



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN ELECTRÓNICA TELECOMUNICACIONES
Y REDES

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
EXTRACCIÓN Y ALMACENAMIENTO DE PARÁMETROS DE
FUNCIONAMIENTO DEL AUTOMÓVIL EN TIEMPO REAL
BASADO EN LOS PROTOCOLOS DE COMUNICACIÓN OBD-II.”**

Trabajo de titulación

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES**

AUTOR: CRISTHIAN JORGE CARDOSO CARRAZCO

DIRECTORA: Ing. Mónica Andrea Zabala Haro MSc.

Riobamba - Ecuador

2021

©2021, Cristhian Jorge Cardoso Carrazco

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, **CRISTHIAN JORGE CARDOSO CARRAZCO** declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 2021

Cristhian Jorge Cardoso Carrazco

060521578-9

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y
REDES

El Tribunal del Trabajo de Titulación certifica que: El Trabajo de titulación: Tipo: Proyecto Técnico, "**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE EXTRACCIÓN Y ALMACENAMIENTO DE PARÁMETROS DE FUNCIONAMIENTO DEL AUTOMÓVIL EN TIEMPO REAL BASADO EN LOS PROTOCOLOS DE COMUNICACIÓN OBD-II.**", de responsabilidad del señor **Cristhian Jorge Cardoso Carrasco**, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal autoriza su presentación.

NOMBRE	FIRMA	FECHA
Ing. Diego Fernando Veloz Cherrez MSc. PRESIDENTE DEL TRIBUNAL	 Firmado electrónicamente por: DIEGO FERNANDO VELOZ CHERREZ	08/04/2021 <hr/>
Ing. Mónica Andrea Zabala Haro MSc. DIRECTORA DEL TRABAJO DE TITULACIÓN	MONICA ANDREA ZABALA HARO  Firmado digitalmente por MONICA ANDREA ZABALA HARO Fecha: 2021.04.15 08:10:40 -05'00'	08/04/2021 <hr/>
Ing. Paúl Alejandro Montúfar Paz MIEMBRO DE TRIBUNAL	PAUL ALEJANDRO MONTUFAR PAZ  Firmado digitalmente por PAUL ALEJANDRO MONTUFAR PAZ Fecha: 2021.04.15 11:02:43 -05'00'	08/04/2021 <hr/>

DEDICATORIA

A Dios por ser mi guía y apoyo en los momentos difíciles, que es una fuente de amor y sabiduría; A mis padres Luis y Judith que son lo más importante en mi vida, han sido mi apoyo incondicional, son mi mayor ejemplo de dedicación y superación, quienes de una u otra forma me han orientado para conseguir mis metas y ayudarme a convertirme en ser la persona que soy; A mis hermanos Alberto, Ana, María, Vinicio, Marcia, Patricia, Luis, Jaime y Evelyn que, gracias a sus consejos, apoyo y compañía en los momentos difíciles, a mis cuñados por sus consejos, por sus palabras de motivación que me han hecho crecer como persona; A mi amor Dani por la comprensión y por todo el apoyo que día a día me brinda para poder cumplir mi meta y mis sueños.

Cristhian

AGRADECIMIENTO

A Dios por darme la fuerza suficiente, su amor y sabiduría para alcanzar mis objetivos; por permitirme gozar y disfrutar de una buena salud; por haberme guiado en cada uno de mis pasos y así permitirme seguir adelante y no ceder para llegar a donde me encuentro ahora; haberme dado la oportunidad de superarme para llegar a terminar una etapa más de mi vida.

A mi familia por guiarme en cada paso, por el amor que siempre me han demostrado, por el apoyo, confianza y fortaleza que me han brindado para enfrentar las dificultades, que con su ejemplo me han enseñado a siempre salir adelante, a luchar por mis sueños y metas.

A la familia Romero Sarmiento por apoyarme en el transcurso de la carrera, por los consejos que me brindaron, por ayudarme y también corregirme cuando lo necesitaba,

A los docentes por impartir sus conocimientos y apoyo durante este tiempo de formación profesional en la FIE de la Escuela Superior Politécnica de Chimborazo.

A mi tutora de tesis Ing. Mónica Zabala por ser una excelente docente y persona, por permitirme recurrir a su apoyo guiarme con su experiencia, los cuales fueron esenciales para culminar de manera triunfante el presente trabajo.

A mis amigos y compañeros, con quienes hemos compartido diferentes momentos amenos, por compartir sus conocimientos y ayuda en momentos buenos y malos.

A todos quienes me ayudaron de forma directa e indirecta, quienes con sus experiencias y enseñanzas me brindaron la oportunidad de formarme como persona y profesionalmente.

Cristhian

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE ANEXOS.....	xv
ÍNDICE DE ABREVIATURAS.....	xvi
RESUMEN.....	xviii
ABSTRACT.....	xix
INTRODUCCIÓN.....	1
CAPÍTULO I	
1 DIAGNÓSTICO DEL PROBLEMA.....	2
1.1 Antecedentes.....	2
1.2 Formulación del Problema.....	4
1.3 Justificación Teórica.....	4
1.4 Justificación Aplicativa.....	5
1.5 Objetivos.....	6
1.5.1 <i>Objetivo General</i>	6
1.5.2 <i>Objetivos Específicos</i>	6
CAPÍTULO II	
2 MARCO TEÓRICO.....	7
2.1 Historia de la Electrónica en el automóvil.....	7
2.2 Sistema de Diagnóstico a Bordo (OBD).....	8
2.2.1 <i>Sistema OBD-I</i>	8
2.2.2 <i>Sistema OBD-II</i>	9
2.2.3 <i>SISTEMA OBD-3</i>	10
2.3 Sensores del Automóvil.....	10
2.3.1 <i>Sensor RPM del motor</i>	11
2.3.2 <i>Sensor de temperatura del líquido refrigerante</i>	11
2.3.3 <i>Sensor MAP (Presión absoluta del colector de admisión)</i>	12

2.3.4	<i>Sensor de Temperatura del aire de Admisión</i>	12
2.3.5	<i>Sensor de Posición del Acelerador</i>	12
2.3.6	<i>Sensor de Velocidad del Automóvil</i>	12
2.4	Protocolos de Comunicación	12
2.5	Modos de Medición OBD-II	15
2.5.1	<i>Modo 01: Solicitud de datos de diagnóstico actuales.</i>	17
2.5.2	<i>Modo 02: Acceso a cuadros de datos congelados.</i>	17
2.5.3	<i>Modo 03: Obtención de los códigos de falla.</i>	18
2.5.4	<i>Modo 04: Borrado de códigos de falla y valores almacenados.</i>	18
2.5.5	<i>Modo 05: Resultado de pruebas de los sensores de oxígeno.</i>	18
2.5.6	<i>Modo 06: Resultados de las pruebas otros sensores.</i>	18
2.5.7	<i>Modo 07: Muestra de códigos de falla pendientes.</i>	19
2.5.8	<i>Modo 08: Control de funcionamiento de componentes.</i>	19
2.5.9	<i>Modo 09: Información del automóvil.</i>	19
2.6	Compatibilidad del Protocolo OBDI	19
2.6.1	<i>Vehículos Compatibles</i>	19
2.7	Lector de Códigos OBD-II (ELM327)	20
2.7.1	<i>Tipos de Comunicación del Interfaz OBD-II (ELM327)</i>	22
2.7.1.1	<i>Comunicación USB - RS232</i>	22
2.7.1.2	<i>Comunicación inalámbrica</i>	22
2.7.2	<i>ELM327 v1.5 Vs ELM327 v2.1</i>	25
2.8	Tarjeta de desarrollo	25
2.8.1	<i>Tarjetas Raspberry Pi 3 modelo B</i>	25
2.9	Servidor Web	26
2.9.1	<i>Introducción</i>	26
2.9.1.1	<i>Servidor HTTP Apache2</i>	26
2.9.1.2	<i>Lenguaje PHP</i>	27
2.9.2	<i>Gestor de base de datos</i>	27
2.9.2.1	<i>Gestor MySQL</i>	28

2.10	Cloud Computing	28
2.10.1	Amazon Web Services (AWS)	28
2.10.1.1	<i>Características de Amazon Web Services</i>	29
2.10.2	Microsoft Azure	30
2.10.2.1	<i>Características de Microsoft Azure</i>	31
2.10.3	Google Cloud Platform	31
2.10.3.1	<i>Características de Google Cloud Platform</i>	32
2.11	Técnica de ponderación	32
2.12	Telefonía móvil celular	36
2.12.1	Estándar GSM (Global System for Mobile communications)	37
2.12.2	Entramado GSM	38
2.12.3	Tarjeta SIM (Subscriber Identify Module)	38
2.12.4	Tecnología GPRS	38
2.12.4.1	<i>Características de la tecnología</i>	38
2.12.4.2	<i>Servicios de GPRS</i>	39
CAPÍTULO III		
3	DISEÑO Y PROTOTIPO	40
3.1	Marco metodológico	40
3.2	Características del sistema	41
3.2.1	Especificaciones de alcance	41
3.2.1.1	<i>Prototipo del sistema</i>	41
3.3	Elección de dispositivos	43
3.3.1	Tarjeta Raspberry Pi 3 modelo B	43
3.3.2	Interprete de protocolos OBD-II (ELM327 v2.1) Bluetooth	44
3.3.3	Adaptador Bluetooth versión 4.0 (Dongle USB)	44
3.3.4	Módulo GPS UBLOX NEO-M8N-0-01	45
3.3.5	Módulo GSM/GPRS (Huawei EC303C)	46
3.3.6	Escenario	47
3.3.6.1	<i>Modelos de vehículos en análisis</i>	47

3.3.6.2	<i>Bluetooth</i>	48
3.3.6.3	<i>Protocolo RFCOMM de Bluetooth</i>	48
3.3.6.4	<i>Protocolo OBD-II</i>	49
3.3.6.5	<i>Extracción de datos brutos</i>	49
3.4	Diseño lógico del Sistema	50
3.4.1	<i>Etapa de Transmisión</i>	50
3.4.2	<i>Etapa de Recepción</i>	51
3.5	Diseño físico del Sistema	52
3.5.1	<i>Algoritmo de conexión del adaptador OBD-II a la Tarjeta Raspberry Pi</i>	52
3.5.2	<i>Etapa de Transmisión</i>	53
3.5.2.1	<i>Configuración de la Raspberry Pi</i>	53
3.5.2.2	<i>Configuración de la librería Bluez de Bluetooth</i>	56
3.5.2.3	<i>Conexión del Adaptador Bluetooth y ELM327</i>	56
3.5.2.4	<i>Pruebas de lectura de datos con ELM327</i>	57
3.5.2.5	<i>Configuración del módulo GPS</i>	58
3.5.2.6	<i>Prueba de funcionamiento del módulo GPS</i>	59
3.5.3	<i>Etapa de Recepción</i>	60
3.5.3.1	<i>Interfaz Web</i>	60
3.5.3.2	<i>Software</i>	60
3.5.3.3	<i>Gestor de Base de Datos</i>	62
3.5.3.4	<i>Lenguaje Python</i>	64
CAPÍTULO IV		
4	PRUEBAS Y RESULTADOS DEL PROTOTIPO	66
4.1	Pruebas de comprobación	66
4.1.1	<i>Características de los automóviles seleccionados</i>	66
4.2	Pruebas de funcionamiento	67
4.2.1	<i>Escenario 1: Chevrolet Aveo Family</i>	67
4.2.2	<i>Escenario 2: Hyundai i10</i>	74
4.2.3	<i>Escenario 3: Chevrolet Aveo Family</i>	80

4.3	Guía de usuario	90
4.3.1	<i>Especificaciones del terminal móvil.....</i>	90
4.3.2	<i>Modo de Instalación</i>	90
	CONCLUSIONES.....	92
	RECOMENDACIONES.....	94
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-2.	Protocolos según el fabricante	14
Tabla 2-2.	Protocolos según el modelo OSI para el estándar OBD-II	15
Tabla 3-2.	Modos de trabajo del estándar OBD-II	16
Tabla 4-2.	Vehículos Compatibles con OBD-II	20
Tabla 5-2.	Protocolos soportados por ELM327 v1.5 vs v2.1	25
Tabla 6-2.	Recursos Informáticos	33
Tabla 7-2.	Almacenamiento	33
Tabla 8-2.	Bases de Datos	34
Tabla 9-2.	Seguridad e Identidad	34
Tabla 10-2.	Redes y Conectividad	34
Tabla 11-2.	Nivel gratuito	35
Tabla 12-2.	Pago por uso del Servicio	36
Tabla 13-2.	Selección por ponderación	36
Tabla 1-3.	Principales características de la Raspberry Pi 3 modelo B	43
Tabla 2-3.	Características del Módulo GPS UBLOX NEO-M8N-0-01	46
Tabla 3-3.	PIDs del Modo 01 a ser extraídos	49
Tabla 1-4.	Características informativas de automóviles probados con el prototipo	67
Tabla 2-4.	Reporte de los datos extraídos - Escenario 1	73
Tabla 3-4.	Reporte de los datos extraídos - Escenario 2	79
Tabla 4-4.	Reporte de los datos extraídos - Escenario 3	84
Tabla 5-4.	Cálculo de tráfico de envío de datos	86
Tabla 6-4.	Cálculo de datos a ser almacenados	86
Tabla 7-4.	Velocidad de las tecnologías móviles	87
Tabla 8-4.	Costo plan básico	89
Tabla 9-4.	Costo plan premium	89

ÍNDICE DE FIGURAS

Figura 1-1.	Arquitectura de la solución propuesta.....	5
Figura 1-2.	Conector ALDL.....	8
Figura 2-2.	Esquema del sistema OBD-II del vehículo.....	9
Figura 3-2.	Estructura del conector de diagnóstico OBD-II.....	10
Figura 4-2.	Sensores del motor de un vehículo.....	11
Figura 5-2.	Trama estándar OBD-II.....	15
Figura 6-2.	Estructura típica del mensaje en protocolo CAN.....	15
Figura 7-2.	Estructura típica del mensaje en protocolo J1850 e ISO.....	16
Figura 8-2.	Estándar DTC.....	18
Figura 9-2.	Comandos AT Generales para OBD.....	21
Figura 10-2.	ELM327 Serial.....	22
Figura 11-2.	ELM327 Bluetooth.....	23
Figura 12-2.	ELM327 Wi-Fi.....	24
Figura 13-2.	Tarjeta Raspberry Pi 3 modelo B.....	25
Figura 14-2.	Estructura de la Raspberry Pi 3 modelo B.....	26
Figura 15-2.	Motores de Almacenamiento de MySQL.....	28
Figura 16-2.	Plataforma Amazon Web Services.....	29
Figura 17-2.	Servicios de almacenamiento en la nube en AWS.....	29
Figura 18-2.	Plataforma Microsoft Azure.....	30
Figura 19-2.	Plataforma Google Cloud.....	31
Figura 20-2.	Esquema del Sistema Celular.....	37
Figura 21-2.	Estructura de la trama GSM.....	38
Figura 1-3.	Diagrama de principios de diseño de Pahl y Beitz.....	40
Figura 2-3.	Diagrama de bloques del sistema de extracción de datos.....	41
Figura 3-3.	EML327 mini v2.1 Bluetooth.....	44
Figura 4-3.	Dongle Bluetooth v4.0.....	45
Figura 5-3.	Módulo GPS UBLOX NEO-M8N-0-01.....	45
Figura 6-3.	Módulo GSM/GPRS Huawei EC303C.....	46
Figura 7-3.	Chevrolet Aveo.....	47
Figura 8-3.	Hyundai i10.....	48
Figura 9-3.	Diagrama lógico del Sistema de extracción de parámetros OBD-II.....	50
Figura 10-3.	Diagrama de bloques de la etapa de transmisión.....	50
Figura 11-3.	Formato del mensaje.....	51
Figura 12-3.	Diagrama de bloques de la etapa de recepción.....	51

Figura 13-3.	Algoritmo de conexión del ELM327 y la Raspberry Pi 3	52
Figura 14-3.	Entorno SD Card Formatter	53
Figura 15-3.	Entorno Win32 Disk Imager	54
Figura 16-3.	Escritorio Raspberry Pi 3	54
Figura 17-3.	Interfaz de conexión de la Raspberry Pi 3.....	55
Figura 18-3.	Interfaz del programa VNC Viewer	55
Figura 19-3.	Interfaz de Raspbian mediante VNC.....	55
Figura 20-3.	Instrucciones de instalación Bluez	56
Figura 21-3.	Listado de puertos USB.....	56
Figura 22-3.	Selección del adaptador Bluetooth.....	56
Figura 23-3.	Configuración serial del ELM327 Bluetooth	57
Figura 24-3.	Asignación del puerto serie a OBD-II.....	57
Figura 25-3.	Lectura y configuración del ELM327 mediante minicom	57
Figura 26-3.	Prueba de lectura de los sensores del vehículo.....	58
Figura 27-3.	Configuración de los puertos.....	58
Figura 28-3.	Visualización de los puertos en uso con ls - lsusb	58
Figura 29-3.	Configuración del puerto serial ttyAMA0.....	59
Figura 30-3.	Modo de conexión del módulo GPS a la Raspberry Pi	59
Figura 31-3.	Prueba de funcionamiento de recepción de datos GPS	59
Figura 32-3.	Página principal del servidor Apache.....	61
Figura 33-3.	Página de inicio de la plataforma de monitoreo de datos.....	62
Figura 34-3.	Página de logeo de PhpMyAdmin.....	63
Figura 35-3.	Gestor de datos de PhpMyAdmin	63
Figura 36-3.	Diagrama de flujo del programa principal	65
Figura 1-4.	Ubicación del conector DLC en un Aveo Family	67
Figura 2-4.	Solicitud de velocidad del vehículo.....	68
Figura 3-4.	Ruta recorrida (Google Earth Pro) – Escenario 1.....	68
Figura 4-4.	Altura y revoluciones del vehículo en tiempo real – Escenario 1	69
Figura 5-4.	Velocidad y temperatura del vehículo en tiempo real – Escenario 1	69
Figura 6-4.	STF y LTF del vehículo en tiempo real – Escenario 1.....	69
Figura 7-4.	Visualización de los datos en tiempo real – Escenario 1.....	70
Figura 8-4.	Coordenadas y Altura (msnm) del vehículo – Escenario 1	70
Figura 9-4.	RPM y velocidad del vehículo – Escenario 1.....	70
Figura 10-4.	Temperatura y STF del vehículo – Escenario 1	71
Figura 11-4.	LTF del vehículo – Escenario 1	71
Figura 12-4.	Representación de los datos obtenidos en el servidor – Escenario 1.....	72
Figura 13-4.	Solicitud de las revoluciones del vehículo	74

Figura 14-4.	Ruta recorrida (Google Earth Pro) – Escenario 2.....	74
Figura 15-4.	Altura y revoluciones del vehículo en tiempo real – Escenario 2	75
Figura 16-4.	Velocidad y temperatura del vehículo en tiempo real – Escenario 2	75
Figura 17-4.	Representación del STF y LTF del vehículo en tiempo real – Escenario 2	75
Figura 18-4.	Visualización de los datos en tiempo real – Escenario 2.....	76
Figura 19-4.	Coordenadas y Altura (msnm) del vehículo – Escenario 2	76
Figura 20-4.	RPM y velocidad del vehículo – Escenario 2.....	76
Figura 21-4.	Temperatura y STF del vehículo – Escenario 2	77
Figura 22-4.	LTF del vehículo – Escenario 2	77
Figura 23-4.	Representación de los datos obtenidos en el servidor – escenario 2	78
Figura 24-4.	Solicitud de la temperatura del vehículo	80
Figura 25-4.	Ruta recorrida (Google Earth Pro) – Escenario 3.....	80
Figura 26-4.	Altura y revoluciones del vehículo en tiempo real – Escenario 3	80
Figura 27-4.	Velocidad y temperatura del vehículo en tiempo real – Escenario 3	81
Figura 28-4.	STF y LTF del vehículo en tiempo real – Escenario 3.....	81
Figura 29-4.	Visualización de los datos en tiempo real – Escenario 3.....	81
Figura 30-4.	Coordenadas y Altura (msnm) del vehículo – Escenario 3	82
Figura 31-4.	RPM y velocidad del vehículo – Escenario 3.....	82
Figura 32-4.	Temperatura y STF del vehículo – Escenario 3	82
Figura 33-4.	LTF del vehículo – Escenario 3	83
Figura 34-4.	Representación de los datos obtenidos en el servidor – Escenario 3.....	83
Figura 35-4.	Cobertura operadora Movistar en la ciudad de Riobamba	87
Figura 36-4.	Cobertura operadora Claro en la ciudad de Riobamba.....	88
Figura 37-4.	Cobertura operadora CNT en la ciudad de Riobamba.....	88
Figura 38-4.	Dispositivo móvil.....	90
Figura 39-4.	Módulo GPS en marcha	91
Figura 40-4.	Módulo GPRS en marcha.....	91
Figura 41-4.	Módulo Bluetooth en marcha.....	91

ÍNDICE DE ANEXOS

- ANEXO A:** PIDs de los modos de operación de OBD-II.
- ANEXO B:** Comandos AT de OBD-II.
- ANEXO C:** Hoja de especificaciones técnicas tarjeta Raspberry Pi 3 model B.
- ANEXO D:** Hoja de especificaciones técnicas módulo GPS UBLOX NEO M-8N.
- ANEXO E:** Configuración en PHP de las páginas web de la plataforma de extracción de datos.
- ANEXO F:** Configuración del programa principal en Python.

ÍNDICE DE ABREVIATURAS

OBD	On Board Diagnostic
ECU	Engine Control Unit
TIC	Tecnología de la Informática y la Comunicación
CAN	Controller Area Network
AWS	Amazon Web Services
GSM	Global System for Mobile communications
GPRS	General Packet Radio Service
GPS	Global Positioning System
CARB	California Air Resources Board
MIL	Malfunction Indicator Light
DTC	Diagnostic Trouble Codes
ECM	Engine Control Module
ALDL	Assembly Line Diagnostic Link
SAE	Society of Automotive Engineers
EPA	Environmental Protection Agency
DLC	Data Link Connector
MCU	Main Control Unit
ASR	Anti-Slip Regulation
ABS	Anti-lock Braking System
EGR	Exhaust Gas Recirculation
VAN	Vehicle Area Network
PSA	Peugeot Société Anonyme
PWM	Pulse Width Modulation
VPW	Variable Pulse Modulation
CSMA	Carrier Sense Multiple Access
PID	Process Identification
PCM	Power Train Control Module
SBC	Single-Board-Computer
ASF	Apache Software Foundation
PHP	Hypertext Pre-Processor
TCP	Transmission Control Protocol
SMS	Short Message Service
TDMA	Time-Division Multiple Access
SIM	Subscriber Identify Module

RFCOMM	Radio Frequency Communication
HTTP	Hypertext Transfer Protocol
NMEA	National Marine Electronics Association
VNC	Viewer Network Control
USB	Universal Serial Bus
PDF	Portable Document Format
CSV	Comma-Separated Values
GPIO	General Purpose Input/Output
LLC	Logical Link Control
RLC	Radio Link Control
MAC	Medium Access Control

RESUMEN

En el presente proyecto de titulación se diseñó un sistema de extracción de parámetros de funcionamiento vehicular basados en el protocolo de comunicación OBD-II y almacenamiento en la nube, para ello se investigó los protocolos de comunicación que soporta el estándar OBD-II, además sobre las tecnologías relacionadas con el almacenamiento y monitoreo en tiempo real. Posteriormente, una vez tomado en cuenta todas las consideraciones técnicas y los requerimientos del usuario se desarrolló el prototipo. El prototipo posee dos etapas, las cuales cumplen funciones específicas y en conjunto permiten un correcto funcionamiento de todo el sistema. La etapa de transmisión compuesta por una tarjeta de desarrollo Raspberry Pi 3, un módulo GPS UBLOX NEO M-8-N-0-01, un módulo GPRS Huawei E303C, un Dongle Bluetooth v4.0 y un lector ELM327 v2.1, contenidos dentro de una caja para facilitar su portabilidad, esta etapa es la encargada de adquirir los datos de los sensores del vehículo mediante el lector OBD-II y con la ayuda del Sistema de Posicionamiento Global (GPS) la adquisición de coordenadas y altura, estos datos fueron procesados con el lenguaje Python en la Raspberry Pi y enviados en tiempo real mediante Internet utilizando la red GPRS de una operadora móvil. La etapa de recepción es básicamente el servidor web, el cual es el encargado de almacenar y visualizar los datos recibidos, creando una página web que permita mostrar los parámetros solicitados en tiempo real mediante medidores y graficas dinámicas. Una vez implementado el prototipo se realizaron pruebas de funcionamiento en tres escenarios simulando ambientes reales, escenarios donde la aplicación es posible por el tráfico que permite la variación de datos de los sensores del vehículo, además la variación en la altura, así determinando su rendimiento y fiabilidad al comparar los datos de los sensores recolectados con los del tablero del vehículo.

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERIA>, <TELECOMUNICACIONES>, <DIAGNÓSTICO VEHICULAR>, <PARÁMETROS DE FUNCIONAMIENTO>, <PROTOCOLOS OBD-II>, <LECTOR ELM327 (HARDWARE)>, <SISTEMA DE POSICIONAMIENTO GLOBAL (GPS)>, <TARJETA DE DESARROLLO (HARDWARE)>.



ELIZABETH
FERNANDA AREVALO
MEDINA



0864-DBRAI-UPT-2021

ABSTRACT

In the current research project, a system for extracting vehicle operating parameters based on the OBD-II communication protocol and cloud storage was designed. For this, the communication protocols supported by the OBD-II standard were investigated and the technologies related to storage and real-time monitoring. Afterward, the prototype was developed once all the technical considerations and user requirements had been taken into account. The prototype has two stages, which fulfill specific functions and correct the entire system's correct operation. The transmission stage is made up of a Raspberry Pi 3 development card, a UBLOX NEO M-8-N-0-01 GPS module, a Huawei E303C GPRS module, a Bluetooth v4.0 Dongle, and an ELM327 v2.1 reader. All of these components are inside a box for easy portability. This stage is responsible for acquiring the data from the vehicle's sensors through the OBD-II reader and coordinating coordinates and height using the Global Positioning System (GPS). These data were processed with the Python language on the Raspberry Pi and sent in real-time through the Internet using a mobile operator's GPRS network. The reception stage is the web server, which is in charge of storing and viewing the received data, creating a web page that allows the requested parameters to be displayed in real-time using meters and dynamic graphs. Once the prototype was implemented, functional tests were carried out in three scenarios simulating natural environments. In these scenarios, the application is possible due to the traffic that allows the variation of data from the vehicle's sensors as well as the variation in height. So, it was possible to determine its performance and reliability when comparing the collected sensor data with those from the vehicle dashboard.

KEYWORDS: <TECHNOLOGY AND ENGINEERING SCIENCE>, <TELECOMMUNICATIONS>, <VEHICLE DIAGNOSTIC>, <PERFORMANCE PARAMETERS>, <OBD-II PROTOCOL>, <ELM327 READER (HARDWARE)>, <GLOBAL POSITIONING SYSTEM (GPS)>, <DEVELOPMENT CARD (HARDWARE) >.

INTRODUCCIÓN

El trabajo de investigación que se propone se basa en la problemática del monitoreo de datos de los sensores de un automóvil en tiempo real a través de una interfaz web, este proceso se ve afectado por el elevado costo que conlleva el contratar de estos servicios privados, para lo cual como solución se plantea: En el Capítulo 1 se realiza una investigación acerca de proyectos basados en esta problemática, en donde solamente logran la adquisición de forma local, también realizan simuladores de las Unidades de Central Electrónica, motivo por el cual se propone crear un sistema de extracción y almacenamiento de datos de los sensores del vehículo en tiempo real, además de su visualización en un entorno web. El Capítulo 2, se desarrolla la investigación relacionada con los dispositivos de comunicación del vehículo y dispositivos para la extracción de los parámetros, investigación y selección del sistema de almacenamiento web que cumpla con las necesidades del proyecto en cuestión, así también el estudio del estándar OBD-II y sus modos de operación. El Capítulo 3, se basa en el marco metodológico, el cual se centra en el diseño lógico y físico del sistema de extracción, de la selección de los dispositivos a ser utilizados en la construcción del prototipo, configuración y montaje de los dispositivos, configuración de la etapa de transmisión y recepción del sistema, pruebas de funcionamiento de cada uno, además un guía práctico para la configuración del sistema. Y el Capítulo 4, se realizan escenarios de prueba para evaluar el funcionamiento y una toma correcta y fiable de los datos en diferentes tipos de rutas o relieves, además en este capítulo se evalúan los resultados del sistema implementado haciendo pruebas de extracción y transmisión de datos hacia el servidor y su visualización en tiempo real, así también un análisis del tráfico enviado, para finalmente concluir y dar algunas recomendaciones con el objetivo de cumplir con todos los requerimientos iniciales del prototipo, además de recomendaciones para proyectos a futuro.

CAPÍTULO I

1 DIAGNÓSTICO DEL PROBLEMA

1.1 *Antecedentes*

Últimamente, la creación de nuevas tendencias tecnológicas en el mundo automovilístico, como, la implementación de nuevos componentes electrónicos, avances de seguridad o interfaces de comunicación que cada vez hacen a los vehículos más confiables, seguros e inteligentes además de los sensores que permiten mantener un control de los parámetros de funcionalidad del vehículo. A partir de ello se sigue investigando y avanzando cada vez más en las redes vehiculares, para las cuales se utilizan dispositivos móviles que analizan y procesan la información obtenida de los vehículos a través de una interfaz OBD-II.(Villén Macías, 2016)

Se han reportados trabajos en torno a la implementación de los sistemas OBD-II orientados a la creación de software mediante la utilización de herramientas de desarrollo, que simulan el funcionamiento de un escáner automotriz.

