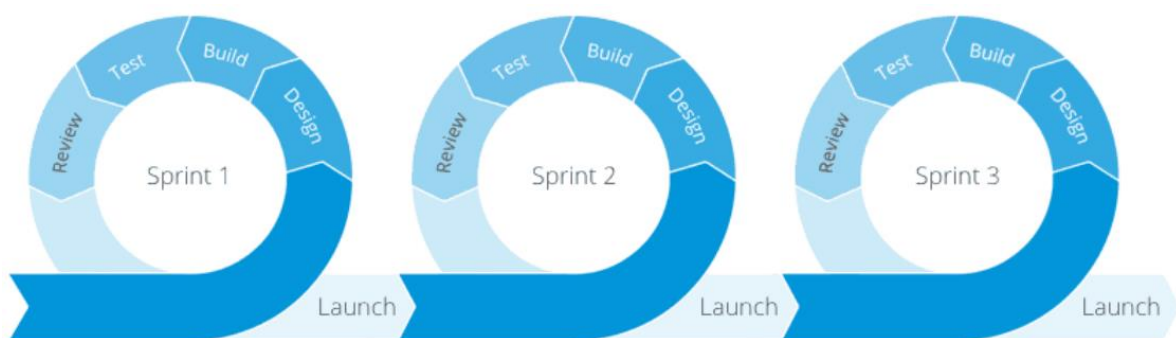
Entre las metodologías más usadas están:

- **Waterfall**
- **Espiral**
- **Prototipo**
- **Incremental**
- **Diseño rápido de aplicaciones**

### 2.3.1.2 Metodologías Ágiles

Las metodologías ágiles se dividen en repeticiones cortas, estos tienen la duración máxima de un mes de calendario, seguido se obtendrá un aumento en el trabajo después de cada iteración, las metodologías ágiles se enfocan en tomar en cuenta los comentarios de los clientes, donde se revisa detalladamente el producto, ya que es importante la opinión del cliente para la agilización del proyecto de forma rápida, haciendo los cambios correspondientes donde se establece rebajar los tiempos de desarrollo, la eliminación de ciertas huecas que exista en el proyecto, mejora de comunicación en el equipo para la obtención de buenos resultados.

Las metodologías ágiles se ven como en la figura 7, dando un enfoque ágil para adaptar la metodología a elegir.



*Figura 7. Metodología Ágil*

Entre las metodologías Ágiles más usadas y debatidas para usar una están:

- **Lean**
- **Programación Extrema (XP)**
- **Kanban**
- **Scrum**

Para saber qué tipo de metodología escoger, hemos procedido hacer una comparativa Tabla 7, de las metodologías ágiles contra las tradicionales ellas para implementarla en el proyecto.

Tabla 7. Cuadro comparativo de Metodologías

Metodología Tradicional	Metodología Ágil
Muy Inflexible	Desarrollo Ágil
No aporta Cambios	Preparada para realizar cambios durante el trabajo
Anticipa a la metodología	Se agiliza ante la metodología
Su entorno son los procesos	Su entorno son las personas
Su comunicación con el cliente es poca	Mantiene una comunicación constante con el cliente para realizar cambios o mejoras en el proyecto.
Tienen sus objetivos definidos	Tienden a mejorar para obtener un producto de calidad.
Tiene pocos roles	Incluye más roles
Contiene un grupo menor a 10 personas	Su distribución de grupos es buena
Es impuesta por el equipo	Es imparte de manera externa
Cuenta con pocos principios y su control es bajo	Sus procesos de controlan por medio de políticas y normas

Adaptado por: Minchala J, Crespo J.

Al tener claro la gran diferencia que hay entre estas metodologías, no cabe duda que las Metodologías Ágiles se acoplan mejor para el desarrollo del proyecto, es por ello que las metodologías ágiles son creadas a través de la queja de muchos desarrolladores, ya que

las metodologías tradicionales no contaban con la fluidez que se requería, es por ello que el Manifiesto para el Desarrollo Ágil de software (©2001) declaro fundamentos para tener un enfoque ágil, dando salida a las metodologías ágiles [34]. Entre las metodologías ágiles que mejor se acoplan al trabajo están:

Tabla 8. Metodologías Agiles

Metodologías Agiles				
	Kanban	Lean	Programación Extrema	Scrum
Trabajo en equipo	X	X	X	X
Existe comunicación constante con el cliente			x	x
Tolerante a cambios	X		x	X
Desarrollo iterativo	x	x	x	x

Adaptado por: Minchala J, Crespo J.

Al tener muchas opciones de metodologías ágiles vistas en la tabla 8, tenemos que saber cuál es la más conveniente a usar ya que:

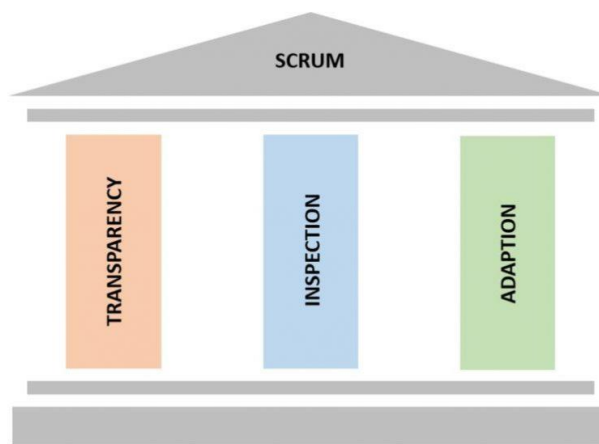
- **Kanban:** “hace uso de tarjetas para gestionar, de manera visual, la realización de determinados procesos”[35]. Al hacer uso de tarjetas, hace que esta metodología sea más fácil a la hora de su implementación, teniendo en cuenta que Kanban otorga tres puntos de vista; lo que hace ahora; lo que ha terminado y lo que se tendrá que hacer a continuación, esto se da gracias a la información recolectada que se transcribe en tarjetas que seguidamente serán colocadas en una tabla para saber en qué parte del proyecto se está situado [36]. Lo cual genera objetivos como:
  - Evitar una sobrecarga de producción.
  - Manejar de una buena manera los inventarios.

- Hacer un producto en un mínimo proceso.
- Controlar el flujo de material.
- **Lean:** Caracterizada por ser una metodología que se enfoca en la entrega del proyecto en un corto periodo de tiempo con un equipo cualificado para el trabajo, esto está conformado por un pequeño grupo de personas que deben estar constantemente relacionadas con lo que se hace, ya sea en aprendizaje y reacciones rápidas en el proceso de trabajo [35].
- **Programación extrema:** Metodología de XP (Programación extrema), se centra en reforzar el trabajo en equipo con el fin de agilizar los procesos encargados para cada equipo, donde se fortalecerá el aprendizaje que genera un buen ambiente en el ámbito laboral [37]. XP hace que exista una constante comunicación entre los clientes y los desarrolladores, siendo una buena práctica para saber lo que se requiere para implementar las distintas soluciones a un problema o generar los cambios pertinentes. “XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico”[37].
- **Scrum:** La metodología Scrum es una de las metodologías más usadas a nivel mundial, ofreciendo la integración de buenas prácticas para ponerla en marcha en un trabajo de equipo colaborativo, dando un mejor objetivo para tener buenos resultados, esto permite que el usuario colabore en el proyecto dando su valoración para realizar algún cambio o agregar apartados.

Una vez repasado las metodologías Agiles más usadas, procedemos a escoger la metodología Scrum por la asignación de roles en un equipo y buen manejo del desarrollo, teniendo a los siguientes puestos:

- ✓ El Product Owner: Su rol es encargarse de agrega un valor extra al producto, además de ser representante de sus clientes y de los intereses que presente [38].
- ✓ El Development Team: Es el encargado de organizar las tareas del equipo de desarrollo del producto y darle un incremento al mismo [38].
- ✓ El Scrum Master: Su rol es ser el líder del equipo Scrum, creando procesos del producto que servirá para alcanzar el objetivo propuesto, por medio de constantes revisiones de los resultados obtenidos de cada Sprint en Scrum [38].

En enfoque de Scrum va orientado al feedback que esta soportado por tres pilares como se ve en la figura 8.



*Figura 8. Pilares Scrum*

### 2.3.2 Fases de un Scrum

Scrum cuenta con 5 fases que muestran el funcionamiento de los procesos, donde cada etapa será parte de las exigencias que plantea los líderes de los proyectos, que a la vez deberán cumplir con el plazo de tiempo para la entrega del trabajo o proyecto. Los pasos de un Scrum son:

### 2.3.2.1 *Inicio*

Esta fase es la primera en actuar, encargándose del estudio y análisis del proyecto a realizar para los sprint. Hay 3 tipos de preguntas que suceden en la fase inicial y son; Que, Como y Cuando lo quiero, siendo las principales constantes a identificar. Para tener un grupo donde que fluyan las ideas, es necesario que este integrado por un grupo de 3 a 5 personas como máximo [39].

### 2.3.2.2 *Planificación*

En esta fase se debe tener en cuenta las historias de usuario, con el fin de comprender qué tipo de producto desea, otro aspecto importante es la identificación de las tareas y por último se debe crear un sprint backlog, ya que la planificación es importante para tener una buena administración en los proyectos que se van realizando [39].

### 2.3.2.3 *Implementación*

En esta fase entra la sala de reuniones que se usa para la discusión del sprint para obtener optimizaciones dentro del desarrollo del trabajo, para su implementación se debe realizar entregables, Daily Stand-Up y obtener una financiación del backlog. Se debe tener en cuenta que una vez ya finalizado el proyecto, este no podrá ser cambiado a última hora [39].

### 2.3.2.4 *Revisión*

En esta fase de revisión se verifican los procesos planteados con el fin de generar opiniones positivas o constructivas acerca del trabajo realizado con el fin de aportar una solución o satisfacción del mismo. Se siguen pasos para demostrar el sprint planteado con su respectiva su retrospectiva [39].

### 2.3.2.5 *Lanzamiento*

Es la parte final de las 5 fases de la metodología Scrum en la que el proyecto será lanzado o entregado con el fin de efectuar el envío de los entregables [39].

#### Sprint

El sprint se le considera un Time-box por tener un tiempo menor igual a un mes, el cual se establece una reunión en la cual se planifican las actividades a realizar en el tiempo del sprint, el equipo tiene que trabajar para crear el producto que será entregado o culminado [40].

## Capítulo 3

### 3.1. Diseño del Sistema

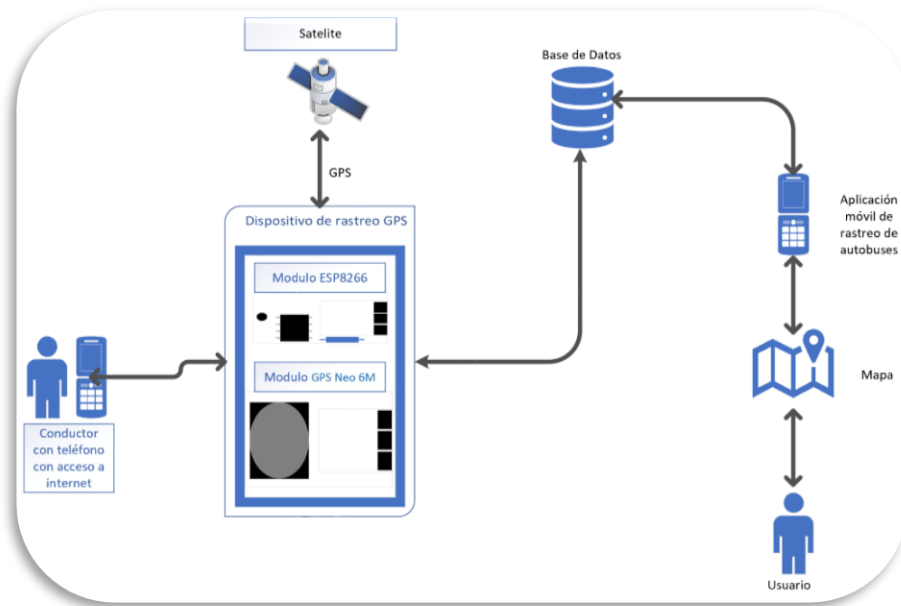
En esta sección se lleva a cabo el diseño del dispositivo de rastreo y la aplicación para visualizarlo, describiendo sus respectivas funciones.

### 3.2. Descripción general

Se diseñó un sistema de rastreo haciendo uso de los Módulos ESP8266 y Modulo GPS Neo 6M que sirve como ayuda para la proporción de datos mostrados en la aplicación de rastreo. La información que se obtenga del Dispositivo de rastreo se almacena en una base de datos, siendo datos que se mostrara en la aplicación del usuario, teniendo la posibilidad de ver los datos de la ubicación del vehículo en movimiento en tiempo real, las rutas que sigue, la distancia de un bus a una parada de llegada y ver paradas cercanas para el usuario.

En la figura 9, se muestra la arquitectura que será como guía del desarrollo del sistema de rastreo y del dispositivo.





*Figura 9. Arquitectura del Sistema*

Este será el proceso que se lleva a cabo para el desarrollo del dispositivo de rastreo como el de la aplicación, tomando en cuenta que el usuario deberá tener la aplicación para poder visualizar su transporte público, esto gracias al servidor que estará recibiendo los datos del dispositivo de rastreo, el dispositivo dependerá del usuario chofer donde deberá brindar una conexión de internet al dispositivo para que este se conecte a la red satelital.

### 3.2.1. Especificaciones de los requerimientos del sistema

Para esta parte es necesario tener unos requisitos a seguir, ya sea para servidor web como en el aplicativo móvil, por ello se procederá a realizar:

- a) Requisitos funcionales: los requisitos funcionales establecerán la funcionalidad del servidor como el de la aplicación [41].
- b) Requisitos no funcionales: realiza descripciones de la forma de comportarse del sistema, teniendo en cuenta la captura de las restricciones de cómo debe actuar un sistema [41].

- c) Requisitos técnicos: se toma en cuenta que tipos de tecnología se harán uso de una manera específica [41].

Los requerimientos que se toman en cuenta son: Servidor Web, Aplicativo Móvil y Dispositivo de rastreo.

### *3.2.1.1. Requerimientos funcionales*

#### **A. Servidor Web**

En nuestro servidor web debe ser capaz de realizar operaciones CRUD, con el fin de poder hacer modificaciones que se verán en ese momento de actualizar datos, por ello es importante tener a un administrador de base de datos que pueda llevar la gestión de la información que este llegando al servidor.

#### **B. Aplicativo Móvil**

El aplicativo móvil deberá ser capaz de tener acceso a internet para poder ver la situación actual del vehículo, además de ser gratuito consta con una buena seguridad OAuth 2.0 que ofrece Google para la seguridad de los usuarios, además de mostrar las paradas, rutas y tiempo de los autobuses, con el fin de establecer una buena comunicación a los ciudadanos.

#### **C. Dispositivo de Rastreo**

Este al integrar módulos que servirán para el envío de información deberá ser capaz de tener una conexión a internet para enviar los respectivos datos a la base, estos están compuestos por: velocidad, latitud, longitud, fecha y hora, esta información se verá reflejado en el dispositivo móvil.

### 3.2.1.2. *Requerimientos no funcionales*

En los requerimientos no funcionales es importante tener en cuenta 3 aspectos primordiales: la calidad, el alcance que tendrá y la operación, en algunos casos no se incluirá algunos aspectos. Con el propósito de generar una descripción de los grupos ya mencionados.

#### **A. Servidor Web**

##### 1) Calidad

- Los puntos primordiales de un servidor web es la seguridad, en este caso los campos deberán ser validados.
- La precisión de los datos es importante para que no existan datos erróneos reflejados en la aplicación
- Tener una buena gestión para obtener datos de manera rápida.

##### 2) Alcance

- El servidor web deberá ser capaz de almacenar todos los datos que reciba del dispositivo, además de tener la información de los buses y otro tipo de información relevante.
- Al momento de enviar y recibir los datos necesarios, este deberá tener una rápida respuesta y reducir la latencia que ocurra en entre dispositivos y servidor.

##### 3) Operación

- En la operación el servidor debe ser capaz de almacenar la información que reciba, sincronizando y mostrando los datos con los usuarios que tengan la aplicación.

#### **B. Aplicativo Móvil**

##### 1) Calidad

- El usuario deberá ser capaz de interactuar con la aplicación sin tener complicaciones.
  - El mantenimiento de una aplicación es importante para corrección de errores o bugs.
  - La compatibilidad con celulares de gama baja capaces de ejecutar la aplicación sin problemas.
- 2) Alcance
- Deberá realizar procesos con mayor fluidez.

### **C. Dispositivo de Rastreo**

- 1) Calidad
- Tendrá que incluir módulos capaces de transmitir y recibir información en cualquier lugar.
  - El dispositivo deberá ser accesible en precio y calidad para ser colocado en todos los transportes públicos.
- 2) Alcance
- La precisión del dispositivo deberá ser reflejada en la aplicación sin tener un radio mayor de 5 metros a la redonda del punto del bus.
- 3) Operación
- Transmite su latitud y longitud transformándolos en datos que serán enviados al servidor.

#### *3.2.1.3. Requerimientos Técnicos*

##### **A. Servidor Web**

- Una base de datos gratuita

##### **B. Aplicativo Móvil**

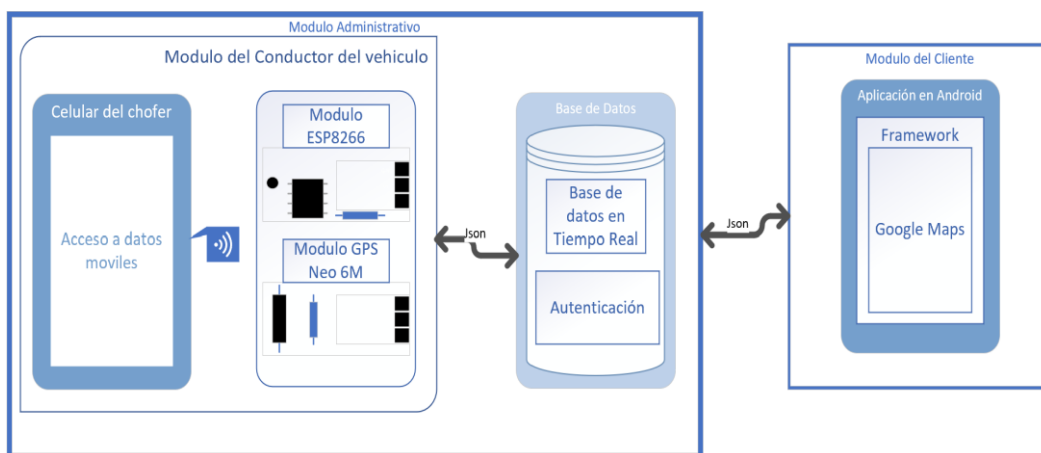
- Disponible para dispositivos Android con un Sistema Operativo (SO) 5.0 en adelante.

