



UNIVERSIDAD UTE

**FACULTAD DE CIENCIAS DE LA INGENIERÍA E
INDUSTRIAS**

CARRERA DE INGENIERÍA MECATRÓNICA

**PUERTA DE ENLACE PARA LA TRANSMISIÓN DE LA
TELEMETRÍA AUTOMOTRIZ HACIA UN SERVIDOR DE
ANÁLISIS DE DATOS**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO MECATRÓNICO**

ISMAEL FERNANDO PAZ MONTERO

DIRECTOR: ING. GUILLERMO MOSQUERA M.SC

Quito, junio 2021

© Universidad UTE. 2021

Reservados todos los derechos de reproducción

FORMULARIO DE REGISTRO BIBLIOGRÁFICO

TRABAJO DE TITULACIÓN

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	1722059993
APELLIDO Y NOMBRES:	Paz Ismael Fernando
DIRECCIÓN:	Nuño de Valderrama N30-78 y Cuero Caicedo
EMAIL:	ismaelpazmontero@gmail.com
TELÉFONO FIJO:	(02)2557065
TELÉFONO MOVIL:	0996060408

DATOS DE LA OBRA				
TÍTULO:	Puerta de enlace para la transmisión de la telemetría automotriz hacia un servidor de análisis de datos.			
AUTOR O AUTORES:	Ismael Fernando Paz Montero			
FECHA DE ENTREGA DEL PROYECTO DE TITULACIÓN:	28-07-2020			
DIRECTOR DEL PROYECTO DE TITULACIÓN:	Ing. Guillermo Mosquera M. Sc.			
PROGRAMA	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">PREGRADO</td> <td style="width: 10%; text-align: center;">x</td> <td style="width: 40%; text-align: center;">POSGRADO</td> </tr> </table>	PREGRADO	x	POSGRADO
PREGRADO	x	POSGRADO		
TÍTULO POR EL QUE OPTA:	Ingeniero Mecatrónico			
RESUMEN: Mínimo 250 palabras	En el presente trabajo se desarrolló el diseño e implementación de un dispositivo tecnológico IoT(Internet de las Cosas) como puerta de enlace para transmitir la telemetría automotriz como temperatura del aceite del motor, temperatura del refrigerante del motor, presión de combustible, velocidad del			

vehículo, RPM del motor, posición del acelerador, entrada de nivel de combustible, datos del acelerómetro, datos de giroscopio, tiempo de ejecución desde el arranque del motor, presión barométrica, vida restante de la batería híbrida, voltaje de la batería, entre otras en tiempo real a través de WiFi o red celular 4G a un servidor de análisis de datos como Amazon Web Services, Microsoft Azure u otros, con el objetivo de monitorear, analizar y estudiar cada una de las variables en bruto recopiladas para luego tratarlas y así puedan estas aportar un valor agregado a las partes interesadas del vehículo. Este dispositivo fue Plug and Play(Conéctalo y Úsalo) ya que bastó con enchufarlo en el puerto OBD-II que se encuentra en la parte inferior del volante de vehículos compatibles con los protocolos CAN 500Kbps/29bit, CAN 250Kbps/29bit, KWP2000 Fast y KWP2000 5Kbps los cuales son los más usados en la industria automotriz, además contó con una memoria flash interna y una externa en microSD para seguir almacenando los datos en lugares remotos donde no exista conexión a internet y cuando esta se restablezca los datos encolados sean enviados para ser analizados, por último se acopló una batería de respaldo que dure por un tiempo

	<p>mínimo de 15 minutos antes del fallo en caso de que el vehículo sea desconectado, su batería se agote o sufra algún accidente.</p>
<p>PALABRAS CLAVES:</p>	<p><TELEMETRÍA>, <SERVIDOR>, <PROTOCOLOS>, <PUERTA DE ENLACE>, <VARIABLES>, <PARTES INTERESADAS></p>
<p>ABSTRACT:</p>	<p>This research is focused on the design and implementation of a technological device as a gateway for transmitting car telemetry such as engine oil temperature, engine coolant temperature, fuel pressure, car speed, engine revolutions per minute, throttle position, fuel level input, accelerometer data, gyroscope data, run time from engine start, barometric pressure, hybrid battery life remaining, battery voltage, among others in real time through WiFi and 4G cellular network to a data analysis server such as Amazon Web Services, Microsoft Azure or others with the objective of monitoring, analyzing and studying each of the raw variables and then treat them and can bring added value to vehicle stakeholders. This device was plug and play, because it was enough to connect it to the OBD-II port of the vehicle that is located at the bottom of the steering Wheel compatible with protocols CAN 500Kbps/29bit, CAN 250Kbps/29bit, KWP2000 Fast and KWP2000 5Kbps</p>

	<p>which are the most used in the automotive industry, it also had an internal flash memory and an external micro SD memory to continue storing data in remote places where there is no internet connection and when there is a connection send the stored data to be analyzed.</p> <p>Finally, a backup battery was inserted that lasts for a minimum time of 15 minutes before failure if the car is disconnected, your battery runs out or suffers an accident.</p>
KEYWORDS	<TELEMETRY>, <SERVER>, <PROTOCOLS>, <GATEWAY>, <VARIABLES>, <STAKEHOLDERS>

Se autoriza la publicación de este Proyecto de Titulación en el Repositorio Digital de la Institución.

f: 

PAZ MONTERO ISMAEL FERNANDO

1722059993


DECLARACIÓN Y AUTORIZACIÓN

Yo, **ISMAEL FERNANDO PAZ MONTERO**, CI 1722059993 autor/a del trabajo de titulación: **PUERTA DE ENLACE PARA LA TRANSMISIÓN DE LA TELEMETRÍA AUTOMOTRIZ HACIA UN SERVIDOR DE ANÁLISIS DE DATOS** previo a la obtención del título de **INGENIERO MECATRÓNICO** en la Universidad UTE.

1. Declaro tener pleno conocimiento de la obligación que tienen las Instituciones de Educación Superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación de grado para que sea integrado al Sistema Nacional de información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Autorizo a la BIBLIOTECA de la Universidad UTE a tener una copia del referido trabajo de titulación de grado con el propósito de generar un Repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Quito, 15 de septiembre de 2021

f: _____


PAZ MONTERO ISMAEL FERNANDO
1722059993

CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor de tesis de grado, certifico que el presente trabajo que lleva por título **PUERTA DE ENLACE PARA LA TRANSMISIÓN DE LA TELEMETRÍA AUTOMOTRIZ HACIA UN SERVIDOR DE ANÁLISIS DE DATOS** para aspirar al título de **INGENIERO MECATRÓNICO** fue desarrollado por **PAZ MONTERO ISMAEL FERNANDO**, bajo mi dirección y supervisión, en la Facultad de Ciencias de la Ingeniería e Industrias; y que dicho trabajo cumple con las condiciones requeridas para ser sometido a la presentación pública y evaluación por parte del Jurado examinador que se designe.



Ing. Guillermo Mosquera M. Sc.

DIRECTOR DEL TRABAJO

C.I.0802613059

DECLARACION JURAMENTADA DEL AUTOR

Yo, Ismael Fernando Paz Montero, portador(a) de la cédula de identidad N.º 1722059993, declaro que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en ese documento.

La Universidad UTE puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

f:


PAZ MONTERO ISMAEL FERNANDO

1722059993

Quito, 08 de junio del 2021

En nombre de INGENESIS IT CONSULTING, quiero mostrar la conformidad por los servicios prestados por el Sr. Ismael Paz con número de cédula 1722059993 para el Proyecto: ***“Desarrollo del dispositivo de puerta de enlace para la transmisión de telemetría automotriz hacia un servidor de análisis de datos”***, se realizó todo de acuerdo con el contrato firmado entre ambas partes y dentro del plazo establecido.

Seguiremos manteniendo contacto, ya que me consta que su trabajo ha sido de gran apoyo para el progreso de la compañía.



Firmado electrónicamente por:

**JUAN
PAZ**

Juan Paz

Gerente general de INGENESIS IT CONSULTING

Mariana de Jesús e7-8 y la Pradera, edificio Business Plus, piso 2, oficina
206 PBX: 02 222 – 5478

www.ingenesis.com.ec

DEDICATORIA

A los grandes héroes de la fe.....

Por la fe Abel presentó a Dios una ofrenda más aceptable que la que presento Caín, fue por la fe que Enoc ascendió al cielo sin morir, por fe Noé construyó un barco para salvar a toda su familia, por fe Abraham obedeció cuando Dios lo llamo para que dejara su tierra y fuera a otra herencia sin saber a dónde iba, fue por fe que Isaac prometió a sus hijos bendiciones para el futuro, fue por fe que cuando nació Moisés. Sus padres lo escondieron durante tres meses sin temor a desobedecer la orden del rey, por fe Moisés cuando fue adulto se rehusó ser llamado hijo del faraón y prefirió ser maltratado con el pueblo de Dios a disfrutar de placeres momentáneos pues considero que era mejor sufrir por causa de Cristo que poseer los tesoros de Egipto, por fe seguía firme en su camino porque tenía los ojos puestos en el invisible. Por fe mi madre viajo a Italia abandonando sus sueños para que los míos y los de mis hermanos se hicieran realidad, por fe mi padre construyo cometas de muchos colores cada verano para enseñarme que la felicidad se encuentra en las cosas más sencillas de esta vida, por fe mi abuelita me hablo de Dios todos los días de su vida, por fe mi profesora de segundo grado "Charito" me enseñó el salmo 23 para que cuando ande en valle de sombra y de muerte no tema mal alguno. Por fe mi hermano Polito compró mi primera bicicleta mostrándome que cada caída solo era parte de la travesía, por fe mi hermana Ely soportó todas mis noches de pesadillas y me ayudo a enfrentar mis fantasmas; por esa misma Fe mi hermana Eleana me enseñó a saludar a decir por favor y gracias, por fe mi hermana menor escucha cada sermón que le doy y aunque parece que no presta atención sabe que es hija del rey y así vengan con carros, dinero o poder ningún hombre la puede envolver, por esa misma fe mi hermano Michael me abrió las puertas de su empresa, de su casa y de su corazón sin esperar nada a cambio y así puedo nombrar a muchos más pero necesitaría demasiado tiempo para hacerlo, Gedeón, Barac, Sansón, Jefté, David, Daniel, Jeremías, Elías, Josué. Por la fe conquistaron reinos, gobernaron con justicia, cerraron bocas de leones, apagaron llamas de fuego, su debilidad se convirtió en fortaleza. Sin embargo, otros fueron torturados porque rechazaron negar a Dios a cambio de la libertad. Algunos fueron ridiculizados y sus espaldas fueron laceradas con látigos; otros fueron encadenados en prisiones. Algunos murieron apedreados, a otros los cortaron por la mitad y a otros los mataron a espada. Algunos anduvieron vestidos con pieles de ovejas y cabras desposeídos oprimidos y maltratados. ¡Este mundo no era digno de ellos ¡.

Pero quien se lleva todo el mérito es quien dio la vida por mí en un madero, Jesús nombre sobre todo nombre.

AGRADECIMIENTOS

Ha sido un largo camino para llegar a este punto, un camino lleno de baches, desiertos, valles, quebradas y paraísos. Normalmente hay que agradecer a las personas que nos ayudaron a salir adelante o a quienes apoyaron nuestro caminar pero yo agradezco al niño que golpeó mi cara por primera vez despertándome del sueño de hadas en el que vivía, agradezco el día en que con mis hermanos nos quedamos dormidos tanto llorar por no haber probado un plato de comida ya que pude entender el dolor que sienten otros y me dio propósito en la vida para servir a los demás, agradezco que me hayan quitado el título de abanderado del pabellón nacional en la escuela y me dieran el segundo lugar; esto me enfocó a buscar títulos eternos y hacer tesoros en el cielo donde ni la polilla ni el orín corrompen y donde los ladrones no minan ni hurtan porque donde está nuestro tesoro ahí está nuestro corazón. Agradezco las burlas que recibí en quinto curso por el acné que tuve , muchas chicas se alejaron de mí y “amigos” me despreciaron por lo que encontré en el deporte un refugio para mi dolor y sin darme cuenta me alejo del tabaco, drogas, sexo y alcohol donde la gran mayoría cayó, gracias a mi primera novia que me abandonó por un chico con carro, buena ropa y dinero, esto me empujo a buscar artículos de mayor lujo que puedan ganar a los que tenía este chico y me encontré con el amor, la paz, la bondad, la amabilidad, la paciencia, la fe, la mansedumbre, el dominio propio y aunque aún no obtengo todos ellos, estoy seguro que estos abren más puertas y oportunidades que el dinero, el poder y conquistar con ellos a una mujer solo pocos hombres hoy en día lo saben hacer, agradezco a todos mis enemigos, detractores y piedras de tropiezo porque sin ellos no hubiera tenido el coraje para levantarme y dar lugar a una batalla más, pero sobre todo quiero agradecer a mi nutricionista que no me dejó morir de hambre jamás, a mi tutor que me enseñó toda ciencia y letra con amor, a mi entrenador que me enseñó a través del deporte el honor y la disciplina siempre me impulsaba a dar un paso más allá, a mi mentor que me mostro que todas las herramientas necesarias para vencer las llevo dentro de mí, a mi doctor que curo cada cicatriz y laceración que recibí por alguien más y por enseñarme a curar las que también hice yo, en fin todos ellos se resumen en uno, en ti Dios, gracias por tantos momentos conmigo, por tantos éxitos y fracasos, por tantos momentos junto al espíritu santo, por tantas lágrimas y sonrisas, gracias por salvarme de mi mismo y aunque sigo siendo un peregrino, vagabundo e indigno en este mundo hoy puedo caminar confiado bajo tu gracia y tu justificación, gracias, muchas gracias por esto y mucho más mi dulce carpintero.

ÍNDICE DE CONTENIDOS

	PÁGINA
RESUMEN	vi
ABSTRACT	2
1. INTRODUCCIÓN	3
2. METODOLOGÍA	14
2.1 REQUERIMIENTOS	15
2.2 DISEÑO DEL SISTEMA	15
2.3 DISEÑO ESPECÍFICO	17
2.3.1 HARDWARE DE CONTROL.....	17
2.3.1.1 Selección del microcontrolador.	17
2.3.1.2 Implementación de los módulos de control.	19
2.3.1.2.1 Conexión con la ecu	19
2.3.1.2.2 Módulo lte cat4.....	20
2.3.1.2.3 Sensor de movimiento mpu-9250 (9-dof).....	22
2.3.1.2.4 Módulo coprocesador stm32f103cx	24
2.3.1.2.5 Módulo y antena gnss integrados m8030 10hz.....	24
2.3.1.2.6 Módulo cp2102 puente usb a uart.....	26
2.3.1.2.7 Módulo w25qxx memoria flash.....	27
2.4 INTEGRACIÓN DE COMPONENTES ELECTRÓNICOS.....	28
2.4.1 CONECTOR MACHO OBD-II.....	28
2.4.2 RANURAS PARA TARJETAS.....	29
2.4.3 MAPA DE E / S	29
2.5 ELABORACIÓN DE LA CARCASA DEL DISPOSITIVO.	32
2.3.2 SOFTWARE DE CONTROL.....	33
2.3.2.1 Desarrollo del código de lectura y procesamiento de datos obd... ..	33

2.3.2.1.1 Librerías principales utilizadas	33
2.3.2.2 Desarrollo del código de inicialización de todos los componentes de registro de datos.....	35
2.3.2.3 Desarrollo del código de envío de datos http api	37
2.3.2.3.1 Alimentación de datos del dispositivo al servidor.....	37
2.3.2.3.2 Notificación de eventos.....	37
2.3.2.3.3 Transmisión de datos.....	37
2.3.2.3.4 Iniciar sesión y cerrar sesión.....	37
2.3.2.3.5 Notificación por udp	38
2.3.2.3.6 Notificación por http	38
2.3.2.3.7 Alimentación de datos por udp.....	39
2.3.2.3.8 Alimentación de datos mediante http get	40
2.3.2.3.9 Alimentación de datos mediante http post	41
2.3.2.3.10 Obteniendo datos del servidor	41
2.3.2.3.11 Obtener datos instantáneos	42
2.3.2.3.12 Secuencia de operación	43
2.3.2.3.13 Obteniendo datos del historial.....	43
2.3.2.4 Programación del código de recopilación y procesamiento de datos.....	46
2.3.2.5 Desarrollo del código de inicialización de la red, mantenimiento de la conexión y transmisiones	48
2.3.3 RESPALDOS DEL DISPOSITIVO.....	50
2.3.3.1 Implementación de la batería de respaldo para el dispositivo.....	50
2.3.2.1.1 Cálculo de la batería de respaldo	50
2.3.2.1.2 Circuito de control de conmutación de la batería principal a la batería de respaldo	52
3. ANÁLISIS DE LOS RESULTADOS.....	57
3.1 COMPARACIÓN DE DATOS.....	57
3.2 PRUEBAS DE FUNCIONAMIENTO	59
3.2.1 ACELERACIÓN.....	59
3.2.2 VELOCIDAD.....	59

3.2.3 VOLTAJE DE LA BATERÍA.....	61
3.2.4 CARGA DEL MOTOR	62
3.2.5 GEOLOCALIZACIÓN	63
3.3 SERVICIOS QUE SE PUEDEN OFRECER A LOS USUARIOS A TRAVÉS DEL DISPOSITIVO DE PUERTA DE ENLACE	63
3.4 ARQUITECTURA DE CONEXIÓN DEL DISPOSITIVO DE PUERTA DE ENLACE	64
4. CONCLUSIONES Y RECOMENDACIONES	65
4.1 CONCLUSIONES	65
4.2 RECOMENDACIONES.....	66
5. BIBLIOGRAFÍA	68

INDICE DE TABLAS

	PÁGINA
Tabla 1. Tiempo de mantenimiento	7
Tabla 2. Costo de mantenimiento.....	7
Tabla 3. Comparaciones del microcontrolador Espressif ESP32 frente al ESP8266 yNodeMCU.....	18
Tabla 4. Especificaciones del módulo LTE Cat4 SIM7600E-H.....	22
Tabla 5. Especificaciones técnicas del Sensor de movimiento MPU-9250 (9- DOF)	23
Tabla 6. Características técnicas del Módulo coprocesador STM32F103CX...	24
Tabla 7. Características técnicas del Módulo y antena GNSS integrados M8030 10Hz.....	25
Tabla 8. Características técnicas del Módulo CP2102	26
Tabla 9. Características técnicas del Módulo W25QXX memoria flash.....	27
Tabla 10. Mapa de pines de conexión de todos los módulos al microcontrolador principal ESP32.....	30
Tabla 11. Especificaciones técnicas de la batería de respaldo LiPoTurnigy nano-tech	51
Tabla 12. Comparación de la velocidad receptada en el servidor de análisis de datos Freematics contra la velocidad apuntada cada 2 segundos del panel de control del vehículo.....	60
Tabla 13. Comparación del voltaje de la batería del vehículo receptado en el servidor de análisis de datos Freematics contra el voltaje medido por un voltímetro previamente instalado en el carro en un lapso de tiempo de cada 2 segundos.	61

ÍNDICE DE FIGURAS

	PÁGINA
Figura 1. Evolución del número de dispositivos conectados al IoT en todo el mundo(Fernández, 2020).....	4
Figura 2. El robo de vehículos en Ecuador	8
Figura 3. AutoConectado, la transformación para innovar en seguros vehiculares.	9
Figura 4. Metodología de diseño en V para el diseño de sistemas mecatrónicos	14
Figura 5. Diagrama de bloques de conexión de todos los elementos.....	15
Figura 6. Flujograma del diseño del dispositivo de puerta de enlace.....	16
Figura 7. Mapa de pines del microcontrolador ESP32	19
Figura 8. Diagrama de conexión del puerto OBD-II estandarizado.....	19
Figura 9. Dimensiones físicas del móduloSIM7600E-H	21
Figura 10. Diseño físico del Sensor de movimiento MPU-9250 (9-DOF)	23
Figura 11. Diseño esquemático del coprocesador STM32F103CX.....	25
Figura 12. Diseño esquemático del Módulo CP2102 puente USB a UART	26
Figura 13. Diseño esquemático del Módulo W25QXX memoria flash.....	28
Figura 14. Diseño esquemático de la integración de todos los módulos de control con el microcontrolador ESP32	31
Figura 15. Resultado final de la carcasa del dispositivo de puerta de enlace. .	32
Figura 16. Flujograma de la programación del código de lectura y procesamiento de datos OBD	35
Figura 17. Flujograma de la programación del código de inicialización de todos los componentes de registro de datos.....	37
Figura 18. Desarrollo del código de envío de datos HTTP API	46
Figura 19. Tarjeta SIM Hologram IoT.....	47
Figura 20. Diagrama del proceso de recopilación y procesamiento de datos...48	
Figura 21. Diagrama de flujo del proceso de inicialización de la red, mantenimiento de la conexión y transmisiones.	50
Figura 22. Diagrama esquemático del circuito de control para la conmutación de la alimentación del dispositivo de puerta de enlace.	53
Figura 23. Dispositivo de puerta de enlace con los puertos externos para conectar la batería LiPo o algún sensor adicional para analizar una variable adicional que desee el cliente.....	54
Figura 24. Dispositivo de puerta de enlace conectado al conector Molex para añadir una fuente de alimentación externa.	55

