



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA DE INGENIERÍA ELECTRÓNICA

**DESARROLLO DE UN DISPOSITIVO PARA LOCALIZACIÓN Y
OBTENCIÓN DE PARÁMETROS DE VEHÍCULOS MEDIANTE UNA
APLICACIÓN MÓVIL**

Trabajo de titulación previo a la obtención del
Título de Ingeniero Electrónico

AUTOR: Wilson Stalin Quimbita Tonato

TUTOR: Gustavo Javier Caiza Guanochanga

Quito-Ecuador

2023

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, Wilson Stalin Quimbíta Tonato, con documento de identificación No. 1722994306; 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, 6 de marzo del año 2023.

Atentamente,



Wilson Stalin Quimbíta Tonato

1722994306

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

Yo, Wilson Stalin Quimbita Tonato, con documento de identificación No. 1722994306, 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 el autor del proyecto técnico: “Desarrollo de un dispositivo para localización y obtención de parámetros de vehículos mediante una aplicación móvil”, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, 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, 6 de marzo del año 2023.

Atentamente,



Wilson Stalin Quimbita Tonato


1722994306

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Gustavo Javier Caiza Guanochanga con documento de identificación No. 1721192191 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UN DISPOSITIVO PARA LOCALIZACIÓN Y OBTENCIÓN DE PARÁMETROS DE VEHÍCULOS MEDIANTE UNA APLICACIÓN MÓVIL, realizado por Wilson Stalin Quimbita Tonato, con documento de identificación No. 1722994306, 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, 6 de marzo del año 2023.

Atentamente,



Ing. Gustavo Javier Caiza Guanochanga, MSc
1721192191

DEDICATORIA

Este trabajo, si bien ha requerido de esfuerzo y mucha dedicación, no hubiese sido posible su finalización sin la cooperación desinteresada de todas y cada una de las personas que me acompañaron en el recorrido laborioso de este trabajo y muchas de las cuales han sido un soporte muy fuerte en momentos de angustia y desesperación, primero y antes que todo, dar gracias a Dios, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio, dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme el haber llegado hasta este momento tan importante de mi formación profesional. A mi padre, por ser el pilar más importante y por demostrarme siempre su cariño y apoyo incondicional sin importar nuestras diferencias de opiniones, a enseñarme que nunca hay de darse por vencido por más difícil que parezca a mis hermanos, que han sido el apoyo fundamental para lograr los objetivos propuestos, ya que, con su ejemplo y amor profundo, me encaminaron a seguir con la propuesta investigativa quienes, siempre me dieron esperanzas y tuvieron fe en mí.

Sobre todo, quiero dedicarle este trabajo tan importante a ese ser de luz que hace que mis días sean maravillosos. Que con ese hermoso regalo que me dio que es mi hermoso hijo me alegran mis días grises. Gracias por ser mi cómplice mi confidente, mi amiga, por ayudarme a crecer por amarme, por ser tan tú, pero sobre todo gracias por nunca cortarme las alas, ¡sí te lo dedico a ti Liliana! Porque mi corazón te pertenece. Te amo

AGRADECIMIENTOS

En estas líneas quiero agradecer a todas las personas que hicieron posible este trabajo de titulación y que de alguna manera estuvieron conmigo en los momentos difíciles, alegres, y tristes. Estas palabras son para ustedes. A mis padres por todo su amor, comprensión y apoyo, pero sobre todo gracias infinitas por la paciencia que me han tenido. No tengo palabras para agradecerles las incontables veces que me brindaron su apoyo en todas las decisiones que he tomado a lo largo de mi vida, unas buenas, otras malas, otras locas. Gracias por darme la libertad de desenvolverme como ser humano.

A mis hermanos por llenarme de alegría día tras día, por todos los consejos brindados, por compartir horas y horas de películas, series, los gritos y herir mi cuerpo de puro amor.

A mis amigos. Con todos los que compartí dentro y fuera de las aulas. Aquellos amigos del cole, que se convierten en amigos de vida y aquellos que serán mis colegas, gracias por todo su apoyo y diversión.

No puedo dejar de agradecerte especialmente a ti Jhonny Mejía, mi compañero fiel de Universidad, y ahora de corazón y vida.

ÍNDICE

1	CAPITULO 1	1
1.1	Problema de estudio	1
1.2	Justificación	2
1.3	Objetivos	4
1.3.1	Objetivo General.....	4
1.3.2	Objetivos Específicos	4
1.4	Fundamentación teórica	5
1.4.1	Sistemas de control.....	5
1.4.2	Internet de las cosas (IoT).....	7
1.4.3	Estructura de una red	8
1.4.4	Protocolo de comunicación en redes IoT.....	10
1.4.5	Red WSN.....	12
1.4.6	Nube.....	13
1.4.7	GPS.....	14
1.4.8	Geolocalización	14
1.4.9	Sistemas embebidos.....	14
2	CAPÍTULO 2	16
2.1	Hardware.....	16
2.1.1	TTGO T-Call ESP32 SIM800L.....	16
2.1.2	SMS chip telefónico	16
2.1.3	Módulo GPS NEO 6M	17
2.1.4	Batería.....	17
2.2	Software	18

2.2.1	Arduino IDE	18
2.2.2	MQTT Corredor (Broker).....	18
2.2.3	Servidor en la Nube basado en Linux.....	19
2.2.4	Servidor web Flask	19
2.2.5	Mapa de geolocalización HERE.....	19
2.2.6	Aplicación de usuario basada en HTML y JavaScript.....	19
3	CAPÍTULO 3	21
3.1	Implementación del sistema.....	21
3.1.1	Datos Móviles.....	21
3.1.2	Librerías Arduino IDE.....	21
3.1.3	Publicación/Subscripción nube.....	26
3.1.4	Servidor en la nube Digital Ocean.....	27
3.1.5	Mqtt Broker utilizado para publicar mensajes en el módulo TTGO T-Call ESP32 SIM800L.....	28
3.1.6	Servidor Web Flask	28
3.1.7	Geomapas HERE.....	29
3.1.8	Interfaz de usuario creada sobre HTML/ Java Script	30
3.1.9	Conexión de usuario con el sistema a través de un dispositivo móvil.....	32
3.1.10	Diagrama completo del sistema.....	32
4	CAPÍTULO 4	34
4.1	Análisis de Funcionamiento.....	34
4.1.1	Cálculo de la precisión y exactitud en una misma locación	34
4.1.2	Cálculo del error absoluto en distintas locaciones.....	36
4.1.3	Descarga de la batería.....	40
4.1.4	Tiempo de respuesta del sistema GPS.....	40

5	CONCLUSIONES.....	42
	RECOMENDACIONES	44
	Lista de Referencias.....	1
	ANEXOS	6

ÍNDICE DE FIGURAS

Figura 1.1. Internet de las cosas	7
Figura 1.2. Capas del modelo OSI.....	9
Figura 1.3. Protocolos de comunicación en entorno doméstico	10
Figura 1.4. Protocolos de comunicación en entorno industrial	11
Figura 2.1. ESP32 SIM Card SIM800L T-Call V1.3	16
Figura 3.1. Diagrama de flujo del programa grabado en el microcontrolador parte1. ...	22
Figura 3.2. Diagrama de flujo del programa grabado en el microcontrolador parte 2. ...	23
Figura 3.3. Diagrama de flujo del programa grabado en el microcontrolador parte 3. ...	24
Figura 3.4. Diagrama de flujo del programa grabado en el microcontrolador parte 4. ...	25
Figura 3.5. Comunicación entre microcontrolador y el servidor en la nube	27
Figura 3.6. Prueba de comunicación con el protocolo MQTT.	28
Figura 3.7. Servidor Web Flask en ejecución.....	29
Figura 3.8. Mapa extraído de Here, que representa a una parte de la ciudad de Quito. ...	30
Figura 3.9. Interfaz gráfica web aplicación final. Esquema constitutivo.	31
Figura 3.10. Interfaz gráfica web aplicación final.	31
Figura 3.11. Validación para el ingreso a la aplicación.....	32
Figura 3.12. Diagrama esquemático del sistema	33
Figura 4.1. Distancias obtenidas en latitud y longitud con respecto al valor real (0,0) del módulo GPS.....	35
Figura 4.2. Imagen comparativa de geoposicionamiento entre los dos sistemas GPS. A) Módulo GPS desarrollado, B) GPS de Google Maps.....	37
Figura 5.3. Imagen de error absoluto para los dos sistemas GPS.....	39
Figura 4.4. Descarga del banco de baterías en función del tiempo.	40
Figura 4.5. Tiempo de respuesta del sistema GPS.....	41

ÍNDICE DE TABLAS

Tabla 2.1. Características del módulo GY-NEO06MV2.....	17
Tabla 2.2. Energía consumida mA.	17
Tabla 4.1. Datos de longitud y latitud calculados a partir de datos de longitud y latitud adquiridos por el módulo GPS.....	35
Tabla 4.2. Resumen de cálculos de exactitud y precisión efectuados para el módulo GPS.	36
Tabla 4.3. Datos de longitud y latitud adquiridos por dos métodos, el desarrollado por nosotros y el de google maps.....	37
Tabla 4.4. Distancia de error en metros calculada de longitud y latitud de los métodos; el desarrollado por nosotros y el de Google Maps.	38

RESUMEN

El presente proyecto alude al Desarrollo de un dispositivo para localización y obtención de parámetros de vehículos mediante una aplicación móvil mediante la puesta en marcha de una red IoT (internet de las cosas por sus siglas en inglés) con el propósito de localizar un dispositivo que proporciona su ubicación exacta mediante GPS incorporado, valiéndose de (1) un sistema embebido microcontrolado, (2) un servidor en la nube y (3) un servicio web con interfaz gráfica. El sistema [embebido] microcontrolado se encarga de la adquisición de datos de ubicación a través de un sensor GPS y conexión a internet por medio de una red GSM [monitoreando los datos de geoposición adquiridos por un sensor], procesa la información, mantiene el sistema en modo reposo con el objetivo de ahorrar energía y se encarga de subir los datos obtenidos a un servidor en la nube. El servicio en la nube se comunica al sistema microcontrolado utilizando la red GSM. Para este servicio se ha rentado capacidad computacional en una empresa y con la ayuda de programas desarrollados y ejecutados sobre este, se logra monitorizar el dispositivo microcontrolado. Este servicio incluye tareas sobre MQTT, los cuales permiten la transmisión y asociación de datos desde el microcontrolador hasta el servicio web. El servicio web levantado sobre el servidor en la nube cumple el propósito de presentar la información ordenada con la ayuda de una interfaz gráfica. Esta interfaz permite visualizar la ubicación del dispositivo en un mapa en tiempo real, ya sea en una computadora o un equipo de telefonía móvil con acceso a internet, con lo cual se asegura un continuo monitoreo del lugar en donde se encuentra el dispositivo, siempre y cuando éste se encuentre encendido.

ABSTRACT

This project refers to the development of a device for locating and obtaining vehicle parameters through a mobile application through the implementation of an IoT network (internet of things for its acronym in English) with the purpose of locating a device that provides its exact location via built-in GPS, using (1) a micro-controlled embedded system, (2) a cloud server, and (3) a web service with a graphical interface. The microcontrolled [embedded] system is in charge of acquiring location data through a GPS sensor and internet connection through a GSM network [monitoring the geoposition data acquired by a sensor], processes the information, keeps the system in sleep mode in order to save energy and is responsible for uploading the data obtained to a server in the cloud. The cloud service communicates to the micro-controlled system using the GSM network. For this service, computing capacity has been rented in a company and with the help of programs developed and executed on it, it is possible to monitor the microcontrolled device. This service includes tasks on MQTT, which allow the transmission and association of data from the microcontroller to the web service. The web service built on the server in the cloud fulfills the purpose of presenting the ordered information with the help of a graphical interface. This interface allows you to view the location of the device on a map in real time, either on a computer or a mobile phone with Internet access, which ensures continuous monitoring of the place where the device is located, as long as when it is turned on.

1 CAPITULO 1

ANTECEDENTES

1.1 Problema de estudio

En la República del Ecuador, según datos publicados por el Ministerio de Gobierno, durante el año 2021 fueron robados un total de 5504 automóviles, registrando un aumento de 20% respecto a los casos registrados del mismo delito durante el año 2020 (González, 2022).

De acuerdo con Sánchez y Pazmiño (2011, pág. 1), los sistemas de alarma con los que los vehículos cuentan de forma general resultan obsoletos debido a dos factores: “los elevados índices de delincuencia (...) y su perfeccionamiento técnico”. Esto, evidentemente, deviene en la necesidad de la creación y actualización de tecnologías modernas y más seguras con las cuales satisfacer las necesidades del mercado vehicular, ya que, debido a sus propias características, es uno de los más susceptibles a los ataques de la delincuencia.

Desde la invención del automóvil con motor eléctrico, cuyos primeros prototipos datan incluso desde 1832-1839 por mano de Robert Anderson (Sanz Arnaiz, 2015, pág. 11), o los de motor por combustión interna, cuya primera patente fue registrada por Karl Benz “el 29 de enero de 1886” (Contreras, 2021), la tecnología automotriz ha evolucionado de tal modo que, hoy en día, los fabricantes de automóviles apuntan ya no solo a la funcionalidad principal de la conmutación, sino que propenden a facilitar la mejor experiencia de conducción; esto implica también características de seguridad, no únicamente en cuanto a salvaguardar la integridad de los ocupantes, sino también la del vehículo, ya que puede tratarse de un objeto realmente valioso, cuyo propietario deseará proteger de las mejores maneras posibles.

Entre otras, una de las tecnologías más fiables que se utilizan en la actualidad para proteger un vehículo es la del rastreo por Sistema de Posicionamiento Global (GPS) Esta tecnología permite conocer la ubicación exacta —estacionaria o en movimiento— de un objeto sobre la superficie terrestre, por lo cual, al ser implementada en los automóviles, facilita su ubicación y rastreo en caso de extravío o hurto.

Con el presente trabajo de investigación se pretende desarrollar e implementar una aplicación móvil que, con el uso de diferentes algoritmos, provea una mejor triangulación de datos de geoposicionamiento para disminuir el margen de error que a propósito tienen los GPS de uso civil y, además, se busca proporcionar acceso a funciones como bloqueo/desbloqueo de portezuelas y

funcionamiento del vehículo, además de un sistema de alertas vía mensaje escrito en caso de vulneración del vehículo. Esto se logrará a través de la instalación de un GPS miniatura fusionado con un sistema GSM que se encuentre en estado de hibernación, lo cual garantizaría la duración de carga de su batería por un tiempo estimado entre 4 y 6 meses sin que el conductor o dueño del vehículo tenga la necesidad de recargarla. Todo esto pretende evitar que el GPS quede inactivo al ser desconectado de una fuente de alimentación del vehículo, al mismo tiempo que evitaría su detección por medios convencionales.

A pesar de sus ventajas, cabe mencionar que, por sus propias características y costo adquisitivo, este tipo de tecnología y productos cuentan con una demanda pequeña en el mercado; en tal virtud, apremia el desarrollo de dispositivos que satisfagan las necesidades de los clientes, ofreciendo productos de alta calidad y confiabilidad a un precio asequible para la mayoría de la población.

1.2 Justificación

Entre los problemas que más afectan a la tecnología de rastreo y geolocalización a través de GPS están: la precisión con la que los objetos pueden geolocalizados y la autonomía de la batería. En tal sentido, la importancia de un proyecto como el propuesto en el presente documento es fundamental, debido a que sus características facultarían el aprovechamiento de la tecnología GPS, a la vez que responderían a la necesidad de evitar fallos debido a cortes inusitados de corriente — ya sean estos accidentales o provocados—, lo cual ocasionaría la desactivación del sistema.