En el 2009, O. Mansilla, de la Universidad Politécnica de Cataluña, España, presenta el diseño y realización de un sistema On Board Diagnostics (OBD-II). Con el objetivo de realizar comunicación estable con cualquier centralita electrónica (ECU) de cualquier vehículo equipado con OBD-II. (Mansilla, Sellar y De, 2009)

A. Velázquez, de la Universidad de Valladolid presenta La aplicación Android OBD-II, el cual presenta una aplicación para dispositivos con sistema operativo Android en el cual podrán visualizar los parámetros obtenidos por el sistema OBD-II. (Velázquez, 2013)

En el 2012, la investigación de Cabala y Gamec aporta con la implementación de una conexión Bluetooth inalámbrica entre el dispositivo ELM327 que se conecta a una interfaz OBD-II y un terminal móvil, mismo que administra la obtención y recepción de datos del auto. Una vez que la información ha sido recibida, el usuario los puede visualizar en diferentes presentaciones en el terminal móvil, dónde su objetivo primordial es dar a conocer el estado actual del vehículo. (Cabala y Gamec, 2012)

En el 2013, Dale S. L. Dolan, Taufik Taufik y Matthew Ducasse. En el Simposio y Exposición Mundial de Vehículos Eléctricos realizado por la IEEE en Barcelona, España. Presentan una Cuantificación de la variabilidad en el uso detallado de la energía en viajes repetidos en el voltio

de Chevrolet, el cual controla para ilustrar y comparar los efectos del comportamiento de conducción en el uso de energía del vehículo. (Dolan, Taufik y Matthew, 2013)

En el 2014, R. Malekian, N. Ruth Moloisane, L Nair, BT(Sunil) Maharaj, y Uche A.K. Chude-Okonkwo, Miembros de la IEEE, presentan el Diseño e implementación de un sistema inalámbrico de gestión de flotas OBD-II, con el principal objetivo de medir la velocidad, la distancia y el consumo de combustible de los vehículos con fines de seguimiento y análisis. Además, estos datos se transmiten a través de Wi-Fi a un servidor remoto. (Malekian et al., 2017)

En el 2015, M. Amarasinghe, S. Kottegoda, A. Arachchi, S. Muramudalige, H. M. N. Dilum Bandara y A. Azeez. En la Decimoquinta conferencia internacional sobre avances en las TIC para regiones emergentes (ICTer) realizada en Colombo, Sri Lanka. Presentan un Monitoreo del conductor en la nube y diagnóstico del vehículo con telemática OBD-II. El principal objetivo es el análisis y adquisición de datos vehiculares basado en la nube para el monitoreo del comportamiento del conductor en tiempo real, el análisis de viajes y el diagnóstico de vehículos, a través del puerto de diagnóstico a bordo (OBD) a dongle Bluetooth, una aplicación móvil que se ejecuta en un teléfono inteligente y un servidor basado en la nube. (Amarasinghe et al., 2015)

En el 2017, Yong Xie, Xin Su, Yifan He, Xuhui Chen, Gengliang Cai, Baisheng Xu y Wenjia Ye, en la 16ª Conferencia Internacional sobre Informática y Ciencia de la Información (ICIS) por IEEE / ACIS realizada en Wuhan China. realizar un sistema de adquisición de datos (DAS) basado en STM32, donde los datos del vehículo transferidos en redes CAN se adquieren a través de la interfaz OBD-II (Diagnóstico en todo el espectro). Y luego, los datos del vehículo adquiridos se analizan y analizan preliminarmente de acuerdo con el protocolo OBD-II, y luego se muestran en un visualizador LED. (Xie et al., 2017)

En el 2010, J. Dimaté y P. Gonzáles de la Universidad Pontificia Bolivariana presentan un diseño de una interfaz gráfica en LabVIEW para el diagnóstico de vehículos por medio de OBD-II. Su objetivo principal fue utilizar el Sistema de Diagnostico a Bordo (OBD-II) para diseñar una interfaz gráfica en LabVIEW que permita realizar por parte del usuario, el diagnostico de los diferentes sistemas del vehículo. (Dimaté Cáceres y Castillo, 2010)

En el 2014, J. Ortiz, de la Universidad de San Carlos de Guatemala presenta el diseño de un escáner automotriz OBD-II multiprotocolo, con el principal objetivo de diseñar el escáner automotriz capaz de obtener información almacenada en al ECU del automóvil. Este escáner hace uso del intérprete de protocolos ELM327, utilizando un microcontrolador 18F4550 como unidad central de proceso, teclado y pantalla LCD. (Ortiz, 2014)

En Ecuador se han realizado varias aplicaciones sobre el sistema OBD-II, entre estos tenemos, M. Caizatoa y X. Méndez de la Universidad de las Fuerzas Armadas (ESPE) quienes presentan el diseño e implementación de un prototipo de monitoreo de automóviles empleando el estándar OBD-II (Caizatoa y Ximena, 2014), W. Simbaña de la Escuela Politécnica Nacional de la ciudad de Quito presenta el diseño e implementación de una solución telemática basada en OBD-II que permitan obtener y procesar la información de los sensores del motor de un automóvil (Simbaña, 2015). C. Carpio de la Universidad del Azuay presenta un manual de procedimientos para interactuar entre protocolos de comunicación automotriz. (Carpio, 2013)

1.2 *Formulación del Problema*

¿Cómo se puede diseñar e Implementar un sistema de extracción y almacenamiento de parámetros de funcionamiento del automóvil en tiempo real basado en los protocolos de comunicación OBD-II?

SISTEMATIZACIÓN DEL PROBLEMA

¿Cómo es posible conocer el funcionamiento de un auto estándar y extraer los datos de verificación?

¿Cuáles son los medios de almacenamiento global y en tiempo real que permitan visualizar datos?

¿Cuál es la mejor forma de validación y presentación de datos en tiempo real y con alta disponibilidad?

1.3 *Justificación Teórica*

La necesidad de conocer cuáles son los datos de los sensores de un vehículo en movimiento, además de rastrearlo y disponer de estos datos en tiempo real en cualquier lugar.

Se han desarrollado proyectos de estas características que ayudan a la extracción de datos de un vehículo con OBD-II, para verificar anomalías y fallas con la ayuda de los sensores del vehículo y dispositivos inteligentes externos. Con el desarrollo de la tecnología GPS una de las maneras más utilizadas para obtener datos sobre la ubicación, en este caso el hombre ha salido beneficiado, puesto que inicialmente se utilizó con fines militares y posteriormente se ponga a disposición de la comunidad civil.

Varios proyectos e investigaciones muestran la extracción de datos del comportamiento de cada sensor de forma general, visualizan un despliegue en general de todos los sensores ubicados en el vehículo, mas no solo aquellos necesarios para un estudio de su comportamiento.

Debido al inconveniente mencionado anteriormente se crea la necesidad de implementar un sistema de extracción de determinados datos de un vehículo, además de que existen plataformas de pago, se propone realizar un sistema de acceso gratuito y en tiempo real a los profesionales en el área para obtener la información de los sensores a convenir para el análisis de estos datos.

1.4 Justificación Aplicativa

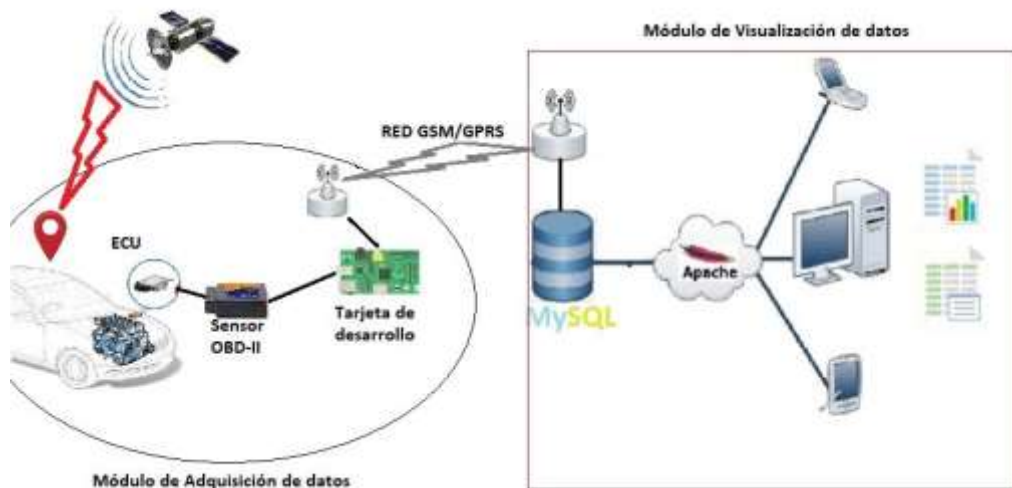


Figura 1-1. Arquitectura de la solución propuesta.

Realizado por: CARDOSO Cristhian, 2019

La motivación principal de este proyecto es llegar a poder diseñar un sistema con el cual poder extraer los datos de los sensores de los vehículos livianos sin tener que recurrir a los costosos servicios oficiales. Esto se debe a que estas marcas tan famosas equipan nuestros coches con sistemas electrónicos capaces de gestionar toda la mecánica y electricidad de que dispone, sin más ayuda que la de un módulo electrónico.

En realidad, estos módulos no son más que una pequeña computadora a la cual si se conoce su funcionamiento se puede acceder, aunque su fabricante intente taparlo, cosa que no les es posible ya que desde hace ya décadas atrás están obligados a implementar un estándar de autodiagnóstico llamado "OBD-II", que se hizo público a consecuencia de las grandes emisiones contaminantes a las que estamos expuestos, implantando así un mecanismo de control de estas.

Estos módulos son el objetivo de este proyecto, ya que al momento que se pueda comunicar con ellos se logrará saber que le ocurre al vehículo y por tanto cuando se tenga que acudir al taller se sabrá, en parte, si nos están estafando, cosa que lamentablemente a veces pasa.

Este proyecto está enfocado a implementar un sistema de extracción de datos en tiempo real de vehículos mediante el protocolo OBD-II, con un prototipo tanto de hardware y software libre.

En el presente trabajo se realizará mediante un Sensor OBD-II el cual estará conectado alámbrica e inalámbricamente por medio de un OBD-II o vía Bluetooth hacia una Raspberry Pi 3 B respectivamente, el cual receptorá y procesará los datos mediante tecnología GPS, el lenguaje de programación que se utilizará es Python, conjuntamente con el módulo GSM/GPRS, que realizará la transmisión de los datos hacia el servidor web de Amazon Web Service donde se alojarán y guardarán los datos que recolecte del mismo, mediante una página web estos datos puedan ser visualizados, por los usuarios, además podrán generar reportes y graficas del comportamiento de los sensores.

El prototipo de extracción de datos de los sensores del vehículo y GPS estará montado dentro de un case donde estarán también el módulo GSM/GPRS y la antena GPS, los cuales estarán alimentados por una batería para sea así un dispositivo que tenga movilidad. Así se obtendrá un sistema de rastreo satelital básico y monitoreo del vehículo en tiempo real a un bajo costo con accesible para los usuarios.

1.5 *Objetivos*

1.5.1 *Objetivo General*

Diseñar e Implementar un sistema de extracción y almacenamiento de parámetros de funcionamiento del automóvil en tiempo real basado en los protocolos de comunicación OBD-II.

1.5.2 *Objetivos Específicos*

- Investigar los parámetros de funcionamiento del protocolo de comunicación OBD-II en vehículos livianos y las tecnologías de almacenamiento disponibles.
- Diseñar el sistema de extracción de parámetros de funcionamiento del automóvil.
- Implementar un sistema de extracción y almacenamiento de los datos brutos del vehículo hacia el servidor.
- Realizar pruebas y verificar resultados del funcionamiento del sistema de extracción los datos y almacenarlos a través de una interfaz web.

CAPÍTULO II

2 MARCO TEÓRICO

En el presente capítulo se explicarán los conceptos teóricos necesarios para la realización del diseño y posteriormente la implementación del prototipo, Así como el estudio de la evolución electrónica en el ámbito automovilístico y los diferentes estándares del diagnóstico a bordo que se han ido creando a raíz de esta evolución. Además de los protocolos existentes, así como los distintos modos de medición del protocolo.

2.1 *Historia de la Electrónica en el automóvil*

Debido a las tendencias del desarrollo del automóvil centradas en materiales ligeros, miniaturización, inteligencia, movilidad, energía y sustentabilidad han ocasionado que la electrónica tome una gran importancia. La evolución de la electrónica de consumo ha dado lugar a requisitos más exigentes para las comunicaciones y las funciones de los vehículos. Por lo que el desarrollo de control de seguridad y comunicaciones y electrónica de entretenimiento pronostican un rápido crecimiento (Saberes y Ciencias, 2017). En vista de esa situación, empresas manufactureras de autos considerarán a la electrónica de automóviles como el factor clave para competir por un mercado de clientes.

Los sistemas automotrices con cada avance sustituyen los sistemas mecánicos a electrónicos, por ejemplo, la ECU es el corazón de un vehículo por a sus múltiples funciones como el control de la combustión; hay una multitud de computadoras en todo el vehículo que controlan diferentes sistemas. En la actualidad un vehículo típicamente contiene alrededor de 25 a 35 microcontroladores, y los de lujo llegan a tener aproximadamente de 60 a 100. (Saberes y Ciencias, 2017)

A finales del siglo XIX Europa fue introducido el automóvil como medio de transporte. Estos primeros vehículos poseían un motor de combustión interna de cuatro tiempos bastante pesado y rudimentario. Mas adelante, Daimmler ideó una variante más ligera el cual sería el precursor de todos los motores a explosión posteriores. Con los años, los automóviles fueron incorporando nuevas actualizaciones que aumentaron su rendimiento y mejoraron sus prestaciones. Estas mejoras incluían el uso de diferencial, correas, baterías, etc., pero en su diseño, el motor de combustión interna no experimento cambios substanciales y será en el siglo XX cuando se introduzca la electrónica. (García, 2004)

2.2 Sistema de Diagnóstico a Bordo (OBD)

En abril de 1985 el CARB (California Air Resources Board), aprobó una regulación para un sistema de diagnóstico a bordo u OBD, (Simbaña, 2015). Esta regulación que se aplica a los automóviles vendidos en el estado de California a partir de 1988 especifica que el Módulo de Control de Motor (ECM) debe monitorizar ciertos componentes del vehículo relacionados con las emisiones de gases para asegurar un correcto funcionamiento, y que se ilumine una lámpara Indicadora de Fallo o MIL en el cuadro de instrumentos cuando se detecte un problema. El sistema OBD también aporta un sistema de Códigos de Error de Diagnóstico (DTC) y unas tablas de errores en los manuales de reparación los cuales ayudan a los técnicos mecánicos a determinar las causas más probables de avería en el motor y problemas en las emisiones. (Villén Macías, 2016)

2.2.1 Sistema OBD-I

Los Sistemas de Diagnóstico a Bordo u OBD se encuentran en la mayoría de los automóviles y vehículos ligeros actuales. Durante la década de los 70 y principio de los 80 se introdujeron componentes electrónicos para cumplir los estándares de emisión de gases, posteriormente la implantación de sistemas OBD para controlar funciones del motor y diagnosticar problemas generó una mayor complejidad en la electrónica integrada en los vehículos.

El sistema OBD inicialmente se denominó ALDL, el cual posee doce pines, de estos se utilizan 9, tal como se muestra en la **Figura 1-2**. Cada uno de estos pines tiene funciones específicas usando un protocolo no estandarizado.

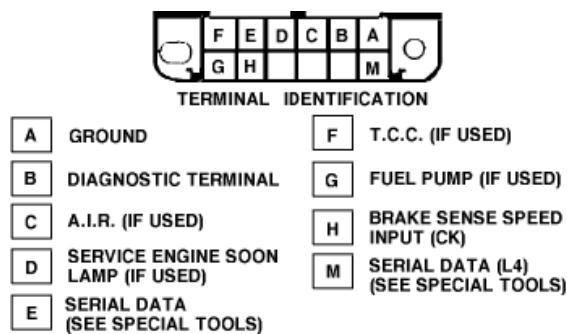


Figura 1-2. Conector ALDL

Fuente: <https://acortar.link/vM483>

Inicialmente hubo varios estándares y cada fabricante tenía sus propios sistemas y códigos (Salas, 2018). En 1988 la Sociedad de Ingenieros de Automoción (SAE) definió un conector estándar OBD de 16 pines y un conjunto de códigos de diagnóstico, cada pin tiene una función específica usando un protocolo no estandarizado, adoptando así la EPA la mayoría de los estándares y recomendaciones SAE sobre aplicaciones OBD. Posteriormente, con OBD-II, un conjunto más

amplio de estándares y sistemas también definidos por la SAE y adoptado por la EPA y el CARB es aprobado para su implementación el 1 de enero de 1996. (Villén Macías, 2016)

2.2.2 Sistema OBD-II

El sistema OBD-II comenzó a ser de forma obligatoria por los nuevos vehículos en los Estados Unidos desde 1996, cuyo objetivo era monitorear los componentes que afecten el sistema de control de emisiones de gases contaminantes, además de medir sus parámetros en tiempo real como: temperaturas, presiones, velocidades entre otros. (Simbaña et al., 2016)

Los componentes del sistema OBD-II son: la ECU conocida como la computadora del automóvil, los transductores encargados de enviar los datos hacia la ECU, la luz indicadora de fallas ubicada en el tablero (MIL) y el conector de diagnóstico (DLC) que sirve de interfaz entre la ECU y los dispositivos de diagnóstico automotriz. (Villén Macías, 2016)

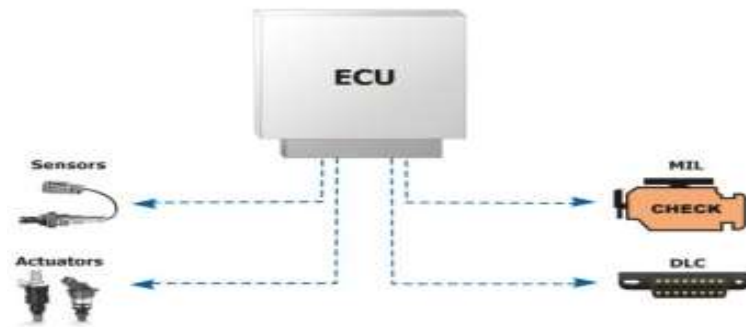


Figura 2-2. Esquema del sistema OBD-II del vehículo.
Fuente: VILLEN José, 2016.

MIL: Este indicador también conocido como check engine light, tiene tres tipos de presentación:

- **Destellos ocasionales:** Este indicativo se presenta si el defecto de funcionamiento es momentáneo. Cuando el defecto es de naturaleza más grave el parpadeo es más continuo.
- **Destellos constantes:** Este indicativo permite determinar que existe un problema que puede causar un daño serio al motor si éste no es reparado inmediatamente. En este caso se detiene el registro de las lecturas de los sensores (denominados *freeze frame*) y se almacenan las últimas lecturas en la memoria del ordenador central del vehículo (MCU – *Main Control Unit*).
- **Destello fijo:** Este indicador se presenta cuando existe problemas muy graves, y permanece activo mientras el vehículo esté funcionando y no se corrija el problema, siendo necesario en algunos casos el vaciado de la memoria de averías.

DLC: Éste es el conector de diagnóstico mediante el cual es posible conectar el coche a un equipo de diagnóstico externo. Actualmente se trata de un conector de 16 pines denominado de manera estandarizada J1962 o conector OBD-II. (Bernias, 2015)



Figura 3-2. Estructura del conector de diagnóstico OBD-II.
 Fuente: BERNIAS Gonzalo, 2015.

2.2.3 SISTEMA OBD-3

OBD-3 es la tercera generación del OBD-II, es un programa diseñado específicamente para minimizar el borrado entre la detección de un mal funcionamiento en el sistema de emisiones por el OBD-II y la reparación del vehículo. Comunica las fallas de manera satelital, puesto que posee un radio comunicador que puede directamente las fallas a la agencia de emisiones (EPA), de esa manera el dueño o usuario puede pedir asistencia de manera remota. (Coronel, 2018)

2.3 Sensores del Automóvil

Para el monitoreo y control de la gran cantidad de sistemas de gestión del motor es necesario el uso de sensores, puesto que es casi imposible realizar esta tarea sin la ayuda de estos, para controlar los sistemas de seguridad y confort (ASR, ABS, airbag, ajuste del cinturón de seguridad, aire acondicionado, etc.). (BlogdeAutomoviles.com, [sin fecha])

Los sensores son dispositivos encargados de monitorear de forma continua el funcionamiento y operación del motor del automóvil, mediante la conversión de una magnitud física (temperatura, revoluciones por minuto (RPM), presión, calidad de aire, nivel de oxígeno, etc.) en magnitud eléctrica. Para la interpretación de señales existe sensor un circuito de adaptación que convierte las señales del sensor en una señal entendible por la unidad de control. (Simbaña, 2015)

Los sensores son importantes, puesto que se usan para registrar de manera precisa los estados reales del motor en funcionamiento, tales como la presión de aceite del motor, la temperatura del motor, el número de revoluciones, temperatura del líquido refrigerante del motor, presión absoluta del colector de admisión, presión barométrica, temperatura del aire de admisión, posición del acelerador, velocidad del automóvil. (Simbaña, 2015)

La **Figura 4-2** indica los principales sensores del motor del automóvil que pueden ser monitoreados por el sistema OBD-II.

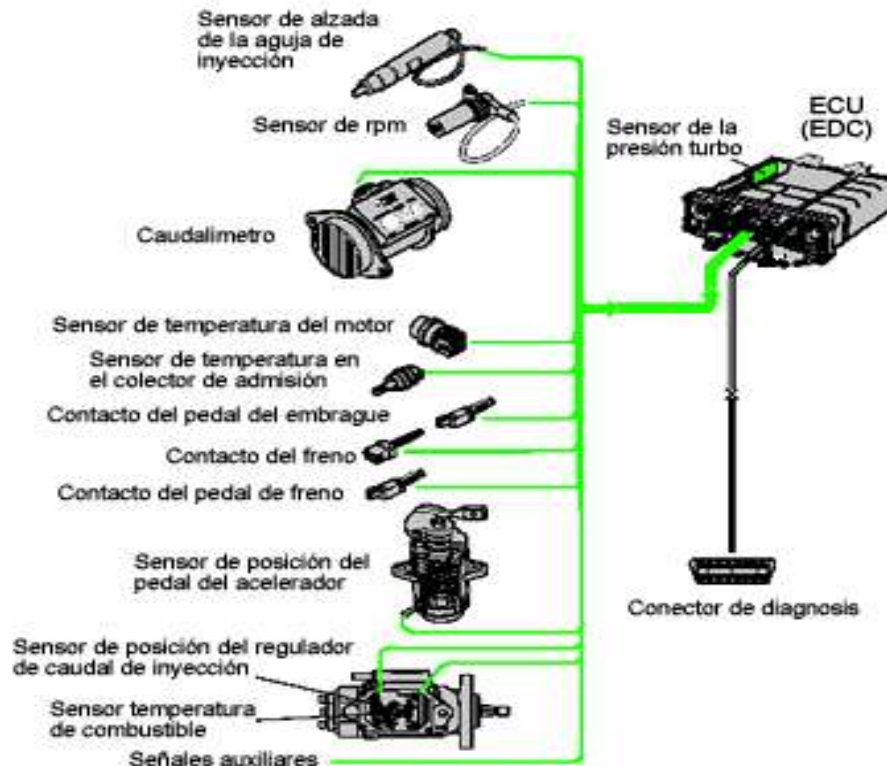


Figura 4-2. Sensores del motor de un vehículo.
Fuente: <https://n9.cl/a29t>

2.3.1 *Sensor RPM del motor*

La función principal es controlar la velocidad en revoluciones del motor, así como los puntos de cambio de la transmisión.

2.3.2 *Sensor de temperatura del líquido refrigerante*

Este sensor informa a la ECU la temperatura del refrigerante del motor para que este a la vez permita el cálculo de entrega de combustible, así también la sincronización del tiempo y el control de la válvula EGR, además la activación y la desactivación del ventilador del radiador.

2.3.3 *Sensor MAP (Presión absoluta del colector de admisión)*

Obtiene la información acerca de los cambios en la presión atmosférica, en el vacío del motor y en el múltiple de admisión, así enviando una señal a la ECU para que este controle el tiempo de ignición y ajustar la mezcla de aire-combustible en las diferentes de carga del motor y altitud sobre el nivel del mar.

2.3.4 *Sensor de Temperatura del aire de Admisión*

Monitorea la temperatura del aire de entrada, la cual posteriormente es enviada a la ECU, con esta señal la realiza ajustes en la mezcla y la duración del pulso del inyector para así determinar la densidad del aire.

2.3.5 *Sensor de Posición del Acelerador*

Provee datos de la posición de la mariposa del cuerpo de aceleración, enviando la información necesaria para que la ECU calcule el pulso del inyector y la curva de avance del encendido, además, regula el flujo de los gases de emisión del escape a través de la válvula EGR.

2.3.6 *Sensor de Velocidad del Automóvil*

La medida voltajes recolectados por este sensor son enviados a la ECU, donde la información recibida es interpretada como: La velocidad de la marcha mínima, el embrague del convertidor de torsión, información para que marque la velocidad en el tablero eléctrico digital, para la función del sistema de control de la velocidad crucero.

2.4 *Protocolos de Comunicación*

Este protocolo es un conjunto de reglas y normas establecidas que permiten una comunicación exitosa entre dos o más dispositivos para el intercambio de información (Ortiz, 2014). En el sistema OBD-II, estos protocolos permiten la comunicación e intercambio de mensajes en forma bidireccional, entre una herramienta de diagnóstico (escáner automotriz) y la ECU del automóvil. (Simbaña, 2015)

Distintos fabricantes de automóviles desarrollan sus propios protocolos, aunque no existe una compatibilidad entre sí, para lo cual, de los varios protocolos existentes, se detallará aquellos que destacan en su aplicación en el mundo del automóvil.

- **Protocolo CAN:** Este protocolo fue diseñado por Robert Bosch y se perfila hoy en día como el líder de mundial en sistemas multiplexados, siendo el que posee mayor aceptación por parte de los fabricantes. En cuanto a sus características, se destaca:

- Alta velocidad de transmisión de hasta 1Mbps.
- Amplio soporte de componentes electrónicos.
- Eficacia y seguridad en su funcionamiento.
- **Protocolo VAN:** Fue desarrollado por el grupo PSA (Peugeot Société Anonyme) y Renault, este protocolo actualmente en desuso por los fabricantes, debido que, ha sido sustituido por el protocolo CAN, el protocolo presenta características como:
 - Es simple y económico.
 - Su velocidad de transmisión es de 259 Kbps.
 - Es el protocolo más eficiente.
- **Protocolo SAE J1850:** Este protocolo fue desarrollado en Estados Unidos, es utilizado por fabricantes como Ford, Chrysler o General Motors. Entre sus características tenemos las siguientes:
 - Su velocidad de transmisión es de 41,6 Kbps.
 - Es apto para aplicaciones de velocidad baja y media.
 - Es la tecnología más económica, gracias al soporte aportado por General Motors y Ford, así como importantes suministradores de componentes (Intel, Motorola, Texas, etc.) determinando su rápida expansión y abaratamiento.

Debido que a veces los protocolos utilizados para la comunicación entre las unidades electrónicas del vehículo (red de comunicación interna del vehículo) no es el mismo utilizado para la conexión con una herramienta de diagnóstico externa. Este tipo de protocolo se origina de dos organizaciones: ISO, SAE. Los protocolos estandarizados por ISO se dividen en tres: ISO 15765-4 (CAN), ISO 9141-2, ISO 14230-4 (KWP2000, Keyword Protocol 2000). En cambio, los protocolos de SAE se dividen en dos: SAE J1939 y SAE J1850 (PWM (Pulse Width Modulation) y VPW (Variable Pulse Modulation)). (Ortiz, 2014)

ISO 9141-2

Este protocolo de comunicación es el más antiguo de todos y fue definido por ISO en 1989, está basado en la comunicación serie asíncrona representado el bit 0 con cero voltios y el bit 1 con 12 voltios. La velocidad de transmisión es de 10400 baudios. Es utilizado por Chrysler, automóviles europeos y asiáticos.

ISO 14230-4

Protocolo de comunicación en serie asíncrona con velocidades de transmisión de 1200 a 10400 baudios. También llamado KWP2000. Este protocolo utiliza la capa física del modelo OSI para redes computacionales, así como la capa de sesión en términos de inicialización, establecimiento y finalización de la comunicación. La capa de este protocolo es idéntica a la del protocolo ISO9141-2.

ISO 15765-4

Este protocolo desarrollado por Bosh permite la comunicación entre dispositivos sin una computadora host, la velocidad máxima de transferencia es de hasta 1Mbit/s o 1000000 baudios en redes con distancia menor a los 40 metros de distancia, al disminuir la velocidad de transmisión aumenta la distancia entre dispositivos.

SAE J1939

Protocolo basado en CAN (Controller Area Network), originalmente utilizado en la industria de camiones pesados, actualmente este se utiliza en vehículos a diesel. SAE J1939 ha sido adoptado por muchos fabricantes de motores a diesel por la creciente demanda de control de emisiones de gases.

SAE J1850

Protocolo de comunicación estandarizado de arquitectura abierta, de bajo costo, generalmente utilizado en vehículos terrestres de carrera y todo terreno. Este protocolo tiene dos variantes: modulación por pulso variable (VPW), cuya modulación en donde la señal tiene un periodo variable para representar un bit 0 y 1. VPW es utilizado para hacer la transición de bit 1 a 0 y de 0 a 1, 64 y 128, además tiene una velocidad de transmisión de 10400 baudios y es el protocolo estándar para General Motors Company. (Ortiz, 2014)

PWM es la técnica de modulación en la cual se modifica el ciclo de trabajo de una señal periódica para transmitir por un canal de comunicación. PWM posee una velocidad de transmisión de 41,7 Kbps, además, para asegurar la integridad de los datos posee (CSMA), este es utilizado por los nodos de una red para detectar y verificar la ausencia de tráfico en el canal de comunicación, antes de poder iniciar la transmisión, PWM es el protocolo estándar para Ford Motor Company. (Ortiz, 2014)

Tabla 1-2. Protocolos según el fabricante

PROTOCOLO	FABRICANTE DEL AUTOMÓVIL
ISO 9141-2/KWP2000-4	Varios automóviles asiáticos y europeos incluyendo Alfa Romeo, Audi, BMW, Citroen, Fiat, Honda, Hyundai, Jeep post-2004, Kia, Land Rover, Mazda, Mercedes, Mitsubishi, Nissan, Peugeot, Renault, Saab, Skoda, Subaru, Toyota, Vauxhall, Volkswagen post-2001, Volvo pre-2004
J1850 PWM	Ford pre-2003. Jaguar pre-2003. Mazda pre-2004.
J1850 VPW	General Motors, Chrysler
ISO 15765-4 (CAN)	Ford, Mazda, Volvo desde 2004.