Figura 25. Api Freematics para la comprobación de todos los módulos y el microcontrolador utilizados en el dispositivo de puerta de enlace antes de ser instalado en un vehículo.....	57
Figura 26. Servidor de análisis de datos Freematics con el token de acceso entregado por la Api de comprobación Freematics	58
Figura 27. Datos del grafico de aceleración tomados en un lapso de 00:03:37 minutos en un diagrama de aceleración (g) vs tiempo (s).....	59
Figura 28. Datos del grafico de velocidad tomados en un lapso de 00:03:27 minutos en un diagrama de velocidad (Km/h) vs tiempo (s).....	59
Figura 29. Datos del grafico del voltaje de la batería tomados en un lapso de 00:03:47 minutos en un diagrama de voltaje de la batería (V) vs tiempo (s).....	61
Figura 30. Datos del grafico de la carga del motor tomados en un lapso de 00:04:39 minutos en un diagrama de carga del motor (%) vs tiempo (s).	62
Figura 31. Geolocalización del automóvil con las gráficas de altitud y velocidad en una distancia de 3.8 Km con una duración de 00:01:31 minutos con un promedio de velocidad de 148,7 Km/h.	63
Figura 32. Arquitectura de conexión del dispositivo de pueta de enlace IoT ..	64

RESUMEN

En el presente trabajo se desarrolló el diseño e implementación de un dispositivo tecnológico IoT (Internet de las Cosas) como puerta de enlace para transmitir la telemetría automotriz como temperatura del aceite del motor, temperatura del refrigerante del motor, presión de combustible, velocidad del vehículo, RPM del motor, posición del acelerador, entrada de nivel de combustible, datos del acelerómetro, datos de giroscopio, tiempo de ejecución desde el arranque del motor, presión barométrica, vida restante de la batería híbrida, voltaje de la batería, entre otras en tiempo real a través de Wi-Fi o red celular 4G a un servidor de análisis de datos como Amazon Web Services, Microsoft Azure u otros, con el objetivo de monitorear, analizar y estudiar cada una de las variables en bruto recopiladas para luego tratarlas y así puedan estas aportar un valor agregado a las partes interesadas del vehículo. Este dispositivo fue Plug and Play (Conéctalo y Úsalo) ya que bastó con enchufarlo en el puerto OBD-II que se encuentra en la parte inferior del volante de vehículos compatibles con los protocolos CAN 500Kbps/29bit, CAN 250Kbps/29bit, KWP2000 Fast y KWP2000 5Kbps los cuales son los más usados en la industria automotriz, además contó con una memoria flash interna y una externa en micro SD para seguir almacenando los datos en lugares remotos donde no exista conexión y cuando esta se restablezca los datos encolados sean enviados para ser analizados, por último se acopló una batería de respaldo que dure por un tiempo mínimo de 15 minutos antes del fallo en caso de que el vehículo sea desconectado, su batería se agote o sufra algún accidente.

Palabras Clave:

<TELEMETRÍA>, <SERVIDOR>, <PROTOCOLOS>, <PUERTA DE ENLACE>, <VARIABLES>, <PARTES INTERESADAS>

ABSTRACT

This research is focused on the design and implementation of a technological device as a gateway for transmitting car telemetry such as engine oil temperature, engine coolant temperature, fuel pressure, car speed, engine revolutions per minute, throttle position, fuel level input, accelerometer data, gyroscope data, run time from engine start, barometric pressure, hybrid battery life remaining, battery voltage, among others in real time through WiFi and 4G cellular network to a data analysis server such as Amazon Web Services, Microsoft Azure or others with the objective of monitoring, analyzing and studying each of the raw variables and then treat them and can bring added value to vehicle stakeholders. This device was plug and play, because it was enough to connect it to the OBD-II port of the vehicle that is located at the bottom of the steering Wheel compatible with protocols CAN 500Kbps/29bit, CAN 250Kbps/29bit, KWP2000 Fast and KWP2000 5Kbps which are the most used in the automotive industry, it also had an internal flash memory and an external micro SD memory to continue storing data in remote places where there is no internet connection and when there is a connection send the stored data to be analyzed. Finally, a backup battery was inserted that lasts for a minimum time of 15 minutes before failure if the car is disconnected, your battery runs out or suffers an accident.

Keywords:

<TELEMETRY>, <SERVER>, <PROTOCOLS>, <GATEWAY>, <VARIABLES>, <STAKEHOLDERS>.

1. INTRODUCCIÓN

El Instituto Nacional de Estadísticas y Censos (INEC) informa el aumento del parque automotor matriculado en Ecuador creció en más de 1,4 millones de vehículos en una década lo que situó la cifra sobre los 2,4 millones de unidades a 2018 prediciendo un aumento exponencial en los siguientes años. Si se compara por provincias, el mayor número de vehículos matriculados se registra en Pichincha con 540 827, seguido por Guayas, con 529 603 unidades. El resto de las provincias acumulan un total de 394380 unidades matriculadas. Por otro lado, Cuando se hace una comparación del parque automotor con países del continente, Chile es el país con el mayor número de vehículos matriculados por cada mil habitantes, seguido por Colombia, Perú, Bolivia y Ecuador(Comercio, 2019).

El sector automotriz ecuatoriano es sin duda muy fuerte según lo que se desprende de las cifras suministradas por la Superintendencia de Compañías(Superintendencia de Compañías, 2016).

En la actualidad existen 4 grandes compañías ensambladoras y 92 firmas de autopartes, mientras que entre las empresas comercializadoras e importadoras hay 1.271 empresas dedicadas a la venta de vehículos nuevos y usados y 137 establecimientos de venta de motocicletas.

Hay 81 empresas dedicadas al área de carrocería, al tiempo que aquellas empresas que se dedican a otras actividades de comercio automotor como el mantenimiento, la reparación y la venta de partes, entre otros regiones, suman un total 3.126.

No obstante, la prueba más contundente de fuerza de la industria automotriz en el Ecuador se mide por la cantidad de puestos de trabajo que genera, de acuerdo con cifras de Encuesta Nacional de Empleo, Desempleo y Subempleo.

Las empresas ensambladoras crean 1.534 empleos, mientras que las firmas autopartistas originan 4.710 puestos de trabajo.

Las compañías dedicadas a comercializar e importar generan unos 13.971 empleos, mientras que las empresas de carrocería crean 592 cargos.

Por otro lado, las empresas orientadas a otras actividades de comercio automotor generan alrededor de 35.994 puestos.

Por todo lo expuesto anteriormente, se tiene un total de 56.801 empleos que genera la industria automotriz en el Ecuador.

Todas estas asombrosas y alentadoras estadísticas dejan en claro que la industria automotriz en el Ecuador está indudablemente en un buen momento(Superintendencia de Compañías, 2016).

Una afirmación aún más contundente es el creciente número de dispositivos que se conectan a Internet. Según la compañía de telefonía Huawei, de los siete mil millones de móviles que actualmente están conectados a internet se pasará a cincuenta mil millones de dispositivos en 2023 gracias a nuevas tecnologías y a la conexión a la red de muchos más objetos de la vida cotidiana, como coches, electrodomésticos, relojes, etc.(Formatjé, 2017).

La revolución del Internet de las Cosas no se detiene. En breve se convertirá, si no es que ya es una realidad, en una tecnología imprescindible en ámbitos como la atención sanitaria, el comercio, el transporte e incluso los servicios públicos.

Para 2025, se prevé que haya cerca de 76.000 millones de dispositivos IoT (Internet de las Cosas) en funcionamiento en todo el mundo(Barchilón, 2019).

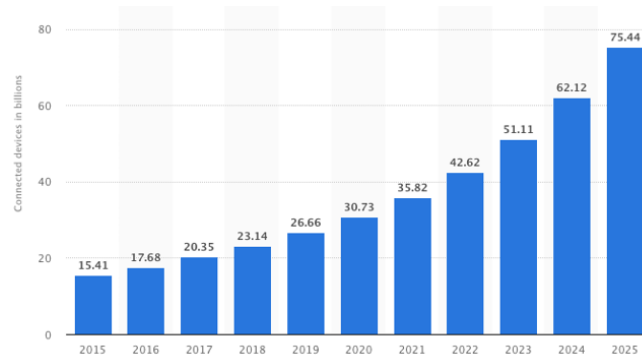


Figura1. Evolución del número de dispositivos conectados al IoT en todo el mundo. (Fernández, 2020)

Internet ha revolucionado la forma tradicional de comunicación persona a persona y con IoT ha revolucionado la forma de interactuar con el entorno, ya que permite la relación digital de objetos cotidianos usados en el día a día y el intercambio de información con otros dispositivos sin intervención humana, dando paso a la comunicación humano-máquina y máquina-máquina adicionando una nueva dimensión al mundo de las comunicaciones, cambiando el conocido modelo de internet. De esta manera es posible capturar grandes cantidades de datos e información facilitando la monitorización, creando experiencias únicas y oportunidades inéditas para personas, empresas y ciudades (Evans, 2011).

Los datos se han convertido en una mina de oro porque permiten a las empresas saber los intereses de sus clientes, optimizar procesos, tendencias a futuro, mejorar la producción, sin mencionar la telemetría de las redes sociales ya que cada minuto se producen 900.000 accesos a Facebook y se suben 46.200 post a Instagram. Cada 24 horas miles de millones de dispositivos e individuos conectados a internet generan 2,5 millones de Terabytes de información según IBM (Alonso, 2019).

Asimismo, la telemetría existe por la necesidad que tiene el ser humano de manipular variables de objetos que se encuentran a una distancia determinada y se puede evidenciar esto en las actividades relacionadas con satélites que toman datos de otro planeta y los envían a la tierra para ser analizados (Gaitán Layza, Mayorga Farfán, Onofre Enero, Reynoso Manrique, & Soto Reyes, 2018).

La problemática está fundamentada en el completo desconocimiento que tiene el conductor sobre el estado actual del vehículo en tiempo real para diagnosticar a tiempo fallas de funcionamiento y predecir futuros daños que pueden ocasionar accidentes fatales antes de que estos sucedan. Datos de la Agencia Nacional de Tránsito (ANT) confirman que en Ecuador los accidentes de tránsito van en aumento. En lo que va del 2018 se han registrado 16.746 siniestros, 2.170 más en comparación con 2017, cuando la cifra fue de 14.322.

Si bien el 97% de los accidentes se dan por fallas humanas (imprudencia e impericia del conductor), los desperfectos mecánicos de los automotores también son causales, señala Guillermo Abad, presidente de la organización Justicia Vial.

El dirigente sostiene que al no existir una herramienta que detecte fallas en el automotor, en generales, el fabricante quien da la guía y la referencia para que el mantenimiento de los vehículos se cumpla en los lapsos que corresponde, tanto preventivo como correctivo. Lo hace a través de un manual “que debe ser manejado por empresas y personas debidamente acreditadas y certificadas. Es decir, talleres y mecánicos autorizados”.

Juan Carlos Mideros es auditor y hace cinco años adquirió un automóvil de segunda mano. Comenta que en el taller de un concesionario un cambio de aceite y una revisión regular bordea los \$ 300, mientras que el mismo trabajo, dice, cuesta \$ 80 donde su mecánico de confianza. El auditor desconoce si el “maestro” que arregla su vehículo lo hace de manera correcta y cuenta con un título académico que certifique sus conocimientos. La única garantía que tiene de que su vehículo está en

buen estado es que es su vecino y que las veces que ha solicitado sus servicios estos no han fallado.

El 97% de personas desconocen si se realizó bien el trabajo a su vehículo, ya que sólo cuentan con la palabra emitida por el mecánico como garantía si quedó bien o mal el carro, y el porcentaje restante corresponde a mecánicos dueños de automóviles que saben lo que hacen y están respaldados por sus conocimientos (Fidalgo, 2017).

Según el administrador del centro de revisión del Valle de los Chillos, todos los técnicos que se encargan de comprobar el buen funcionamiento de los carros son ingenieros mecánicos o especialistas en electrónica y en mecatrónica. Las principales fallas por las que los vehículos no pasan la revisión son la emisión de gases y fallas en el sistema de frenos y luces. Esto se debe, principalmente, a la falta de mantenimiento de los automóviles por el desconocimiento del estado del automotor(Granda, 2018).

Otro problema es satisfacer a tiempo la demanda que exigen los clientes dueños de vehículos livianos en plazos viables, sin retrasos, adoptando las normas, metodología y rediseño de las grandes marcas enfocadas al servicio express para aumentar la velocidad y eficacia en los servicios automotrices (Bustamante, 2018).

Según Ángel Bernal Matute (2013), en el caso de un mantenimiento correctivo, que es un mantenimiento no planificado que se da luego de ocurrir una falla o avería cuando se reparan aquellos componentes que dejaron de funcionar o están dañados; la planificación del tiempo que estará el vehículo fuera de operación no es predecible porque no se tiene un diagnostico fiable de las causas que provocan las fallas, pues se ignora si falló por maltrato, abandono, por desconocimiento del manejo o por desgaste natural. En estas condiciones el técnico pierde tiempo en la ubicación del componente específico y en recibir el mantenimiento correctivo por no tener una historia detallada del vehículo ni un diagnostico general de su estado actual para localizar el problema.

Con respecto a un mantenimiento preventivo, que es un mantenimiento planificado, tiene lugar antes de que ocurra un fallo o avería. Este se efectúa bajo condiciones controladas sin la existencia de algún error en el sistema del vehículo, siguiendo un programa previamente elaborado donde se detalla el procedimiento a seguir y las actividades a realizar, a fin de tener herramientas y repuestos necesarias a la mano. Por lo regular se realiza en 90 minutos a 4 horas como máximo, sin embargo, este debería

ir de la mano con un mantenimiento predictivo para no llegar a un mantenimiento correctivo, pero para esta técnica se necesita en todo instante observar la condición mecánica y eléctrica real del vehículo junto con un historial completo para convertir los mantenimientos correctivos en preventivos de tal manera que se pueda ahorrar tiempo y costos por mantenimiento.

A continuación, se puede observar datos del tiempo y costo aproximado de un mantenimiento preventivo vs correctivo de un automóvil convencional bajo un proceso normal.

En la Tabla 1 se puede evidenciar cuanto tiempo en horas dependiendo de la operación que se realiza en el vehículo, se demora realizar un mantenimiento preventivo frente a un mantenimiento correctivo.

Tabla 1. Tiempo de mantenimiento

Mantenimiento preventivo		Operación	Mantenimiento correctivo	
Trabajando (h)	Parado (h)		Trabajando (h)	Parado (h)
1	1	Desmontaje	4	5
		Examen inicial	0.25	0.25
1.5	1.5	Remolque		3
1.5	1.5	Montaje y prueba	4.5	5.5
		Trabajo externo		
2.5	2.5	Tiempo total	8.75	13.75

(Matute, 2013)

En la Tabla 2 se puede evidenciar cuánto dinero en dólares dependiendo del detalle de cada vehículo, cuesta realizar un mantenimiento preventivo frente a un mantenimiento correctivo.

Tabla 2. Costo de mantenimiento

Costo de Mantenimiento Preventivo (\$)	Detalle	Costo de Mantenimiento Correctivo (\$)
30	Costo de trabajo	60
	Costo de trabajo externo	50
50	Repuestos	130
5.20	Costo de vehículo parado	28.65
85.80	Costo total	268.65

(Matute, 2013)

Por otra parte el hurto o pérdida de vehículos que no logran ser rastreados o localizados y desaparecen sin tener la posibilidad de poderlos recuperar o saber su paradero para buscarlos y dar con su ubicación exacta es sumamente alta en comparación con los autos recuperados(Lozano Rodriguez, Marín Castañeda, & Rodriguez Campos, 2015).

Es importante señalar que entre el 1 de enero de 2015 hasta el 16 de julio de 2017, la Policía Judicial registra en sus informes el robo de 13.271 automotores y se recuperaron 2.363. Además, el hurto de 15.966 motos y recuperación de 1.032, solo en 2015 y 2016. Las estadísticas muestran que en los 2 últimos años los delincuentes se llevaron 26.336 partes y accesorios de vehículos, los cuales tienen su mercado ilegal, mayoritariamente, en ciudades grandes como Quito y Guayaquil (TELÉGRAFO, 2017).

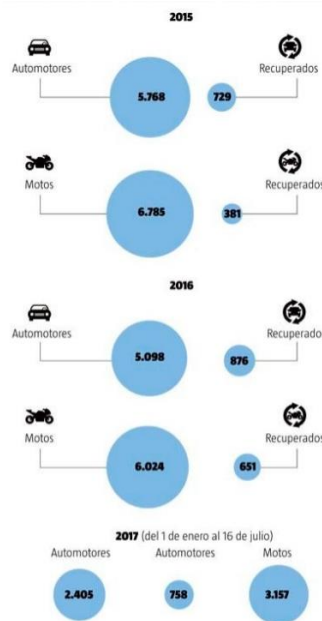


Figura 2. El robo de vehículos en Ecuador. (Judicial, 2017)

Finalmente, los datos de la telemetría vehicular en el Ecuador son muy limitados, cuenta Andrea Acosta (2019), directora del producto AutoConectado de Seguros Equinoccial, en 2016 junto con CarSync e IBM desarrollaron un producto enfocado en la información de sus clientes a fin de premiar los buenos hábitos de conducción. En el 2017 realizaron cambios internos y trabajo con proveedores externos a fin de desarrollar Driving Score, App y sistemas operativos internos, para mayo 2018 AutoConectado sale al mercado ecuatoriano, como el primer producto de su categoría, en el 2019 tienen ventas sostenidas, análisis de datos de

14000 vehículos, desarrollan un modelo de riesgo, mejoran la experiencia de usuario, mejoran el Driving Score, implementan CrashAlert, nueva App y para el 2020 realizan una oferta simplificada basada en análisis de datos y entendimiento del cliente. Andrea comenta que existen varios problemas en este proceso, el dispositivo que usan para tomar la data de los vehículos de sus clientes es invasivo, toma aproximadamente unas cuatro horas instalarlo ya que lo insertan en el tablero y esto tiene un riesgo para el automóvil que pierde la garantía por parte de la casa automotriz si es manipulado por terceras personas; por otra parte para instalar el dispositivo necesitan una considerable logística y talleres donde realizar este trabajo elevando los precios de comercialización de las primas, presentando un coste mucho más alto frente a la competencia, así mismo los datos que se toman de los vehículos pertenecen a Seguros Equinoccial pero necesitan pedir a IBM la autorización de poderlos utilizar y cuando necesitan algún reporte reciben la información en tablas de Excel y después de varios días, lo que hace nada productivo el proceso de analítica, IBM trabaja con una arquitectura de datos muy cerrada que no permite que otros proveedores puedan participar en el tratamiento de los datos para crear nuevos modelos de analítica, usuarios arquetipo, geofencing y alianzas con establecimiento al identificar rutas frecuentes de clientes y otros tipos de análisis menciona Andrea Acosta.



Figura 3. AutoConectado, la transformación para innovar en seguros vehiculares. (Acosta, 2019)

La tecnología está cambiando rápidamente para todas las industrias y el sector automotriz no es la excepción ya que existen nuevos requerimientos del consumidor, haciendo que el acceso a la web y a muchas otras aplicaciones en la nube pueda llevarse a cabo desde cualquier parte del mundo, por lo que los fabricantes necesitan extender la conectividad IoT a su oferta (Ramírez & Guzmán Ruiz, 2016).

El presente proyecto tiene como finalidad desarrollar un dispositivo tecnológico que satisfaga estas necesidades, adaptando esta nueva tecnología IoT a los vehículos de manera fácil, sencilla y no invasiva, por

medio de una puerta de enlace que se conecte al puerto OBD (Diagnostico a Bordo) del vehículo tomando la telemetría de la ECU automotriz y enviando esta información a un servidor de análisis de datos que se encargue del manejo de cada una de las variables para brindar resultados relevantes, seguros y en tiempo real sobre el estado, monitoreo y control del vehículo.

Esta nueva herramienta estará ahora al alcance de las manos del usuario quien ya no sólo dependerá del manual, la guía y referencias que le da el fabricante del automotor para que talleres, empresas o mecánicos autorizados sean los encargados de un mantenimiento adecuado y oportuno, sino el mismo conductor tendrá la oportunidad de participar activamente en el cuidado y protección de su vehículo con datos reales que notifiquen el momento indicado cuando el automóvil necesita un mantenimiento tanto predictivo como correctivo.