Este proyecto, por tanto, se enfoca en crear un GPS totalmente autónomo, ya que no dependería de la energía del vehículo, ya que, al estar programado para permanecer en modo hibernación, su funcionamiento sería ejecutado únicamente al momento en que el usuario lo active mediante su aplicativo móvil. Así, en caso de robo, por ejemplo, los dispositivos facilitarían la ubicación y detección del automotor en tiempo real mientras que quienes estén conduciendo el vehículo no sospecharían, siquiera que están siendo rastreados. Dicha información podría ser consecuentemente compartida igualmente en tiempo real con las autoridades competentes, facilitando dar con el paradero del vehículo en el menor tiempo posible para proceder con su recuperación.

Otra aplicación posible relacionada con la seguridad de los dueños y conductores de automóviles, es la del cuidado a través de rastreo y seguimiento para evitar casos de secuestro o extravío. En caso de sentir alguna situación de peligro, el conductor sería capaz de compartir sigilosamente con personas y/o entidades de confianza su ubicación a través del GPS vehicular, lo cual facilitaría su

rastreo de forma inmediata para proceder a realizar un acompañamiento y seguimiento oportuno en caso de ser necesario.

Lo mismo sucedería con el rastreo y geolocalización de valores y mercaderías, ya que esta tecnología ayudaría a evitar desvíos innecesarios o sospechosos por parte de conductores contratados ya que, en caso de que por alguna razón se produzcan cambios inadvertidos de las rutas programadas, el sistema de alertas notificará de inmediato a los interesados sobre las novedades ocurridas, a fin de que se tomen las correcciones necesarias según los intereses de los clientes. Con el seguimiento de flotas por GPS, se evita muchos errores a medida que se recopila datos exactos de conmutación. Esta tecnología avanzada debería ser totalmente adoptada por las agencias de flotas, si lo que se desea es mejorar la gestión y los servicios al cliente (Geotab, 2023)

Particularmente, en el caso del Ecuador, y como ya se dijo sobre la alarmante situación de los robos de vehículos automotores, se vuelve necesario desarrollar este tipo de dispositivos de protección antirrobo con el fin de minimizar las pérdidas de automóviles y autopartes, creando así una sensación de mayor seguridad en la ciudadanía, ya que, al contar con esta tecnología, las personas pueden movilizarse con mayor confianza sabiendo que, en caso de robo de su vehículo, existe un alto porcentaje de probabilidades de que sea recuperado (El Universo, 2009).

En cuanto a las posibles fuentes de error en los cálculos realizados por GPS, es necesario mencionar que, además de la degradación intencional de precisión con la que cuentan los GPS de uso civil debido a razones de seguridad, lo cual ocasiona que la exactitud de la geolocalización tenga un margen de error de entre 6 y 15 metros, existe el fenómeno conocido como efecto multiproyecto. Este se produce debido al rebote de la señal satelital GPS sobre estructuras cercanas como edificios y montañas que están por encima del objeto rastreado. Esto implica, en efecto, que, el receptor GPS detecte dos veces una misma señal en diferentes rangos; sin embargo, la utilidad del proyecto propuesto también apunta a responder a dichas fuentes de error ya que, un receptor GPS adecuadamente diseñado puede alcanzar una precisión horizontal de 3m o mejor y una precisión vertical de 5m o mejor el 95% del tiempo durante el rastreo. Los sistemas GPS aumentados pueden proporcionar precisión submétrica (GISGeography, 2022)

1.3 Objetivos

1.3.1 Objetivo General

- Desarrollar un dispositivo para localización y obtención de parámetros de vehículos mediante un aplicativo móvil.

1.3.2 Objetivos Específicos

- Realizar la revisión bibliográfica para la determinación del estado del arte de soluciones del margen de exactitud de los dispositivos GPS.
- Desarrollar un dispositivo para la localización de los vehículos para el cumplimiento de los requerimientos definidos en el estado del arte.
- Implementar un algoritmo de triangulación que ayude a reducir el margen de error de distancia que presentan dispositivos GPS convencionales.
- Desarrollar una aplicación para dispositivo móvil que permita operar de manera remota y así dar seguimiento de su funcionamiento.
- Analizar los resultados, mediante pruebas experimentales para la validación del dispositivo

1.4 Fundamentación teórica

1.4.1 Sistemas de control

Los sistemas de control son conjuntos de varios elementos y/o subsistemas interconectados y dispuestos de tal modo que, “ante unos objetivos determinados responde[n] con una serie de actuaciones” (Ñeco, et al, 2003, pág. 4). En palabras de Brunete, et al, “un sistema de control es aquel en el que las variables de salida se comportan según las órdenes dadas por las variables de entrada” (2020, pág. 26); y estos sistemas de control se producen a fin de garantizar la obtención de resultados deseados y, al mismo tiempo, facultar la eliminación o reducción de fallos.

Además, la necesidad de la presencia de un sistema de control en el área de la automatización responde a la búsqueda del cumplimiento de al menos tres objetivos; estos son:

- a. Garantizar la estabilidad y, particularmente, ser robusto frente a perturbaciones y errores en los modelos.
- b. Ser tan eficiente como sea posible, según un criterio preestablecido. Normalmente este criterio consiste en que la acción de control sobre las variables de entrada sea realizable, evitando comportamientos bruscos e irreales.
- c. Ser de fácil implementación y cómodo de operar en tiempo real con ayuda de un ordenador (Álvarez Brotons, 2004, pág. 5).

1.4.1.1 Elementos de un sistema de control

No parece existir un acuerdo entre los distintos autores que han trabajado sobre el tema de los sistemas de control y sobre cuáles serían sus elementos; sin embargo, la mayoría coincide en que son:

Entrada

Este concepto se refiere a la cantidad y tipo de información independiente que puede ser controlada y manipulada a fin de que cumpla un objetivo determinado; es decir que, los módulos de entrada analógica convierten cantidades analógicas en números almacenados en variables internas que facilita la obtención de los resultados deseados en el siguiente elemento (salida). Es necesario mantener en mente que, esta conversión se realiza con una precisión o resolución específica (número de bits) y en un intervalo de tiempo específico (período de muestreo). Los módulos de entrada analógica pueden leer voltaje o corriente, parte de la señal de entrada se toma en cuenta también a las señales no deseadas, ya que estas pueden ocasionar perturbaciones externas que, de una u otra forma podrían afectar la variable y, en consecuencia, afectar los datos esperados del

sistema. (Perez & Zamanillo, 2003, pág. 209)

Salida

Esta es la señal que contribuye para que el control que ha sido propuesto se pueda llevar a cabo. Esta variable debe ser elegida de modo tal que, mediante su estudio [permita analizar] si el sistema cumple o no con los objetivos propuestos (Pérez, Pérez, & Pérez, 2007, pág. 7).

Planta

El elemento planta puede estar constituido por cualquier objeto físico que, por sus propias características en cuanto a capacidad de almacenamiento de información, velocidad de procesamiento y ejecución de programas, se destina a realizar operaciones determinadas (Lascorz, 2009, pág. 6), en función de las señales de entrada y salida previamente descritas.

Sensor

También reconocido como bloque de retroalimentación, es el elemento de entrada “que convierte la variable de salida manipulable tomada de una variable física medida, en otra variable manejable tal como desplazamiento, presión, voltaje, estos dispositivos envían señales eléctricas a la salida ya sean análogas o digitales, por lo cual es uno de los dominios tangibles más utilizados en sistemas de medidas y que, en última instancia, se encargará de direccionar la señal de salida hacia el comparador. (Corona, Abarca, & Carreño, 2014, pág. 17)

Señal de referencia

Tal y como ya se dijo sobre el concepto de señal de entrada, esta puede ser manipulada o modificada de acuerdo con parámetros establecidos que propendan a conseguir los resultados deseados. En tal sentido, la señal de referencia es aquella que sirve como valor comparativo hasta que sea equiparado con el “valor de la salida del sistema” (Pérez, Pérez, & Pérez, 2007, pág. 7)

Actuador

Actuador en un término general con el que se define un mecanismo de control, se trata de un elemento físico de potencia, el cual se utilizará para dar una orden a un dispositivo que convierte las señales en señal de entrada a la planta de acuerdo con la señal o acción de control, a fin de que la señal de salida se aproxime a la señal de entrada de referencia (alfaomega, 2016, pág. 78). Así, la variable que ingresa a la planta estará adecuadamente regulada de acuerdo con las mejoras de la variable de salida.

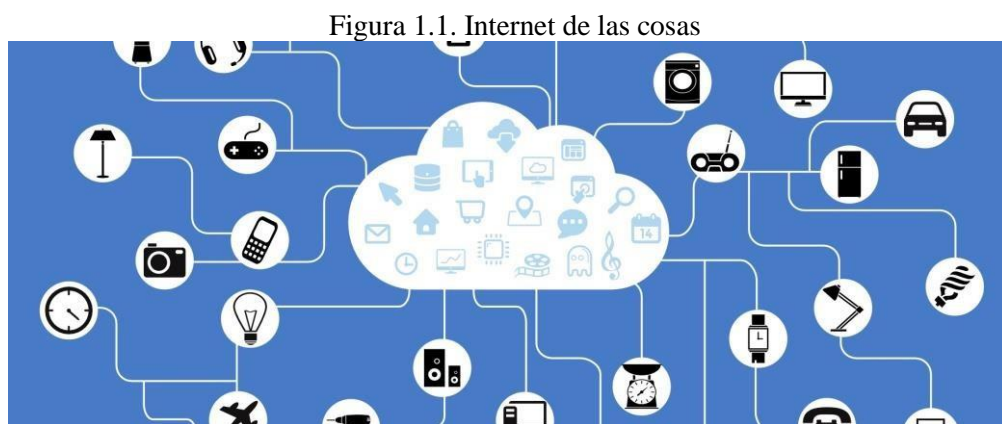
Controlador

El elemento controlador, distinguido por una acción inicial y una respuesta rápida la cual proporciona una variable que se encarga de comparar los valores reales de salida con los valores de la señal de referencia de modo que, “Es posible emplear un solo tipo de controlador para ejecutar distintas dificultades de control” (Oscar & Luis, 2014, pág. 20) que no infiera negativamente en todo el proceso.

1.4.2 Internet de las cosas (IoT)

Internet de las cosas (IoT por sus siglas en inglés) es una arquitectura emergente basada en la Internet global que facilita el intercambio de bienes y servicios entre redes de la cadena de suministro (Weber, 2010, pág. 23). Esta arquitectura, además, debido a sus características y prestaciones, “tiene un impacto importante en la seguridad y privacidad de los actores involucrados” (Salazar & Silvestre, 2017, pág. 7).

En términos sencillos, esta arquitectura permite que, a través de las conexiones que faculta la tecnología, cualesquier objetos pueden mantener una conectividad sistémica a través de la red de internet global con cualesquier otros objetos, tal y como se puede apreciar en la Figura 1.1



Fuente: (Bejerano, 2015)

En cuanto a sus aplicaciones, estas van desde la posibilidad de controlar el funcionamiento de sensores, cámaras de vigilancia, dispositivos móviles, artículos caseros, instrumental médico, vehículos, etc. Es más, si se tratara de objetos que por sus características no cuentan con conectividad intrínseca en su diseño, igualmente pueden ser susceptibles de contar con esta tecnología ya que, con la adaptación a través de sensores y otros mecanismos, se les puede adjudicar las capacidades de interconexión, de modo que los alcances de la tecnología podrían

abarcar aplicaciones como la geolocalización de objetos como textos de libro, llaves, billeteras, la caducidad de víveres comestibles o medicamentos, el daño de redes de electricidad o agua potable, etc. Lo único necesario para esto sería, como se dijo, garantizar su capacidad de interconexión, lo cual se puede realizar a través de identificación por radiofrecuencia (RFID por sus siglas en inglés) o redes de sensores (WSN) (João, Serralvo, & De Souza, 2019)

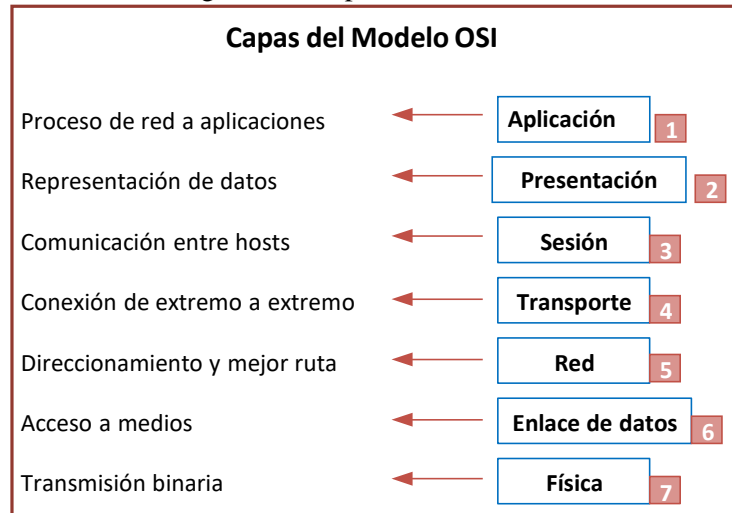
Una de las ventajas más destacables de esta arquitectura corresponde a que no se necesita de gran cantidad de operadores humanos que deban encargarse del proceso de control, sino que, gracias a procesos de automatización, esta tecnología permite monitorear las diferentes variables de procesos de forma telemática y responder a emergencias según ameriten las alertas emitidas también de forma automática, las cuales, a su vez, podrían ser también programadas para que, al presentarse en un futuro, sean los mismos sistemas los que se encarguen de solucionar los desperfectos (Mongay Batalla, Mastorakis, Mavromoustakis, & Pallis, 2017, pág. 27).

1.4.3 Estructura de una red

Al hablar de interconectividad se hace referencia a la capacidad de intercomunicación entre al menos dos objetos que comparten un canal de comunicación tanto de entrada como de salida de información. En tal virtud, para garantizar que dicha información fluya de forma adecuada e ininterrumpida en ambas direcciones, es necesario contar con el hardware de red, software de red y software de aplicaciones como los tres componentes imprescindibles para formar una red de comunicación. A la estructura conformada por estos tres niveles de componentes se la llama estructura de red.

En cuanto a los modelos para las comunicaciones en red, el principal es el modelo de referencia de Interconexión de Sistemas Abiertos (OSI, por sus siglas en inglés), el cual “es un marco que se puede utilizar para comprender cómo viaja la información a través de una red” (Rigotti, 2022, pág. 5) y que está, a su vez, conformado por “siete capas, cada una de las cuales ilustra una función de red específica” (Rigotti, pág. 5). Según el mismo Rigotti, tal como se muestra en la figura 1.2.

Figura 1.2. Capas del modelo OSI



Fuente: Elaboración propia, con base en los aporte de CISCO (2010).

Capa 1. Capa de aplicación. Es la capa de interacción directa con el usuario y no proporciona los servicios a ninguna otra de las capas del modelo.

Capa 2. Capa de presentación. Esta garantiza la lectura entre capas de aplicación de diferentes sistemas. En esta capa, incluso, se realiza las traducciones desde diferentes lenguajes a uno que sea compartido por ambos sistemas informáticos.

Capa 3. Capa de sesión. Esta presta servicio a la capa de presentación, “sincroniza el diálogo entre las capas de presentación de los dos hosts y administra su intercambio de datos [y] regula la sesión” (Rigotti, pág. 6).

Capa 4. Capa de transporte. Esta capa y las descendentes ejecutan procesos de conmutación de datos. En esta, específicamente, los datos provenientes del host emisor son divididos para volver a ensamblarlos de modo que puedan llegar al host receptor.

Capa 5. Capa de red. Esta capa se encarga de garantizar la interconexión entre dos o más redes al seleccionar las rutas más adecuadas.

Capa 6. Capa de enlace de datos. Esta capa busca garantizar la confiabilidad de la conmutación de datos por medio de enlaces físicos, ocupándose así del “direccionamiento físico, la topología de red, el acceso a la red, la notificación de errores, entrega ordenada de tramas y control de flujo” (Rigotti, pág. 7).