Fuente: <https://n9.cl/z5oa>

Realizado por: CARDOSO Cristhian, 2021. Adaptado de (Ortiz, 2014)

Tabla 2-2. Protocolos según el modelo OSI para el estándar OBD-II

Niveles OSI	OBD-II			
Físico (Nivel 1)	SAE J1850	ISO 9141-2	ISO 14230-1	ISO 11898 ISO 15765-4
Enlace (Nivel 2)	SAE J1850	ISO 9141-2	ISO 14230-1	ISO 11898 ISO 15765-4
Red (Nivel 3)	ISO 11898 ISO 15765-4
Transporte (Nivel 4)
Sesión (Nivel 5)
Presentación (Nivel 6)	ISO 15765-4
Aplicación (Nivel 7)	ISO 15031-5	ISO 15031-5	ISO 15031-5	ISO 15031-5

Realizado por: CARDOSO Cristhian, 2021. Citado en: (Ortiz, 2014)

2.5 Modos de Medición OBD-II

OBD-II puede trabajar en **diez** modos de medición, estos modos son comunes en todos los vehículos y permiten desde registrar datos para su verificación, extraer códigos de averías, borrarlos y realizar pruebas dinámicas de actuadores. Los modos en que se presentan la información se hallan estandarizados, al igual que el tamaño de la trama OBD-II. (Beltrán, 2015)

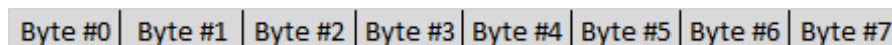


Figura 5-2. Trama estándar OBD-II

Realizado por: CARDOSO Cristhian, 2021. Citado en: (Beltrán, 2015)

El contenido de cada “Byte” dependerá de si la trama es de petición o de respuesta y el modo en el que se esté trabajando. La excepción es el Byte #0, que indica el número de bytes que contienen información OBD-II de la trama en cuestión, (Sánchez, 2017). La existencia de este byte dependerá del protocolo sobre el que se envía el mensaje, ya que de los cinco protocolos que define el estándar OBD-II el protocolo CAN soporta ocho bytes de datos. (Beltrán, 2015)

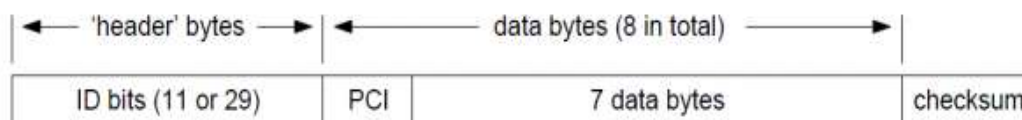


Figura 6-2. Estructura típica del mensaje en protocolo CAN

Realizado por: CARDOSO Cristhian, 2021. Citado en: (Caizatoa y Ximena, 2014)

El protocolo J1850 e ISO utilizan el mismo formato de mensaje, el cual está compuesto por 3 bytes de cabecera al inicio de la trama, hasta 7 bytes de datos y 2 bytes de checksum al final de la trama. En la **Figura 7-2**. Se puede visualizar las partes del formato del mensaje.

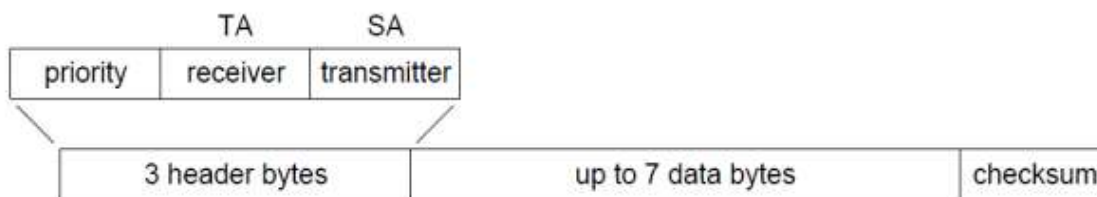


Figura 7-2. Estructura típica del mensaje en protocolo J1850 e ISO

Realizado por: CARDOSO Cristhian, 2021. Citado en: (Simbaña, 2015)

La principal función de cada uno de los parámetros del mensaje se detalla a continuación:

- Priority: Proporciona información sobre la prioridad del mensaje.
- Receiver (TA, Target Address): Indica la dirección destino a donde va dirigido el mensaje.
- Transmitter (SA, Source Address): Indica la dirección de donde se origina el mensaje.
- Data: Lleva la información que se va a enviar, ya sea un parámetro o petición.
- Checksum: Detecta los errores que pueden haber sido introducidos durante el envío del mensaje y en caso de detectarlos solicita retransmisión de este.

Dependiendo de la información a la que se desee acceder, se necesita un modo diferente. Una vez dentro de ese modo de trabajo, se ofrece un extenso número de parámetros para acceder a dicha información. Se incluyen los siguientes modos. (Meseguer, 2012)

Tabla 3-2. Modos de trabajo del estándar OBD-II.

MODO	CARACTERÍSTICAS
Modo 01	Identificación de Parámetro (PID). Es el acceso a datos en tiempo real de valores analógicos o digitales de salidas y entradas a la ECU (Engine Control Unit).
Modo 02	Acceso a Cuadro de Datos Congelados. La ECU toma una muestra de todos los valores relacionados con las emisiones en el momento exacto de ocurrir un fallo.
Modo 03	Permite extraer de la memoria de la ECU todos los códigos de fallo (DTCs (Códigos de Diagnóstico de Error)) almacenados. Estos códigos ayudan al usuario a determinar la causa fundamental de un problema.
Modo 04	Se pueden borrar todos los códigos almacenados en la PCM (Power Train Control Module) en la ECU incluyendo los DTCs y el cuadro de datos grabado, este modo también apaga el indicador luminoso MIL.
Modo 05	Devuelve los resultados de las pruebas realizadas a los sensores de oxígeno para determinar el funcionamiento de estos y la eficiencia del convertidor catalítico, el cual es vital para el control de las emisiones del vehículo y para el correcto funcionamiento de este.
Modo 06	Permite obtener los resultados de todas las pruebas de abordó.
Modo 07	Permite leer de la memoria de la ECU todos los DTCs pendientes.

Continuación de la **Tabla 3-2**

Modo 08	Realiza pruebas de actuadores, en el cual puede activar o desactivar actuadores como bombas de combustible, válvula de relentí, entre otros actuadores del sistema automotriz.
Modo 09	Modo opcional, no todos los vehículos lo soportan, permite obtener información del vehículo como el número de serie, numero de calibrado y verificación, información de la ECU, etc.
Modo 0A	Permite solicitar códigos de problemas permanentes.

Fuente: BERNIAS Gonzalo, Aplicación para la monitorización y diagnosis de automóviles.

Realizado por: CARDOSO Cristhian, 2021. Citado en: (Bernias, 2015)

Al acceder a cada modo, la respuesta a la petición será en hexadecimal, para interpretar la información se deberá realizar la conversión a decimal y luego aplicar formulas establecidas por el estándar para obtener el valor deseado. Estos datos se conocen como PID (Parameter ID), un PID es un código de solicitud de informaciones cual es enviado a la ECU según el modo OBD-II. Para cada modo de operación sus PID son estandarizados, dependiendo del modo y del PID las respuestas serán interpretadas de distintas formas. (Ortiz, 2014)

Para la obtención de PID se sigue el siguiente proceso:

- Introducir el PID
- Envío del PID por el protocolo de comunicación (detección automática por el OBD-II)
- Obtención de la información al ser reconocido el PID por la ECU
- Envío de la información solicitada por medio del protocolo de comunicación

2.5.1 Modo 01: Solicitud de datos de diagnóstico actuales.

Este modo contiene información en tiempo real de todos los sensores que cuenta equipado el automóvil. Al realizar la petición de información de algún PID de este modo, este devolverá un valor en hexadecimal, el cual se debe convertir a decimal y aplicar la fórmula correspondiente para obtener el resultado en las dimensionales ya estandarizadas para el PID. (Ortiz, 2014)

La información que se puede obtener por medio de este modo incluye: estado de combustible, temperatura del refrigerante del motor, presión del múltiple de admisión, RPM del motor, velocidad, temperatura del aire de admisión, posición absoluta del acelerador, distancia recorrida con la luz indicadora de mal funcionamiento (MIL) encendida, entre otras. (Simbaña, 2015, p. 17)

2.5.2 Modo 02: Acceso a cuadros de datos congelados.

El modo 2 accede a los datos que contienen algún fallo ocurrido en el automóvil, los cuales se encuentran almacenados en la memoria de la ECU, su funcionamiento es similar al modo 1, la diferencia radica en que se acceden a datos pasados y no en tiempo real. (Simbaña, 2015, p. 17)

2.5.3 Modo 03: Obtención de los códigos de falla.

Este modo permite el acceso a todos los códigos de falla (DTC, *Data Trouble Codes*) almacenados en la memoria de la ECU, estas fallas se producen cuando la lectura de algún sensor está fuera de rango con respecto a los valores óptimos que se encuentran grabados en la memoria de la ECU. En el caso de no existir fallos se recibe una respuesta nula o un mensaje indicando que no existen DTCs almacenados. (Simbaña, 2015, p. 17)

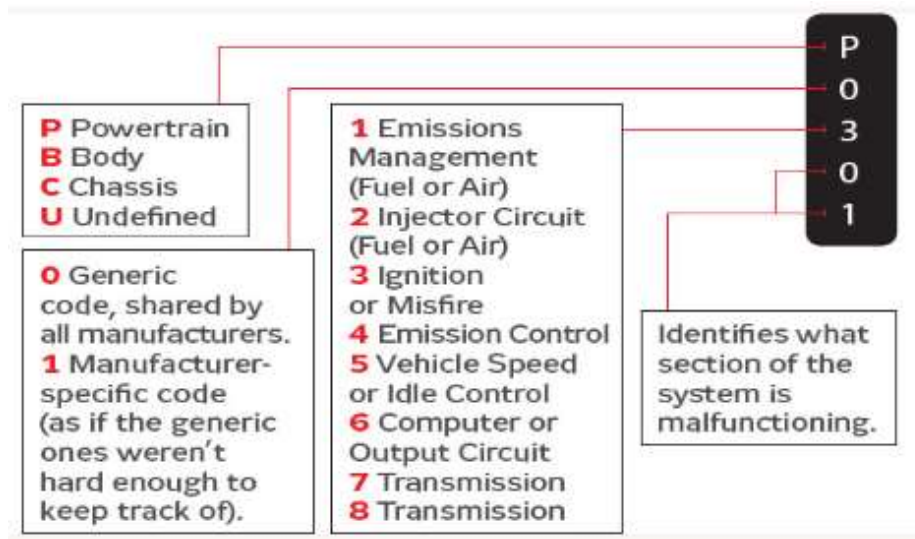


Figura 8-2. Estándar DTC

Fuente: (Villén Macías, 2016)

2.5.4 Modo 04: Borrado de códigos de falla y valores almacenados.

Con este modo se pueden borrar todos los códigos de falla almacenados en la memoria de la ECU del automóvil, una vez ejecutado el borrado, los registros de la memoria de la ECU quedan en cero y la luz MIL se apaga. En el caso de no corregir correctamente el problema, los códigos de falla volverán a aparecer al igual que la luz MIL se encenderá nuevamente. (Simbaña, 2015, p. 18)

2.5.5 Modo 05: Resultado de pruebas de los sensores de oxígeno.

El modo de operación 05 devuelve los resultados de las pruebas realizadas a los sensores de oxígeno para verificar el funcionamiento de estos y la eficiencia del convertidor catalítico, un dato vital para el control de emisiones del vehículo. (Villén Macías, 2016, p. 28)

2.5.6 Modo 06: Resultados de las pruebas otros sensores.

Este modo permite obtener todos los resultados del sistema completo, además los monitores del sistema identifican que sistemas pueden ser monitoreados y en el caso de que se pueda, el estado de estos. (Villén Macías, 2016, p. 28–29) Las pruebas más comunes que se efectúan son las siguientes:

- Fallas de combustión (referente a cables, bujías, encendido, inyectores, etc.).
- Sistema de combustible (sensores de oxígeno, sensores de ciclo cerrado, retroalimentación de combustible y sistema de encendido).
- Catalizador.
- Calentamiento del catalizador.
- Sistema evaporativo.
- Sistema secundario de aire.
- Refrigerante del aire acondicionado.
- Sensores de oxígeno.
- Calentamiento del sensor de oxígeno.
- Recirculación de gases de escape (EGR).

2.5.7 Modo 07: Muestra de códigos de falla pendientes.

Este modo permite leer todos los códigos de falla pendientes que no hayan sido reparados o borrados previamente, que se encuentran almacenados en la memoria de la ECU. (Villén Macías, 2016, p. 29)

2.5.8 Modo 08: Control de funcionamiento de componentes.

El modo 08 está destinado a la realización de pruebas de actuadores (bombas de combustible, válvula de ralentí) para poder activarlos o desactivarlos, es recomendable que solo el personal autorizado realice el uso de este modo de operación. (Villén Macías, 2016, p. 29)

2.5.9 Modo 09: Información del automóvil.

Este modo es opcional, puesto que no todos los vehículos se encuentran equipados con el mismo, la función principal de este modo es pedir información sobre el vehículo como el número de serie e inclusive información extra sobre el mismo. (Villén Macías, 2016, p. 29) La función de ID del vehículo es aplicable a los vehículos del año 2000 y posteriores.

Los detalles de los modos y los PIDs se detallan en el **Anexo A**

2.6 Compatibilidad del Protocolo OBDI

2.6.1 Vehículos Compatibles

El sistema OBD-II escanea vehículos americanos a gasolina y diesel desde 1996, vehículos Europeos, Asiáticos y Japoneses a gasolina y Diesel desde el 2000, de la categoría liviana.

Las Marcas de vehículos soportados por la herramienta de diagnóstico se detalla a continuación.

Tabla 4-2. Vehículos Compatibles con OBD-II

ABARTH	HONDA	NISSAN
ACURA	HUMMER	OPEL
ALFA ROMEO	HYUNDAI	PEUGEOT
ASTON MARTIN	ISUZU	PONTIAC
AUDI	IVECO	PORSCHE
BMW	JAGUAR	RENAULT
BUIK	JEEP	ROVER
CADILLAC	KIA	SAAB
CATERHAM	LADA	SATURN
CITROEN	LANCIA	SEAT
CHEVROLET	LAND ROVER	SECMA
CHRYSEL	LEXUS	SKODA
DACIA	LINCOLN	SMART
DAEWOO	LOTUS	SSANGYONG
DAIHATSU	MAHINDRA	SUBARU
DOGDE	MASERATI	SUZUKI
FAW	MAZDA	TATA
FERRARI	MERCEDES BENZ	TOYOTA
FIAT	MERCURY	VAUXHALL
FORD	MG	VOLKSWAGEN
GMC	MINI	VOLVO
GREAT WALL	MITSUBISHI	
HOLDEN	MORGAN	

Fuente: <https://n9.cl/z5oa>

Realizado por: CARDOSO Cristhian, 2021. Citado en: (Bernias, 2015)

2.7 Lector de Códigos OBD-II (ELM327)

El lector de códigos OBD (ELM327) es un hardware que permite la lectura de los datos de la ECU para posteriormente ser transmitidos hacia algún dispositivo de procesamiento a través de una interfaz, estos pueden ser por: puerto serial, puerto USB o inalámbrico.

Para lograr una comunicación con la ECU y el OBD-II es necesario el uso de comandos AT, para poder realizar las distintas peticiones a la ECU, ELM327 usa comandos con números hexadecimales. A continuación, se muestra una lista con los principales comandos AT para ELM327.

General Commands

<CR>	repeat the last command
BRD hh	try Baud Rate Divisor hh
BRT hh	set Baud Rate Timeout
D	set all to Defaults
E0, E1	Echo off, or on*
FE	Forget Events
I	print the version ID
L0, L1	Linefeeds off, or on
LP	go to Low Power mode
M0, M1	Memory off, or on
RD	Read the stored Data byte
SD hh	Save Data byte hh
WS	Warm Start (quick software reset)
Z	reset all
@1	display the device description
@2	display the device identifier
@3 cccccccccc	store the @2 identifier

OBD Commands

AL	Allow Long (>7 byte) messages
AMC	display Activity Monitor Count
AMT hh	set the Activity Mon Timeout to hh
AR	Automatically Receive
AT0, 1, 2	Adaptive Timing off, auto1*, auto2
BD	perform a Buffer Dump
BI	Bypass the Initialization sequence

OBD Commands

DP	Describe the current Protocol
DPN	Describe the Protocol by Number
FT	Filter for Transmitter off*
FT hh	Filter for Transmitter = hh
H0, H1	Headers off*, or on
IA	Is the protocol Active?
MA	Monitor All
MR hh	Monitor for Receiver = hh
MT hh	Monitor for Transmitter = hh
NL	Normal Length messages*
PC	Protocol Close
R0, R1	Responses off, or on*
RA hh	set the Receive Address to hh
S0, S1	printing of Spaces off, or on*
SH xyz	Set Header to xyz
SH xxyyzz	Set Header to xxyyzz
SH wwxxyyzz	Set Header to wwxxyyzz
SP h	Set Protocol to h and save it
SP Ah	Set Protocol to Auto, h and save it
SP 00	Erase stored protocol
SR hh	Set the Receive address to hh
SS	use Standard Search order (J1978)
ST hh	Set Timeout to hh x 4 msec
TA hh	set Tester Address to hh
TP h	Try Protocol h
TP Ah	Try Protocol h with Auto search

Figura 9-2. Comandos AT Generales para OBD

Fuente: (Elm Electronics Inc., 2014)

Los comandos AT generales disponibles se detallan en el **Anexo B**

Entre las principales características del lector ELM327 se tiene:

- Mantiene el mismo funcionamiento que un escáner automotriz.
- Soporta los 5 protocolos de comunicación OBD-II.
- Trabaja con los 9 modos de medición del sistema OBD-II.
- La velocidad de señal es de 38400 baudios.
- El voltaje de operación es 12V (batería del automóvil).

2.7.1 Tipos de Comunicación del Interfaz OBD-II (ELM327)

2.7.1.1 Comunicación USB - RS232

La comunicación OBD-PC será por medio del puerto serie RS-232, el cual define una interconexión serie entre un dispositivo transmisor de datos y receptor de datos. Básicamente estuvo orientado a conexiones punto a punto, pero se fue introduciendo al entorno industrial para comunicaciones entre captadores y sistemas de adquisición de datos. (Gómez, 2016)



Figura 10-2. ELM327 Serial
Fuente: <https://n9.cl/9ej51>

Protocolos soportados por el OBD-II USB:

- ISO 15765-4 (CAN BUS).
- ISO 14230-4 (KWP2000).
- ISO 9141-2 (Key Word Protocol).
- J1850 VPW (Variable Pulse Width).
- J1850 PWM (Pulse Width Modulation).

2.7.1.2 Comunicación inalámbrica

La comunicación interfaz-PC puede ser de tipo inalámbrica, puesto que ELM327 dispone versiones de dispositivos bluetooth y wifi, además existen aplicaciones gratuitas que pueden ser instaladas en dispositivos móviles como Smartphone y Tablet. Cabe mencionar que en este proyecto se utilizará un dispositivo bluetooth.

ELM327 BLUETOOTH

El adaptador de ELM327 Bluetooth se conecta al vehículo y ordenador / Smartphone / Tablet y deja que ellas se comuniquen entre sí a través de software OBD. Este adaptador Bluetooth también funciona en todas las aplicaciones disponibles en Google Play OBD tienda para dispositivos Android, no compatible con sistema operativo IOS (Apple). Al igual que el dispositivo OBD-II

serial, permite diagnosticar los problemas de los vehículos y leer los datos de los sensores disponibles en tiempo real.



Figura 11-2. ELM327 Bluetooth

Fuente: <https://n9.cl/ioqx7> <https://n9.cl/waazr>

Características del adaptador ELM327 Bluetooth

- Alcance de Bluetooth: 5-15 metros. Clase 2 de transmisión Bluetooth con control de potencia adaptable.
- Velocidad de transmisión: 9600 o 38400 baudios
- Tensión de funcionamiento: 12V
- Corriente nominal de reposo: 45 mA
- Temperatura de funcionamiento: -4° a 131° F (-20° a 55°C)
- Humedad de funcionamiento: 10 a 85% (sin condensación)

Protocolos soportados por el adaptador ELM327 Bluetooth

- J1850 PWM (Vehículos Ford)
- J1850 VPW (Vehículos GM)
- ISO 9141-2 (Asia, Europa, vehículos de Chrysler)
- ISO 14230-4 (Kwp2000)
- ISO 15765-4 (CAN)

ELM327 WIFI

El adaptador ELM327 WIFI permite la exploración y diagnóstico del vehículo en tiempo real, de una manera que el adaptador ELM327 USB y Bluetooth, soporta todos los protocolos de OBD-II, EOBD y CAN. (Morocho, 2017)

El equipo se puede utilizar con PC, Smartphone (Android, IOS) e IPAD mediante conexión inalámbrica Wi-Fi, dentro de las tiendas de los dispositivos existen numerosas aplicaciones compatibles con esta herramienta (Aplicaciones OBD).



Figura 12-2. ELM327 Wi-Fi

Fuente: <https://n9.cl/kx8k5>

Incorpora la última versión que es compatible con la mayoría de los vehículos, a diferencia de las versiones anteriores que no lo son. Su conexión y funcionamiento es muy sencillo, se conecta el adaptador ELM327 Wi-Fi al conector de diagnóstico del vehículo y se procede a instalar la aplicación de mayor interés en tu Smartphone, Tablet, IPAD p PC. El dispositivo permite realizar diferentes operaciones, (Mercado Libre, 2020), entre ellas se tiene:

- Lee códigos de diagnóstico de averías, genéricos y específicos del fabricante.
- Permite borrar los códigos de averías de motor y apagar la luz de avería.
- Muestra los datos de los sensores relacionados con el módulo de motor en tiempo real.

Características del adaptador ELM327 Wi-Fi

- SSID: OBDII
- IP: 192.168.0.10
- Puerto: 35000
- Subred: 255.255.255.0
- Antena: Interna
- WIFI Rango: 50 pies (15 metros aproximadamente)
- Estándar de Wifi: 802.11 a / b / g
- Corriente nominal de reposo: 45 mA
- Temperatura de funcionamiento: -15° a 100° C

Protocolos soportados por el adaptador ELM327 Wi-Fi

- J1850 PWM
- J1850 VPW
- ISO 9141-2
- ISO 14230-4 (Kwp2000)
- ISO 15765-4 (CAN)

2.7.2 ELM327 v1.5 Vs ELM327 v2.1

Protocolos soportados

Tabla 5-2. Protocolos soportados por ELM327 v1.5 vs v2.1

ELM237 v1.5	ELM237 v2.1
SAE J180 PWM (41.6 kbaud)	-
SAE J180 VPW (10.4 kbaud)	-
ISO 9141-2 (5 baud init, 10.4 kbaud)	ISO 9141-2 (5 baud init, 10.4 kbaud)
ISO 141230-4 KWP (5 baud init, 10.4 kbaud)	ISO 141230-4 KWP (5 baud init, 10.4 kbaud)
ISO 141230-4 KWP (fast init, 10.4 kbaud)	ISO 141230-4 KWP (fast init, 10.4 kbaud)
ISO 15765-4 CAN (11 bit ID, 500 kbaud)	ISO 15765-4 CAN (11 bit ID, 500 kbaud)
ISO 15765-4 CAN (29 bit ID, 500 kbaud)	ISO 15765-4 CAN (29 bit ID, 500 kbaud)
ISO 15765-4 CAN (11 bit ID, 250 kbaud)	ISO 15765-4 CAN (11 bit ID, 250 kbaud)
ISO 15765-4 CAN (29 bit ID, 250 kbaud)	ISO 15765-4 CAN (29 bit ID, 250 kbaud)

Fuente: <https://obd2-elm327.es/elm327-version-2.1>

Realizado por: CARDOSO Cristhian, 2021.

Las mejoras que presenta ELM327 v2.1 con respecto a la versión 1.5 es que ofrece todo lo que poseen las versiones anteriores también cuenta con:

- Una memoria de transmisión RS232 de 512 bytes.
- Mejoramiento en velocidad y la fiabilidad.
- Monitor de actividad, que permite ponerse en modo ahorro de energía.
- Parámetros programables y comandos AT.
- Igualación de frecuencias para evitar envíos no deseados CAN durante la búsqueda.
- Capacidad para procesar las respuestas de “Respuestas pendientes”.

2.8 Tarjeta de desarrollo

2.8.1 Tarjetas Raspberry Pi 3 modelo B



Figura 13-2. Tarjeta Raspberry Pi 3 modelo B

Fuente: (Sislema, 2018, p. 39)

Raspberry Pi es una placa de desarrollo reducida o placa simple (SBC) como se muestra en la **Figura 13-2**, es una de las mejores opciones considerando su calidad y precio, posee un sistema operativo de código abierto adaptado a la versión Debian, denominada Raspbian, la se encuentra en constantes actualizaciones. La función principal de esta placa es ser el controlador principal del dispositivo, así como de los módulos de seguimiento.

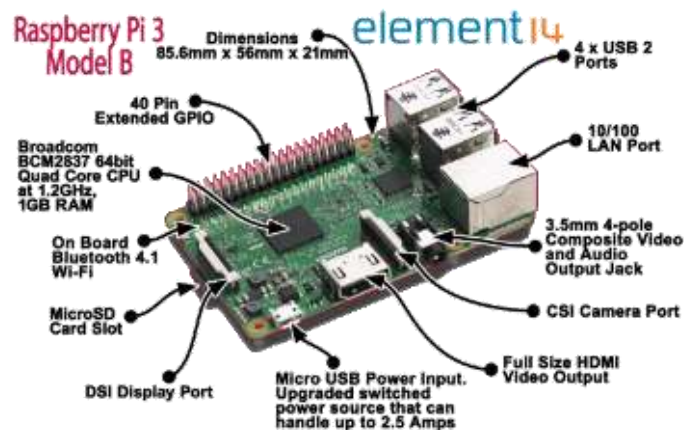


Figura 14-2. Estructura de la Raspberry Pi 3 modelo B
Fuente: <https://n9.cl/epzem>

2.9 Servidor Web

2.9.1 Introducción

Un servidor de datos es un ordenador u otro tipo de equipo informático el cual es el encargado de proporcionar a los clientes quienes pueden ser personas u otros dispositivos conectados a él. La información puede ser múltiple y variada, que va desde un archivo de texto, imágenes, videos y hasta programas de TV informativos, base de datos, entre otros, el rol que desempeña es ser el cerebro principal de un sistema informático, debido que constituye un elemento capaz de almacenar información.

2.9.1.1 Servidor HTTP Apache2

El servidor HTTP Apache2 es un servidor Web de software libre o código abierto desarrollado por la Apache Software Foundation (ASF), Apache se ha convertido en uno de los servidores Web más populares del mundo gracias a que es un software que responde a las solicitudes de los navegadores web, gracias a su alta calidad, además es un servidor robusto y con un ciclo de desarrollo muy rápido. (Mifsuf, 2017, p. 11)

Además, es un servidor estable, eficiente, extensible y multiplataforma.

- **Estable:** Gracias a su robustez se impide caídas o cambios inesperados en el servidor.

- **Flexible y eficiente:** Compatible con la mayor parte de extensiones web que existen en la actualidad, como son los módulos PHP. SSL, CGI, SSI, proxy, entre otros.
- **Extensible:** Su funcionalidad es ampliada de una gran manera gracias a la cantidad de módulos que posee.
- **Multiplataforma:** Se encuentra disponible para diferentes plataformas como GNU/Linux, Windows, MacOS.

En la web existen varios sitios de ofrecen alojamiento web quienes cuentan con aplicaciones adicionales como MySQL y PHP, entre estas tenemos AWS, Hostinger, GoDaddy. (Sistema, 2018, p. 31)

2.9.1.2 *Lenguaje PHP*

PHP (*Personal Home Page tools o Herramientas para páginas iniciales personales*). Es un lenguaje de programación para entornos web que cuenta con funciones muy similares a las de ASP y JSP, el cual es utilizado en servidores Linux para personalizar la información enviada a los usuarios que tienen acceso a un sitio web. Desde el punto de vista técnico, PHP es un lenguaje interpretado de alto nivel, similar en construcciones léxicas y sintácticas a C, C++, Java y Perl, por lo que para quienes ya tienen un conocimiento de estos lenguajes les resultará muy fácil comenzar a escribir código PHP. (Millet, 2013, p. 9–10)

PHP es un lenguaje embebido en páginas HTML, es decir, este es un lenguaje de programación que se introduce dentro de las páginas HTML. El código PHP se interpreta en el lado del servidor web, desde donde se genera la página HTML solicitada antes de efectuar su transmisión hacia el navegador. De esta forma, se puede programar aplicaciones asociadas al servidor web, aumentando así, la funcionalidad de dicho servidor y transformándolo en un sistema de desarrollo de aplicaciones cliente / servidor mucho más completo. Su principal objetivo es hacer que los desarrolladores de aplicaciones basadas en la web puedan escribir páginas que se generan dinámicamente de una forma sencilla y rápida. (Millet, 2013, p. 9–10)

En resumen, PHP gracias a sus constantes actualizaciones mejoraron el rendimiento y prestaciones, con la versión 5, se ha rediseñado completamente para crear un lenguaje completamente OO (Orientado a Objetos), así agilizando su funcionamiento, y extrayendo la compatibilidad con MySQL en un módulo externo. (Millet, 2013, p. 9–10)

2.9.2 *Gestor de base de datos*

Un gestor o servidor de base de datos dota servicios de base de datos a otros programas o computadoras, tal y como es definido por el esquema cliente-servidor. Además de manejar grandes y complejos volúmenes de datos, y al mismo tiempo necesitan compartir información

con un conjunto de clientes de manera confiable y segura. Para lo cual requiere de un SGBD (Sistema de Gestión de Base de Datos), uno de los más conocidos es MySQL el cual será utilizado en el presente proyecto. (Sislema, 2018)

2.9.2.1 Gestor MySQL

MySQL es el gestor de base de datos más popular, de código abierto, confiabilidad, además de ser sencillo y rápido que se adapta perfectamente a entornos en los que el volumen de datos sea el orden de los megabytes. En los últimos años destacadas compañías de software han desarrollado aplicaciones SQL de código abierto (*Open Source*), MySQL se ha convertido en la principal opción de base de datos para aplicaciones basadas en la web, debido que utiliza propiedades web de alto perfil. (Millet, 2013, p. 9–10)

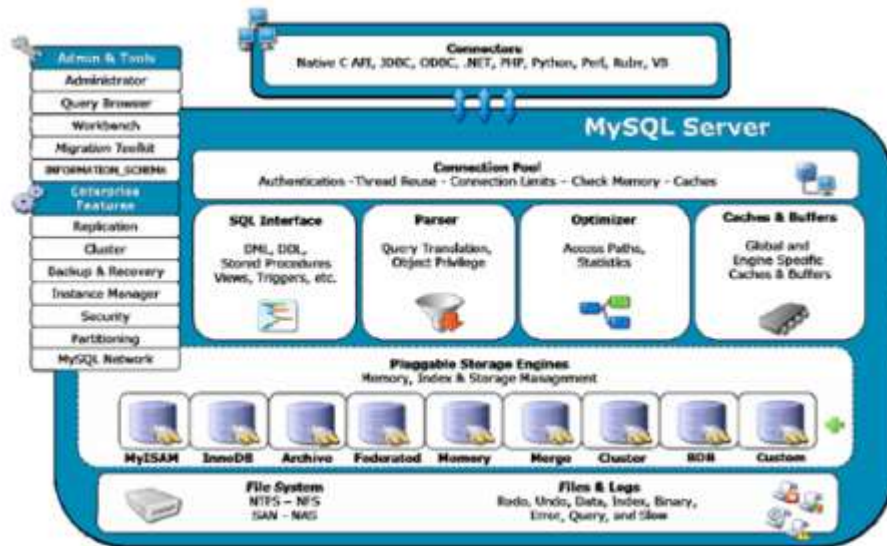


Figura 15-2. Motores de Almacenamiento de MySQL
Fuente: (Sislema, 2018, p. 32)

2.10 Cloud Computing

Es una tecnología que permite acceso remoto a software, almacenamiento de archivos y procesamiento de datos por medio de Internet, siendo esta la alternativa a la ejecución en una computadora personal o servidor local. En el modelo de nube, no hay necesidad de instalar aplicaciones en forma local ya que en su contrato establece varias aplicaciones.