Por otra parte, se disminuirá considerablemente la incertidumbre que tienen los usuarios vehiculares cuando el carro sale del taller tanto si es del propio concesionario como de un genérico, ya que podrán tener un contacto cercano actualizado con todas las variables que intervienen en un correcto mantenimiento recibido por parte de los mecánicos, advirtiéndolo de esta manera si el servicio recibido corrigió o no los desperfectos electrónicos y mecánicos por el cual el vehículo llegó al taller.

Así mismo, gracias a los beneficios de este dispositivo, se disminuirá considerablemente la tasa de mortalidad actual a causa de accidentes de tránsito por desperfectos mecánicos porque cada conductor será consciente del estado de su carro para tomar decisiones correctivas y preventivas que mejoren la salud vehicular de la ciudad.

También este dispositivo servirá como una herramienta de mantenimiento predictivo y proactivo para reducir considerablemente en un 70% la cantidad de vehículos que pasan por un mantenimiento correctivo, apalancando este porcentaje a un mantenimiento preventivo que como se evidenció en la Tabla 1 y Tabla 2 ahorra mucho más tiempo y dinero al conductor; además de extender la vida útil del automotor utilizando una práctica de detección y corrección de las desviaciones justo a tiempo, según el programa de mantenimiento proactivo asignado y verificando que el automóvil se encuentre dentro de los rangos de funcionamiento normal para tomar acciones que permitan al equipo regresar a las condiciones establecidas para desempeñarse adecuadamente por más tiempo.

Adicionalmente la puerta de enlace ofrece al dueño del vehículo una geolocalización exacta de dónde se encuentra su carro en todo momento, incrementando las probabilidades de rastrear, localizar y encontrar un vehículo perdido o robado reduciendo de esta forma las cifras exponenciales que se reflejan en la Figura 2 de carros no recuperados para brindar una mayor esperanza de tener de vuelta el automotor por medio de esta tecnología.

Por último el proyecto catapulta la investigación e innovación en el país para desarrollar nuevas ideas de negocio enfocadas en el estudio, análisis y control de la telemetría automotriz que den un valor agregado a esta industria produciendo más fuentes de trabajo y nuevas áreas del conocimiento para el desarrollo y crecimiento del mercado laboral.

En Guayas provincia del Ecuador ya se está usando la tecnología IoT en favor de las personas no videntes. Pérez Andrés(Fernando, 2019)diseñó un chaleco con sensores ultrasónicos el cual permite a las persona con capacidades visuales limitadas sentir una vibración poco intensa a muy intensa dependiendo de qué tan cerca o lejos se encuentra de un objeto para evitar que colisionen y sufran accidentes al movilizarse a través de las calles de Guayaquil; además, si sufren un accidente por medio de un botón de pánico pueden enviar una alerta a sus seres queridos con las dirección exacta donde se encuentran para que reciban ayuda inmediata, facilitando la localización, tranquilidad y confianza tanto del usuario como sus familiares al transportarse de manera autónoma de un lugar a otro. Para elaborar este dispositivo se usó como materiales una placa de Arduino uno, módulo sensor ultrasónico Hc-sr04, Infrarrojo, módulo SIM900, módulo chip GPRS GSM900, módulo GPS NEO-6m/v2, módulo L293, motor de vibración y baterías recargables.

Como resultado los usuarios del chaleco pueden anticipar los obstáculos en un rango determinado a una distancia de 1 metro como máximo y un mínimo de 20 centímetros y en caso de algún percance al pulsar el botón de pánico se envían mensajes y llamadas con las coordenadas del siniestro con un margen de error de 4 metros en un tiempo máximo de 10 segundos después de la señal de alerta.

Otro caso que mencionan Javier García y Lluís Gifre (García Fernández & Gifre Renom, 2019) es el desarrollo de un dispositivo IoT para la adquisición de medidas ambientales para el control y monitoreo de determinadas áreas naturales que necesitan un cuidado especial y vigilancia constante con el fin de analizar cada una de las variables de un entorno y estudiarlas para pronosticar futuras afecciones al medio

ambiente y controlar este problema antes de que suscite, usando un protocolo de comunicación MQTT (Message Queuing Telemetry Transport) el cual no necesita una conexión continua a la red tomando variables como presión, altitud, humedad, temperatura, luz ambiental, aceleración, rotación y balanceo del acelerómetro utilizando como hardware un Wipy de Pycom, Pysense de Pycom, sensores de luz, humedad, temperatura, presión, altura, acelerómetro, una Raspberry Pi 3b y como software Grafana con InfluxDB obteniendo como resultado el desarrollo de un software para dispositivos IoT, que obtuviera medidas del entorno y, además, permitiera la conexión de dispositivos a la red de forma exitosa, empezando por la detección de la activación de dispositivos, continuando con el envío de muestras tomadas por los sensores del dispositivo hacia la aplicación a petición de esta última, y el almacenamiento y representación gráfica de dichas muestras.

Adicionalmente comentan Jorge E. Gómez y sus colaboradores (E Gomez, Castaño, Mercado, Fernandez, & Garcia, 2017) sobre el dispositivo IoT creado en el 2017 para monitorear cultivos protegidos como invernaderos, se trata de un artefacto que toma variables como el contenido volumétrico de agua en el suelo, temperatura y humedad relativa así como también la radiación con el objetivo de recolectar datos que permitan en tiempo real verificar su estado y a la vez pueda generar alertas al usuario que lleve a una adecuada toma de decisiones o acciones automatizadas como suministro de agua al cultivo o algún tipo de control ambiental, este proceso lo realizan en cuatro capas, la primera es la que escucha permanentemente los datos que provienen de cada sensor por medio del protocolo MQTT, la segunda se encarga de obtener las peticiones del cliente desde la web móvil y muestra la respuesta a los clientes mediante HTTP, la tercera esta encargada de la suscripción al bróker por un cliente MQTT el cual recibe los datos leídos y se conecta con la base de datos para su almacenamiento y por último la cuarta capa está diseñada para recuperar y entregar información, dependiendo de las solicitudes y respuestas de los usuarios, así como de los sensores. Los resultados obtenidos fueron los parámetros en tiempo real a través del sistema que es mostrado en la web site o en su móvil, accediendo a los diferentes métodos elegidos para visualizar y analizar la información en tablas con los datos de los parámetros o gráficos para el claro análisis de estos.

En Ecuador, según Arbaiza (Arbaiza, 2010), el uso de datos a través de telemetría se presencia en CNEL Manabí que por medio de medidores de energía con una comunicación GPRS se puede obtener una información

veraz y en tiempo real de variables como voltaje, energía reactiva, energía activa y corriente, logrando un control más eficiente de los clientes.

Para continuar con los avances y por las razones publicadas anteriormente el objetivo principal de la investigación es:

- Diseñar e implementar un dispositivo IoT (Internet de las Cosas) en el vehículo, que sirva como puerta de enlace para la transmisión de la telemetría de la ECU (Unidad de Control del Motor) automotriz, hacia un servidor de análisis de datos para la empresa INGENESIS.

Y los objetivos específicos de la investigación son:

- Desarrollar e implementar el hardware de control del dispositivo de puerta de enlace.
- Desarrollar y programar el software de control del dispositivo de puerta de enlace.
- Desarrollar e implementar el protocolo de comunicación del vehículo al dispositivo de puerta de enlace y del dispositivo de puerta de enlace al servidor de análisis de datos.
- Comprobar y asegurar la veracidad de la información transmitida al servidor de análisis de datos.
- Desarrollar e implementar respaldos para el dispositivo de puerta de enlace y la información transmitida por el mismo, como son batería y memoria de almacenamiento interno.

2. METODOLOGÍA

Se utilizó la metodología de diseño mecatrónico en V, solicitando en primera instancia los requerimientos y necesidades que tiene el cliente para despejar las dudas y preocupaciones con propuestas que se adapten a brindar una solución efectiva al problema que se propone, en este caso, obtener todas las variables de un automóvil para brindar al usuario información relevante que ayude al cuidado y mantenimiento del vehículo a través del análisis de datos.

Una vez que se tuvo claro lo que necesita el cliente se procedió a diseñar el sistema general del dispositivo, creando un panorama amplio de cada una de las partes que se va a necesitar desarrollar para cumplir con cada uno de los objetivos propuestos por el cliente.

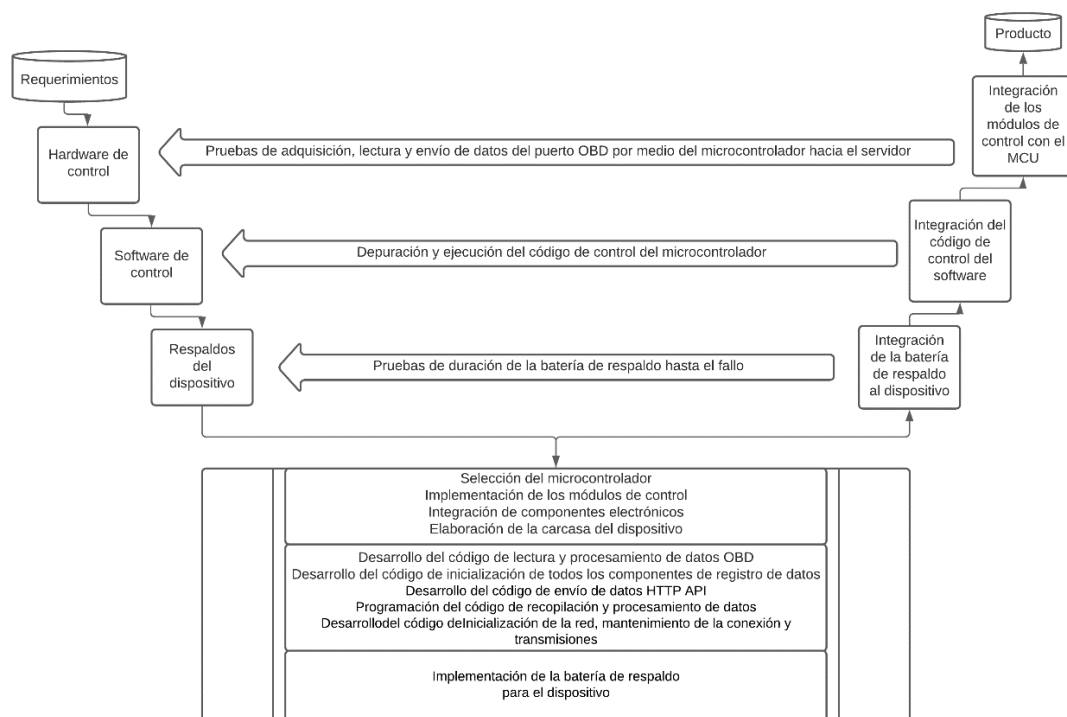


Figura 4. Metodología de diseño en V para el diseño de sistemas mecatrónicos. (Deutscher, 2004)

Una vez que se ha trazado el mapa general de todo lo que se necesita para elaborar el dispositivo se enfoca el desarrollo en el diseño específico de cada uno de los elementos que conforman el dispositivo completo dividiendo en tres ramas las cuales son el hardware de control, el software de control y los respaldos del dispositivo, a continuación se realizó la integración de todos los componentes electrónicos en un solo circuito compacto que cumpla con todas las funciones determinadas necesarias para satisfacer los requerimientos propuestos al comenzar el proyecto, este ciclo en V está en una constante

mejora continua para garantizar las propiedades del dispositivo permitiéndole de esta manera mejorar y escalar sus funciones en el transcurso del tiempo.

2.1 REQUERIMIENTOS.

Tipo de conexión: Plug and Play.

Datos de lectura: Aceleración, velocidad, voltaje de la batería, carga del motor, geolocalización.

Dimensiones del recinto: 60x48x20 mm.

Forma de comunicación: WI-FI, Bluetooth y red celular 4G.

Almacenamiento de datos: Memoria flash interno y externo en tarjeta micro SD.

Actualización de geolocalización: GPS/GLONASS.

Protocolo de acceso a la información: PID OBD-II estándar, DTC, VIN desde la ECU del vehículo.

Protocolo de rastreo de la información: Rastreo de datos del bus CAN.

Grados de libertad de los sensores de movimiento: 9-DOF

Tipo de seguridad de transmisión de datos: Transporte SSL/TLS y autenticación por usuario y contraseña o mediante certificado.

Batería de respaldo: Duración mínima 15 minutos después del fallo.

2.2 DISEÑO DEL SISTEMA.

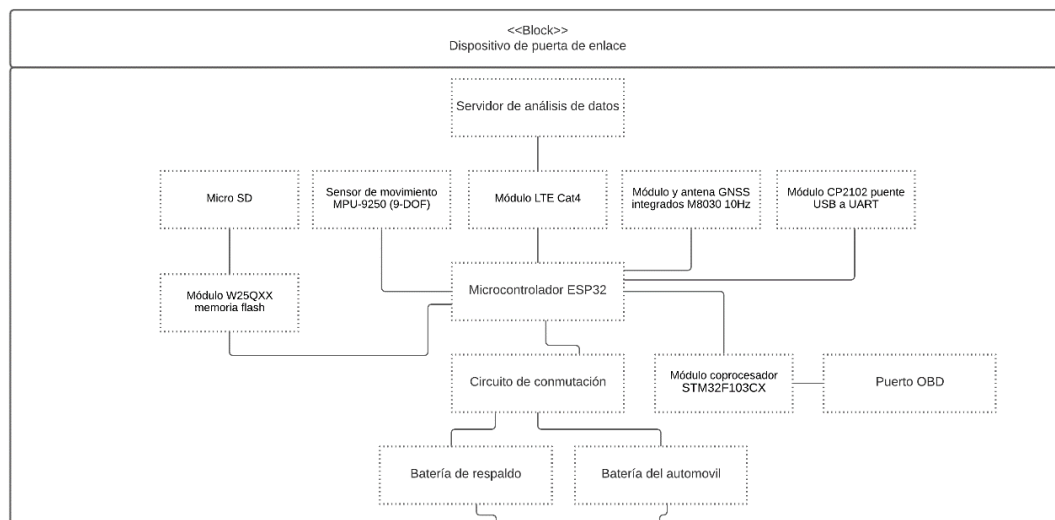


Figura 5. Diagrama de bloques de conexión general de todos los elementos.

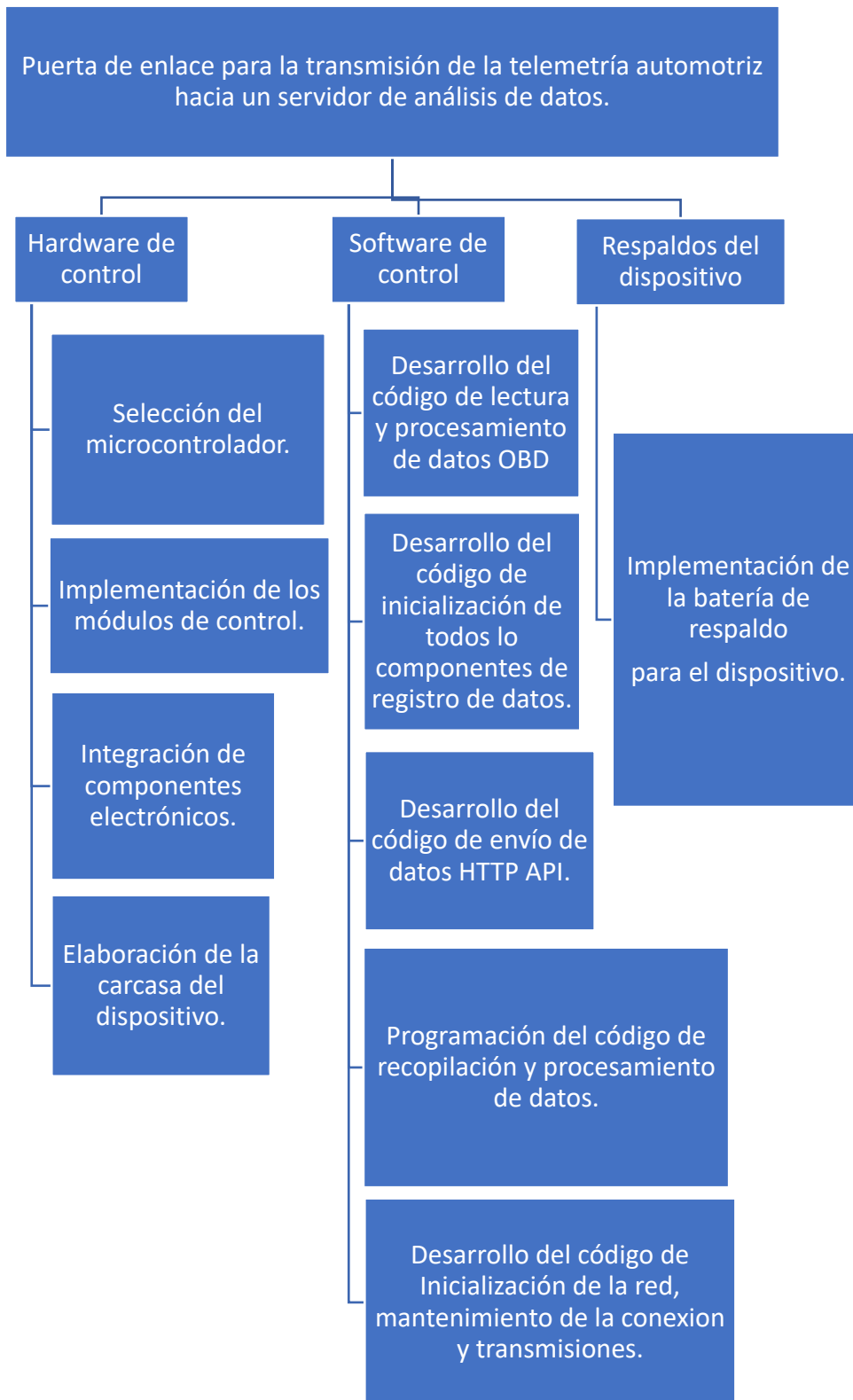


Figura 6. Flujoograma del diseño del dispositivo de puerta de enlace.

En la figura 5 y 6 se puede observar el diagrama de conexión de bloques junto con el flujograma del diseño del dispositivo de puerta de enlace los cuales permiten presentar un panorama general de todos los elementos y como están interconectados entre sí.

2.3 DISEÑO ESPECÍFICO.

2.3.1 HARDWARE DE CONTROL.

2.3.1.1 Selección del microcontrolador.

Se buscó un microcontrolador que trabaje con el IDE de Arduino el cual se le pueda acoplar un sistema OBD-II para poder recopilar la información del Bus Can del vehículo por medio de la Ecu automotriz.

Se eligió el microcontrolador que más se adapte al dispositivo a desarrollar basado en siete características fundamentales que son necesarias para cumplir con los objetivos propuestos, estas son:

Bits: se requirió un espacio de direcciones para el bus de datos y el bus de instrucciones por lo que se necesita como mínimo 32 bits.

RAM: se necesitó mínimo 8 MB de memoria RAM porque va a realizar varias acciones al mismo tiempo.

Flash: la mayoría de los microcontroladores no cuentan con mucho espacio por lo que independientemente de la memoria interna se va a necesitar una memoria adicional externa, por lo tanto, con que cuente mínimo con 16 MB de almacenamiento para mantener procesos en tiempo real es suficiente.

GPIO: los pines de entrada y salida de propósito general para abordar a cada uno de los módulos que se utilizó deben ser de al menos 30 pines físicos.

Conectividad: el dispositivo se basa en una solución IoT por lo que necesariamente debe contar con conexión a Internet por lo menos a través de Wi-Fi y tener la opción de acoplar un módulo GSM para conexión portátil a internet.

Consumo de energía: tiene varios sensores conectados y debe depender de una batería de respaldo en el caso que falle la fuente de alimentación principal del carro, por lo que en un modo de bajo consumo debe trabajar a 20mA y como máximo a 160mA, este rango permite que dure un tiempo prudente el dispositivo conectado a la batería de respaldo antes del fallo.

Herramientas de desarrollo y comunidad: contar con un conjunto de herramientas maduras que cuenten con soporte y documentación comunitario para poder desarrollar códigos personalizados que se ejecute en el

microcontrolador, Arduino cumple con este objetivo sin embargo TI, Samsung, Raspberry Pi son buenos tableros de desarrollo, pero se debe elegir el más adecuado dependiendo de los factores que intervienen en el dispositivo.