Capa 7. Capa física. Esta define las especificaciones eléctricas, mecánicas, de procedimiento y funcionales para activar, mantener y desactivar el enlace físico entre sistemas finales. Las características tales como niveles de voltaje, temporización de cambios de voltaje, velocidad de datos físicos, distancias de transmisión máximas, conectores físicos y otros atributos similares son definidos por las especificaciones de la capa física (Rigotti, pág. 7).

1.4.4 Protocolo de comunicación en redes IoT

Como se mencionó anteriormente, para que la arquitectura del Internet de las Cosas pueda ser establecida, es necesario que los objetos cumplan con ciertas características que garanticen su interconectividad. Esto se vuelve posible gracias a los protocolos comunicacionales, los cuales deben ser dispuestos de modo que superen y satisfagan las necesidades de acuerdo a factores como la naturaleza y la distancia geográfica existente entre dispositivos, la compatibilidad de su tecnología, los posibles desencadenantes de fallo, etc.

Entre los protocolos de comunicación más conocidos constan la tecnología Bluetooth, Wifi y Zigbee. Sin embargo, existen otras que, aunque menos conocidas por la población en común, son muy utilizadas y que, según Ignacio Porro Sáez (2019), pueden clasificarse de la siguiente manera:

1.4.4.1 Clasificación de los protocolos de comunicación

Protocolos IoT de entorno doméstico

Figura 1.3. Protocolos de comunicación en entorno doméstico



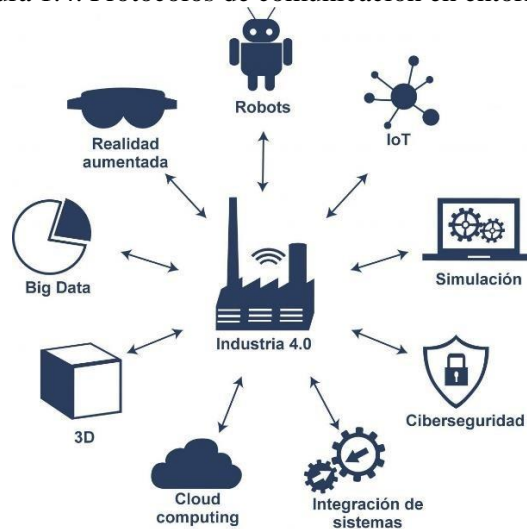
Fuente: Porro, Ignacio, 2019.

Como su nombre sugiere, este tipo de protocolos son utilizados para garantizar la interconectividad de dispositivos generalmente usados a nivel doméstico. Entre estos están:

- AllJoyn
- HomePlug y HomeGrid
- MFi (Made For iPhone/iPod/iPad)
- OCF (Open Connectivity Foundation)
- Thread (network protocol) (Porro Sáez, 2019)

Protocolos IoT de entorno industrial

Figura 1.4. Protocolos de comunicación en entorno industrial



Fuente: Porro, Ignacio, 2019.

Como igualmente se sugiere en su nombre, este tipo de protocolos son utilizados para garantizar la interconectividad de dispositivos generalmente usados a nivel industrial. Entre estos están:

- AMQP (Advanced Message Queuing Protocol)
- CoAP (Constrained Application Protocol)
- DDS (Data Distribution Service)
- HTTP (REST/JSON) (Hypertext Transfer Protocol)
- MQTT (Message Queuing Telemetry Transport)
- OPC UA (Unified Architecture)
- XMPP (Extensible Messaging and Presence Protocol) (Porro Sáez, 2019).

1.4.5 Red WSN

Una red de sensores inalámbricos (WSN por sus siglas en inglés) está conformada por muchos “dispositivos distribuidos espacialmente, que utilizan sensores para controlar diversas condiciones en distintos puntos, entre ellas la temperatura, el sonido, la vibración, la presión y movimiento o los contaminantes” (Aakvaag & Frey, 2006, pág. 39).

En cuanto a la conectividad de los dispositivos (sensores) que conforman la red WSN, la información recolectada es enviada a un nodo central a través de ondas electromagnéticas.

En cuanto a los elementos de una red WSN se menciona al menos los siguientes:

1.4.5.1 Sistema de adquisición de datos

Este sistema está constituido principalmente por los sensores, que son los dispositivos que recogen la información de acuerdo con sus propias características y diferencias según la naturaleza y tecnología de diseño, transformándola a electricidad para enviarla a los siguientes elementos: las motas. Este proceso, básicamente, consta de tres etapas:

1. Conversión de la magnitud a una señal eléctrica.
2. Adaptación de la señal eléctrica para su lectura digital.
3. Sistema hardware de adquisición de datos, generalmente a través de un computador (Mora, Gil, Signes, & Colom, 2016, pág. 1).

1.4.5.2 Motas

Los dispositivos conocidos como motas son aquellos donde se procesa la información recibida de parte de los sensores y desde los cuales se envía dicha información a las estaciones base.

De acuerdo con Fernández Barcell (2020), las motas estarían compuestas por: baterías, CPU, memoria SSD, memoria para datos, placa de sensores, radio de comunicación inter-mota y ADC (conversor analógico-digital).

1.4.5.3 Gateway

Para Piñeros y González, (2004, pág. 50), un Gateway también conocido como pasarela, es un “dispositivo activo que trabaja cubriendo hasta el nivel 7 (aplicación) del modelo OSI y que permite transferencia de datos y la interconexión de redes que tienen funciones similares pero

implantaciones diferentes, como IP y X.25.” En consecuencia, los Gateway son routers, dispositivos donde los paquetes de información recibidos experimentarán una reestructuración de formato con el fin de que dicha información llegue a ser comprensible por la red destinataria.

1.4.5.4 Estación base

Este elemento se compone de un PC o un sistema embebido donde se recolectan los datos.

Por otro lado, entre los parámetros que generalmente caracterizan una red WSN están: precisión, seguridad, robustez, interconectividad, compatibilidad, costo-beneficio, gasto de recursos, facilidad de instalación, etc. (Fernández Barcell, 2020).

1.4.6 Nube

Si bien es cierto que apenas desde 1990 el término nube empezó a popularizarse hasta alcanzar su lugar en el lenguaje común para hacer referencia al “lugar” donde se encontraría alojada gran cantidad de información computacional y a la cual es posible acceder desde cualquier lugar en el mundo, siempre y cuando se cuente con un dispositivo computacional y conexión a internet, los orígenes de esta tecnología parecen remontarse incluso hasta 1960, época en la cual JCR Licklider —uno de los miembros del equipo desarrollador de ARPANET— y John McCarthy —a quien se le acredita la acuñación del término “inteligencia artificial”— dieran los primeros esbozos de la que concebían como Red Computacional Intergaláctica, y desde ya vaticinaban que la información estaría alojada en redes compuestas por dispositivos alojados a lo largo y ancho del mundo desde los cuales las personas colaborarían, ya que el servicio de computación habría de llegar a considerarse como un servicio público.

Es así que, desde esos vaticinios, la arquitectura de nube ha llegado a ser lo que es hoy, una interconexión de millones de dispositivos en donde la información puede alojarse de forma particionada a tal punto que puede accederse a ella desde cualquier lugar y momento del mundo sin que se dependa de uno solo o pocos dispositivos, sino que todos conforman una base universal de datos (Sturm, 2012).

Hoy en día, el servicio de nube más conocido es el proporcionado por Google, Google Drive; sin embargo, existen muchos otros, entre los cuales destacan Dropbox, OneDrive, SkyDrive, etc.

1.4.7 GPS

A pesar de que esta tecnología fue originalmente diseñada con fines exclusivamente militares, debido a sus características y multiplicidad de aplicaciones, el Sistema de Posicionamiento Global, o GPS por sus siglas en inglés, ha llegado a ser ampliamente utilizado, aunque con varias limitaciones, en un contexto plenamente civil. La tecnología GPS, en última instancia, faculta que uno o más elementos previamente dispuestos establezcan interconexiones que permitan el reconocimiento de su geolocalización. Para dar cumplimiento a esta tarea, el sistema GPS debe contar con los siguientes componentes:

1. Espacial. Formado por 24 satélites situados a unos 20200 km de la Tierra y que pasan por el mismo lugar cada 12 horas.
2. Control. Formado por una serie de estaciones de observación cercanas al ecuador encargados de controlar la posición orbital de los satélites y calibrar y sincronizar los relojes
3. Receptor GPS. Entre las actividades la navegación marítima o terrestre, excursionismo, topografía, control de maquinaria, etc. (Quispe, Castillo, Tejada, & García, 2022, pág. 2).

1.4.8 Geolocalización

El término geolocalización se refiere a la posibilidad de conocer la ubicación de cualesquier objetos en “un punto cualquiera en un espacio de coordenadas (x, y, z), partiendo del cálculo de las distancias del punto a un mínimo de tres satélites cuya localización es conocida.” (Pozo-Ruz, y otros, 2020, pág. 2)

1.4.9 Sistemas embebidos

Los sistemas embebidos son sistemas computacionales diseñados para responder a necesidades computacionales en tiempo real y, aunque prácticamente se esté haciendo referencia a las conocidas generalmente como computadoras, la diferencia radica en que, mientras estas cubren “un amplio rango de necesidades, los sistemas embebidos, [en cambio], se diseñan para cubrir necesidades específicas” (UNED, 2017, pág. 3).

Entre sus características a destacar, se puede establecer que estos sistemas “permiten conocer y dotar de inteligencia a las interacciones que se producen con el entorno” (Mora, Gil, Signes, & Colom, 2016, pág. 1) y que, en cuanto su programación, esta puede darse de forma directa “en el

lenguaje ensamblador del microcontrolador o microprocesador incorporado sobre el mismo, o también, utilizando los compiladores específicos, pueden utilizarse lenguajes como C o C++” (UNED, 2017, pág. 3)

2 CAPÍTULO 2

En este apartado se describirán detalladamente los elementos constitutivos del sistema GPS a implementar, tanto en la parte de hardware como también en el software.

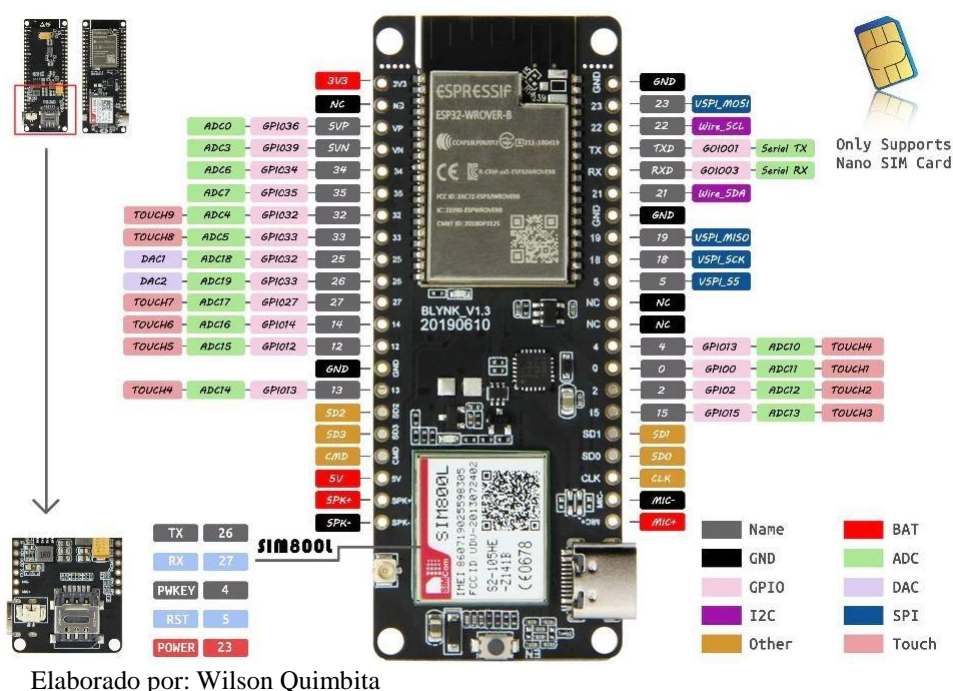
2.1 Hardware

Con respecto al hardware se tienen los siguientes elementos:

2.1.1 TTGO T-Call ESP32 SIM800L

Este dispositivo condensa los microchips ESP32 y SIM800L como se muestra en la figura 2.1. Lo cual permite enlazar a internet al ESP32 empleando una tarjeta sim con plan de datos o a su vez, realizar comunicación por medio de mensajes SMS y también llamadas debido a la integración del módulo SIM800L, existiendo la limitación de ser compatible con redes 2G o GSM. (Bernal & Ladino, 2020, pág. 30) Las especificaciones se muestran en el anexo 3.

Figura 2.1. ESP32 SIM Card SIM800L T-Call V1.3



2.1.2 SMS chip telefónico

Las siglas en inglés de SIM proviene de la palabra módulo de identificación de abonado. Consiste en un chip embebido en una tarjeta plástica, la cual, almacena de forma segura la clave de servicio

del beneficiario con el fin de identificarse en la red móvil telefónica. En general la capacidad de almacenaje de una SIM varía desde 2kb hasta 1Gb, no obstante, las más utilizadas están entre 16 y 32 kb (Shatilin, 2016)

2.1.3 Módulo GPS NEO 6M

El módulo GPS NEO 6M es una buena opción en precisión - costo beneficio. Su tamaño reducido facilita la portabilidad en proyectos. La comunicación se realiza mediante el puerto serial UART (*Universal Asynchronous Receiver - Transmitter*), con lo cual, se puede utilizar junto a varios dispositivos como son: Arduino, RaspberryPi y computadora. Las magnitudes que mide este dispositivo son: latitud, longitud, velocidad, altitud. En la tabla 2.1 se describe en detalle las características técnicas del dispositivo.

Tabla 2.1. Características del módulo GY-NEO06MV2.

Datos	Especificaciones
Coporta estándares	WAAS/EGNOS/MSAS/GAGAN
Frecuencia de actualización	5Hz
velocidad de desplazamiento máxima	500m/seg
Antena incorporada	18.2 x 18.2 x 4.0 mm
Rango de temperatura	-40° to 85° C
Tamaño	30mm x20mm x 11.4mm
Corriente de funcionamiento	Alrededor de 40mA.

Elaborado por: Wilson Quimbita.

2.1.4 Batería

Con el objetivo de obtener un correcto dimensionamiento de la energía consumida por el equipo en la tabla 2.2 se detalla el consumo total de energía estimado.

Tabla 2.2. Energía consumida mA.

Elemento	Consumo mínimo mA	Consumo máximo mA
SIM800L	10	50
Esp32-wrover	50	50
NEO-6M-0-001		40
Consumo		140
Consumo total	+ 5%	147

Elaborado por: Wilson Quimbita.

Ahora bien, por los datos proporcionados en la tabla 2.2, si se desea que el equipo funcione durante

17 días, es necesario transformar los meses a horas, tal como se observa en la ecuación 2.1.

Ecuación 2.1. Conversión de jornada diaria a horaria

$$\begin{aligned} \text{horas} &= \text{horas de un día} \times \text{días de funcionamiento} && \text{Ec (2.1)} \\ \text{horas} &= 24 \times 17 = 408 \end{aligned}$$

Elaborado por: Wilson Quimbita.

Con el fin de encontrar el consumo estimado de amperaje por hora, se multiplica la cantidad de horas que se desea el dispositivo esté encendido por el consumo de energía en mA, tal como indica la ecuación 2.2.

Ecuación 2.2 Cálculo de consumo de amperaje

$$\begin{aligned} \text{consumo}_{mAh} &= \text{consumo en horas} \times \text{energía consumida en miliamperios} && \text{Ec. (2.2)} \\ \text{consumo}_{mAh} &= 408 \times 147 = 59976 \end{aligned}$$

Elaborado por: Wilson Quimbita.

