



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA DE TELECOMUNICACIONES

**DESARROLLO DE UN DISPOSITIVO IoT EN CLOUD PARA
DETERMINAR EL TIEMPO DE ARRIBO DE UN BUS DE
TRANSPORTE PÚBLICO A LA ESTACIÓN EN LA CIUDAD DE QUITO.**

**Trabajo de titulación previo a la obtención del
Título de Ingeniero en Telecomunicaciones**

AUTOR: MARJORIE GABRIELA MULLO TARCO

TUTOR: LUIS GERMÁN OÑATE CADENA

Quito, Ecuador

2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Marjorie Gabriela Mullo Tarco con documento de identificación N° 0550062319 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 04 de marzo del año 2024

Atentamente,



Marjorie Gabriela Mullo Tarco
0550062319

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

Yo, Marjorie Gabriela Mullo Tarco con documento de identificación No. 0550062319, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del proyecto técnico “Desarrollo de un dispositivo IoT en cloud para determinar el tiempo de arribo de un bus de transporte público a la estación en la ciudad de Quito.”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Quito, 04 de marzo del año 2024

Atentamente,



Marjorie Gabriela Mullo Tarco

0550062319

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Luis Germán Oñate Cadena con documento de identificación N° 1712157401, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESSARROLLO DE UN DISPOSITIVO IoT EN CLOUD PARA DETERMINAR EL TIEMPO DE ARRIBO DE UN BUS DE TRANSPORTE PÚBLICO A LA ESTACIÓN EN LA CIUDAD DE QUITO realizado por Marjorie Gabriela Mullo Tarco con documento de identificación N° 0550062319, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 04 de marzo del año 2024

Atentamente,



Ing. Luis Germán Oñate Cadena, MSc
1712157401

DEDICATORIA

Yo Marjorie Gabriela Mullo Tarco estoy muy orgulloso de dedicar mi trabajo a mis padres, Marco Mullo y Bertha Tarco, por el apoyo que me han brindado y los grandes sacrificios que han hecho para permitirme alcanzar todas mis metas y sueños, gracias por su ayuda. He establecido un camino bueno y profesional de autoprotección en la sociedad, gracias por su educación y especial atención hacia mí, así como por su confianza y amor por mí.

Asimismo, quisiera dedicar mi trabajo a mi hermana y a mi sobrina, quienes siempre me han dado la fuerza para seguir adelante con sus palabras de aliento, y cuya humildad, empatía y encanto saben alegrar mi vida.

AGRADECIMIENTO

En primer lugar, expreso mi gratitud a Dios por concederme salud y perseverancia durante mi tiempo en la universidad. También, deseo agradecer a la compañía de transporte urbano Distrito del Sur DISUTRAN S.A por facilitar la implementación de mi proyecto técnico en sus vehículos y por la confianza depositada en mi desde el principio.

Quiero expresar mi agradecimiento a la Universidad Politécnica Salesiana y a todos los profesores que compartieron sus conocimientos, contribuyendo así a que alcanzara otra meta en mi vida.

Además, agradezco a mi tutor, Luis Germán Oñate, por sus valiosas contribuciones, orientaciones, comentarios y por compartir su experiencia en el desarrollo de mi proyecto técnico.

ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN	1
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA	2
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.....	3
DEDICATORIA.....	1
AGRADECIMIENTO	2
ÍNDICE GENERAL	3
ÍNDICE DE FIGURAS	5
ÍNDICE DE TABLAS	6
RESUMEN.....	7
ABSTRACT.....	8
INTRODUCCIÓN	9
CAPÍTULO 1.....	11
1.1. Planteamiento del problema.....	11
1.2. Justificación	12
1.3. Objetivos.....	13
1.3.1. Objetivo General.....	13
1.3.2. Objetivos Específicos.....	13
1.4. Metodología.....	13
1.4.1. Metodología Descriptiva.....	13
1.4.2. Metodología Analítica	13
1.4.3. Metodología Experimental.....	14
CAPÍTULO 2	15
2. Fundamentación teórica.....	15
2.1. Sistemas Embebidos	15
2.2. Sistemas de Transporte Público	15
2.3. Tecnologías IoT	16
2.3.1. IoT en el Transporte público.....	17
2.4. Sistema de Posicionamiento global	18
2.4.1. Módulo GPS NEO 6M.....	18
2.5. Módulo ESP32.....	19

2.6.	IDE de Arduino.....	19
2.7.	Cloud.....	20
2.8.	Firestore Storage.....	20
CAPÍTULO 3		21
3.	Desglose funcional del dispositivo	21
3.1.	Capas de la arquitectura del dispositivo	21
3.1.1.	Capa de Percepción.....	22
3.1.2.	Capa de procesamiento y almacenamiento	22
3.1.3.	Capa de transporte.....	22
3.1.4.	Capa de aplicación	22
3.2.	Diagrama de bloques del dispositivo y de la aplicación.....	23
3.3.	Diagrama Esquemático	24
3.4.	Diagrama de Flujo de la aplicación	24
3.5.	Resultados Obtenidos	25
3.5.1.	Pruebas de operatividad del dispositivo.....	25
3.5.2.	Resultados del cálculo de tiempos de arribo entre estaciones de la ruta R34...27	27
CAPÍTULO 4		30
ANÁLISIS DE COSTOS		30
4.1.	Costos de Materiales	30
4.2.	Servicios Adicionales	30
4.3.	Costo Fijo.....	31
4.4.	Costo Variable	31
4.5.	Punto de Equilibrio	31
CAPÍTULO 5.....		34
CONCLUSIONES.....		34
RECOMENDACIONES		36
BIBLIOGRAFÍA		37

ÍNDICE DE FIGURAS

Figura 1: Diagrama General de un Sistema IoT.....	16
Figura 2: IoT en el transporte público.....	17
Figura 3: Modulo GPS NEO 6M.....	18
Figura 4: ESP32	19
Figura 5: Diagrama de comunicación entre el ESP32 y la aplicación	21
Figura 6: Diagrama de bloque del dispositivo	23
Figura 7: Diagrama esquemático del circuito	24
Figura 8: Diagrama de flujo del desarrollo de la aplicación	24
Figura 9: Ventana principal de la aplicación.....	25
Figura 10: Ubicación y datos del Gps	26
Figura 11: Tiempo de arribo de una estación a otra.....	27
Figura 12: Grafica Punto de Equilibrio	33

ÍNDICE DE TABLAS

Tabla 1: Tiempos entre estaciones Viaje de partida.....	27
Tabla 2: Tiempo entre estaciones Viaje de retorno	28
Tabla 3: Módulos y Materiales.....	30
Tabla 4: Costo total del proyecto.....	31
Tabla 5: Punto de Equilibrio.....	33

RESUMEN

El servicio de transporte público en la ciudad de Quito ha sido objeto de diversos cuestionamientos en su funcionamiento, convirtiéndose en una preocupación para las autoridades correspondientes. Es crucial destacar que este servicio enfrenta varios desafíos en términos de innovación, como la redundancia de rutas, la falta de satisfacción del indicador mínimo de comodidad y el incumplimiento de tiempos de ruta. En este contexto surge el siguiente proyecto, centrado principalmente en abordar el control de tiempos de los buses de transporte urbano de la ciudad de Quito, específicamente enfocado en la compañía DISUNTRAN S.A. Para lo cual se propone el diseño un dispositivo portátil que utilice el sistema de posicionamiento global (GPS) con el fin de recopilar datos, los mismos que se enviarán a través de Internet de las cosas (IoT) mediante la implementación de módulos como el ESP32 encargado de gestionar otros módulos como el GPS y el GSM. Cabe destacar que dichos módulos han sido programados utilizando el software Arduino. Para luego almacenar la información en la nube mediante una base de datos en este caso Firebase.

La información, presentada en formato de cadena (string), permitirá conocer variables relevantes para el cálculo preciso del tiempo transcurrido entre estaciones mientras el bus se encuentra en movimiento dentro de una ruta asignada.

Para el desarrollo de la aplicación móvil, se emplea un software de desarrollo denominado Visual Studio. Con esta herramienta, se logra gestionar de forma remota la ubicación y otras variables del bus, permitiendo así el cálculo en tiempo real de los intervalos entre estaciones. Esta funcionalidad se integra de manera conjunta con la plataforma Firebase, en colaboración con otras dependencias y librerías compatibles, garantizando así una aplicación multiplataforma.

Palabras clave: transporte público, tiempos, IoT, Cloud, string, GSM, GPS, ESP32, Fir

ABSTRACT

The public transport service in the city of Quito has been subject to various questions in its operation, becoming a concern for the corresponding authorities, and it is crucial to highlight that this service faces several challenges in terms of innovation, such as route redundancy, failure to meet the minimum comfort indicator, and non-compliance with route times. In this context, the following project arises, focused mainly on addressing the time control of urban transport buses in the city of Quito, specifically focused on the company DISUNTRAN S.A. For which we propose the design of a portable device that uses the global positioning system (GPS) in order to collect data, which will be sent through the Internet of Things (IoT) through the implementation of modules such as ESP32 responsible for managing other modules such as GPS and GSM. It should be noted that these modules have been programmed using Arduino software. Subsequently, the information is stored in the cloud via a database, in this case Firebase.

The information, presented in string format, will provide relevant variables for the precise calculation of the time elapsed between stations while the bus is running.

For the development of the mobile application, a development software called Visual Studio is used. With this tool, it is possible to remotely manage the location of the bus, thus enabling real-time calculation of the intervals between stations. This functionality is integrated jointly with the Firebase platform, in collaboration with other dependencies and compatible libraries, thus guaranteeing a cross-platform application.