### C. Dispositivo de Rastreo

- Disponible a precios accesible para su desarrollo e implantación.

### 3.3.Funcionalidad del sistema de geolocalización de autobuses

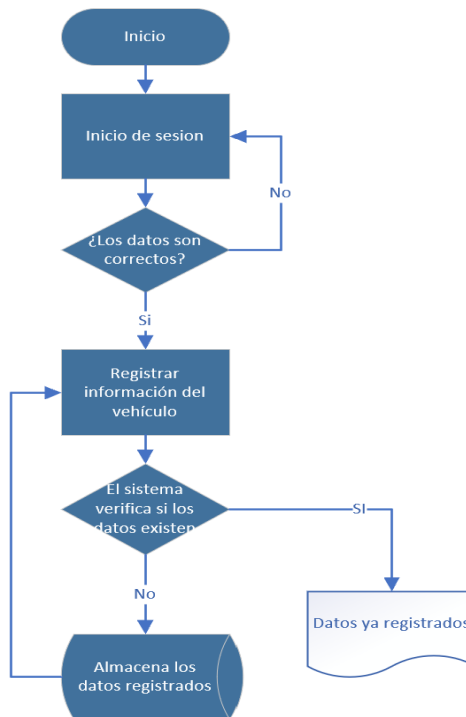
Para entender cómo funciona el sistema de seguimiento de los autobuses, el sistema será dividido en tres módulos, siendo los siguientes: Módulo del Chofer, Módulo Administrativo y Módulo del cliente, como se observa en la siguiente figura 10.



*Figura 10. Vista del diagrama general*

#### Módulo Administrativo

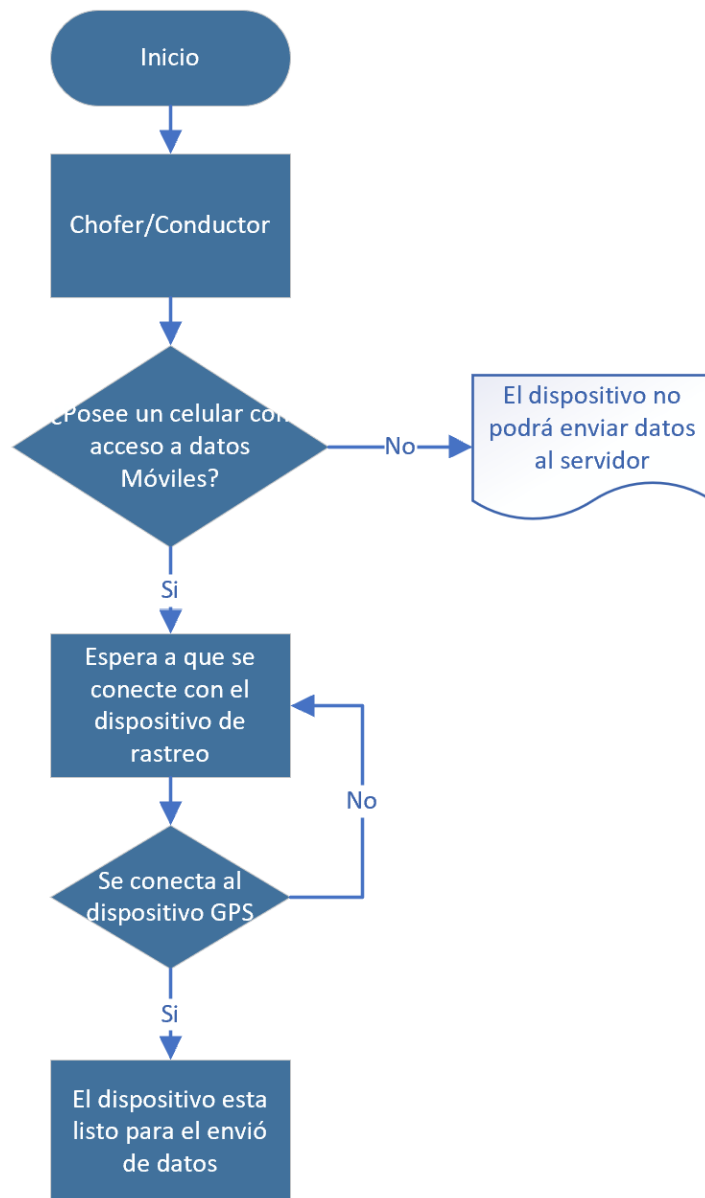
El módulo administrativo es el encargado de actualizar la información de los autobuses y rutas que estos tomen, por medio de la administración de podrán modificar los parámetros que sean relevantes para la aplicación, dando al administrador una vista general de que tipo de información se va a manejar. En la siguiente figura 11 se puede observar un diagrama de flujo demostrando el proceso de administración.



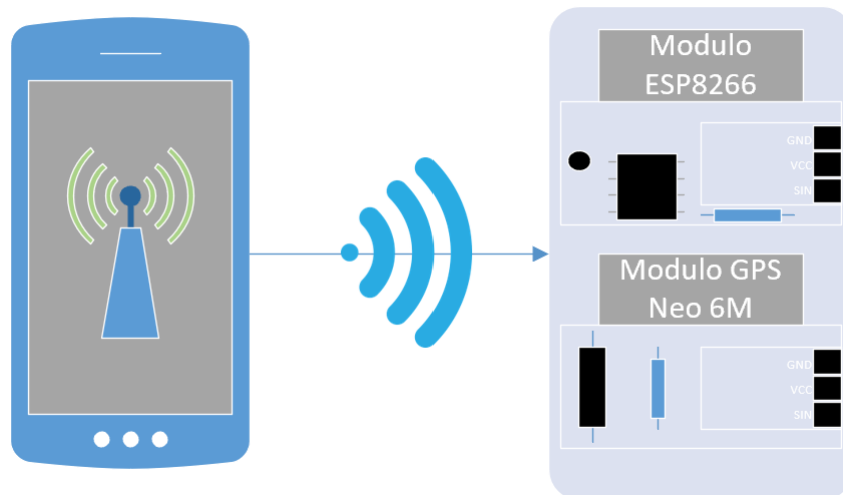
*Figura 11. Mapa de procesos del administrador*

### Módulo del Piloto del vehículo

En este módulo el conductor será el responsable en brindar una conexión de red al dispositivo de rastreo, siendo fundamental para que los datos sean enviados al servidor cuando el chofer se ponga en marcha con el vehículo, para entenderlo mejor en las figuras 12 y 13 se ve el proceso que este sigue.



*Figura 12. Mapa de Procesos del Conductor*



*Figura 13. Visualización de la actividad del conductor*

#### Módulo del consumidor

El módulo del consumidor corresponde al posicionamiento vehicular y vista general de las rutas y paradas de autobús, logrando crea un sistema interactivo para el usuario. Al hacer uso de una API de Google nos permite usar los servicios de Mapas, por otro lado, al tener un dispositivo de rastreo que envíe información al servidor, los datos serán reflejados en forma de objeto mostrándose en el mapa de la aplicación del usuario, teniendo un punto exacto del vehículo en movimiento, como se observa en las figuras 14 y 15.

#### Componentes del Módulo del Usuario

- ✓ Integra los frameworks que son los encargados de mostrar el mapa.
- ✓ Integra los botones de navegación como menú.
- ✓ Integra marcadores de iconos que muestra el vehículo en movimiento y las paradas de los buses.
- ✓ Contiene el método de Polyline, que se usa para el mapeo de las rutas de los vehículos.
- ✓ Contiene el método de búsqueda de lugares.



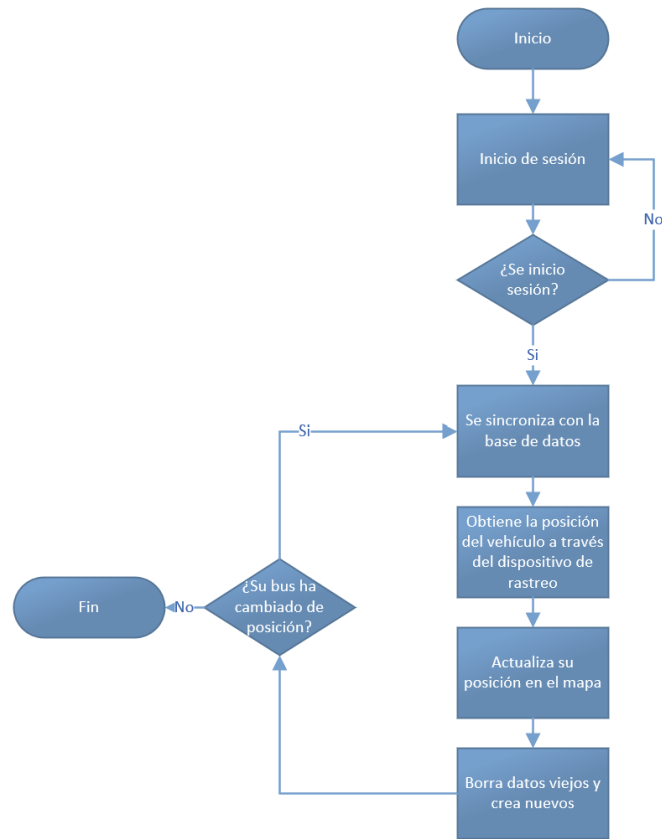


Figura 14. Mapa de Procesos del Usuario

Boceto del Usuario



Figura 15. Vista de las actividades que puede hacer el usuario

## Capítulo 4

### 4.1.Propuesta

En este capítulo se da paso a la creación de la propuesta, aplicando la metodología Scrum, mostrando como se fue creando todo el proceso para realizar la aplicación y que actividades fueron aplicadas.

De igual manera, se muestra el resultado final de la creación del software y hardware, hecho por el equipo de trabajo que está conformado por 2 personas, haciendo uso de las metodologías planteadas anteriormente para obtener un resultado satisfactorio.

### 4.2.Desarrollo de la Propuesta

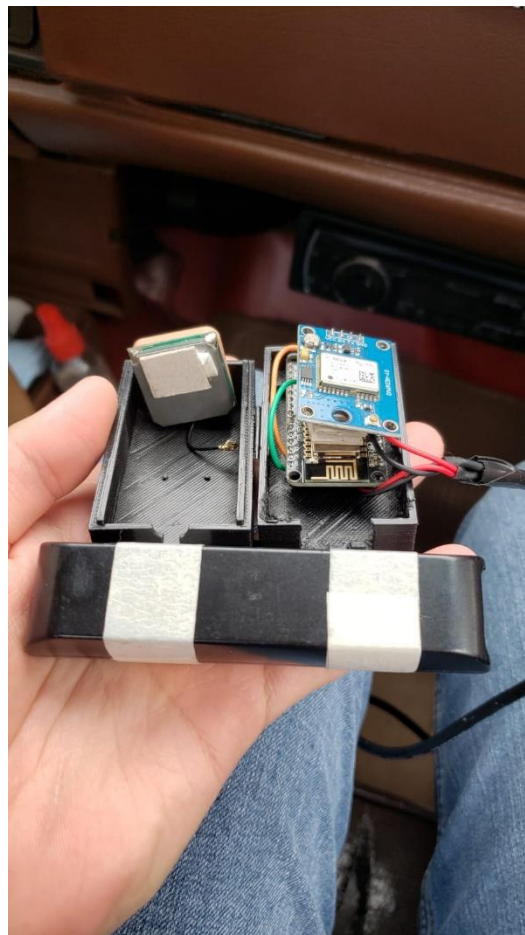
Para desarrollar el sistema de información para el rastreo de los buses de transporte público de la empresa San Francisco por medio de un dispositivo GPS, se hace uso de un hardware accesible en propiedades requeridas y el software se creó haciendo uso componentes de desarrollo para aplicaciones móviles. Como primer punto del desarrollo será:

- ❖ Configuración del dispositivo de rastreo por medio de Arduino, mismo que servirá para el envío de datos al servidor.
- ❖ Desarrollo de un servicio web con Sprintboot, mismo que servirá para el cambio mutuo de datos que transmitirá el dispositivo de rastreo al servidor.
- ❖ Uso de la base de datos PostgreSQL para almacenar los datos que serán recibidos del servicio web.
- ❖ Una aplicación para dispositivos Android que será desarrollado en Android Studio, donde mostrará al usuario la Ubicación de los autobuses en tiempo real, las paradas, las rutas/ recorridos y distancia del bus al punto del usuario.

- ❖ Desarrollo del sistema haciendo uso de la metodología Scrum, la cual partes del proyecto será dividido en pequeñas partes que se acoplen a los Sprint, ubicando las historias de usuario a especificar, clasificándolas en un rango de las más prioritarias a las menos prioritarias.

#### 4.2.1. Desarrollo del dispositivo de Rastreo

Antes de comenzar con el primer Sprint, es necesario entender el Hardware del dispositivo de rastreo y cómo funciona cada uno de sus sistemas. Como se puede ver en la figura 16, integra los módulos ya mencionados en el Cap.2 siendo una de las piezas fundamentales que este requiere para el envío de datos.



*Figura 16. Dispositivo de Rastreo.*

### 4.2.2. Creación del Dispositivo

Antes de iniciar, primero debemos saber las entradas que tiene el módulo NEO 6M, ya que, si se conecta mal un cable, este puede generar problemas en la placa, es por ello que en la tabla 9 se detalla que tipo de entradas tiene.

Tabla 9. Pinout del módulo NEO 6M

TX	TX es el PIN de transmisión, se usa para la comunicación en serie.
RX	RX es el PIN receptor, se usa para la comunicación en serie.
GND	Es el PIN que servirá como conexión de tierra del ESP8266.
VCC	Es el Pin encargado de recibir una energía de 3.3 V del módulo ESP8266.

Adaptado por: Minchala J, Crespo J.

Primero se debe hacer las conexiones correspondientes del ESP8266 al NEO 6M, iniciamos haciendo uso de los cables DuPont de hembra/macho que serán fundamentales para la conexión de placas, quedando de la siguiente manera:

1. En la placa ESP8266 conectamos D1 al RX del NEO 6M.
2. Se conecta D2 de la placa ESP8266 al TX del NEO 6M.
3. Se conecta GND de ambas placas por igual.
4. Se conecta EVE de la placa ESP8266 al VCC del NEO 6M.

En la figura 17 se muestra un esquema del resultado.

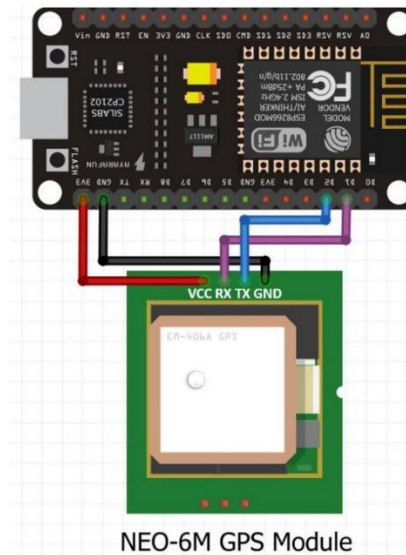


Figura 17. Esquema de conexión de módulos

Al tener armado el dispositivo de rastreo se invirtió un aproximado de 30 dólares viendo sus valores en la tabla 10, al ser elementos de bajo coste nos permiten realizar cambios en caso de quemaduras en los circuitos y reemplazarlos con nuevos módulos.

Tabla 10. Precio de los componentes

Precio de los componentes del dispositivo de rastreo	
1 módulo ESP8266	12 dólares
1 módulo NEO 6M	12 dólares
10 cables DuPont	10 centavos cada uno

Adaptado por: Minchala J, Crespo J.

Al tener armado el dispositivo de rastreo armado se procede a su configuración, dando inicio a la implementación de Scrum.

#### 4.3. Implementación de Scrum

Para el desarrollo de la metodología se debe aplicar los respectivos roles, el Product Owner es el encargado de saber acerca del proyecto, el Master Scrum es responsable

de ser el autor del proyecto y por último el development Team se encargará del desarrollo del mismo.

Al aplicar Product Backlog, este integra las historias de usuario que serán divididas en Sprint, expresando que son pequeñas definiciones que se integra en el proyecto de software. A continuación, se muestra el Product Backlog con la siguiente estructura y visualizado en la tabla 11:

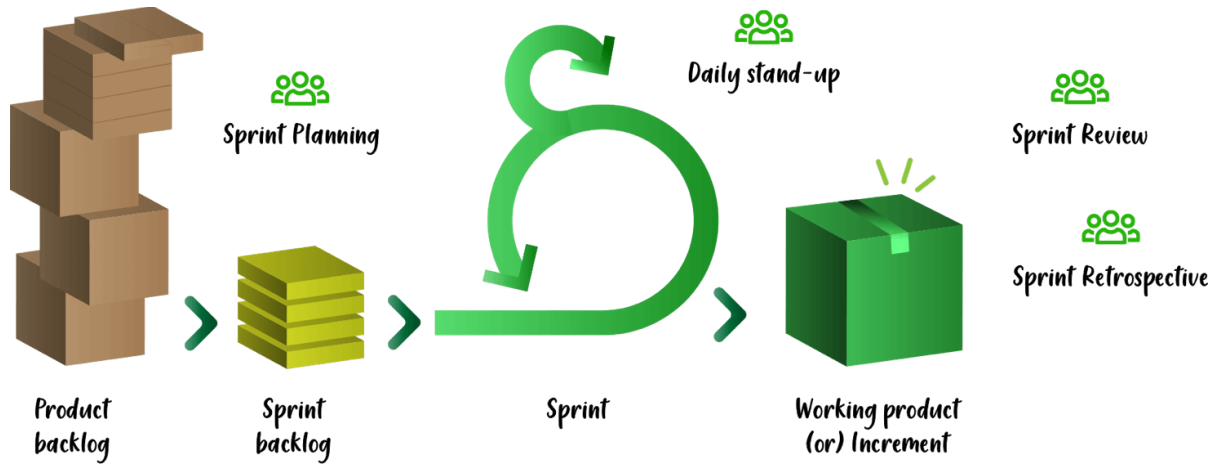
- ❖ Numeración: La numeración identificará la historia de usuario en la se encuentra.
- ❖ Historia de Usuario: Se identificador del nombre de usuario.
- ❖ Estimación: El tiempo que se toma para realizar el trabajo dividido.
- ❖ Prioridad: Priorizará la historia de usuario que tenga más importancia, siendo 5 muy prioritario y 1 no muy prioritario.

Tabla 11. Product Backlog.