Una vez encontrada la cantidad de miliamperios hora que debe permanecer encendido el equipo se seleccionaron seis baterías modelo 18560 las cuales proporcionan 3,6v y 10000mAh, con lo cual se cumple con el dimensionamiento realizado.

2.2 Software

2.2.1 Arduino IDE

Para iniciar a programar la placa TTGO T-Call ESP32 SIM800L es necesario utilizar un IDE que consiste en un grupo de herramientas de software libre y disponible para descargas en la web, las cuales posibilitan a los desarrolladores grabar el código necesario para que el microcontrolador funcione como necesitamos. Este IDE facilita la escritura, depuración, edición y grabación de los programas en el microcontrolador de una manera fácil y a la cual puede acceder cualquier persona que busque producir sus proyectos interactivos (ARDUINO, 2023).

2.2.2 MQTT Corredor (Broker)

Eclipse Mosquitto es un broker de mensajes en código abierto que implementa varias versiones del protocolo MQTT. Mosquitto al ser liviano, es apropiado usarlo en varios dispositivos como computadores de escritorio hasta servidores. Este protocolo MQTT suministra una forma ligera de

enviar mensajes empleando un modelo de publicación/suscripción. Por tal motivo lo convierte en una opción adecuada en la mensajería de IoT (internet de las cosas), como son los sensores de baja consumo de potencia o dispositivos móviles como teléfonos, computadoras o microcontroladores. Mosquitto además proporciona una librería en C para implementar clientes MQTT basado en línea de comandos `mosquitto_pub` y `mosquitto_sub` (Mosquitto, 2023).

2.2.3 Servidor en la Nube basado en Linux

DigitalOcean es una nube de creación de aplicaciones para programadores. Esto permite simplificar la creación de aplicaciones para programadores que trabajan de forma independiente, en proyectos medianos y pequeños. Las soluciones que brinda de infraestructura y plataforma como servicio está entre las más utilizadas ya que armoniza potencia y simplicidad con el uso de código abierto para que el desarrollo del software sea más globalizado. En nuestro caso se instaló un servidor con sistema operativo Ubuntu 20.0, por el cual pasarán todos los datos de entrada y salida del sistema GPS (DigitalOcean, 2023) .

2.2.4 Servidor web Flask

Flask es un microframework basado en Python que permite crear aplicaciones web de toda índole de una manera rápida y efectiva. Habitualmente se enuncia que Django es mejor framework para proyectos de cierta extensión, en tanto que, Flask se utiliza para prototipos y pequeñas aplicaciones, tal es el caso del sistema GPS, el cual no es proyecto sumamente extenso y calza adecuadamente en este microframework (FLASK, 2023).

2.2.5 Mapa de geolocalización HERE

El proveedor del mapa, HERE por su nombre en inglés, es un servicio en la nube, en el cual los datos y servicios de localización se encuentran almacenados dentro de servidores remotos, para que los usuarios puedan hacer consultas desde el dispositivo cliente. HERE provee mapas de 200 países en 46 idiomas diferentes con sus respectivas coordenadas que captura contenido de localización tales como: calles, carreteras, edificios, parques, entre otros. Dichos mapas pueden ser usados desde el teléfono móvil o desde la página web (HERE, 2023).

2.2.6 Aplicación de usuario basada en HTML y JavaScript

2.2.6.1 Java Script

A decir de Menéndez-Barzanallana, decano de la Universidad de Murcia, “JavaScript es un lenguaje de programación de scripts (secuencia de comandos) orientado a objetos” (2020, pág. 3). Entiéndase, además, que, al hablar de lenguaje de programación, se está haciendo referencia al lenguaje a través del cual los desarrolladores pueden escribir y o modificar el código fuente que se destinará a ser analizado por un computador. Asimismo, el código fuente que ha sido desarrollado a través del lenguaje de programación, puede ser usado de tres maneras:

1. Lenguaje compilado
2. Lenguaje precompilado
3. Lenguaje interpretado (Menéndez-Barzanallana, 2020, pág. 3)

2.2.6.2 HTML

El Lenguaje de marcado hipertexto, HTML por sus siglas en inglés, que significan: *HiperText Markup Language*, corresponde al lenguaje que se utiliza para construir y desplegar una página web con su contenido. Gracias al uso del lenguaje HTML, los programadores no necesitan acceder desde programas o software especiales para la creación y desarrollo de páginas web, pudiendo, por tanto, desarrollar dichas páginas desde el mismo lenguaje desarrollado en forma de texto (Ferrer, García, & García, 2007).

3 CAPÍTULO 3

3.1 Implementación del sistema

3.1.1 Datos Móviles

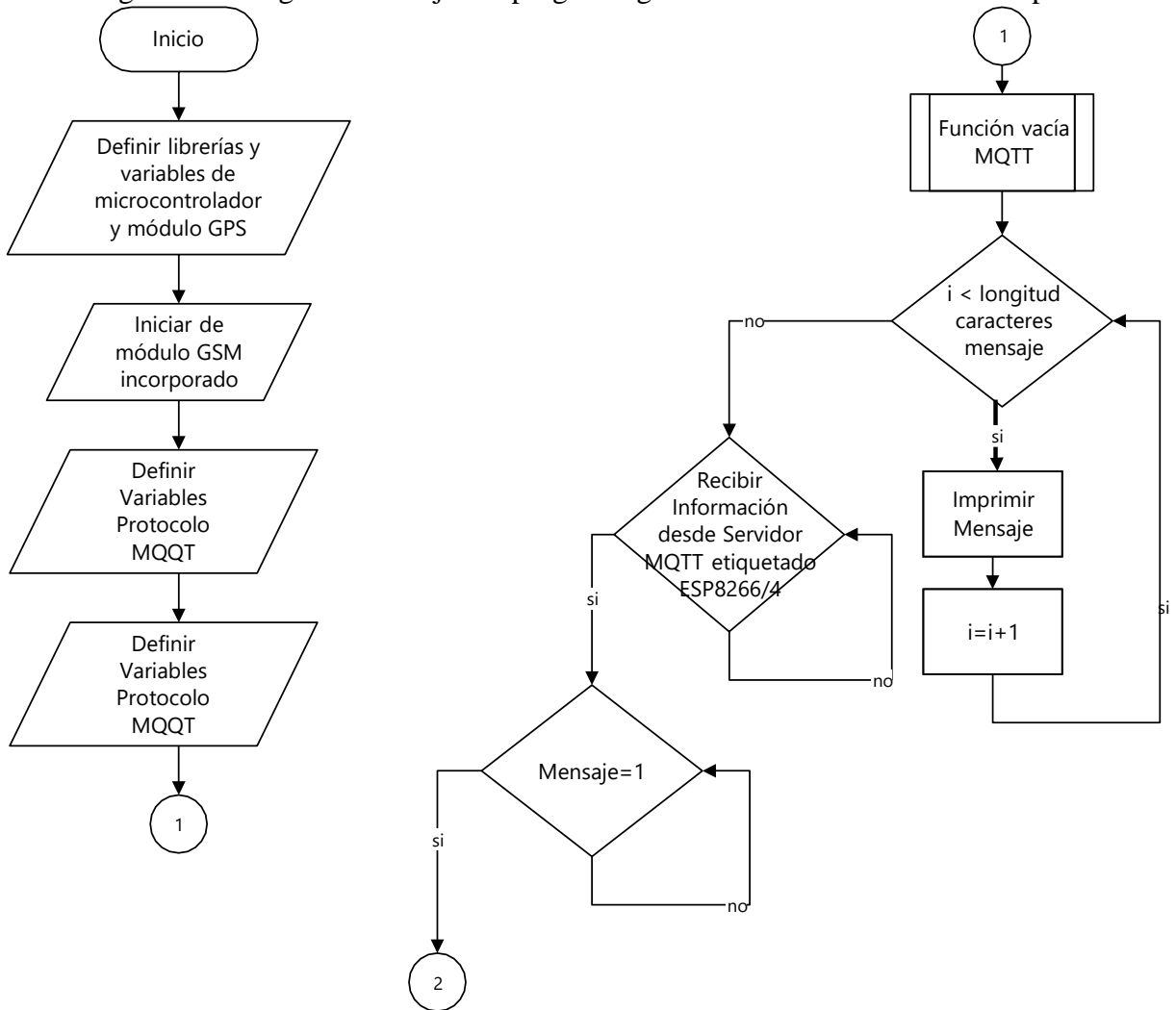
Con la finalidad de conectar la tarjeta SIM a internet, se debe conocer detalladamente la APN (access point network) del proveedor del plan telefónico (CNT-Ecuador), un nombre de usuario y una contraseña. Si se busca la configuración en la página web “apnsettings.org” la siguiente información: “GPRS APN CNT Ecuador”, se conseguirá la información necesaria. La cual viene dada por la APN “internet3gsp.alegro.net.ec”, el usuario y la contraseña sin definir por lo cual se deja en blanco.

3.1.2 Librerías Arduino IDE

Para la compilación del programa y su posterior grabación en el microcontrolador, es necesario instalar varias librerías en el IDE de Arduino como son: TinyGSM, TinyGPS, PubSubClient y Async-mqtt-client. También se debe tener instalado el complemento TTGO T1 el cual permite la comunicación entre el dispositivo embebido y el IDE de programación con el fin de poder grabar el programa desarrollado (Libraries, 2023). En las figuras 3.1, 3.2, 3.3 y 3.4 se aprecia un resumen organizado del algoritmo utilizado en la programación del microcontrolador, que consta de las funciones necesarias para cumplir con el objetivo de enviar las coordenadas de posicionamiento hacia el servicio en la nube.

En la figura 3.1 se observa la inicialización de las librerías y las variables necesarias para el funcionamiento del Módulo GSM, GPS y poder enviar y recibir datos hacia la nube mediante el uso del protocolo MQTT. Además, se muestra la validación de caracteres alfanuméricos y la verificación de la longitud de mensajes que se van a enviar.

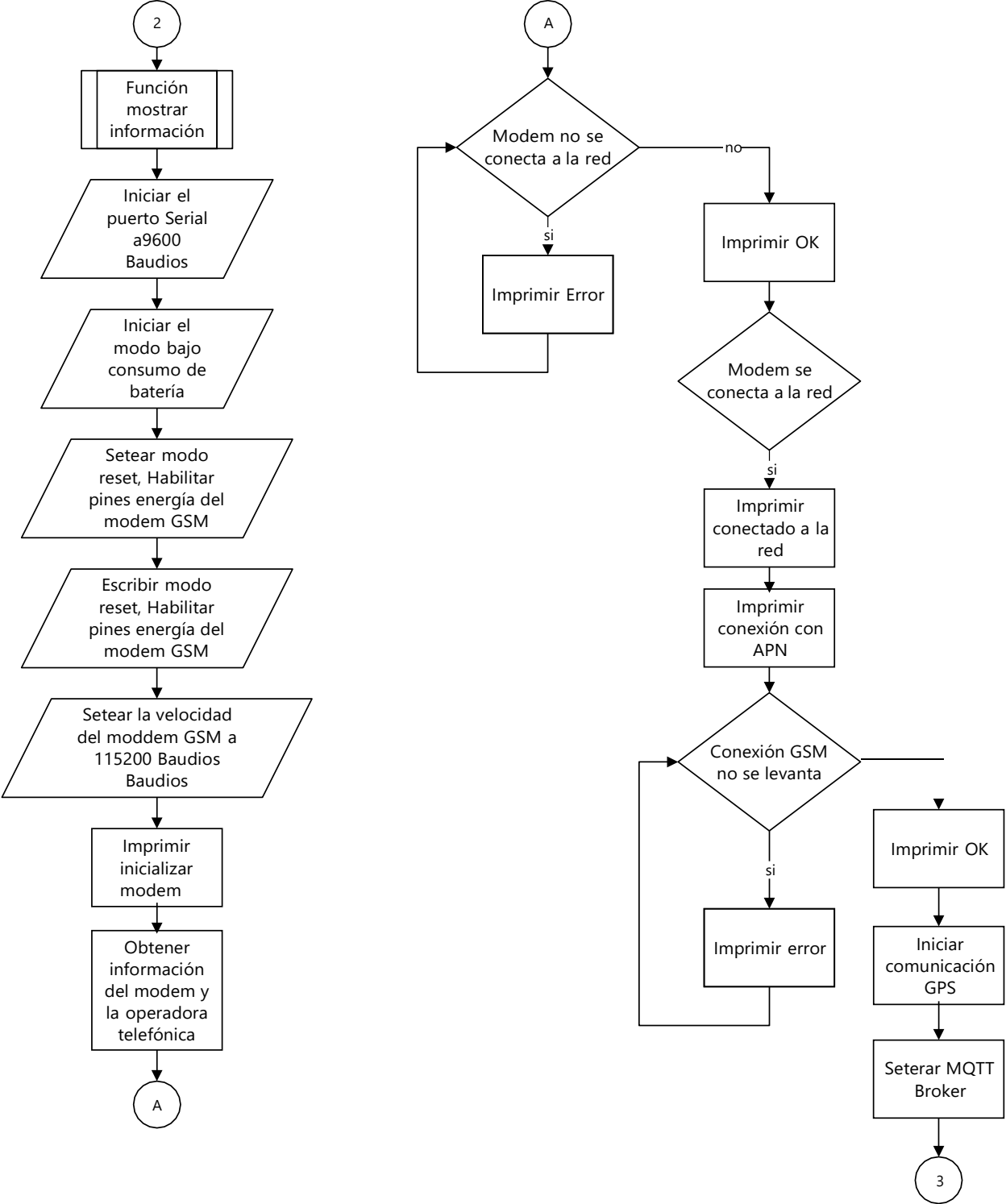
Figura 3.1. Diagrama de flujo del programa grabado en el microcontrolador parte1.



Elaborado por: Wilson Quimbita.

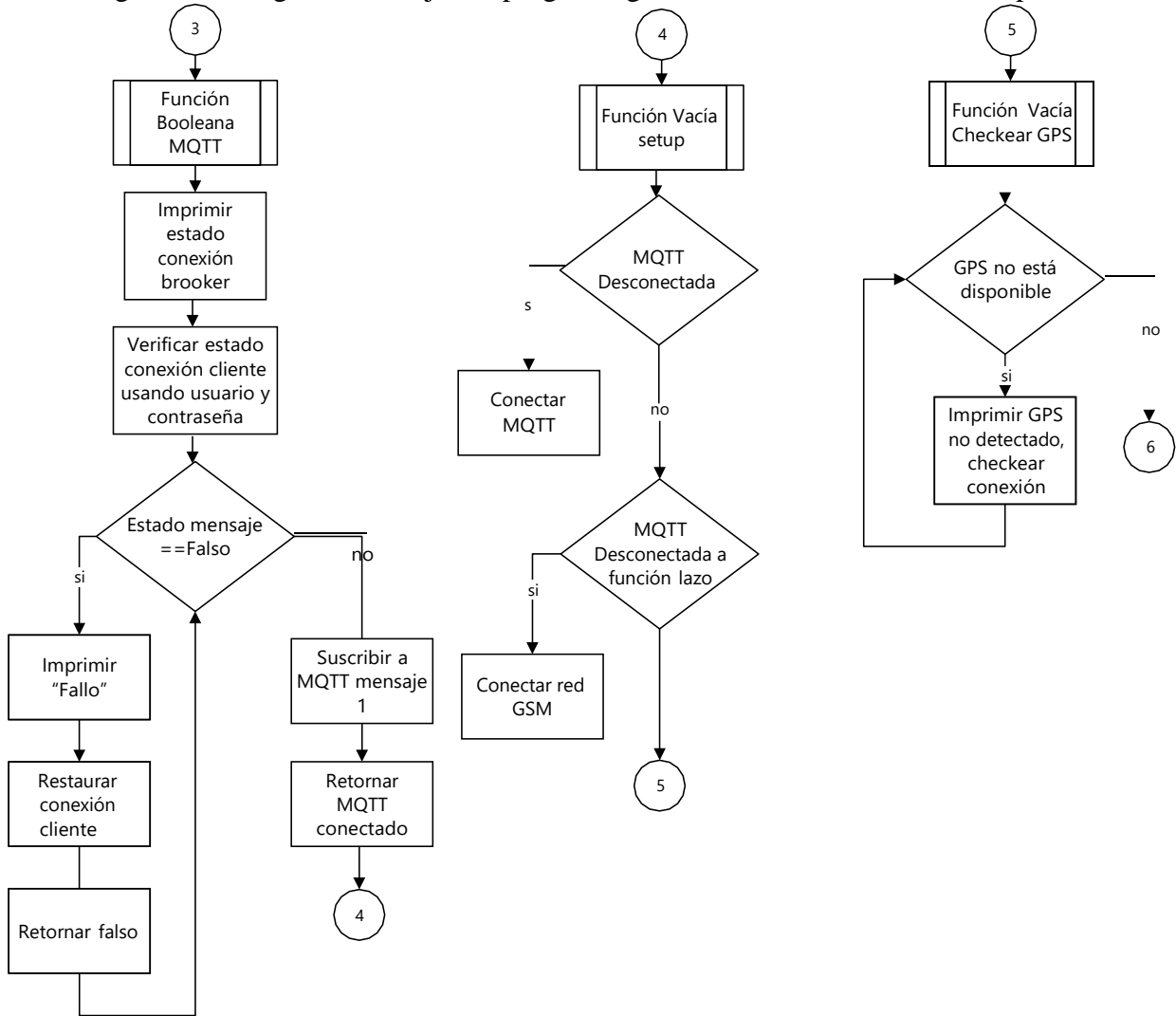
En la figura 3.2 se muestra el proceso en el cual se imprime y verifica la información de inicialización del: puerto serial a una velocidad de 9600 baudios, el módulo GSM a 11500 baudios, el modo ahorro de energía del TTGO T1, la conexión del modem GSM de la operadora telefónica hacia la nube por medio del protocolo MQTT.