Keywords: public transport, timing, IoT, Cloud, string, GSM, GPS, ESP32, Firebase.

INTRODUCCIÓN

El transporte público en Quito desempeña un papel fundamental en la movilización de la población, ya que el 75% de los desplazamientos dentro de la ciudad se lleva a cabo mediante este servicio. Este dato cobra relevancia debido a la diversidad de motivos que impulsan a las personas a desplazarse, tales como trabajo, educación, turismo, entre otros.

En el contexto de la elaboración de una nueva ordenanza de transporte para Quito, la Secretaría de Movilidad ha expuesto los principales desafíos que enfrenta el transporte público en la ciudad, destacando aspectos como el control de tiempos de ruta, la velocidad promedio y los intervalos entre cada unidad. Estos problemas han generado inconvenientes en la interacción entre los usuarios y el servicio de transporte público.

En respuesta a una de estas problemáticas, la empresa DISUTRAN S.A ha reconocido la necesidad de incorporar tecnologías avanzadas en su flota de buses. Es fundamental destacar que las operadoras deben asumir la responsabilidad de adquirir estas innovaciones de manera independiente. En este sentido, se propuso el desarrollo de un dispositivo IoT en la nube para el control de tiempos entre estaciones. Este mecanismo se encargará de proporcionar datos de variables obtenidas a través de un sistema GPS, haciendo uso del módulo Gsm Gprs SIM800L, el cual se encarga de transmitir datos a través de internet. Este enfoque permitirá calcular los tiempos estimados entre una estación y otra mientras el bus se desplaza dentro de la ruta asignada por la operadora.

Los datos generados se almacenarán en tiempo real en una base de datos en la nube, lo que facilitará la obtención del tiempo estimado y la ubicación actual del bus. Para acceder a esta información de manera amigable, se desarrollará una aplicación en Visual Studio con un entorno de fácil manejo. Este conjunto de soluciones tecnológicas tiene como objetivo mejorar el control de tiempos y la eficiencia del servicio de transporte público, beneficiando tanto a los usuarios como a la operadora.

Este trabajo consta de cinco capítulos, los cuales se describen a continuación:

En el Capítulo 1: Planteamiento del problema, justificación, objetivo general y específicos y la metodología.

En el Capítulo 2: Fundamentación teórica, las características técnicas y físicas de los módulos para el desarrollo del dispositivo.

En el Capítulo 3: Se expone el diseño y la parte física y lógica del dispositivo.

En el Capítulo 4: Se analizan los costos y la aceptación del equipo.

En el Capítulo 5: Se formula la conclusión y las recomendaciones, se añaden las referencias bibliográficas y los anexos correspondientes al desarrollo del software en el lenguaje propio del IDE de arduino y Visual Studio.

CAPÍTULO 1

En este capítulo se describe el planteamiento del problema, la justificación, los objetivos y la metodología a seguir.

1.1. Planteamiento del problema

En la actualidad el deficiente servicio de transporte público está identificado como uno de los problemas principales que afectan a la ciudad de Quito (Lastra Villamarin & Tasinchana Guanotasig, 2019). Por la falta de innovación tecnológica en los sistemas de los buses de transporte, así como el poco conocimiento de aplicaciones o dispositivos hechos para satisfacer las necesidades tanto de los usuarios como de las unidades de transporte de pasajeros.

Los usuarios del transporte público de Quito pierden, en promedio, unos 77 minutos en movilizarse de un lugar a otro. Según datos que se desprenden del Plan Maestro de Movilidad Sostenible del Municipio de Quito (Jonathan, 2022), ya que es común que pasen de 15 a 35 minutos de pie, varias veces dependiendo del clima o poniéndose en riesgo de ser víctimas de un asalto en las estaciones esperando la llegada del transporte público (Machado, 2021).

Al generarse incertidumbre por la demora en la llegada de la unidad de servicio de transporte público, ya que la información de la ubicación, ruta o tiempos del bus respecto a la estación en la que se encuentra el usuario, solo se reporta a un centro de control, lo que causa incertidumbre, que es un aspecto negativo en la calidad del servicio. Por lo que se plantea un prototipo que sea capaz de enfrentar la problemática de la localización y monitoreo constante para conocer su posición geográfica en tiempo real además del tiempo del tiempo de llegada del bus a la estación

1.2. Justificación

Con la finalidad de mejorar la calidad del servicio en el transporte público siendo este parte de uno de los objetivos de desarrollo sostenible de la ONU (Ecuador en directo, 2022). Debido a la reestructuración del mismo sistema en la ciudad de Quito basada en indicadores de calidad (Quito informa, 2021), se propone un sistema de monitoreo que dará seguimiento a los buses para poder informar sobre los tiempos de llegada entre estaciones en tiempo real a los usuarios para que puedan organizar sus periodos de espera en la estación gracias a la información proporcionada por el dispositivo. Disminuyendo así el nivel de estrés e inconformidad, además de evitar que el usuario se vea obligado a tomar otros servicios de transporte de coste elevado o en un caso particular se exponga al esperar prolongadamente en una estación que presenta peligro de robo u otras acciones negativas para el usuario.

Al implementar el prototipo del sistema de control de tiempos de arribo entre estaciones con el uso de las IoT en cloud se permitirá a la empresa obtener los tiempos exactos para llegar a las estaciones además de obtener datos como la velocidad, posición geográfica y otros que ayudaran a la planificación en el servicio de transporte.

1.3. Objetivos

1.3.1. Objetivo General

- Desarrollar un dispositivo en IoT y Cloud que indique el tiempo de arribo entre estaciones de una unidad de transporte público.

1.3.2. Objetivos Específicos

- Analizar las características y restricciones necesarias, para el cumplimiento de los requerimientos de la geolocalización en un bus de transporte público en la ciudad de Quito.
- Diseñar un dispositivo IoT en cloud con una interfaz intuitiva que muestre a los usuarios la posición geográfica y el tiempo de arribo a la estación de un bus de transporte público.
- Implementar el dispositivo IoT en cloud para la comprobación de su correcto funcionamiento al momento de determinar la posición de un bus y el tiempo de arribo.
- Realizar un análisis de los costos para la implementación del dispositivo IoT.

1.4. Metodología

1.4.1. Metodología Descriptiva

La siguiente metodología se utilizará para establecer todas las características de las tecnologías usadas dentro de la medición de tiempos entre estaciones de un bus de transporte público.

1.4.2. Metodología Analítica

Dentro del enfoque analítico utilizado, se llevará a cabo la evaluación de viabilidad del proyecto, considerando el análisis de los costos y presupuestos que se generen durante su implementación.

1.4.3. Metodología Experimental

Mediante la ejecución de experimentos, será posible verificar la eficacia del dispositivo al evaluar si sus respuestas concuerdan con los tiempos reales de llegada de un autobús de transporte público a una estación.

CAPÍTULO 2

2. Fundamentación teórica

En este capítulo, se presenta de manera resumida la composición esencial del dispositivo a desarrollar, destacando elementos clave como el microcontrolador ESP32 integrado al Internet de las cosas (IoT) y la nube (Cloud), el Sistema de Posicionamiento Global (GPS). Asimismo, se aborda el software base que será fundamental para la implementación de la interfaz.

2.1. Sistemas Embebidos

Es un sistema que dentro del ámbito de la computación se concibe como una entidad diseñada para llevar a cabo una o varias funciones especializadas, frecuentemente en tiempo real. Su enfoque principal está orientado hacia la consecución de objetivos específicos, enfatizando la disminución de tamaño, costos y consumo energético.

Un sistema embebido en general se compone de un microprocesador, un software que requiere una asignación precisa de cierta cantidad de memoria, así como entradas y salidas que permiten la conectividad con el entorno externo.(Torres Rodríguez & Patiño López, 2019)

2.2. Sistemas de Transporte Público

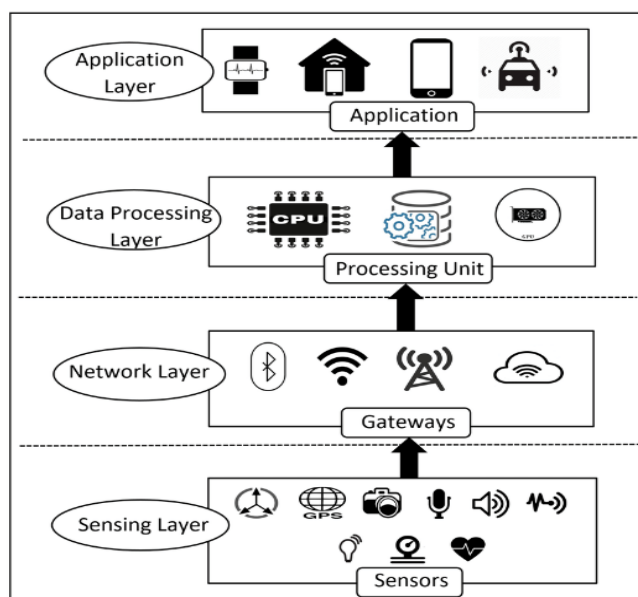
El crecimiento vertiginoso de las grandes ciudades a nivel global trae consigo el incremento sustancial en las demandas para implementar y adecuar sistemas de transporte masivos. Este fenómeno está directamente relacionado con la necesidad de mitigar el impacto ambiental, mejorar la habitabilidad en centros urbanos congestionados y optimizar la utilización de la infraestructura urbana, lo que ha impulsado a muchos países alrededor del mundo a avanzar en nuevas tecnologías, buscando abordar eficazmente los desafíos viales derivados de este crecimiento urbano. La aplicación de estas tecnologías en el manejo de extensas redes de transporte urbano se denomina Sistemas de Transporte Inteligente (ITS), que son esencialmente la fusión de avances en la informática, información tecnológica y telecomunicaciones unidas al sector automotriz experto y de transporte. Las tecnologías de comunicaciones son de gran importancia y potencial en el desarrollo de las ITS: Comunicaciones y multimedias personalizadas y portátiles, comunicaciones inalámbricas, sistemas detectores y de sensores y rastreo de vehículos.(Phil Sayeg & Phil Charles, 2013)

2.3. Tecnologías IoT

El concepto básico de IoT es la interconexión de los mundos físico, social y cibernético. Esto ha tenido un enorme impacto en la sociedad y los seres humanos, así como en las redes sociales. El motor de este impacto es la capacidad de IoT para convertir los sistemas tradicionales en sistemas inteligentes, por ejemplo, los servicios de movilidad inteligentes que proporcionan a los ciudadanos herramientas para planificar con precisión sus viajes en transporte público. (Borgia, 2014) El IoT aporta beneficios tangibles, principalmente a industrias físicas como la agricultura, la fabricación, la energía, el transporte y la sanidad. (Masinde et al., n.d.)