<b>Product Backlog</b>		
<b>Numero</b>	<b>Historia de Usuario</b>	<b>Prioridad</b>
<b>1</b>	Configuración de Módulos ESP8266 Y GPS NEO 6M	5
<b>2</b>	Desarrollo de un Servicio Web e implementación de la Data Base	4
<b>3</b>	Inicio de sesión de Usuario	3
<b>4</b>	Localización de Autobuses	2
<b>5</b>	Mapeo de Rutas	1
	Parada de Autobuses	

Adaptado por: Minchala J, Crespo J.

## Proceso Scrum



*Figura 18. Proceso Scrum*

Al tener hecha la tabla del Product Backlog, es importante asegurarse de la distribución de cada historia de usuario, ya que, al momento de la entrega de un sprint, este debe estar culminado, caso contrario puede generar problemas a lo largo del proyecto, además es necesario priorizar las historias de usuario más importantes por el esfuerzo que requerirá hacerlos, por otro lado, se debe llevar a cabo las tareas del Product Backlog para saber a detalle cada su función.

### 4.3.1. Sprint 1

Tabla 12. Historia de Usuario 1

Historia de Usuario	
Numero: 1	Prioridad: 5
Nombre de historia de usuario: Configuración de Módulos ESP8266 Y GPS NEO 6M	Responsable: Crespo j, Minchala J
Descripción: Como primera demostración de la historia de usuario correspondiente a la configuración de Módulos ESP8266 Y GPS NEO 6M, la cual se les debe asignar las librerías correspondientes para su funcionamiento, establecer una ID que identifique el dispositivo, agregar el controlador del GPS, establecer la conexión con el servidor, agregar un usuario y contraseña para que se pueda conectar a una red de internet.	

Adaptado por: Minchala J, Crespo J.

#### 4.3.1.1. Especificación del Sprint

Se procede a iniciar las tareas del primer Sprint tabla 12, donde se definirá la duración del trabajo a realizar, teniendo un tiempo de aproximadamente 137 horas.

#### 4.3.1.2. Sprint Backlog

Como primera historia de usuario corresponde a la configuración de Módulos ESP8266 Y GPS NEO 6M, integrando una lista de tareas que se proceden a realizar en cada Sprint con el fin de calcular el avance del proyecto y tener el proyecto ya culminado para entregarlo observándose en las tablas 13, 14 y 15.



Tabla 13. Sprint Backlog 1

<b>Product Backlog</b>		
<b>Numero</b>	<b>Historia de Usuario</b>	<b>Prioridad</b>
<b>1</b>	Configuración de Módulos ESP8266 Y GPS NEO 6M	5

Adaptado por: Minchala J, Crespo J.

Tabla 14. Tareas del Primer Sprint

<b>Tareas del Primer Sprint</b>			
<b>Sprint</b>	<b>Tiempo del proyecto en horas</b>	<b>Historia de Usuario</b>	<b>Estimación</b>
1	137	Configuración de Módulos ESP8266 Y GPS NEO 6M	
<b>S.B</b>	<b>Tareas</b>	<b>Situación</b>	31/10/22 - 8/11/22
1	Importar librerías para controlador	Terminado	
	Asignación de Usuario y contraseña para Arduino	Terminado	
	Método para habilitar el internet	Terminado	
	Asignación del servidor	Terminado	
	Trasformación de datos de Arduino	Terminado	

Adaptado por: Minchala J, Crespo J.

Tabla 15. Tareas Culminadas del primer Sprint

Tareas Culminadas	
Tareas	Fechas
Importar librerías para controlador	31/10/22
Asignación de Usuario y contraseña para Arduino	01/11/22
Método para habilitar el internet	02/11/22
Asignación del servidor	03/11/22
Trasformación de datos de Arduino	07/11/22

Adaptado por: Minchala J, Crespo J.

#### 4.3.1.3. Diagrama de Gantt

Al usar el diagrama de Gantt, nos muestra de manera detallada las fechas de cada tarea, dándonos el resultado del avance del trabajo hasta una fecha límite, a continuación, en la figura 19, se observa cómo se fue clasificando las fechas por medio del diagrama ya mencionado.



Figura 19. Diagrama d Gantt Sprint 1

#### 4.3.1.4. Explicación del Sprint

Como justificante del primer Sprint se ha desarrollado las tareas asignadas como se observa en las Ilustraciones;

- ❖ En la figura 20 se ve las librerías que son necesarias para que el dispositivo funcione.
- ❖ En la figura 21 se ve la asignación de usuario como la de contraseña que son parámetros necesarios para la conexión a internet del dispositivo.
- ❖ En la figura 22 se ve que método se usó para que el dispositivo tenga habilitado la red Wifi.
- ❖ En la figura 23 se observa la asignación del servidor al que ira dirigido los datos del dispositivo y finalmente en la figura 24 se ve los datos que se usan para enviar al servidor y su transformación de a String.

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include <TinyGPS++.h>
#include <base64.h>
```

*Figura 20. Librerías y Controladores.*

```
TinyGPSPlus gpsplus;
TinyGPS gps;
SoftwareSerial ss(D1, D2); //rx=4 tx=3 ESP8266 rx=D1 tx=D2
const char* ssid = "GPS Wifi";
const char* password = "1234567899";
```

*Figura 21. Asignación de usuario y contraseña.*

```

void setup() {
  Serial.begin(115200);
  ss.begin(9600);
  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());

  Serial.println("Timer set to 5 seconds (timerDelay variable), it will take 5 seconds before publishing the first reading.");
}

```

*Figura 22. Método para habilitar Wifi.*

```

//Your Domain name with URL path or IP address with path
const char* serverName = "http://192.168.0.192:8080/api/buses";

```

*Figura 23. URL del servidor.*

```

// If you need an HTTP request with a content type: application/json, use the following:
http.addHeader("Content-Type", "application/json");
String auth = "AdminBuses:Jicg8266";
auth.replace("\n", "");
auth.replace("\r", "");
http.addHeader("Authorization", "Basic " + base64::encode(auth));

float speedKnots = gpsplus.speed.kmph();
float speedKPH = speedKnots * 1.852;
float latitude = gpsplus.location.lat();
float longitude = gpsplus.location.lng();
int year = gpsplus.date.year();
int month = gpsplus.date.month();
int day = gpsplus.date.day();

String Lat = String(latitude, 8);
String Lon = String(longitude, 8);
String velo = String(speedKPH, 2);
String dia = String(day);
String mes = String(month);
String ano = String(year);
int httpResponseCode = http.POST("{\"nombus\":\"1\", \"latitud\":\"" + Lat + "\", \"longitud\":\"" + Lon +
 "\", \"velobus\":\"" + velo + "\", \"fecha\":\"" + dia + "-" + mes + "-" + ano + "\"}");

```

*Figura 24. Transformación de datos.*

#### 4.3.1.5. Resultado del Sprint

A continuación, en la siguiente figura 25 se muestra en la consola de Arduino los datos que se obtienen del dispositivo en tiempo real. Estos datos serán los que se muestren en forma de objeto dentro de la aplicación para ver la ubicación del vehículo.

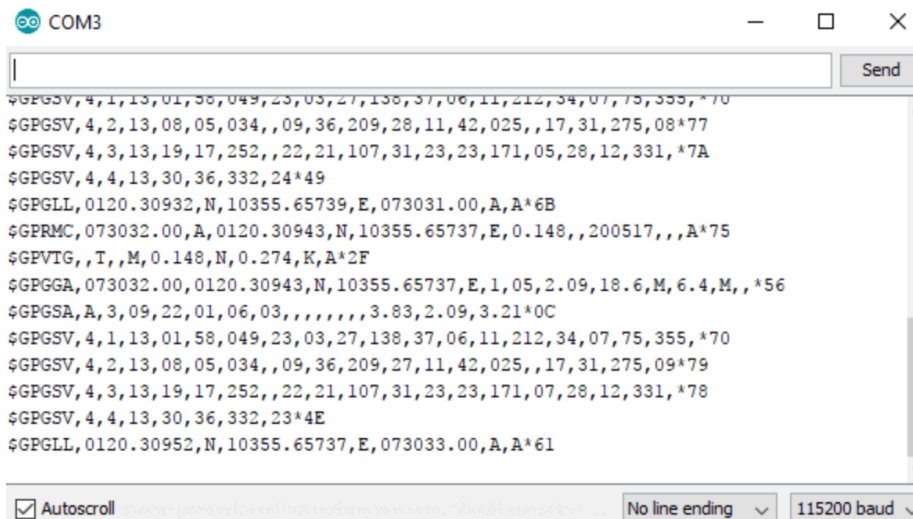


Figura 25. Coordenadas obtenidas del dispositivo.

### 4.3.2. Sprint 2

Tabla 16. Historia de usuario 2

Historia de Usuario	
Numero: 2	Prioridad: 4
Nombre de historia de usuario: Desarrollo de un Servicio Web e implementación de la Data Base	Responsable: Crespo j, Minchala J
Descripción: En este segundo Sprint se procedió a realizar el Desarrollo de un Servicio Web que sea de ayuda para el envío y almacenado de datos, en este caso para que el dispositivo de rastreo sea el que los envíe, almacenándolas en una base de datos.	

Adaptado por: Minchala J, Crespo J.

#### 4.3.2.1. Especificación del Sprint

Es plantea la misma ideología del primer Sprint tomando en cuenta las tareas que se deberán realizar en este product backlog tabla 16, teniendo un tiempo de aproximadamente 13 días.

#### 4.3.2.2. *Sprint Backlog*

Como segunda historia corresponde al Desarrollo de un Servicio Web tabla 17, que de igual manera se detalla las tareas hechas en las tablas 18 y 19, para saber que procesos fueron implementado y calculando el tiempo que se ha invertido en el Sprint.

Tabla 17. Product Backlog 2

<b>Product Backlog</b>		
<b>Numero</b>	<b>Historia de Usuario</b>	<b>Prioridad</b>
<b>2</b>	Desarrollo de un Servicio Web e implementación de la Data Base	<b>4</b>

Adaptado por: Minchala J, Crespo J.

Tabla 18. Tareas del segundo Sprint.

<b>Tareas del Segundo Sprint</b>			
<b>Sprint</b>	<b>Tiempo del proyecto en horas</b>	<b>Historia de Usuario</b>	<b>Estimación</b>
2	309	Desarrollo de un Servicio Web	
<b>S.B</b>	<b>Tareas</b>	<b>Situación</b>	
2	Crear un proyecto en SprintBoot con el nombre de gbuslocation	Terminado	09/11/22 - 30/11/22
	Creación de la clase WebConfig	Terminado	
	Creación de la clase BusControlador	Terminado	
	Creación de la clase SecurityConfiguration y método Spring Security	Terminado	
	Creación de la clase Datos Bus	Terminado	
	Creación del repositorio	Terminado	

Adaptado por: Minchala J, Crespo J.

Tabla 19. Tareas culminadas del Segundo Sprint.

<b>Tareas Culminadas</b>	
<b>Tareas</b>	<b>Fechas</b>
Crear un proyecto en SprintBoot con el nombre de gbuslocation	09/11/22
Creación de la clase WebConfig	10/11/22
Creación de la clase BusControlador	14/11/22
Creación de la clase SecurityConfiguration y método Spring Security	17/11/22
Creación de la clase Datos Bus	21/11/22
Creación del repositorio	24/11/22

Adaptado por: Minchala J, Crespo J.

#### 4.3.2.3. Diagrama de Gantt

Al usar el diagrama de Gantt para el segundo Sprint podemos ver como se ha distribuido las actividades con sus respectivas fechas, a continuación, en la figura 26 se observa cómo se fue avanzado con las tareas propuestas.

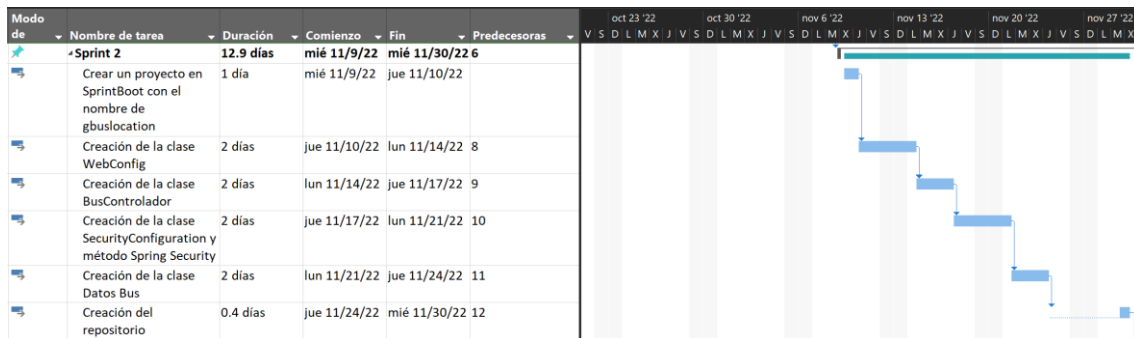


Figura 26. Diagrama de Gantt Sprint 2

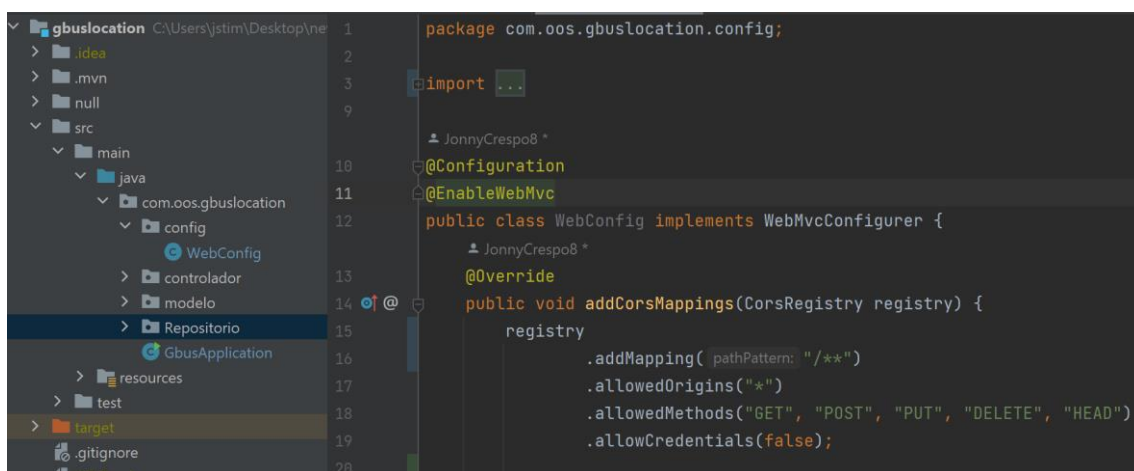
#### 4.3.2.4. Explicación del Sprint

Como segundo justificante del Sprint, podemos definir el desarrollo culminado de las tareas propuestas como se ve a continuación:

- ❖ En la siguiente figura 27 se observa la creación de la clase WebConfig que sirve como credenciales que usa para hacer el envío de datos.
- ❖ En las figuras 28 y 29 se observa la creación de la estructura de la base de datos, con su respectivo constructor que servirá para el envío de datos y los métodos get y set.
- ❖ En las figuras 30 y 31, se ve cómo se implementó la clase SecurityConfiguration, con el fin de ningún tipo de usuario pueda acceder a la URL de los datos y la implementación del método Sprint Security.



- ❖ En la figura 32 se observa la creación de la interfaz que sirve para recoger los datos que serán enviados a la clase BusControlador y en la figura 33 se observa la conexión a la base de datos.
- ❖ La base de datos está configurada para llegar a un cierto número de datos que almacenara los datos registrados por el dispositivo GPS, al llegar a la cifra de 500 datos, los datos se reiniciarán, es decir que iniciara en 1 son afectar la posición del bus.



```
1 package com.oos.gbuslocation.config;
2
3 import ..;
4
5
6
7
8
9
10 @Configuration
11 @EnableWebMvc
12 public class WebConfig implements WebMvcConfigurer {
13     JonnyCrespo8 *
14     @Override
15     public void addCorsMappings(CorsRegistry registry) {
16         registry
17             .addMapping( pathPattern: "/*")
18             .allowedOrigins("*")
19             .allowedMethods("GET", "POST", "PUT", "DELETE", "HEAD")
20             .allowCredentials(false);
21     }
22 }
23
24
25
26
```

*Figura 27. Clase WebConfig.*

```
1 package com.oos.gbuslocation.controlador;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17 JonnyCrespo8 *
18 @RestController
19 @RequestMapping("/api")
20
21
22
23
24
25
26
27
28
29 JonnyCrespo8
30 @GetMapping("/getbuses")
31 public List<DatosBus> buses()
32 {
33     return busRepositorio.findAll();
34 }
35
36 JonnyCrespo8 *
37 @PreAuthorize("hasAuthority('BASIC_AUTH')")
38 @PostMapping("/buses")
39 public DatosBus insertaNuevoBus(@Valid @RequestBody DatosBus bus)
40 {
41     return busRepositorio.save(bus);
42 }
```

Figura 28. Clase BusControlador.