Por las características mencionadas anteriormente se eligió el microcontrolador ESP32 de la empresa Espressif que proporciona un manual de referencias técnicas muy completo para la integración con otros módulos y un puerto macho OBD-II, las especificaciones del MCU (microcontrolador) se presentan en la siguiente tabla:

Tabla 3. Comparaciones del microcontrolador Espressif ESP32 frente al ESP8266 y NodeMCU

Especificaciones	Espressif ESP32	ESP8266	NodeMCU
Espacio de direcciones	4 GB (32-bit)	4 GB (32-bit)	4 GB (32-bit)
Memoria flash SPI	Admite hasta 16 MB	Admite hasta 16 MB	Admite hasta 4 MB
Memoria SRAM SPI	Admite hasta 8 MB	Admite hasta 50 kB	Admite hasta 32 kB
Pines de E/S de uso general	34 pads GPIO físicos	Tiene 17 puertos GPIO, pero solo se pueden usar 9 o 10.	9 pines GPIO con I2C y SPI
Conectividad	Transceptor Wi-Fi 802.11 b / g / n HT40 incorporado y Bluetooth de modo dual integrado (clásico y BLE)	Transceptor Wi-Fi 802.11 b / g / n	Transceptor Wi-Fi 802.11 b / g / n
Potencia nominal típica a 80Mhz	20mA (WIFI inactivo) / 50mA (WIFI activo) / 150mA (LTE activo)	0,5mA (dispositivo apagado) / 170mA (cuando transmitimos a tope la señal)	25mA (WIFI inactivo) / 75mA (WIFI activo) / 195mA (LTE activo)
Frecuencia de muestreo	60 Hz superior	de 8 Hz a 860 Hz	8.928 KHz
Numero de núcleos del procesador	doble	Único	Único
Capacidad de memoria externa	Hasta 64 Gb	Hasta 32 Gb	Hasta 64 Gb

(Systems, 2020)

En la siguiente figura se muestra un esquema del mapa de los pines del Microcontrolador ESP32:

- 5) Conexión a tierra del equipo.
- 6) Protocolo CAN High J-2284.
- 7) Protocolo ISO 9141-2.
- 8) Libre para el fabricante.
- 9) Libre para el fabricante.
- 10) Protocolo SAE-J1850 Bus-.
- 11) Libre para el fabricante.
- 12) Libre para el fabricante.
- 13) Libre para el fabricante.
- 14) Protocolo CAN Low J-2284.
- 15) Protocolo ISO 9141-2.
- 16) Batería+.

Los datos que se obtienen de este sistema OBD-II y la ECU se presentan de la siguiente manera:

Parámetros de Identificación (PID): Son la representación de forma analógica o digital de la información en vivo que se reciben de las entradas y salidas a la ECU automotriz.

Códigos de fallas almacenados: Esta opción permite adquirir todos los DTC (Data Trouble Code) códigos de fallo recopilados en la ECU del vehículo.

Eliminación de códigos de falla almacenados: Permite eliminar todos los códigos de error almacenados.

Pruebas a bordo: Esta opción permite observar los resultados de todas las pruebas realizadas mientras el automóvil se encuentra en funcionamiento.

2.3.1.2.2 Módulo LTE Cat4.

Este módulo admite varios modos de comunicación inalámbrica de LTE-TDD / LTE-FDD / HSPA + / GSM / GPRS / EDGE, etc. Además, tiene una velocidad de enlace descendente máxima de 150 Mbps y una velocidad de enlace ascendente de 50 Mbps, además permite una migración sin problemas de tecnología 3G a LTE facilitando el diseño de dispositivos más compatibles según las necesidades del cliente.

En la Figura 9 se observa las dimensiones físicas del módulo celular SIM7600E-H 4G LTE CAT4 que se incorporará al microcontrolador ESP32.

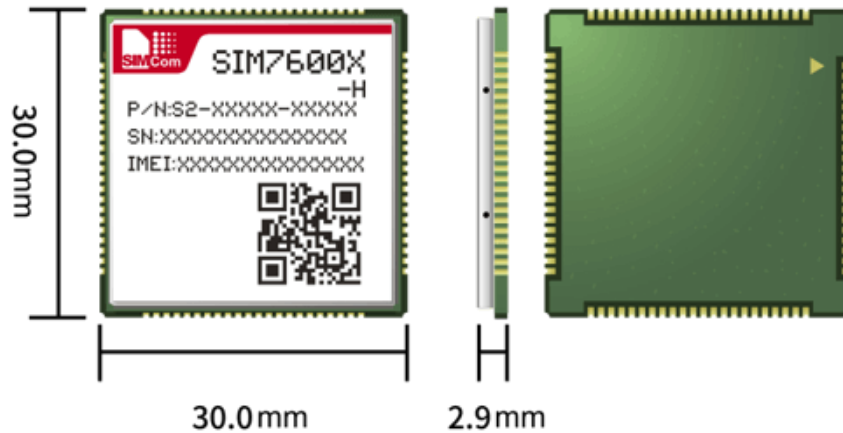


Figura 9. Dimensiones físicas del módulo SIM7600E-H.
(Sim, 2018)

Esta serie SIM7600E-H relaciona varios sistemas GNSS de posicionamiento satelital de alta precisión con múltiples protocolos de red, admitiendo controladores para los principales sistemas operativos (Windows, Linux, Android entre otros) además integra las principales interfaces estándar industriales como UART, USB, I2C, GPIO siendo una solución adecuada para las principales aplicaciones IoT en telemetría, telemática, dispositivos de vigilancia, enrutadores industriales entre otros.

A continuación, en la Tabla 4 se detalla las especificaciones del módulo LTE Cat4 SIM7600E-H con todas las características físicas, bandas de frecuencia, características eléctricas, transferencia de datos, funciones de software, e interfaces, con todas estas especificaciones se procura que no exista ningún problema al momento de integrar el módulo LTE Cat4 con el microcontrolador ESP32, la ventaja de este modulo es que funciona para transferir lo datos en diferentes bandas permitiendo de esta manera que no exista limitaciones en la cobertura y tecnologías de las diferentes operadoras de celular que operan en el Ecuador.

Tabla 4. Especificaciones del módulo LTE Cat4 SIM7600E-H

Módulo LTE Cat4 SIM7600E-H	
Características físicas	
Ítems	Detalle
Factor de forma	LCC, 87PIN
Dimensiones	30.0*30.0*2.9
Bandas de frecuencia	
Ítems	Detalle
Bandas de frecuencia LTE-FDD	B1/B3/B5/B7/B8/B20
Bandas de frecuencia LTE-TDD	B38/B40/B41
Bandas de frecuencia WCDMA	B1/B5/B8
Bandas de frecuencia GSM	900/1800MHz
Temperatura de funcionamiento	-40°C ~ +85°C
Características eléctricas	
Ítems	Detalle
Voltaje de suministro	3.4~ 4.2
Transferencia de datos	
Ítems	Detalle
LTE(Mbps)	150(DL)/50(UL)
HSPA+(Mbps)	42(DL)/5.76(UL)
WCDMA(Kbps)	384(DL)/384(UL)
GPRS/EDGE(Kbps)	236.8(DL)/236.8(UL)
Funciones de software	
Ítems	Detalle
Protocolo	TCP/IP/IPV4/IPV6/ Multi-PDP/FTP/FTPS/ HTTP/HTTPS/DNS
Android RIL	Android 5.0/6.0/ 7.0/8.0/9.0
USB Driver	Microsoft Windows 7/8/10 Linux/Android
MBIM	Win8/Win10
NDIS	Linux/Windows 7/8/10
Firmware Upgrade	USB/FOTA
Interfaces	
Ítems	Detalle
SIM Card	1.8V/3.0V

(Sim, 2018)

2.3.1.2.3 Sensor de movimiento MPU-9250 (9-DOF)

Este módulo es multi-chip (MCM) compuesto por un giroscopio de 3 ejes, un acelerómetro de 3 ejes, un magnetómetro de 3 ejes AK8963 y de esta manera se convierte en un dispositivo de seguimiento de movimiento de 9 ejes con un procesador de movimiento digital (DMP), también cuenta con un sensor de temperatura integrado con y un regulador de voltaje de 3.3V incorporado.

La Figura 10 permite ver el sensor de movimiento MPU-9250 integrado (9-DOF)

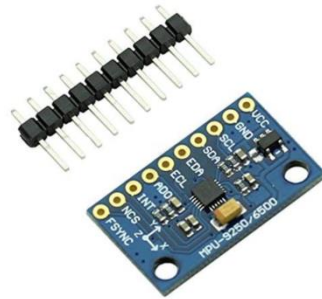


Figura 10. Diseño físico del Sensor de movimiento MPU-9250 (9-DOF)
(Amazon, 2020)

A continuación, en la Tabla 5 se puede observar las especificaciones técnicas por las cuales se eligió el sensor de movimiento MPU-9250 para acoplarlo al microcontrolador ESP32, el cual brinda los datos de seguimiento de movimiento del vehículo al dispositivo de puerta de enlace y receptara también esta información para enviar al servidor web donde se analizarán los datos de movimiento del vehículo.

Tabla 5. Especificaciones técnicas del Sensor de movimiento MPU-9250 (9-DOF)

Sensor de movimiento MPU-9250 (9-DOF)	
Acelerómetro de 3 ejes.	
Ítems	Detalle
Rango	hasta ± 0.56 oz
Sensibilidad	hasta 16.384 LSB/g
Giroscopio de 3 ejes.	
Ítems	Detalle
Rango	hasta ± 70.55 oz/seg
Sensibilidad	hasta 131 LSB/deg/seg
Magnetómetro de 3 ejes	
Ítems	Detalle
Rango	$\pm 4800 \mu\text{T}$
Sensibilidad	0,6 $\mu\text{T/LSB}$
Características del dispositivo	
Ítems	Detalle
Voltaje de alimentación:	4,4 a 6,5 V o 3,3 V
Interfaz	I2C
Dimensiones de la tabla	1.004 in de largo x 0.606 in de ancho, 0.118 in de diámetro interior de agujeros de montaje
Peso	2,72 g (0,096 oz)

(Amazon, 2020)

2.3.1.2.4 Módulo coprocesador STM32F103CX

Este módulo permite conectar al bus CAN del automóvil para leer los datos que recopila la ECU automotriz, este MCU es de 32 bits basado en ARM de línea de rendimiento de alta densidad con 256 a 512 KB flash, USB, CAN, 11 temporizadores, 3ADC, 13 interfaces de comunicación, a continuación en la Tabla 6 se especifica las características de cada una de las partes que lo conforman para tener un mejor conocimiento de su función y la manera en la que se debe conectar al ESP32 que es el microcontrolador principal.

Tabla 6. Características técnicas del Módulo coprocesador STM32F103CX

Módulo coprocesador STM32F103CX	
Núcleo: CPU ARM Cortex de 32 bits	
Ítems	Detalle
Frecuencia máxima	72 MHz
Multiplicación y hardware	Ciclo único división
Recuerdos	
ítems	Detalle
Memoria Flash	256 a 512 KB
Memoria SRAM	64KB
Gestión de reloj, reinicio y suministro	
ítems	Detalle
Oscilador de cristal	4 a 16 MHz
Suministro de aplicaciones	De 2,0 a 3,6 V
Convertidores A/D de 3 x 12 bits	
ítems	Detalle
Rango de conversión: 0 a 3,6V	4,4 a 6,5 V o 3,3 V
Capacidad de retención	Muestro triple
Temporizadores básicos	2x16 bits para controlar el DAC
Interfaz CAN	2.0B activo

(ALLDATASHEET.COM, 2018)

2.3.1.2.5 Módulo y antena GNSS integrados M8030 10Hz

La solución M8030 10Hz GNSS con antena de cerámica activa está integrada y administrada por un coprocesador que también decodifica el flujo NMEA y hace que la información decodificada esté lista para que ESP32 la obtenga.

En la Figura 11 se puede apreciar la solución M8030 10Hz GNSS con antena de cerámica activa y el coprocesador STM32F103CX.

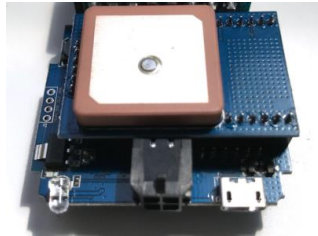


Figura 11. Diseño esquemático del coprocesador STM32F103CX.
(Amazon, 2020)

Este módulo permite tomar la posición exacta donde se encuentra el automóvil mediante una técnica de triangulación con una precisión aproximada de cinco metros y enviar esta información al MCU ESP32 para conectarse al servidor de análisis de datos por medio del módulo GSM que permite enlazar al internet desde cualquier lugar donde se encuentre el vehículo, la antena de cerámica activa permite tener un mayor alcance para que la precisión del dispositivo sea mayor y como ventaja proporcione la integración dentro del mismo dispositivo sin la necesidad de colocarlo fuera de él o en otro lugar del automóvil, facilitando de esta manera la compactación del sistema en una sola caja donde interactúen todos los componentes desde un solo lugar, en la Tabla 7 se detalla las especificaciones técnicas del Módulo y antena GNSS integrados M8030 10Hz.

Tabla 7. Características técnicas del Módulo y antena GNSS integrados M8030 10Hz

Módulo y antena GNSS integrados M8030 10Hz	
Núcleo: CPU ARM Cortex de 32 bits	
Parámetro	Especificaciones
Chipset	u-blox M8030-KT
Formato de recepción	GPS, GLONASS, Galileo, BeiDou, QZSS y SBAS
Frecuencia	GPS L1, GLONASS L1, BeiDou B1, SBAS L1, Galileo E1
Canales	72 canales de búsqueda
Precisión	6.6 ft CEP 2D RMS SBAS Enable (típico cielo abierto)
Velocidad	0.3 ft/seg 95% (desconexión de SA)
Tasa de datos y actualización	4800 bps a 9600 bps, por defecto 9600 bps.
Nivel de datos	TTL o RS-232, Nivel de TTL por defecto
Protocolo de datos	NMEA-0183 o UBX, por defecto NMEA-0183
Límites operacionales de altitud	50 0.0 ft máx.
Límites operacionales de velocidad	1,689.6 ft/s máx.
Consumo de energía	Voltaje DC: 3,0 V-5,5 V, típico: 5,0 V.

Continúa...

Continuación...

	70 mA/5,0 V.
Dimensiones	1.102 x 1.102 x 0.394 in.
Peso	0.42 oz.
Temperatura de funcionamiento	-40.0 °F ~ +185.0 °F

(Amazon, 2020)

2.3.1.2.6 Módulo CP2102 puente USB a UART

Este conversor permite al microcontrolador ESP32 y una PC comunicarse utilizando el protocolo USB de forma sencilla para facilitar la integración de nuestros proyectos con programas de PC como Matlab, Labview y otros, de esta manera el dispositivo puede interactuar directamente con un ordenador para realizar pruebas y verificar los datos de salida que transmite la ECU del vehículo.

La Figura 12 muestra el modelo físico del CP2102 con cada uno de los pines que se necesita para acoplar al MCU esta función y facilitar la conexión entre los componentes que forman parte del dispositivo de puerta de enlace.



Figura 12. Diseño esquemático del Módulo CP2102 puente USB a UART.
(LABS, 2018)

En la Tabla 8 se puede observar las especificaciones y características técnicas del CP2102 para una correcta instalación y puesta en marcha del dispositivo.

Tabla 8. Características técnicas del Módulo CP2102

Módulo CP2102 puente USB a UART	
Parámetro	Especificaciones
Velocidad	USB 2.0 de velocidad completa de 12 Mbps
Tipo de conexión	Micro USB tipo B (smartphone)
Pines salida (TTL)	+3.3V, RST, TXD, RXD, GND y + 5V
Cristal oscilador	Integrado
Regulador de voltaje	3.3V interno
Buffer de recepción	576 Bytes
Buffer de transmisión	640 Bytes

Continuación...

Continúa...

Temperatura de trabajo	-40° a 80°C
Sistemas Operativos soportados	Windows 10, 8, Vista, 7, XP, 2000, 98SE y Linux 2.40(en adelante)
Dimensiones	21 mm x 16 mm

(LABS, 2018)

2.3.1.2.7 Módulo W25QXX memoria flash

Es un módulo dataflash serie con una interfaz de SPI a bordo, funciona como una memoria en serie o Flash del dispositivo que soporta el estándar/Dual/Quad SPI & QPI, a continuación, se describe las especificaciones y características técnicas de este módulo en la siguiente tabla:

Tabla 9. Características técnicas del Módulo W25QXX memoria flash

Módulo W25QXX memoria flash	
Parámetro	Especificaciones
Capacidad	128m-bit/16m-byte
Frecuencia máxima	104 MHz
Tensión de funcionamiento	2.7 ~ 3.6 V
Temperatura de trabajo	-40 °C ~ + 85 °C
Dimensión	20.6mm x 38.9mm

(ALLDATASHEET.COM, 2018)

El módulo W25QXX memoria flash es un respaldo más para el dispositivo ya que si la microSD falla o la memoria flash del microcontrolador tiene algún error esta memoria almacena los datos y los mantiene hasta que se recupere nuevamente alguna de las memorias con las que cuenta la puerta de enlace, en la siguiente Figura 13 se puede apreciar el diseño físico del módulo W25QXX con la descripción necesaria en cada uno de los pines para realizar una correcta instalación e integración.

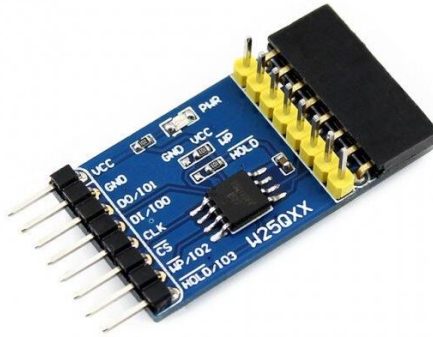


Figura 13. Diseño esquemático del Módulo W25QXX memoria flash.
(ALLDATASHEET.COM, 2018)

2.4 INTEGRACIÓN DE COMPONENTES ELECTRÓNICOS

Cada uno de los elementos que se describieron anteriormente fueron integrados y compactados de tal manera que el dispositivo cumpla con los requerimientos solicitados por las partes interesadas, más adelante se puede observar el diagrama de conexión de cada uno de los módulos mencionados junto con el microcontrolador y cada una de las partes que conforman el dispositivo de puerta de enlace OBD-II

Primero que nada, se presenta brevemente las interfaces físicas con las que cuenta el dispositivo de puerta de enlace:

- Conector macho OBD-II (16 pines)
- Toma GPIO (4 pines)
- ranura para tarjeta microSD
- Ranura para tarjeta micro SIM
- Puerto micro USB
- xBee Socket
- Indicador LED

Ahora una corta descripción de cada una de las interfaces físicas del dispositivo junto con el mapeo de los pines.

2.4.1 CONECTOR MACHO OBD-II

El conector macho OBD-II permite que el dispositivo de puerta de enlace se conecte directamente al puerto hembra OBD-II que está disponible en la

mayoría de los automóviles actuales. El puerto OBD-II brinda una conexión entre el dispositivo de puerta de enlace y la ECU automotriz, así como la fuente de alimentación que es la batería del vehículo.

Toma GPIO

El dispositivo de puerta de enlace cuenta con una toma de pines de entrada y salida de 4 pines que contiene 2x GPIO (del ESP32), VCC y GND. La definición de línea del conector es la siguiente.

- GND
- GPIO32 / Rx2 (ESP32)
- VCC (5 V)
- GPIO33 / Tx2 (ESP32)

Esta toma GPIO permitió la instalación de receptores GPS externos como accesorios aparte, sin embargo, en este caso se va a incluir un receptor GPS interno dentro del dispositivo de puerta de enlace por lo que se ocupó estos mismos pines del Microcontrolador ESP32 para instalar una receptor GPS dentro de la misma carcasa del dispositivo de puerta de enlace, se menciona esto ya que para mejoras posteriores de este dispositivo se puede colocar fácilmente en estos mismos pines un receptor externo para una mejor recepción de señal u otros fines requeridos por el usuario.

2.4.2 RANURAS PARA TARJETAS

El dispositivo cuenta con una ranura para tarjeta microSD y una ranura para tarjeta micro SIM debajo de la cubierta de plástico en un lado de la carcasa. Usando cada uno de los módulos descritos anteriormente se integraron estas ranuras al microcontrolador ESP32 con los pines que se detallan más adelante en la Tabla 10.

2.4.3 MAPA DE E / S

Todos los periféricos están conectados al controlador principal ESP32 de la siguiente manera.

Tabla 10. Mapa de pines de conexión de todos los módulos al microcontrolador principal ESP32

Mapa de E / S	
Periférico	Pines ESP32
OBD-II (STM32)	SPI: GPIO18 (SCK), GPIO19 (MISO), GPIO23 (MOSI) STM32 CS: GPIO2, STM32 LISTO: GPIO13
MPU-9250	I2C: GPIO21 (SDA), GPIO22 (SCL)
GPS	UART2: GPIO32 (RX) GPIO33 (TX) Control de energía: GPIO15
Módulo celular (xBee)	UART1: GPIO16 (RX) GPIO17 (TX) Interruptor de encendido del módulo: GPIO27
microSD	SPI: GPIO18 (SCK), GPIO19 (MISO), GPIO23 (MOSI) SD CS: GPIO18
Serie USB	UART0: GPIO3 (RX) GPIO1 (TX)
Indicador LED	GPIO4

(ALLDATASHEET.COM, 2018)

En la Tabla 10 se puede observar todos los mapas de entradas y salidas digitales y analógicas que tiene el microcontrolador ESP32, así mismo todos los periféricos junto con los puertos GPIO que se necesitan para conectar los diferentes módulos de control que son necesarios para el dispositivo de puerta de enlace como es el puerto OBD-II (STM32), el sensor de movimiento MPU-9250, modulo celular (xBee), la tarjeta micro SD, el transformador de puerto a serie USB y el indicador Led que va conectado al puerto GPIO4 que es una salida digital como notificación para saber si el dispositivo esta habilitado o deshabilitado para enviar datos.