Al considerar la implementación de la IoT, se identifican requisitos clave, tales como la gestión de la heterogeneidad, el soporte al dinamismo, la escalabilidad, la compatibilidad con protocolos de comunicación interoperables, la rentabilidad y características como la autoconfiguración (englobando la autoorganización, autoadaptación, auto reacción a eventos y estímulos, autodescubrimiento de entidades y servicios, y auto procesamiento de Big Data). Otros aspectos cruciales incluyen la flexibilidad en la gestión dinámica y reprogramación de dispositivos o grupos de dispositivos, la garantía de calidad del servicio (QoS), la mejora de la calidad de la experiencia (QoE), el conocimiento del contexto, la capacidad de toma de decisiones inteligentes y la adhesión a entornos seguros. (Masinde et al., n.d.)

Figura 1: Diagrama General de un Sistema IoT



Autor: (Croce, 2023)

2.3.1. IoT en el Transporte público

El sector del transporte público constituye una de las áreas de mayor impacto económico a nivel global. La incorporación de la Internet de las cosas (IoT) en este sector propicia una mejora significativa en la eficiencia tanto de las industrias asociadas como del propio sistema de transporte. La funcionalidad principal de la IoT en este contexto se concentra en la monitorización de las rutas de transporte, asegurando el cumplimiento de los estándares de calidad establecidos en cada fase de la cadena de suministro.(Muñoz, 2023)

La integración de IoT en este sector permite una mayor eficiencia de las industrias y del transporte. Su funcionalidad se centra en monitorizar las rutas de transporte y comprobar que los estándares de calidad del productor son respetados en cada eslabón de la cadena.

Un sistema de transporte público inteligente, como solución interconectada, ofrece a los pasajeros la facilidad y comodidad para disfrutar de sus desplazamientos. Este sistema integra tecnologías clave como 5G, computación en la nube, paradas de autobús inteligentes y autobuses públicos inteligentes, permitiendo la participación remota de usuarios a través de aplicaciones móviles o web. A través de esta plataforma, los usuarios pueden acceder a horarios precisos y realizar un seguimiento en tiempo real de la ubicación de los autobuses, facilitando la reserva anticipada de asientos y la llegada puntual a las paradas.(Conure, 2023)

Figura 2: IoT en el transporte público



Autor:(Samar José, 2020)

2.4. Sistema de Posicionamiento global

“El Sistema de Posicionamiento Global (GPS) es un sistema de radionavegación de los Estados Unidos de América, basado en el espacio, que proporciona servicios fiables de posicionamiento, navegación, y cronometría gratuita e ininterrumpidamente a usuarios civiles en todo el mundo. A todo el que cuente con un receptor del GPS, el sistema le proporcionará su localización y la hora exacta en cualesquiera condiciones atmosféricas, de día o de noche, en cualquier lugar del mundo y sin límite al número de usuarios simultáneos.”(GPS.GOV, 2018)

2.4.1. Módulo GPS NEO 6M

Los módulos GPS NEO-6, son parte del Sistema de Posicionamiento Global (GPS), desempeñan un papel esencial al proporcionar información. Estos módulos ofrecen datos fundamentales como efemérides, almanaque, última posición, tiempo y estado del satélite.(Cuenca Sáenz & León Cando, 2017) Su precisión abarca desde metros hasta centímetros, y su funcionalidad se ve enriquecida por la posibilidad de proporcionar una señal opcional de sincronización de tiempo. (Ortiz Ordoñez & Pillalaza Quijia, 2018)Este añadido no solo reduce significativamente el tiempo necesario para la fijación inicial, sino que también mejora la sensibilidad de la adquisición. Es importante destacar que todos los módulos NEO-6 son compatibles con servicios en línea y AssistNow Offline A-GPS de u-blox 11, y cumplen con la normativa OMA SUPL.(Cuenca Sáenz & León Cando, 2017)

Figura 3: Modulo GPS NEO 6M



Autor:(AV Electronics, n.d.)

2.5. Módulo ESP32

El ESP32, desarrollado por Espressif Systems, se distingue por ser un microcontrolador de bajo costo que integra tecnologías WiFi y Bluetooth, cuenta con dos núcleos de procesamiento y eficiencia energética. Este dispositivo, además, incorpora el chip CP2102N, el cual soporta velocidades de transmisión de hasta 3Mbit/s en el modo USB/UART. Asimismo, presenta dos pulsadores con funciones de reset y boot. Su versatilidad no solo se limita a estas características, ya que también posee la capacidad de permitir que otros microcontroladores se conecten a redes inalámbricas y establezcan conexiones TCP/IP mediante comandos AT. La combinación de estas funcionalidades y su accesible precio lo convierten en una elección altamente recomendada para el desarrollo de aplicaciones orientadas a la Internet de las Cosas.(Herranz, 2019)

Figura 4: ESP32



Autor:(*Naylampmechatronics.*, n.d.)

2.6. IDE de Arduino

Un entorno de desarrollo integrado, conocido como IDE (acrónimo en inglés de integrated development environment), se define como un programa informático que reúne un conjunto de herramientas de programación. Este software puede estar diseñado exclusivamente para un lenguaje de programación específico o ser versátil para varios. En términos prácticos, un IDE se presenta como una aplicación única que incluye un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). En el caso particular del IDE de Arduino, no solo proporciona estas funcionalidades generales, sino que también incorpora las herramientas necesarias para cargar el programa compilado en la memoria flash del hardware Arduino. Es crucial comprender el funcionamiento de este IDE para trabajar eficientemente con los dispositivos Arduino.(Wordpress, 2016)

2.7. Cloud

La computación en la nube proporciona servicios en forma de infraestructuras, plataformas o software, permitiendo aprovechar el potencial creativo humano. En este contexto, los datos generados, las herramientas utilizadas y las visualizaciones creadas se vuelven secundarias, maximizando el potencial del Internet de las Cosas en diversas áreas. La nube abarca todos los aspectos de la ubicuidad al proporcionar almacenamiento escalable y herramientas para la creación de negocios. La interacción entre nubes, especialmente a través de Google Cloud, es esencial para los desarrolladores al consolidar la información, algoritmos y visualización. Además, se destaca la importancia de que un nuevo marco para aplicaciones IoT ofrezca soporte para la lectura de flujos de datos y la expresión fácil de la lógica de análisis en infraestructuras en la nube.(Gubbi et al., 2013)

2.8. Firebase Storage

Firestore de Firebase se origina como un servicio integrado en Google Cloud Platform, diseñado específicamente para la plataforma Firebase. Aunque comparte la naturaleza de base de datos NoSQL con Realtime Database, presenta notables diferencias en su estructura. Firestore se organiza mediante documentos que se agrupan en colecciones, y estos documentos pueden contener una variedad de campos que admiten diversos tipos de datos, como cadenas de texto, números, coordenadas geográficas, referencias internas a la base de datos, arrays, booleanos, marcas de tiempo e incluso objetos personalizados.(Reyes Pacheco, 2019)