2.10.1 Amazon Web Services (AWS)

Amazon Web Services brinda una plataforma de informática en la nube escalable, con gran disponibilidad y fiabilidad, el cual proporciona herramientas que permiten a los clientes ejecutar una variedad de aplicaciones, además de brindar confianza hacia los clientes, en AWS el proteger

la confidencialidad, integridad y disponibilidad de los sistemas y de los datos es de suma importancia. (AWS, 2017)



Figura 16-2. Plataforma Amazon Web Services
Fuente: <https://www.masip.es/wp-content/uploads/2020/08/aws-logo.png>

La plataforma AWS proporciona servicios unificados. AWS soporta lenguajes de programación como Java, Python, Ruby, Perl, entre otros, además ofrece un servicio web RDS (Relational Database Service) que elimina las tareas de administración de bases de datos.

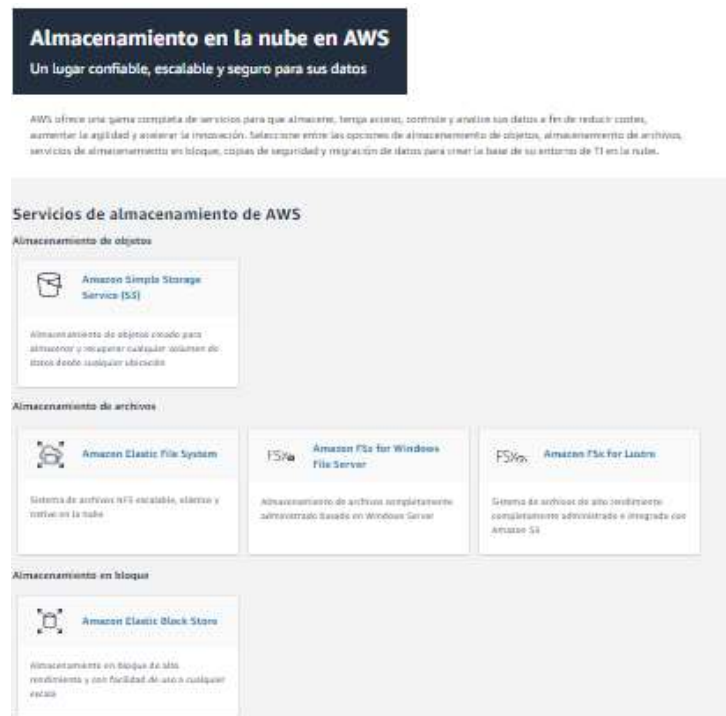


Figura 17-2. Servicios de almacenamiento en la nube en AWS
Fuente: <https://aws.amazon.com/es/products/storage/>

2.10.1.1 Características de Amazon Web Services

- **Computación:** Suministra servicios virtuales en la nube.
- **Almacenamiento:** Almacenamiento escalable y bajo costo.
- **Bases de datos:** Dispone de MySQL, PostgreSQL, Oracle, SQL Server y MariaDB, DynamoDB.

- **Migración:** Se pueden realizar migraciones de servidores en un tiempo de actividad mínimo, incluso transferencias a gran escala.
- **Redes:** Permite tener un servicio web DNS escalable y de alta disponibilidad.
- **Herramientas para desarrolladores:** Permite almacenar códigos en repositorios privados o la interfaz de línea de comando de AWS que sirve para administrar los servicios de AWS.
- **Seguridad, identidad y conformidad:** Administra las claves de usuario y claves de cifrado, aprovisiona, administra e implementa certificados, filtra el tráfico web malintencionado.
- **Análisis:** Extrae conocimientos y estadísticas de los datos.
- **Inteligencia artificial:** Permite crear interfaces de comunicación para apps con voz, texto, imágenes, traductores de texto en habla.
- **Servicios móviles:** Servicios orientadas al desarrollo de aplicaciones móviles.
- **Mensajería:** Envío y recepción de correo electrónico.
- **Internet de las cosas:** Servicio como AWS Greengrass y AWS IoT, con procesamiento de flujos de datos en tiempo real procedentes de dispositivos IoT.
- **Desarrollo de juegos:** Motor de juegos gratuito y multiplataforma.
- **Evaluación gratuita:** Periodo de 12 meses, 85 productos, además de productos gratuitos en forma indefinida.
- **Pago por uso:** Se facturará si se utilizan los servicios bajo esta modalidad.

2.10.2 Microsoft Azure



Figura 18-2. Plataforma Microsoft Azure

Fuente: <https://www.afcsoftware.com/wp-content/uploads/2020/08/Mangaard-azure.png>

Microsoft Azure es la plataforma de aplicaciones para la nube alojada en los data centers de Microsoft, este se caracteriza por su apertura y flexibilidad. Uno de sus puntos fuertes es, precisamente, que es una plataforma que no se limita al conocido entorno de Microsoft, sino que permite desarrollar software en cualquier lenguaje. Además, que permite escalar aplicaciones con facilidad y ofreciendo un autoservicio de manera flexible en función de las necesidades y pagando solamente por los recursos que usa la aplicación. (Olcina Valero, 2017)

2.10.2.1 Características de Microsoft Azure

- **Computación:** Servicios de máquinas virtuales.
- **Redes:** Permite tener un servicio web DNS escalable y de alta disponibilidad.
- **Almacenamiento:** Almacenamiento escalable.
- **Web y móvil:** Servicios para crear e implementar aplicaciones web críticas.
- **Bases de datos:** Dispone de SQL Database y NoSQL
- **Inteligencia + Análisis:** Herramienta de análisis predictivo.
- **Internet de las cosas:** Centro de IoT de Azure, proceso en tiempo real.
- **Seguridad + Identidad:** Protege el acceso a los datos y aplicaciones con un nivel adicional de autenticación.
- **Supervisión + Administración:** Administra y supervisa todos los productos de Azure a través de una consola unificada (Portal de Microsoft Azure).
- **Herramientas para desarrolladores:** Permite almacenar códigos en repositorios privados
- **Evaluación gratuita:** Periodo de 12 meses.
- **Pago por uso:** Se facturará si se utilizan los servicios.
- **Uso de licencia Open:** Dirigido a pequeñas y medianas empresas que compran productos y servicios Microsoft a través de licencias Open.
- **Soporte técnico Developer:** Soporte técnico online, con suscripción de 6 meses, costo \$29/mes.
- **Soporte técnico Estándar:** Incluido soporte telefónico limitado, costo \$100/mes.
- **Soporte Profesional Direct:** Amplia el soporte estándar a un número ilimitado de llamadas y con atención prioritaria, costo \$1000/mes.

2.10.3 Google Cloud Platform

Esta plataforma es la que reúne todas las aplicaciones de desarrollo web que Google ofrecía por separado. Google Cloud Platform permite crear soluciones en la nube y es utilizada en su gran mayoría por arquitectos y desarrolladores de software especializados en Java, Python o REST services. (Olcina Valero, 2017)



Figura 19-2. Plataforma Google Cloud

Fuente: <https://ucloudstore.com/wp-content/uploads/2019/11/gcloud-slider02.png>

Está basada en la infraestructura sobre la que se ejecuta el motor de búsqueda de Google, Youtube, Gmail y otros servicios. Permite la conexión bajo demanda en la nube,

2.10.3.1 *Características de Google Cloud Platform*

- **Recursos informáticos:** Posibilidad de crear máquinas virtuales propias.
- **Almacenamiento y bases de datos:** Almacenamiento escalable y de alto rendimiento.
- **Redes:** Ofrece productos en la red de fibra privada de Google.
- **Big Data:** Almacenamiento de datos totalmente administrativos.
- **Aprendizaje automático:** Suministra servicios rápidos y escalables, con modelos ya creados o a su vez crear modelos a partir de datos.
- **Herramientas para administración:** Supervisión en tiempo real para escalar y diagnosticar problemas.
- **Herramientas para desarrolladores:** Disposición de librerías que ayudaran a desarrollar e implementar las aplicaciones.
- **Identidad y seguridad:** Controla el acceso y visibilidad de los recursos.
- **Evaluación gratuita:** Periodo de 90 días.
- **Pago por uso:** Se facturará si se utilizan los servicios.

2.11 *Técnica de ponderación*

Esta técnica permite determinar imparcialmente los aspectos contrapuestos de una cuestión o el equilibrio entre el peso de dos o más cosas, es decir, tomar la mejor decisión cuando en la argumentación concurren razones justificadoras conflictivas y del mismo valor. (CUELLO y SARDOTH, 1981)

La plataforma de almacenamiento para la implementación del sistema de extracción y almacenamiento de parámetros vehiculares para su visualización en tiempo real con el sistema OBD-II debe cumplir con características y parámetros básicos que se detallan a continuación:

1. Recursos informáticos
2. Almacenamiento
3. Bases de datos
4. Seguridad e identidad
5. Redes y conectividad
6. Nivel gratuito
7. Costo del servicio luego finalizar el periodo gratuito.

Para seleccionar una plataforma de las disponibles en el mercado, se pondera o se asigna un valor de acuerdo con la escala presentada a continuación, la cual refleja la inexistencia o el cumplimiento de los requerimientos necesarios establecidos para la implementación del sistema.

5 Cumple con los requerimientos solicitados respecto al parámetro analizado

0 No cumple con los requerimientos solicitados respecto al parámetro analizado

Este valor de ponderación se aplica para los requisitos:

1. Recursos informáticos
2. Almacenamiento
3. Bases de datos
4. Seguridad e identidad

Tabla 6-2. Recursos Informáticos

<i>Servicios</i>	AWS	Valor	Azure	Valor	Google	Valor
<i>Gestión y mantenimiento de servidores</i>	Nube de computación elástica (EC2)	5	Máquinas virtuales de Azure	5	Motor de cálculo	5
<i>Servidores virtuales privados</i>	LightSail - Vela ligera	5	Imágenes de máquinas virtuales	5	<i>No disponible</i>	0
<i>Servicio gestionado de contenedores</i>	EC2 Servicio de contenedores (ECS)	5	<i>Registro de contenedores de Azure</i>	5	<i>Registro de contenedores</i>	5
<i>Escalado automático de instancias</i>	Escalado automático de AWS	5	Conjunto de escala de máquina virtual Autoescalado	5	Grupos de instancias	5
Total de ponderación		20		20		15

Realizado por: CARDOSO Cristhian, 2021.

Tabla 7-2. Almacenamiento

<i>Servicios</i>	AWS	Valor	Azure	Valor	Google	Valor
<i>Disco para instancias</i>	Tienda de bloques elásticos (EBS)	5	Almacenamiento de disco	5	Disco persistente	5
<i>Almacenamiento de ficheros</i>	Tienda de bloques elásticos (EBS)	5	Almacenamiento de disco	5	Almacén de archivos en la nube	5
<i>Soluciones para grandes transferencias de datos</i>	DataSync Borde de bola de nieve Motonieve	5	Importación y exportación Caja de datos de Azure	5	Servicio de transferencia de almacenamiento	5
<i>Respaldo</i>	Glacier Puerta de enlace de almacenamiento	5	Copia de seguridad de Azure	5	Línea fría	5
<i>Servicio de Recuperación ante Desastres</i>	Recuperación del sitio	5	<i>Recuperación del sitio de Azure</i>	5	<i>Recuperación ante desastres de Hystax</i>	5
Total de ponderación		25		25		25

Realizado por: CARDOSO Cristhian, 2021.

Tabla 8-2. Bases de Datos

Servicios	AWS	Valor	Azure	Valor	Google	Valor
<i>Bases de datos relacionales</i>	RDS Amazon Aurora	5	SQL Database	5	Cloud SQL Cloud Spanner	5
<i>NoSQL</i>	DynamoDB	5	Azure Cosmos DB	5	Cloud Datastore Cloud Bigtable	5
<i>NoSQL: clave-valor</i>	DynamoDB SimpleDB	5	Almacenamiento de mesa	5	Cloud Datastore	5
<i>Migración de bases de datos</i>	Database Migration Service	5	Azure Database Migration Service	5	<i>Google Database Migration Service</i>	5
Total de ponderación		20		20		20

Realizado por: CARDOSO Cristhian, 2021.

Tabla 9-2. Seguridad e Identidad

Servicios	AWS	Valor	Azure	Valor	Google	Valor
<i>Autenticación y autorización</i>	Gestión de identidades y accesos (IAM) Organizaciones	5	Directorio Activo Active Directory Premium	5	Cloud IAM Proxy de Cloud Identity-Aware	5
<i>Protección de la información</i>	Cifrado y protección de datos	5	Protección de la información de Azure	5	<i>Protección y control</i>	5
<i>Firewall</i>	Cortafuegos de aplicaciones web	5	Puerta de enlace de aplicaciones	5	<i>Políticas de firewall</i>	5
<i>Evaluación de la integridad</i>	Inspector	5	Security Center	5	<i>Cloud Monitoring</i>	5
<i>Administrador de identidades</i>	Cognito	5	Azure Active Directory B2C	5	<i>Administrador de identidades y acceso</i>	5
<i>Protección a DDoS</i>	Escudo de AWS	5	Servicio de protección DDoS	5	Armadura de nube	5
Total de ponderación		30		30		30

Realizado por: CARDOSO Cristhian, 2021.

Tabla 10-2. Redes y Conectividad

Servicios	AWS	Valor	Azure	Valor	Google	Valor
<i>Entornos virtuales de red aislados</i>	Nube virtual privada	5	Red Privada	5	Nube virtual privada	5
<i>Conexión con entornos on-prem</i>	VPN administrada por AWS	5	VPN Gateway	5	Cloud VPN	5
<i>DNS administrativo</i>	Route 53	5	Azure DNS	5	Google Cloud DNS	5
<i>Redirección de tráfico y mejorar el rendimiento</i>	Gestor de tráfico	5	<i>Administrador de tráfico de Azure</i>	5	<i>Equilibrio de carga en la nube</i>	5
<i>Red privada para conectar la nube con los entornos locales</i>	Conexión Directa	5	ExpressRoute	5	Cloud Interconnected	5
<i>Balaneo de carga</i>	Equilibrio de carga elástico	5	Balancedador de carga	5	Equilibrio de carga en la nube	5
Total de ponderación		30		30		30

Realizado por: CARDOSO Cristhian, 2021.

Respecto a las características del nivel gratuito y pago por servicio la ponderación aplicada se mantiene en escala del 0-5 pero al analizar valores cuantitativos se establece lo siguiente:

Para Nivel gratuito:

En el ítem tiempo de prueba:

- 5 Gratuidad en la utilización de los servicios HASTA 12 meses
- 3 Gratuidad en la utilización de los servicios HASTA 6 meses
- 1 Gratuidad en la utilización de los servicios HASTA 3 meses
- 0 Gratuidad en la utilización de los servicios menor a 3 meses

En el ítem de tiempo de procesamiento en horas/mes:

- 5 Tiempo de procesamiento mayor o igual a 750 horas/mes
- 0 Tiempo de procesamiento menor a 750 horas/mes

En el ítem almacenamiento:

- 5 Almacenamiento mayor a 5 GB
- 3 Almacenamiento igual a 5 GB
- 0 Almacenamiento menor a 5 GB

Tabla 11-2. Nivel gratuito

<i>Servicios</i>	AWS	Valor	Azure	Valor	Google	Valor
<i>Tiempo de prueba</i>	12 meses	5	12 meses	5	3 meses	1
<i>Tiempo de procesamiento en horas/mes</i>	750 horas al mes	5	750 horas al mes	5	750 horas al mes	5
<i>Almacenamiento</i>	Hasta 20 GB	5	5 GB	3	5GB	3
<i>Total de ponderación</i>		15		13		9

Realizado por: CARDOSO Cristhian, 2021.

Para Pago por uso del servicio:

En el ítem Cobro por año:

Para este ítem selecciona el cobro estimado por año, debido a la escalabilidad y duración del proyecto, además es el costo del servicio por año es más económico.

- 5 Costo menor o igual a 0.15 centavos de dólar por año
- 3 Costo menor o igual a 0.25 centavos de dólar por hora
- 0 Costo menor o igual a 0.35 centavos de dólar por hora

Tabla 12-2. Pago por uso del Servicio

Servicios	AWS	Valor	Azure	Valor	Google	Valor
<i>Cobro por hora/año</i>	\$0,147	5	\$0,238	3	\$0,331	0
Total de ponderación		5		3		0

Realizado por: CARDOSO Cristhian, 2021.

El resultado total de ponderación de las plataformas disponibles en el mercado según las características fundamentales requeridas para el proyecto se resume a continuación.

Tabla 13-2. Selección por ponderación

Características del sistema	AWS	Azure	Google
<i>Recursos Informáticos</i>	20	20	15
<i>Almacenamiento</i>	25	25	25
<i>Bases de Datos</i>	20	20	20
<i>Seguridad e Identidad</i>	30	30	30
<i>Redes y Conectividad</i>	30	30	30
<i>Nivel gratuito</i>	15	13	9
<i>Pago por uso</i>	5	3	0
Total	145	141	129

Realizado por: CARDOSO Cristhian, 2021.

Debido a que Amazon Web Services, presenta como mejor opción ya que es este sistema de almacenamiento cumple con uno de los requisitos fundamentales para la implementación del proyecto el cual son, buen almacenamiento y escalable, seguridad de la información ante ataques y sobre todo que sea de bajo costo, esto a pesar de que el modo gratuito concluya, la tasa de cobro es el más económico con respecto a Microsoft Azure y Google Cloud.

2.12 Telefonía móvil celular

La telefonía celular se diseñó para proporcionar de conexiones estables entre dos dispositivos móviles o entre una unidad móvil y una estacionaria (fija). Un proveedor de servicios debe ser capaz de localizar y hacer un seguimiento al que llama, asignando un canal a la llamada y transfiriendo la señal de un canal a otro a medida que el dispositivo se mueve fuera del rango de un canal y dentro del rango de otro canal. (Forouzan, 2007, p. 202)

Para que este seguimiento sea posible, el área de servicio se divide en regiones pequeñas denominadas células. Cada una de estas células contienen una antena controlada por una central denominada central de célula, la cual a su vez está controlada por una central de comunicación de telefonía móvil, quien es el encargado de coordinar las comunicaciones entre todas las centrales de célula y la central telefónica. (Forouzan, 2007, p. 203), véase **Figura 20-2**.

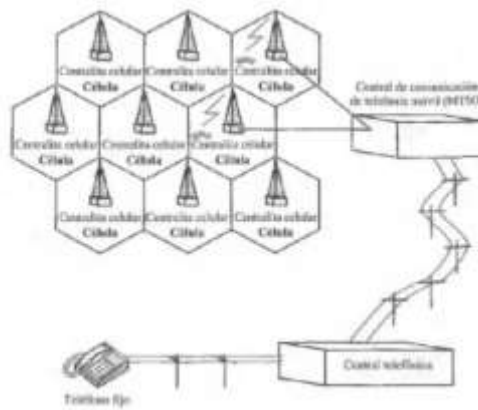


Figura 20-2. Esquema del Sistema Celular
Fuente: (Forouzan, 2007, p. 203)

El tamaño de las células no es fijo y puede variar dependiendo de la población del área. El radio típico de una célula está entre 2 y 20 kilómetros. Las áreas con alta densidad necesitan más células geográficamente más pequeñas para satisfacer las demandas de tráfico que las áreas de baja densidad de población. Una vez calculado, el tamaño de las células se puede optimizar para prevenir interferencias de las señales de las células adyacentes. La potencia de la señal se mantiene baja para prevenir que su señal interfiera a las demás células. (Forouzan, 2007, p. 203)

2.12.1 Estándar GSM (*Global System for Mobile communications*)

GSM es un estándar mundial para teléfonos móviles digitales creado por la CEPT y desarrollado por el ETSI como un estándar para los teléfonos móviles europeos, para desarrollar una normativa que fuera adoptada mundialmente. (Velasco, 2005, p. 70)

Este estándar tiene cuatro versiones principales basadas en las bandas: GSM-850, GSM-900, GSM-1800 y GSM-1900. GSM-900 (900 MHz) y GSM-1800 (1,8 GHz) son las versiones que se utilizan en la mayor parte del mundo, salvo Estados Unidos, Canadá y el resto de América Latina, donde se utilizan las bandas GSM-850 y GSM-1900 (1,9 GHz), ya que en Estados Unidos las bandas de 900 y 1800 MHz están destinadas para uso militar. Debido a la rápida expansión, el espacio radioeléctrico se vio saturado en torno a la frecuencia de 900 MHz, por lo que las redes de telecomunicación pública se enfocaron en las frecuencias de 1800 y 1900 MHz, por lo que es habitual que los equipos móviles de hoy en día sean tribanda. (Velasco, 2005, p. 70)

El único servicio ofrecido por GSM y que no se encuentra en los sistemas analógicos más antiguos es el que realmente nos interesa para el presente proyecto, el servicio de mensajes cortos o SMS (Short Message Service). SMS es un servicio bidireccional para mensajes alfanuméricos cortos de hasta 160 bytes. (Velasco, 2005, p. 70)

2.12.2 Entramado GSM

Para la transmisión de “bits” entre la estación base y una estación móvil se realiza mediante canales físicos, determinados por un numero de slot y una portadora. Dentro de cada portadora, capaz de capaz de transportar una multitrama se multiplexan en el tiempo 8 ranuras, formando una trama TDMA. En el gráfico de la **Figura 17-2** se detalla una de estas tramas de tráfico, aunque las hay de otros tipos. Cabe recalcar que, dentro de los 26 slots de la multitrama, el 12 está reservado para señalización y el 25 no se utiliza, siendo el resto canales de datos. (Velasco, 2005, p. 72)

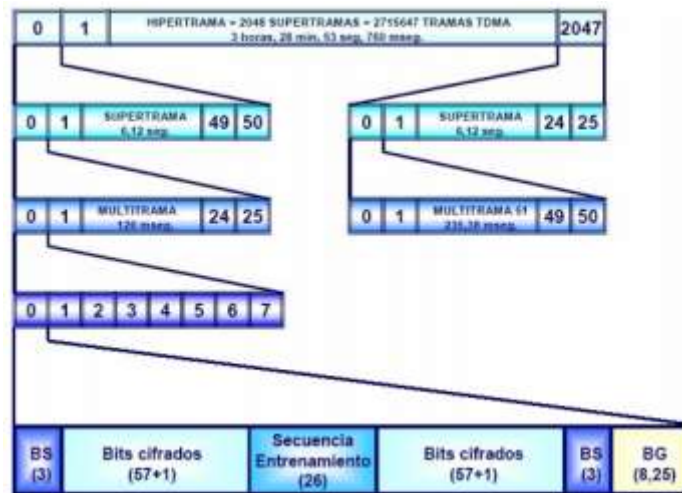


Figura 21-2. Estructura de la trama GSM
Fuente: (Velasco, 2005, p. 72)

2.12.3 Tarjeta SIM (Subscriber Identify Module)

La tecnología GSM viene provista de una tarjeta SIM o conocida como chip, la cual almacena información importante del usuario o abonado, además de funcional como un perfil de usuario facilitando el cambio de equipo terminal conservando sus datos, así también como el cambio de operadora conservando su equipo. (Sislema, 2018, p. 26)

2.12.4 Tecnología GPRS

GPRS esta tecnología es una extensión de GSM, principalmente usada para la transmisión de datos por conmutación de paquetes, también conocida como GSM-IP, con el cual se permite tasas de transmisión de 56 a 144 Kbps, además de permitir el uso de los servicios que brinda TCP/IP y HTTP.

2.12.4.1 Características de la tecnología

Entre las características que ofrece GPRS, la principal es la efectividad al momento de cargar los datos, puesto que es mucho mayor que el Servicio de Mensajes Cortos (SMS), donde el número

de caracteres está limitado a 160. Gracias a la combinación de la velocidad y capacidad convierte a GPRS en el medio ideal para servicios tales como Protocolo de Aplicaciones Inalámbricas (WAP) y Mensajería Multimedia (MMS), así ofreciendo una mayor variedad de servicios innovadores y generadores de facturación. (Naranjo y Chuquitarco, 2012, p. 10–11)

2.12.4.2 *Servicios de GPRS*

Entre los principales servicios que la tecnología GPRS ofrece se destacan las siguientes:

- Servicios basados en SMS.
- Internet.
- Conexiones a Intranets.
- Servicios específicos para GPRS.
- Servicios de localización.
- Aplicaciones WAP.

CAPÍTULO III

3 DISEÑO Y PROTOTIPO

3.1 Marco metodológico

En el presente capítulo, el desarrollo del prototipo se regirá al modelo de Principios de Diseño de Pahl y Beitz.

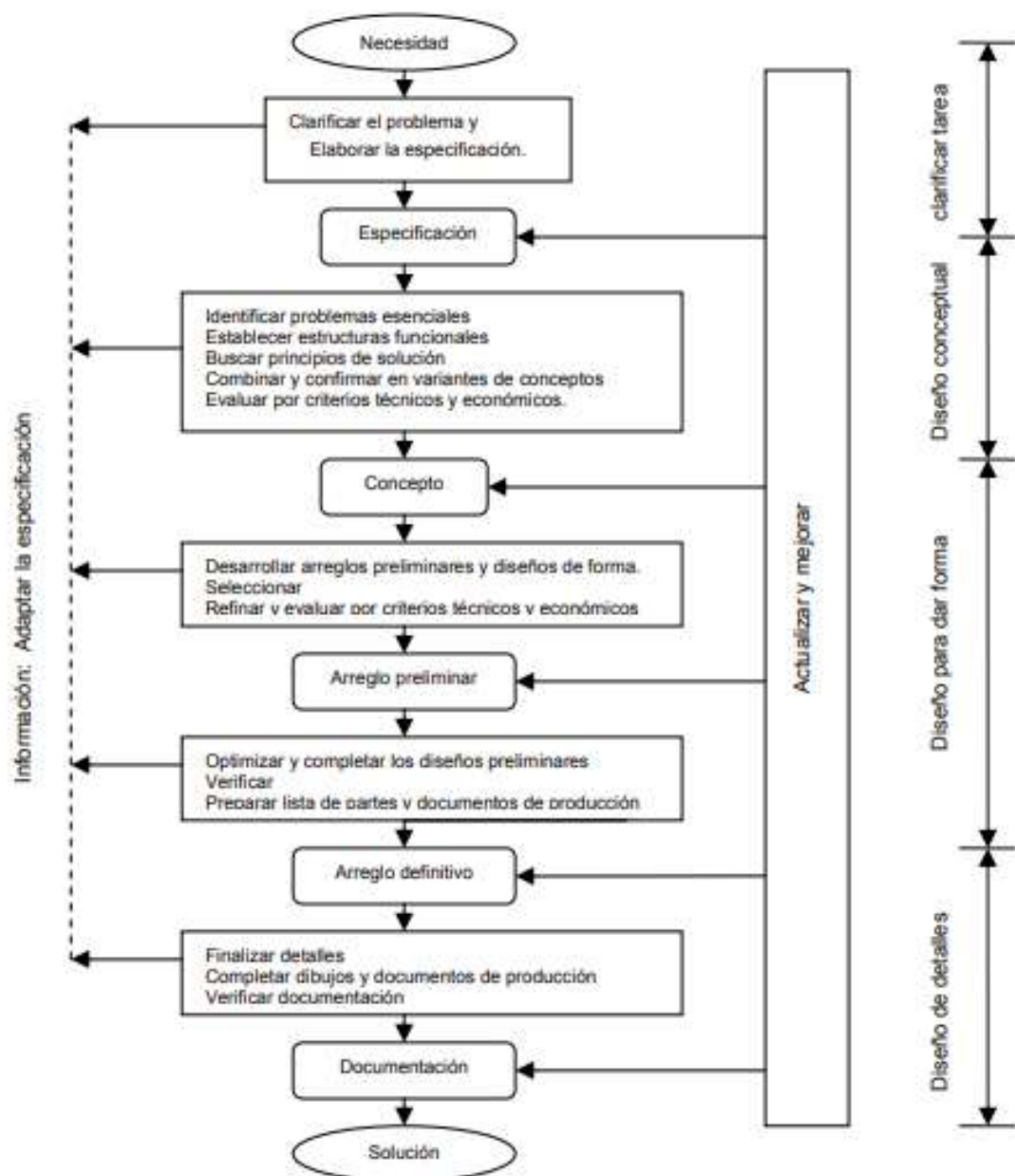


Figura 1-3. Diagrama de principios de diseño de Pahl y Beitz
Fuente: (PAHL Y BEITZ, 1996)

3.2 Características del sistema

El sistema de extracción de datos debe cumplir con las siguientes características:

- Uso de una tarjeta de desarrollo.
- Alimentación del equipo de 5V/3A.
- Adaptador bluetooth.
- Uso del scanner OBD-II.
- Uso de red GPRS.
- Utilización del sistema de posicionamiento global (GPS).
- Uso de los datos obtenidos del OBD-II por comandos AT.
- Módulos GPS y GPRS/GSM de tamaños reducidos y de bajo consumo de energía.
- Dispositivo de recepción compacto y portable.
- Desarrollo de una página web para la presentación de los datos.
- Utilizar una base de datos para el almacenaje de los datos recibidos por el dispositivo móvil.
- Acceso y recolección de datos del OBD-II y GPS.

3.2.1 Especificaciones de alcance

Se toma en base a los requerimientos de una flota de vehículos que cumplen con las características de compatibilidad del scanner OBD-II, para el presente trabajo se toman en cuenta los vehículos livianos de la gama liviana y del año 2005 en adelante.

3.2.1.1 Prototipo del sistema



Figura 2-3. Diagrama de bloques del sistema de extracción de datos.
Realizado por: CARDOSO Cristhian, 2021.

Para el desarrollo de la plataforma de extracción de datos del automóvil implica estructurarlo desde su base, desde el dispositivo de extracción y localización hasta su funcionamiento, para lo cual hay que definir y organizar cada uno de ellos. La **Figura 2-3** muestra el diagrama de bloques del sistema de extracción de datos del vehículo donde se puede observar los diferentes componentes que conforman el dispositivo móvil, el modo de conexión para la extracción de datos, la red de transmisión de datos, el servidor y la presentación de los datos al usuario mediante el internet.