En la Figura 14 se muestra el diseño esquemático completo con todos los módulos conectados al microcontrolador ESP32, guiados por la Tabla 8 donde se describe el mapa de pines con cada uno de los componentes que van a interactuar con el microcontrolador.

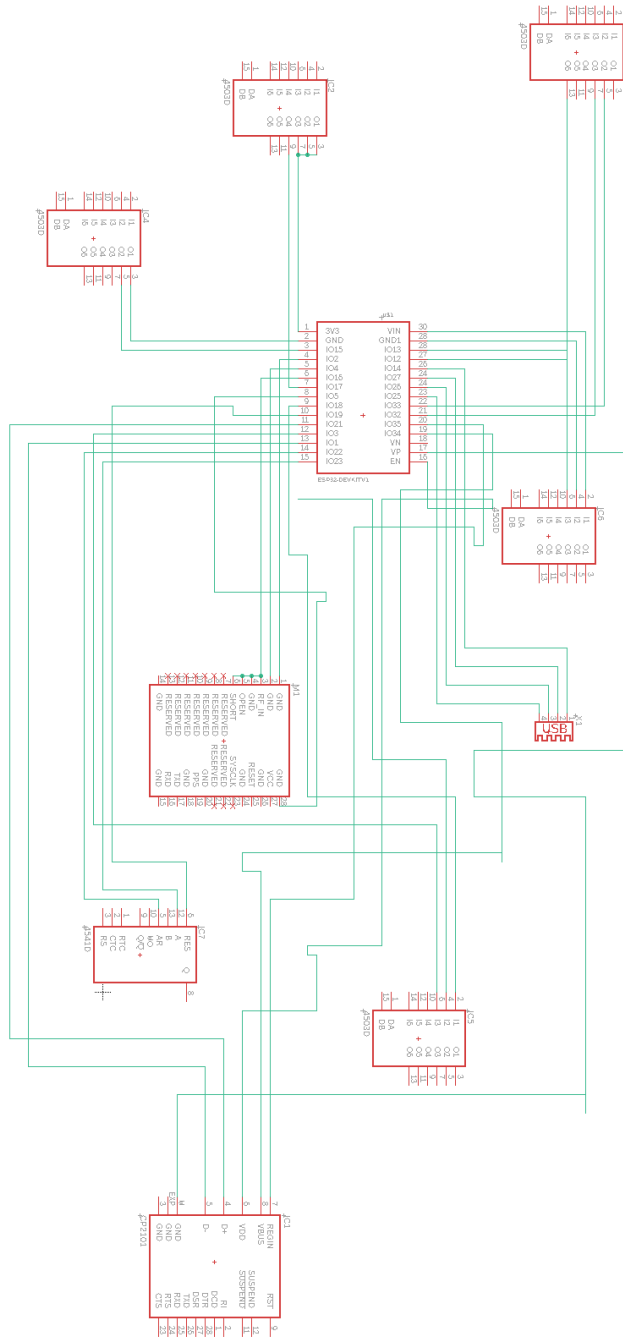


Figura 14. Diseño esquemático de la integración de todos los módulos de control con el microcontrolador ESP32 (ALLDATASHEET.COM, 2018)

2.5 ELABORACIÓN DE LA CARCASA DEL DISPOSITIVO.

Para cumplir con los requerimientos de las partes interesadas se desarrolló una carcasa con un recinto de las siguientes dimensiones: 60 mm de largo por 48 mm de ancho y 20 mm de alto, todos los componentes del dispositivo fueron adquiridos e integrados de tal manera que ocupen esta dimensión en el espacio asignado por la carcasa, a continuación, en la Figura 15 se muestra el resultado final de la carcasa listo para ser impreso en 3D con las cotas mencionadas.

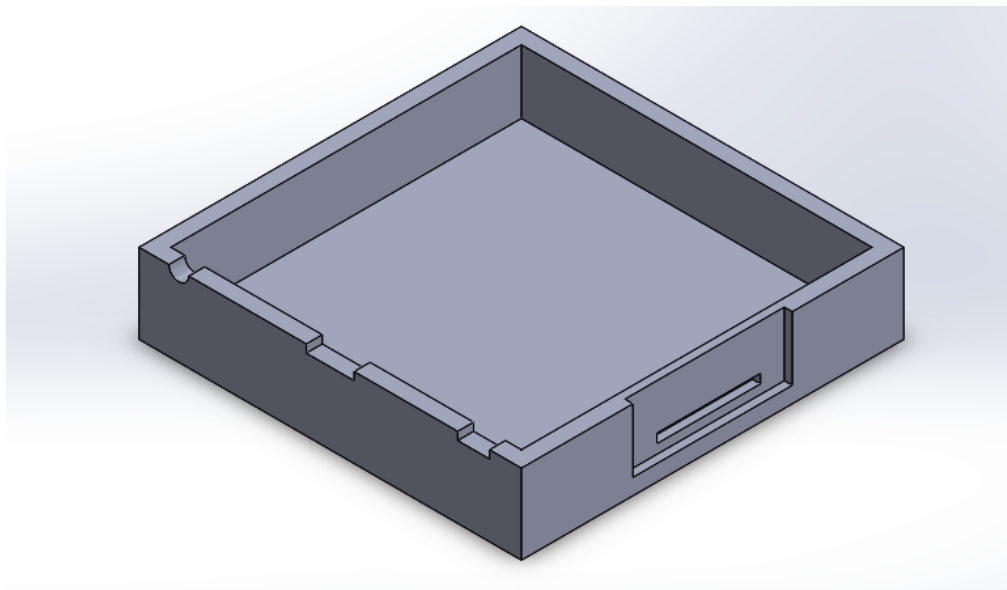


Figura 15. Resultado final de la carcasa del dispositivo de puerta de enlace.

Con la carcasa del dispositivo queda finalizado la primera etapa correspondiente al desarrollo del hardware de control, en adelante se observará la etapa de desarrollo del software de control, veracidad de la información y respaldos del dispositivo, todas estas etapas se elaboraron ya con el hardware de control funcionando de manera correcta y ensamblado, listo para ser usado y probado en un automóvil real que sea compatible con los protocolos de comunicación CAN descritos anteriormente.

2.3.2 SOFTWARE DE CONTROL

2.3.2.1 Desarrollo del código de lectura y procesamiento de datos OBD

2.3.2.1.1 Librerías principales utilizadas

Para desarrollar el código de toma de datos del microcontrolador ESP32 se utilizó una librería ya existente <FreematicsPlus.h> ya que permite tener acceso a datos telemétricos muy importantes de la ECU automotriz y transmitir a través del dispositivo a un monitor serial para comenzar y luego enviar a un servidor de análisis de datos como objetivo final, los PID (Parameter ID) para comunicarse con el sistema de diagnóstico a bordo de un vehículo que se usaron en la programación del código de acuerdo a cada grupo del carro son los siguientes:

Para el sistema eléctrico del vehículo:

PID_HYBRID-BATTERY_PERCENTAGE (este PID permite tomar el porcentaje de batería del vehículo)

PID_CONTROL_MODULE_VOLTAGE (este PID permite tomar el voltaje de la batería del vehículo)

Para el sistema de conducción del vehículo:

PID_AMBIENT_TEMP (este PID permite tomar la temperatura del ambiente dentro del vehículo)

PID_THROTTLE (este PID permite tomar la aceleración del vehículo)

Para el motor del vehículo:

PID_RPM (este PID permite tomar las revoluciones por minuto del motor)

PID_ENGINE_LOAD (este PID permite tomar la carga del motor)

PID_COOLANT_TEMP (este PID permite tomar la temperatura del refrigerante)

PID_TIMING_ADVANCE (este PID permite tomar el tiempo de encendido y apagado del motor)

PID_ENGINE_OIL_TEMP (este PID permite tomar la temperatura del aceite del motor)

PID_ENGINE_TORQUE_PERCENTAGE (este PID permite tomar el porcentaje del torque del motor)

Para la velocidad y tiempo del vehículo:

PID_SPEED (este PID permite tomar la velocidad a la cual va el vehículo en un viaje)

PID_RUNTIME (este PID permite tomar el tiempo que duro un viaje de un punto a otro)

PID_DISTANCE (este PID permite tomar la distancia recorrida en un viaje de un punto hacia otro)

Para la presión y flujos del vehículo:

PID_INTAKE_TEMP (este PID permite tomar la temperatura del aire de admisión)

PID_INTAKE_PRESSURE (este PID permite tomar la presión del aire de admisión)

PID_MAF_FLOW (este PID permite tomar el flujo del aire de admisión)

PID_BAROMETRIC (presión barométrica del aire de admisión)

Adicionalmente se ocupó también la librería <httpd.h> para la configuración del Servidor Apache HTTP con las siguientes directrices:

ServerRoot

La directriz ServerRoot formula el directorio de nivel superior que tiene el argumento web. Por defecto, ServerRoot está configurado a "httpd" para servidores seguros y no seguros.

PidFile

Menciona el archivo en el que el servidor registra los PID del listado solicitado del puerto OBD según los PID instaurados como se explicó anteriormente.

Timeout

Su servicio es definir, en segundos, el tiempo que el servidor aguarda por recibir y transmitir durante la comunicación.

KeepAlive

Resuelve si el servidor permitirá más de una petición por conexión y se puede usar para prever a un cliente consumir demasiados recursos del servidor.

MaxKeepAliveRequests

Esta directriz determina el número máximo de peticiones permitidas por cada conexión permanente.

KeepAliveTimeout

La directriz KeepAliveTimeout establece el número de segundos que el servidor aguarda tras haber dado servicio a una petición, antes de cancelar la conexión.

A continuación, se muestra en la Figura 16 el flujograma de la programación del código de lectura y procesamiento de datos OBD.

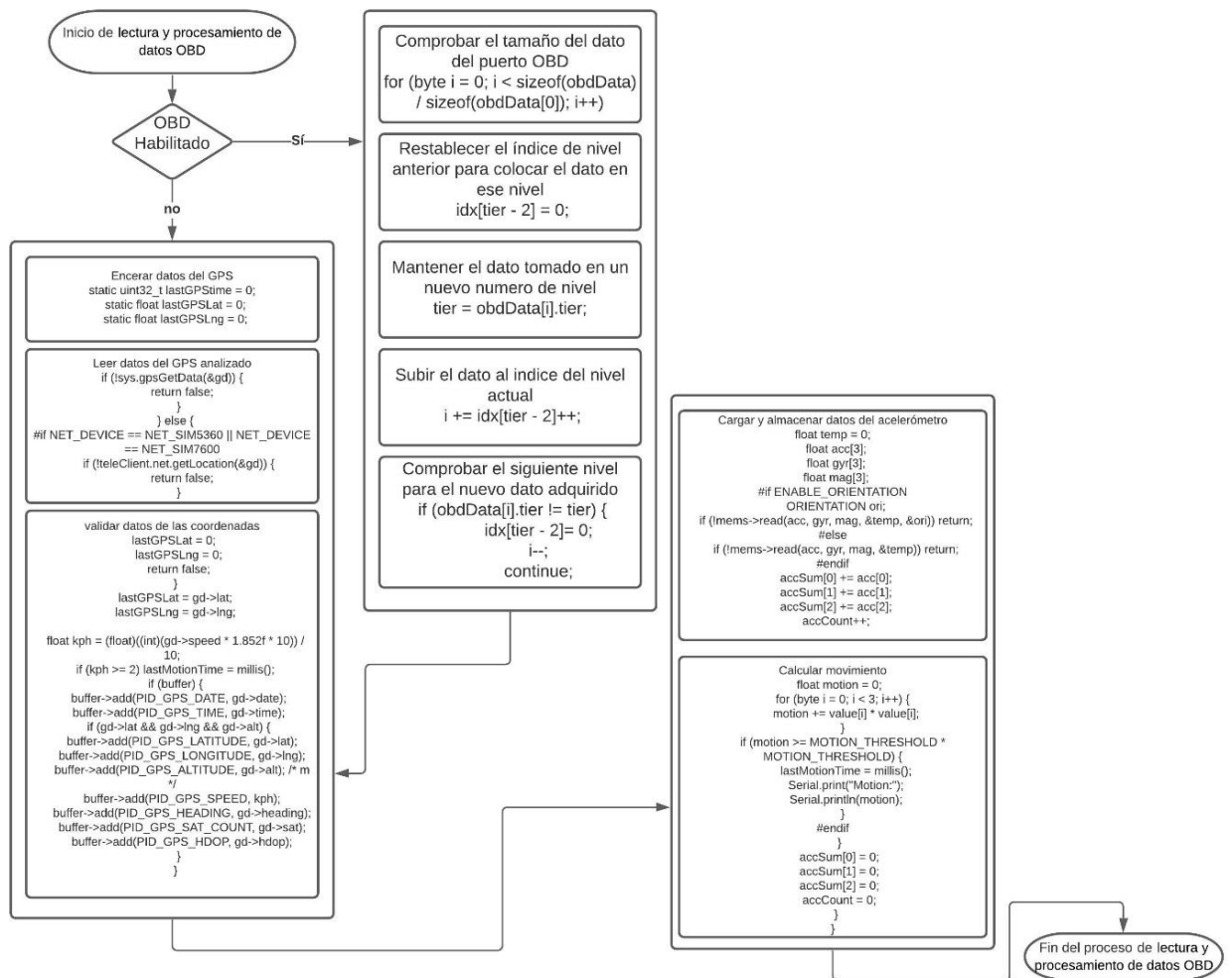


Figura 16. Flujograma de la programación del código de lectura y procesamiento de datos OBD.

2.3.2.2 Desarrollo del código de inicialización de todos los componentes de registro de datos.

Esta sección de código está enfocada en activar todos los módulos que forman parte del dispositivo y que están conectados al ESP32, lo primero que se hizo fue encender el zumbador a una frecuencia de 2000Hz y una vez que llegue a esta frecuencia desactivarlo, en seguida se purga todo el bus de datos para absorber la nueva trama de datos que ingresa, a continuación se inicializa el receptor GPS y luego la comunicación OBD esperando el movimiento del GPS siempre y cuando el OBD no esté conectado, después se inicializa el almacenamiento interno y externo del dispositivo, después se vuelve a intentar

conectarse con el puerto OBD si no se estableció una conexión y por último se comprueba la hora del sistema .

A continuación, en la Figura 17 se puede observar el diagrama de flujo que permitió activar todos los componentes del dispositivo para comenzar a tomar, visualizar y enviar todos los datos a un servidor.



Figura 17. Flujograma de la programación del código de inicialización de todos los componentes de registro de datos.

2.3.2.3 Desarrollo del código de envío de datos HTTP API

Esta porción del código es utilizada para poder postear los datos receptados por el dispositivo conectado al puerto OBD-II del vehículo a Freematics Hub que es

un servidor de datos de telemetría que permite conectar varios dispositivos y recibir datos de telemetría remotos para almacenarlos en la memoria cache o en archivos de disco y suministrar estos datos de forma organizada a pedido, estos datos se transmiten desde el dispositivo como datagramas UDP (User Datagram Protocol) encapsulado de capa 4 o de transporte del modelo OSI a través de la red sin que se haya establecido previamente una conexión ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera y están disponibles para los pares que los requieran mediante API HTTP/REST simples.

2.3.2.3.1 Alimentación de datos del dispositivo al servidor

El dispositivo de puerta de enlace recopila los datos y los transmite a Freematics Hub mediante el envío de datagramas UDP o solicitudes HTTP GET o POST continuas de la siguiente manera.

- 1) Notifica al servidor de análisis de datos que el dispositivo de puerta de enlace está comenzando a recopilar datos o a iniciado sesión.
- 2) Envía datos al servidor
- 3) Notifica al servidor de análisis de datos que el dispositivo de puerta de enlace ha dejado de recopilar datos o a cerrado sesión
- 4) Hace ping al servidor de un análisis de datos en un intervalo de tiempo determinado para informar al servidor si el dispositivo de puerta de enlace esta fuera de línea o conectado

2.3.2.3.2 Notificación de eventos

Se utiliza el siguiente datagrama:

<ID de dispositivo> # EV = <ID de evento>, SK = <clave de servidor>, TS = <marca de tiempo del dispositivo>, VIN = <ID de vehículo> * <suma de comprobación>

2.3.2.3.3 Transmisión de datos

Se utiliza el siguiente datagrama:

<ID de dispositivo> # <PID 1>: <datos 1>, <PID 2>: <datos 2>, <PID 3>: <datos 3 valor 1>; <datos 3 valor 2>; <datos 3 valor 3> ,... \$ <Suma de comprobación>

2.3.2.3.4 Iniciar sesión y cerrar sesión

Se puede mandar una notificación de evento mediante UDP o HTTP GET.

El evento de inicio de sesión (ID de evento 1) menciona que la sesión de transmisión de datos está a punto de iniciar y que se envían algunas identificaciones, incluido el número de identificación del vehículo (VIN) que se emplea como identificador único. El evento de cierre de sesión (ID de evento 2)

determina que la sesión de transmisión de datos está a punto de finalizar, lo más posible es que haya terminado un viaje.

2.3.2.3.5 Notificación por UDP

Un aviso de notificación típica de UDP tiene la carga útil siguiente.

ABCDEFGH # EV = 1, TS = 39539, VIN = A1JC5444R7252367 * XX

ABCDEFGH es ID de dispositivo

EV = 1 indica un suceso de inicio de sesión

TS =... la marca de tiempo actual (normalmente el contador del temporizador del dispositivo)

VIN =... determina el VIN, que debe ser único y se emplea para asignar la identificación del feed

* XX es la suma de comprobación (en hexadecimal) de todos los bytes anteriores

Una vez que el servidor acepta una solicitud UDP, responde un datagrama UDP que transporta una carga útil como seguir, como respuesta.

1 # EV = 1, RX = n, TS = 39539 * X

El primer número (en hexadecimal) es el ID del feed (arbitrario)

EV = n es el ID de evento de la solicitud

TS =... es la marca de tiempo de la solicitud

RX =... es el número de solicitudes / transmisiones de datos recibidas

* XX es la suma de validación (en hexadecimal) de todos los bytes anteriores

2.3.2.3.6 Notificación por HTTP

La actividad de notificación también se puede enviar mediante HTTP o una solicitud HTTP más certera con el siguiente formato de solicitud. Las definiciones de parámetros son idénticas a las de UDP.

http: // <host del servidor> / api / notificar / [ID DE DISPOSITIVO]? EV = <ID de evento>& TS = <marca de tiempo del dispositivo>& VIN = <número de ID del vehículo>

Respuesta HTTP en caso de éxito

```
{"resultado": "hecho", "id": n}
```

n es un número entero arbitrario que se emplea para la alimentación de datos posterior denominado ID de feed .

Respuesta HTTP en caso de error

```
{"resultado": "fallido", "error": "Clave de servidor no válida"}
```

```
{"resultado": "fallido", "error": "VIN no válido"}
```

2.3.2.3.7 Alimentación de datos por UDP

El uso de UDP para sustentar los datos en vivo es lo más eficiente para un dispositivo con una comunicación de red no siempre estable. De esta manera, las cargas útiles que contienen un grupo de datos recopilados y suma de validaciones se transmiten como datagramas UDP, para realizar una carga útil es necesario utilizar un formato de datos empaquetados que está basado en texto empleado para la transmisión de datos mediante UDP o HTTP POST. La secuencia son fragmentos de pares PID / valores separados por comas. A continuación, se muestran algunos PID de modo 01 de uso común en el estándar OBD-II.

- 0x104 - Carga del motor
- 0x105 - Temperatura del refrigerante del motor
- 0x10a - Presión de combustible
- 0x10b - Presión absoluta del colector de admisión
- 0x10c - RPM del motor
- 0x10d - Velocidad del vehículo
- 0x10e - avance de tiempo
- 0x10f - Temperatura del aire de admisión
- 0x110 - Caudal de aire MAF
- 0x111 - Posición del acelerador
- 0x11f - Tiempo de ejecución desde el arranque del motor
- 0x121 - Distancia recorrida con luz indicadora de mal funcionamiento
- 0x12f - Entrada de nivel de combustible
- 0x131 - Distancia recorrida desde que se borraron los códigos
- 0x133 - Presión barométrica
- 0x142 - Voltaje del módulo de control
- 0x143 - Valor de carga absoluto
- 0x15b - Duración restante de la batería híbrida
- 0x15c - Temperatura del aceite del motor

➤ 0x15e - Tasa de combustible del motor

Para la transmisión UDP, se aumenta un encabezado simple que contiene la identificación del dispositivo y una cola de suma de validación como sigue. Ambos están representados en hexadecimal.