CAPÍTULO 3

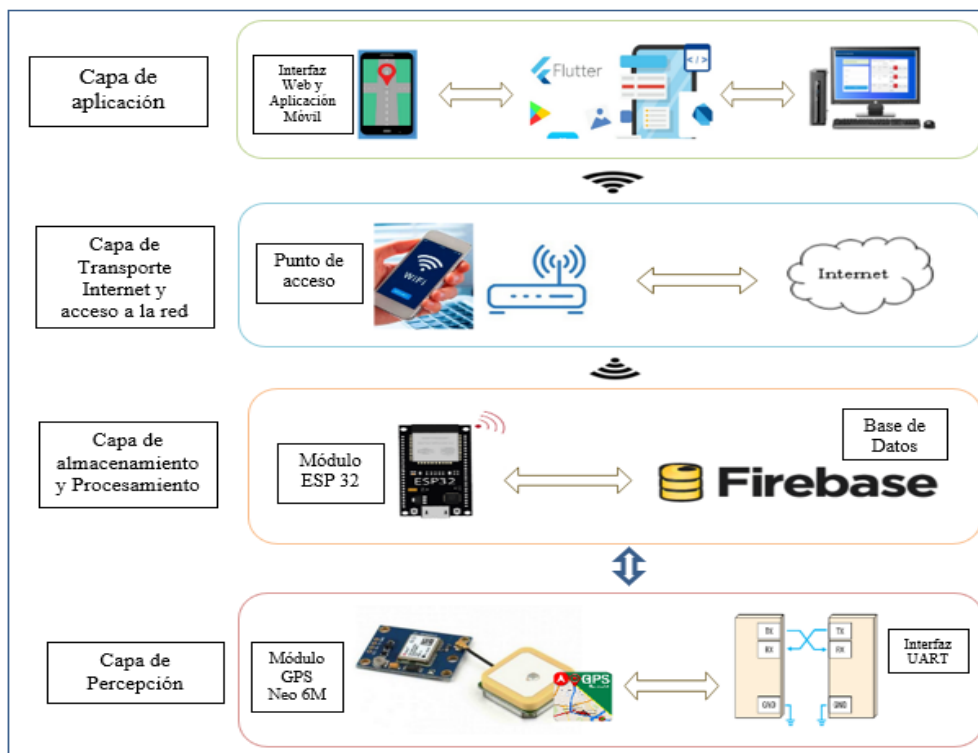
Este capítulo, trata del diseño e implementación del dispositivo, abordando tantos aspectos relacionados con el funcionamiento y la estructura, los mismos que serán detallados mediante la creación de diagramas de bloque y de flujo, que proporcionarán información sobre el diseño. Además, de examinar por bloques las diversas configuraciones de cada módulo, lo que permite evaluar y verificar las señales de transmisión y recepción.

3. Desglose funcional del dispositivo

La tecnología IoT no se rige a una arquitectura única; su estructura varía según la función que debe desempeñar. El número de capas en la arquitectura dependerá de la tarea específica. En el Foro Mundial de IoT de 2014, tanto IBM como Intel mencionaron que se pueden tener hasta siete capas. En la figura 5 se pueden distinguir cuatro de estas capas, que podrían considerarse las principales sobre las cuales se desarrollan las diversas capacidades. (Pisano, 2018)

3.1. Capas de la arquitectura del dispositivo

Figura 5: Diagrama de comunicación entre el ESP32 y la aplicación



Elaborado por: Marjorie Mullo

3.1.1. Capa de Percepción

Dentro de esta capa se integra el GPS, cuya programación se ha llevado a cabo en el IDE de Arduino, lo que permite que el módulo desempeñe la función de recopilar información, como latitud, longitud y tiempo. Estos datos son fundamentales para determinar la posición y los tiempos de desplazamiento del bus de transporte público en el que se ha instalado el dispositivo. La recopilación de estos datos se realiza mediante una interfaz UART, que es un estándar de comunicación serial utilizado para la transmisión y recepción de datos entre dispositivos, lo que asegura la interacción entre el módulo GPS y el resto del sistema, permitiendo así una monitorización precisa de la ubicación y el tiempo real del desplazamiento.

3.1.2. Capa de procesamiento y almacenamiento

Esta capa está compuesta por un microcontrolador que ayuda al desarrollo de sistemas IoT, siendo en este caso un ESP32 que se encarga de recibir y procesar los datos provenientes de la capa de percepción, específicamente del GPS, además de facilitar la conexión de dispositivos físicos a la nube a la par que establece una conexión a la red. El módulo hace conexión con la base de datos Firebase que es una estructura de la nube que almacena, sincroniza y recupera datos en tiempo real mediante Firebase Realtime Database.

3.1.3. Capa de transporte

En esta capa, se gestionan las comunicaciones generales entre los dispositivos y la nube, que se lleva a cabo a través de puntos de acceso WIFI o de una manera más técnica por medio de la implementación de los protocolos TCP O UDP sobre el protocolo IP, los mismos que facilitan la comunicación en el entorno de la red.

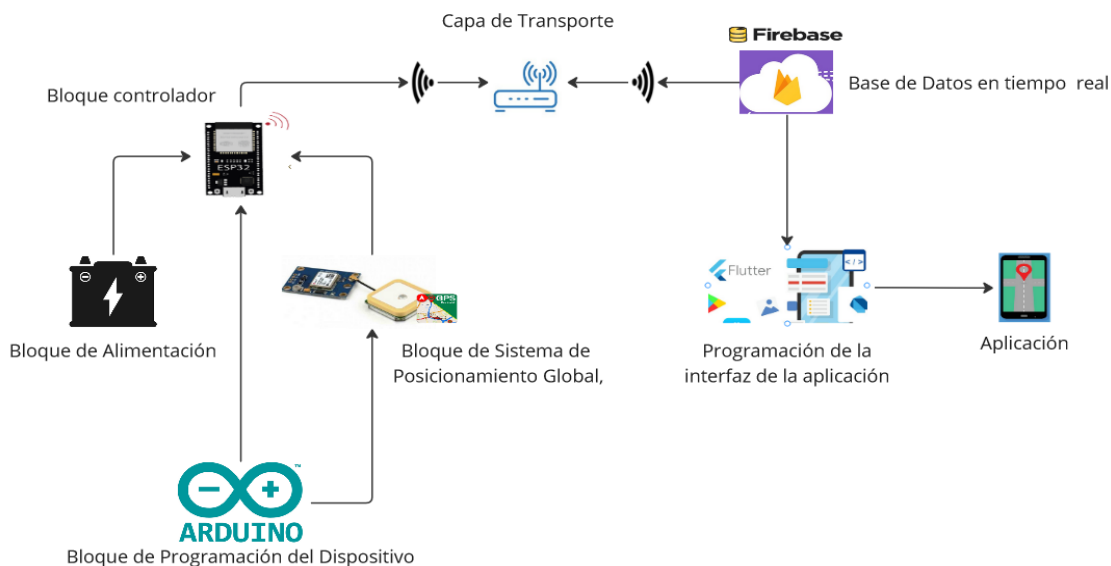
3.1.4. Capa de aplicación

Esta capa representa el nivel más alto de la arquitectura dentro de los sistemas o arquitecturas IoT , ya que dentro de esta se encuentra la interfaz a través de la cual los usuarios podrán interactuar mientras tienen acceso a internet , es decir que pueden visualizar los datos como latitud longitud y tiempo estimado de llegada entre estaciones de un bus de transporte público, así como la ubicación en tiempo real de esa unidad en específico, estos datos han sido procesados gracias a la interacción de los dispositivos, a la conexión con la nube y el uso de herramientas como Visual Studio Code como editor de código para la aplicación, y Flutter, que es un

framework que ayuda en el desarrollo rápido de aplicaciones móviles.

3.2. Diagrama de bloques del dispositivo y de la aplicación

Figura 6: Diagrama de bloque del dispositivo



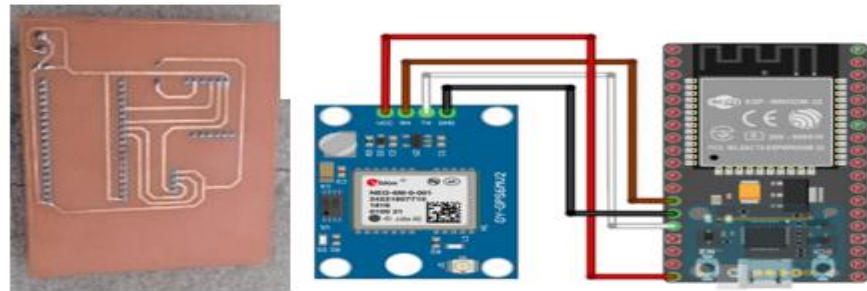
Elaborado por: Marjorie Mullo

En la Figura 6, se presentan todos los elementos que integran este proyecto, abarcando tanto la parte del dispositivo con el uso de módulos como el ESP32 y el GPS NEO 6M, como también un bloque de alimentación encargado de suministrar la energía necesaria para el funcionamiento continuo del dispositivo, por ejemplo, proporcionando 5 Voltios para la alimentación de los módulos GPS. Además, se muestra el bloque de programación del dispositivo, el cual fue llevado a cabo en la plataforma de IDE de Arduino. Esta plataforma cuenta con la flexibilidad necesaria para la instalación de bibliotecas, beneficiando tanto al bloque controlador como al bloque del GPS.

Asimismo, se tiene la sección del envío de datos a Firebase. Esta plataforma en la nube facilita la conexión con la aplicación mediante el uso de Visual Studio Code que es un entorno de desarrollo que en este caso maneja el lenguaje de programación Dart, con el cual trabaja Flutter para la creación de aplicaciones móviles. Este enfoque garantiza la eficiencia en el intercambio de datos entre el dispositivo, la nube y la interfaz de usuario móvil.