```

1 package com.oos.gbuslocation.modelo;
2
3
4 import ...
5
6
7
8
9 @Entity
10 @Table(name="buslocation")
11 public class DatosBus {
12
13
14
15     @Id
16     @GeneratedValue(strategy = GenerationType.IDENTITY)
17     private Integer id;
18
19     @Column(name = "nombus",length= 60,nullable=false)
20     private Integer nombus;
21
22     @Column(name = "latitud",precision = 8, scale = 8,nullable=false)
23     private Float latitud;
24
25     @Column(name = "longitud",precision = 8, scale = 8,nullable=false)
26     private Float longitud;
27
28     @Column(name = "velobus",length= 60,nullable=false)
29     private Float velobus;
30
31     @Column(name = "fecha",length= 60,nullable=false)
32     private String fecha;
33
34 }

```

Figura 29. Clase DatosBus

```

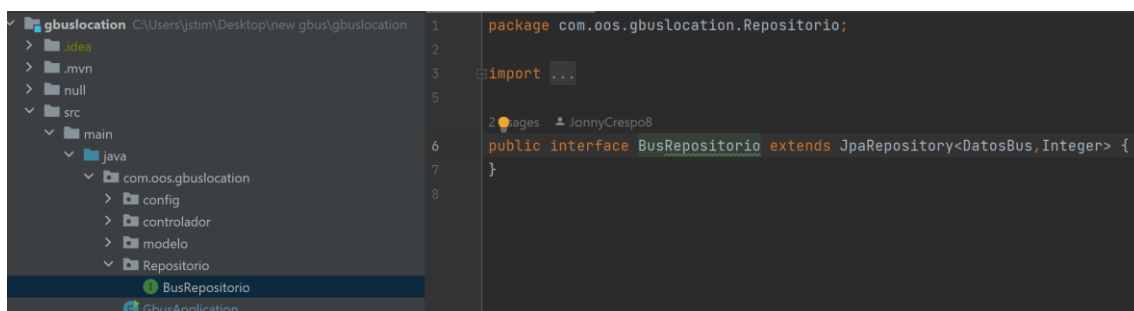
1 package com.oos.gbuslocation.controlador;
2
3 import ...
4
5
6
7
8
9
10
11
12 @Configuration
13 @EnableWebSecurity
14 public class SecurityConfiguration extends WebSecurityConfigurerAdapter {
15
16     @Override
17     protected void configure(HttpSecurity http) throws Exception {
18         /* http
19
20
21         http
22             .csrf().disable() HttpSecurity
23             .authorizeRequests() ExpressionUrlAuthorizationConfigurer<...>.ExpressionInterceptUriRegistry
24             .anyRequest().authenticated()
25             .and() HttpSecurity
26             .httpBasic();
27         }
28
29     @Autowired
30     public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
31         auth
32             .inMemoryAuthentication()
33             .withUser( username: "AdminBuses").password("{noop}Jicg8266").roles("USER");
34     }
35 }

```

Figura 30. Clase SecurityConfiguration.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

Figura 31. Método Security

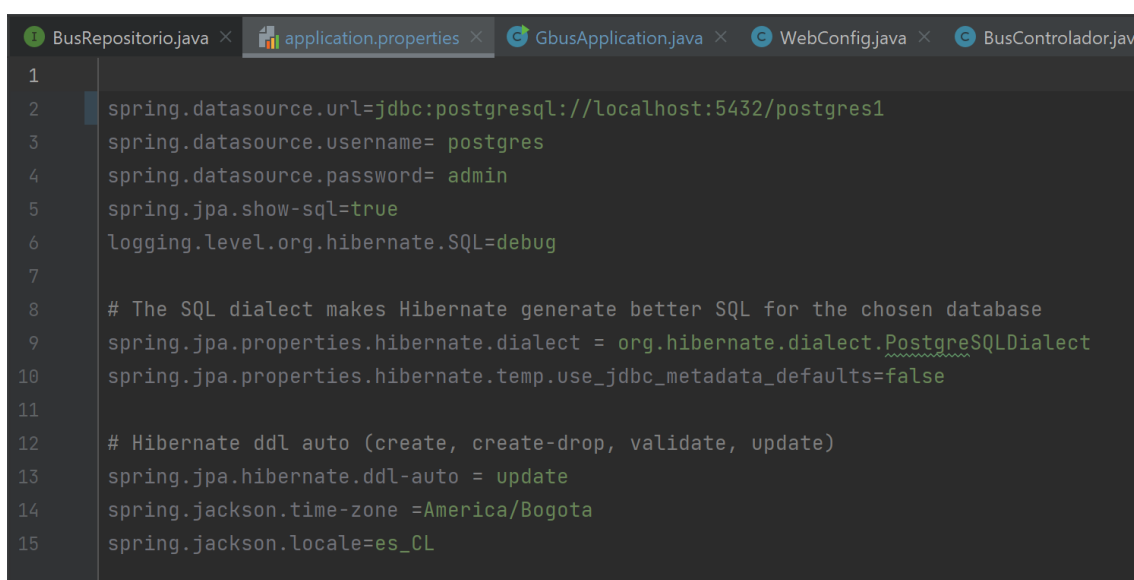


```
package com.oos.gbuslocation.Repositorio;

import ...

public interface BusRepositorio extends JpaRepository<DatosBus, Integer> {
}
```

Figura 32. Interfaz BusRepositorio.



```
1
2 spring.datasource.url=jdbc:postgresql://localhost:5432/postgres1
3 spring.datasource.username= postgres
4 spring.datasource.password= admin
5 spring.jpa.show-sql=true
6 logging.level.org.hibernate.SQL=debug
7
8 # The SQL dialect makes Hibernate generate better SQL for the chosen database
9 spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.PostgreSQLDialect
10 spring.jpa.properties.hibernate.temp.use_jdbc_metadata_defaults=false
11
12 # Hibernate ddl auto (create, create-drop, validate, update)
13 spring.jpa.hibernate.ddl-auto = update
14 spring.jackson.time-zone =America/Bogota
15 spring.jackson.locale=es_CL
```

Figura 33. Conexión a la base de datos

#### 4.3.2.5. Resultado del Sprint

A continuación, se observa los resultados correspondientes al Sprint 2 como se ve en las figuras 34, 35 y 36, demostrando la seguridad de la página y en servidor que en este caso es Heroku PostgreSQL.



### 4.3.3. Sprint 3

Tabla 20. Historia de Usuario 3

Historia de Usuario	
Numero: 3	Prioridad: 3
Nombre de historia de usuario: Inicio de sesión de usuario	Responsable: Minchala J, Crespo J.
Descripción: El sprint 3 corresponde al Inicio de sesión del usuario a la aplicación móvil donde seguido accederá a los mapas, además de integrar más funciones como cerrar sesión y el (onNavigationItemSelected) que sirve para navegar entre fragmentos.	

Adaptado por: Minchala J, Crespo J.

#### 4.3.3.1. Especificación del Sprint

Se sigue haciendo uso del sprint para saber que tipos de tareas corresponde a la creación del Inicio de sesión de usuario tabla 20, además de estimar un tiempo de 10 días para el desarrollo de todas las tareas.

#### 4.3.3.2. Sprint Backlog

Se detallan las tareas hechas en las tablas 22 y 23 con su respectivo tiempo de producción teniendo de cabecera al inicio de sesión de usuario en el Product backlog de la tabla 21.

Tabla 21. Product Backlog 3

Product Backlog		
Numero	Historia de Usuario	Prioridad
3	Inicio de sesión de usuario	3

Adaptado por: Minchala J, Crespo J.

Tabla 22. Tareas del tercer Sprint

<b>Tareas del Tercer Sprint</b>			
<b>Sprint</b>	<b>Tiempo del proyecto en horas</b>	<b>Historia de Usuario</b>	<b>Estimación</b>
3	252	Inicio de sesión de usuario	
<b>S.B</b>	<b>Tareas</b>	<b>Situación</b>	
3	Creación del Clase home	Terminado	01/12/22 - 18/12/22
	Agregar los servicios de Google play	Terminado	
	Usar los servicios de Firebase	Terminado	
	Uso de Imágenes para las actividades	Terminado	
	Creación de layout activity_main	Terminado	
	Creación de la clase Login	Terminado	
	Creación de layout activity_login	Terminado	
	Creación de cerrar sesión	Terminado	
	Creación de Layout activity_cerrar_sesion	Terminado	

Adaptado por: Minchala J, Crespo J.

Tabla 23. Tareas culminadas del tercer Sprint

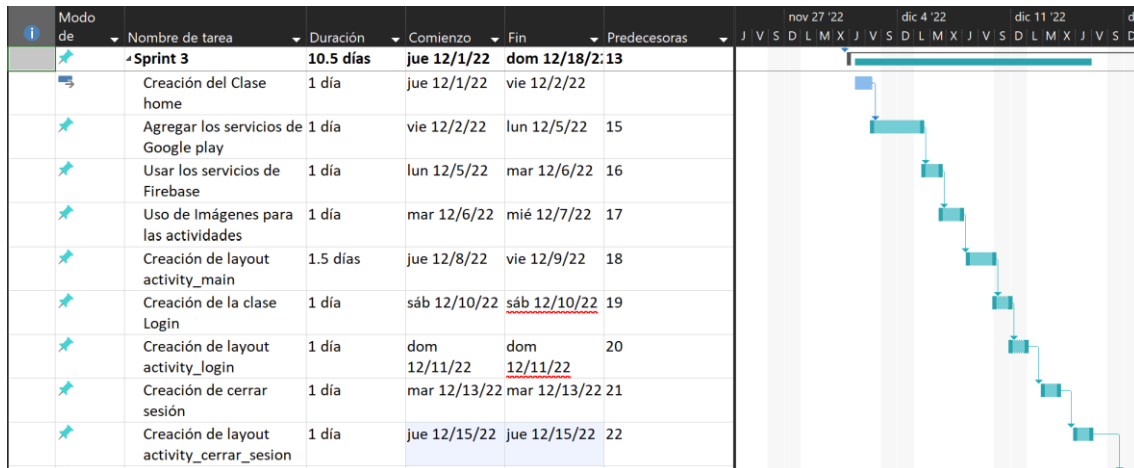
<b>Tareas Culminadas</b>	
<b>Tareas</b>	<b>Fechas</b>
Creación del Clase home	01/12/22
Agregar los servicios de Google play	02/12/22
Usar los servicios de Firebase	05/12/22
Uso de Imágenes para las actividades	06/12/22
Creación de layout activity_main	08/12/22
Creación de la clase Login	10/12/22
Creación de layout activity_login	11/12/22
Creación de cerrar sesión	13/12/22
Creación de Layout activity_cerrar_sesion	15/12/22

Adaptado por: Minchala J, Crespo J.

#### 4.3.3.3. Diagrama de Gantt

De igual manera se hace uso del diagrama de Gantt para el cálculo de tiempo de las actividades realizadas en el tercer Sprint, viéndose de la siguiente manera en la figura 37.





*Figura 37. Diagrama de Gantt Sprint 3*

#### 4.3.3.4. Explicación del Sprint

El tercer Sprint podemos observar las actividades realizadas que embarca a:

- ❖ En las primeras figuras 38, 39 y 40 es la clase MainActivity es la actividad principal donde está programado para movernos entre fragmentos.
- ❖ En las figuras 41, 42, 43 y 44 es la creación de la clase login, lo cual hace uso de OAuth 2.0 de Google como seguridad de la aplicación.
- ❖ En las siguientes figuras 45 y 46 es la creación de la clase cerrar sesión.

```

Login.java x MainActivity.java x
1 package com.example.gbus;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6     private BottomNavigationView bnvMenu;
7     private Fragment fragment;
8     private FragmentManager manager;
9
10
11 @Override
12 protected void onCreate(Bundle savedInstanceState) {
13     super.onCreate(savedInstanceState);
14     setContentView(R.layout.activity_main);
15     initView();
16     initValues();
17     initListener();
18 }
19
20 private void initView() { bnvMenu = findViewById(R.id.bnvMenu); }
21
22 private void initValues() {
23     manager = getSupportFragmentManager();
24     loadFirstFragment();
25 }
26
27

```

Figura 38. Clase MainActivity

```

36 }
37
38 private void initListener() {
39     bnvMenu.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
40         @Override
41         public boolean onNavigationItemSelected(@NonNull MenuItem item) {
42             int idMenu = item.getItemId();
43             switch (idMenu) {
44                 case R.id.menuira:
45                     fragment = Ira.newInstance();
46
47                     openFragment(fragment);
48                     return true;
49                 case R.id.menuParadas:
50                     fragment = Paradas.newInstance();
51                     openFragment(fragment);
52                     return true;
53                 case R.id.menuLineas:
54                     fragment = Rutas.newInstance();
55                     openFragment(fragment);
56                     return true;
57             }
58             return false;
59         }
60     });
61 }
62 private void openFragment(Fragment fragment) {
63     manager.beginTransaction()
64         .replace(R.id.fragmentContainer, fragment)
65         .commit();
66 }
67 private void loadFirstFragment() {
68     fragment = Ira.newInstance();
69     openFragment(fragment);
70 }
71 }

```

Figura 39. Clase MainActivity

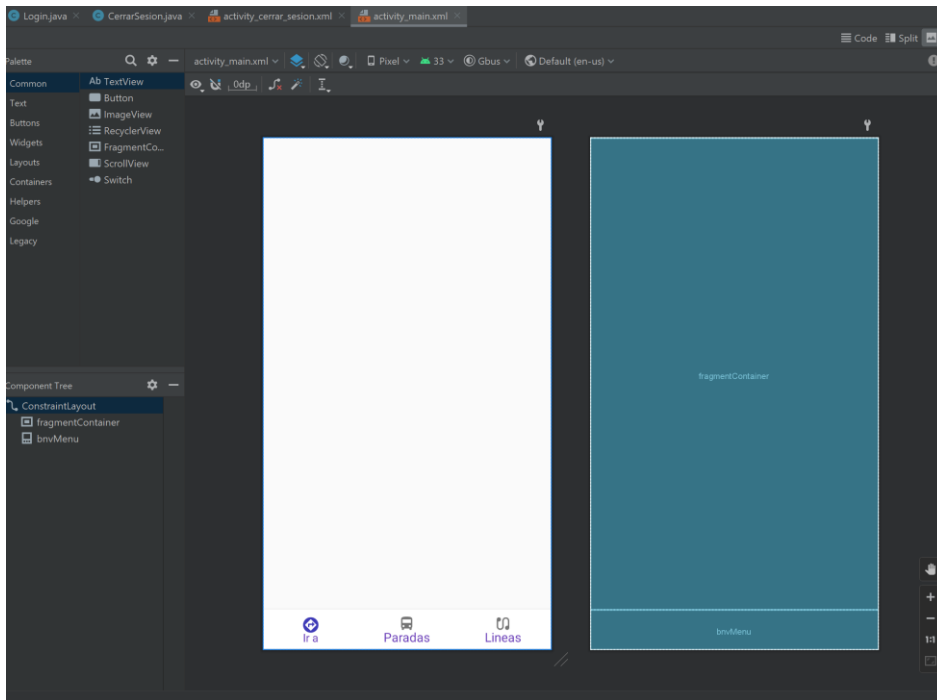


Figura 40. Layout MainActivity

```

1 package com.example.gbus;
2
3 import ...
4
22
23 public class Login extends AppCompatActivity {
24
25
26
27     private static final String TAG = "GoogleActivity";
28     private static final int RC_SIGN_IN = 9001;
29
30     // [START declare_auth]
31     private FirebaseAuth mAuth;
32     // [END declare_auth]
33
34     private GoogleSignInClient mGoogleSignInClient;
35     @SuppressWarnings("MissingInflatedId")
36     @Override
37     protected void onCreate(Bundle savedInstanceState) {
38         setTheme(R.style.Theme_gbus);
39         super.onCreate(savedInstanceState);
40         setContentView(R.layout.activity_login);
41         findViewById(R.id.btnIniciarSesion).setOnClickListener(new View.OnClickListener() {
42             @Override
43             public void onClick(View v) { signIn(); }
44         });
45     };
46
47     GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
48         .requestIdToken("594005154004-cu1j5pvdud3v6giqhnafmhsg97hjcc0k.apps.goog...")
49         .requestEmail()
50         .build();
51     mGoogleSignInClient = GoogleSignIn.getClient(activity, this, gso);
52     // [END config_signin]
53     // [START initialize_auth]
54     // Initialize Firebase Auth
55     mAuth = FirebaseAuth.getInstance();
56     // [END initialize_auth]
57 }
58

```

Figura 41. Clase Login

```

59 // [START on_start_check_user]
60 @Override
61 public void onStart() {
62     super.onStart();
63     // Check if user is signed in (non-null) and update UI accordingly.
64     FirebaseUser currentUser = mAuth.getCurrentUser();
65     updateUI(currentUser);
66 }
67 // [END on_start_check_user]
68 // [START onactivityresult]
69 @Override
70 public void onActivityResult(int requestCode, int resultCode, Intent data) {
71     super.onActivityResult(requestCode, resultCode, data);
72
73     // Result returned from launching the Intent from GoogleSignInApi.getSignInIntent(...);
74     if (requestCode == RC_SIGN_IN) {
75         Task<GoogleSignInAccount> task = GoogleSignIn.getSignedInAccountFromIntent(data);
76         try {
77             // Google Sign In was successful, authenticate with Firebase
78             GoogleSignInAccount account = task.getResult(ApiException.class);
79             Log.d(TAG, msg: "firebaseAuthWithGoogle:" + account.getId());
80             firebaseAuthWithGoogle(account.getIdToken());
81         } catch (ApiException e) {
82             // Google Sign In failed, update UI appropriately
83             Log.w(TAG, msg: "Google sign in failed", e);
84         }
85     }
86 }
87 // [END onactivityresult]

```

Figura 42. Clase Login