Figura 3.2. Diagrama de flujo del programa grabado en el microcontrolador parte 2.



Elaborado por: Wilson Quimbita.

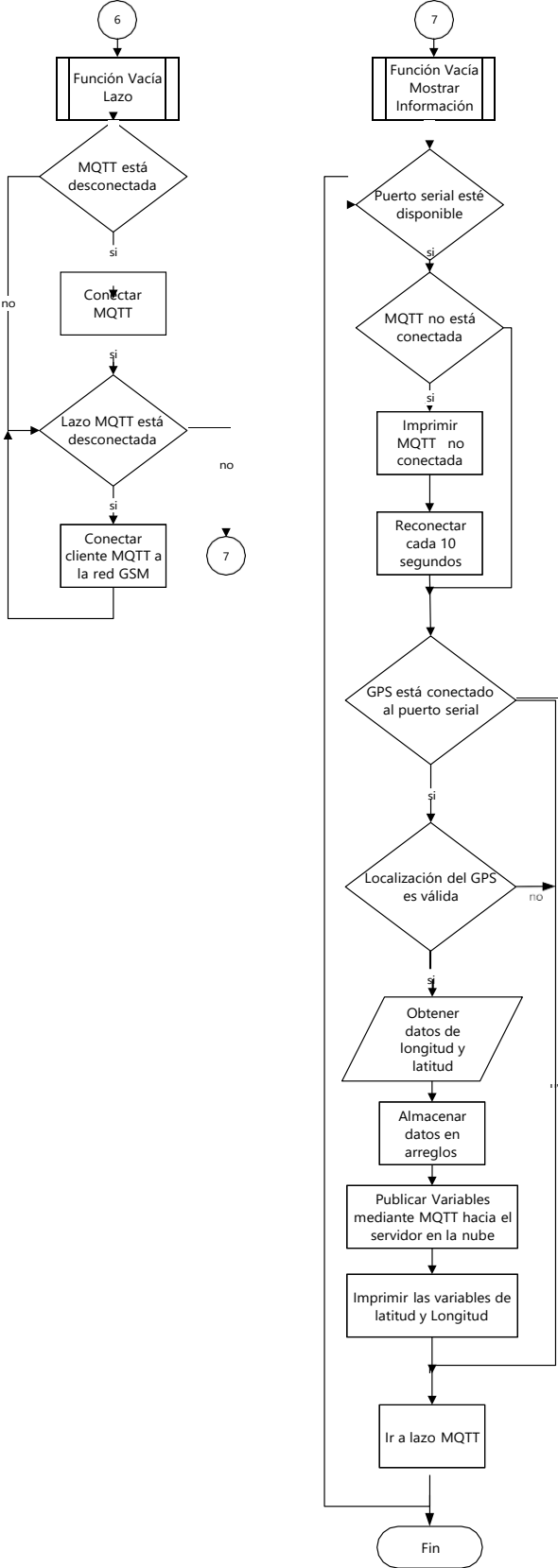
Figura 3.3. Diagrama de flujo del programa grabado en el microcontrolador parte 3.



Elaborado por: Wilson Quimbita.

En la figura 3.3 se muestra la parte del diagrama de flujo en donde se comprueba el estado de conexión del broker MQTT a través de una función booleana y se procede a verificar el usuario y contraseña del cliente. Una vez verificada las credenciales del cliente MQTT, se procede a realizar la conexión con el servicio en la nube utilizando la red GSM. Al final, en esta sección se procede a verificar el estado del módulo GPS, para intentar una conexión con este dispositivo.

Figura 3.4. Diagrama de flujo del programa grabado en el microcontrolador parte 4.



Elaborado por: Wilson Quimbita.

En la figura 3.4 me muestra la última parte del programa en la que de no realizarse la conexión con el dispositivo GPS se vuelve a probar conexión de la red telefónica GSM con la nube mediante el protocolo MQTT. Cuando existe conexión entre el módulo GSM y el servicio de la nube, se comprueba la conexión del módulo GPS por medio del puerto serial hacia el microcontrolador. De ser válida la conexión se realiza la petición de datos de latitud y longitud del módulo para publicarlos en la nube mediante el protocolo MQTT.

Este proceso se repite varias veces hasta terminar la conexión apagando el dispositivo o si existe un fallo en alguno de los módulos que permiten en conjunto la transmisión y recepción de datos. De suscitarse dicho fallo el programa intentará establecer una nueva conexión desde el principio, verificando la correcta inicialización de cada módulo.

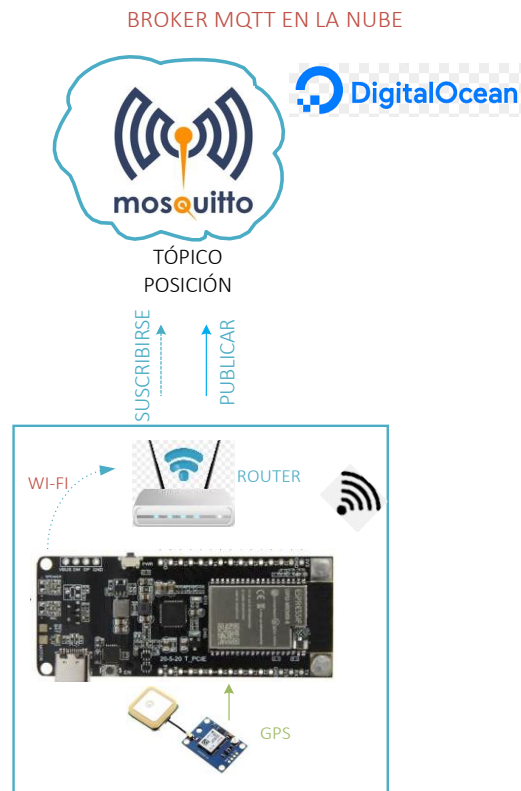
3.1.3 Publicación/Subscripción nube

Se intercambiarán los datos valiéndonos del protocolo MQTT (Message Queue Server Telemetry Transport) por sus siglas en inglés. Consiste en un sistema liviano de publicación y suscripción, en el cual, se puede publicar y recibir mensajes como cliente. Un corredor(broker) de MQTT es el encargado de recibir y filtrar los mensajes, además decidir quién se encuentra interesado en cada mensaje para posteriormente publicarlos a todos los clientes suscritos.

Existen en la web varios corredores que se pueden utilizar. Para este proyecto se utilizó Mosquito MQTT, el cual es de fácil instalación, configuración y uso. Este corredor utilizado en la nube (servicio proporcionado por Digital Ocean) permite tener múltiples dispositivos IoT que se comunican entre sí incluso si se encuentran en diferentes redes en todo el mundo.

El microcontrolador TTGO T-Call ESP32 SIM800L debe estar conectado a internet mediante un enrutador para conectarse al corredor, lo cual se logra utilizando los datos móviles de la operadora de telefonía CNT. Esto permite publicar y suscribir datos desde y hacia la nube como se observa en la figura 3.5.

Figura 3.5. Comunicación entre microcontrolador y el servidor en la nube.



3.1.4 Servidor en la nube Digital Ocean

Con el objetivo de ejecutar programas en la nube como: Mosquitto, Flask, here JS y HTML, se debe utilizar un servicio de hospedaje en esta, el cual debe conceder acceso a línea de comandos y permitir instalar cualquier software que se requiera. En este sentido Digital Ocean brinda un servidor Ubuntu, el que se puede administrar a través de una línea de comandos.

Con el objetivo de crear una nueva máquina virtual se procede de la siguiente forma:

- En la esquina superior derecha se presiona el botón “Create” y se selecciona la opción “Drops” (Digital Ocean nombra Droplets a sus máquinas virtuales).
- Se selecciona el sistema operativo Ubuntu 20,04 (LTS)x64.
- En el menú de recursos de máquina virtual, se elige el plan más económico por 5 usd al mes. Con el fin de ejecutar MQTT Broker, Flask, Java Script y HTML este plan económico nos sirve sin inconvenientes.

- Al final se elige un nombre de host para identificar con qué máquina virtual se está trabajando. En este caso el Droplet es “mqtt-server”. Se presiona el botón verde para crear el drop y finalizar este proceso.

Con el objetivo de acceder por medio de consola de máquina virtual Ubuntu se debe seguir los siguientes pasos:

- Abrir el servidor creado previamente llamado “mqtt-server”
- Previamente se debe tener programado el microcontrolador con la ayuda de Arduino IDE.

3.1.5 Mqtt Broker utilizado para publicar mensajes en el módulo TTGO T-Call ESP32 SIM800L

El siguiente esquema realiza la suscripción desde el módulo TTGO a un tema MQTT en el servicio de la nube para recibir mensajes. En el IDE de Arduino se debe tener insertada las credenciales de red GSM, así también los detalles del broker MQTT (la dirección IP de la máquina virtual y el nombre de usuario y la contraseña del broker) (RandomNerdTutorials.com, 2023).

Con el propósito de comprobar si el módulo está recibiendo mensajes MQTT, en la consola de Digital Ocean empieza a publicar diferentes mensajes, por ejemplo: “Hello world!”, “gps”, etc como se aprecia en la figura 3.6 ventana 1. El módulo debería recibir cada mensaje e imprimirlo en el Monitor Serial, como se muestra en la figura 3.6 ventana 2.

Figura 3.6. Prueba de comunicación con el protocolo MQTT.

Ventana 1

```

pi@raspberrypi:~$ mosquitto_pub -d -t testTopic -m "Hello world!"
Client mosqpub/868-raspberrypi sending CONNECT
Client mosqpub/868-raspberrypi received CONNACK
Client mosqpub/868-raspberrypi sending PUBLISH (d0, q0, r0, m1, 'testTopic', ... (12 bytes))
Client mosqpub/868-raspberrypi sending DISCONNECT

```

Ventana 2

```

pi@raspberrypi:~$ mosquitto_sub -d -t testTopic
Client mosqsub/867-raspberrypi sending CONNECT
Client mosqsub/867-raspberrypi received CONNACK
Client mosqsub/867-raspberrypi sending SUBSCRIBE (Mid: 1, Topic: testTopic, QoS: 0)
Client mosqsub/867-raspberrypi received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/867-raspberrypi received PUBLISH (d0, q0, r0, m0, 'testTopic', ... (12 bytes))
Hello world!

```

Elaborado por: Wilson Quimbata.

3.1.6 Servidor Web Flask

Como complemento al servicio MQTT, los mensajes publicados por este servicio pasan a través del servidor web flask, el cual administra websockets tanto de entrada como de salida, con el

propósito de controlar la información que recibe desde Mosquitto. El servidor controla también el flujo de información que llega desde los aplicativos: Here, JS, y HTML, con la finalidad de publicar mensajes de regreso hacia el broker MQTT. En la figura 3.7 se aprecia el inicio de la ejecución del servidor web flask en una terminal por consola.

Figura 3.7. Servidor Web Flask en ejecución

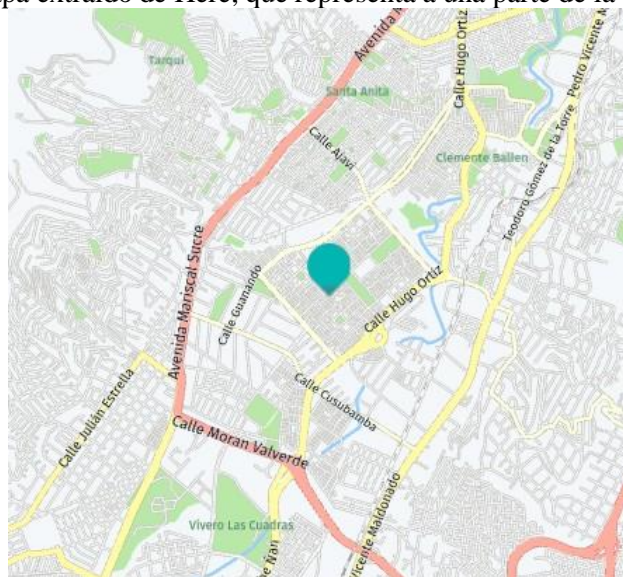
```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production
  deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:8181/ (Press CTRL+C to quit)
* Restarting with stat
WebSocket transport not available. Install eventlet or gevent and gevent-
websocket for improved performance.
Connected with result code 0
* Debugger is active!
* Debugger PIN: 197-343-719
200.125.231.222 - - [27/Aug/2022 01:22:02] "GET / HTTP/1.1" 200 -
200.125.231.222 - - [27/Aug/2022 01:22:03] "GET /favicon.ico HTTP/1.1"
404 -
200.125.231.222 - - [27/Aug/2022 01:22:03] "GET /socket.io/?EIO=3&tran
sport=polling&t=1661563323571-0 HTTP/1.1" 200 -
200.125.231.222 - - [27/Aug/2022 01:22:03] "POST /socket.io/?EIO=3&tra
nsport=polling&t=1661563323784-1&sid=e56b0e58734e4a1db4fde69a9b03d50e
HTTP/1.1" 200 -
received json data here: {'data': "I'm connected!"}
200.125.231.222 - - [27/Aug/2022 01:22:17] "GET /socket.io/?EIO=3&tran
sport=polling&t=1661563337697-0 HTTP/1.1" 200 -
```

Elaborado por: Wilson Quimbita.

3.1.7 Geomapas HERE

Con el objetivo de realizar conexión con el servicio de geoposición prestado por here, se procedió a suscribirse en su dominio web (HERE, 2023) (Here, n.d.). Con esto se consiguió generar unas “API KEY”, las cuales permiten el acceso a su vasta base de datos de mapas de todo el mundo, y se logró graficar en un mapa las coordenadas provistas anteriormente por el dispositivo GPS embebido tal y como se observa en la figura 3.8.

Figura 3.8. Mapa extraído de Here, que representa a una parte de la ciudad de Quito.