- **Dispositivo de extracción o equipo terminal**

El dispositivo de extracción de datos del vehículo debe ser desarrollado en una tarjeta de desarrollo Raspberry Pi 3 modelo B, la cual será el elemento base del dispositivo donde se realizan las respectivas configuraciones y será el encargado de controlar todo el proceso de gestión de la información recopilada, se utilizará el lenguaje PHP para la misma.

Se usará un adaptador Bluetooth, un adaptador OBD-II (ELM327 v2.1), un módulo GPS y un módulo GPRS, con los cuales se realizará la adquisición de los datos de la ECU del automóvil y del sistema de posicionamiento satelital GPS, además de la transmisión respectivamente, haciendo uso de la red GPRS, todos estos dos módulos y el adaptador Bluetooth se conectan a la Raspberry Pi 3 mediante la conexión serial USB, así aprovechando su alimentación por la misma conexión, mientras tanto, el ELM327 se conecta a la Raspberry Pi 3 mediante Bluetooth y el puerto rfcomm.

La alimentación de la tarjeta de desarrollo será de 5V/3A por el puerto mini USB y conectado a un adaptador que será conectado a la alimentación disponible en el vehículo o a su vez la utilización de un banco de baterías se el vehículo no cuenta con una alimentación USB disponible.

Tanto los módulos, adaptadores y la tarjeta de desarrollo hacen parte del dispositivo de extracción móvil, los cuales serán colocados dentro de una caja haciéndolo de dimensiones y peso portables, además el dispositivo será de fácil instalación por parte del usuario.

- **Servidor de monitoreo de datos**

Para recopilar la información adquirida por el dispositivo de extracción, es necesario contar con un servidor el cual nos permita almacenar y recibir la información enviada por el dispositivo, este servidor dependerá de las características y complejidad del sistema, como la interacción con un ambiente gráfico y consulta de la información adquirida.

- **Interfaz de usuario o cliente**

De la misma forma se creará una página web directamente en el servidor, el cual desplegará los datos adquiridos por el dispositivo de extracción y será posteriormente visualizado con dispositivo o equipo con conexión a internet.

La página web contará con las siguientes características:

- Ambiente gráfico.
- Acceso mediante un nombre de usuario y contraseña.
- Tabla de datos recolectados del dispositivo.
- Graficas de cada uno de los datos recolectados.

- Conexión a la base de datos previamente mencionada.
- Servicios adicionales que interesen al usuario sobre el vehículo en cuestión como su latitud, longitud, rpm, velocidad, temperatura, etc.
- Hosting gratuito.

3.3 Elección de dispositivos

En base al diseño conceptual previamente descritos en el capítulo anterior para el desarrollo del sistema además de tomar en cuenta la disponibilidad de los servicios requeridos en el dispositivo móvil, deberá integrar un receptor GPS, un adaptador Bluetooth, un ELM327 junto con un sistema de comunicación GSM/GPRS que trabaje con las operadoras disponibles en el país.

3.3.1 Tarjeta Raspberry Pi 3 modelo B

Como se detalló en el capítulo anterior, la Raspberry Pi es una placa de desarrollo reducida, una de las mejores considerando su calidad y precio. Se eligió una Raspberry Pi 3 por su tamaño, alimentación USB y bajo consumo, además de estar a un precio accesible a comparación de otros dispositivos con similares características, ya que posee una capacidad de una microcomputadora y capacidad de añadir funcionalidades. La placa cuenta con un chip Broadcom BMC2837 de 64 bits.

Tabla 1-3. Principales características de la Raspberry Pi 3 modelo B

PROCESADOR	ARM C�rtex A53 (4 n�cleos) de 1,2Ghz
RAM	1 Gb
PUERTOS	GPIO 40 pines HDMI 4 x USB 2.0 Micro SD, Micro USB DSI (Pantalla t�ctil)
CONECTIVIDAD	IEEE 802.11.b/g/n Bluetooth 4.1
ALIMENTACION	3,3V / 5V a 2,5A
DIMENSIONES	85 x 56 mm
PESO	45g

Realizado por: CARDOSO Cristhian, 2021.

En el **Anexo C** se detallan las especificaciones Generales de la tarjeta Raspberry Pi 3 modelo B.

3.3.2 *Interprete de protocolos OBD-II (ELM327 v2.1) Bluetooth*

Se optó por un OBD-II Bluetooth versión 2.1, debido a las características que posee, la compatibilidad que posee con todas y cada una de las aplicaciones de Google Play OBD existentes en la tienda para Android, además de poseer compatibilidad con la mayor gama de vehículos livianos, es de bajo costo, con una velocidad de transmisión de 9600 o 38400 baudios, alcance de 5-15 metros, clase 2 de transmisión Bluetooth, además de su alimentación es de 12/24 V el cual es proporcionado por el vehículo.



Figura 3-3. EML327 mini v2.1 Bluetooth

Fuente: <https://n9.cl/ioqx7>

3.3.3 *Adaptador Bluetooth versión 4.0 (Dongle USB)*

Para que sea posible la comunicación entre la tarjeta Raspberry Pi 3 y el intérprete de protocolos OBD-II, es necesaria la conexión y configuración de un adaptador bluetooth 4.0, debido a su compatibilidad de conexión con el ELM327, ya que la tarjeta Raspberry Pi 3 a pesar de poseer un módulo Bluetooth integrado no es posible una conexión hacia el ELM327, puesto que tienen versiones diferentes e incompatibles, la tarjeta Raspberry Pi 3 posee un módulo Bluetooth versión 4.2 la cual incluye el concepto de Bluetooth Smart y el ELM327 consta con Bluetooth versión 2.0 o 2.0 + EDR (Enhanced Data Rate o tasa mejorada de datos) que permite transmisión de datos de algo más de 2 Mb/s.

Al tener un Bluetooth en versión 4.2, solamente nos permite una comunicación hacia dispositivos de la misma categoría o Bluetooth Smart, ya que si se trata de versiones anteriores lo hace incompatible, motivo por el cual se optó por un *Dongle Bluetooth v4.0*, el cual posee compatibilidad con las versiones anteriores a esta como 1, 2 y 3, además de tener un bajo consumo de energía, bajo costo, conexión USB y su tasa de transferencia de hasta 3 Mb/s.

Entre sus principales características se tiene:

- Transceptor Bluetooth Smart Ready compatible con Bluetooth 4.0 LE.
- Dispositivo de Clase 1 con un rango de eficacia de 50 metros.
- Soporte para EDR (Enhanced Data Rate).

- Tasa de transmisión de hasta 3 Mbps.
- Bajo consumo de energía.
- Retro compatible con dispositivos con Bluetooth versiones 3.0, 2.0, 1.2, 1.1



Figura 4-3. Dongle Bluetooth v4.0

Fuente: <https://n9.cl/6d0q>

3.3.4 *Módulo GPS UBLOX NEO-M8N-0-01*

Para que el dispositivo de extracción tenga datos de posicionamiento y la capacidad de adquirir los datos del Sistema de satélites GPS, se optará por un módulo GPS el cual realizara el proceso mencionado y entregando datos NMEA a la Raspberry Pi.

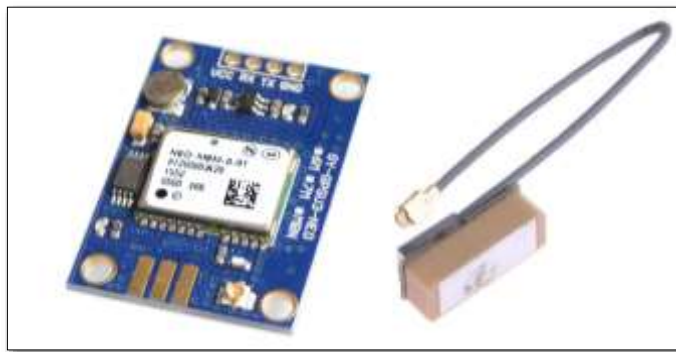


Figura 5-3. Módulo GPS UBLOX NEO-M8N-0-01

Fuente: <https://n9.cl/thty>

En la **Figura 5-3** se puede observar el módulo receptor GPS seleccionado, el cual con las antenas que tiene incluida recibe las señales de los satélites que están alrededor de la Tierra, motivo por el cual se tomó en cuenta como principal característica para la elección del dispositivo frente a otros de características similares que necesitan una configuración adicional mediante comandos AT para su configuración, activación y comenzar a recibir los datos.

El módulo funciona con un voltaje de alimentación de (3.5 - 5) voltios, para que la comunicación sea posible con la Raspberry Pi basta con conectar los pines TX y RX para recibir los datos NMEA.

Tabla 2-3. Características del Módulo GPS UBLOX NEO-M8N-0-01

Tamaño	25x35 mm
Tiempo de arranque	26 s
Precisión de posición	2.0 m
Velocidad de transmisión	9600

Realizado por: CARDOSO Cristhian, 2021.

En el **Anexo D** se detalla las especificaciones del Módulo GPS.

3.3.5 Módulo GSM/GPRS (Huawei EC303C)

Este dispositivo es el medio de conexión entre el dispositivo de extracción (parámetros OBD-II y posicionamiento) y el servidor de datos, cuando los datos obtenidos del interprete OBD-II y la señal GPS hayan sido procesados, este dispositivo será el encargado de enviar los datos al servidor mediante una conexión TCP, de esta manera permitiendo almacenarlos.

En la actualidad existe una gran gama de módulos GSM/GPRS, pero se ha optado por este módulo de comunicaciones celulares por su compatibilidad con las tarjetas de desarrollo Raspberry Pi, además que permite trabajar con lenguaje PHP y las librerías con las que cuenta, a su vez trabaja con cualquier operadora disponible en el país usando las 4 bandas de comunicación celular 850, 900, 1800 y 1900 MHz.



Figura 6-3. Módulo GSM/GPRS Huawei EC303C

Fuente: (Sislema, 2018, p. 42)

La **Figura 6-3** representa al módulo GSM/GPRS el cual soporta la red GPRS cuatribanda GSM, disponibilidad para enviar mensajes SMS de transmisión remota, su tamaño es compacto y de un bajo consumo de corriente, comunicación con la Raspberry Pi mediante el puerto USB, incluye 3GPP y SIMCOM mejorado con comandos AT, además cuenta con una ranura para insertar un SIM en la parte interna del módulo.

El módulo GSM/GPRS usa el protocolo TCP/IP para compartir remotamente información entre cliente-servidor. Al momento de transmitir la información lo hace de modo autenticado, es decir, que requiere de un usuario y contraseña para acceder, y de manera bidireccional, pero sin seguridad alguna debido a que no utiliza ningún cifrado. Usa por defecto los puertos TCP 20 o

puerto de datos, de acceso a FTP y 21 o puerto de control, donde se especifican parámetros de conexión como puertos, direcciones, etc. (Alvarez y etc, 2019) .

3.3.6 *Escenario*

Se ha considerado pertinente determinar los diferentes materiales a ser usados al momento de realizar el prototipo de extracción de parámetros OBD-II, además considerar el tipo de datos a ser extraídos mediante el sistema.

3.3.6.1 *Modelos de vehículos en análisis*

Dentro de la alta gama de vehículos que poseen una entrada hacia la ECU para la obtención de datos mediante el uso del lector de parámetros OBD-II, se ha considerado vehículos livianos de las marcas más reconocidas y disponibles en el país, además del año de fabricación se ha considerado realizar en vehículos del año 2010 en adelante, para así evitar que el ELM327 v2.1 tenga problemas de compatibilidad con estos.

Entre los vehículos a usar el prototipo se tiene:

- **Chevrolet:** Se considero esta marca puesto que en el país existe una gran variedad de modelos de estos vehículos, además de su compatibilidad con el ELM327, ya que usan el protocolo de comunicación (KW-FAST o CAN 11bit 5000Kb), además que entre estos modelos el que más destaca es el Chevrolet Aveo y sus diferentes versiones, dando lugar a un gran número de vehículos a ser considerados dentro de esta marca, el año de fabricación como se especificó será a partir del 2010.



Figura 7-3. Chevrolet Aveo

Fuente: <https://n9.cl/akxhc>

- **Hyundai:** Esta marca al igual que la anterior es considerada por la disponibilidad dentro del país, el modelo considerado en este proyecto es el *Hyundai i10*, los protocolos de comunicación con el ELM327 de esta marca son similares a los de la marca Chevrolet, de igual manera el año de fabricación a tomar en cuenta será a partir del 2010.



Figura 8-3. Hyundai i10
Fuente: <https://n9.cl/wjto>

3.3.6.2 *Bluetooth*

Bluetooth es una especificación industrial para Redes de Área Personal (WPAN) el cual posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda de los 2.4 GHz, fue diseñado para dispositivos de bajo consumo, que requieren corto alcance de emisión y basados en transceptores de bajo costo.

Bluetooth permite el uso de dos o más dispositivos en un área reducida (o hasta 100 metros si se hace uso de la versión de largo alcance) sin la necesidad de ancho de banda, además Bluetooth ofrece perfiles de servicios más detallados como por ejemplo la emulación de línea serie, el cual será aplicado en el presente proyecto, por esta razón y debido al consumo con respecto a otras tecnologías inalámbricas es por lo que se ha optado Bluetooth como base de comunicaciones entre el ELM327 y la tarjeta de desarrollo.

3.3.6.3 *Protocolo RFCOMM de Bluetooth*

RFCOMM (*Radio Frequency Communication o también Comunicación por radio frecuencia*) es un conjunto simple de protocolos de transporte que se apoya sobre el protocolo L2CAP, el cual proporciona conexiones simultáneas para dispositivos bluetooth emulando puertos serie RE-232. Este protocolo es denominado “emulador de puertos serie”, el cual soporta hasta 60 conexiones simultáneas entre dispositivos Bluetooth. El protocolo como tal sólo se encarga de las conexiones entre dispositivos mediante conexión directa más las conexiones entre el dispositivo y un módem para conexiones de red, por lo que permite que las aplicaciones que usa comunicaciones por puerto serie puedan ser fácilmente trasladadas al uso de este protocolo.

Debido a la sencillez de uso y manejo de una comunicación por puertos serie, que se basa en una interfaz de comunicación de datos digitales donde la información es transmitida bit a bit y debido que el requerimiento se adecua perfectamente a este tipo de comunicación ya que lo que se va a transmitir hacia la ECU serán caracteres con peticiones para la extracción de datos que son interpretadas correctamente por la ECU, motivo por el cual se tomó la decisión de usar este protocolo en el proyecto.

3.3.6.4 Protocolo OBD-II

Como ya se detalló acerca del sistema OBD-II en el capítulo anterior, cabe recalcar que el uso de este sistema y protocolo para la extracción de los datos de la ECU hacia la tarjeta de desarrollo, son indispensables, puesto que actualmente todos los vehículos poseen este sistema integrado como un requerimiento obligatorio, lo cual permite el uso del mismo para determinar los distintos datos que pueden ser obtenidos mediante el intérprete del protocolo (ELM327), además que el ELM327 posee una conectividad inalámbrica Bluetooth, permitiendo así hacer uso de los sistemas de comunicaciones ya conocidos e implementarlos en este proyecto.

3.3.6.5 Extracción de datos brutos

Entre los distintos datos a extraer de los modos ya mencionados en la sección 2.5, el presente proyecto está enfocado a la extracción 5 datos preestablecidos con el uso del Modo 01 de operación del OBD-II, cada dato se encuentra relacionado con un PID, la estructura de los datos a extraer se detalla a continuación.

Tabla 3-3. PIDs del Modo 01 a ser extraídos

Modo 01							
PIDs (hex)	PIDs (Dec)	Bytes de datos devueltos	Descripción	Valor mínimo	Valor máximo	Unidades	Fórmula
06	6	1	Recorte de combustible a corto plazo	-100 (Reducir combustible: demasiado rico, exceso de gasolina o falta de aire)	99.2 (Agregar combustible: demasiado pobre, escases de gasolina o exceso de aire)	%	$\frac{100}{128}A - 100$
07	7	1	Recorte de combustible a largo plazo				$(\frac{A}{1.28} - 100)$
0C	12	2	Velocidad del motor	0	16,383.75	rpm	$\frac{256A + B}{4}$
0D	13	1	Velocidad del vehículo	0	255	km/h	A
0F	15	1	Temperatura en la toma de aire	-40	215	°C	A-40

Realizado por: CARDOSO Cristhian, 2021.

Adicionalmente el sistema proveerá de información de posición obteniendo la latitud, longitud y altura por medio de GPS.

3.4 Diseño lógico del Sistema

El dispositivo móvil tiene instalado el sistema operativo Raspbian, y mediante lenguaje de programación Python se configura, para disponer de los diferentes servicios, que incluye la recolección y envío de datos al servidor, responsable del monitoreo y visualización de los datos extraídos.



Figura 9-3. Diagrama lógico del Sistema de extracción de parámetros OBD-II
Realizado por: CARDOSO Cristhian, 2021.

Para la construcción de la estructura del prototipo se hará uso de varios recursos que se encuentran disponibles en el país, no obstante, se realizan diversas tareas para que los parámetros cumplan con cierta confiabilidad tanto en lo físico como en lo lógico y así ofrecer un servicio eficiente de toma y muestra de parámetros OBD-II. El software provee herramientas que nos facilitan la administración y acceso para habilitar los servicios de internet.

3.4.1 Etapa de Transmisión

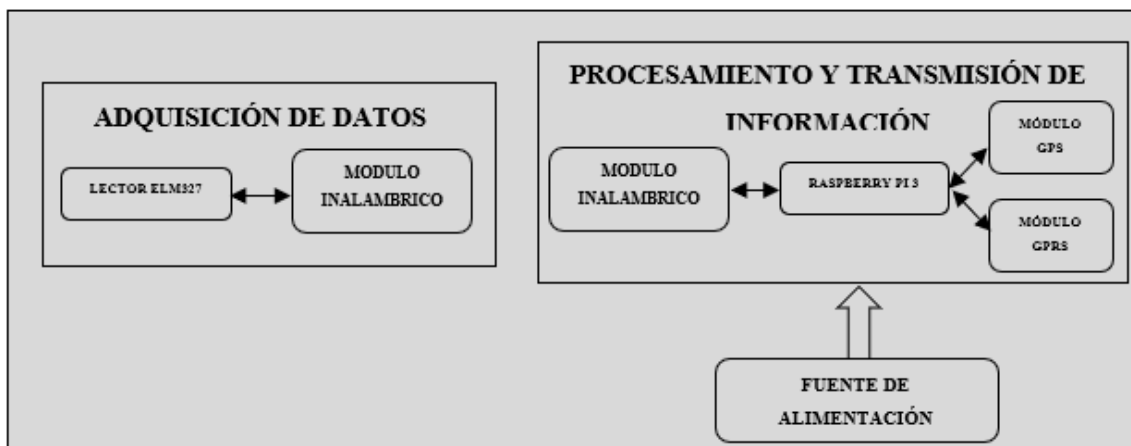


Figura 10-3. Diagrama de bloques de la etapa de transmisión
Realizado por: CARDOSO Cristhian, 2021.

La etapa de transmisión adquiere los datos de los sensores del motor del automóvil además de los datos del módulo GPS, el cual los procesa y envía hacia el servidor remoto por medio de la red GPRS de una operadora móvil. Este bloque se basa en la conexión del ELM327 hacia la ECU mediante una comunicación inalámbrica (Bluetooth), adicionalmente se crea un puerto serial para la comunicación mediante comandos en lenguaje AT, para la lectura de los códigos OBD, así solicitando los datos definidos en la sección 3.3.6.5 que producen los sensores de un automóvil y soportar el Modo 01 del sistema OBD-II, además que permitirá monitorear las medidas en tiempo real.

Posteriormente, la Raspberry Pi recolecta, procesa y envía el mensaje hacia el servidor remoto mediante el lenguaje de programación Python. Este mensaje está conformado por los siguientes campos: fecha y hora de la toma del dato, datos del módulo GPS (latitud, longitud, altura) y valores medidos por los sensores del motor del automóvil.

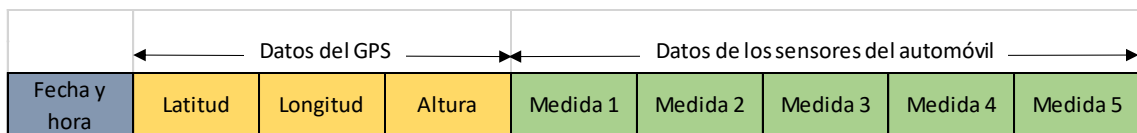


Figura 11-3. Formato del mensaje
Realizado por: CARDOSO Cristhian, 2021.

3.4.2 Etapa de Recepción

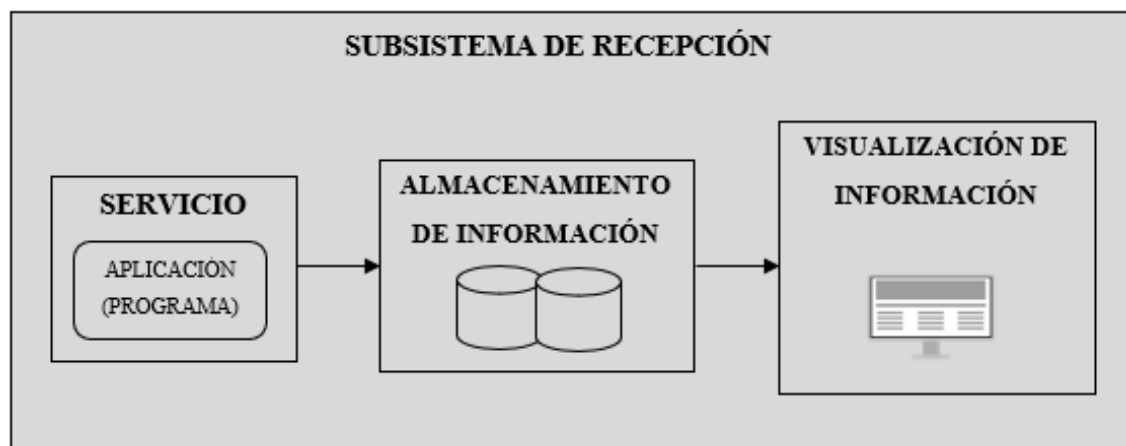


Figura 12-3. Diagrama de bloques de la etapa de recepción
Realizado por: CARDOSO Cristhian, 2021.

La etapa de recepción es el encargado de recibir la información enviada de forma continua por el subsistema de transmisión. La etapa de recepción consta de:

- **Bloque de servicio:** Está constituido por un programa que se ejecuta en segundo plano, quien es encargado de escuchar por un puerto a la espera de una conexión proveniente del subsistema de transmisión y guarda la información en forma ordenada y correcta en un sistema de almacenamiento.

- **Bloque de almacenamiento:** Encargado de recolectar, organizar y guardar toda la información enviada desde el subsistema de transmisión, de tal manera que se pueda garantizar consistencia, integridad y disponibilidad de dicha información.
- **Bloque de visualización de información:** Proporciona una interfaz web básica, la cual permite al usuario acceder y verificar que los valores medidos por los sensores del motor de un automóvil son recibidos y almacenados correctamente. La información que se muestra en la interfaz es aquella que se ha sido guardada en la base de datos durante el periodo de puesta en funcionamiento del prototipo.

3.5 Diseño físico del Sistema

3.5.1 Algoritmo de conexión del adaptador OBD-II a la Tarjeta Raspberry Pi

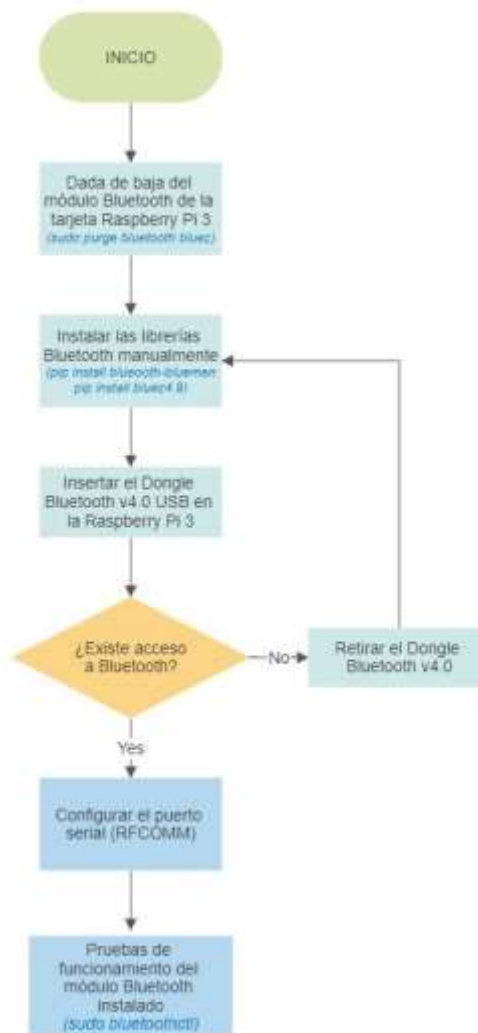


Figura 13-3. Algoritmo de conexión del ELM327 y la Raspberry Pi 3
Realizado por: CARDOSO Cristhian, 2021.

3.5.2 Etapa de Transmisión

3.5.2.1 Configuración de la Raspberry Pi

- **Materiales requeridos**

Para la configuración y puesta en funcionamiento la tarjeta Raspberry Pi es necesario los siguientes materiales:

- Memoria microSD de 32 GB.
- Adaptador microSD a SD o USB.
- Teclado.
- Mouse.
- Cable HDMI o un adaptador activo de HDMI a VGA.
- Monitor o televisor con entrada HDMI.

- **Software**

Paso 1: Preparar la tarjeta microSD para cargar el Sistema Operativo

Para no tener inconvenientes al momento de cargar el sistema operativo hacia la microSD y que este se encuentre en un formato válido se requiere un software adicional (SD Card Formatter), el cual permite formatear la tarjeta y que esta quede en el formato requerido para el proceso de carga del sistema.

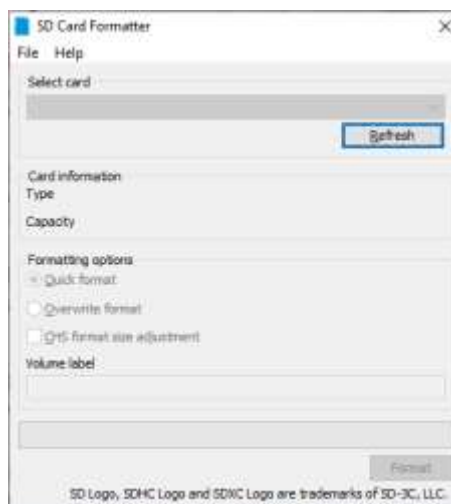


Figura 14-3. Entorno SD Card Formatter
Realizado por: CARDOSO Cristhian, 2021.

Paso 2: Descarga del Sistema Operativo y cargar hacia la tarjeta microSD

Para cargar el sistema operativo (Raspbian) disponible en <https://www.raspberrypi.org/software/> se utilizará la versión *2020-02-13-raspbian-buster-full*, se requiere un software adicional (Win32

Disk Imager), este software permite importar y grabar la imagen ISO que contiene el sistema operativo Raspbian.

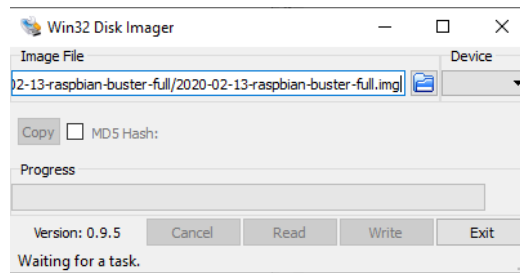


Figura 15-3. Entorno Win32 Disk Imager
Realizado por: CARDOSO Crishian, 2021.

El software es sencillo de utilizar, se selecciona la imagen respectiva con SO, determinar el dispositivo al que va a ser cargado, en este caso la microSD generalmente es la unidad (F:), y dar click en *copy*, este proceso tardara algunos minutos ya que el tamaño de la imagen es de 7 GB aproximadamente.

Paso 3: Configuración del sistema

Después de haber sido cargado el sistema a la tarjeta microSD, se procede a insertarla en la Raspberry Pi junto con todos los periféricos, se conecta el cargador al puerto micro USB de la placa y el sistema operativo comienza a instalarse automáticamente.

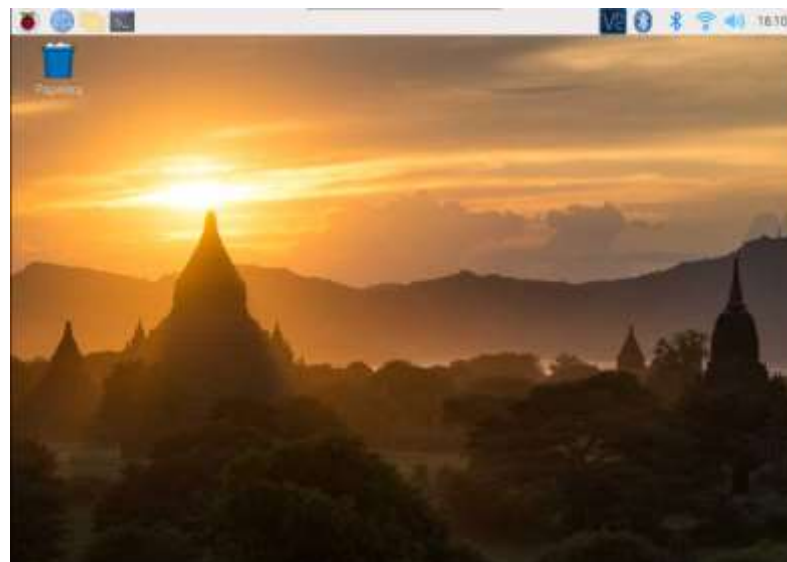


Figura 16-3. Escritorio Raspberry Pi 3
Realizado por: CARDOSO Crishian, 2021.

Una vez instalada se procede a configurar las interfaces puertos desde la configuración de la Raspberry en la interfaz gráfica de Raspbian, donde se activan según las necesidades del usuario, como se muestra a continuación.

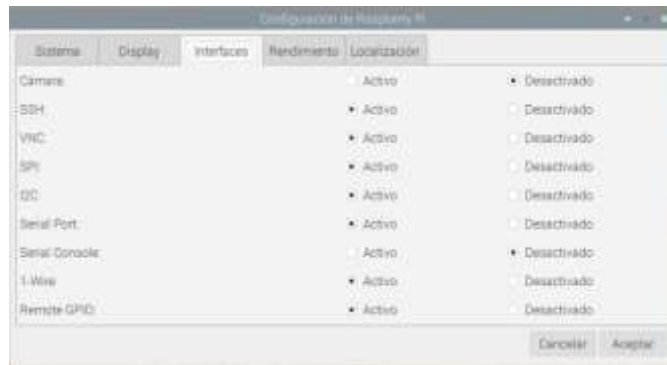


Figura 17-3. Interfaz de conexión de la Raspberry Pi 3
Realizado por: CARDOSO Cristhian, 2021.