Ejemplo:

<ID de dispositivo> # <PID 1>: <valor 1>, <PID 2>: <valor 2>, <PID 3>: <valor 3>, ... * <Suma de comprobación>

ABCDEFGH # 0: 68338,10D: 79,30: 1010,105: 199,10C: 4375,104: 56,111: 62,20: 0; -1; 95,10: 6454200, A: -32.727482, B: 150.150301, C: 159, D: 0, F: 5, 24: 1250 * 7^a

2.3.2.3.8 Alimentación de datos mediante HTTP GET

Usar una solicitud HTTP GET para alimentar datos en vivo es sencillo de probar (incluso desde el navegador). Se aceptan diversos pares de valores y PID como argumentos de URL para una solicitud. Los números PID están en formato hexadecimal y los valores están en formato decimal. Es necesario transportar datos con la misma marca de tiempo mediante una transacción HTTP GET.

Solicitar URL

http:// <host del servidor> / api / push / [ID DE DISPOSITIVO]? TS = [marca de tiempo] & [pid1] = [valor1] & [pid2] = [valor2] ...

Respuesta (éxito)

```
{"resultado": n}
```

n es el número de pares PID / valores aceptados.

Respuesta (errónea)

```
{"resultado": "fallido", "error": "Clave de servidor no válida"}
```

```
{"resultado": "fallido", "error": "ID de FEED no válido"}
```

2.3.2.3.9 Alimentación de datos mediante HTTP POST

El uso de HTTP POST para alimentar datos tiene la superioridad de seguridad y flexibilidad. Se puede transferir un grupo de datos con diferentes marcas de tiempo mediante una solicitud como UDP.

Solicitar URL

http: // <host del servidor> / api / post / [ID DE DISPOSITIVO]

Respuesta (éxito)

```
{"resultado": n}
```

n es el número de pares PID / valor de datos aceptados.

Respuesta (errónea)

```
{"resultado": "fallido", "error": "Clave de servidor no válida"}
```

```
{"resultado": "fallido", "error": "ID de FEED no válido"}
```

2.3.2.3.10 Obteniendo datos del servidor

El software que está diseñado para ilustrar, procesar, analizar o almacenar datos de forma permanente, utiliza un grupo de solicitudes HTTP SOAP API para alcanzar datos de Freematics Hub. La GUI de Freematics Hub se aplica mediante la API.

Obtención de lista de feeds de datos

Solicitar URL

http: // <host del servidor> / api / channels

Respuesta

La respuesta es una matriz de objetos en la que cada elemento corresponde a una fuente de datos.

```
{"canales": [
```

```
  {"id": "1", "devid": "A3HNZMJG", "recv": 26253, "tick": 505269, "age": 2250, "flags": 1, "csq": 22},
```

```
{"id": "2", "devid": "ES9129GD", "recv": 432432, "tick": 1205269, "age": 1125, "flags": 1, "csq": 25},
```

...

```
]}]
```

Propiedades:

- id: ID de feed
- vin - número de identificación del vehículo
- recv - datos recibidos (bytes) durante la sesión / viaje actual
- tick - tic del reloj del sistema del dispositivo (ms)
- edad: tiempo transcurrido (ms) desde la última entrada de datos recibida hasta que se solicita
- csq - índice de intensidad de la señal (en dB)
- bandera - 1 para estado activo / en ejecución, 0 para estado inactivo / estacionado

2.3.2.11 Obtener datos instantáneos

Esto es para solicitar datos en vivo de una fuente de datos.

Solicitar URL

`http://<host del servidor>/api/get/[ID DE DISPOSITIVO]`

Ejemplo de respuesta:

```
{"stats": {"tick": 431865078, "devtick": 818627, "elapsed": 70, "age": {"data": 3234, "ping": 0}, "estacionado": 0},
```

```
" datos ": [[32, [0,0,0], 3234], [36,1229,3234], [130,44,18300], [260,87,3234], [261,61,13283], [266,549,3234], [268,7500,3234], [269,0,3234]]]
```

En la respuesta, las estadísticas contienen información estadística que incluye:

- tick - marca de tiempo del servidor (en ms)
- devtick - marca de tiempo del dispositivo (en ms)
- transcurrido - tiempo de ejecución del viaje / sesión
- age.data - tiempo (en ms) transcurrido desde la última transmisión de datos
- age.ping - tiempo (en ms) transcurrido desde el último ping de regreso (cuando el dispositivo está en modo de espera)

Los datos de la matriz abarcan todos los datos instantáneos manifiestos en una matriz de 3 elementos que incluyen:

- PID - en decimal

- valor de datos (puede ser un número entero, flotante o una matriz de 3 elementos para datos de 3 ejes)
- edad: tiempo (en ms) transcurrido desde su recopilación desde el extremo del dispositivo

2.3.2.3.12 Secuencia de operación

La secuencia básica para conseguir datos instantáneos es la siguiente.

- 1) Obtener la lista de fuentes de datos (que contiene el ID y algunas propiedades de una fuente de datos)
- 2) Obteniendo datos instantáneos (con ID de feed)
- 3) Analizar la carga útil JSON de respuesta y gestionar los datos
- 4) Feedback paso 2 y 3

2.3.2.3.13 Obteniendo datos del historial

Una cola de datos es una serie de datos con marca de tiempo. La solicitud de una cola de datos puede determinar el intervalo de marca de tiempo o el tiempo para revertir. Los datos están ordenados como una matriz JSON de 2 dimensiones. Esto es útil para representar una tabla de datos o una ruta en el mapa con muestras de datos completas, de modo que la línea en la tabla o la forma de la ruta se vea suave. Esto también es necesario para almacenar en caché o almacenar datos de una fuente de datos.

Solicitar URL

`http: // <host del servidor> / api / pull / <ID DE DISPOSITIVO>? [ts = <ms>] [&endts = <ms>] [&rollback = <ms>]`

Solicitar argumentos

- ts se utiliza para especificar la hora de inicio de la cola de datos solicitada. El tiempo se presenta en el tic del reloj del sistema del dispositivo en ms.
- endts se utiliza para especificar la hora de finalización de la cola de datos solicitada. El tiempo se presenta en el tic del reloj del sistema del dispositivo en ms.
- la reversión establece la hora de inicio con el tiempo relativo en ms sin necesidad de conocer el tic del reloj local del dispositivo. Por ejemplo, rollback = 60000 indica que se solicitan datos de hace 1 minuto.

Descripción de propiedades:

- stats.tick - tic del reloj del sistema del dispositivo (ms)
- stats.recv- datos recibidos (bytes) en el viaje actual
- stats.elapsed - tiempo transcurrido (segundos) en el viaje actual

- stats.age - tiempo transcurrido (ms) desde la última entrada de datos recibida hasta que se solicita
- stats.flag - 1 para conducir / activo, 0 para estacionado / inactivo
- datos []: cola de datos representada como una matriz de matriz de 3 elementos que consta de marca de tiempo, tipo de datos y valor de datos
- eos: indica si este es el final de la transmisión (no hay más datos) en el momento en que se realiza la solicitud

Respuesta (errónea)

```
{"resultado": "fallido", "error": "Clave de servidor no válida"}
```

```
{"resultado": "fallido", "error": "ID de FEED no válido"}
```

Secuencia de operación

La secuencia para obtener la cola de datos completa es la siguiente.

- 1) Obtener la lista de feeds de datos (que contiene el ID del feed y algunas propiedades del feed de datos)
- 2) Obtener una cola de datos con tiempo de reversión especificado
- 3) Analizar la carga útil JSON de respuesta, tratar los datos y mantener la marca de tiempo de la última muestra de datos
- 4) Obtener la siguiente cola de datos con la marca de tiempo especificada (hora de inicio establecida como la última muestra de datos más 1)
- 5) Feedback paso 3 y 4

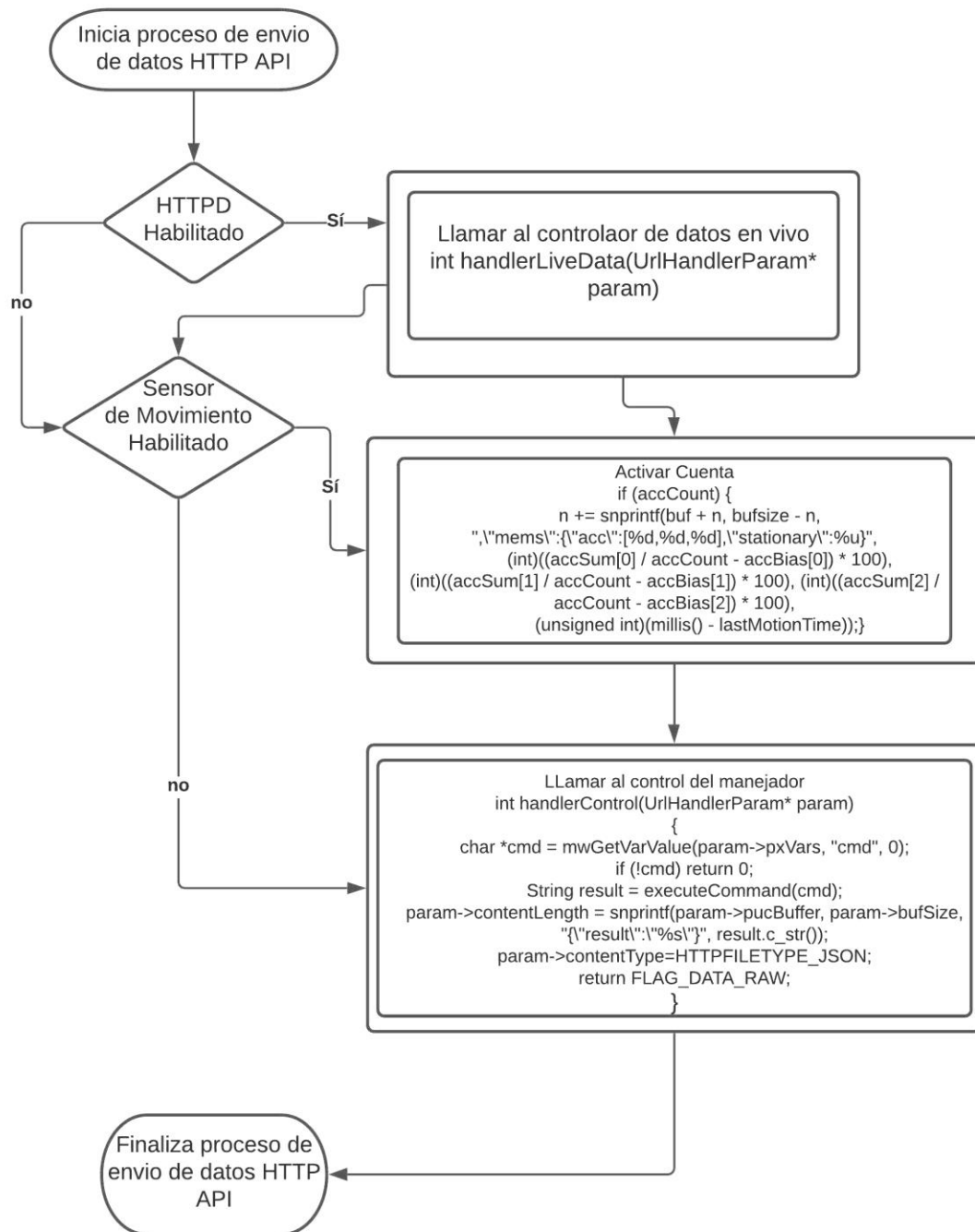


Figura 18. Desarrollo del código de envío de datos HTTP API.

En la Figura 18 se puede observar el flujograma del proceso de envío de datos mediante la API de Freematics hacia el servidor de análisis de datos.

2.3.2.4 Programación del código de recopilación y procesamiento de datos.

Esta parte está enfocada en la obtención de datos por medio del puerto OBD-II, así como el sensor de movimiento del dispositivo y el GPS para transmitirlos a través de la red WI-FI o GSM, en este proceso es necesario contar con una SIM Card en el caso de transmitir los datos vía GSM o un proveedor de internet si lo deseamos hacer por WI-FI, para que el dispositivo sea útil se elige enviar vía GSM Cat 4G utilizando la tarjeta SIM global IoT Hologram mostrada en la Figura 19, la ventaja de esta tarjeta SIM es que está basada en una solución orientada al internet de las cosas por lo que cuenta con un panel de administrador donde podemos ver todos los dispositivos que están conectados así como los datos que se están transfiriendo,



Figura 19. Tarjeta SIM Hologram IoT.
(Hologram, 2020)

En la Figura 19 se puede evidenciar la tarjeta SIM Hologram IoT, tarjeta con la cual se va a trabajar como medio para transferir toda la información de la puerta de enlace al servidor de análisis de datos vía GSM.

A continuación, en la Figura 20 se puede apreciar el flujograma de trabajo del inicio del proceso de recopilación y procesamiento de datos del dispositivo de puerta de enlace para la transmisión de la información al servidor de análisis de datos.

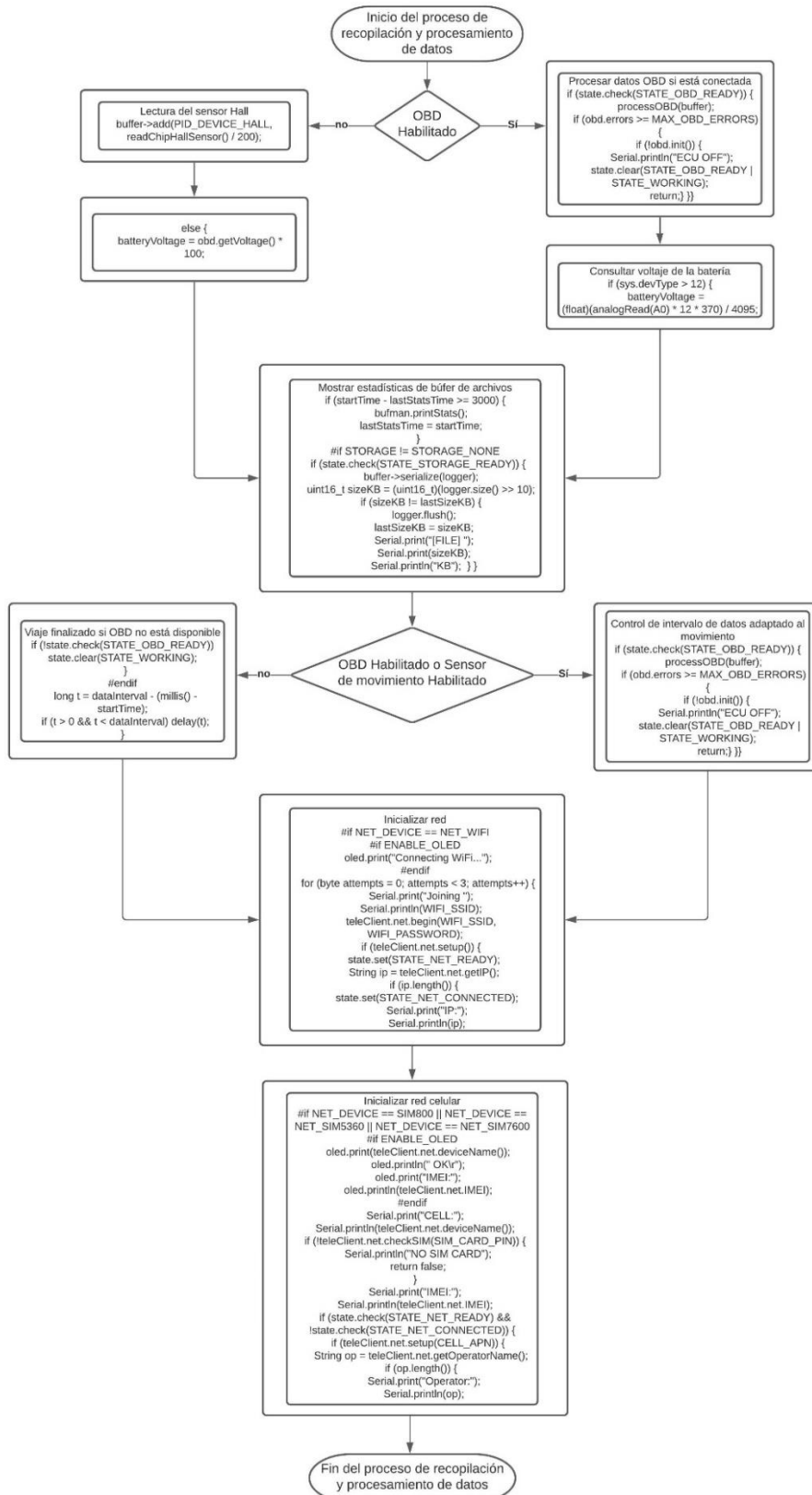


Figura 20. Diagrama del proceso de recopilación y procesamiento de datos.

2.3.2.5 Desarrollo del código de inicialización de la red, mantenimiento de la conexión y transmisiones

En este segmento del código se explica cómo se logró conectar el dispositivo con la red enviando un Pingy revisando en el puerto serial para verificar si tuvo éxito o no el envío de este Ping. Esto permitió generar un enlace entre el dispositivo y la red para realizar el envío de la telemetría por medio del protocolo UDP, asimismo, para el mantenimiento de la conexión y transmisión de datos se declaró una alerta en caso de que el dispositivo sobrepase la temperatura de trabajo, el dispositivo notificará que está demasiado caliente y procederá al enfriamiento de este, pausando el envío de la telemetría del automóvil hasta que se encuentre en una temperatura prudente de trabajo.

En la Figura 21 se puede observar el diagrama de flujo del proceso de inicialización de la red, mantenimiento de la conexión y transmisiones.

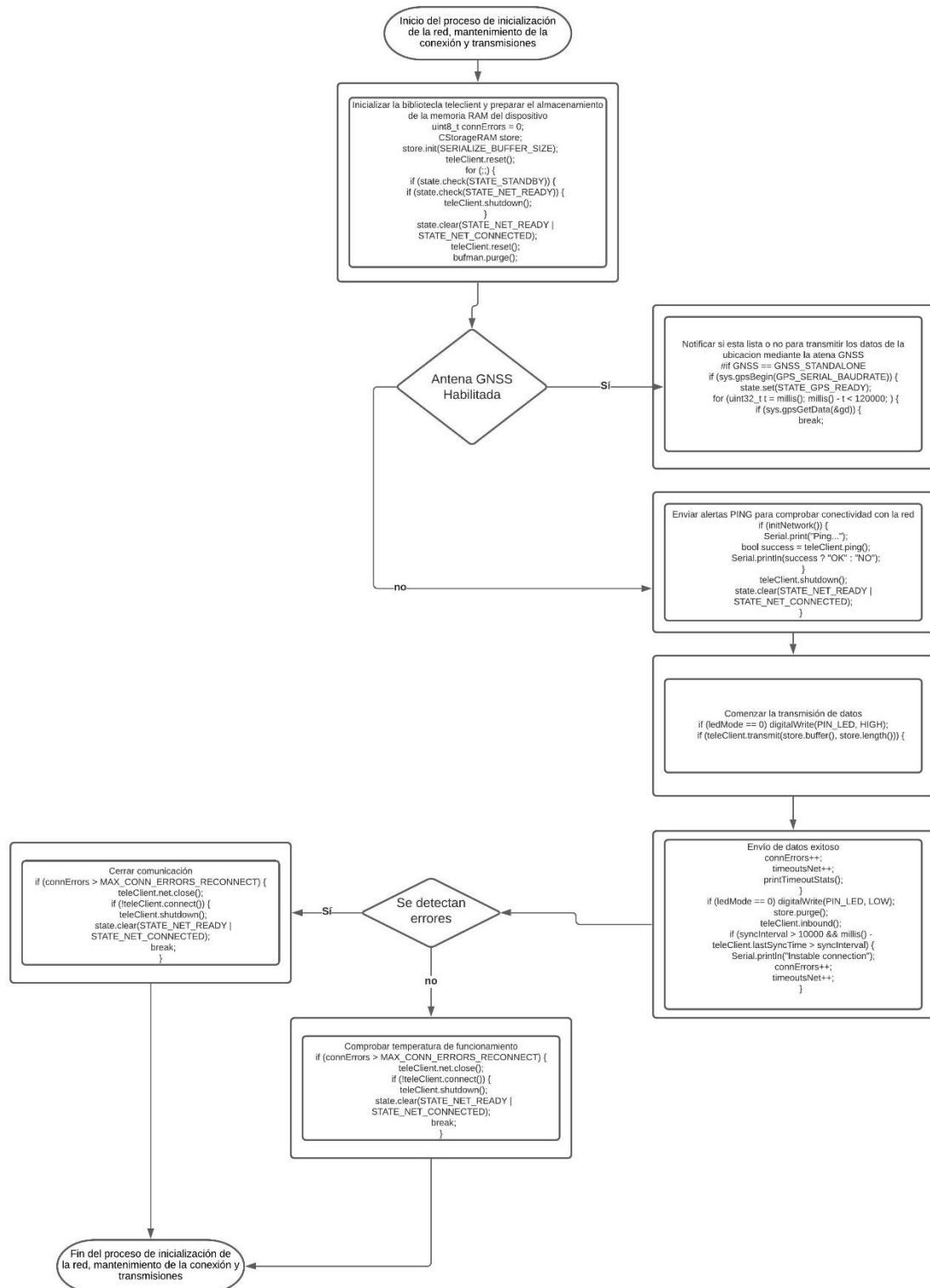


Figura 21. Diagrama de flujo del proceso de inicialización de la red, mantenimiento de la conexión y transmisiones.