3.3. Diagrama Esquemático

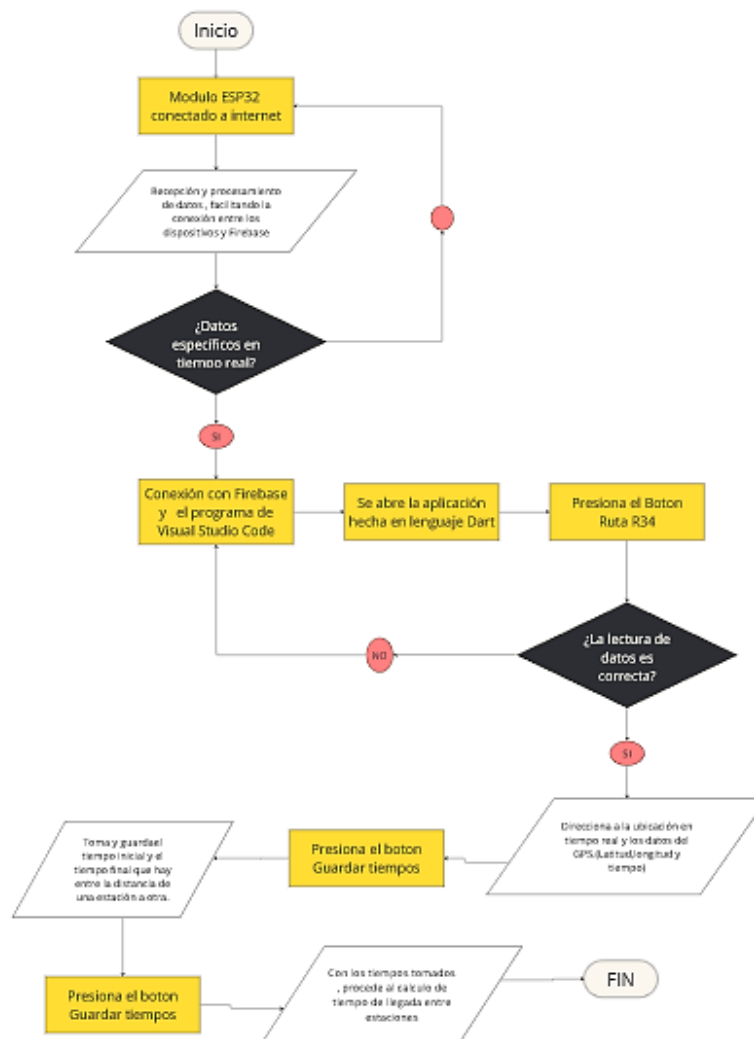
Figura 7: Diagrama esquemático del circuito



Elaborado por el autor

3.4. Diagrama de Flujo de la aplicación

Figura 8: Diagrama de flujo del desarrollo de la aplicación



Elaborado por el autor

3.5. Resultados Obtenidos

Esta sección aborda los resultados derivados de los procesos antes mencionados, abarcando tanto la parte física, representada por el dispositivo en cuestión, como la dimensión lógica, la cual comprende la aplicación y la interfaz con la que los usuarios pueden interactuar.

3.5.1. Pruebas de operatividad del dispositivo

La aplicación, representa la capa final en este servicio de IoT, permitiendo la visualización de datos previamente procesados por los módulos físicos utilizados. Al abrir la aplicación, la primera ventana principal muestra el nombre y logotipo distintivo de la empresa de transporte público DISUTRAN S.A.

Figura 9: Ventana principal de la aplicación



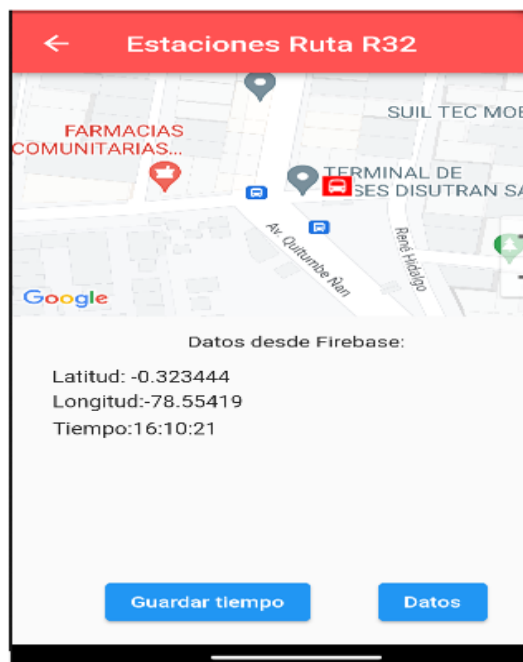
Elaborado por el autor

Además, presenta un título que destaca la selección de una de las rutas convencionales de la empresa, que cuenta con dos opciones. Razón por la cual se añadió un botón denominado "Ruta R34" indicando que la ruta parte desde Nueva Aurora hasta El Tejar, esta acotación es relevante ya que la compañía también opera otra ruta, la R32, que inicia en el Terminal Quitumbe y concluye en El Tejar. Además, cuenta con un subtítulo que aclara el propósito de la aplicación: determinar el tiempo de arribo de los buses entre estaciones.

Es importante destacar que al presionar el botón de la Ruta R34, se abrirá una nueva ventana

que integra el mapa de Google Maps. Este recurso proporciona la ubicación en tiempo real del autobús mientras se desplaza, facilitando así la visualización precisa de su posición actual. En esta pantalla además se incorpora información esencial para el desarrollo del proyecto, la cual es recopilada mediante el módulo GPS NEO 6M suministrando así datos como latitud, longitud y tiempo los mismos que pasan por la base de datos en tiempo real para luego ser reflejados dentro de la aplicación como se observa en la Figura 10.

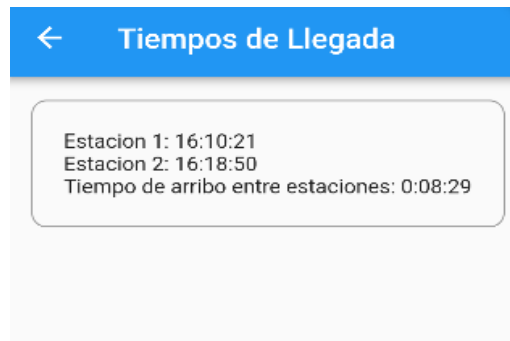
Figura 10: Ubicación y datos del Gps



Elaborado por el autor

En la siguiente ventana se encuentra el cálculo de la medición del tiempo, específicamente para calcular los tiempos de llegada entre estaciones, teniendo así un tiempo inicial, un tiempo final y un tiempo de arribo entre estaciones, los mismos que son calculados al realizar la resta de ambos valores. Se usa etiquetas de Estación 1 y Estación 2 ya que así el usuario podrá elegir en donde empieza y finaliza su ruta.

Figura 11: Tiempo de arribo de una estación a otra



Elaborado por el autor

3.5.2. Resultados del cálculo de tiempos de arribo entre estaciones de la ruta R34

Tabla 1: Tiempos entre estaciones Viaje de partida

Ruta R34 (Viaje de partida)		Tiempo Inicial	Tiempo Final	Tiempo entre estaciones	
1	Nueva Aurora	El Beaterio	16:10:21	16:18:50	0:08:29
2	El Beaterio	Capulí	16:18:34	16:27:49	0:09:15
3	Capulí	Puente de Guajalo	16:27:49	16:30:21	0:02:32
4	Puente de Guajalo	C.C. de Negocios Andinos	16:30:21	16:35:30	0:05:09
5	C. C. de Negocios Andinos	Pronaca(Sur)	16:35:30	16:37:55	0:02:25
6	Pronaca(Sur)	Mercado Mayorista	16:37:55	16:41:41	0:03:46
7	Mercado Mayorista	Unidad Educativa Consejo Provincial	16:41:41	16:45:53	0:04:12
8	Unidad Educativa Consejo Provincial	Cardenal de la Torre	16:45:53	16:46:57	0:01:04
9	Cardenal de la Torre	Ajavi	16:46:57	16:50:26	0:03:29
10	Ajavi	C.C Atahualpa	16:50:26	17:03:45	0:13:19
11	C.C Atahualpa	El Pintado	17:03:45	17:07:05	0:03:20
12	El Pintado	La Magdalena	17:07:05	17:10:29	0:03:24
13	La Magdalena	Mascota	17:10:29	17:14:16	0:03:47
14	Mascota	Panecillo	17:14:16	17:19:59	0:05:43
15	Panecillo	San Roque	17:19:59	17:27:06	0:07:07
16	San Roque	El Tejar	17:27:06	17:34:41	0:07:35
Tiempo de duración del viaje					1:24:36
Promedio de tiempo entre estaciones					0:05:17

Elaborado por el autor

Como se puede observar en la Tabla 1, la ruta se sigue de sur a norte empezando desde la Nueva Aurora, esta ruta consta de 17 paradas o estaciones.

Para el cálculo del tiempo entre estaciones se recurre a tomar datos del tiempo inicial y final, teniendo en cuenta que la resta de ambas nos da el intervalo de tiempo entre las estaciones, proceso que se realizó también dentro de la aplicación.

Tabla 2: Tiempo entre estaciones Viaje de retorno

Ruta R34 (Viaje de Retorno)			Tiempo Inicial	Tiempo Final	Tiempo entre estaciones
1	El Tejar	San Roque	17:45:35	17:52:48	0:07:13
2	San Roque	Panecillo	17:52:48	17:57:51	0:05:03
3	Panecillo	Mascota	17:57:51	18:05:35	0:07:44
4	Mascota	La Magdalena	18:05:35	18:08:40	0:03:05
5	La Magdalena	El Pintado	18:08:40	18:13:05	0:04:25
6	El Pintado	C.C Atahualpa	18:13:05	18:16:34	0:03:29
7	C.C Atahualpa	Ajavi	18:16:34	18:20:03	0:03:29
8	Ajavi	Cardenal de la Torre	18:20:03	18:28:20	0:08:17
9	Cardenal de la Torre	Unidad Educativa Consejo Provincial	18:28:20	18:31:40	0:03:20
10	Unidad Educativa Consejo Provincial	Mercado Mayorista	18:31:40	18:40:41	0:09:01
11	Mercado Mayorista	Pronaca(Sur)	18:40:41	18:45:40	0:04:59
12	Pronaca(Sur)	C. C. de Negocios Andinos	18:45:40	18:50:10	0:04:30
13	C.C. de Negocios Andinos	Puente de Guajalo	18:50:10	19:01:35	0:11:25
14	Puente de Guajalo	Capulí	19:01:35	19:05:20	0:03:45
15	Capulí	El Beaterio	19:05:20	19:10:18	0:04:58
16	El Beaterio	Nueva Aurora	19:10:18	19:17:18	0:07:00
Tiempo de duración del viaje					1:31
Promedio de tiempo entre estaciones					0:05:44

Elaborado por el autor

En esta tabla en cambio se tiene la ruta de norte a sur o de retorno, esta consta de las mismas estaciones, si se realiza una comparación entre la Tabla 1 y la Tabla 2, se nota que los tiempos de arribo entre estaciones son similares ya que solo se cambia de dirección.