Paso 3: Configuración para el acceso remoto a la interfaz gráfica

Existen varios métodos de conexión directa y remota hacia la interfaz de la Raspberry Pi, entre estos se utilizará VNC puesto que es un recurso disponible en el SO Raspbian y por ser un software de uso libre para PC, Smartphones, Tablet, se procese a instalar el software VNC Viewer en la PC para controlarla sin necesidad de periféricos externos, el único requerimiento es la conexión a la misma red que disponga de internet.



Figura 18-3. Interfaz del programa VNC Viewer
Realizado por: CARDOSO Cristhian, 2021.

Al seleccionar la interfaz de la Raspberry Pi conectada, se abrirá una ventana solicitando las credenciales, las cuales son las que se configuraron al inicio del arranque del sistema operativo, después de un corto lapso de tiempo se abrirá una pantalla correspondiente a la pantalla de la Raspberry Pi.



Figura 19-3. Interfaz de Raspbian mediante VNC
Realizado por: CARDOSO Cristhian, 2021.

3.5.2.2 Configuración de la librería Bluez de Bluetooth

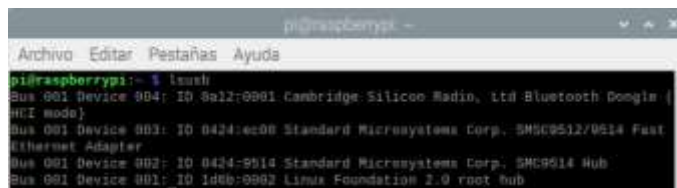
Debido a las constantes actualizaciones del sistema operativo Raspbian, algunas librerías han quedado obsoletas, motivo por el cual se ha optado en una instalación manual de las mismas mediante línea de comandos, para así habilitar que el dispositivo bluetooth permita conexiones sin ningún inconveniente.

```
wget https://www.kernel.org/pub/linux/bluetooth/bluez-5.9.tar.xz
tar xvf bluez-5.9.tar.xz
sudo apt-get install libusb-dev libdbus-1-dev libglib2.0-dev libudev-dev
-dev libreadline-dev
cd bluez-5.9
export LDFLAG5=-lrt
./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var --enable
y --disable-systemd
make
sudo make install
sudo cp attrib/gatttool /usr/bin/
cd -
reboot
```

Figura 20-3. Instrucciones de instalación Bluez
Realizado por: CARDOSO Cristhian, 2021.

3.5.2.3 Conexión del Adaptador Bluetooth y ELM327

Paso 1: Listar los puertos USB disponibles y verificar la disponibilidad del Dongle Bluetooth mediante el comando *lsusb*.



```
pi@raspberrypi:~$ lsusb
Bus 001 Device 084: ID 0a12:0001 Cambridge Silicon Radio, Ltd Bluetooth Dongle (HCI mode)
Bus 001 Device 083: ID 1042:ec08 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 082: ID 1042:9514 Standard Microsystems Corp. SMSC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Figura 21-3. Listado de puertos USB
Realizado por: CARDOSO Cristhian, 2021.

Paso 2: Habilitar Adaptador Bluetooth desde el administrador de dispositivos y verificar que este activo solamente el del Dongle Bluetooth.

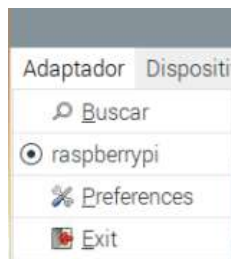


Figura 22-3. Selección del adaptador Bluetooth
Realizado por: CARDOSO Cristhian, 2021.

Paso 3: Para configurar el ELM327 Bluetooth se lo puede realizar por línea de comando o usando la interfaz de dispositivos que nos proporciona la herramienta Bluetooth, realizamos la acción Buscar hasta que se encuentre el dispositivo a ser conectado, donde se debe ingresar el código de emparejamiento que por lo general es “1234 o 0000”, por último darle acceso mediante puerto serial



Figura 23-3. Configuración serial del ELM327 Bluetooth
Realizado por: CARDOSO Cristhian, 2021.

Una vez realizado de forma correcta la conexión nos mostrara un mensaje que contiene el puerto asignado para el dispositivo ELM327 Bluetooth, como se muestra a continuación.



Figura 24-3. Asignación del puerto serie a OBD-II
Realizado por: CARDOSO Cristhian, 2021.

3.5.2.4 Pruebas de lectura de datos con ELM327

Con el uso de *minicom* ingresaremos de forma manual hacia la ECU mediante el intérprete ELM327 que está configurado en el puerto *rfcommX*, donde *X* representa el número del puerto al cual fue asignado el dispositivo bluetooth. Por medio de comandos AT ya especificados en el capítulo anterior, se accede a realizar peticiones de versión del ELM327, voltaje de alimentación, protocolo del vehículo, y parámetros del vehículo, los cuales se muestran en valores hexadecimales, dando resultado lo siguiente:



Figura 25-3. Lectura y configuración del ELM327 mediante minicom
Realizado por: CARDOSO Cristhian, 2021.

Para la visualización de la información obtenida de los sensores del motor se creó un script .py de prueba para realizar la extracción de la información y verificar el correcto funcionamiento de este, además que los datos obtenidos sean correctos y no emita información errónea.

```

pi@raspberrypi:~$ python rev.py
Iniciando...
794 rpm @ k/h 51 C
771 rpm @ k/h 51 C
790 rpm @ k/h 51 C
2121 rpm @ k/h 51 C
2529 rpm @ k/h 51 C
2338 rpm @ k/h 51 C
3132 rpm @ k/h 51 C
3218 rpm @ k/h 51 C
3168 rpm @ k/h 51 C
3208 rpm @ k/h 51 C
1915 rpm @ k/h 51 C
875 rpm @ k/h 51 C
823 rpm @ k/h 51 C
825 rpm @ k/h 51 C
833 rpm @ k/h 51 C

```

Figura 26-3. Prueba de lectura de los sensores del vehículo
Realizado por: CARDOSO Cristhian, 2021.

3.5.2.5 Configuración del módulo GPS

El módulo GPS Ublox trabaja con los pines UART de los puertos serie GPIO, los cuales deben ser habilitados, para esto se edita el archivo “config.txt”, digitando *sudo nano /boot/config.txt* en el terminal y añadir los permisos necesarios.

```

pi@raspberrypi ~
Archivo Editar Pestañas Ayuda
GNU nano 3.2 /boot/config.txt
dtoverlay=vc4-fkms-v3d
max_framebuffers=2

[all]
dtoverlay=vc4-fkms-v3d
enable_uart=1
dtoverlay=wl-gpio

dtparam=spi-on
dtoverlay=pi3-disable-bt
core_freq=250
force_turbo=1

```

Figura 27-3. Configuración de los puertos
Realizado por: CARDOSO Cristhian, 2021.

Los pines 14 y 15, son los puertos UART o puertos serie ttyAMA0 o ttyS0, dependiendo de la tarjeta Raspberry que se utilice, para determinar cuál es el correcto se debe fijar aquel que lleve el nombre “serial 0”, mediante el comando *ls -l lsusb*

```

drwxr-xr-x 2 root root          60 dic 31 1969 raw
crw-rw-r-- 1 root netdev    10, 242 ene 27 10:24 rfkill
lrwxrwxrwx 1 root root         7 ene 27 10:24 serial0 -> ttyAMA0
lrwxrwxrwx 1 root root         5 ene 27 10:24 serial1 -> ttyS0
drwxrwxrwt 2 root root          40 feb 14 2019

```

Figura 28-3. Visualización de los puertos en uso con ls - lsusb
Realizado por: CARDOSO Cristhian, 2021.

Como se muestra en la **Figura 28-3** el puerto serie es el ttyAMA0, este puerto debe ser habilitado para que el módulo GPS funcione de manera correcta, la habilitación debe ser mediante los comandos:

```
sudo systemctl stop serial-getty@ttyAMA0.service
sudo systemctl disable serial-getty@ttyAMA0.service
```

Figura 29-3. Configuración del puerto serial ttyAMA0
Realizado por: CARDOSO Cristhian, 2021.

Después de realizar los cambios, guardar y reiniciar la tarjeta para que los cambios realizados tengan efecto.

3.5.2.6 Prueba de funcionamiento del módulo GPS

El pin GPIO 15 (Rx) de la Raspberry estará conectado de forma cruzada con el módulo GPS, es decir Rx con el Tx del módulo.

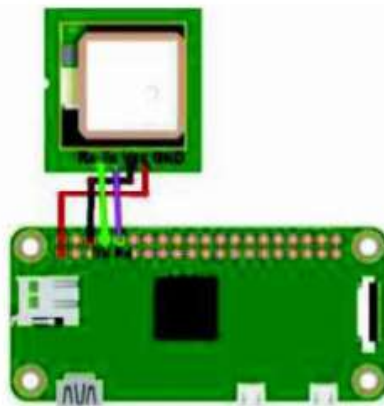


Figura 30-3. Modo de conexión del módulo GPS a la Raspberry Pi
Realizado por:(ACOSTA CURIPALLO, 2020)

Una vez configurado y verificado correctamente el puerto y las conexiones se procede a ejecutar el script .py en el cual se aloja el código para visualizar los datos de latitud y longitud tal como se muestra en la **Figura 31-3**.

```
pi@raspberrypi ~$ python gps.py
Latitud=-1.6766533333 Y Longitud=-78.6228483333
Latitud=-1.6766533333 Y Longitud=-78.6228458333
Latitud=-1.67665616667 Y Longitud=-78.622845
Latitud=-1.67666033333 Y Longitud=-78.6228371667
Latitud=-1.676661 Y Longitud=-78.6228355
Latitud=-1.67665933333 Y Longitud=-78.622838
Latitud=-1.6766595 Y Longitud=-78.6228468333
Latitud=-1.67666016667 Y Longitud=-78.6228496667
Latitud=-1.6766605 Y Longitud=-78.6228386667
Latitud=-1.67665916667 Y Longitud=-78.6228485
Latitud=-1.6766595 Y Longitud=-78.6228386667
Latitud=-1.67665883333 Y Longitud=-78.6228385
Latitud=-1.6766595 Y Longitud=-78.6228365
Latitud=-1.67665966667 Y Longitud=-78.6228351667
Latitud=-1.6766633333 Y Longitud=-78.622833
Latitud=-1.6766616667 Y Longitud=-78.6228333333
Latitud=-1.6766606667 Y Longitud=-78.6228328333
Latitud=-1.6766615 Y Longitud=-78.6228318333
```

Figura 31-3. Prueba de funcionamiento de recepción de datos GPS
Realizado por: CARDOSO Cristhian, 2021.

3.5.3 *Etapa de Recepción*

3.5.3.1 *Interfaz Web*

Para el desarrollo de la interfaz o página web se utilizará Visual Studio Code, el cual permitirá crear scripts escritos en lenguaje PHP, HTML, Java, entre otros, cada una de las paginas contará con diferentes tipos de contenido como:

- Los datos de posicionamiento y de los sensores del motor del vehículo.
- Una tabla de datos almacenados.
- Función para descarga de los datos que el usuario requiera.
- Grafica de cada uno de los datos almacenados.
- Grafica en tiempo real de los datos de los sensores del motor del vehículo.
- Medidores en tiempo real de velocidad, revoluciones y temperatura del vehículo.
- Programa para insertar en la base de datos.
- Funciones java.

3.5.3.1.1 *Diseño en Frontend*

Hace referencia a las tecnologías que se emplean en la parte del cliente, es decir aquello que se ejecuta en un navegador web.

Los lenguajes empleados en Frontend son: HTML, CSS y JavaScript.

3.5.3.1.2 *Diseño en Backend*

Hace referencia a referencia a aquello que se ejecuta por el lado del servidor, incluido las bases de datos, todo este trabajo no es posible ser visualizado por parte del cliente, siendo invisible pero muy poderoso.

Entre los lenguajes usados en Backend se tiene: PHP, Python, Javascript

3.5.3.2 *Software*

Entre uno de los softwares utilizados se tiene el *Servidor Apache*, el cual es un programa que transporta los datos de hipertexto, páginas web con todos sus complementos, este tipo de servidores utilizan el protocolo http, lo que están alojados en el ordenador que posee conexión a internet a la espera que algún navegador realice alguna petición para así responder y enviar un código HTML (Hypertext Markup Language – Lenguaje de Marcas de Hipertexto).

El servidor Apache es completamente libre y de código abierto, puede trabajar con diferentes sistemas operativos sin que se vea afectado su rendimiento, además soporta diferentes lenguajes

como Perl, PHP, Python, entre otros. En la **Figura 32-3** se puede observar la pantalla principal del servidor Apache una vez instalada.



Figura 32-3. Página principal del servidor Apache
Realizado por: CARDOSO Cristhian, 2021.

Para el servidor web se utiliza la instancia EC2 (Amazon Elastic Compute Cloud) de AWS, el cual es un servicio web que proporciona capacidad informática en una nube segura y de tamaño modificable. Está diseñado para simplificar el uso de la informática en la nube a escala web para los desarrolladores.

El desarrollo del servidor web se realiza para el seguimiento del terminal móvil, consulta de datos almacenados, graficas de los datos recibidos, visualización de la lectura de datos en tiempo real, con la creación de páginas web, además de los programas de inserción, consulta, a continuación, se describen los pasos:

1. Crear una cuenta en AWS.
2. Configurar Amazon EC2.
3. Configurar una instancia RDS para el almacenamiento de datos.
4. Instalar el servidor web Apache.
5. Crear una página web (conexión.php) para establecer la conexión con la base de datos.
6. Crear una página web (DatoSensores.php) para la lectura de los datos en tiempo real.
7. Crear una página web (login.php) para el inicio de sesión.
8. Crear una página web (registrer.php) para el registro de los usuarios.
9. Crear una página web inicial (principal.php) con información sobre el proyecto.
10. Crear la página web principal (tabla.php) donde se visualizan los datos cargados al servidor.
11. Crear una página web (grafica.php) donde permite visualizar las gráficas de los datos almacenados.
12. Crear una página web (grafreal.php) donde permite visualizar las gráficas de los nuevos datos adquiridos por el terminal móvil en tiempo real.

13. Crear una página web (medidores.php) donde se visualizan los datos de velocidad, rpm y temperatura de forma dinámica en tiempo real.
14. Crear una página web (insertar.php) donde se realiza la conexión del terminal móvil con la base de datos para la inserción de estos.

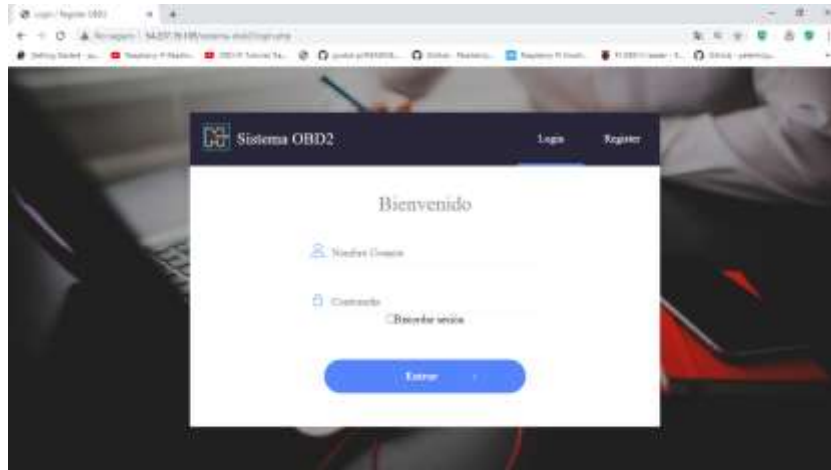


Figura 33-3. Página de inicio de la plataforma de monitoreo de datos
Realizado por: CARDOSO Cristhian, 2021.

En el **Anexo E** se muestran las paginas antes mencionadas.

3.5.3.3 Gestor de Base de Datos

El servidor de Base de Datos se encuentra montado dentro de una instancia RDS en Amazon Web Services, su característica principal es su almacenamiento de 20 GB el cual puede escalar con un plan de servicios, administrado mediante PhpMyAdmin.

3.5.3.3.1 Herramienta PhPMYAdmin

PhpMyAdmin es una herramienta que permite la administración de MySQL a través de las páginas web utilizando internet, haciéndolo de fácil uso, confiable y de un muy buen rendimiento, esto hace de PhpMyAdmin una de las aplicaciones web de administración más popular de todo el mundo.



Figura 34-3. Página de logeo de PhpMyAdmin
Realizado por: CARDOSO Cristhian, 2021.

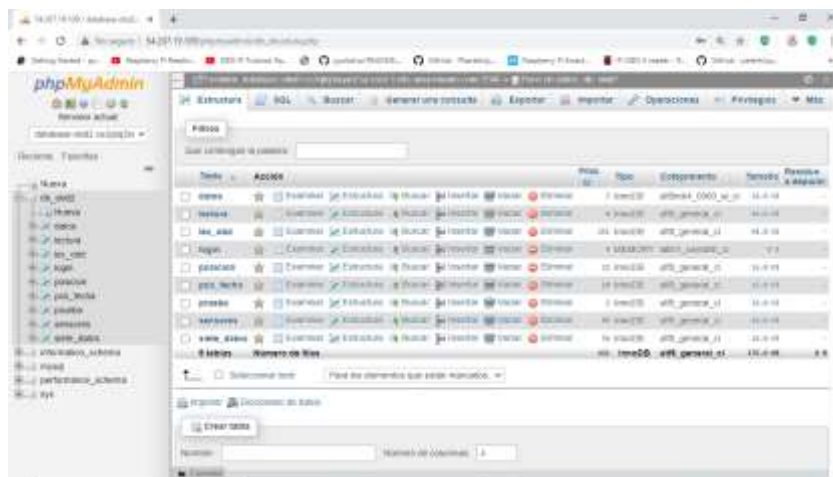


Figura 35-3. Gestor de datos de PhpMyAdmin
Realizado por: CARDOSO Cristhian, 2021.

Mediante la ayuda de la interfaz gráfica en la web para el manejo de los datos MySQL es posible la ejecución de diferentes ordenes como crear, modificar, eliminar y administrar la base de datos, una de las características importantes es la importación de varios formatos. En las **Figuras 34-3** y **35-3**. se puede observar el entorno PhpMyAdmin en el cual se maneja la base de datos.

PhpMyAdmin es el administrados de la base de datos donde permite administrar las tablas de la información del usuario, información de la página web y la tabla donde se almacenan los datos extraídos del dispositivo móvil, fecha y hora, coordenadas, altura, rpm, velocidad, temperatura, entre otros, el ingreso a este gestor está restringido para los usuarios, a continuación, los pasos realizados para la configuración de la base de datos:

1. Instalar y configurar la instancia RDS de Amazon.
2. Colocar un usuario y contraseña para el ingreso.
3. Crear una base de datos nueva.
4. Elegir el tipo de base de datos que se va a utilizar (utf8_general_ci) para los campos.

5. Guardar la base de datos.
6. Crear una tabla para guardar la información del usuario (usuarios).
7. Colocar los campos dentro de la tabla y su tipo (id, correo, usuario, clave).
8. Crear una tabla para almacenar los datos recibidos del terminal móvil (lec_obd).
9. Colocar los campos dentro de la tabla y su tipo (fech, lat, lng, alt, rpm, vel, entre otros).

3.5.3.4 Lenguaje Python

Python es un lenguaje de programación independiente multiplataforma y orientado a objetos, es una opción recomendada para el desarrollo de software libre, es un lenguaje de alto nivel ya que contiene implícitas algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, los cuales permiten realizar tareas complejas en pocas líneas de código y de manera legible. Lo realmente importante de utilizarla en la Raspberry Pi es la capacidad de aprovechar los pines GPIO para conectar lo físico con lo digital.

El lenguaje Python es empleado en la elaboración del programa principal del terminal móvil, programando el acceso a la ECU y al módulo GPS, procesando los datos obtenidos, separándolos para utilizar los datos que se requieren y posteriormente transmitirlos por la red GPRS, el procedimiento seguido es:

1. Se instala las librerías de Bluetooth (bluez, bluettoth-blueman).
2. Se instalan las librerías parse, urllib y pynmea2.
3. Dentro del script se declaran las librerías a utilizar.
4. Se declara la variable (scan) que accederá al puerto rfcomm asignado para el ELM327.
5. Se declara la variable (ser) que accederá al puerto asignado al módulo GPS.
6. Se utiliza la función parse para dividir los datos NMEA recibidos.
7. Se utiliza los comandos (scan.write y sca.readall()) para la petición y lectura de los datos de los sensores del motor del vehículo.
8. Dentro del script con los datos obtenidos con la variable (scan) se aplican las conversiones y fórmulas respectivas para cada dato.
9. Se crea la variable **campo** donde se introducirá toda la información a enviar.
10. Se crea la variable **url** donde se adjuntará la dirección del servidor.
11. Utilizamos la librería **urllib** que accede a la URL del servidor previamente declarada.
12. Se imprime la respuesta del servidor (Opcional para verificación de almacenamiento de información).
13. Se crea excepciones para el manejo de errores.

En el **Anexo F** se puede ver de forma más detallada la programación del programa principal, el cual funciona de acuerdo con el diagrama de flujo que se presenta en la **Figura 36-3**.

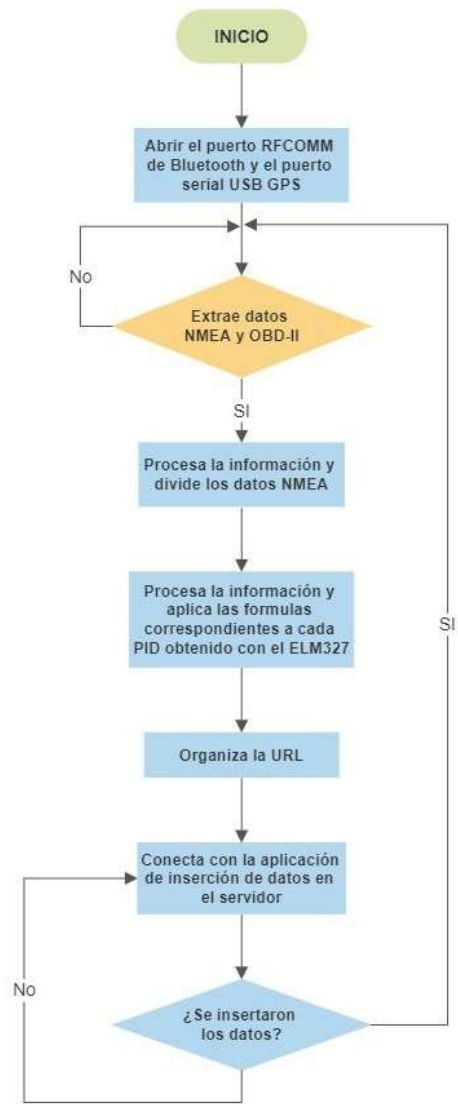


Figura 36-3. Diagrama de flujo del programa principal
 Realizado por: CARDOSO Cristhian, 2021.

CAPÍTULO IV

4 PRUEBAS Y RESULTADOS DEL PROTOTIPO

En el presente capítulo se analizan los resultados obtenidos de la implementación del prototipo de extracción de parámetros OBD y datos GPS, estos resultados se enfocaron en la aplicación, rendimiento y fiabilidad del sistema de extracción, partiendo del ambiente el cual está basado en las destrezas, capacidades y características de los módulos empleados en el prototipo, las pruebas se detallan secuencialmente a una toma de rutas dentro de la ciudad de Riobamba, realizándolo en tres diferentes modelos de vehículos los cuales ya fueron detallados en sección 3.3.6.1, vehículos en los cuales fue aplicado el terminal de rastreo para de esta forma evaluar y validar la eficiencia del sistema.

4.1 *Pruebas de comprobación*

Las pruebas fueron realizadas con la finalidad de validar el desempeño específico del sistema en condiciones reales de funcionamiento, además de corroborar si cumple con las especificaciones de diseño para las cuales fue construido.

Mediante el protocolo de pruebas se verificaron aspectos importantes sobre la implementación y su desempeño, los cuales son:

- Demostrar que el dispositivo terminal cumple con la finalidad para la cual fue diseñado y construido, además de garantizar el funcionamiento y fiabilidad del sistema de extracción de parámetros OBD-II del automóvil y su posicionamiento durante las pruebas de campo.
- Verificar el rendimiento que ofrece el dispositivo y verificar si es apto para cumplir con las características del diseño.
- Verificar y comprobar la fiabilidad de los datos obtenidos del dispositivo cargando las rutas a GoogleEarth.

Una vez determinados estos factores se demostrará si el prototipo está apto o no para su uso, en caso de no ser así, se deberá realizar un rediseño o correcciones necesarias al sistema.

4.1.1 *Características de los automóviles seleccionados*

El dispositivo puede ser instalado en automóviles que cuenten con el conector DLC para poder conectarse a la ECU. Ver sección 4.3.6.1.

Los automóviles utilizados para las pruebas de funcionamiento del dispositivo son un Chevrolet Aveo Family STD TM y un Hyundai i10, sus características informativas se presentan en la **Tabla 1-4**.

Tabla 1-4. Características informativas de automóviles probados con el prototipo

CARACTERÍSTICA	AUTOMÓVIL 1	AUTOMÓVIL 2
Marca	Chevrolet Aveo Family STD TM	Hyundai i10
Año	2013	2010
Color	Negro	Plateado

Realizado por: CARDOSO Cristhian, 2021.

4.2 Pruebas de funcionamiento

4.2.1 Escenario 1: Chevrolet Aveo Family

Previo a la realización de pruebas se instaló dentro del vehículo el dispositivo de extracción de parámetros OBD-II, para la instalación del lector ELM327 se encontró la ubicación exacta del conector DLC. Por lo general se encuentra en la parte inferior del tablero en el lado del conductor del vehículo y cerca de los pedales, en la **Figura 1-4** se puede observar la ubicación del conector DLC para los dos automóviles de prueba.



Figura 1-4. Ubicación del conector DLC en un Aveo Family
Realizado por: CARDOSO Cristhian, 2021.

El cargador USB para celulares del automóvil sirve de fuente de alimentación para la tarjeta Raspberry Pi y sus componentes.

- **Pruebas de adquisición de datos con el lector ELM327**

En la **Figura 2-4** se muestra el ejemplo del envío del PID mediante comandos *AT ATDP* y *010D* y el uso de minicom hacia el lector ELM327, donde el primer comando permite determinar el protocolo que posee el vehículo para la comunicación con el lector ELM327, mientras que el segundo comando cuyos valores son interpretados por separado, el primer valor hace referencia al modo que solicita, en este caso el modo 01 y el otro valor corresponde al PID que solicita la velocidad del motor el automóvil. La respuesta a esta petición viene dada en hexadecimal con el valor del PID que se recibe desde la ECU, el cual es *41 0D XX*, cuyos datos son procesados posteriormente por la tarjeta de desarrollo mediante el lenguaje Python aplicando las fórmulas correspondientes a cada PID.