2.3.3 RESPALDOS DEL DISPOSITIVO

2.3.3.1 Implementación de la batería de respaldo para el dispositivo.

2.3.2.1.1 Cálculo de la batería de respaldo

La batería de control incluye a todos los sensores, módulos y el microcontrolador ESP32

Cargas de cada elemento:

$$I_{instrumentación} = I_{ESP32} + I_{Módulo LTE CAT4} + I_{MPU 9250} + I_{Módulo STM32F103CX} + I_{GNSS M8030} + I_{Módulo CP2102} + I_{Módulo W25QXX}$$

$$I_{instrumentación} = I_{150 mA} + I_{5mA} + I_{10mA} + I_{80mA} + I_{70mA} + I_{5mA} + I_{10mA}$$

$$I_{instrumentación} = 330 mA$$

El dispositivo de puerta de enlace debe tener una autonomía mínimo de 15 minutos después del fallo de la batería del automóvil.

La corriente total que consume es de 330 mA

Agregando un factor de seguridad del 25%

$$I_{consumo total} = 1.25 * 330 mA = 412,5 mA$$

La capacidad de la batería del dispositivo de puerta de enlace está determinada por:

$$Ab(\text{consumo total}) = \frac{\text{capacidad de la batería}}{\text{consumo esperado}}$$

$$\text{capacidad de la batería} = Ab * \text{consumo esperado}$$

$$\text{capacidad de la batería} = 412,5 (mA) * 0,33 (h) = 136 (mAh)$$

Para seleccionar el voltaje de la batería se toma en cuenta el consumo principal del microcontrolador ESP32 que va en un rango de 3.3V a 5V, por las

consideraciones que se mencionó anteriormente, es recomendable utilizar una batería LiPoTurnigy nano-tech 300 mAh de 7.4V ya que con esta capacidad el sistema estará en funcionamiento por un tiempo mayor igual a 15 minutos, en la siguiente Tabla 11 se describe las especificaciones técnicas de esta batería.

Tabla 11. Especificaciones técnicas de la batería de respaldo LiPoTurnigy nano-tech

LiPoTurnigy nano-tech	
Especificaciones	Descripción
Capacidad	300 mAh
Voltaje	2S1P/2Cell/7.4V
Descarga	35C Constantes/70C Ráfagas
Peso	17g
Dimensiones	44x12x17mm
Tipo de enchufe	JST-XH
Enchufe de descarga	JST

(Mercado Libre, 2020)

Existen dos posibilidades para cambiar el modo de alimentación del dispositivo de puerta de enlace, una mediante programación y otra mediante un circuito de control que ayude a detectar cuando el dispositivo se queda sin alimentación de la fuente principal del vehículo y active la batería LiPo para que se alimente de esa fuente, tomando la segunda opción a continuación en la Figura 22 se explica mediante un diagrama esquemático el circuito de control que permite la conmutación automática para alimentar el dispositivo de una fuente a otra cuando detecte que la batería del vehículo quedó inhabilitada por diferentes motivos.

2.3.2.1.2 Circuito de control de conmutación de la batería principal a la batería de respaldo

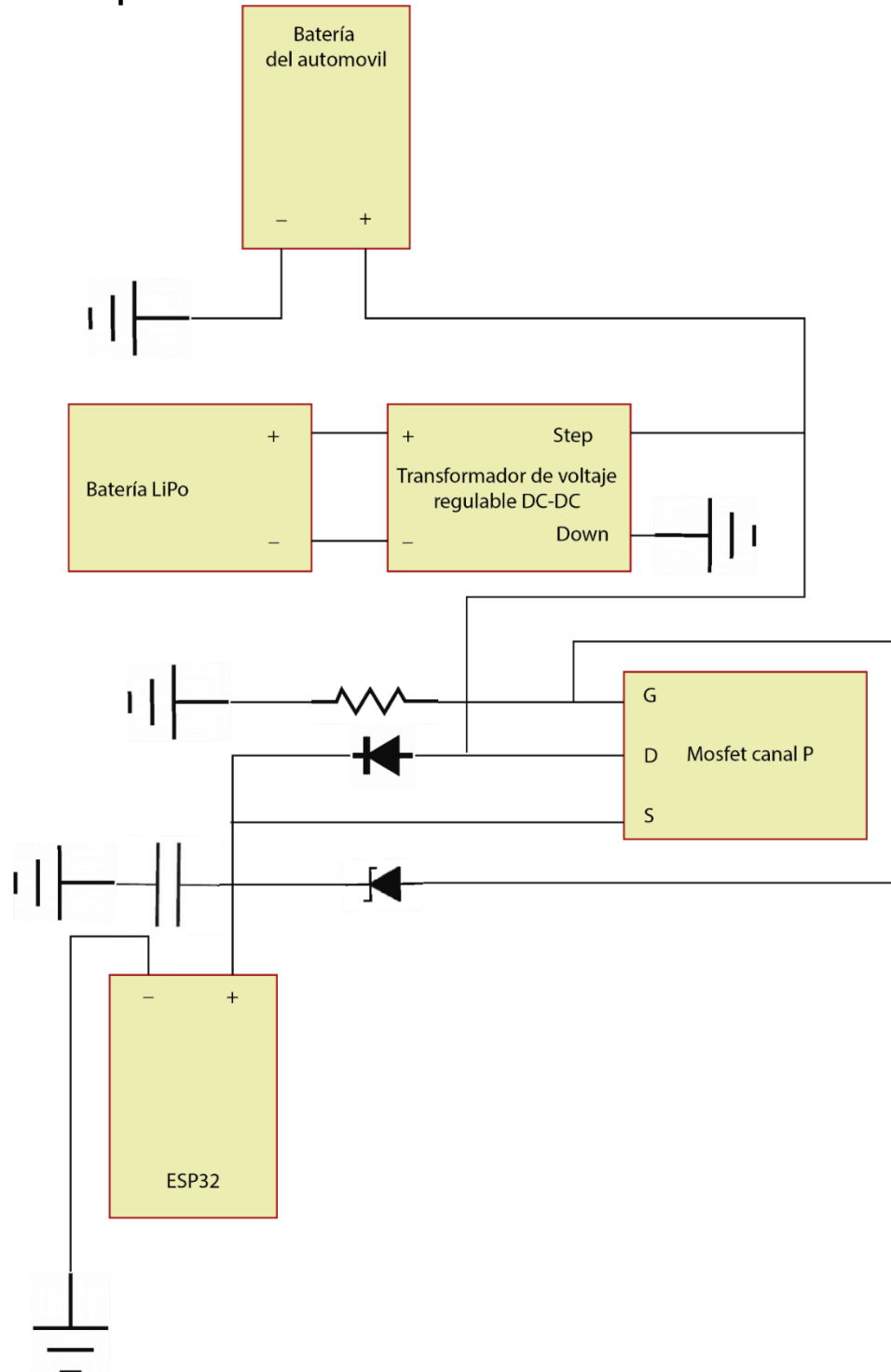


Figura 22. Diagrama esquemático del circuito de control para la conmutación de la alimentación del dispositivo de puerta de enlace.

Para conectar el dispositivo al circuito de control y la batería LiPo se utilizó una extensión externa con un zócalo de 4 pines que contiene 2 pines de E/S para los puertos de propósito general (GPIO26 y GPIO34 del ESP32), VCC (controlable por GPIO12 de ESP32) y GND, como sigue a continuación en la Figura 23.



Figura 23. Dispositivo de puerta de enlace con los puertos externos para conectar la batería LiPo o algún sensor adicional para analizar una variable adicional que desee el cliente.

Con un cable de conversión de conector Molex a Dupont 2.54, los sensores externos o una fuente de alimentación externa se pueden conectar fácilmente. Usando los colores negros para GND, rojo para VCC, blanco para GPIO34 y verde para GPIO26.

Más adelante en la Figura 24 se puede evidenciar el dispositivo de puerta de enlace acoplado a un conector Molex, el cual nos permite incorporar la batería externa del dispositivo para que este siga funcionando después de que el vehículo quede sin alimentación y pueda continuar enviando la telemetría automotriz al servidor de análisis de datos por un determinado tiempo antes del fallo.

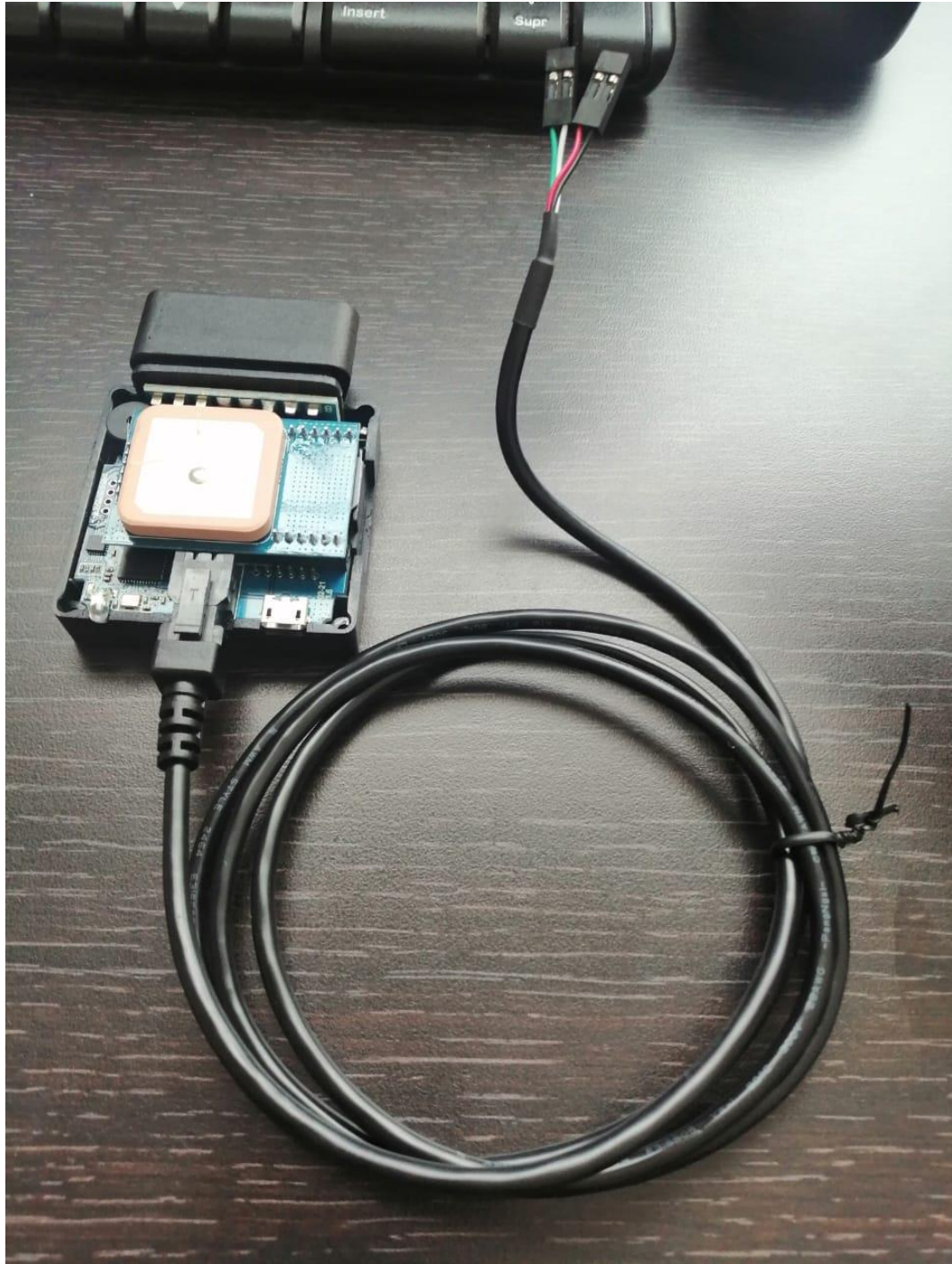


Figura 24. Dispositivo de puerta de enlace conectado al conector Molex para añadir una fuente de alimentación externa.

3. ANÁLISIS DE LOS RESULTADOS

3.1 COMPARACIÓN DE DATOS

El dispositivo de puerta de enlace fue instalado para el desarrollo de pruebas y comprobación del funcionamiento de todo el sistema en un vehículo Volkswagen Jetta año 2018. Se utilizó este vehículo ya que es compatible con el protocolo de comunicación KWP2000 mencionado anteriormente como uno de los protocolos compatibles utilizado para los diagnósticos a bordo con puerto OBD-II que satisface a los requerimientos de comunicación interna necesaria para el dispositivo de puerta de enlace.

Lo primero que se realizó fue cargar todo el código de Arduino IDE al microcontrolador y para verificar que todos los componentes están funcionando correctamente se pasó el programa y el dispositivo por una API de Freematics que utiliza el desarrollador de Arduino como motor principal de depuración, esta API permite visualizar como puerto serial todos los datos que estamos obteniendo pero de una manera clara y ordenada, a continuación se muestra la Figura 25 donde se muestra la interfaz que brinda este widget de Freematics.

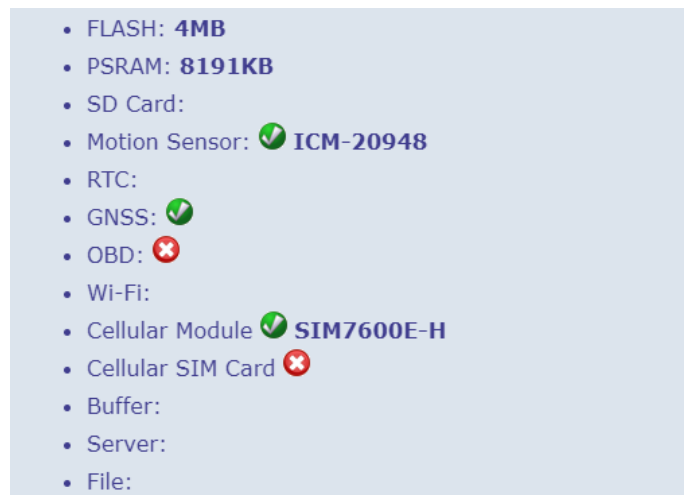


Figura 25. Api Freematics para la comprobación de todos los módulos y el microcontrolador utilizados en el dispositivo de puerta de enlace antes de ser instalado en un vehículo.

En la Figura 26 se puede observar la interfaz de usuario mediante la cual se puede revisar todos los elementos que están habilitados en el dispositivo para ser utilizados cuando este entre en conexión con el vehículo.

Como se puede observar la memoria flash del dispositivo es de 4MB, la RAM de 8191KB, así mismo el sensor de movimiento está activo y enviando datos, también el GPS está en línea, pero el puerto OBD está desconectado ya que está conectado el dispositivo de puerta de enlace al computador por medio de un cable USB, el módulo WI-FI está apagado ya que tiene el módulo GSM activo para conectarse a Internet.

Una vez que se comprobó que todos los componentes funcionan correctamente, se procedió a cargar el código al microcontrolador nuevamente, pero esta vez por medio de esta plataforma, para verificar que ya está en línea el dispositivo conectamos mediante un token que da la API de Freematics para usar el servidor web ya desarrollado por ellos y visualizar todos los datos en tiempo real. En la Figura 26 está el token junto con la pantalla principal para conectar el dispositivo de puerta de enlace con el servidor de análisis de datos de Freematics que es una plataforma de código abierto para poderle modificar o personalizar según las necesidades del cliente, lo que hace al dispositivo accesible y flexible para poder comercializar.



Figura 26. Servidor de análisis de datos Freematics con el token de acceso entregado por la Api de comprobación Freematics.

Este servidor es totalmente gratuito para pruebas y si más adelante se desarrolla una solución a medida del cliente se puede personalizar toda la interfaz y conectar con motores externos de análisis de datos como Microsoft Azure, Amazon Web Services, PowerBI entre otros, por lo que es bastante flexible y escalable para soluciones más complejas o concretas, claro que dejaría de ser gratuito si se requiere para uso comercial, pero por el momento para pruebas y aprendizaje es adecuado.

3.2 PRUEBAS DE FUNCIONAMIENTO

Una vez implementado el dispositivo se puede observar toda la telemetría automotriz recopilada por la puerta de enlace para ser enviada al servidor de análisis de datos Freematics, a continuación, se describe cada una de las variables tomadas para ser analizadas y se compara en tiempo real con la información que proporciona vehículo.

3.2.1 ACELERACIÓN

En un tiempo recorrido de 00:03:37 con 22 KB recibidos con 600 muestras por minuto a una temperatura de trabajo del dispositivo de 42 °C en la fecha 2021-02-10 se toma la siguiente información representada en la Figura 27 en una gráfica aceleración (g) vs tiempo (s) como se puede observar existió nueve cambios bruscos de aceleración en donde el automóvil al inicio varió su aceleración de 0 a 1m/s^2 y en cada pico alcanzó una aceleración máxima de 2m/s^2 .

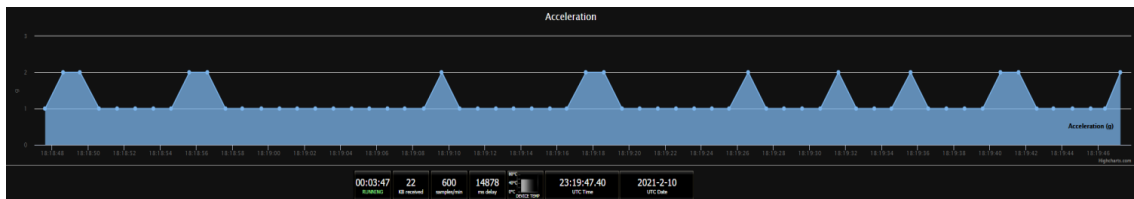


Figura 27. Datos del gráfico de aceleración tomados en un lapso de 00:03:37 minutos en un diagrama de aceleración (g) vs tiempo (s).

3.2.2 VELOCIDAD

En un tiempo recorrido de 00:03:27 con 20 KB recibidos con 600 muestras por minuto a una temperatura de trabajo del dispositivo de 42 °C en la fecha 2021-02-10 se toma la siguiente información representada en la Figura 28 en una gráfica velocidad (Km/h) vs tiempo (s), como se puede observar en esta muestra el vehículo inicia con una velocidad de 45 Km/h y llegó en este tramo a una velocidad máxima 110 Km/h.

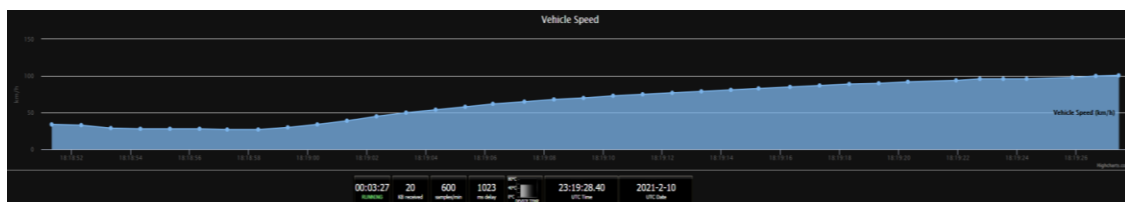


Figura 28. Datos del gráfico de velocidad tomados en un lapso de 00:03:27 minutos en un diagrama de velocidad (Km/h) vs tiempo (s).

A continuación, se muestra la Tabla 12 con la comparación de la velocidad receptada en el servidor de análisis de datos Freematics en contraste con la velocidad que marca el velocímetro del vehículo en el panel de control.

Tabla 12. Comparación de la velocidad receptada en el servidor de análisis de datos Freematics contra la velocidad apuntada cada 2 segundos del panel de control del vehículo.