```

Login.java
91 AuthCredential credential = GoogleAuthProvider.getCredential(idToken, accessToken: null);
92 mAuth.signInWithCredential(credential)
93     .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
94         @Override
95         public void onComplete(@NonNull Task<AuthResult> task) {
96             if (task.isSuccessful()) {
97                 // Sign in success, update UI with the signed-in user's information
98                 Log.d(TAG, msg: "signInWithCredential:success");
99                 FirebaseUser user = mAuth.getCurrentUser();
100                 updateUI(user);
101             } else {
102                 // If sign in fails, display a message to the user.
103                 Log.w(TAG, msg: "signInWithCredential:failure", task.getException());
104                 updateUI( user: null);
105             }
106         }
107     });
108 }
109 // [END auth_with_google]
110 // [START signin]
111 private void signIn() {
112     Intent signInIntent = mGoogleSignInClient.getSignInIntent();
113     startActivityForResult(signInIntent, RC_SIGN_IN);
114 }
115 // [END signin]
116 private void updateUI(FirebaseUser user) {
117     FirebaseUser user1=FirebaseAuth.getInstance().getCurrentUser();
118     if (user1 != null){
119         Intent intent=new Intent( packageContext: this,MainActivity.class);
120         startActivity(intent);
121         finish();
122         // Toast.makeText(this, "Ingreso", Toast.LENGTH_SHORT).show();
123     }
124 }
125 // [END check_current_user]
126 }

```

Figura 43. Clase Login

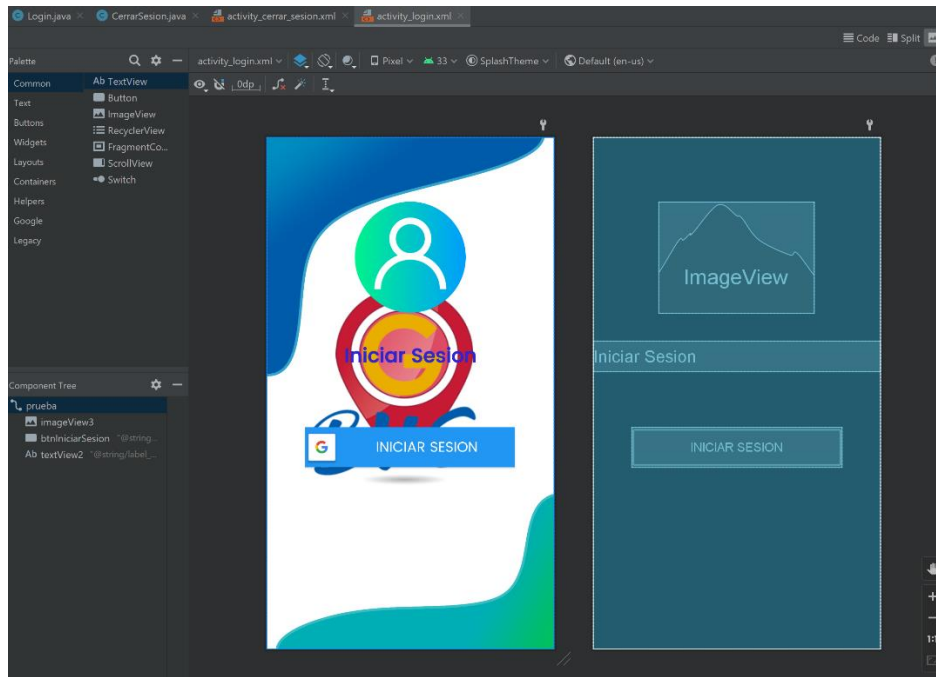


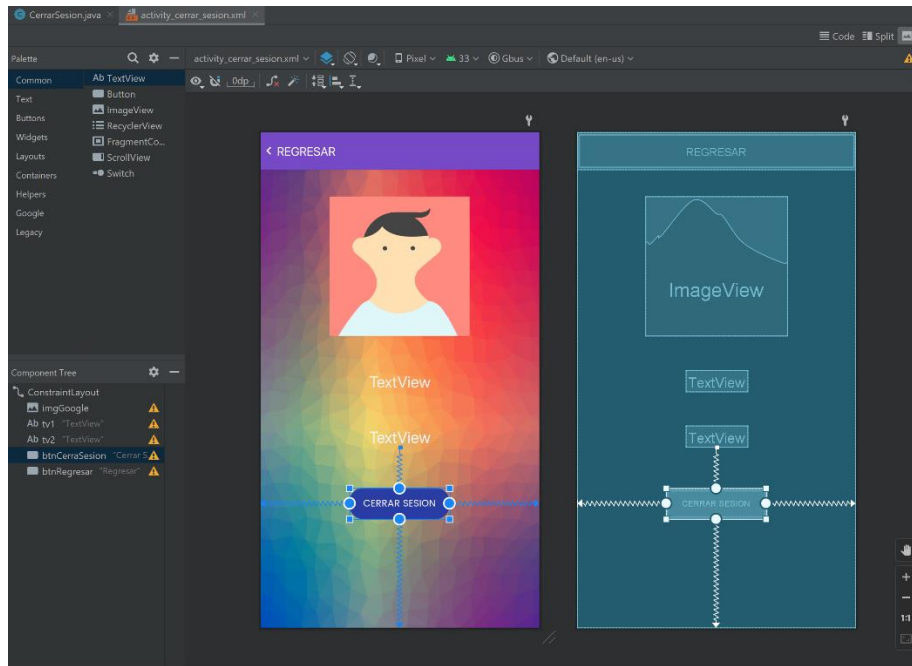
Figura 44. Layout Login

```

CerrarSesion.java x activity_cerrar_sesion.xml x
15 public class CerrarSesion extends AppCompatActivity {
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_cerrar_sesion);
21         ImageView img= findViewById(R.id.imgGoogle);
22         TextView tv1 = findViewById(R.id.tv1);
23         TextView tv2 = findViewById(R.id.tv2);
24         Button btnCerrarSesion = findViewById(R.id.btnCerraSesion);
25         Button btnRegresar = findViewById(R.id.btnRegresar);
26         FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
27
28         if (user != null) {
29             String name = user.getDisplayName();
30             String gmail = user.getEmail();
31
32             tv1.setText("Usuario\n" + name);
33             tv2.setText("Correo\n" + gmail);
34             Picasso.get().load(user.getPhotoUrl()).placeholder(R.drawable.ic_user).into(img);
35         }else {
36             getApplicationContext();
37         }
38         btnCerrarSesion.setOnClickListener(new View.OnClickListener() {
39             @Override
40             public void onClick(View v) {
41                 FirebaseAuth.getInstance().signOut();
42                 finishAffinity();
43             }
44         });
45         btnRegresar.setOnClickListener(new View.OnClickListener() {
46             @Override
47             public void onClick(View v) {
48
49                 finish();
50             }
51         });
52     }

```

Figura 45. Clase Cerrar sesión



*Figura 46. Layout Cerrar sesión*

#### *4.3.3.5. Resultado del Sprint*

Como resultado del tercer Sprint tenemos la creación del aplicativo móvil que será como lo vera el usuario, en la figura 47 tenemos el login de la aplicación que funciona con Google, en la siguiente figura 48 tenemos cerrar sesión, esto por si el usuario quiere mantener iniciado sesión y finalmente en la figura 49 nos muestra la navegación de controles.



*Figura 47. Inicio de sesión de la Aplicación*



*Figura 48. Cerrar sesión de la Aplicación*

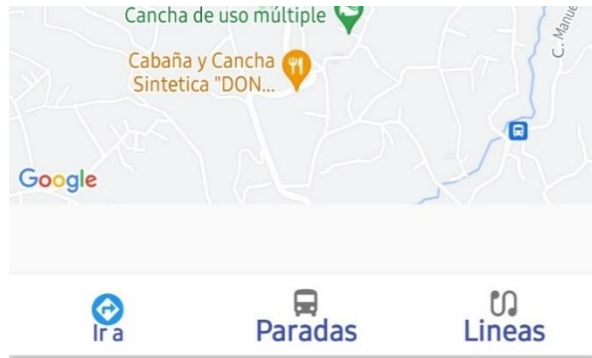


Figura 49. Menú para cambio de pantalla

#### 4.3.4. Sprint 4

Tabla 24. Cuarta historia de Usuario

Historia de Usuario	
Numero: 4	Prioridad: 3
Nombre de historia de usuario: Localización de Autobuses	Responsable: Minchala J, Crespo J.
Descripción: En el cuarto Sprint se plantea a realizar la localización de los Autobuses, continuando con el desarrollo de la aplicación, teniendo en cuenta que se hará uso de los datos obtenidos a través del dispositivo de rastreo que han sido almacenados en el servidor web.	

Adaptado por: Minchala J, Crespo J.

##### 4.3.4.1. Especificación del Sprint

De la misma manera que los otros Sprint tabla 25, se evalúan las tareas a realizar para la integración del aplicativo Móvil, teniendo un tiempo de aproximadamente 10 días.

##### 4.3.4.2. Sprint Backlog

A continuación, se declaran las tareas a realizar para el cuarto Sprint como se observa en las tablas 26 y 27, para la distribución de las actividades que se darán en el tiempo declarado por el Product backlog de la tabla 25.



Tabla 25. Cuarto Product Backlog

<b>Product Backlog</b>		
<b>Numero</b>	<b>Historia de Usuario</b>	<b>Prioridad</b>
<b>4</b>	Localización de Autobuses	<b>3</b>

Adaptado por: Minchala J, Crespo J.

Tabla 26. Tareas del Cuarto Sprint

<b>Tareas del Cuarto Sprint</b>			
<b>Sprint</b>	<b>Tiempo del proyecto en horas</b>	<b>Historia de Usuario</b>	<b>Estimación</b>
4	240	Localización de Autobuses	
<b>S.B</b>	<b>Tareas</b>	<b>Situación</b>	
4	Creación de la clase Ira	Terminado	18/12/22 - 03/01/23
	Creación del Método del movimiento del bus	Terminado	
	Creación del diseño del Layout	Terminado	
	Validación de Campos	Terminado	
	Validación de conexión a internet	Terminado	
	Permisos de almacenamiento	Terminado	
	Permisos de Ubicación del Usuario	Terminado	
	Método de parada de bus más cercana	Terminado	
	Validación de paradas más cercanas solo para el sector Guapan	Terminado	

Adaptado por: Minchala J, Crespo J.

Tabla 27. Tareas terminadas del cuarto Sprint

<b>Tareas Culminadas</b>	
<b>Tareas</b>	<b>Fechas</b>
Creación de la clase Ira	18/12/22
Creación del Método del movimiento del bus	19/12/22
Creación del diseño del Layout	21/12/22
Validación de campos	22/12/22
Validación de conexión a internet	26/12/22
Permisos de almacenamiento	27/12/22
Permisos de Ubicación del Usuario	27/12/22
Método de parada de bus más cercana	28/12/22
Validación de paradas más cercanas solo para el sector Guapan	30/12/22

Adaptado por: Minchala J, Crespo J.

#### *4.3.4.3. Diagrama de Gantt*

En el siguiente diagrama de Gantt podemos observar la distribución de tareas, de igual manera de tener sus respectivas fechas, observándose de la siguiente manera en la figura 50.

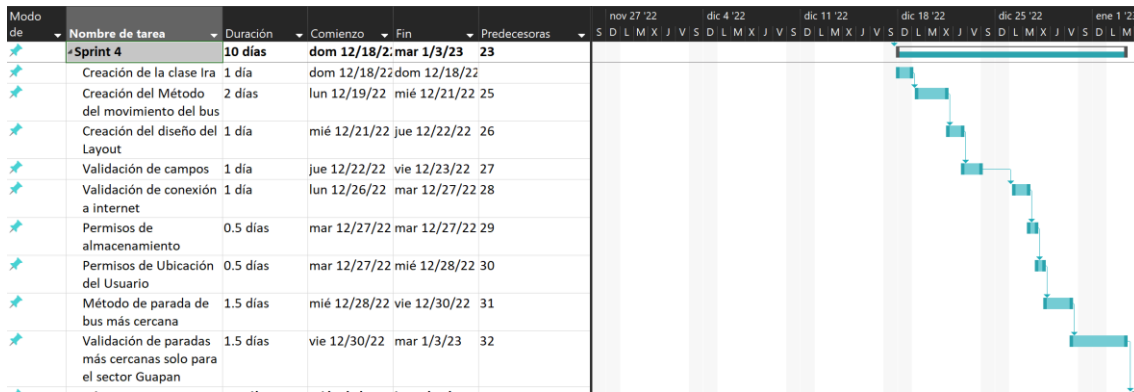


Figura 50. Diagrama de Gantt Sprint 4

#### 4.3.4.4. Explicación del Sprint

A continuación, se observa en las figuras del 51 al 64, la creación de la clase Ira, esta contiene los permisos de ubicación del usuario que es esencial para saber nuestra posición y la del vehículo. Además, integra el método que hace que el bus este en constante movimiento. Por otro lado, incluye el método de la ruta más cercana que añade una polyline como sugerencia.

```

1 package com.example.gbus.ControladorTab;
2
3
4 import ...
5
6 public class Ira extends Fragment implements OnMapReadyCallback, GoogleApiClient.OnConnectionFailedListener, SearchView.OnQueryTextListener {
7
8     private GoogleMap mMap;
9     private List<Coordenadas> listaCoordenadas;
10    private ArrayList<Marker> mapRealTimeMarkers = new ArrayList<>();
11    private View mView;
12    private MapView mMapView;
13    private Button btnVerBus;
14    TextView textViewInicio;
15    LatLng coordDest, coordOrigen;
16    ParadasPuntos obj = new ParadasPuntos();
17    RutasPuntos rte = new RutasPuntos();
18    TextView tvNoConectado;
19    int numList = 0, busSelect, x = 1;
20    private AutoCompleteTextView autoOrigen, autoDestino;
21    private String origenItemSelect, destItemSelect, baseUrl = "http://192.168.0.192:8080", userLogin = "AdminBuses", passURL = "jicq266";
22    private static final int PERMISSION_REQUEST_LOCATION = 1;
23    @Override
24    public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
25    }
26    public Ira() {
27    }
28    @
29    public static Ira newInstance() {
30        Bundle args = new Bundle();
31        Ira fragment = new Ira();
32        fragment.setArguments(args);
33        return fragment;
34    }
35 }

```

Figura 51. Clase Ira

```
xml x irajava x
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
    busSelect = 0;
    mView = inflater.inflate(R.layout.fragment_ira, container, attachToRoot: false);
    autoCOrigen = mView.findViewById(R.id.buscarOrigen);
    autoCDestino = mView.findViewById(R.id.buscarDestino);
    imagenSinCo = mView.findViewById(R.id.imgSinCo);
    btnVerBus = mView.findViewById(R.id.btnVerBus);
    imagenSinCo.setVisibility(View.INVISIBLE);
    tvNoConectado = mView.findViewById(R.id.tvNoConectado);
    return mView;
}

@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    ConnectivityManager con = (ConnectivityManager) getActivity().getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = con.getActiveNetworkInfo();
    if (networkInfo != null && networkInfo.isConnected()) {
        imagenSinCo.setVisibility(View.INVISIBLE);
        tvNoConectado.setVisibility(View.INVISIBLE);
        int permissionCheck = ContextCompat.checkSelfPermission(getActivity(),
            Manifest.permission.ACCESS_FINE_LOCATION);
        if (permissionCheck == PackageManager.PERMISSION_DENIED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(getActivity(),
                Manifest.permission.ACCESS_FINE_LOCATION)) {
            } else {
                ActivityCompat.requestPermissions(getActivity(),
                    new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                    requestCode: 1);
            }
        }
    }
}
```

Figura 52. Clase Ira

```

mMapView = mView.findViewById(R.id.map);
if (mMapView != null) {
    mMapView.onCreate( savedInstanceState: null);
    mMapView.onResume();
    mMapView.getMapAsync( callback: this);
}
List<String> items = new ArrayList<>();
List<String> itemsOrigen = new ArrayList<>();
for (int i = 0; i < obj.paradasMarcadores().size(); i++) {
    items.add(obj.paradasMarcadores().get(i).getTitulo());
}
itemsOrigen.add("MI UBICACION ACTUAL");
for (int i = 0; i < obj.paradasMarcadores().size(); i++) {
    itemsOrigen.add(obj.paradasMarcadores().get(i).getTitulo());
}
ArrayAdapter<String> adapterOrigen = new ArrayAdapter<>(getActivity(),
    android.R.layout.simple_dropdown_item_1line, itemsOrigen);
ArrayAdapter<String> adapterDestino = new ArrayAdapter<>(getActivity(),
    android.R.layout.simple_dropdown_item_1line, items);
autoCOrigen.setAdapter(adapterOrigen);
autoCOrigen.setThreshold(1);
autoCDestino.setAdapter(adapterDestino);
autoCDestino.setThreshold(1);
autoCOrigen.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        autoCOrigen.showDropDown();
        autoCOrigen.performCompletion();
    }
});

autoCDestino.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        autoCDestino.showDropDown();
        autoCDestino.performCompletion();
    }
});
});

```

Figura 53. Clase Ira

```

autoCDestino.setEnabled(false);
autoCOrigen.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent1, View view, int position, long id) {

        coordsOrigen = null;
        origenItemSelect = (String) parent1.getItemAtPosition(position);
        int posItem = 0;
        for (int i = 0; i < itemsOrigen.size(); i++) {
            if (itemsOrigen.get(i).equals(origenItemSelect)) {
                posItem = i;
                break;
            }
        }
        if (posItem == 0) {
            if (CheckGps()) {
                if (ContextCompat.checkSelfPermission(getActivity(), Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
                    // Si el permiso no está concedido, se muestra un diálogo de alerta para solicitarlo.
                    if (ActivityCompat.shouldShowRequestPermissionRationale(getActivity(), Manifest.permission.ACCESS_FINE_LOCATION)) {
                        // Si el usuario ha denegado el permiso anteriormente y ha elegido la opción "No volver a preguntar",
                        // se proporciona una explicación al usuario de por qué es necesario el permiso.
                        new AlertDialog.Builder(getActivity())
                            .setTitle("Solicitud de permiso")
                            .setMessage("Es necesario el permiso de ubicación para obtener tu ubicación actual.")
                            .setPositiveButton("Entiendo", new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog, int which) {
                                    // Se proporciona al usuario un enlace a la página de configuración de permisos.
                                    Intent intent = new Intent();
                                    intent.setAction(Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
                                    Uri uri = Uri.fromParts("package", getActivity().getPackageName(), null);
                                    intent.setData(uri);
                                    startActivity(intent);
                                }
                            })
                            .create()
                            .show();
                    }
                }
            }
        }
    }
});

```

Figura 54. Clase Ira

```

    } else {
        // Si no se proporciona una explicación al usuario, se solicita el permiso directamente.
        requestPermissions(new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, PERMISSION_REQUEST_LOCATION);
    }
} else {
    LocationManager locationManager = (LocationManager) getActivity().getSystemService(LOCATION_SERVICE);
    if (ActivityCompat.checkSelfPermission(getActivity(), Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
        && ActivityCompat.checkSelfPermission(getActivity(), Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        return;
    }
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, minTimeMs: 0, minDistanceM: 0, new LocationListener() {
        @Override
        public void onLocationChanged(Location location) {
            coordsOrigen = new LatLng(location.getLatitude(), location.getLongitude());
            autoCDestino.setEnabled(true);
        }
        @Override
        public void onStatusChanged(String provider, int status, Bundle extras) {
        }
        @Override
        public void onProviderEnabled(String provider) {
        }
        @Override
        public void onProviderDisabled(String provider) {
        }
    });
} else {
    Toast.makeText(getActivity(), text: "Obteniendo ubicación GPS", Toast.LENGTH_SHORT).show();
}
} else {
    autoCDestino.setEnabled(true);
}
});

```

Figura 55. Clase Ira

```

autoCDestino.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        mMap.clear();
        destItemSelect = (String) parent.getItemAtPosition(position);
        int posicion1 = 0;
        for (int i = 0; i < itemsOrigen.size(); i++) {
            if (itemsOrigen.get(i).equals(origenItemSelect)) {
                posicion1 = i;
                break;
            }
        }
        if (posicion1 != 0) {
            coordsOrigen = obj.paradasMarcadores().get(posicion1 - 1).getCoords();
            autoCDestino.setEnabled(true);
        }
        coordsDest = null;
        String a = (String) parent.getItemAtPosition(position);
        int posicion = -1;
        for (int i = 0; i < obj.paradasMarcadores().size(); i++) {
            if (obj.paradasMarcadores().get(i).getTitulo().equals(a)) {
                posicion = i;
                break;
            }
        }
        coordsDest = obj.paradasMarcadores().get(posicion).getCoords();
        numList = posicion;
        if (coordsOrigen != null) {
            paradaMasCercana(coordsOrigen, mMap, getActivity());
            rutaMasCercana();
        } else {
            Toast.makeText(getActivity(), text: "Encienda el GPS", Toast.LENGTH_SHORT).show();
        }
    }
});

```

Figura 56. Clase Ira