```
AUTO, ISO 14230-4 (KWP FAST)
>01 0D
>41 0D 0C
```

Figura 2-4. Solicitud de velocidad del vehículo
Realizado por: CARDOSO Cristhian, 2021.

- **Ruta Ejecutada**

El escenario corresponde a un pequeño tramo recorrido por el centro de la ciudad de Riobamba y posteriormente hacia la parte posterior de la Escuela Superior Politécnica de Chimborazo, realizando un tracking y monitoreo de los sensores vehículo Aveo Family,



Figura 3-4. Ruta recorrida (Google Earth Pro) – Escenario 1
Realizado por: CARDOSO Cristhian, 2021.

- **Visualización de datos en tiempo real: Lectura Gráfica**

Una vez colocado el dispositivo móvil en el vehículo se visualiza el estado de cada sensor en tiempo real como se muestran en las **Figuras 4-4 a 7-4**.

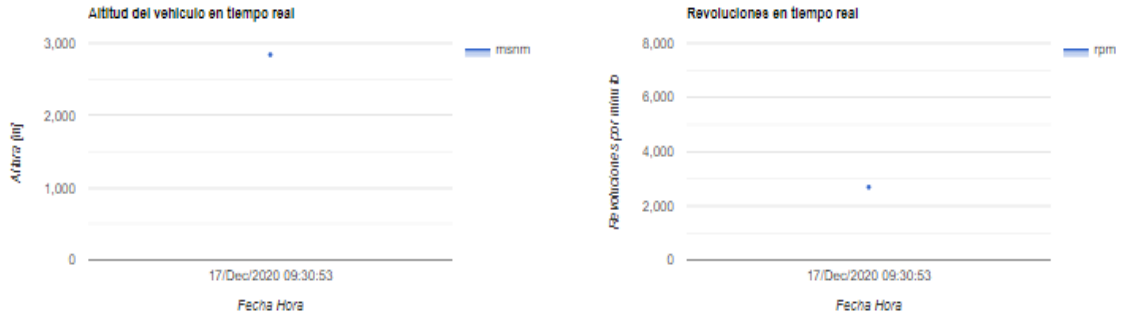


Figura 4-4. Altura y revoluciones del vehículo en tiempo real – Escenario 1
Realizado por: CARDOSO Cristhian, 2021.

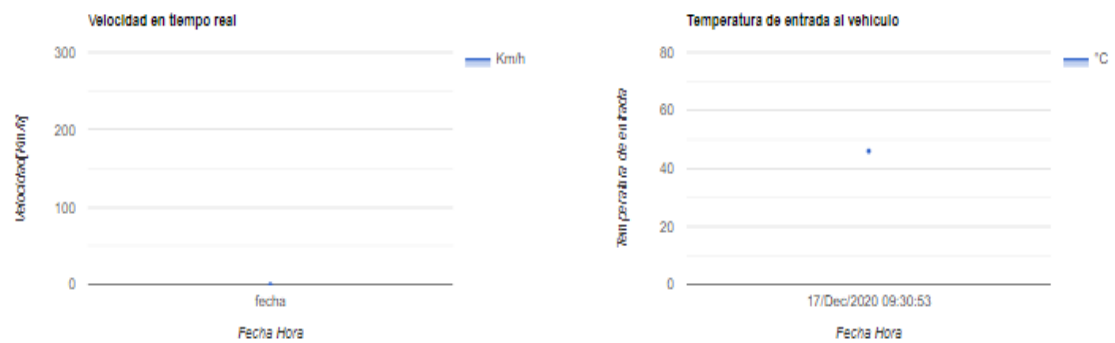


Figura 5-4. Velocidad y temperatura del vehículo en tiempo real – Escenario 1
Realizado por: CARDOSO Cristhian, 2021.

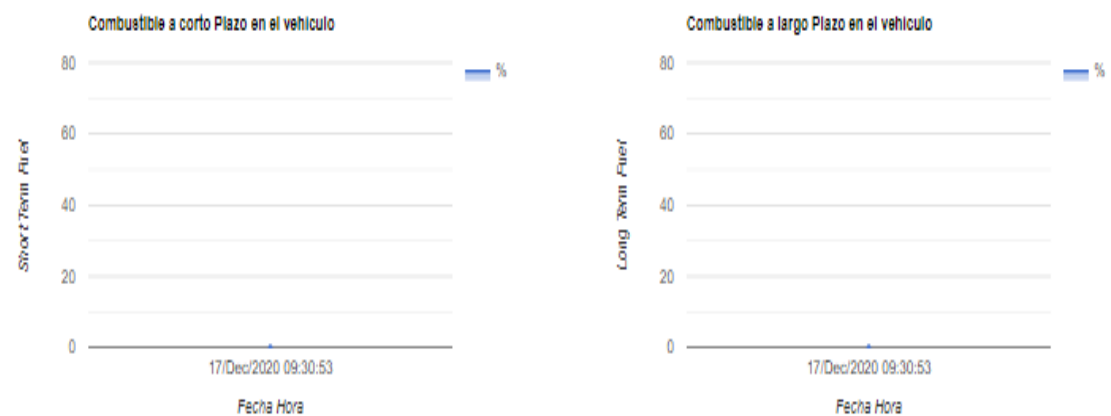


Figura 6-4. STF y LTF del vehículo en tiempo real – Escenario 1
Realizado por: CARDOSO Cristhian, 2021.



Figura 7-4. Visualización de los datos en tiempo real – Escenario 1
Realizado por: CARDOSO Cristhian, 2021.

En las **Figuras 8-4 a 11-4** se puede visualizar las gráficas de cada uno los datos obtenidos en el tramo recorrido por el vehículo como sus coordenadas, altura sobre el nivel del mar, RPM, velocidad, temperatura, consumo de combustible a corto plazo y consumo de combustible a largo plazo.

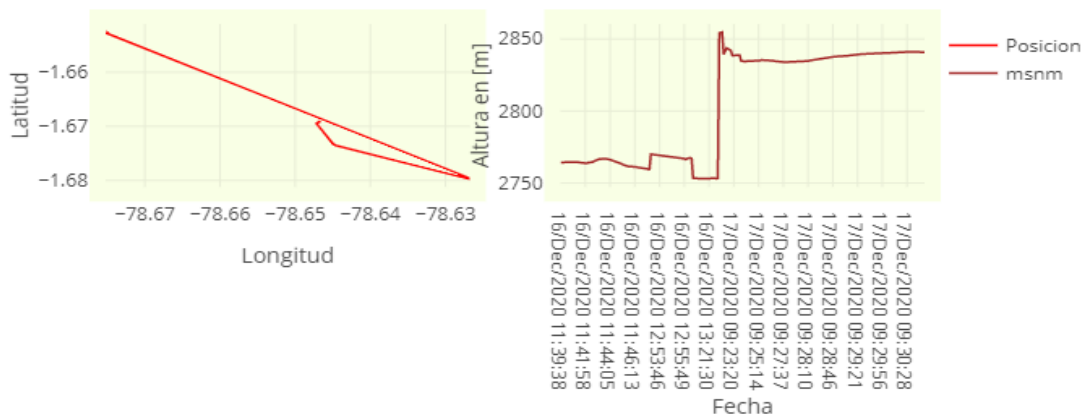


Figura 8-4. Coordenadas y Altura (msnm) del vehículo – Escenario 1
Realizado por: CARDOSO Cristhian, 2021.

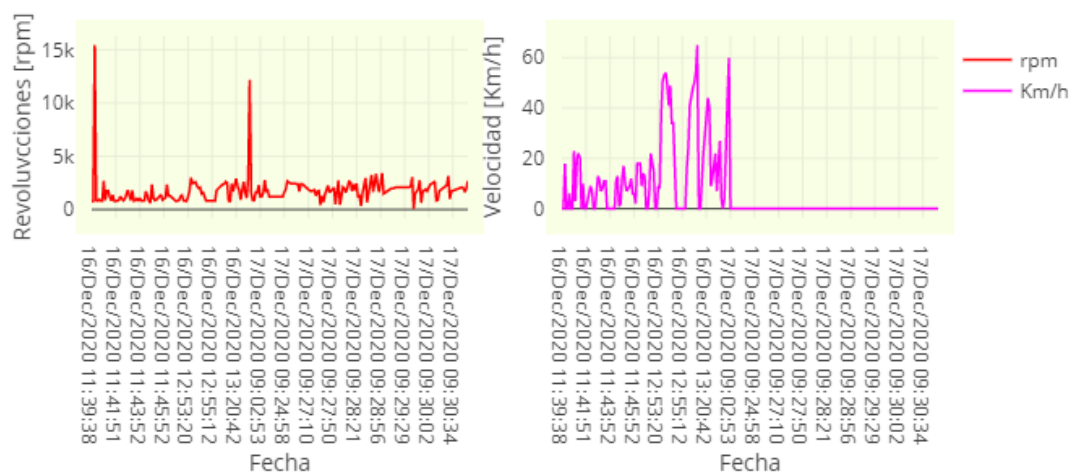


Figura 9-4. RPM y velocidad del vehículo – Escenario 1
Realizado por: CARDOSO Cristhian, 2021.

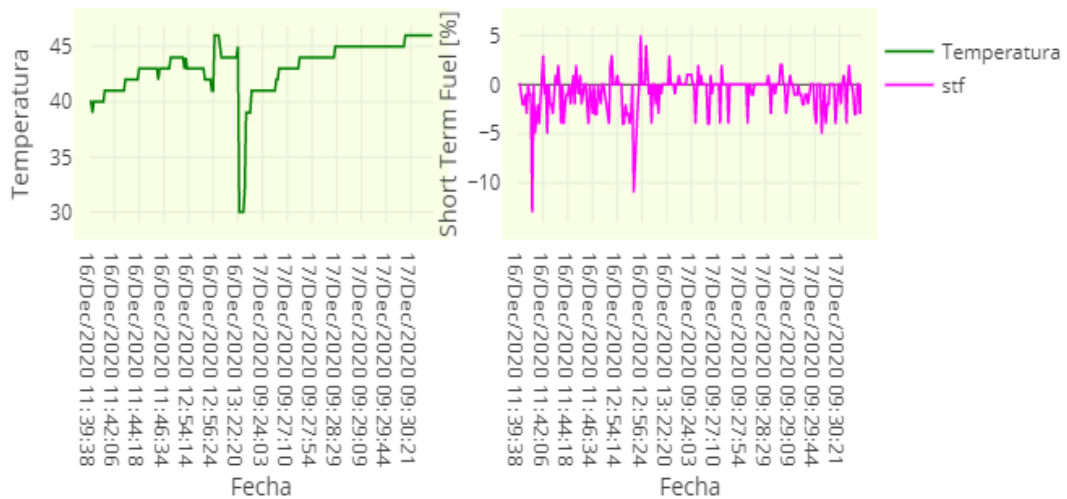


Figura 10-4. Temperatura y STF del vehículo – Escenario 1
 Realizado por: CARDOSO Cristhian, 2021.

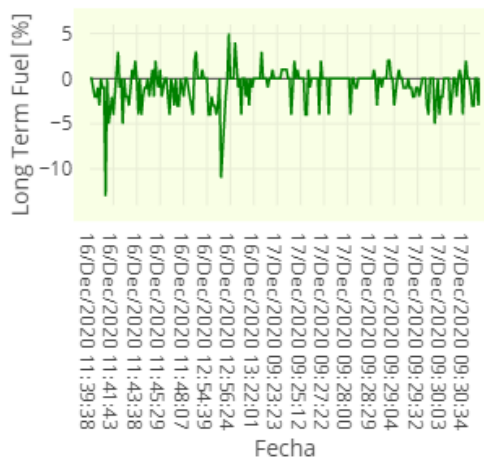


Figura 11-4. LTF del vehículo – Escenario 1
 Realizado por: CARDOSO Cristhian, 2021.

- **Visualización de datos en tiempo real: Lectura Analítica**

En la **Figura 12-4** se puede observar el modo de presentación los datos almacenados en la página web, los cuales se ordenan a medida que se cargan los datos, además que permite búsqueda de datos ya se por fecha o algún otro parámetro disponible dentro de la tabla, incluyendo opciones de descarga de los registros en diferentes formatos (EXCEL, CSV, PDF) y dos características adicionales las cuales son imprimir y copiar los datos.

LECTURA DE DATOS EN TIEMPO REAL CON OBD-II

Actualizar Tabla Regresar Graficas ▾

Fecha - Hora	Latitud	Longitud	Altitud [m]	RPM	Velocidad [Km/h]	Temperatura [°C]	Short Term Fuel [%]	Short Term Fuel [%]
16/Dec/2020 11:39:38	-1.66903383333	-78.6467038333	2764.4	777	0	40	0	0
16/Dec/2020 11:39:44	-1.6690345	-78.646699	2764.4	786	0	40	0	0
16/Dec/2020 11:39:54	-1.66903666667	-78.6466946667	2764.4	15428	18	39	-1	-1
16/Dec/2020 11:40:00	-1.66903333333	-78.6466941667	2764.4	859	0	40	-2	-2
16/Dec/2020 11:40:08	-1.6690335	-78.6466905	2764.5	809	0	40	-2	-2
16/Dec/2020 11:40:15	-1.66903683333	-78.6466896667	2764.6	941	6	40	-1	-1
16/Dec/2020 11:40:21	-1.669035	-78.6466883333	2764.7	803	0	40	-3	-3
16/Dec/2020 11:40:28	-1.66903433333	-78.6466865	2765.1	800	0	40	0	0
16/Dec/2020 11:40:35	-1.66903333333	-78.6466875	2765.1	2674	23	40	-1	-1
16/Dec/2020 11:40:44	-1.66903116667	-78.6466885	2765.0	871	3	40	-1	-1

18 =

Anterior 1 2 3 4 5 ... 26 Siguiente

Figura 12-4. Representación de los datos obtenidos en el servidor – Escenario 1
Realizado por: CARDOSO Cristhian, 2021.

La **Tabla 2-4** muestra una parte de los datos extraídos y almacenados en el transcurso de la ruta especificada en el escenario 1. Como se puede observar en el servidor los datos extraídos se almacenaron en un intervalo de 7 segundos, tiempo que fue definido al momento de ejecutar el dispositivo móvil, la variación de tiempo se debe a la duración de extracción de los datos de los sensores mediante el lector ELM327, puesto que posee un retardo de 500 milisegundos por petición de PID además del proceso de estos para ser posteriormente almacenados en el servidor.

El tiempo de retraso hace referencia al tiempo de envío de la petición del lector ELM327 mediante comandos AT hacia la ECU del vehículo y otros 500 milisegundos en respuesta de la ECU hacia el lector ELM327, los datos a extraer son cinco, adicionalmente el tiempo de procesamiento de 1000 milisegundos, por lo cual el tiempo de retraso total es de:

$$t_T = (t_p + t_r) * n + t_{pros}$$

$$t_T = ((500 + 500) * 5 + 1000)ms$$

$$t_T = 7000 ms = 7 seg.$$

Donde,

t_T : es el tiempo total de extracción y procesamiento.

t_p : es el tiempo que tarda en cada envío de la petición.

t_r : es el tiempo que tarda en cada respuesta a la petición.

t_{pros} : tiempo de procesamiento de los datos.

n : número de peticiones.

Tabla 2-4. Reporte de los datos extraídos - Escenario 1

Fecha -Hora	Latitud	Longitud	Altitud [m]	RPM	Velocidad [Km/h]	Temperatura [°C]	Short Term Fluel [%]	Short Term Fluel [%]
16/Dec/2020 11:39:37	-1,6690338	-78,6467038	2764,4	777	0	40	0	0
16/Dec/2020 11:39:44	-1,6690345	-78,646699	2764,4	786	0	40	0	0
16/Dec/2020 11:39:53	-1,6690367	-78,6466947	2764,4	15428	18	39	-1	-1
16/Dec/2020 11:40:00	-1,6690333	-78,6466942	2764,4	859	0	40	-2	-2
16/Dec/2020 11:40:07	-1,6690355	-78,6466905	2764,5	809	0	40	-2	-2
16/Dec/2020 11:40:14	-1,6690368	-78,6466897	2764,6	941	6	40	-1	-1
16/Dec/2020 11:40:21	-1,669035	-78,6466883	2764,7	803	0	40	-3	-3
16/Dec/2020 11:40:28	-1,6690343	-78,6466865	2765,1	800	0	40	0	0
16/Dec/2020 11:40:35	-1,6690333	-78,6466875	2765,1	2674	23	40	-1	-1
16/Dec/2020 11:40:42	-1,6690312	-78,6466885	2765	871	3	40	-1	-1
16/Dec/2020 11:40:49	-1,669032	-78,646687	2764,9	1751	20	40	-13	-13
16/Dec/2020 11:40:56	-1,6690345	-78,6466857	2764,7	1759	22	41	0	0
16/Dec/2020 11:41:03	-1,6690357	-78,646686	2764,5	1188	20	41	-5	-5
16/Dec/2020 11:41:10	-1,6690367	-78,6466883	2764,3	830	0	41	-3	-3
16/Dec/2020 11:41:17	-1,6690507	-78,6466852	2764,1	1361	10	41	-2	-2
16/Dec/2020 11:41:24	-1,6690767	-78,6466885	2764	798	0	41	-4	-4
16/Dec/2020 11:41:31	-1,6690998	-78,6467263	2763,9	790	0	41	-2	-2
16/Dec/2020 11:41:38	-1,669139	-78,6467463	2763,6	784	3	41	0	0
16/Dec/2020 11:41:45	-1,6691807	-78,646768	2763,7	864	6	41	3	3
16/Dec/2020 11:41:52	-1,6692062	-78,6468133	2764,2	1222	9	41	-1	-1
16/Dec/2020 11:41:59	-1,6692365	-78,6468547	2764,2	897	7	41	0	0
16/Dec/2020 11:42:06	-1,6692687	-78,6468898	2764,4	837	0	41	-5	-5
16/Dec/2020 11:42:13	-1,6693027	-78,6469193	2764,6	984	0	41	0	0

Realizado por: CARDOSO Cristhian, 2021.

4.2.2 Escenario 2: Hyundai i10

- **Pruebas de adquisición de datos con el lector ELM327**

Se repite el proceso, como en la sección 4.2.1.

```
>AUTO, ISO 14230-4 (KWP FAST)
>01 0C
>41 0C 0C C8
```

Figura 13-4. Solicitud de las revoluciones del vehículo

Realizado por: CARDOSO Cristhian, 2021.

- **Ruta Ejecutada**

El escenario corresponde a un tramo recorrido por el bus urbano de la ciudad de Riobamba Línea 15 (del sector la Plaza de las hierbas hacia el Parque General Barriga), realizando tracking y monitoreo de los sensores del motor del vehículo Hyundai i10.



Figura 14-4. Ruta recorrida (Google Earth Pro) – Escenario 2

Realizado por: CARDOSO Cristhian, 2021.

- **Visualización de datos en tiempo real: Lectura Gráfica**

Se verifica en la plataforma la adquisición de la información vehicular extraída. Ver **Figuras 15-4 a 18-4.**

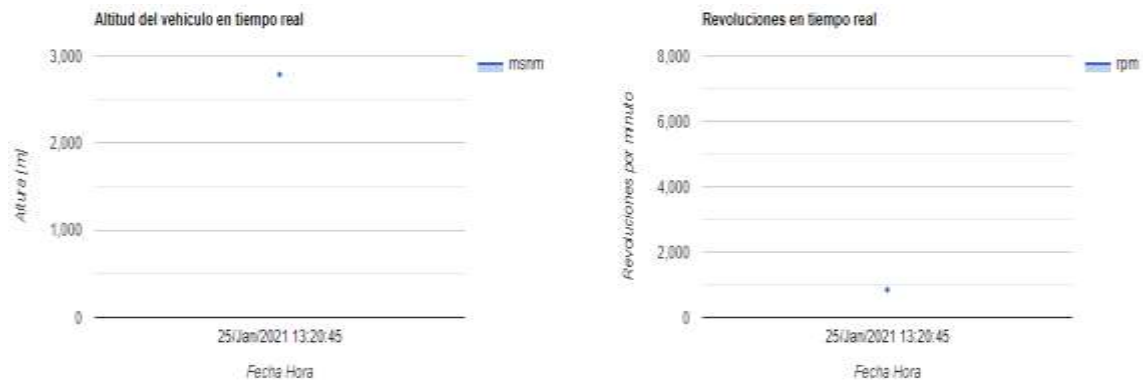


Figura 15-4. Altura y revoluciones del vehículo en tiempo real – Escenario 2
Realizado por: CARDOSO Cristhian, 2021.

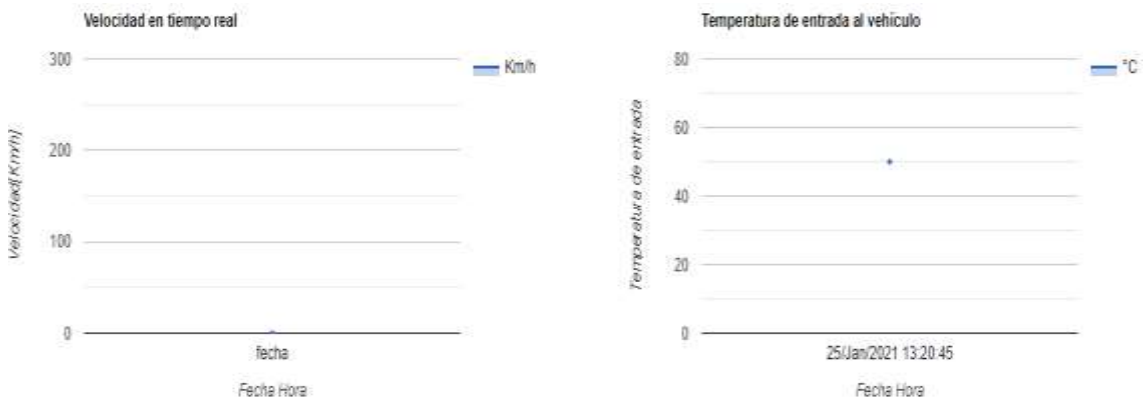


Figura 16-4. Velocidad y temperatura del vehículo en tiempo real – Escenario 2
Realizado por: CARDOSO Cristhian, 2021.

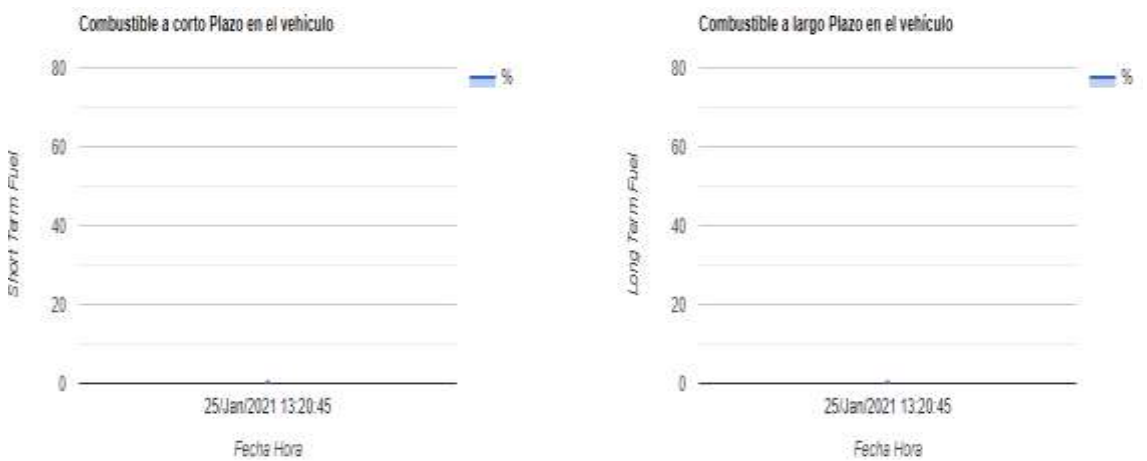


Figura 17-4. Representación del STF y LTF del vehículo en tiempo real – Escenario 2
Realizado por: CARDOSO Cristhian, 2021.



Figura 18-4. Visualización de los datos en tiempo real – Escenario 2
 Realizado por: CARDOSO Cristhian, 2021.

En las **Figuras 19-4 hasta 22-4**, se puede apreciar las gráficas obtenidas de los datos adquiridos por el dispositivo mediante el recorrido antes mencionado.

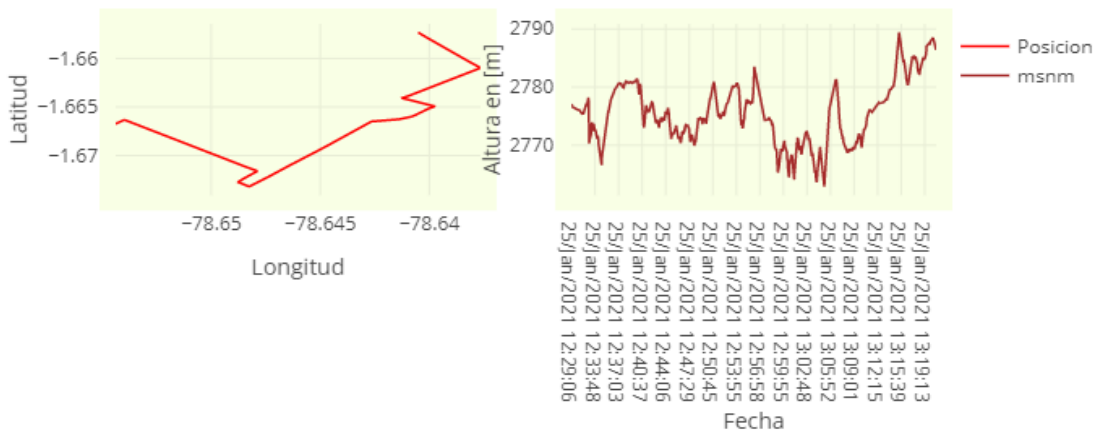


Figura 19-4. Coordenadas y Altura (msnm) del vehículo – Escenario 2
 Realizado por: CARDOSO Cristhian, 2021.

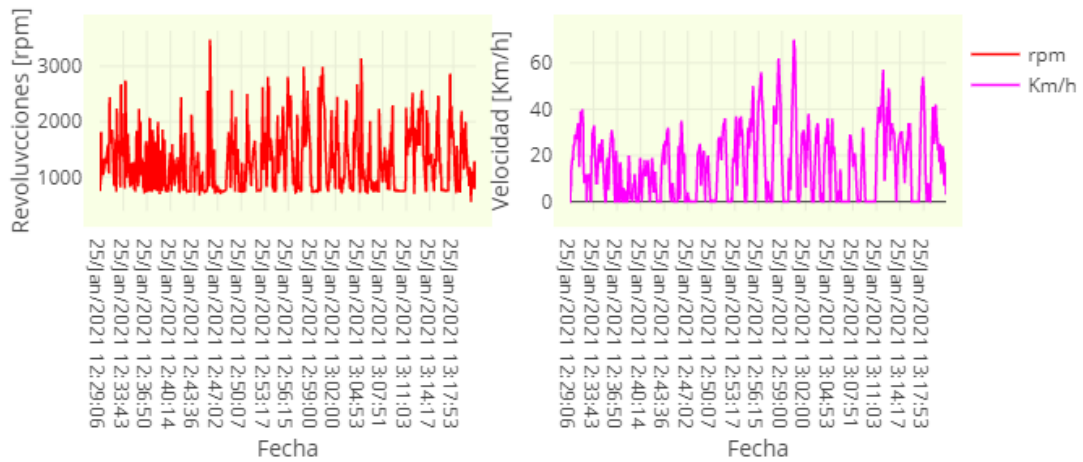


Figura 20-4. RPM y velocidad del vehículo – Escenario 2
 Realizado por: CARDOSO Cristhian, 2021.

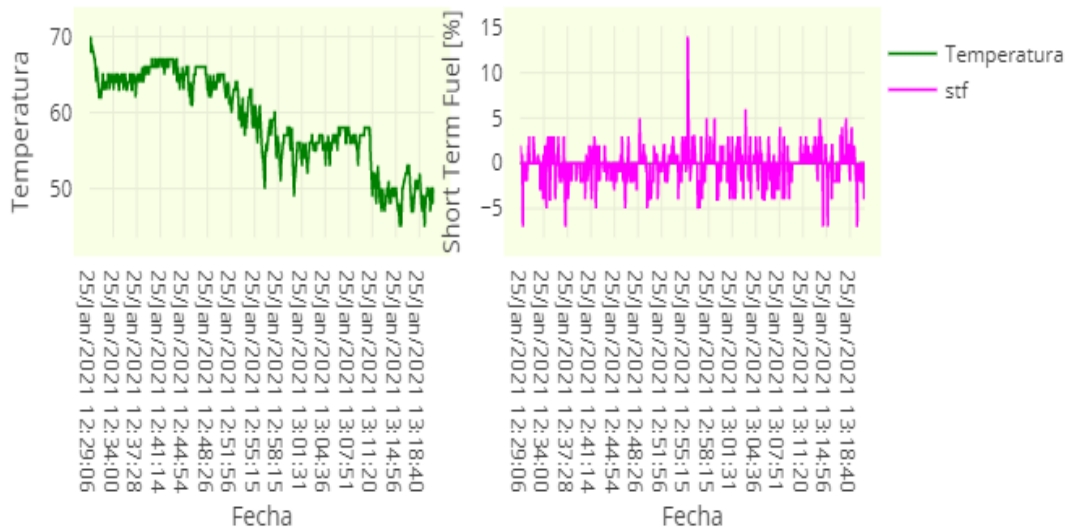


Figura 21-4. Temperatura y STF del vehículo – Escenario 2
 Realizado por: CARDOSO Cristhian, 2021.

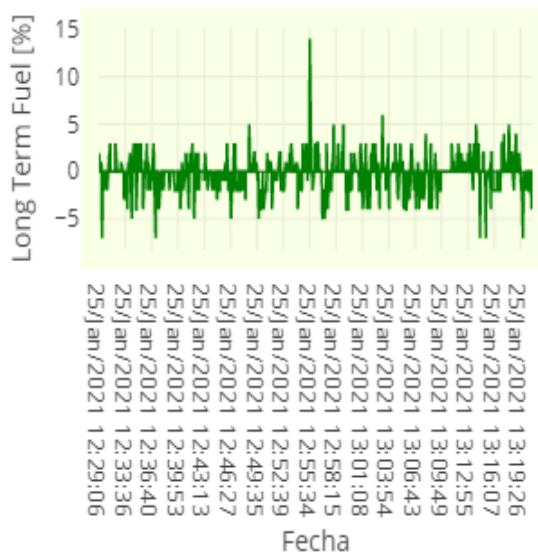


Figura 22-4. LTF del vehículo – Escenario 2
 Realizado por: CARDOSO Cristhian, 2021.

- **Visualización de datos en tiempo real: Lectura Analítica**

Se visualiza de forma ordenada en la plataforma la adquisición de los datos del vehículo extraídos mediante el sistema, como se muestra en la **Figura 23-4**, con sus respectivas características adicionales de la página web.

LECTURA DE DATOS EN TIEMPO REAL CON OBD-II

Actualizar Tabla Regresar Graficas

Fecha - Hora	Latitud	Longitud	Altitud [m]	RPM	Velocidad [Km/h]	Temperatura [°C]	Short Term Fuel [%]	Short Term Fuel [%]
25/Jan/2021 12:29:06	-1.657323333333	-78.640507	2777.1	750	0	69	2	2
25/Jan/2021 12:30:25	-1.657323833333	-78.6405056667	2776.6	765	0	70	0	0
25/Jan/2021 12:30:30	-1.657323833333	-78.6405056667	2776.6	904	6	69	1	1
25/Jan/2021 12:30:32	-1.657324	-78.6405056667	2776.3	951	7	70	0	0
25/Jan/2021 12:30:35	-1.657324	-78.6405058333	2776.5	1811	14	68	0	0
25/Jan/2021 12:30:37	-1.657326666667	-78.6405046667	2776.3	1101	15	68	0	0
25/Jan/2021 12:30:39	-1.657336333333	-78.6404995	2776.3	1125	15	69	1	1
25/Jan/2021 12:30:43	-1.65735116667	-78.6404908333	2776.4	1472	19	68	-5	-5
25/Jan/2021 12:30:46	-1.6573665	-78.6404816667	2776.4	1307	17	69	-7	-7
25/Jan/2021 12:30:50	-1.65738116667	-78.6404733333	2776.4	1322	18	69	0	0

Anterior 1 2 3 4 5 ... 106 Siguiente

Figura 23-4. Representación de los datos obtenidos en el servidor – escenario 2
 Realizado por: CARDOSO Cristhian, 2021.

La **Tabla 3-4** muestra una parte de los datos extraídos y almacenados en el transcurso de la ruta especificada en el escenario 2. Como se puede observar en el servidor los datos extraídos se almacenaron en un intervalo de 3 segundos, tiempo que fue definido al momento de ejecutar el sistema, la variación de tiempo se debe a la duración de extracción de los datos de los sensores mediante el lector ELM327, puesto que posee un retardo de 250 milisegundos, tiempo que fue mejorado para cada petición de PID.

El tiempo de retraso hace referencia al tiempo de envío de la petición del lector ELM327 mediante comandos AT hacia la ECU del vehículo y otros 250 milisegundos en respuesta de la ECU hacia el lector ELM327, los datos a extraer son cinco, adicionalmente el tiempo de procesamiento de 500 milisegundos, por lo cual el tiempo de retraso total es de:

$$t_T = (t_p + t_r) * n + t_{pros}$$

$$t_T = ((250 + 250) * 5 + 500)ms$$

$$t_T = 3000 ms = 3 seg.$$

Tabla 3-4. Reporte de los datos extraídos - Escenario 2

Datos Registrados								
Fecha -Hora	Latitud	Longitud	Altitud [m]	RPM	Velocidad [Km/h]	Tem Peratura [°C]	Short Term Fluel [%]	Short Term Fluel [%]
25/Jan/2021 12:30:32	-1,657324	-78,6405057	2776,5	951	7	70	0	0
25/Jan/2021 12:30:35	-1,657324	-78,6405058	2776,5	1811	14	68	0	0
25/Jan/2021 12:30:37	-1,65732667	-78,6405047	2776,5	1101	15	68	0	0
25/Jan/2021 12:30:39	-1,65733633	-78,6404995	2776,5	1125	15	69	1	1
25/Jan/2021 12:31:06	-1,657492	-78,6403965	2776,2	1304	28	67	0	0
25/Jan/2021 12:31:08	-1,6575225	-78,6403718	2776,2	1347	29	67	0	0
25/Jan/2021 12:31:11	-1,65755267	-78,6403468	2776,2	1310	28	67	0	0
25/Jan/2021 12:31:16	-1,6575825	-78,6403227	2776,2	1238	27	66	0	0
25/Jan/2021 12:31:18	-1,65761417	-78,6402978	2776,2	1341	29	65	0	0
25/Jan/2021 12:31:21	-1,6576485	-78,6402702	2776,1	1371	30	64	0	0
25/Jan/2021 12:31:23	-1,65768767	-78,6402398	2776,2	1441	31	64	-2	-2
25/Jan/2021 12:31:28	-1,65772883	-78,6402085	2776,1	1554	34	65	0	0
25/Jan/2021 12:31:31	-1,65776967	-78,6401778	2776	1476	32	66	0	0
25/Jan/2021 12:31:33	-1,65780933	-78,6401472	2776	1577	34	65	0	0
25/Jan/2021 12:31:36	-1,65784567	-78,6401173	2776	1271	27	66	0	0
25/Jan/2021 12:31:39	-1,6578765	-78,6400915	2775,9	1143	25	66	2	2
25/Jan/2021 12:31:48	-1,657904	-78,6400697	2775,7	1062	15	64	1	1
25/Jan/2021 12:31:50	-1,65793117	-78,6400475	2775,6	1875	26	63	3	3
25/Jan/2021 12:31:52	-1,657964	-78,6400202	2775,5	2445	29	63	0	0
25/Jan/2021 12:31:55	-1,6580035	-78,6399878	2775,5	1609	36	62	0	0
25/Jan/2021 12:31:59	-1,65804567	-78,639952	2775,5	1806	39	62	0	0
25/Jan/2021 12:32:02	-1,65808983	-78,639913	2775,3	1606	35	63	0	0
25/Jan/2021 12:32:04	-1,658137	-78,6398738	2775,5	1646	36	63	0	0
25/Jan/2021 12:32:06	-1,65818433	-78,6398337	2775,5	1725	37	62	0	0
25/Jan/2021 12:32:11	-1,65823367	-78,6397932	2775,3	1853	40	62	0	0
25/Jan/2021 12:32:14	-1,65828233	-78,6397507	2775,5	1830	40	62	0	0
25/Jan/2021 12:32:16	-1,65833133	-78,639711	2775,7	1687	36	63	0	0
25/Jan/2021 12:32:18	-1,65838283	-78,6396733	2776	1550	34	63	0	0
25/Jan/2021 12:32:23	-1,65843517	-78,6396335	2776,1	1343	17	64	0	0
25/Jan/2021 12:32:25	-1,65848783	-78,6395943	2776,2	917	12	64	0	0

Realizado por: CARDOSO Cristhian, 2021.

4.2.3 Escenario 3: Chevrolet Aveo Family

- **Pruebas de adquisición de datos con el lector ELM327**

Se realiza el proceso de comprobación, como en la sección 4.2.1.