Dentro de las tablas presentadas se encuentra el cálculo promedio de los tiempos entre estaciones, el cual se lleva a cabo mediante la suma de todos los tiempos y su posterior división por el número de muestras. Este proceso arroja un estimado de 0:05:44 minutos en el viaje de ida y de 0:05:17 minutos en el viaje de retorno como tiempo promedio entre estaciones.

Este resultado es significativo, ya que demuestra que tanto en los viajes de ida como de retorno, los autobuses cumplen con un tiempo previsto máximo de una 1:30:00 minutos. Este cumplimiento es esencial para el adecuado funcionamiento de la operación de la compañía, especialmente considerando que cuenta con 80 unidades de transporte público que se distribuyen en las dos rutas ofrecidas a los usuarios, lo que contribuye a mantener una gestión eficiente por parte de la compañía, beneficiando a los usuarios y asegurando un funcionamiento óptimo en sus desplazamientos diarios.

CAPÍTULO 4

ANÁLISIS DE COSTOS

Este capítulo detalla los costos de los componentes y módulos utilizados en el desarrollo del dispositivo.

Estos costos proporcionarán una perspectiva sobre si el dispositivo es adecuado para su uso en el transporte público. La viabilidad del proyecto se reflejará en los costos fijos, los costos variables y el punto de equilibrio.

4.1. Costos de Materiales

Para la elaboración del dispositivo se utilizó módulos que cumplan con las necesidades propuestas y se detallan a continuación en la Tabla 3.

Tabla 3: Módulos y Materiales

EQUIPO	CANTIDAD	VALOR UNITARIO USD	VALOR TOTAL
GPS NEO 6M V2	1	15,6	15,6
Capacitor 100 uF	2	0,1	0,2
Capacitor 1uF	1	0,6	0,6
Placa PCB	1	25	25
bateria 4,5 ma	1	12	12
Diseño 3D	1	17	17
Impresión 3D	1	30	30
ESP32	1	18	18
Servicio de datos	1	10	10
	TOTAL		128,4

Elaborado por: Marjorie Mullo

4.2. Servicios Adicionales

En la Tabla 4 se determina el precio final del dispositivo que controla los tiempos entre estaciones, en dicha tabla se suma los costos de servicio de Internet, equipo y mano de obra.

Tabla 4: Costo total del proyecto.

INVERSIÓN	VALOR (USD)
Equipo	128,4
Instalación	20
Servicio en la nube	3
Mano de Obra	25
TOTAL	176,4

Elaborado por: Marjorie Mullo

4.3. Costo Fijo

Son aquellos costos que no son sensibles a pequeños cambios en los niveles de actividad, sino que se mantienen constantes ante estos cambios. Lo opuesto a los costos fijos son los costos variables. Distinguir entre costos fijos y variables es fundamental para obtener información adecuada para tomar decisiones basadas en costos. Los costos fijos generalmente están asociados con la producción de una estructura, por lo que a menudo también se les llama costos estructurales y se utilizan para preparar informes sobre el grado de uso de la estructura. (VASQUEZ, 2020)

4.4. Costo Variable

Un costo variable es aquel que se modifica de acuerdo a variaciones del volumen de producción (o nivel de actividad), se trate de bienes o servicios. Es decir, si el nivel de actividad decrece, estos costos decrecen, mientras que, si el nivel de actividad aumenta, también lo hace esta clase de costos. (VASQUEZ, 2020)

4.5. Punto de Equilibrio

El Punto de Equilibrio me dice cuánto tengo que vender para cubrir mis Costos Totales. Refleja la situación en que los ingresos son iguales a los costos y no hay ni pérdida ni utilidad en el negocio. (VASQUEZ, 2020)

Con los datos de los costos de la tabla 3 se procede a obtener los costos fijos y los costos variables con el fin de determinar el punto de equilibrio, con un margen de utilidad del 30% y un precio estimado de venta al público de 229.32 USD.

DATOS

PVP	\$	229,32
COSTOS FIJOS	\$	128,40
COSTOS VARIABLES	\$	48,00

a) Punto de equilibrio (unidades)

$$PE(u) = \frac{CF}{PV(u) - CV(u)}$$

$$PE(u) = \frac{128,40}{181,32}$$

$$PE(u) = 0,71$$

Punto de equilibrio (precio)

$$\$PE = \frac{CF}{\frac{P - CV}{P}}$$

$$PE(\$) = \frac{\$ 128,40}{0,790685505}$$

$$PE(\$) = \$ 162,39$$

b) Margen de contribución %

$$MC = PVP - CV$$

$$MC = \$181,32$$

$$\%MC = \frac{MC}{IT} * 100$$

$$MC\% = \frac{\$ 181,32}{\$ 229,32} * 100$$

$$79\%$$

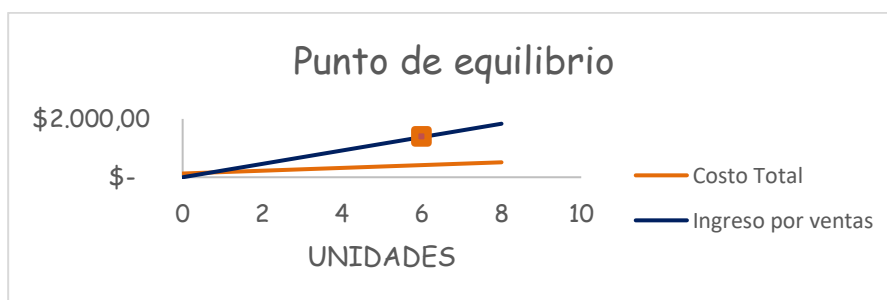
c) Punto de equilibrio (gráfica)

Tabla 5: Punto de Equilibrio

Unidades	Costo Fijo	Costo Variable	Costo Total	Ingreso por ventas	Utilidad
0	\$ 128,40	\$ -	\$ 128,40	\$ -	
1	\$ 128,40	\$ 48,00	\$ 176,40	\$ 229,32	\$ 52,92
2	\$ 128,40	\$ 96,00	\$ 224,40	\$ 458,64	\$ 234,24
3	\$ 128,40	\$ 144,00	\$ 272,40	\$ 687,96	\$ 415,56
4	\$ 128,40	\$ 192,00	\$ 320,40	\$ 917,28	\$ 596,88
5	\$ 128,40	\$ 240,00	\$ 368,40	\$ 1.146,60	\$ 778,20
6	\$ 128,40	\$ 288,00	\$ 416,40	\$ 1.375,92	\$ 959,52
7	\$ 128,40	\$ 336,00	\$ 464,40	\$ 1.605,24	\$ 1.140,84
8	\$ 128,40	\$ 384,00	\$ 512,40	\$ 1.834,56	\$ 1.322,16

Elaborado por: Marjorie Mullo

Figura 12: Grafica Punto de Equilibrio



Elaborado por: Marjorie Mullo

En la tabla 5 y la figura 12 se evidencia que para obtener un logro de beneficio cero y proceder a obtener ganancias con la distribución y producción del proyecto se deben vender 6 dispositivos con un total en ventas de 1375.92 USD.

CAPÍTULO 5

CONCLUSIONES

- Con el desarrollo y la implementación de este proyecto, se analizó más a fondo las características del Internet, que es una tecnología de tipo ubicua, por lo que fue una opción muy acertada al momento de trabajar con dispositivos que funcionan con conexión Wi-Fi como el microcontrolador ESP32 que contribuyó con la recepción, procesamiento de datos del GPS y la transmisión de los mismos a la base de datos para ser procesados nuevamente y obtener los resultados que se aprecian en la Figura 11 con el cálculo de los tiempos de arribo entre estaciones de un bus de transporte público. Estas diversas tecnologías apoyan a los avances de los sistemas de transporte inteligente o también conocido por sus siglas ITS que están enfocadas primordialmente en la eficiencia y seguridad de sus procesos.
- Se diseñó un dispositivo para el control de tiempos entre estaciones de un bus de transporte público en este caso una unidad de la compañía DISUTRAN S.A, cuya aplicación cuenta con una arquitectura en la que se diferencian varias etapas o también llamadas capas dentro del IoT, que abarca tanto la parte física como lógica, como es el tener un punto de acceso o red Wi-Fi para la conexión de los módulos usados para la transmisión de datos. Esto facilitó el diseño de una interfaz intuitiva y de fácil manejo procurando el acceso a la mayor cantidad de usuarios de diferentes edades.
- Las pruebas de implementación, se las realizó en la unidad 71 de la compañía antes mencionada, teniendo en cuenta un montaje que no obstaculice o moleste tanto a conductores, ayudantes y usuarios. La compañía cuenta con un estudio de tiempos generales entre sus estaciones por lo que el dato más relevante que se tenía era que el recorrido de ida y de retorno duraba alrededor de 1:20:00 o 1:30:00 cada una, dependiendo de distintos factores como es el día, la hora, el clima, los feriados, el tráfico etc. Lo mismo que se constató al momento de hacer las pruebas pues se tomó el tiempo en las 17 paradas que tienen dentro de su recorrido como se puede observar en la Tabla 1 y 2, las mismas que arrojaron que el dato antes mencionado coincidía con las muestras tomadas por el dispositivo ya que se obtuvo un tiempo aproximado de viaje de 1:24:36.