```

public Latlng paradaMasCercana(LatLng coordsPM, GoogleMap gMap, Activity activity) {
    List<LatLng> marcadores = new ArrayList<>();
    for (int i = 0; i < obj.paradasMarcadores().size(); i++) {
        marcadores.add(obj.paradasMarcadores().get(i).getCoords());
    }
    LatLng posicionMasCercana = null;
    double distanciaActual = Double.MAX_VALUE;
    for (int i = 0; i < marcadores.size(); i++) {
        double distancia = SphericalUtil.computeDistanceBetween(coordsPM, marcadores.get(i));
        if (distanciaActual > distancia) {
            posicionMasCercana = marcadores.get(i);
            distanciaActual = distancia;
        }
    }
    //Calcular distancia
    Location location1 = new Location("provider");
    location1.setLatitude(coordsPM.latitude);
    location1.setLongitude(coordsPM.longitude);
    Location location2 = new Location("provider");
    location2.setLatitude(posicionMasCercana.latitude);
    location2.setLongitude(posicionMasCercana.longitude);
    float distanceInMeters = location1.distanceTo(location2);
    if (distanceInMeters < 1000) {
        gMap.addMarker(new MarkerOptions().position(posicionMasCercana).title("Parada Mas Cercana").icon(BitmapDescriptorFactory.fromResource(R.drawable.imgparado)));
        CameraUpdate locationBus = CameraUpdateFactory.newLatLngZoom(posicionMasCercana, zoom: 15);
        gMap.moveCamera(locationBus);
        trazolinea(coordsPM, posicionMasCercana, activity, gMap);
    } else {
        Toast.makeText(activity, "Esta muy lejos de la ubicacion", Toast.LENGTH_SHORT).show();
    }
    return posicionMasCercana;
}

```

Figura 57. Método Parada más cercana

```

private void rutaMasCercana() {
    List<Polyline> polyLines = new ArrayList<>();
    Polyline polyline1 = mMap.addPolyline(rts.RutaGuapan);
    Polyline polyline2 = mMap.addPolyline(rts.RutaCachipamba);
    Polyline polyline3 = mMap.addPolyline(rts.RutaZhindilig);
    Polyline polyline4 = mMap.addPolyline(rts.RutaQuinua);
    Polyline polyline5 = mMap.addPolyline(rts.RutaBuil);
    Polyline polyline6 = mMap.addPolyline(rts.Ruta0Ecuatoriano);
    Polyline polyline7 = mMap.addPolyline(rts.RutaAguilan);
    polyLines.add(polyline1);
    polyLines.add(polyline2);
    polyLines.add(polyline3);
    polyLines.add(polyline4);
    polyLines.add(polyline5);
    polyLines.add(polyline6);
    polyLines.add(polyline7);
    mMap.clear();
    paradaMasCercana(coordsOrigen, mMap, getActivity());
    PolylineOptions options = new PolylineOptions().color(Color.GREEN)
        .width(15);
    PolylineOptions options1 = new PolylineOptions().color(Color.GREEN)
        .width(15);
    if (destItemSelect == "Terminal Terrestre") {
        //Polyline que se encuentra cerca del origen
        double minDistance1 = Double.MAX_VALUE;
        Polyline closestPolyline1 = null;
        for (Polyline polyline : polyLines) {
            List<LatLng> points = polyline.getPoints();
            for (LatLng point : points) {
                double distance = SphericalUtil.computeDistanceBetween(point, coordsOrigen);
                if (distance < minDistance1) {
                    minDistance1 = distance;
                    closestPolyline1 = polyline;
                }
            }
        }
        options1.addAll(closestPolyline1.getPoints());
        mMap.addPolyline(options1);
    }
}

```

Figura 58. Método ruta más cercana

```

    } else {
        //Polyline que se encuentra cerca del destino
        double minDistance = Double.MAX_VALUE;
        Polyline closestPolyline = null;
        for (Polyline polyline : polylines) {
            List<LatLng> points = polyline.getPoints();
            for (LatLng point : points) {
                double distance = SphericalUtil.computeDistanceBetween(point, coordsDest);
                if (distance < minDistance) {
                    minDistance = distance;
                    closestPolyline = polyline;
                }
            }
        }
        options.addAll(closestPolyline.getPoints());
        mMap.addPolyline(options);
    }
    mMap.addMarker(new MarkerOptions().position(coordsDest).title(obj.paradasMarcadores().get(numList).getTitulo())
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.imgdestino)));
}
}

```

*Figura 59. Método ruta más cercana*

```

public void trazoLinea(LatLng posicionMasCercana, LatLng coordsDest, Activity activity, GoogleMap mMap) {
    String url = "https://maps.googleapis.com/maps/api/directions/json?origin=" + posicionMasCercana.latitude + ","
        + posicionMasCercana.longitude + "&destination=" + coordsDest.latitude + "," + coordsDest.longitude + "&key=AlzaSycyK5tagfULYN3rN5LCA6Q0VWxaRUT0eFs";
    RequestQueue queue = Volley.newRequestQueue(activity);
    StringRequest stringRequest = new StringRequest(Request.Method.GET, url, new com.android.volley.Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jso = new JSONObject(response);
                trazarRuta(jso, mMap);
                Log.i("tag: jsonRuta: ", "msg: " + response);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }, new com.android.volley.Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
        }
    });
    queue.add(stringRequest);
}

```

*Figura 60. Método Trazo de línea*

```

public void trazarRuta(JSONObject jso, GoogleMap mMap) {
    JSONArray jRoutes;
    JSONArray jLegs;
    JSONArray jSteps;
    try {
        jRoutes = jso.getJSONArray("routes");
        for (int i = 0; i < jRoutes.length(); i++) {
            jLegs = ((JSONObject) (jRoutes.get(i))).getJSONArray("legs");
            for (int j = 0; j < jLegs.length(); j++) {
                jSteps = ((JSONObject) jLegs.get(j)).getJSONArray("steps");
                for (int k = 0; k < jSteps.length(); k++) {
                    String polyline = "" + ((JSONObject) ((JSONObject) jSteps.get(k)).get("polyline")).get("points");
                    Log.i("tag: end", "msg: " + polyline);
                    List<LatLng> list = PolyUtil.decode(polyline);
                    mMap.addPolyline(new PolylineOptions().addAll(list).color(Color.GRAY).width(10));
                }
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

```

*Figura 61. Método Trazar Ruta*



```

private boolean CheckGps() {
    LocationRequest locationRequest = LocationRequest.create();
    locationRequest.setInterval(10000);
    locationRequest.setFastestInterval(5000);
    locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()
        .addLocationRequest(locationRequest)
        .setAlwaysShow(true);
    Task<LocationSettingsResponse> task = LocationServices.getSettingsClient(getActivity())
        .checkLocationSettings(builder.build());
    task.addOnCompleteListener(new OnCompleteListener<LocationSettingsResponse>() {
        @Override
        public void onComplete(@NonNull Task<LocationSettingsResponse> task) {
            try {
                LocationSettingsResponse response = task.getResult(ApiException.class);
                // Toast.makeText(getActivity(), "GPS ACTIVADO", Toast.LENGTH_SHORT).show();
            } catch (ApiException e) {
                if (e.getStatusCode() == LocationSettingsStatusCodes.RESOLUTION_REQUIRED) {
                    ResolvableApiException resolvableApiException = (ResolvableApiException) e;
                    try {
                        Toast.makeText(getActivity(), text: "Debe Activar el GPS para continuar", Toast.LENGTH_SHORT).show();
                        resolvableApiException.startResolutionForResult(getActivity(), requestCode: 101);
                    } catch (IntentSender.SendIntentException sendIntentException) {
                        sendIntentException.printStackTrace();
                    }
                }
                if (e.getStatusCode() == LocationSettingsStatusCodes.SETTINGS_CHANGE_UNAVAILABLE) {
                    Toast.makeText(getActivity(), text: "GPS No Activado", Toast.LENGTH_SHORT).show();
                }
            }
        }
    });
    return true;
}
}

```

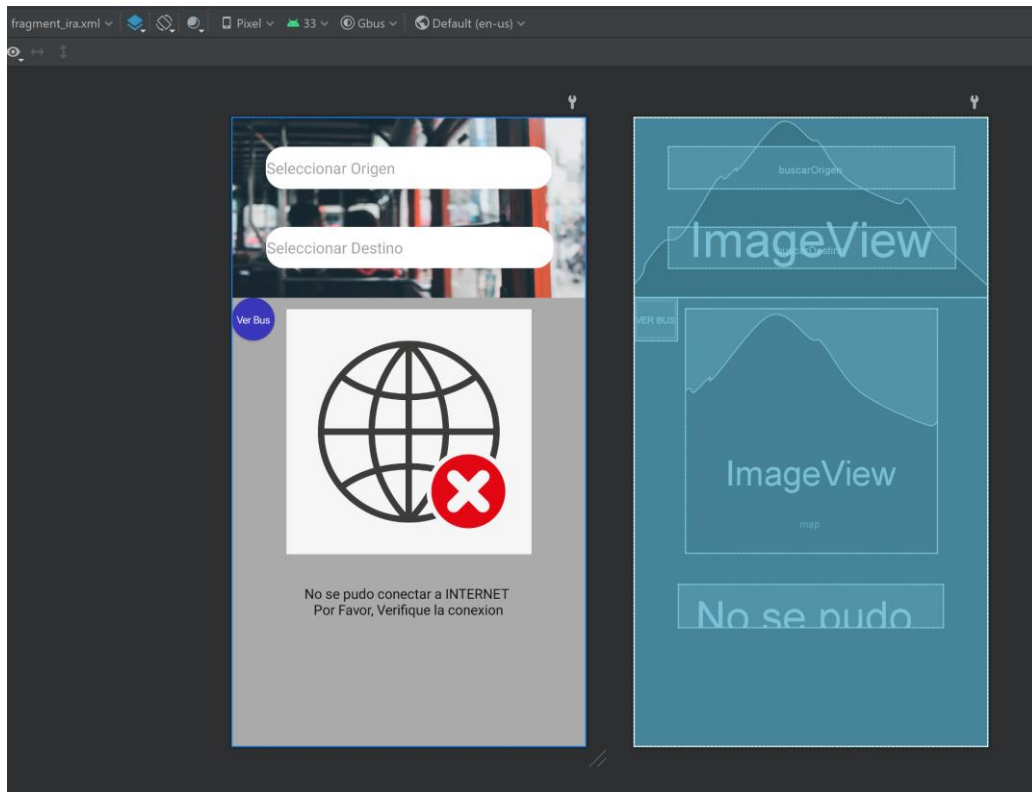
Figura 62. Método check GPS

```

public void getBusLocation() {
    // Crear una instancia de OkHttpClient y establecer las credenciales en el encabezado de autorización
    OkHttpClient client = new OkHttpClient.Builder()
        .authenticator((route, response) -> {
            String credentials = Credentials.basic(usuarioURL, passURL);
            return response.request().newBuilder().header("Authorization", credentials).build();
        })
        .build();
    Retrofit retrofit = new Retrofit.Builder()
        .baseUrl(urlPG)
        .client(client)
        .addConverterFactory(GsonConverterFactory.create())
        .build();
    RICOordenadas riCoordenadas = retrofit.create(RICOordenadas.class);
    Call<List<Coordenadas>> call4 = riCoordenadas.findAll();
    call4.enqueue(new Callback<List<Coordenadas>>() {
        @Override
        public void onResponse(Call<List<Coordenadas>> call, Response<List<Coordenadas>> response) {
            listaCoordenadas = response.body();
            Coordenadas coordenadas = listaCoordenadas.get(listaCoordenadas.size() - x);
            MarkerOptions markerOptions = new MarkerOptions();
            markerOptions.position(new LatLng(Double.valueOf(coordenadas.getLatitude()), Double.valueOf(coordenadas.getLongitude())));
            .title("Bus N° " + coordenadas.getNombus()).snippet(coordenadas.getVelobus() + " Km/h")
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.imgbus)).anchor(u: 0.0f, v: 1.0f);
            for (Marker marker : tmpRealTimeMarkers) {
                marker.remove();
            }
            tmpRealTimeMarkers.add(mMap.addMarker(markerOptions));
        }
        @Override
        public void onFailure(Call<List<Coordenadas>> call, Throwable t) {
        }
    });
}
}

```

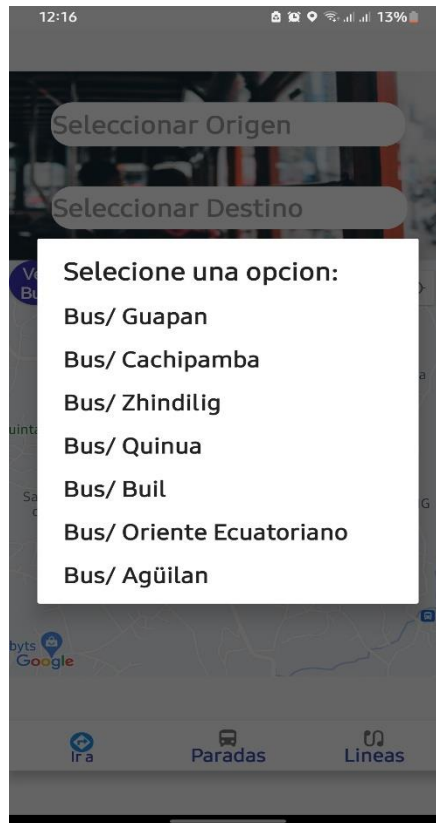
Figura 63. Obtener localización del bus



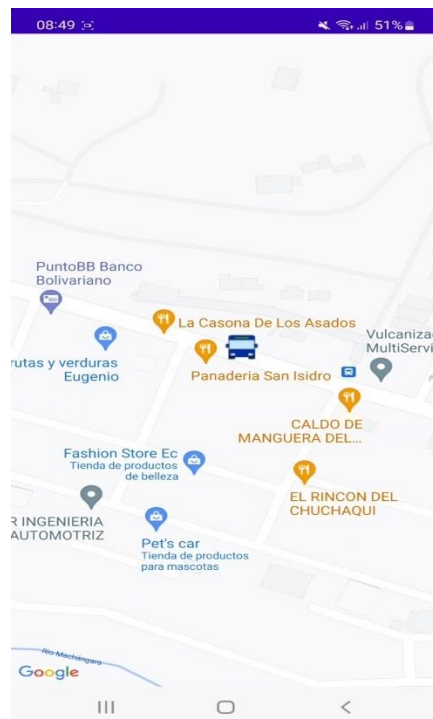
*Figura 64. Layout de la clase Ira*

#### *4.3.4.5. Resultado del Sprint*

Como resultado del desarrollo de las tareas se obtiene una vista de la localización del bus como se ve en la siguiente figura 66, además de mostrar las distintas líneas de autobús que hay como se ve en la figura 65 y por último un buscador en caso de querer ir a un sitio en específico como se ve en la figura 67.



*Figura 65. Lista de buses*



*Figura 66. Vista del bus en Movimiento en tiempo real*

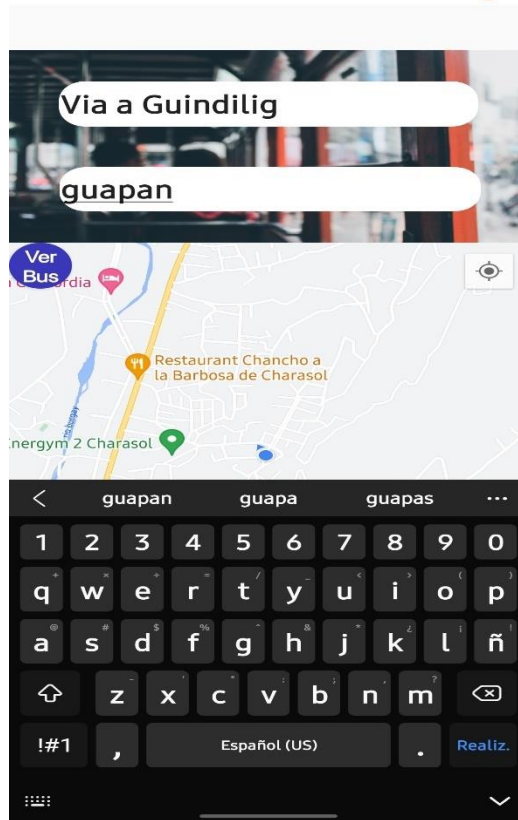


Figura 67. Buscador de sitios

#### 4.3.5. Sprint 5

Tabla 28. Historia de Usuario del quinto Sprint

Historia de Usuario	
Numero: 5	Prioridad: 3
Nombre de historia de usuario: Mapeo de Rutas y parada de autobuses	Responsable: Minchala J, Crespo J.
Descripción: es importante para la gente saber las respectivas rutas de los autobuses en caso de no conocer el lugar y tener la oportunidad de saber las respectivas paradas de autobuses.	

Adaptado por: Minchala J, Crespo J.

#### 4.3.5.1. Especificación del Sprint

Este quinto Sprint se ejecuta el mapeo de rutas y paradas de autobuses a realizar como se observa en las tablas 28 y 29, tomando en cuenta el tiempo aproximado de 5 días para la elaboración de las actividades.

#### 4.3.5.2. Sprint Backlog

En este Quinto y último Sprint se procede a realizar las ultimas tareas del proyecto, en la tabla 30 y 31 se muestran las tareas a ejecutar, mismas que serán esenciales para la culminación con el proyecto.

Tabla 29. Quinto Product Backlog

<b>Product Backlog</b>		
<b>Numero</b>	<b>Historia de Usuario</b>	<b>Prioridad</b>
<b>5</b>	Configuración de Módulos ESP8266 Y GPS NEO 6M	<b>3</b>

Adaptado por: Minchala J, Crespo J.

Tabla 30. Tareas del quinto Sprint

<b>Tareas del Quinto Sprint</b>			
<b>Sprint</b>	<b>Tiempo del proyecto en horas</b>	<b>Historia de Usuario</b>	<b>Estimación</b>
5	137	Mapeo de Rutas y parada de autobuses	
<b>S.B</b>	<b>Tareas</b>	<b>Situación</b>	04/01/23 - 12/01/23
5	Recolección de coordenadas para las rutas	Terminado	
	Recolección de coordenadas para las paradas	Terminado	
	Guía de rutas	Terminado	
	Ajustes de Mapa	Terminado	
	Ajustes de posición de la cámara del mapa	Terminado	

Adaptado por: Minchala J, Crespo J.

Tabla 31. Tareas culminadas del quinto Sprint

<b>Tareas Culminadas</b>	
<b>Tareas</b>	<b>Fechas</b>
Recolección de coordenadas para las rutas	04/01/23
Recolección de coordenadas para las paradas	05/01/23
Guía de rutas	06/01/23
Ajustes de mapa	09/01/23
Ajustes de posición de la cámara del mapa	10/01/23

Adaptado por: Minchala J, Crespo J.

### 4.3.5.3. Diagrama de Gantt

En la figura 68 del diagrama de Gantt, nos muestra el tiempo de las tareas realizadas a lo largo del Sprint, mostrando el avance que se tuvo a lo largo del proyecto para poder culminar con el trabajo.



Figura 68. Diagrama de Gantt Sprint 5

### 4.3.5.4. Explicación del Sprint

- ❖ Como resultado de las tareas del Sprint, tenemos en las figuras 69 al 73 la creación de la Clase Paradas y ParadasPuntos, que integra las configuraciones de posición de la cámara en el mapa. Además, integra el añadido de las paradas con sus respectivos métodos y coordenadas que serán mostradas en el mapa.
- ❖ En las figuras 74 y 75 se visualiza la creación de la clase Rutas y RutasPuntos, integrando las coordenadas de las rutas y el dibujo de las rutas por el método de Polyline.
- ❖ En las figuras 76 muestra el método de llamada Get y Set que será representado en el mapa además de mostrar el método de listado en la figura 79.

- ❖ En las figuras 77 y 78 se muestra la clase Rutas Marcadas, la cual consiste en poner señales en el mapa que serán de ayuda a la guía de las rutas.