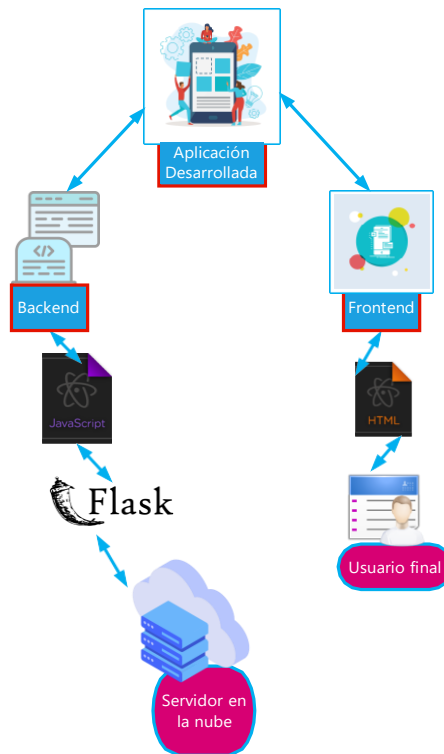
Elaborado por: Wilson Quimbita.

3.1.8 Interfaz de usuario creada sobre HTML/ Java Script

La interfaz creada consta de dos partes: la primera es el backend (figura 3.9) desarrollada sobre java script. En lo referente a este lenguaje, se lo utilizó para encapsular los mensajes de peticiones y respuestas desde y hacia el servidor web flask, al igual que la conexión con el servicio de mapas here.

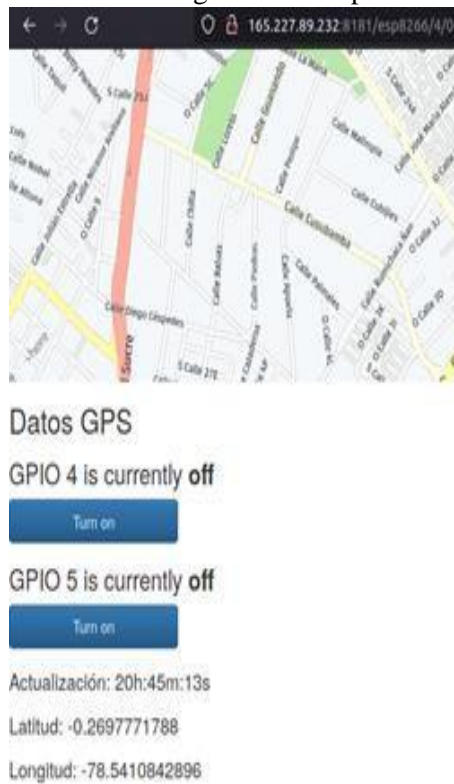
La otra parte es el frontend (figura 3.10), para lo cual se utilizó HTML. En este lenguaje se toman los datos de entrada y de salida agrupados en java script con el fin de ser mostrados en de una manera ordenada dentro de una interfaz gráfica web con el fin de interactuar con el usuario final, ya sea en una computadora o en un dispositivo móvil como se observa en la figura 3.10 se observa en la parte superior el mapa desplegado, el cual se obtuvo invocando al servicio de geolocalización HERE, en la parte baja de observa los botones interactivos GPIO4 y GPIO5, los cuales al presionarlos inician el proceso de petición de datos desde el módulo GPS hasta la interfaz gráfica para visualizarlos al final de la aplicación.

Figura 3.9. Interfaz gráfica web aplicación final. Esquema constitutivo.



Elaborado por: Wilson Quimbita.

Figura 3.10. Interfaz gráfica web aplicación final.



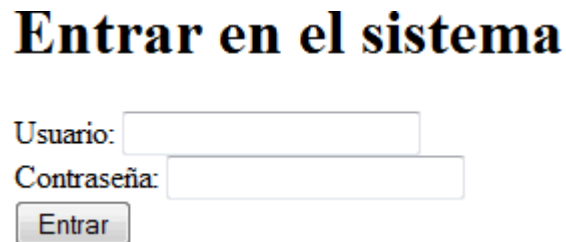
Elaborado por: Wilson Quimbita.

3.1.9 Conexión de usuario con el sistema a través de un dispositivo móvil

Una vez levantado todo el servicio sobre la nube, lo que resta es realizar la comunicación entre este servicio y un usuario externo, utilizando el protocolo http para lo cual se debe ingresar la dirección IP con la cual se encuentra configurada la máquina virtual, además del puerto sobre el cual se encuentra trabajando el servicio web flask de la siguiente manera (<http://165.227.89.232:8181>). Este acceso se debe realizar desde un teléfono celular o una computadora con acceso a internet.

Para una mayor seguridad antes de ingresar a la página principal, se implementó la petición de una contraseña. Esto con el fin de evitar que cualquier usuario desconocido con acceso a internet pueda rastrear el dispositivo libremente como se aprecia en la figura 3.11.

Figura 3.11. Validación para el ingreso a la aplicación.



Entrar en el sistema

Usuario:

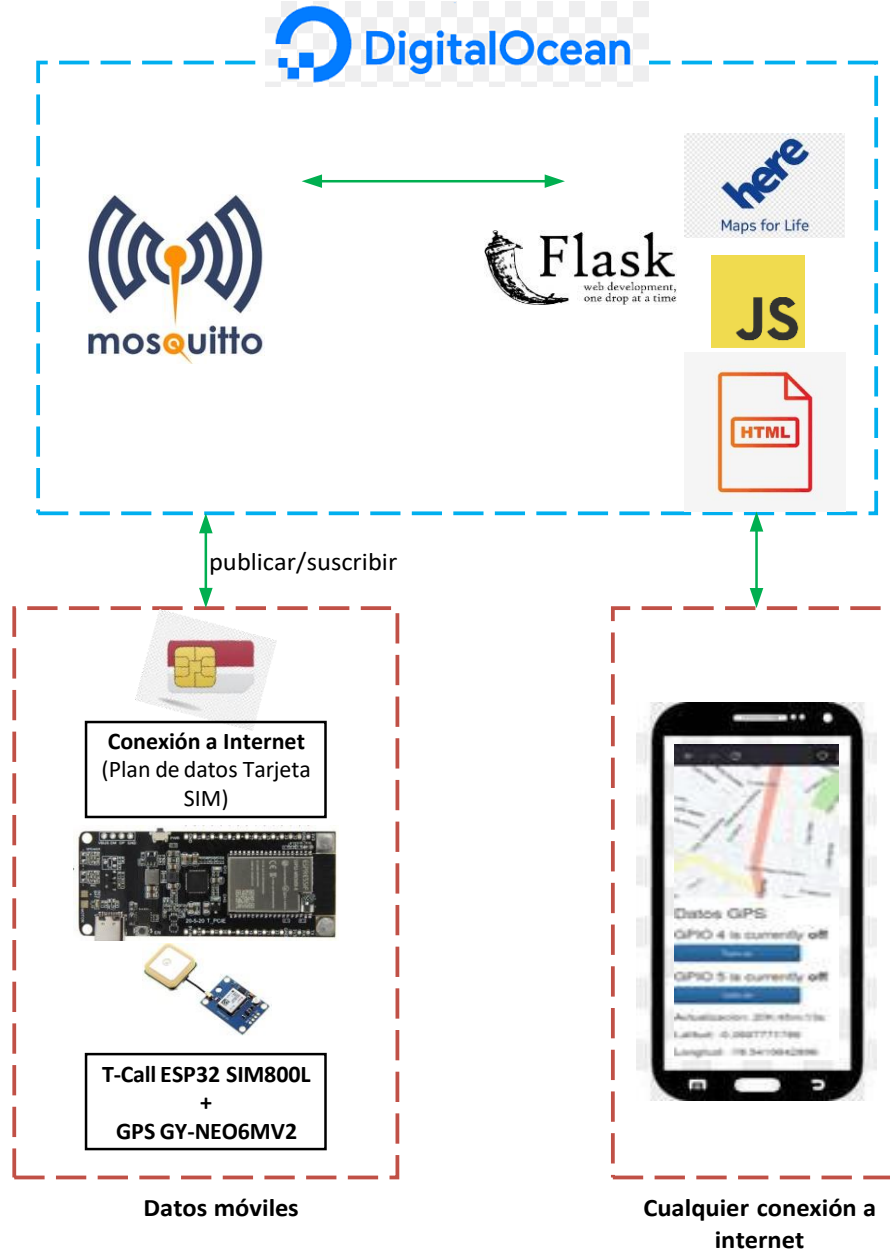
Contraseña:

Elaborado por: Wilson Quimbita.

3.1.10 Diagrama completo del sistema

En la figura 3.12 se puede apreciar ya el sistema completo de funcionamiento tanto de envío como de recepción de la información desde el dispositivo microcontrolado embebido hasta el teléfono móvil. Pasando por los datos de geoposición enviados a través de la red GSM hasta la nube y descargados mediante consulta por un aplicativo web desde un teléfono móvil con acceso a internet.

Figura 3.12. Diagrama esquemático del sistema.



Elaborado por: Wilson Quimbita.

4 CAPÍTULO 4

4.1 Análisis de Funcionamiento

Con el propósito de comprobar el correcto funcionamiento del módulo GPS implementado, se realizó varias pruebas de campo, según se manifiesta a continuación:

4.1.1 Cálculo de la precisión y exactitud en una misma locación

La primera prueba se realizó de la siguiente manera:

- Se adquirió información latitudinal y longitudinal en un lugar específico escogido dentro del Distrito Metropolitano de Quito (Solanda), a través del encendido y puesta en funcionamiento del dispositivo, asegurándose que el servicio en la nube se encontraba activo y en modo en línea ingresando a la aplicación móvil a través del terminal celular.
- Se guardó capturas de imágenes de la posición mostrada por el sistema GPS (figura 18), así como también del tamaño (ancho en metros) de la calle cercana a la posición.
- Se tomó la medida en metros de latitud y longitud de la posición real del módulo GPS con referencia a la calle aledaña más cercana.
- Se posicionó la distancia real de longitud y latitud en los mapas de las imágenes capturadas y las medidas tomadas previamente.
- Se calculó la distancia de error en metros de la longitud y latitud del módulo GPS, tomando como punto de referencia el valor real la coordenada (0, 0) figura 4.1. En la tabla 4.1 se observa un resumen de estos cálculos, los cuales se repitieron 20 veces.

Figura 4.1. Distancias obtenidas en latitud y longitud con respecto al valor real (0,0) del módulo GPS.



Elaborado por: Wilson Quimbita.

Tabla 4.1. Datos de longitud y latitud calculados a partir de datos de longitud y latitud adquiridos por el módulo GPS.

Distancia en metros desde la ubicación real (0,0)					
Sector Solanda					
Nro	Lat (m)	Long (m)	Nro	Lat (m)	Long (m)
1	12.787	17.825	11	14.2235	26.2588
2	12.035	18.23	12	19.3294	26.2588
3	10.2118	20.4235	13	8.7529	2.55
4	11.5976	21.882	14	8.7529	5.4705
5	6.929	21.882	15	13.05647	6.3647
6	8.53	21.08	16	8.7529	2.5529
7	8.68	22.6112	17	14.369	1.458
8	0.8753	27.79	18	14.58	1.458
9	0.364	27.712	19	13.0564	1.094
10	5.908	28.0824	20	13.4941	0.583

Elaborado por: Wilson Quimbita.

Concluido el cálculo de distancias de longitud y latitud con respecto al valor real (0,0), se procedió a calcular la precisión y exactitud del módulo GPS. Para esto, se continuó con los siguientes cálculos:

- Se calculó la exactitud para las medidas de latitud y longitud obtenidas por el módulo GPS, para esto, se utilizó la fórmula de la ecuación 4.1, la cual indica que la exactitud es la división entre la sumatoria del conjunto de valores obtenidos y el total de elementos que componen dicho conjunto (tabla 4.2).
- Se calculó la precisión para el conjunto de datos de latitud y longitud utilizando la ecuación 4.2, que representa la desviación estándar (tabla 4.2).

Tabla 4.2. Resumen de cálculos de exactitud y precisión efectuados para el módulo GPS.

Exactitud		Precisión	
$\frac{\sum_i^n X_i}{n} = \bar{X}$ Ec. (4.1)		$s = \sqrt{\frac{\sum_i^n (X_i - \bar{X})^2}{n - 1}}$ Ec. (4.2)	
media lat (m)	media long (m)	Lat (m)	Long (m)
10.3143	15.0783	4.5838	10.8349

Elaborado por: Wilson Quimbita.

Al finalizar esta primera prueba se puede apreciar que para la exactitud y la precisión en ambos casos la variable de latitud es la que más se acerca al valor real.

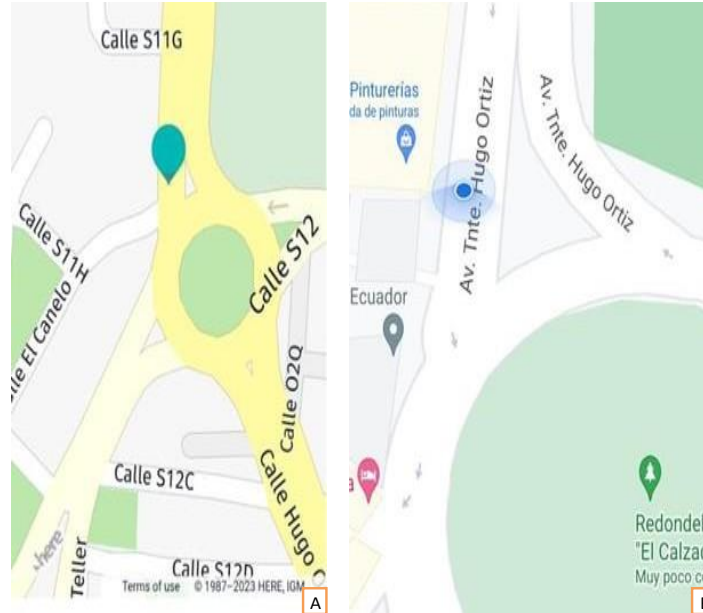
4.1.2 Cálculo del error absoluto en distintas locaciones

Para este segundo ensayo se procedió de la siguiente manera:

- La adquisición de datos tanto de latitud y longitud con el módulo GPS fue similar al procedimiento anterior con la diferencia de que se llevó a cabo en varias locaciones dentro del Distrito Metropolitano de Quito y no solo en una.
- Adicionalmente se ingresó con el celular de forma paralela a la aplicación de Google Maps y se obtuvo información en el mismo sitio que el módulo, obteniéndose así datos adicionales de latitud y longitud, desde una fuente de referencia posicional diferente.
- Se guardó capturas de imágenes de la posición mostrada por los sistemas GPS (figura 19), así como también del tamaño (ancho en metros) de las calles cercanas a la posición.
- Se tomó la medida en metros de latitud y longitud de la posición real de los dos dispositivos superpuestos con referencia a la calle aledaña más cercana.

En la figura 4.2a y 4.2b, se observa la adquisición de datos en un mismo lugar por medio de los dos dispositivos. Estas pruebas de campo fueron repetidas por 10 ocasiones más; en la tabla 4.3 se aprecia los datos adquiridos de longitud y latitud.

Figura 4.2. Imagen comparativa de geoposicionamiento entre los dos sistemas GPS. A) Módulo GPS desarrollado, B) GPS de Google Maps.



Elaborado por: Wilson Quimbita.

Tabla 4.3. Datos de longitud y latitud adquiridos por dos métodos, el desarrollado por nosotros y el de google maps.

Nro	Ubicación	Módulo GPS desarrollado		GPS Google Maps	
		Coordenadas en grados decimales (DD)		Coordenadas en grados decimales (DD)	
		Latitud (DD)	Longitud (DD)	Latitud (DD)	Longitud (DD)
1	El Recreo	-0.252054	-78.522087	-0.252003	-78.522017
2	Redondel Atahualpa	-0.249051	-78.531669	-0.249094	-78.531709
3	Redondel Calzado	-0.253796	-78.531181	-0.249094	-78.531709
4	IESS Quito Sur	-0.259528	-78.524811	-0.249094	-78.531709
5	El Comercio	-0.27254	-78.527679	-0.272572	-78.527646
6	Av Princesa Toa	-0.282059	-78.51651	-0.252572	-78.527646
7	Camino Conocoto	-0.282222	-78.491462	-0.272572	-78.527646
8	Parque Conocoto	-0.291965	-78.478256	-0.272572	-78.527646
9	El Cinto	-0.248931	-78.521015	-0.247921	-78.569052
10	Solanda	-0.269789	-78.541069	-0.269767	-78.541087

Elaborado por: Wilson Quimbita.