- Al desarrollar el dispositivo IoT se tomó en cuenta el impacto de la tecnología sobre los avances de este sector económico, ya que por sus precios accesibles al público y por ende la posibilidad de producir este tipo de dispositivos a gran escala este representa una solvencia y rentabilidad muy alta.
- Considerando el análisis de los costos al momento del diseño e implementación del dispositivo, se ha concluido que el proyecto logra alcanzar el punto de equilibrio al comercializar 6 unidades a un precio de venta al público de 229.32 usd, el cual indica que el proyecto cubre sus costos y además obtiene ganancias y no se incurre en pérdidas, manteniendo una rentabilidad del 30% , además de ser un precio accesible, lo que promoverá su aceptación en el mercado principalmente ante usuarios potenciales específicamente las compañías de transporte público de la ciudad de Quito.

RECOMENDACIONES

- Se recomienda que el dispositivo opere con una batería individual ya que así se tiene una optimización energética del consumo de energía, además que se mejora la eficiencia y se extiende la vida útil del dispositivo, ya que así los módulos recibirán la alimentación requerida según sus datos de fabricación, además de garantizar una operación continua.
- Para previas mejoras en la interfaz y aplicación del dispositivo, se sugiere considerar la implementación de protocolos de seguridad más robustos dentro de las bases de datos. Una opción que, si brinda Firebase, con un costo adicional que varía según la cantidad de memoria requerida. La elección de estos protocolos son la ventaja de tener un nivel de seguridad más avanzado que ayudara a salvaguardar los datos, garantizando la confidencialidad contra amenazas cibernéticas.

BIBLIOGRAFÍA

- Departamento de Ingeniería Eléctrica, E. y. (2017). “*Sistemas Embebidos. Controladores industriales de diseño de alto nivel.*” . Obtenido de http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia_I SE5_3_1.pdf. 2017.
- Ecuador en directo.* (11 de 11 de 2022). Obtenido de <https://ecuadorendirecto.com/2022/11/11/inseguridad-atermoriza-a-miles-de-usuarios-en-las-paradas-de-buses-de-quito/>
- Jonathan, M. (8 de 11 de 2022). *PRIMICIAS*. Recuperado el 2 de 11 de 2022, de <https://www.primicias.ec/noticias/sociedad/quito-tiempo-transporte-publico-hora-pico/>
- Machado, J. (7 de 5 de 2021). *PRIMICIAS*. Recuperado el 30 de 11 de 2022, de <https://www.primicias.ec/noticias/sociedad/quito-transporte-publico-acoso-movilidad/>
- Mishell, L. V.-T. (2020). Análisis de calidad del servicio de transporte convencional urbano del Distrito. Quito.
- P., S. P. (2013). *Sistemas de transporte inteligentes*. Alemania.
- Quito informa.* (8 de 06 de 2021). Obtenido de <http://www.QUITOINFORMA.gob.ec/2021/08/26/tecnologia-la-clave-para-la-modernizacion-del-transporte-publico/>
- RN2903, M. (s.f.). *Microchip*. Obtenido de <http://ww1.microchip.com/downloads/en/DeviceDoc/50002390E.pdf>
- VASQUEZ, J. C. (2020). COSTOS. En J. C. VASQUEZ, *COSTOS*. AGUILAR.
- AV Electronics.* (n.d.). Retrieved January 27, 2024, from <https://avelectronics.cc/producto/modulo-gps-neo-6m/>
- Borgia, E. (2014). *The Internet of Things vision: Key features, applications and open issues*. <https://doi.org/10.1016/j.comcom.2014.09.008>
- Conure. (2023). *IoT and 5G: Transforming Public Transportation System*. <https://www.iotforall.com/iot-and-5g-transforming-public-transportation-system>
- Croce, J. I. (2023). *IOT aplicado a las comunicaciones del sector del transporte*. Universidad Politécnica de Madrid.
- Cuenca Sáenz, L. E., & León Cando, J. A. (2017). “IMPLEMENTACIÓN DE UN SISTEMA DE NAVEGACIÓN TRAYECTORIA DENTRO DE LA ESPOCH.” ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO .

- GPS.GOV. (2018). *Bienvenidos a GPS.gov*. <https://www.gps.gov/spanish.php>
- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). *Future Generation Computer Systems Internet of Things (IoT): A vision, architectural elements, and future directions*. *Future Generation Computer Systems*, 29, 1645–1660. <https://doi.org/10.1016/j.future.2013.01.010>
- Herranz, Á. B. (2019). *Desarrollo de aplicaciones para IoT con el módulo ESP32*. Universidad de Alcalá.
- Lastra Villamarin, L. E., & Tasinchana Guanotasig, K. M. (2019). *Análisis de calidad del Servicio de Transporte Convencional Urbano del Distrito Metropolitano de Quito en el año 2017*. Quito: UCE. <http://www.dspace.uce.edu.ec/handle/25000/20425>
- Masinde, M., Shoman, A., & Mostafa, M. H. (n.d.). *Internet of Things-Based Framework for Public Transportation Fleet Management in Non-smart City*. https://doi.org/10.1007/978-3-030-05198-3_1
- Muñoz, M. R. (2023). *Plataforma IoT para la provisión de servicios en procesos industriales* Doctorado en Informática.
- Naylampmechatronics. (n.d.) , from <https://naylampmechatronics.com/espressif-esp/1011-nodemcu-32-38-pin-esp32-wifi.html>
- Ortiz Ordoñez, J. E., & Pillalaza Quijia, M. A. (2018). *DESARROLLO DE MÓDULOS CON RASPBERRY PI UTILIZANDO COMUNICACIONES INALÁMBRICAS, PARA EL LABORATORIO DE MICROPROCESADORES DE LA ESFOT. ESCUELA POLITÉCNICA NACIONAL* , 7.
- Phil Sayeg, & Phil Charles. (2013). *Transport Intelligent Transport Systems Module 4e Sustainable Transport: A Sourcebook for Policy-makers in Developing Cities*. Ministerio Federal de Cooperación Económica y Desarrollo de Alemania. <http://www.sutp.org>
- Pisano, A. (2018). *Internet de la Cosas*. Universidad de San Andrés .
- Reyes Pacheco, J. H. (2019). *APLICACIÓN ANDROID DE ALERTA TEMPRANA A POSIBLES RIESGOS CON LOS SERVICIOS BÁSICOS. UNIVERSIDAD MAYOR DE SAN ANDRÉS*.
- Samar José. (2020, December 22). *Cómo el “IoT” puede mejorar el transporte público. – movilidad conectada. Tecnología, V2V/V2I* . <https://movilidadconectada.com/2020/12/22/como-el-iot-puede-mejorar-el-transporte-publico/>
- Torres Rodríguez, J. A., & Patiño López, L. F. (2019). *DISEÑO E IMPLEMENTACIÓN*

DE UN SISTEMA EMBEBIDO BASADO EN IoT PARA LA GESTIÓN DEL TRANSPORTE PÚBLICO. UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS FACULTAD DE INGENIERÍA.

Wordpress. (2016, December 11). IDE Arduino | Aprendiendo Arduino. <https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/>

ANEXOS

ANEXO 1. Programación en el IDE de Arduino

Anexo 1.1. Inclusión de Bibliotecas

```
#include <Wire.h>
#include <TinyGPS++.h>
#include <WiFi.h>
#include <ArduinoHttpClient.h>
```

Anexo 1.2. Declaración de constantes

```
const char FIREBASE_HOST[] = "gpsmullo-default-rtdb.firebaseio.com";
const String FIREBASE_AUTH = "pjYNB8fY8rmYkLctY6YDOFrthmyfLnAWiczeJrIk";
const String FIREBASE_PATH = "/";
const int SSL_PORT = 443;
```

Anexo 1.3. Definición de pines

```
// GPS Module RX pin to ESP32 17
// GPS Module TX pin to ESP32 16
#define RXD2 16
#define TXD2 17
HardwareSerial neogps(2);
TinyGPSPlus gps;

HttpClient http_client;
```

Anexo 1.4. Configuración de WIFI

```
const char *ssid = "Red";
const char *password = "1234567890";
```

Anexo 1.5. Inicio de la comunicación serial, conexión a la red Wi-Fi utilizando las credenciales

```
void setup() {
  Serial.begin(115200);
  Serial.println("ESP32 serial initialize");

  // Inicia la conexión Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  neogps.begin(9600, SERIAL_8N1, RXD2, TXD2);
  Serial.println("GPS serial initialize");
  delay(3000);

  http_client.setHttpResponseTimeout(90 * 1000);
}
```

Anexo 1.6 Verificación del cliente HTTP está conectado

```
void loop() {
  while (true) {
    if (!http_client.connected()) {
      Serial.println();
      http_client.stop(); // Shutdown
      Serial.println("HTTP not connect");
      break;
    } else {
      gps_loop();
    }
  }
}
```