```
Paradas.java
84     mMapView = mView.findViewById(R.id.mapParadas);
85     if (mMapView != null) {
86         mMapView.onCreate(savedInstanceState);
87         mMapView.onResume();
88         mMapView.getMapAsync(this);
89     }
90     locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
91     if (ActivityCompat.checkSelfPermission(getContext(), Manifest.permission.ACCESS_FINE_LOCATION)
92         != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(getActivity(),
93         Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
94         return;
95     }
96     locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, minTimeMs: 0, minDistanceM: 0, new LocationListener() {
97         @Override
98         public void onLocationChanged(Location location) {
99             coords = new LatLng(location.getLatitude(), location.getLongitude());
100        }
101        @Override
102        public void onStatusChanged(String provider, int status, Bundle extras) {
103        }
104        @Override
105        public void onProviderEnabled(String provider) {
106        }
107        @Override
108        public void onProviderDisabled(String provider) {
109        }
110    });
111    paradaCercana.setOnClickListener(new View.OnClickListener() {
112        @Override
113        public void onClick(View v) {
114            CheckGps();
115            if (coords != null) {
116                iraFragment.paradaMasCercana(coords, mMap, getActivity());
117            } else {
118                Toast.makeText(getActivity(), text: "Obteniendo Ubicacion", Toast.LENGTH_SHORT).show();
119            }
120        }
121    });
```

Figura 69. Clase Parada





```

170 public void Antut(GoogleMap googleMap) {
171     mMap = googleMap;
172     List<MarkerOptions> markers = new ArrayList<>();
173     for (int i = 0; i < obj.paradasMarcadores().size(); i++) {
174         markers.add(new MarkerOptions().position(obj.paradasMarcadores().get(i)
175             .getCoords()).title(obj.paradasMarcadores().get(i).getTitulo()));
176     }
177     for (MarkerOptions marker : markers) {
178         mMap.addMarker(marker.icon(BitmapDescriptorFactory.fromResource(R.drawable.impparada)));
179     }
180 }
181 private boolean CheckGps() {
182     LocationRequest locationRequest = LocationRequest.create();
183     locationRequest.setInterval(10000);
184     locationRequest.setFastestInterval(5000);
185     locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
186     LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()
187         .addLocationRequest(locationRequest)
188         .setAlwaysShow(true);
189     Task<LocationSettingsResponse> task = LocationServices.getSettingsClient(getActivity())
190         .checkLocationSettings(builder.build());
191     task.addOnCompleteListener(new OnCompleteListener<LocationSettingsResponse>() {
192         @Override
193         public void onComplete(@NonNull Task<LocationSettingsResponse> task) {
194             try {
195                 LocationSettingsResponse response = task.getResult(ApiException.class);
196             } catch (ApiException e) {
197                 if (e.getStatusCode() == LocationSettingsStatusCodes.RESOLUTION_REQUIRED) {
198                     ResolvableApiException resolvableApiException = (ResolvableApiException) e;
199                     try {
200                         Toast.makeText(getActivity(), "Debe Activar el GPS para continuar", Toast.LENGTH_SHORT).show();
201                         resolvableApiException.startResolutionForResult(getActivity(), requestCode: 101);
202                     } catch (IntentSender.SendIntentException sendIntentException) {
203                         sendIntentException.printStackTrace();
204                     }
205                 }
206             }
207         }
208     });
209 }

```

Figura 72. Check GPS

```

9 public class ParadasPuntos {
10
11     private LatLng coords;
12     private String titulo;
13     private ArrayList<ParadasPuntos> paradasMarcadores;
14
15
16     public ParadasPuntos(LatLng coords, String titulo) {
17         this.coords = coords;
18         this.titulo = titulo;
19     }
20
21     public LatLng getCoords() { return coords; }
22
23     public void setCoords(LatLng coords) { this.coords = coords; }
24
25     public String getTitulo() { return titulo; }
26
27     public void setTitulo(String titulo) { this.titulo = titulo; }
28
29     public ArrayList<ParadasPuntos> paradasMarcadores() { return paradasMarcadores; }
30
31     public void setListaPersonas(ArrayList<ParadasPuntos> listaPersonas) {
32         this.paradasMarcadores = listaPersonas;
33     }
34
35     @SuppressWarnings("SuspiciousIndentation")
36     public ParadasPuntos() {
37
38         paradasMarcadores = new ArrayList<>();
39         paradasMarcadores.add(new ParadasPuntos(new LatLng( latitude: -2.7369937415097225, longitude: -78.84867425983633), titulo: "Terminal Terrestre"));
40         paradasMarcadores.add(new ParadasPuntos(new LatLng( latitude: -2.727182485962482, longitude: -78.84861561184668), titulo: "Cesantos Guapán"));
41         paradasMarcadores.add(new ParadasPuntos(new LatLng( latitude: -2.72166458628844, longitude: -78.84838915628152), titulo: "Intructor"));
42         paradasMarcadores.add(new ParadasPuntos(new LatLng( latitude: -2.7181494124684455, longitude: -78.84747230719789), titulo: "Ferretería González"));
43         paradasMarcadores.add(new ParadasPuntos(new LatLng( latitude: -2.712635656076812, longitude: -78.84848576538569), titulo: "Parque Central de Guapán"));
44         paradasMarcadores.add(new ParadasPuntos(new LatLng( latitude: -2.7158239878331334, longitude: -78.8501223735835), titulo: "Iglesia Virgen de las Nieves Guapán"));
45         paradasMarcadores.add(new ParadasPuntos(new LatLng( latitude: -2.720091809808559, longitude: -78.84703189687679), titulo: "Vía a Guindilla"));
46     }
47 }

```

Figura 73. Coordenada de las paradas

```

42 @Override
43 public View onCreateView(@Nullable LayoutInflater inflater, @Nullable ViewGroup container,
44                          @Nullable Bundle savedInstanceState) {
45     // Inflate the layout for this fragment
46     View vista = inflater.inflate(R.layout.fragment_lineas, container, attachToRoot: false);
47
48     btnLRTGuapan = vista.findViewById(R.id.btnGuapan);
49     btnLRTCachipamba = vista.findViewById(R.id.btnGuindilig);
50     btnLRTZhindilig = vista.findViewById(R.id.btnZhindilig);
51     btnLRTQuinua = vista.findViewById(R.id.btnQuinua);
52     btnLRTBuil = vista.findViewById(R.id.btnBuil);
53     btnLRT0Ecuatoriano = vista.findViewById(R.id.btn0Ecuatoriano);
54     btnLRTAguilan = vista.findViewById(R.id.btnAguilan);
55     btnCerrarSesion = vista.findViewById(R.id.btnConfig);
56     btnCerrarSesion.setOnClickListener(new View.OnClickListener() {
57
58         @Override
59         public void onClick(View v) {
60             Intent intent = new Intent(getActivity(), CerrarSesion.class);
61             startActivity(intent);
62         }
63     });
64
65     presionaBtn = 0;
66     btnLRTGuapan.setOnClickListener(new View.OnClickListener() {
67
68         @Override
69         public void onClick(View v) {
70             presionaBtn = 0;
71             Intent intent = new Intent(getActivity(), RutasMarcadas.class);
72             startActivity(intent);
73         }
74     });
75
76     btnLRTCachipamba.setOnClickListener(new View.OnClickListener() {
77
78         @Override
79         public void onClick(View v) {
80             presionaBtn = 1;
81             Intent intent = new Intent(getActivity(), RutasMarcadas.class);

```

Figura 74. Método de las rutas de autobuses

```

47 public void setRutaCachipamba(PolylineOptions rutaCachipamba) {
48     RutaCachipamba = rutaCachipamba;
49 }
50
51 public PolylineOptions getRutaQuinua() { return RutaQuinua; }
52 public void setRutaQuinua(PolylineOptions rutaQuinua) { RutaQuinua = rutaQuinua; }
53 public PolylineOptions getRutaBuil() { return RutaBuil; }
54 public void setRutaBuil(PolylineOptions rutaBuil) { RutaBuil = rutaBuil; }
55 public PolylineOptions getRuta0Ecuatoriano() { return Ruta0Ecuatoriano; }
56 public void setRuta0Ecuatoriano(PolylineOptions ruta0Ecuatoriano) {
57     Ruta0Ecuatoriano = ruta0Ecuatoriano;
58 }
59
60 public PolylineOptions getRutaAguilan() { return RutaAguilan; }
61 public void setRutaAguilan(PolylineOptions rutaAguilan) { RutaAguilan = rutaAguilan; }
62 public RutasPuntos(LatLng coords, String nomRuta) {
63     this.coords = coords;
64     NomRuta = nomRuta;
65 }
66
67 public RutasPuntos() {
68
69     //RutaGuapan
70     RutaGuapan = new PolylineOptions();
71     RutaGuapan.add(new LatLng( latitude: -2.7370033270365806, longitude: -78.84861665234936));
72     RutaGuapan.add(new LatLng( latitude: -2.737026060371308, longitude: -78.84904907898175));
73     RutaGuapan.add(new LatLng( latitude: -2.736112937744707, longitude: -78.84899218074442));
74     RutaGuapan.add(new LatLng( latitude: -2.7353021145770335, longitude: -78.84888597069177));
75     RutaGuapan.add(new LatLng( latitude: -2.7349914251712044, longitude: -78.8489314892829));
76     RutaGuapan.add(new LatLng( latitude: -2.734775458107373, longitude: -78.84899597395615));
77     RutaGuapan.add(new LatLng( latitude: -2.73448750195485, longitude: -78.84912873651582));
78     RutaGuapan.add(new LatLng( latitude: -2.7342488014133313, longitude: -78.84927287872661));
79     RutaGuapan.add(new LatLng( latitude: -2.7337410889762124, longitude: -78.8494511598989));
80     RutaGuapan.add(new LatLng( latitude: -2.732638518710957, longitude: -78.84958012925263));
81     RutaGuapan.add(new LatLng( latitude: -2.731858004435718, longitude: -78.8502363566283));
82     //terminal
83     RutaGuapan.add(new LatLng( latitude: -2.7313995467729266, longitude: -78.85019842353537));
84     RutaGuapan.add(new LatLng( latitude: -2.731104011994339, longitude: -78.85011876598709));

```

Figura 75. Coordenadas de las Rutas

```

1 package com.example.gbus.models;
2
3 public class Coordenadas {
4     private int id;
5     private String nombus;
6     private String longitud;
7     private String latitud;
8     private double velobus;
9     private String fecha;
10
11     public Coordenadas(int id, String nombus, String longitud, String latitud, double velobus, String fecha) {
12         this.id = id;
13         this.nombus = nombus;
14         this.longitud = longitud;
15         this.latitud = latitud;
16         this.velobus = velobus;
17         this.fecha = fecha;
18     }
19
20     public int getId() { return id; }
21     public void setId(int id) { this.id = id; }
22     public String getNombus() { return nombus; }
23     public void setNombus(String nombus) { this.nombus = nombus; }
24     public String getLongitud() { return longitud; }
25     public void setLongitud(String longitud) { this.longitud = longitud; }
26     public String getLatitud() { return latitud; }
27     public void setLatitud(String latitud) { this.latitud = latitud; }
28     public double getVelobus() { return velobus; }
29     public void setVelobus(double velobus) { this.velobus = velobus; }
30     public String getFecha() { return fecha; }
31     public void setFecha(String fecha) { this.fecha = fecha; }
32
33 }

```

Figura 76. Clase Coordenadas

```

60 public void onMapReady(@NonNull GoogleMap mMap) {
61     MapsInitializer.initialize(context, this);
62     mGoogleMap = mMap;
63     mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
64     PolylineOptions options = new PolylineOptions()
65         .color(Color.BLUE)
66         .width(15);
67     switch (obj.presionaBtn) {
68         case 0:
69             options.addAll(rts.RutaGuapan.getPoints());
70             texto.setText("Guapan");
71             break;
72         case 1:
73             options.addAll(rts.RutaCachipamba.getPoints());
74             texto.setText("Cachipamba - Guindilig");
75             break;
76         case 2:
77             options.addAll(rts.RutaZhindilig.getPoints());
78             texto.setText("Zhindilig");
79             break;
80         case 3:
81             options.addAll(rts.RutaQuinua.getPoints());
82             texto.setText("Quinua");
83             break;
84         case 4:
85             options.addAll(rts.RutaBuil.getPoints());
86             texto.setText("Buil Guapan");
87             break;
88         case 5:
89             options.addAll(rts.Ruta0Ecuatoriano.getPoints());
90             texto.setText("Oriente Ecuatoriano");
91             break;
92         case 6:
93             options.addAll(rts.RutaAguilan.getPoints());
94             texto.setText("Aguilan");
95             break;
96     }

```

Figura 77. Obtener Coordenadas de rutas

```

Coordenadas.java  RutaMarcadas.java
28 texto.setText("Oriente Ecuatoriano");
29 break;
30 case 6:
31 options.addAll(nts.RutaAguilan.getPoints());
32 texto.setText("Aguilan");
33 break;
34
35 default:
36 Toast.makeText(this, "No se ha encontrado la RUTA", Toast.LENGTH_SHORT).show();
37 break;
38 }
39
40 if (bus.prestacion Ruta == 0 || bus.prestacion Ruta == 6) {
41 mMap.addMarker(new MarkerOptions().position(options.getPoints().get(0)).title("Inicio/Fin").icon(BitmapDescriptorFactory.fromResource(R.drawable.ingorigendestino)));
42 mMap.addMarker(new MarkerOptions().position(options.getPoints().get(options.getPoints().size() - 1)).title("Continua por la Derecha")
43 .icon(BitmapDescriptorFactory.fromResource(R.drawable.imgflechaderecha)).anchor(U 0.2f, V 0.2f));
44 mMap.addMarker(new MarkerOptions().position(options.getPoints().get(options.getPoints().size() - 6)).title("Continua por Aguil")
45 .icon(BitmapDescriptorFactory.fromResource(R.drawable.imgflechabajos)));
46 mMap.addMarker(new MarkerOptions().position(options.getPoints().get(options.getPoints().size() - 2)).title("Continua por Abajo")
47 .snippet("Finaliza en el Terminal").icon(BitmapDescriptorFactory.fromResource(R.drawable.imgflechabajos)));
48 } else {
49 mMap.addMarker(new MarkerOptions().position(options.getPoints().get(0)).title("Inicio").icon(BitmapDescriptorFactory.fromResource(R.drawable.ingorigendestino)));
50 mMap.addMarker(new MarkerOptions().position(options.getPoints().get(options.getPoints().size() - 1)).title("Fin")
51 .icon(BitmapDescriptorFactory.fromResource(R.drawable.ingorigendestino)));
52 }
53
54 CameraPosition position = CameraPosition.builder().target(options.getPoints().get(0)).zoom(16).bearing(0).tilt(45).build();
55 mMap.moveCamera(CameraUpdateFactory.newCameraPosition(position));
56
57 mGoogleMap.addPolyLine(options);
58 }

```

Figura 78. Señales guía de rutas