Terminada la adquisición de datos, se procedió a encontrar el error absoluto de posición para los

dos sistemas GPS. Con este fin, se concretaron las siguientes actividades:

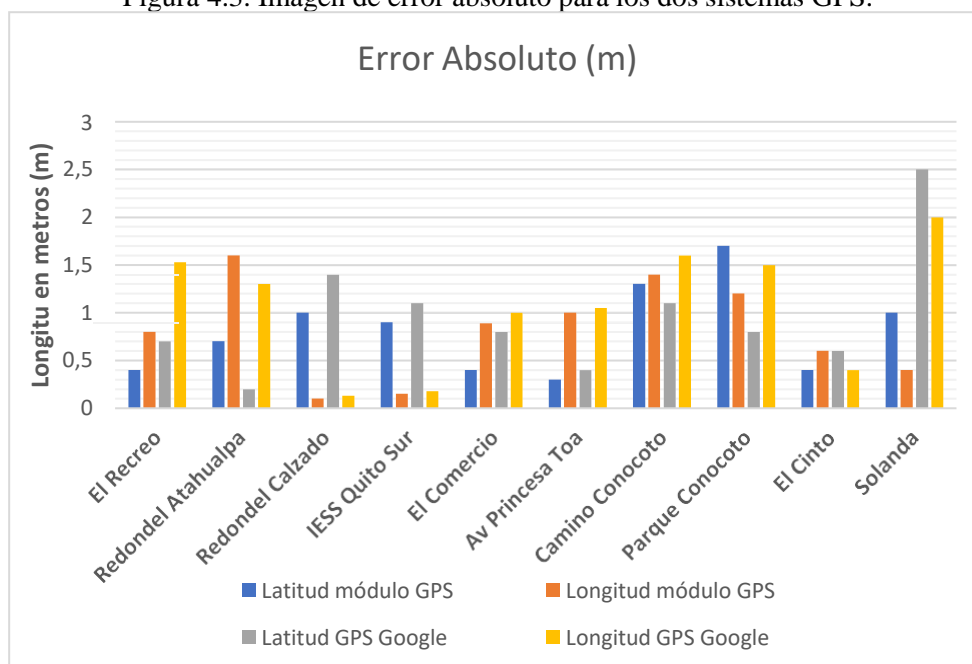
- Se posicionó la distancia real de longitud y latitud en los mapas de las imágenes capturadas y las medidas tomadas previamente.
- Se calculó la distancia de error en metros de la longitud y latitud de los dos sistemas GPS. En este caso estas distancias vienen a ser el error absoluto, ya que tomamos como punto de referencia el valor real la coordenada (0, 0) (figura 18) en cada una de las 10 locaciones. En la tabla 4.4 se observa un resumen de estas distancias en metros, que, a su vez, también son el error absoluto figura 4.3.

Tabla 4.4. Distancia de error en metros calculada de longitud y latitud de los métodos; el desarrollado por nosotros y el de Google Maps.

Nro.	Ubicación	Distancias en metros			
		Módulo Gps		Gps Google	
		Lat (m)	Long (m)	Lat (m)	Long (m)
1	El Recreo	0.4	0.8	0.7	1.53
2	Redondel Atahualpa	0.7	1.6	0.2	1.3
3	Redondel Calzado	1	0.1	1.4	0.13
4	IESS Quito Sur	0.9	0.15	1.1	0.18
5	El Comercio	0.4	0.89	0.8	1
6	Av Princesa Toa	0.3	1	0.4	1.05
7	Camino Conocoto	1.3	1.4	1.1	1.6
8	Parque Conocoto	1.7	1.2	0.8	1.5
9	El Cinto	0.4	0.6	0.6	0.4
10	Solanda	1	0.4	2.5	2

Elaborado por: Wilson Quimbita.

Figura 4.3. Imagen de error absoluto para los dos sistemas GPS.



Elaborado por: Wilson Quimbita.

En última instancia, en la figura 4.3 se muestra los resultados de un total de 40 mediciones entre latitud y longitud tomadas entre los dos dispositivos, los mismos que se han ponderado bajo los rangos de: bastante alto, si el error absoluto supera los 2m; alto, si el error absoluto se encuentra entre 1.5 y 2m; medio, si el error absoluto se encuentra entre 0.5 y 1.5m y; bajo, si el error absoluto está por debajo de 0.5m. Así:

2 mediciones corresponden a una ponderación de bastante alto, ya que superaron los 2m de error absoluto. Estas se alcanzaron únicamente con el GPS de Google.

- 4 mediciones corresponden a una ponderación de alto, ya que superaron los 1.5m de error absoluto. De estas, 2 pertenecen al módulo GPS construido y los 2 restantes al GPS de Google 23 mediciones corresponden a una ponderación de medio, ya que se encuentran entre 0.5 y 1.5m de error absoluto. 11 medidas son del módulo GPS y 12 del GPS de Google.
- 12 mediciones corresponden a una ponderación de bajo, ya que se encuentran por debajo de los 0.5m de error absoluto. Siete medidas pertenecen al módulo GPS y cinco al GPS de Google.

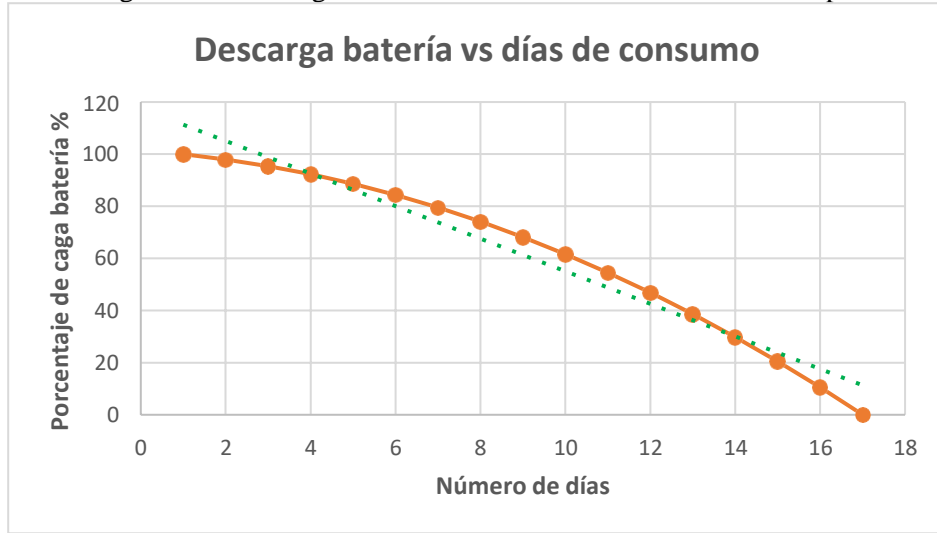
Nótese, por tanto, que, en 7 de las 10 locaciones donde se realizó las mediciones, el módulo GPS implementado por los autores del presente documento arroja un error relativo menor al del GPS de Google, lo cual apunta a la existencia de un rendimiento superior a nivel global.

4.1.3 Descarga de la batería

Para continuar el análisis, en el tercer ensayo se realizó la siguiente tarea:

- Se guardó información a diario del porcentaje de descarga con la ayuda del display de carga que posee este equipo. Los datos resultantes se aprecian en la figura 4.4.

Figura 4.4. Descarga del banco de baterías en función del tiempo.



Elaborado por: Wilson Quimbita.

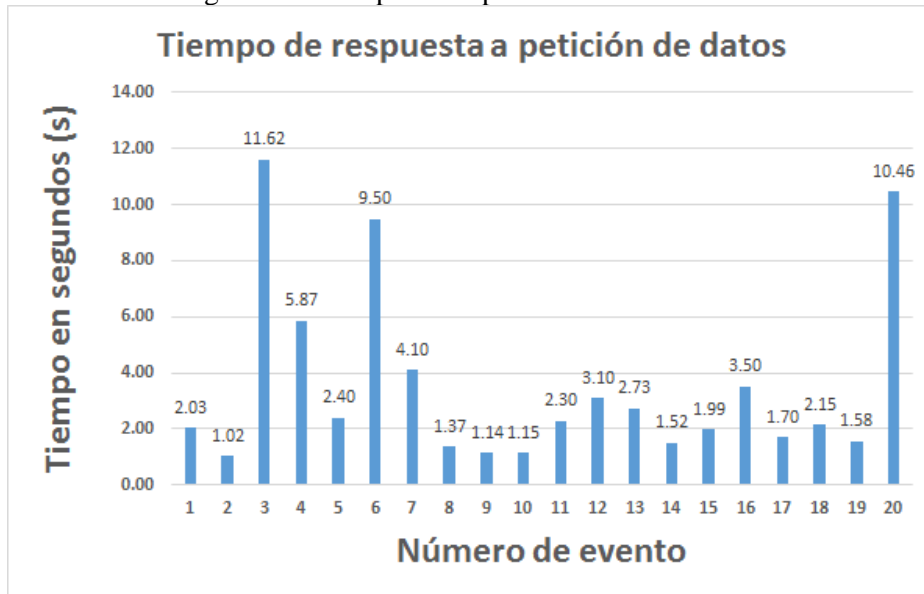
En la figura 4.4 se observa que la descarga de la batería no ocurre de una manera lineal como muestra la tendencia proyectada de la línea verde punteada. Más bien, sigue un patrón exponencial de descarga como muestra la línea anaranjada continua, descargándose menos porcentaje al principio, para terminar con un porcentaje mayor de descarga al final del ciclo. Así, en el primer día de trabajo —transcurridas 24 horas—, se produce un nivel de descarga de energía del 2%; para el noveno día — transcurridas 216 horas—, ocurre una descarga del 31.87% y; al completarse el décimo séptimo día —transcurridas 408 horas—, se produce una descarga del 99.9975%.

4.1.4 Tiempo de respuesta del sistema GPS

Para finalizar el análisis, en el cuarto experimento se tomó los tiempos de respuesta del sistema desde el momento en que comienza a funcionar el equipo hasta que se realiza una petición de ubicación desde el aplicativo web o móvil. Para conseguir esto se realizaron 20 peticiones en un mismo sitio, de estas 4 fueron con el equipo totalmente apagado y puesto en línea nuevamente luego de dos horas o más luego de permanecer apagado totalmente, los otros 16 restantes fueron con el equipo en línea, apagándolo y encendiéndolo en un tiempo de 10 minutos y en estado de

reposo.

Figura 4.5. Tiempo de respuesta del sistema GPS.



Elaborado por: Wilson Quimbita.

Respecto al tiempo de respuesta alcanzado con el sistema GPS, se ha realizado 20 mediciones y, tal y como se muestra en la figura 4.5, los valores registrados van desde un mínimo de 1.02 segundos hasta un máximo de 11.62 segundos. Teniendo en cuenta el valor de todas y cada una de las mediciones realizadas, se establece que la media esperada para recibir un tiempo de respuesta con el dispositivo GPS es de 3.56 segundos. La placa electrónica GY-NEO06MV2 que fue utilizada para realizar las mediciones, ha facultado que los tiempos de respuesta obtenidos no superen los 11.62 segundos luego de realizarse las peticiones de coordenadas hechas en la interfaz de usuario.

5 CONCLUSIONES

- Una red IoT (internet of things) posibilita la supervisión eficiente de la posición del módulo en diferentes lugares a través de la transmisión de datos desde el sensor GPS con tiempos de trabajo no mayores a 12 segundos hasta llegar al servidor en la nube MQTT y su respectiva exhibición en la interfaz gráfica desarrollada, la cual, permite operar de manera remota y dar seguimiento en tiempo real al funcionamiento global del sistema. La implementación total del sistema utilizando código abierto y el uso de sistemas microcontrolados permite abaratar los costos de licencias de programas y, del mismo modo, una colaboración externa de posibles mejoras si se sube los programas a repositorios GIT.
- La exactitud obtenida por el módulo GPS mediante la adquisición y tabulación de datos detalladas en la sección 4.1.1 en el presente trabajo se halla dentro de los parámetros de la revisión bibliográfica, es decir, la exactitud para la latitud es de 10.3 metros, mientras que para la longitud es de 15,1 metros. Adicionalmente, se obtuvo una sensibilidad del equipo GPS para latitud de un +- 4.6m y de longitud +-10.83 metros. Lo cual nos indica que el parámetro de latitud posee un menor error absoluto.
- El algoritmo de triangulación implementado en este módulo GPS, el cual consiste, en calcular la media aritmética de un paquete de 8 datos de longitud y 8 de latitud que llegan desde el sensor GPS hacia el microcontrolador, los cuales, se registran en los primeros 200 milisegundos en el microprocesador, ayuda a reducir el margen de error absoluto de los dos casos mencionados. Esto debido a que se toma en cuenta 8 mediciones por cada variable y no solamente la primera o la última.
- El dimensionamiento del banco de baterías es una parte importante del sistema, ya que nos permite conocer el tiempo de autonomía óptimo para el mismo. El ciclo de descarga del banco de baterías del módulo GPS coincide con los mostrados por los fabricantes de baterías secas de ciclo profundo (Padrón Jabib, pág. 15). Para calcular el porcentaje de descarga no lineal se midió el voltaje de descarga diario durante 18 días continuos con el equipo encendido.
- En consecuencia, al analizar y exponer los valores obtenidos de precisión, exactitud y tiempos de respuesta por medio de pruebas experimentales de adquisición y tabulación de datos detalladas en los apartados 4.1.1 y 4.1.2 del presente trabajo, se valida el funcionamiento del sistema GPS acorde al objetivo planteado en este documento. Debido

a que al comparar con el funcionamiento de un sistema GPS un teléfono móvil presenta valores más cercanos a la realidad, principalmente en la variable de latitud.

RECOMENDACIONES

- A fin de mantener una conexión inalámbrica estable entre el módulo GPS y la nube, se recomienda utilizar un chip GSM que proporcione la mayor cobertura posible a nivel nacional con el propósito de conseguir conectividad en la mayor cantidad de locaciones a nivel nacional.
- En futuros proyectos se puede alimentar la interfaz gráfica con trazados de líneas en el mapa por cambios de posición en segundos debido al movimiento del vehículo.
- Para siguientes investigaciones, se recomienda tomar en cuenta que las mediciones de tiempos de respuesta alcanzados y registrados en el presente documento se produjeron cuando el sistema había sido previamente inicializado de forma completa; de modo que, parecería factible ponderar los tiempos de respuesta a alcanzarse cuando el sistema no ha sido previamente inicializado, donde se habrá de evaluar de acuerdo con los tiempos en que la placa GPS GY-NEO06MV2 toma hasta conectarse a los diferentes satélites para luego recibir información. Además, otras variables a tomarse en cuenta en próximas investigaciones son el tiempo que toma hasta lograr la conexión a la red GSM y posteriormente a la nube tras iniciarse el sistema desde cero, luego que ha sido apagado o desconectado por lapsos superiores a dos horas.
- Finalmente, se considera pertinente que, para próximas aplicaciones y uso de este dispositivo o sus variantes, se considere la adquisición de una cuenta de pago para el servidor en la nube, ya que estos servicios facilitan la optimización de recursos y la actualización tecnológica y de seguridad que difícilmente se puede garantizar al contar únicamente con servidores físicos.