Comparación de la velocidad tomada por el dispositivo de puerta de enlace contra la velocidad tomada del panel de control del vehículo		
Velocidad del servidor de análisis de datos (Km/h)	Velocidad del panel de control del vehículo (Km/h)	Tiempo(s)
45,78	46	2
41,25	41	4
41,08	41	6
42,31	42	8
44,87	45	10
48,53	48	12
53,21	53	14
68,95	69	16
74,41	74	18
76,87	77	20
80,65	80	22
84,98	85	24
88,23	88	26
91,56	92	28
94,78	95	30
99,98	100	32
109,79	110	34

3.2.3 VOLTAJE DE LA BATERÍA

En un tiempo recorrido de 00:03:47 con 22 KB recibidos con 600 muestras por minuto a una temperatura de trabajo del dispositivo de 42 °C en la fecha 2021-02-10 se toma la siguiente información representada en la Figura 29 en una gráfica voltaje (V) vs tiempo (s), como se puede observar en esta muestra la batería del vehículo se mantiene casi constante con unas variaciones muy pequeñas en todo el tramo de tiempo medido, lo que quiere decir que está funcionando correctamente el sistema de carga del vehículo comprendido por el alternador y por supuesto que la batería se encuentra en estados óptimos de funcionamiento y por el momento no necesita ser cambiada, en la gráfica se puede observar una pequeña fluctuación de 14,82 volts a 14,17 mientras permanezca en este rango el vehículo está siendo correctamente alimentado.

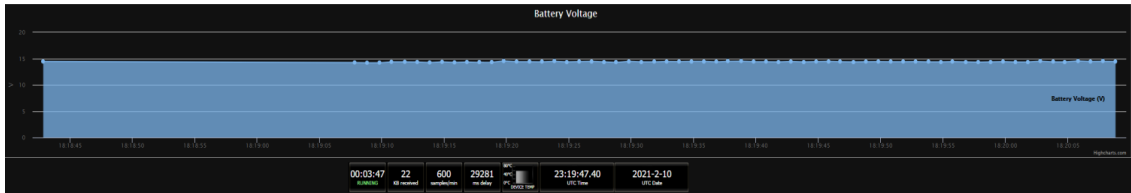


Figura 29. Datos del grafico del voltaje de la batería tomados en un lapso de 00:03:47 minutos en un diagrama de voltaje de la batería (V) vs tiempo (s).

A continuación, en la Tabla 13 se puede observar la comparación del voltaje medido por el dispositivo de puerta de enlace en contraste con el voltaje medido por un voltímetro instalado previamente en el carro.

Tabla 13. Comparación del voltaje de la batería del vehículo receptado en el servidor de análisis de datos Freematics contra el voltaje medido por un voltímetro previamente instalado en el carro en un lapso de cada 2 segundos.

Comparación de la velocidad tomada por el dispositivo de puerta de enlace contra la velocidad tomada del panel de control del vehículo		
Voltaje receptado en el servidor de análisis de datos (V)	Voltaje tomado del voltímetro del vehículo (V)	Tiempo(s)
14,25	14,18	2
14,38	14,31	4
14,52	14,56	6
14,67	14,64	8
14,78	14,75	10
14,56	14,52	12

Continúa...

14,34	14,28	14
14,23	14,20	16
14,35	14,30	18
14,82	14,88	20
14,44	14,39	22
14,71	14,79	24
14,63	14,65	26
14,74	14,83	28
14,24	14,19	30
14,17	14,25	32
14,38	14,50	34

3.2.4 CARGA DEL MOTOR

En un tiempo recorrido de 00:04:39 con 28 KB recibidos con 601 muestras por minuto a una temperatura de trabajo del dispositivo de 43 °C en la fecha 2021-02-10 se toma la siguiente información representada en la Figura 30 en un gráfico de porcentaje de carga (%) vs tiempo (s), como se puede observar en esta muestra la carga del motor tiene un pequeño descenso de la carga del minuto 18:20:32 a 18:20:35 esto se da ya que se presionó el acelerador a fondo para que la mariposa quede totalmente abierta subiendo de esta manera el voltaje del sensor MAF(Mass Air Flow Sensor), las RPM y los gramos de aire que entran por segundo, disminuyendo la carga del motor de un 90% a un 55%, así se comprueba en tiempo real el funcionamiento del vehículo tenga concordancia con los datos proporcionados por el servidor de análisis de datos.

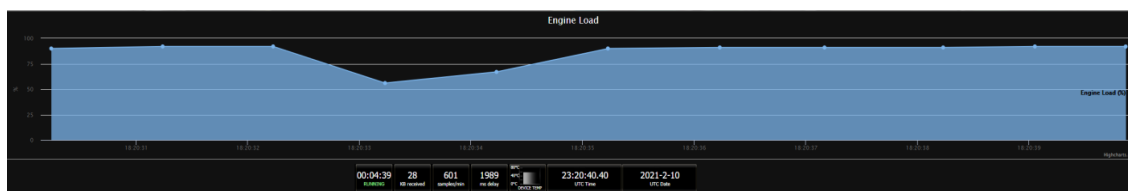


Figura 30. Datos del gráfico de la carga del motor tomados en un lapso de 00:04:39 minutos en un diagrama de carga del motor (%) vs tiempo (s).

3.2.5 GEOLOCALIZACIÓN

En una distancia de 3.8 Km con una duración de 00:01:31 minutos con un promedio de velocidad de 148,7 Km/h se obtiene la localización y las gráficas de altitud y velocidad que se muestran en la Figura 31 con el mapa actualizado y comparado con Google maps para obtener un margen de error de cuatro metros a la redonda desde el punto actual donde se encuentra el vehículo.

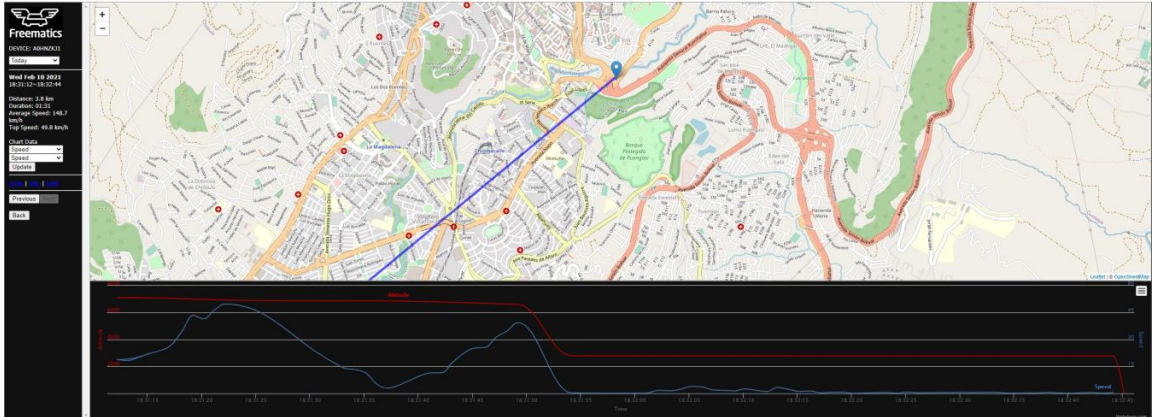


Figura 31. Geolocalización del automóvil con las gráficas de altitud y velocidad en una distancia de 3.8 Km con una duración de 00:01:31 minutos con un promedio de velocidad de 148,7 Km/h.

Con esta pequeña demostración de los resultados obtenidos se elabora una reseña de lo que este dispositivo de puerta de enlace puede ofrecer a los stakeholders.

3.3 SERVICIOS QUE SE PUEDEN OFRECER A LOS USUARIOS A TRAVÉS DEL DISPOSITIVO DE PUERTA DE ENLACE

A través del dispositivo de puerta de enlace se puede ofrecer estos servicios:

- Ubicación en tiempo real
- Comportamiento de conducción
- Alarma de geovalla
- Alarma de movimiento
- Geofencing
- Indicadores de frenados y acelerados bruscos
- Indicadores de velocidad, distracciones
- Nivel de batería, aceite y combustible
- Apagado remoto
- Historial de rutas
- Riesgo de siniestralidad por usuario
- Mantenimiento predictivo
- Localización de la zona de choque

- Plug and play
- Hábitos de manejo

3.4 ARQUITECTURA DE CONEXIÓN DEL DISPOSITIVO DE PUERTA DE ENLACE

A continuación, en la Figura 32 se muestra la arquitectura de conexión del dispositivo de puerta de enlace para saber cómo funciona.

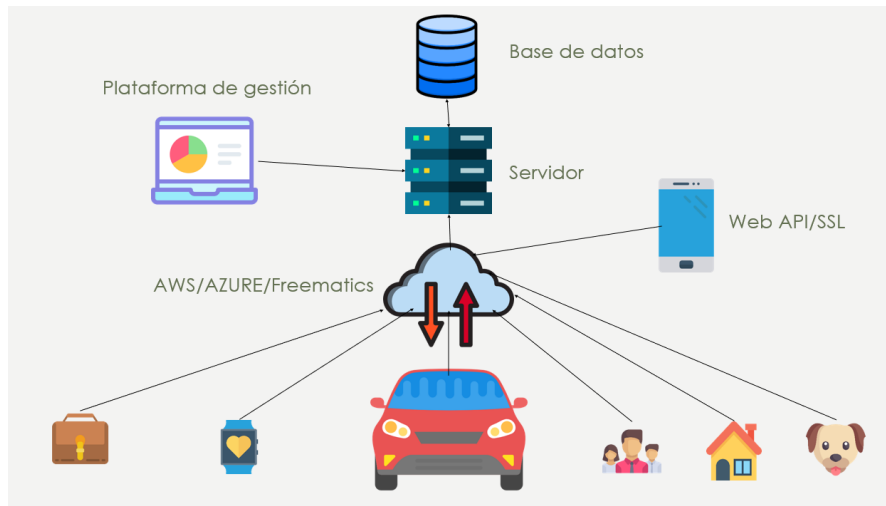


Figura 32. Arquitectura de conexión del dispositivo de puerta de enlace IoT.

Como se puede observar el dispositivo se conecta con la nube y esta envía los datos hacia un servidor o una Web Api para ser analizados.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Mediante varias pruebas de funcionamiento del dispositivo de puerta de enlace conectado al automóvil se comprobó y aseguró la veracidad de la información transmitida al servidor de análisis de datos mediante la comparación de la información del panel de control del automóvil contra la información almacenada en el servidor de análisis de datos, de 30 viajes recorridos con el dispositivo conectado y tomando 600 muestras por minuto en cada uno de ellas se estableció que el 98% de lecturas son correctas teniendo un margen de error del 2% del promedio total de lecturas, a pesar de que los datos del dispositivo de puerta de enlace son mucho más precisos y detallados que los datos que brinda el panel de control del vehículo. Se produce este error cuando el dispositivo queda fuera de una zona de conectividad hasta que se vuelva a restablecer la comunicación para transmitir los datos encolados en tiempo real almacenados anteriormente.
- La batería de respaldo permitió que el dispositivo de puerta de enlace siga funcionando y enviando información al servidor de análisis de datos por un tiempo de 00:21:58 representado un 40% más del tiempo previsto, este exceso se da porque en el tiempo calculado se necesitaba una batería con una carga mínima para satisfacer el tiempo de 15 minutos de 136 mAh y en el mercado la batería que más se ajustaba a las necesidades es una con carga de 300 mAh, esto justifica los 6 minutos con 58 segundos adicionales que permaneció activo el dispositivo.
- El uso del protocolo de comunicación del dispositivo de puerta de enlace hacia el servidor de análisis de datos más eficiente es el de datagramas UDP que permite alimentar datos en vivo con cargas útiles que contienen un grupo de información recopilada y suma de comprobaciones que se transmiten a través de este protocolo ya que ocupa un formato de datos empaquetados basado en texto que junto con los fragmentos de pares PID permite transmitir en tiempo real la data necesaria para analizar cada uno de los parámetros del vehículo.

- La programación del microcontrolador ESP32 permite la adquisición para recolectar todas las variables necesarias por medio de las librerías de Arduino, facilitando de esta manera la obtención de datos del automóvil.

4.2 RECOMENDACIONES

- Desarrollar una aplicación que informe al usuario el estado del vehículo en tiempo real con gráficos sencillos de comprender y alertas tempranas que adviertan futuros posibles daños o algún fallo actual en el sistema del automóvil, teniendo así el control total del vehículo al alcance de su mano, generando de esta manera un valor adicional al cuidado del vehículo evitando accidentes y daños que pueden ser previstos y solucionados antes de que sucedan.
- Añadir un circuito de carga en la batería de respaldo, ya que si esta lleva mucho tiempo sin ser utilizada, cuando el dispositivo de puerta de enlace necesite alimentarse de esta batería, quedará sin funcionar por estar totalmente descargada y perderá toda la telemetría y los datos que se puedan tomar en el transcurso del tiempo en que el automóvil quede desconectado; por lo que es necesario que esta batería esté en constante recarga para que funcione en cualquier momento en el que el carro quede expuesto a un corte de la fuente principal de energía del vehículo.
- Implementar una extensión o un mecanismo que permita conectar al dispositivo de puerta de enlace alejado del puerto OBD con el fin de esconderlo de terceras personas que maliciosamente deseen retirar el dispositivo para robarse el vehículo, con el fin de que el dueño pierda todo rastro, telemetría y localización de donde se encuentra el vehículo. De esta manera no será tan sencillo para una persona retirar el dispositivo haciendo de esta forma que sea mucho más seguro y confiable.
- Buscar un módulo que extienda la posibilidad de conectar el dispositivo de puerta de enlace con más vehículos compatibles con otros

estándares de comunicación interna con la ECU automotriz, de esta manera el dispositivo se podría conectar con transporte pesado que por lo general necesita ser monitoreado todo el tiempo para que la carga llegue a su destino sin ninguna novedad o simplemente expandir la oportunidad que otros vehículos también puedan disfrutar de esta tecnología.

- Instalar en el dispositivo un accionador u alarma que permita conectar al vehículo directamente con el ECU 911 en caso de que sufra algún accidente o suceso que ponga en peligro la integridad del usuario o del vehículo junto con sus activos, para que cada viaje sea más seguro y placentero sabiendo que el automóvil es rastreado y notificado de cualquier eventualidad fortuita que pueda suceder a personas que pueden brindar una pronta ayuda en caso de cualquier emergencia.

5. BIBLIOGRAFÍA

BIBLIOGRAFÍA

- A. Acosta (comunicación personal, 12 de Julio de 2019). Auto Conectado. (D. Gallardo, J. Paz, & I. F. Paz Montero, Entrevistadores)
- AEADE. (18 de Marzo de 2016). *AEADE*. Obtenido de Asociación de Empresas Automotrices del Ecuador: <https://www.aeade.net/perfil-del-sector-automotor-del-ecuador/>
- Alonso, T. P. (2019). *El Automóvil 4.0*. Sevilla: Universidad de Sevilla.
- Arbaiza, P. N. (2010). *Estudio de factibilidad para la creación de una empresa proveedora de tecnología de Telemetría como solución al problema de la recuperación de cartera de las empresas eléctricas del Guayas*. Guayaquil: Universidad Ecotec.
- Arévalo, J. A. (2016). El "Internet de las cosas...". (págs. 24,25). Mexico: Desiderata LAB.
- Ávila, R., & Soto, A. (2013). *Propuesta de implementación de una empresa de control y monitoreo para el mantenimiento preventivo de vehículos, mediante telemetría*. Cuenca: Universidad Politécnica Salesiana.
- Barchilón, M. (01 de Marzo de 2019). *La Vanguardia*. Obtenido de <https://www.lavanguardia.com/vida/junior-report/20190301/46752655177/internet-cosas-dispositivos-conectados-iot.html>
- Bustamante, J. A. (2018). *Diseño de un Taller Automotriz de Mantenimiento Express para Vehículos Livianos en la Ciudad de Guayaquil*. Guayaquil: UIDE.
- Comercio, D. E. (01 de Noviembre de 2019). *ElComercio.com*. Obtenido de <https://www.elcomercio.com/actualidad/parque-automotor-ecuador-crecimiento-decada.html>
- Coriat, B. (2010). *El taller y el robot Ensayos sobre el fordismo y la producción en masa en la era de la electrónica*. México: Siglo xxi.
- Desarrollo, S. N. (14 de Ebero de 2017). *Plan Nacional para el Buen Vivir 2017-2021*. Obtenido de [http://planparatodoscloud.planificacion.gob.ec/objetivo-5#:~:text=Objetivo%20%2D%20Plan%20Nacional&text=Impulsar%](http://planparatodoscloud.planificacion.gob.ec/objetivo-5#:~:text=Objetivo%20%2D%20Plan%20Nacional&text=Impulsar%20)

20la%20productividad%20y%20competitividad,ciudadanos%20sean%20los%20principales%20beneficiarios.

- Dodge, J. M. (2011). *Telemetría usando Redes de Datos de Telefonía Celular*. Barranquilla: NGENIARE, Universidad Libre-Barranquilla.
- E Gomez, J., Castaño, S., Mercado, T., Fernandez, A., & Garcia, J. (2017). *Sistema de internet de las cosas (IoT) para el monitoreo de cultivos protegidos*. Córdoba: Revista Ingeniería e Innovación.
- Ecuador, P. N. (2017). *Rendición de cuentas*. Quito: Ministerio del Interior.
- Evans, D. (2011). *Internet de las cosas Cómo la próxima evolución de Internet lo cambia todo*. New York: CISCO.
- Fernández, R. (2020). *revisión de los dispositivos conectados a Internet a nivel mundial*. Statista.
- Fernando, P. F. (2019). *Implementación de un dispositivo inteligente demovilidad usando IoT para personas con discapacidad visual*. Guayaquil: UNIVERSIDAD DE GUAYAQUIL .
- Formatjé, N. B. (2017). *En 2025 habrá 100.000 millones de móviles conectados a internet gracias al 5G*. Barcelona: Universidad Oberta de Catalunya.
- Gaitán Layza, F. E., Mayorga Farfán, W., Onofre Enero, O., Reynoso Manrique, E. M., & Soto Reyes, J. P. (2018). *Propuesta de un Modelo de Aplicación de IoT y Telemetría en los Procesos de Servicios de Taller para Empresas Concesionarias Automotrices*. Lima: Esan Business.
- García Fernández, J., & Gifre Renom, L. (2019). *Desarrollo de un dispositivo IoT para la adquisición de medidas ambientales*. Madrid: UNIVERSIDAD AUTONOMA DE MADRID.
- González, M. Á. (2019). *Sensores en los sistemas de seguridad del automóvil*. Valladolid: UNIVERSIDAD DE VALLADOLID.
- Granda, A. (28 de Octubre de 2018). *Diario EL TELÉGrafo*. Obtenido de <https://www.eltelegrafo.com.ec/noticias/quito/1/mecanicos-talleres-avalados-ecuador>
- Judicial, P. (2017). *Rendición de cuentas*. Quito: Policía Nacional del Ecuador.
- Kogan, E. (06 de Octubre de 2018). *THE MORNING CALL*. Obtenido de <https://www.mcall.com/hoyla-aut-cuantos-autos-hay-en-el-mundo-y-cuantos-se-fabrican-anualmente-20160923-story.html>

- Lozano Rodriguez, K. V., Marín Castañeda, N. G., & Rodriguez Campos, F. E. (2015). *Sistema de Rastreo de Vehículos de Transporte por Medio de Tecnología GPS*. Torreón: Ciencia, Ingeniería y Desarrollo Tec Lerdo.
- Matute, Á. A. (2013). *Manejo y Optimización de las Operaciones de Mantenimiento Preventivo y Correctivo en un Taller Automotriz*. Guayaquil: Escuela Superior Politécnica del Litoral .
- P, A. A. (12 de Julio de 2019). Auto Conectado. (Ingenesis, Entrevistador)
- Parada, R., Cárdenas Tacoronte, D., Monzo, C., & Meliá Seguí, J. (2017). *Analizador de cobertura de área de Internet of Things (ITHACA) para escenarios topográficos complejos*. Barcelona: MDPI.
- Ramírez, L. M., & Guzmán Ruiz, M. L. (2016). *Desarrollo de Sistema de Seguridad para Automóviles con IOT y Smartphone*. Bogotá: UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS .
- Schwab, K. (2016). *La cuarta revolución industrial*. Barcelona: Debate.
- Superintendencia de Compañías, V. y. (2016). *Breve Análisis Societario del Sector Automotriz del Ecuador Años 2014-2015*. Quito: Superintendencia de Compañías, Valores y Seguros.
- TELÉGRAFO, D. E. (30 de julio de 2017). *eltelegrafo*. Obtenido de <https://www.eltelegrafo.com.ec/noticias/judicial/12/en-2-anos-y-medio-la-policia-registro-el-robo-de-13-271-carros>
- Val Román, J. L. (2016). *Industria 4.0: la transformación digital de la industria*. Deusto: CODOii.