```

RICOordenadas.java
1 package com.example.gbus.models.retrofit;
2
3 import ...
4
5
6
7
8
9
10 public interface RICOordenadas {
11     @GET("/api/getbuses")
12     Call<List<Coordenadas>> findAll();
13 }
14

```

Figura 79. Interfaz de obtencion de coordenadas

#### 4.3.5.5. Resultado del Sprint

Como resultado del quinto Sprint, tenemos la visualización de las rutas, paradas y guía de los buses, teniendo una vista general de los tres apartados en las siguientes figuras 81, 81, 82 y 83.



Figura 80. Parada de Autobuses

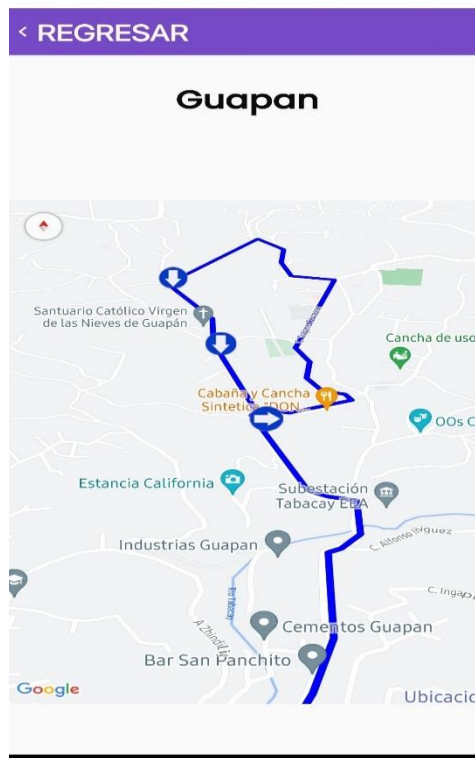


Figura 81. Guía de rutas



Figura 82. Lista de rutas a escoger

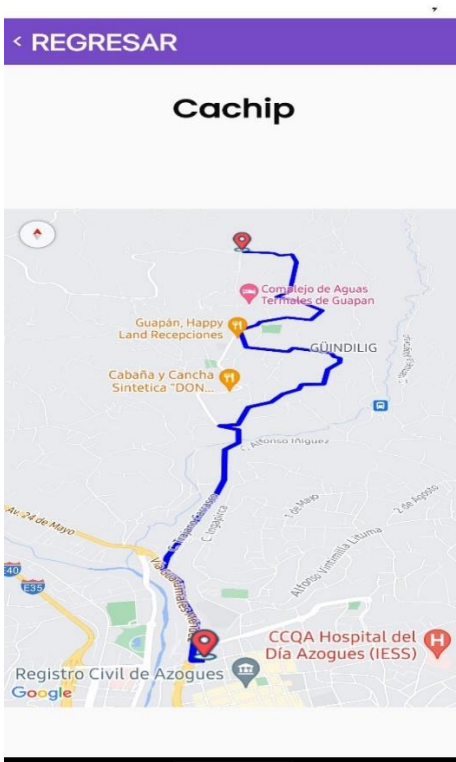


Figura 83. Vista de una de las rutas

## Capítulo 5

### 5.1. Conclusiones

- Se identificaron y evaluaron diferentes conceptos y tecnologías ya implementadas, en prototipos desarrollados para mitigar los problemas relacionados al tema, lo que nos permitió seleccionar las herramientas más óptimas para el desarrollo de nuestra aplicación.
- Las expectativas de los usuarios, choferes y desplazamiento de unidades de transporte nos ayudaron a comprender las acciones y comportamientos en el entorno y registrar lo observado, recopilando la información de forma precisa y detallada.
- Se desarrollo la aplicación móvil Android haciendo uso de la metodología planteada para cumplir con los objetivos propuestos en cuanto a diseño, funcionalidad y seguridad del aplicativo. Además, se logró aplicar estándares de calidad para mejorar las expectativas de los usuarios finales.
- Finalmente, se confirmó la correcta funcionalidad del sistema mediante pruebas exhaustivas con enfoque hacia la mejora continua.

### 5.2. Recomendaciones

- Se recomienda hacer uso de un disipador de energía para el dispositivo de rastreo en caso de ser conectado a la fuente de energía del vehículo.
- Se recomiendo hacer uso de datos móviles o internet para la conexión con el dispositivo de rastreo.
- Si el trabajo es de interés público, se recomienda su implementación, ya que esto podría incorporar una gran mejora para la empresa en sí y para el pueblo.



## Bibliografías

- [1] P. F. Alzamora Ramirez and A. J. Bautista Ramirez, “Control y Monitorización del recorrido de los buses de transporte público mediante tecnología GPS y GSM,” p. 24, 2010, [Online]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/2356/14/UPS-GT000128.pdf>
- [2] V. A. Villa Villa, “DESARROLLO DE UN SISTEMA DE CONTROL DE RECORRIDOS USANDO EQUIPOS GPS PARA LA COOPERATIVA ATAHUALPA LÍNEA 21,” UNIVERSIDAD DE GUAYAQUIL, Guayaquil, 2011. Accessed: Jan. 20, 2023. [Online]. Available: <http://repositorio.ug.edu.ec/bitstream/redug/6604/1/TesisCompleta%20-%20399%20-%202011.pdf#:~:text=Este%20sistema%20est%C3%A1%20destinado%20a%20resolver%20de%20forma,puntos%20de%20marcaci%C3%B3n%20a%20trav%C3%A9s%20de%20las%20Geo-cercas.>
- [3] J. P. Hernández Rojas, “DESARROLLO DE UN PROTOTIPO FUNCIONAL PARA LA APLICACIÓN MÓVIL Q-BUS PARA LA PLATAFORMA IOS QUE BRINDE INFORMACIÓN DE LAS RUTAS DE TRANSPORTE PÚBLICO EN LA CIUDAD DE QUITO UTILIZANDO BLUETOOTH LOW ENERGY, CÓDIGOS QR Y GEO POSICIONAMIENTO TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO,” UNIVERSIDAD TECNOLÓGICA EQUINOCCIAL, Quito, 2016. [Online]. Available: [http://repositorio.ute.edu.ec/bitstream/123456789/16857/1/67755\\_1.pdf](http://repositorio.ute.edu.ec/bitstream/123456789/16857/1/67755_1.pdf)
- [4] L. E. Betancourt Rodríguez, “SISTEMA DE GEOLOCALIZACIÓN MÓVIL PARA LA OPTIMIZACIÓN DEL TRANSPORTE PÚBLICO BRINDADO POR LA COOPERATIVA FLECHA VERDE - RUTA 55 EN LA CIUDAD DE GUAYAQUIL,” Universidad de Guayaquil, Guayaquil, 2018. Accessed: Jan. 20, 2023. [Online]. Available: <http://repositorio.ug.edu.ec/bitstream/redug/36202/1/TESIS%20GEOLOCALIZACION%20%20MOVIL%20R7.pdf>
- [5] Á. Y. San Andrés, “Aplicación Android para Buses urbanos de Guadalajara,” Universidad de Alcalá Escuela Politécnica Superior, Guadalajara, 2018. [Online]. Available: [https://ebuah.uah.es/dspace/bitstream/handle/10017/33948/TFG\\_Yelamos\\_SanAndres\\_2018.pdf?sequence=1&isAllowed=y](https://ebuah.uah.es/dspace/bitstream/handle/10017/33948/TFG_Yelamos_SanAndres_2018.pdf?sequence=1&isAllowed=y)
- [6] R. U. Hurtado Suárez, “PROTOTIPO DE SISTEMA DE LOCALIZACIÓN DE UN BUS DE TRANSPORTE URBANO MEDIANTE GPS Y UNA APLICACIÓN ANDROID,” UNIVERSIDAD TECNOLÓGICA ISRAEL, Quito, 2018. [Online]. Available: <http://repositorio.uisrael.edu.ec/bitstream/47000/1622/1/UISRAEL-EC-ELDT-378.242-2018-038.pdf>
- [7] L. J. Quinde Pozo, “Desarrollo de un sistema de mapeo y visualización de rutas de buses urbanos de la provincia de Santa Elena para la Agencia nacional de tránsito. Módulo: Aplicación móvil,” UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA, La Libertad, 2019. [Online]. Available: <https://repositorio.upse.edu.ec/bitstream/46000/5019/1/UPSE-TIN-2019-0015.pdf>
- [8] Devsu, “Manual de Usuario Aplicación Móvil proyecto Movilízate UIO,” 2015, Accessed: Jan. 16, 2023. [Online]. Available: <https://secretariademovilidad.quito.gob.ec/images/MOVILIDAD/MovilizateUIO/Manual-AppMovil.pdf>

- [9] Moovit, "Sobre Moovit," *Moovit*, 2022. <https://moovit.com/es/about-us-es/> (accessed Jan. 16, 2023).
- [10] P. Aguayo, "INTRODUCCIÓN AL MICROCONTROLADOR," Nov. 2004, Accessed: Jan. 25, 2023. [Online]. Available: <https://d1wqtxts1xzle7.cloudfront.net/39407044/micro-libre.pdf?1445752291=&response-content-disposition=inline%3B+filename%3DMicro.pdf&Expires=1674780362&Signature=J5irDGj-f7ygzqO9yRjQ2rGtceVYFWIb9HMdrBWw8vr0vuJVuFoSgd2xzAYJi~6EXu9h-6X8WjT1VzZoedlzK5jN2bPQQfJpZLOKzxeUdJg2w-FY81nb7gLzCXw16HOsxG2~6yIJuRkaEOjWSx98Oi0JNovDeW8DMz0zoXJlmr2x6jsKqU2ANXGz4e1xgnQkELfpwaX7G-6Mgs1aOJ8kr16tuq0kwWOKd0GG-ni20EcoHfv0kwUALE1el6ZJ3WhPbQLaynSeluBifcl9lgy-VYNHDYql5pt38nxpmU~cuQJ0gKWserJvGxywLF1zoMUHy~f51-x8XWlnJHrQ>
- [11] R. Enríquez Herrador, *Guía de Usuario de Arduino*. 2009. [Online]. Available: [https://foros.giltesa.com/otros/arduino/fc/docs/01-guia\\_de\\_usuario\\_arduino.pdf](https://foros.giltesa.com/otros/arduino/fc/docs/01-guia_de_usuario_arduino.pdf)
- [12] J. Quintana, "¿Qué es Arduino?," *Librería CATEDU*, Apr. 21, 2022. <https://libros.catedu.es/books/arduinoblocks-en-el-aula/page/que-es-arduino>
- [13] "Comparación de Arduino vs ESP8266 vs ESP32," *elosciloscopio*, 2022. <https://elosciloscopio.com/comparacion-arduino-vs-esp8266-vs-esp32/>
- [14] "Módulo GPS Arduino con Neo 6M," *BUGEADOS*, Oct. 2020. <https://bugeados.com/arduino/modulo-gps-arduino-con-neo-6m/>
- [15] A. L. Morán Cabezas, "Análisis e Implementación de un Sistema de Rastreo Satelital aplicado a mascotas mediante software libre con tecnología GPS y GSM," Universidad Católica de Santiago de Guayaquil, Guayaquil, 2021. [Online]. Available: <http://201.159.223.180/bitstream/3317/16332/1/T-UCSG-POS-MTEL-192.pdf>
- [16] A. Pozo-Ruz, A. Ribeiro, M. C. García-Alegre, L. García, D. Guinea, and F. Sandoval, "SISTEMA DE POSICIONAMIENTO GLOBAL (GPS): DESCRIPCIÓN, ANÁLISIS DE ERRORES, APLICACIONES Y FUTURO," Madrid. [Online]. Available: [http://www.oocities.org/es/foro\\_gps/infografia/gps5.pdf](http://www.oocities.org/es/foro_gps/infografia/gps5.pdf)
- [17] L. J. Sacristán Gutiérrez, "EL GPS ESTELAR," *Rev Gen Mar*, vol. 283, pp. 525–538, Oct. 2022, [Online]. Available: [https://armada.defensa.gob.es/archivo/rgm/2022/10/rgmoct2022\\_cap08.pdf](https://armada.defensa.gob.es/archivo/rgm/2022/10/rgmoct2022_cap08.pdf)
- [18] F. Cristancho, "¿Qué es un framework en programación?," *talently*, Feb. 08, 2022. <https://talently.tech/blog/que-es-un-framework-en-programacion/>
- [19] A. Rodríguez, "Qué es un SDK y sus ventajas dentro del App Marketing," *EMMA*. <https://emma.io/blog/que-es-un-sdk-y-sus-ventajas/> (accessed Sep. 14, 2022).
- [20] Northware, "Desarrollo multiplataforma," *Northware*. <https://www.northware.mx/blog/desarrollo-multiplataforma/> (accessed Sep. 14, 2022).
- [21] Android Studio, "Introducción a Android Studio," *developers*, Nov. 22, 2022. <https://developer.android.com/studio/intro?hl=es-419>
- [22] Java, "¿Qué es Java y por qué lo necesito?," *Java*. [https://www.java.com/es/download/help/whatis\\_java.html](https://www.java.com/es/download/help/whatis_java.html) (accessed Nov. 19, 2022).

- [23] VMware Tanzu, “¿Por qué Spring?,” *Spring*, 2022. <https://spring.io/why-spring>
- [24] C. A. Morales Machuca, “Estado del Arte: Servicios Web,” Universidad Nacional de Colombia, Bogotá, 2010. [Online]. Available: [https://d1wqtxts1xzle7.cloudfront.net/63618879/articulo2\\_120200613-43645-uxrrro-libre.pdf?1592078483=&response-content-disposition=inline%3B+filename%3DArticulo2\\_1.pdf&Expires=1673842252&Signature=NHrR~QYfGIfcu5VkSG4~fHICRDHQmFwTV2-RXWfXnsOKdo9CaAo-yciVCWclzUKL8KzdnlWEkQUcnyfnCwHG9PfdU8PxGxgFEgqAhfJI0ceycBm6ZfOIdkU7A5AjqN~ppuzWkG1awB6~Ub8KluCmQcpJw4h8nBiHSwqHCYyBPSDWaltPJxcQniJFFN-CiiM28PwNODeOulWlsVmFjm-w1CwrqnsUWgdHdmOkakNAFmgEsOjKRgvWYyu3-8b2RDj-2taB1G2Lb8II-Lk2XCooowtVZdM7rsCKAQJdohKWIE1zykZfrXriQ2ARRznXrfj7EYujapSscFo-kBogyGVmw\\_\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://d1wqtxts1xzle7.cloudfront.net/63618879/articulo2_120200613-43645-uxrrro-libre.pdf?1592078483=&response-content-disposition=inline%3B+filename%3DArticulo2_1.pdf&Expires=1673842252&Signature=NHrR~QYfGIfcu5VkSG4~fHICRDHQmFwTV2-RXWfXnsOKdo9CaAo-yciVCWclzUKL8KzdnlWEkQUcnyfnCwHG9PfdU8PxGxgFEgqAhfJI0ceycBm6ZfOIdkU7A5AjqN~ppuzWkG1awB6~Ub8KluCmQcpJw4h8nBiHSwqHCYyBPSDWaltPJxcQniJFFN-CiiM28PwNODeOulWlsVmFjm-w1CwrqnsUWgdHdmOkakNAFmgEsOjKRgvWYyu3-8b2RDj-2taB1G2Lb8II-Lk2XCooowtVZdM7rsCKAQJdohKWIE1zykZfrXriQ2ARRznXrfj7EYujapSscFo-kBogyGVmw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA)
- [25] Amazon Web Service, “¿Qué es una API?,” *AWS*, 2023. <https://aws.amazon.com/es/what-is/api/>
- [26] N. Postman, “¿Qué es Postman?,” *Postman*, 2023. <https://www.postman.com/product/what-is-postman/> (accessed Jan. 14, 2023).
- [27] Google, “Firebase,” *Firebase*, 2022. <https://firebase.google.com/> (accessed Jan. 14, 2023).
- [28] Microsoft, “Introducción al protocolo de autenticación y sincronización ,” *Microsoft*. 2023. Accessed: Jan. 25, 2023. [Online]. Available: <https://learn.microsoft.com/es-es/azure/active-directory/fundamentals/auth-sync-overview>
- [29] OAuth, “OAuth 2.0,” *OAuth*. <https://oauth.net/2/> (accessed Jan. 16, 2023).
- [30] Developers, “Autenticación en servicios OAuth2,” *Developers*, Jun. 03, 2020. <https://developer.android.com/training/id-auth/authenticate#java> (accessed Jan. 16, 2023).
- [31] P. López Herrera, “Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL,” Universidad Autónoma del Estado de México, Texcoco, 2016. Accessed: Jan. 15, 2023. [Online]. Available: <https://core.ac.uk/download/pdf/80528621.pdf>
- [32] PostgreSQL, “¿Qué es PostgreSQL?,” *PostgreSQL*, 2023. <https://www.postgresql.org/about/> (accessed Jan. 15, 2023).
- [33] J. G, “Metodologías tradicionales vs metodologías ágiles,” *LinkedIn*, Oct. 22, 2021. <https://www.linkedin.com/pulse/metodolog%C3%ADas-tradicionales-vs-%C3%A1giles-jos%C3%A9-gabal%C3%B3n/?originalSubdomain=es> (accessed Jan. 15, 2023).
- [34] K. Beck *et al.*, “Principios del Manifiesto Ágil,” *Manifiesto por el Desarrollo Ágil de Software*, 2001. <http://agilemanifesto.org/iso/es/principles.html> (accessed Jan. 20, 2023).
- [35] C. Coello, “Las mejores metodologías ágiles para la creación de software,” *Digital55*, Apr. 12, 2018. Accessed: Jan. 15, 2023. [Online]. Available: <https://digital55.com/blog/mejores-metodologias-agiles-creacion-software/>


- [36] L. Castellano Lendínez, “Kanban. Metodología para aumentar la eficiencia de los procesos,” vol. 8, no. 1, pp. 30–41, 2019, doi: 10.17993/3ctecno/2019.
- [37] P. Letelier Torres and E. A. Sánchez López, “Metodologías Ágiles en el Desarrollo de Software,” *jisdb*, Nov. 2003, Accessed: Jan. 15, 2023. [Online]. Available: [http://fcaenlinea1.unam.mx/anexos/1728/Unidad\\_1/u1\\_act2\\_1.pdf](http://fcaenlinea1.unam.mx/anexos/1728/Unidad_1/u1_act2_1.pdf)
- [38] L. Paez, “¿Qué es un Sprint en Scrum?,” *Crehana*, Apr. 2021, Accessed: Jan. 15, 2023. [Online]. Available: <https://www.crehana.com/blog/estilo-vida/scrum-sprint/>
- [39] G. Mancuzo, “Fases de la Metodología Scrum,” *ComparaSoftware*, Jul. 10, 2020. <https://blog.comparasoftware.com/fases-metodologia-scrum/> (accessed Jan. 16, 2023).
- [40] K. Schwaber and J. Sutherland, “La Guía de Scrum,” Jul. 2013, Accessed: Jan. 18, 2023. [Online]. Available: <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- [41] E. Y. Marco, “Aplicación Web y Móvil para el seguimiento de autobuses escolares,” Universitat Politècnica de València, 2015. Accessed: Jan. 17, 2023. [Online]. Available: <https://riunet.upv.es/bitstream/handle/10251/55654/YAGO%20-%20Aplicaci%3%b3n%20Web%20y%20M%3%b3vil%20para%20el%20seguimiento%20de%20autobuses%20escolares.pdf?sequence=1&isAllowed=y>

## Glosario

GPS	= Global Positioning System
DGPS	= Differential Global Positioning System
IDE	= Entorno de Desarrollo Interactivo
IoT	= Internet of Things
OTA	= Over The Air
MVVM	= Model View ViewModel
HTML	= Hyper Text Markup Language
CSS	= Cascading Style Sheets
POO	= Programación Orientada a Objetos
SDK	= Software Development Kit
SQL	= Structured Query Language
No-SQL	= Not Only SQL
CRUD	= (Create, Read, Update and Delete)
SO	= Sistema Operativo
SoC	= System on Chip
ANT	= Agencia Nacional de Transito
CPU	= Central Processing Unit

**Jonnathan Ismael Crespo Guillén** portador de la cédula de ciudadanía N° **0302633581**. En calidad de autor y titular de los derechos patrimoniales del trabajo de titulación **“Diseño de un sistema de información seguro para el rastreo de los buses de transporte público de la parroquia Guapán mediante un dispositivo GPS”** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de este trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Azogues, **8 de marzo de 2023**

F: 

**Jonnathan Ismael Crespo Guillén**

**C.I. 0302633581**



**Jacobo Stive Minchala González** portador de la cédula de ciudadanía N° **0350007571**. En calidad de autor y titular de los derechos patrimoniales del trabajo de titulación **“Diseño de un sistema de información seguro para el rastreo de los buses de transporte público de la parroquia Guapán mediante un dispositivo GPS”** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de este trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Azogues, **8 de marzo de 2023**

F: .....

**Jacobo Stive Minchala González**

**C.I. 0350007571**