Lista de Referencias

- Aakvaag, N., & Frey, J. (enero de 2006). Redes de sensores inalámbricos. *Revista ABB*(2), 39-42. Obtenido de https://www.researchgate.net/publication/28135303_Redес_de_sensores_inalambricos
- alfaomega. (2016). *Sistmas Electrico y Electronico del automovil*. Mexico: MARCOMBO, S.A.
- Álvarez Brotons, X. (2004). *Control predictivo de canales de riego utilizando modelos de predicción de tipo Muskingum (primer orden) y de tipo Hayami (segundo orden)*. Barcelona: Universidad Politécnica de Cataluña. Obtenido de <https://upcommons.upc.edu/handle/2099.1/3330>
- ARDUINO. (19 de enero de 2023). *What is Arduino?* Obtenido de [arduino.cc: https://www.arduino.cc/en/Guide/Introduction](https://www.arduino.cc/en/Guide/Introduction)
- Bejerano, P. (2015). *ThinkBig*. Recuperado el octubre de 2022, de 5 útiles ejemplos de Internet de las cosas que pronto podrás probar: <https://blogthinkbig.com/5-utiles-ejemplos-de-internet-de-las-cosas-que-pronto-podras-probar/>
- Bernal, H., & Ladino, D. (2020). *Prototipo IOT para la Gestión de Actividades Cognitivas y Manejo de Información de Salud en Personas que Sufren de Alzheimer*. Bogotá: Universidad Distrital Francisco José De Caldas.
- Brunete, A., San Segundo, P., & Herrero, R. (2020). *Introducción a la Automatización Industrial*. Madrid: Universidad Politécnica de Madrid. Obtenido de https://bookdown.org/alberto_brunete/intro_automatica/
- Cerda Sánchez, D., & Pazmiño Días, I. (2011). *Diseño e Implementación de un Sistema con GPS y Control de Seguridad Vehicular con Comunicación GSM*. Latacunga: Escuela Politécnica del Ejército. Recuperado el 2022, de <http://repositorio.espe.edu.ec/handle/21000/3784>
- CISCO. (agosto de 2010). *Microsoft Internet Explorer*. Recuperado el octubre de 2022, de CISCO Network Academy: <https://www.netacad.com/courses/networking>
- Contreras, M. (2021). *El primer automóvil cumple 135 años: esta es la historia de su creación*. (J. A. Arévalo Miró Quesada, Ed.) Lima: El Comercio-Perú. Recuperado el octubre de 2022, de <https://elcomercio.pe/ruedas-tuercas/internacional/el-primer-automovil-cumple-135->

anos-asi-fue-la-historia-de-su-creacion-karl-benz-motorwagen-noticia/?ref=ecr

Corona, L., Abarca, G., & Carreño, J. (2014). *Sensores y Actuadores aplicados en arduino*. Azcapotzalco, Mexico D.F: Patria S.A.

DigitalOcean. (20 de enero de 2023). *Droplets*. Obtenido de www.digitalocean.com:
<https://www.digitalocean.com/products/droplets>

El Universo. (12 de 04 de 2009). Del delito a la contravención, el robo y hurto son más viables. *El Universo*. Recuperado el octubre de 2022, de <https://www.eluniverso.com/2009/04/12/1/1422/06BFC9D060D54F55B110FCB2AC8330D8.html>

Fernández Barcell, M. (2020). *WSN*. Obtenido de M.F.BARCELL:
<http://www.mfbarcell.es/conferencias/wsn.pdf>

Ferrer, J., García, V., & García, R. (2007). *Curso completo de HTML*. Madrid: TLDP-ES/LuCAS. Obtenido de <http://es.tldp.org/>

FLASK. (19 de enero de 2023). *User's guide*. Obtenido de <https://flask.palletsprojects.com>:
<https://flask.palletsprojects.com/en/2.2.x/>

Frohr, F., & Orttenburger, F. (1986). *SIEMENS "introducción al control electrónico"*. Barcelona, España: Marcombo.

Geotab. (2023). *Localización y seguimiento de flotas en tiempo real*. Recuperado el 20 de 07 de 2022, de <https://www.geotab.com/es/soluciones-gesti%C3%B3n-flotas/seguimiento-activos/>

GISGeography. (21 de 07 de 2022). *GPS Accuracy: HDOP, PDOP, GDOP, Multipath & the Atmosphere*. Obtenido de <https://gisgeography.com/gps-accuracy-hdop-pdop-gdop-multipath/>

González, J. (28 de 01 de 2022). El delito del robo de vehículos se incrementó 20% en 2021. *El Comercio*. Recuperado el 2022, de <https://www.elcomercio.com/actualidad/seguridad/delito-robo-vehiculos-delincuencia-victimas.html>

HERE. (19 de enero de 2023). *About Here*. Obtenido de www.here.com:
<https://www.here.com/company/about-us>

- HERE. (2023). *About Here*. Recuperado el 19 de enero de 2023, de https://developer.here.com/documentation/map-image/dev_guide/topics/examples-map-uncertainty.html
- João, B. D., Serralvo, F. A., & De Souza, C. L. (Octubre de 2019). A systematic review of smart cities and the internet of things as a research topic. *Cad. EBAPE.BR*, 17(4), 1115-1130. doi:<https://doi.org/10.1590/1679-395174442x>
- Lascorz, P. (2009). *Robotica y Automatica en el Aula: Trabajando con Microcontroladores Picaxe*. Colombia.
- Libraries, A. (2023). *Libraries Arduino*. Recuperado el 20 de 07 de 2022, de <https://www.arduino.cc/reference/en/libraries/category/communication/>
- Menéndez-Barzanallana, R. (2020). *Javascript*. Murcia: Universidad de Murcia.
- Mongay Batalla, J., Mastorakis, G., Mavromoustakis, C., & Pallis, E. (Edits.). (2017). *Beyond the Internet of things: everything interconnected*. Cham-Suiza: Springer.
- Mora, H., Gil, D., Signes, M. T., & Colom, J. F. (2016). *Método de Gestión del Internet de las Cosas Para la provisión de procesamiento flexible por Sistemas Cloud Computing*. Alicante: Universidad de Alicante. Obtenido de <chrome-extension://efaidnbmnnnibpcajpcgclefindmkaj/https://rua.ua.es/dspace/bitstream/10045/19119/1/Sistemas%20de%20adquisici%C3%B3n%20y%20Procesamiento%20de%20datos.pdf>
- Mosquitto. (20 de enero de 2023). *Eclipse Mosquitto*. Obtenido de [Mosquitto.org: https://mosquitto.org/](https://mosquitto.org/)
- Ñeco, G. R., Reinoso, O., García, N., & Aracil, R. (2003). *Apuntes de sistemas de control*. Elche: Club Universitario. Obtenido de <https://www.apuntesdeelectronica.com/industrial/apuntes-de-sistemas-de-control.htm>
- Padrón Jabib, F. A. (2014). *MANUAL DE BATERÍAS Y ACUMULADORES*. Colombia: Universidad Pontificia Bolivariana. Obtenido de <https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=0CAQQw7AJahcKEwiw34HxjuD8AhUAAAAAHQAAAAAQAg&url=https%3A%2F%2Frepository.upb.edu.co%2Fbitstream%2Fhandle%2F20.500.11912%2F1536%2FMANUAL%2520DE%2520B%25C3%259ASQUEDA%2520R%25>

- Perez, C., & Zamanillo, S. (2003). *Fundamentos de Television Analogica y Digital*. Cantabria: Graficas Calima, S.A.
- Pérez, M., Pérez, A., & Pérez, E. (2007). *Introduccion a los Sistemas de Control y Modelo Matemático para Sistemas Lineales invariantes en el tiempo*. San Juan: Universidad Nacional de San Juan.
- Piñeros, K., & González, D. (2004). *Dispositivos de Interconexión de Redes y Medios de Transmisión*. Cartagena de Indias: Universidad Tecnológica de Bolivar. Obtenido de <https://biblioteca.utb.edu.co/notas/tesis/0026243.pdf>
- Porro Sáez, I. (07 de febrero de 2019). *IoT: protocolos de comunicación, ataques y recomendaciones*. Recuperado el octubre de 2022, de incibe-cert: <https://www.incibe-cert.es/blog/iot-protocolos-comunicacion-ataques-y-recomendaciones>
- Pozo-Ruz, A., Ribeiro, A., García-Alegre, M., García, L., Guinea, D., & Sandoval, F. (2020). *Sistema de Posicionamiento Global (GPS): Descripción, Análisis De Errores, Aplicaciones y Futuro*. Madrid: Instituto de Automática Industrial. Obtenido de <https://www.peoplesmatters.com/Archivos/Descargas/GPS.pdf>
- Quispe, M., Castillo, C., Tejada, P., & García, F. (2022). Design of a Mobile Surveillance System using GPS Navigation and Trajectory Control via waypoints. *Latin American and Caribbean Consortium of Engineering Institutions*, 1-10. Obtenido de <https://laccei.org/LACCEI2022-BocaRaton/meta/FP508.html>
- Rigotti, G. (febrero de 2022). *Modelo OSI/ISO*. Recuperado el octubre de 2022, de COMUNICACION DE DATOS I: <chrome-extension://efaidnbmnnnibpajpcglclefindmkaj/https://users.exa.unicen.edu.ar/catedras/comdat1/material/ElmodeloOSI.pdf>
- Salazar, J., & Silvestre, S. (2017). Internet de las cosas. *TechPedia*, 7-34. Obtenido de <https://www.researchgate.net/file.PostFileLoader.html?id=55c4147460614b25168b45a3&assetKey=AS%3A273981783904256%401442333755312>
- Sanz Arnaiz, I. (2015). *Análisis de la Evolución y el Impacto de los Vehículos Eléctricos en la Economía Europea*. Madrid: Universidad Pontificia Comillas. Recuperado el octubre de 2022, de <chrome-extension://efaidnbmnnnibpajpcglclefindmkaj/https://repositorio.comillas.edu/xmlui/bitstream/handle/11531/3803/TFG001112.pdf>

- Shatilin, I. (5 de enero de 2016). *La evolución de la tarjeta SIM*. Obtenido de kaspersky daily:
<https://latam.kaspersky.com/blog/sim-card-history/6493/>
- Sturm, C. (27 de abril de 2012). *El origen de: La Nube*. Obtenido de fayerwayer:
<https://www.fayerwayer.com/2012/04/el-origen-de-la-nube/>
- Tutorials, R. N. (2023). *RandomNerdTutorials.com*. Recuperado el 19 de 07 de 2022, de
<https://randomnerdtutorials.com/cloud-mqtt-mosquitto-broker-access-anywhere-digital-ocean/>
- UNED. (2017). *Controladores industriales de diseño de alto nivel. Sistemas embebidos*. Madrid:
UNED. Obtenido de
http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia_ISE5_3_1.pdf
- Weber, R. (January de 2010). Internet of Things – New security and privacy challenges.
ScienceDirect, 26(1), 23-30. doi:<https://doi.org/10.1016/j.clsr.2009.11.008>

Anexo 2.

Prueba de campo del módulo GPS locación número 9 “El Cinto”.



Elaborado por: Wilson Quimbita.

Anexo 3.

Características del módulo TTGO T-Call ESP32 SIM800L.

Datos	Especificaciones
Voltaje	2.7 V-3.6V.
Chipset	ESP32 240 MHz Xtensa®Microprocesador LX6 de 32 bits de doble núcleo
Flash	QSPI flash 4 MB/PSRAM 8 MB.
Sram	SRAM de 520 kB.
WiFi	4 802.11 b/g/n (2.4 GHz) hasta 150 Mbps.
Bluetooth	4.2 Classic + Bluetooth de baja energía (BLE).
Conectividad	2G GSM/GPRS a través del módulo SIM800L + ranura para tarjeta Nano SIM.
USB a TTL	Chip CP2104.
Puerto	USB-C para alimentación y programación.
Corriente de funcionamiento	Alrededor de 70mA.
Interfaz:	UART, SPI, SDIO, I2C, PWM, I2S, IRGPIO, sensor táctil de condensador, ADC, DAC.
Puerto USB-C	5V 1A.
Cabecera JST	Batería de litio de 3.7V (corriente de carga de 500 mA).
Dimensiones	78.83mm × 28.92mm × 8.6mm.

Elaborado por: Wilson Quimbita.

Anexo 4

Proceso de creación de cuenta para uso de Servidor en la nube Digital Ocean

Si no se dispone de una cuenta, se puede registrarse en la web de Digital Ocean y se puede probar durante 60 días en los cuales la plataforma otorga crédito de manera gratuita para ser usada.

Para abrir una cuenta de Digital Ocean, es necesario dirigirse a su portal web y al apartado: “Registrarse”.

Open incident Network Connectivity in the APAC region To learn more, [check our status page](#)

MANAGE 165.227.89.232 Create ? ? Estimated costs: \$0.00

Create Droplets

Droplets are virtual machines that anyone can setup in seconds. You can use droplets, either standalone or as part of a larger, cloud based infrastructure.

Choose Region

New York San Francisco Amsterdam
Singapore London Frankfurt

\$4.00/month
\$0.006/hour

[CREATE VIA COMMAND LINE](#) Create Droplet

Se procede a ingresar los datos solicitados para crear una cuenta. Los datos son: correo electrónico, usuario, contraseña y número de tarjeta de crédito válida.

Open incident Network Connectivity in the APAC region To learn more, [check our status page](#)

MANAGE 165.227.89.232 Create ? ? Estimated costs: \$0.00

VPC Network - **default-nyc1** DEFAULT

All resources created in this datacenter will be members of the same VPC network. They can communicate securely over their Private IP addresses.

Choose an image

OS Marketplace Custom images

Ubuntu Fedora Debian CentOS AlmaLinux Rocky Linux

\$4.00/month
\$0.006/hour

[CREATE VIA COMMAND LINE](#) Create Droplet

Open incident Network Connectivity in the APAC region To learn more, [check our status page](#) ✕

MANAGE 165.227.89.232 Create Estimated costs: \$0.00

Apps
Droplets
Functions
Kubernetes
Volumes
Databases
Spaces
Container Registry
Images
Networking
Monitoring
Add-Ons
Billing

Droplet Type

SHARED CPU	DEDICATED CPU			
Basic (Currently selected)	General Purpose	CPU-Optimized	Memory-Optimized	Storage-Optimized

Basic virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and dev/test environments.

CPU options

Regular Disk type: SSD	Premium Intel NEW	Premium AMD NEW
----------------------------------	----------------------	--------------------

\$4.00/month
\$0.006/hour

[CREATE VIA COMMAND LINE](#) Create Droplet

Al hacerlo se recibe un crédito de 100usd, el cual puede ser usado a lo largo de 60 días, aunque se puede cancelar la cuenta en cualquier momento si ya no se está interesado en usar el servicio.

Open incident Network Connectivity in the APAC region To learn more, [check our status page](#) ✕

MANAGE 165.227.89.232 Create Estimated costs: \$0.00

Apps
Droplets
Functions
Kubernetes
Volumes
Databases
Spaces
Container Registry
Images
Networking
Monitoring
Add-Ons
Billing

Regular
Disk type: SSD

Premium Intel NEW
Disk: NVMe SSD

Premium AMD NEW
Disk: NVMe SSD

\$4/mo \$0.006/hour	\$6/mo \$0.009/hour	\$12/mo \$0.018/hour	\$18/mo \$0.027/hour	\$24/mo \$0.036/hour	\$48/mo \$0.071/hour
512 MB / 1 CPU 10 GB SSD Disk 500 GB transfer	1 GB / 1 CPU 25 GB SSD Disk 1000 GB transfer	2 GB / 1 CPU 50 GB SSD Disk 2 TB transfer	2 GB / 2 CPUs 60 GB SSD Disk 3 TB transfer	4 GB / 2 CPUs 80 GB SSD Disk 4 TB transfer	8 GB / 4 CPUs 160 GB SSD Disk 5 TB transfer

\$4.00/month
\$0.006/hour

[CREATE VIA COMMAND LINE](#) Create Droplet

Show all plans