Anexo 1.7. Bloque de inicialización del gps

```
void gps_loop() {
  boolean newData = false;
  for (unsigned long start = millis(); millis() - start < 2000;) {
    while (neogps.available()) {
      if (gps.encode(neogps.read())) {
        newData = true;
        break;
      }
    }
  }
}

if (newData) {
  String latitude, longitude;
  unsigned long time, speed;

  latitude = String(gps.location.lat(), 6);
  longitude = String(gps.location.lng(), 6);
  time = gps.time.value();
  speed = gps.speed.kmph();

  Serial.print("Latitude= ");
  Serial.print(latitude);
  Serial.print(" Longitude= ");
  Serial.println(longitude);

  Serial.print("Time= ");
  Serial.print(time);
  Serial.print(" Speed= ");
  Serial.println(speed);

  String gpsData = "{";
  gpsData += "\"lat\": " + latitude + ",";
  gpsData += "\"lng\": " + longitude + ",";
  gpsData += "\"time\": " + String(time) + ",";
  gpsData += "\"speed\": " + String(speed);
  gpsData += "}";

  PostToFirebase("PATCH", FIREBASE_PATH, gpsData, &http_client);
}
```

Anexo 1.8 Firebase

```
void PostToFirebase(const char *method, const String &path, const String &data, HttpClient *http) {
  String response;
  int statusCode = 0;
  http->connectionKeepAlive();

  String url;
  if (path[0] != '/') {
    url = "/";
  }
  url += path + ".json";
  url += "?auth=" + FIREBASE_AUTH;

  String contentType = "application/json";
  http->put(url, contentType, data);

  statusCode = http->responseStatusCode();
  Serial.print("Status code: ");
  Serial.println(statusCode);
  response = http->responseBody();
  Serial.print("Response: ");
  Serial.println(response);

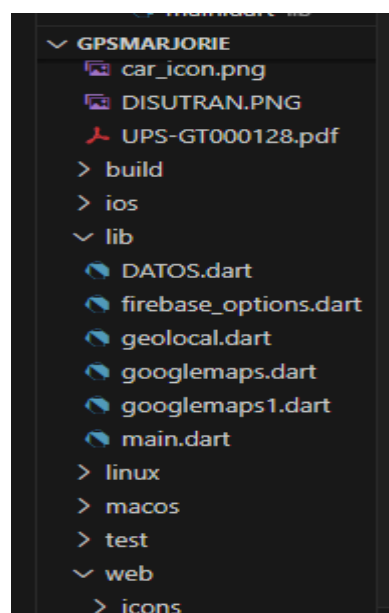
  if (!http->connected()) {
    Serial.println();
    http->stop();
    Serial.println("HTTP POST disconnected");
  }
}
|
```

Anexo 2. Programación en Visual Studio

Anexo 2.1 Partes de la aplicación

Programas

Lugar donde se cargan las imágenes que van a ser usadas dentro de la aplicación y los archivos creados para su precio uso.



Anexo 2.2. Ventana de la aplicación:

Importación de Paquetes y librerías

```
import 'package:flutter/material.dart';  
import 'package:gpsmarjorie/googlemaps.dart';  
//import 'package:firebase_core/firebase_core.dart';  
//import 'package:cloud_firestore/cloud_firestore.dart';
```

Anexo 2.3 Creación de una Clase

Devuelve la representación visual del widget.

```
class HomePage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    double screenHeight = MediaQuery.of(context).size.height;  
    double imageHeight =  
      screenHeight * 0.4; // Ajusta este valor  
      screenHeight * 0.05; // Ajusta este valor
```

Anexo 2.4 Aplicación de Widget

```
return Scaffold(  
  appBar: AppBar(  
    title: Container(  
      color: Colors.orange, // Fondo naranja para el título  
      padding: EdgeInsets.all(  
        10), // Espaciado interno para el contenido del Container  
      child: Text(  
        "DISUTRAN S.A", // Agrega el título "DISTBUS"  
        textAlign: TextAlign.center,  
        style: TextStyle(  
          fontSize:  
            titleFontSize, // Tamaño de fuente ajustado según el tamaño  
de la pantalla  
          fontWeight: FontWeight.bold,  
          color: Colors.white, // Texto en color blanco  
        ),  
      ),  
    ),  
  ),  
  centerTitle: true, // Centra el título en la AppBar
```

```

body: Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Image.asset(
      'assets/DISUTRAN.PNG', // Ruta de la imagen en assets
      height: imageHeight,
    ),
    SizedBox(
      height:

```

```

      SizedBox(
        height:
          20, // Espacio entre la imagen y el texto "Ruta convencional"
      ),
      Text(
        'Ruta convencional', // Nuevo texto "Ruta convencional"
        style: TextStyle(
          fontSize: 24, // Tamaño de fuente mayor
          fontWeight: FontWeight.bold,
        ),
      ),
      SizedBox(
        height:
          20, // Espacio entre el texto "Ruta convencional" y "Tiempo
estimado de llegada"
      ),
      Text(
        'Tiempo estimado de llegada',
        style: TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
  ),

```

```

    ),
  ),
  SizedBox(
    height: 20,
  ), // Espacio entre el texto "Tiempo estimado de llegada" y el botón
  ElevatedButton(
    onPressed: () {
      // Navegar a la página de Google Maps cuando se presione el botón
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => GoogleMapPage()),
      );
    },
    style: ElevatedButton.styleFrom(
      primary: Colors.orange,
    ),
    child: Text('Ruta R34'),
  ),
),
),
);
}

```


ANEXO 3

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_database/firebase_database.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'package:gpsmajorie/DATOS.dart'; // Asegúrate de importar el archivo
DATOS.dart correctamente

class GoogleMapPage extends StatefulWidget {
  const GoogleMapPage({Key? key}) : super(key: key);

  @override
  _GoogleMapPageState createState() => _GoogleMapPageState();
}

class _GoogleMapPageState extends State<GoogleMapPage> {
  final databaseReference = FirebaseDatabase.instance.reference();
  Map<String, dynamic> dataFromFirebase = {};
  late GoogleMapController mapController;

  @override
  void initState() {
    super.initState();
    Firebase.initializeApp();
    _readDataFromFirebase();
  }

  void _readDataFromFirebase() {
    databaseReference.onValue.listen((event) {
      if (event.snapshot.value != null) {
        setState(() {
          dataFromFirebase = event.snapshot.value as Map<String, dynamic>;
        });
      }
    }, onError: (error) {
      print("Error leyendo los datos: $error");
    });
  }

  void _navegarADatos() {
    // Navegar a la pantalla DATOS.dart
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => DATOSPage(data: dataFromFirebase),
      ),
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Estaciones Ruta R32"),
        backgroundColor: Colors.redAccent,
      ),
      body: Column(
        children: [
          Expanded(
            child: GoogleMap(
```

```

child: Container(
  padding: EdgeInsets.all(16.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text("Datos desde Firebase: "),
      Text("Longitud: ${dataFromFirebase['longitud'] ?? 'N/A'}"),
      Text("Latitud: ${dataFromFirebase['latitud'] ?? 'N/A'}"),
      Text("Tiempo: ${dataFromFirebase['tiempo'] ?? 'N/A'}"),
    ],
  ),
),
),
),
Container(
  padding: EdgeInsets.all(16.0),
  child: Center(
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
        ElevatedButton(
          onPressed: () {
            // Código para guardar tiempo
          },
          child: Text("Guardar tiempo", style: TextStyle(fontSize: 14)),
        ),
        ElevatedButton(
          onPressed: () {
            _navegarADatos(); // Llama a la función cuando se presiona el botón "Datos"
          },
          child: Text("Datos", style: TextStyle(fontSize: 14)),
        ),
      ],
    ),
  ),
),
),
),
],
),
);
}

void main() {
  runApp(MaterialApp(
    home: GoogleMapPage(),
  ));
}

```

Anexo 4 Llave de la base de Datos

```
"project_info": {
  "project_number": "325122479055",
  "firebase_url": "https://[redacted]aseio.com",
  "project_id": "gpsmullo",
  "storage_bucket": "gpsmullo.appspot.com"
},
"client": [
  {
    "client_info": {
      "mobilesdk_app_id": "1:325122479055:android:08874b76634209ee1138ba",
      "android_client_info": {
        "package_name": "com.example.gpsmarjorie"
      }
    },
    "oauth_client": [
      {
        "client_id": "325122479055-eg7qhsuv92dvvfef5eskvbvhik2f9nrdm.apps.googleusercontent.com",
        "client_type": 3
      }
    ],
    "api_key": [
      {
        "current_key": "AIzaSyBJ89i7snfYwkGFsY9M8lhUBtNI45mT9fk"
      }
    ],
    "services": {
      "appinvite_service": {
        "other_platform_oauth_client": [

```

```
{ } firebase_app_id_file.json > ...
Click here to ask Blackbox to help you code faster
{
  "file_generated_by": "FlutterFire CLI",
  "purpose": "FirebaseAppID & ProjectID for this Firebase app in this directory",
  "GOOGLE_APP_ID": "1:325122479055:ios:7cdacc71fda25bf51138ba",
  "FIREBASE_PROJECT_ID": "gpsmullo",
  "GCM_SENDER_ID": "325122479055"
}
```

Especificación de paquetes en Flutter

Pubspec.yaml: Google maps en Flutter - Cloud Firestore

```
# Of the product and file versions while build number is used as the build suffix.
version: 1.0.0+1

environment:
  sdk: '>=3.0.6 <4.0.0'

# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  google_maps_flutter: [redacted]
  location: ^5.0.0
  geolocator: ^9.0.2
  #firebase_core: ^2.15.0
  cloud_firestore: ^4.8.3
  #firebase_database: ^10.2.4

flutter:
  sdk: flutter

# The following adds the Cupertino Icons font to your application.
```