```
>AUTO,ISO 15765-4 (CAN 11/500)
>01 0F
>41 0F 4C
```

Figura 24-4. Solicitud de la temperatura del vehículo
Realizado por: CARDOSO Cristhian, 2021.

- **Ruta Ejecutada**

El escenario corresponde a un tramo del sector la Dávalos hacia la parte posterior de la Universidad Nacional de Chimborazo recorrido por el vehículo en la ciudad de Riobamba, realizando tracking y monitoreo de los sensores del motor del vehículo Chevrolet Aveo Family.



Figura 25-4. Ruta recorrida (Google Earth Pro) – Escenario 3
Realizado por: CARDOSO Cristhian, 2021.

- **Visualización de datos en tiempo real: Lectura Gráfica**

Se verifica la adquisición de la información vehicular extraída en la plataforma. Ver las **Figuras 26-4 a 29-4**.

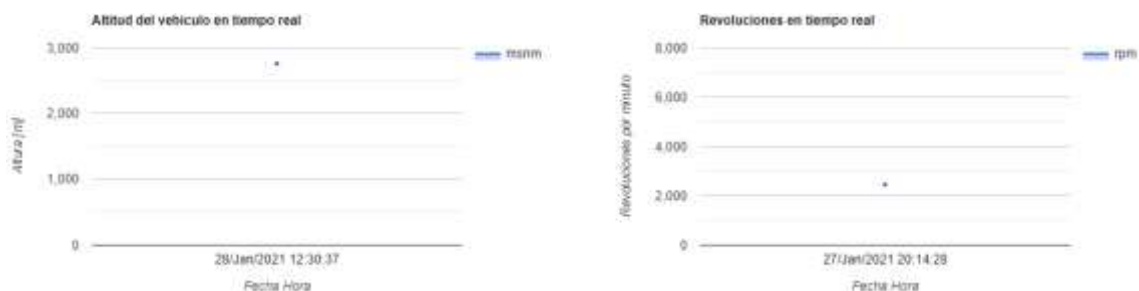


Figura 26-4. Altura y revoluciones del vehículo en tiempo real – Escenario 3
Realizado por: CARDOSO Cristhian, 2021.

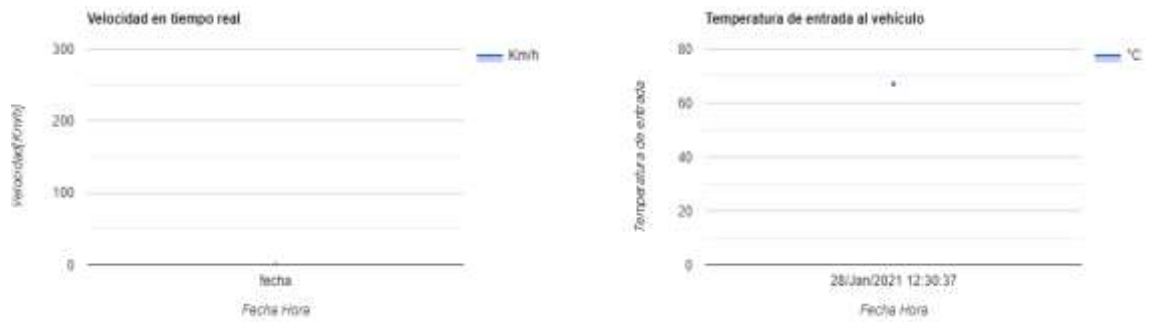


Figura 27-4. Velocidad y temperatura del vehículo en tiempo real – Escenario 3
Realizado por: CARDOSO Cristhian, 2021.

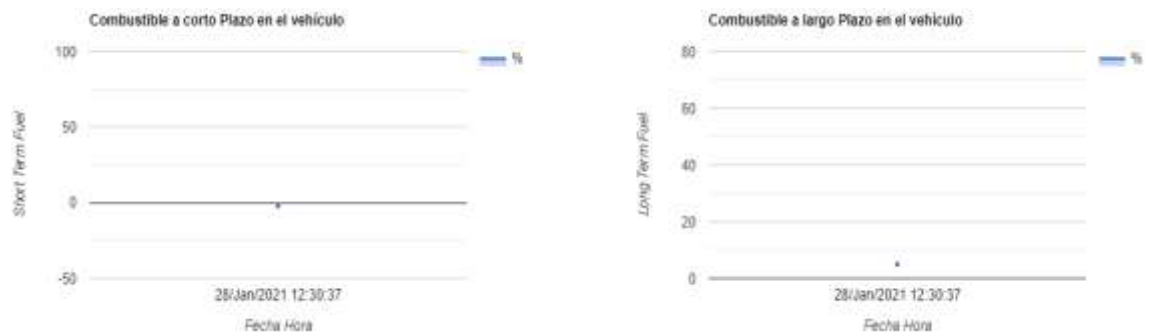


Figura 28-4. STF y LTF del vehículo en tiempo real – Escenario 3
Realizado por: CARDOSO Cristhian, 2021.



Figura 29-4. Visualización de los datos en tiempo real – Escenario 3
Realizado por: CARDOSO Cristhian, 2021.

En las **Figuras 30-4 a 33-4** se puede apreciar las gráficas generadas con todos los datos adquiridos por el sistema.

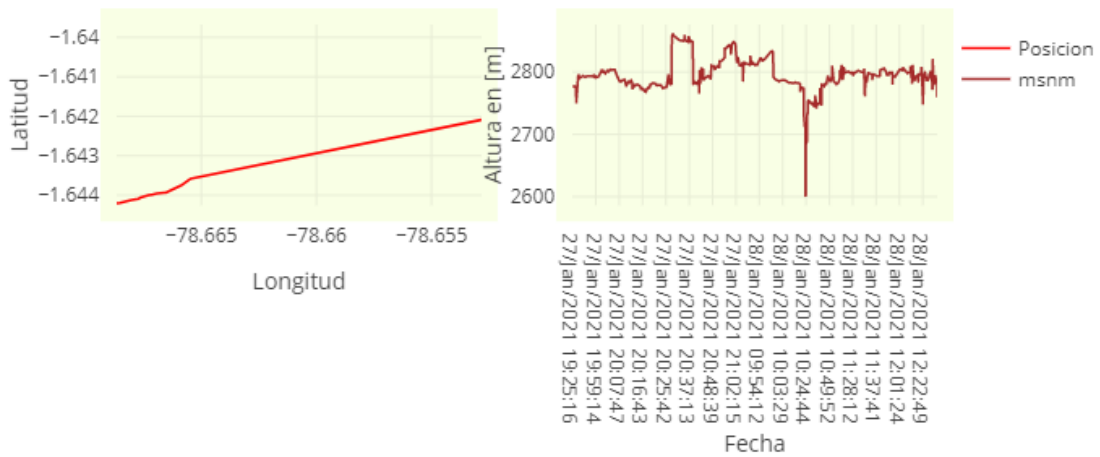


Figura 30-4. Coordenadas y Altura (msnm) del vehículo – Escenario 3
 Realizado por: CARDOSO Cristhian, 2021.

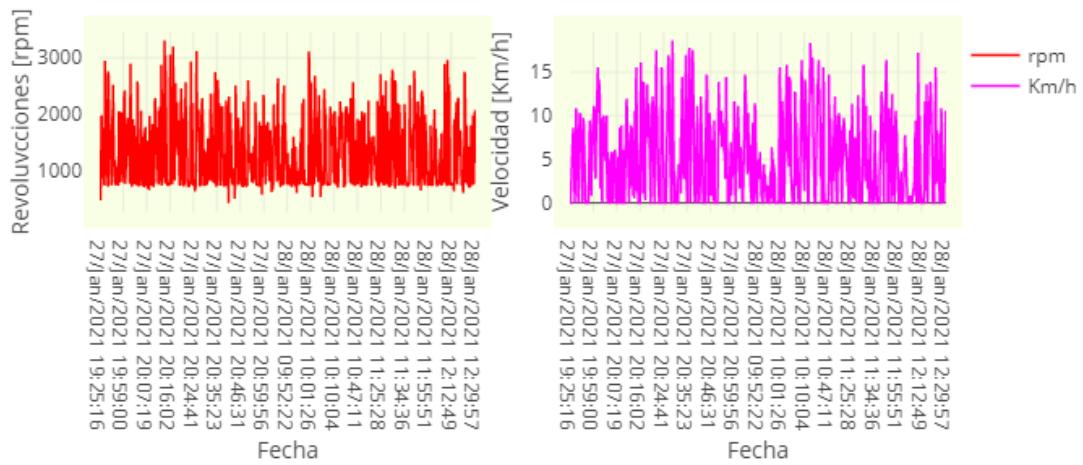


Figura 31-4. RPM y velocidad del vehículo – Escenario 3
 Realizado por: CARDOSO Cristhian, 2021.

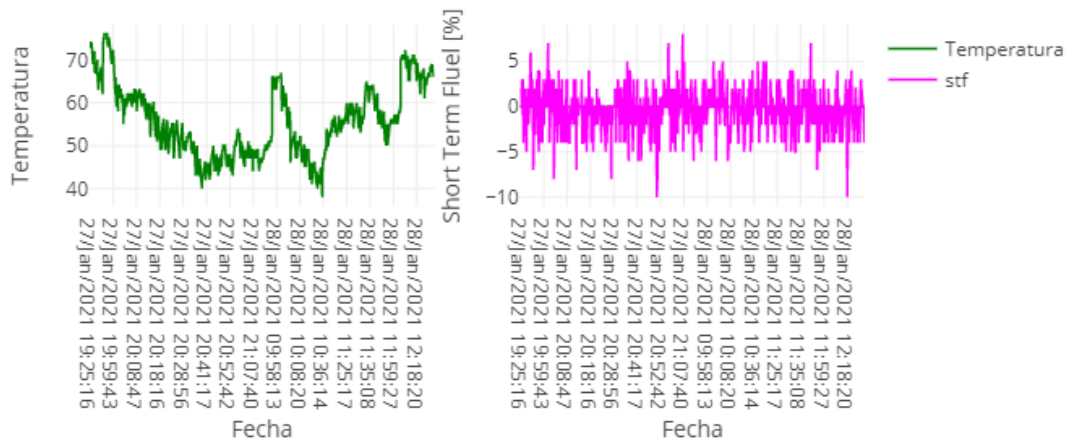


Figura 32-4. Temperatura y STF del vehículo – Escenario 3
 Realizado por: CARDOSO Cristhian, 2021.

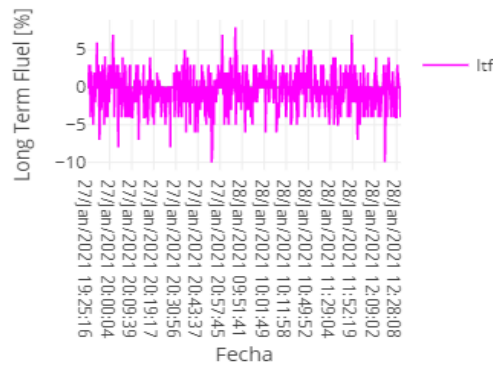


Figura 33-4. LTF del vehículo – Escenario 3
Realizado por: CARDOSO Cristhian, 2021.

- **Visualización de datos en tiempo real: Lectura Analítica**

Se visualiza de forma ordenada en la plataforma la adquisición de los datos del vehículo extraídos mediante el sistema, como se muestra en la **Figura 34-4**, con sus respectivas características adicionales de la página web.

LECTURA DE DATOS EN TIEMPO REAL CON OBD-II

Actualizar Tabla Regresar Graficas

Fecha-Hora	Latitud	Longitud	Altitud [m]	RPM	Velocidad [Km/h]	Temperatura [°C]	Short Term Fuel [%]	Short Term Fuel [%]
27/Jan/2021 19:25:16	-1.657621	-78.6402375	2777.0	765.0	0.0	74	2	5
27/Jan/2021 19:25:18	-1.65762166667	-78.640238	2776.9	762.75	0.0	74	0	5
27/Jan/2021 19:25:20	-1.65762233333	-78.640238	2776.8	743.25	0.0	74	0	5
27/Jan/2021 19:25:22	-1.65762266667	-78.640238	2776.8	764.25	0.0	74	0	5
27/Jan/2021 19:25:25	-1.65762383333	-78.6402386667	2776.8	758.5	0.0	74	0	3
27/Jan/2021 19:25:27	-1.65762483333	-78.6402386667	2776.7	1107.25	0.0	74	0	5
27/Jan/2021 19:25:29	-1.6576255	-78.6402391667	2776.7	746.75	0.0	74	0	5
27/Jan/2021 19:25:31	-1.657626	-78.6402396667	2776.7	738.0	0.0	74	0	-2
27/Jan/2021 19:25:34	-1.6576265	-78.6402396667	2776.7	468.5	0.833333333333	74	0	5
27/Jan/2021 19:25:36	-1.65762633333	-78.6402391667	2776.7	979.5	1.94444444444	73	0	0

10

Anterior 1 2 3 4 5 ... 344 Siguiente

Figura 34-4. Representación de los datos obtenidos en el servidor – Escenario 3
Realizado por: CARDOSO Cristhian, 2021.

La **Tabla 4-4** muestra una parte de los datos extraídos y almacenados en el transcurso de la ruta especificada en el escenario 3. Como se puede observar en el servidor los datos extraídos se almacenaron en un intervalo de 2 a 3 segundos, tiempo que fue definido al momento de ejecutar el dispositivo móvil.

La variación de tiempo se debe a la duración de extracción de los datos de los sensores mediante el lector ELM327, puesto que posee un retardo de 100 milisegundos, tiempo que fue mejorado para cada petición de PID, además del proceso de estos para ser posteriormente almacenados en el servidor, este tiempo de retardo no pudo ser mejorado a menos de lo mencionado debido que si se ingresa un valor menor no existirá respuesta y no se podrá extraer ningún tipo de dato con el lector ELM327. El tiempo que tarda se detalla a continuación:

$$t_T = (t_p + t_r) * n + t_{pros}$$

$$t_T = ((100 + 100) * 5 + 500)ms$$

$$t_T = 2500 ms = 2,5 seg.$$

Tabla 4-4. Reporte de los datos extraídos - Escenario 3

Datos Registrados								
Fecha -Hora	Latitud	Longitud	Altitud [m]	RPM	Velocidad [Km/h]	Temperatura [°C]	Short Term Fuel [%]	Short Term Fuel [%]
27/Jan/2021 19:25:16	-1,657621	-78,6402375	2777	765	0	74	2	5
27/Jan/2021 19:25:18	-1,65762167	-78,640238	2776,9	762,75	0	74	0	5
27/Jan/2021 19:25:20	-1,65762233	-78,640238	2776,8	743,25	0	74	0	5
27/Jan/2021 19:25:22	-1,65762267	-78,640238	2776,8	764,25	0	74	0	5
27/Jan/2021 19:25:25	-1,65762383	-78,6402387	2776,8	758,5	0	74	0	3
27/Jan/2021 19:25:27	-1,65762483	-78,6402387	2776,7	1107,25	0	74	0	5
27/Jan/2021 19:25:29	-1,6576255	-78,6402392	2776,7	746,75	0	74	0	5
27/Jan/2021 19:25:31	-1,657626	-78,6402397	2776,7	738	0	74	0	-2
27/Jan/2021 19:25:34	-1,6576265	-78,6402397	2776,7	468,5	3	74	0	5
27/Jan/2021 19:25:36	-1,65762633	-78,6402392	2776,7	979,5	7	73	0	0
27/Jan/2021 19:25:38	-1,65762633	-78,6402387	2776,8	1094,5	7	73	0	5
27/Jan/2021 19:25:40	-1,657626	-78,6402377	2776,8	1136,75	8	73	0	5
27/Jan/2021 19:25:43	-1,65762583	-78,6402367	2776,7	1109,75	7	73	0	-2
27/Jan/2021 19:25:45	-1,65762683	-78,6402358	2776,6	1157,25	8	73	1	-2
27/Jan/2021 19:25:47	-1,65762717	-78,6402343	2776,5	1545,75	11	73	0	-2
27/Jan/2021 19:25:49	-1,657628	-78,6402333	2776,5	1875	13	73	3	-3
27/Jan/2021 19:25:52	-1,657629	-78,6402318	2776,5	1967	14	73	0	-3
27/Jan/2021 19:25:54	-1,65763033	-78,6402308	2776,5	1965	14	73	0	5
27/Jan/2021 19:25:56	-1,65763367	-78,6402295	2776,6	1005	13	73	0	3

Continuación de la **Tabla 4-4**

27/Jan/2021 19:26:00	-1,65764083	-78,6402258	2776,8	1014,75	14	74	0	1
27/Jan/2021 19:26:02	-1,65765583	-78,6402238	2777,1	1120,75	15	73	0	0
27/Jan/2021 19:26:04	-1,65767283	-78,6402187	2777	1205,25	16	73	0	0
27/Jan/2021 19:26:07	-1,6576885	-78,6402085	2776,7	1610,5	22	72	0	0
27/Jan/2021 19:26:09	-1,65770533	-78,6401965	2776,7	1572,25	21	72	-1	0
27/Jan/2021 19:26:11	-1,6577195	-78,6401827	2776,5	1763	24	72	0	0
27/Jan/2021 19:26:14	-1,65773583	-78,6401683	2776,4	1131,25	24	72	0	5
27/Jan/2021 19:26:17	-1,657775	-78,640153	2776,6	1069	23	73	0	0
27/Jan/2021 19:26:19	-1,6577655	-78,6401373	2776,3	1175	26	72	2	0
27/Jan/2021 19:26:22	-1,657781	-78,640122	2776,1	1323	29	71	3	0
27/Jan/2021 19:26:24	-1,657796	-78,6401075	2776,1	1406,5	31	70	0	0
27/Jan/2021 19:26:26	-1,65781233	-78,6400933	2775,7	1420	31	69	0	5
27/Jan/2021 19:26:28	-1,6578305	-78,6400787	2775,5	985,5	28	70	-1	0
27/Jan/2021 19:26:31	-1,65784967	-78,6400622	2775,3	988	29	70	0	0

Realizado por: CARDOSO Cristhian, 2021.

• **Cálculo de tráfico**

El sistema trabaja bajo la arquitectura TCP/IP sobre internet, en donde UDP fue el protocolo de transporte utilizado. A continuación, se detalla el cálculo del tráfico producido durante la transmisión de la información hacia el servidor, los resultados son de gran utilidad para estimar costos de comunicación y almacenamiento, estos resultados varían dependiendo el escenario, puesto que cada uno envía los datos a diferentes intervalos de tiempo.

La cantidad de bytes utilizados para los mensajes UDP a enviarse es de 33 bytes, los cuales se obtienen de las 5 medidas obtenidas de los sensores del vehículo (6 bytes), la fecha de extracción (13 bytes), de los datos de posicionamiento (6 bytes) y de los separadores de cada dato (8 bytes). Estos datos independientemente del protocolo que posea el vehículo tendrán el mismo tamaño y a su vez la respuesta a la petición será la misma en cualesquiera de los dos vehículos de prueba, motivo por el cual se realizó las pruebas en 2 escenarios con el vehículo Aveo y 1 con el Hyundai.

Capa de transporte:

Bytes de información: 33 bytes.

Bytes de cabecera para mensajes UDP: 8 bytes.

Total bytes del mensaje UPD: 41 bytes.

Capa de internet:

Bytes de información: 41 bytes.

Bytes de cabecera para datagrama IP: 24 bytes.

Total bytes para datagrama IP: 65 bytes.

Para la capa de acceso a la red de arquitectura GPRS se dispone de dos subcapas: LLC (*Logical Link Control*) y RLC/MAC (*Radio Link Control / Medium Access Control*), donde cada uno de estas posee su propio formato de trama.

Subcapa LLC:

Bytes de información: 65 bytes.

Bytes de cabecera LLC: 37 bytes.

Bytes de cola LLC: 3 bytes.

Total bytes para LLC: 105 bytes.

Subcapa RLC/MAC:

Bytes de información: 105 bytes.

Numero de segmentos: 105 bytes/23 bytes. = 4,56

Numero de segmentos: 5 segmentos.

Bytes total de información: 5*57 bytes.

Bytes de información: 285 bytes.

Considerando el tiempo de envío de los datos para cada escenario, se calcula el total que se produce en un día, en un mes y a la vez el que se producirá en un año, como se detalla en la **Tabla 5-4** y en la **Tabla 6-4** se detalla el cálculo del almacenamiento de información en la base de datos.

Tabla 5-4. Cálculo de tráfico de envío de datos

	Escenario 1 (7 segundos)	Escenario 2 (3 segundos)	Escenario 3 (2,5 segundos)
<i>Bytes de información en minutos</i>	2565	5700	6840
<i>Bytes de información en días</i>	3693600	8208000	9849600
<i>Bytes de información en meses</i>	110808000	246240000	295488000
<i>Bytes de información en días</i>	3,523 Mb	7,828 Mb	9,393 Mb
<i>Bytes de información en meses</i>	105,675 Mb	234,833 Mb	281,80 Mb

Realizado por: CARDOSO Cristhian, 2021.

Tabla 6-4. Cálculo de datos a ser almacenados

	Escenario 1 (7 segundos)	Escenario 2 (3 segundos)	Escenario 3 (2,5 segundos)
<i>Bytes de información en minutos</i>	297	660	792
<i>Bytes de información en días</i>	427680	950400	1140480
<i>Bytes de información en meses</i>	12830400	28512000	34214400
<i>Bytes de información en años</i>	153964800	342144000	410572800
<i>Bytes de información en días</i>	417,656 Kb	928,125 Kb	1,088 Mb
<i>Bytes de información en meses</i>	12,236 Mb	27,191 Mb	32,630 Mb
<i>Bytes de información en años</i>	146,832 Mb	326,294 Mb	391,553 Mb

Realizado por: CARDOSO Cristhian, 2021.

- **Presupuesto de Operadoras móviles**

La transmisión de información de los parámetros mencionados extraídos por medio del OBD-II deben ser enviados a la plataforma de almacenamiento como se demostró en la sección 4.2. Es importante definir la tecnología de comunicación afín a las características de funcionamiento del sistema. La lectura y extracción de datos se debe realizar cuando el vehículo se encuentra en movimiento por lo que se opta por un medio de comunicación inalámbrico por sus ventajas y facilidad de conexión. De éstas se considera los servicios móviles avanzados SMA, debido a la cobertura que ofrece, actualmente la tecnologías disponibles en el país por las principales operadoras ofrecen velocidades de hasta 100 Mbps en LTE 4G.

Tabla 7-4. Velocidad de las tecnologías móviles

<i>Operadora Movil</i>	4G	3 G	2 G
<i>Movistar, Claro, CNT</i>	100 Mbps hasta 1Gbps	Hasta 2Mbps	Hasta 64 kbps

Realizado por: CARDOSO Cristhian, 2021.

- **Cobertura de las operadoras**

En las **Figuras 35-4 a 37-4** se puede visualizar la cobertura brindada por las tres operadoras principales del país, las cuales fueron enfocadas en la ciudad de Riobamba, se puede decir que el uso de la red móvil dentro de la ciudad es bastante viable, ya que la cobertura de cualesquiera de estas operadoras es bastante buena.

Movistar

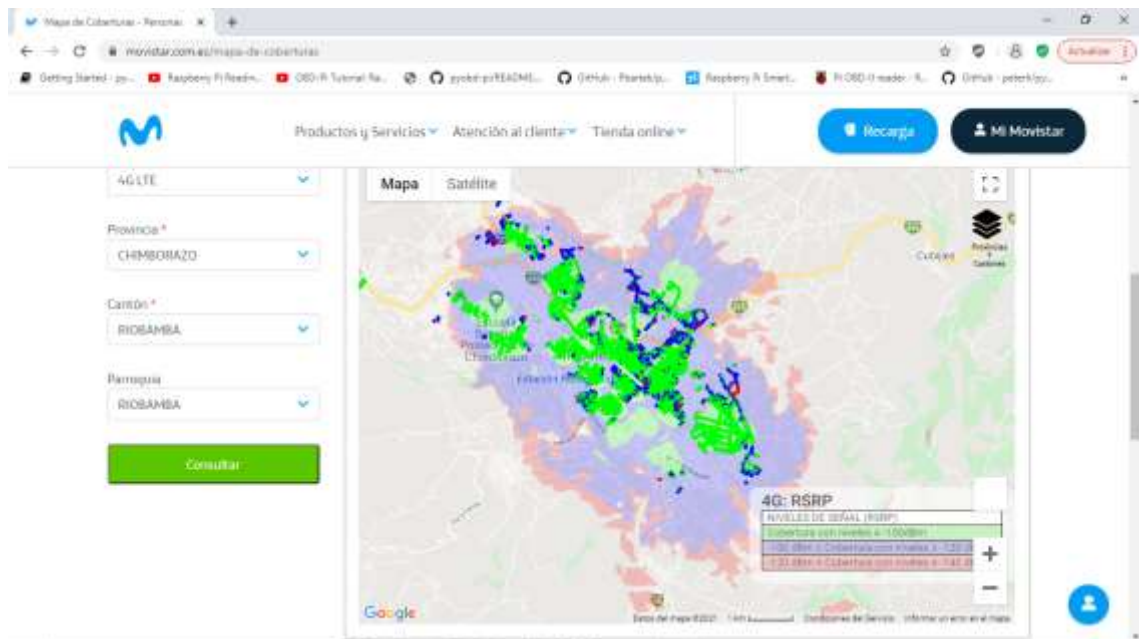


Figura 35-4. Cobertura operadora Movistar en la ciudad de Riobamba

Realizado por: CARDOSO Cristhian, 2021.

Claro

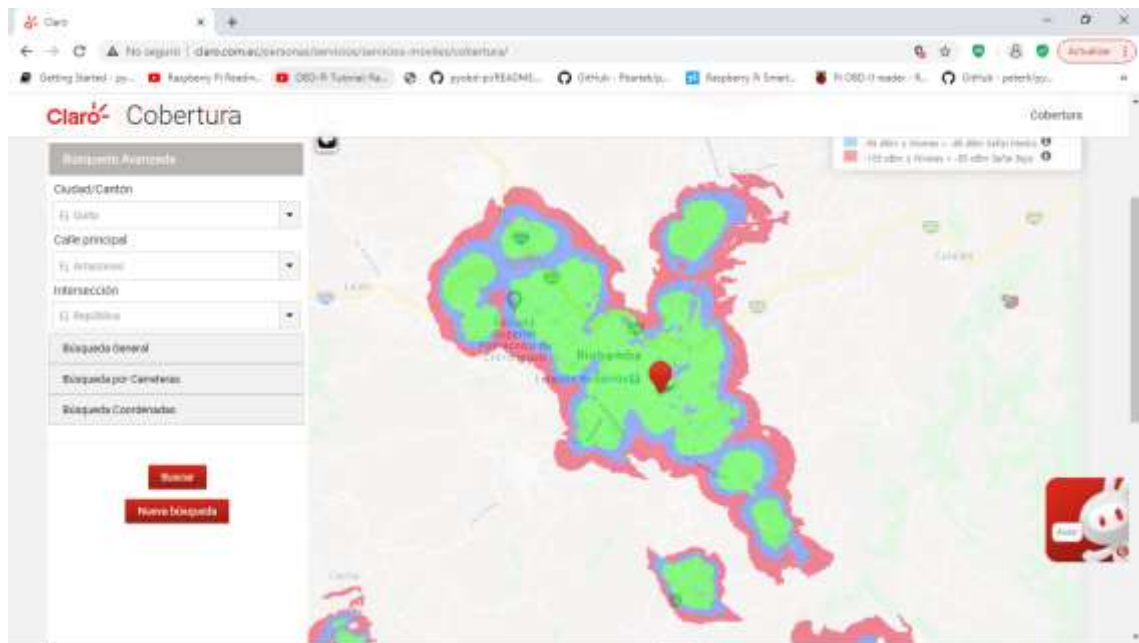


Figura 36-4. Cobertura operadora Claro en la ciudad de Riobamba
Realizado por: CARDOSO Cristhian, 2021.

CNT

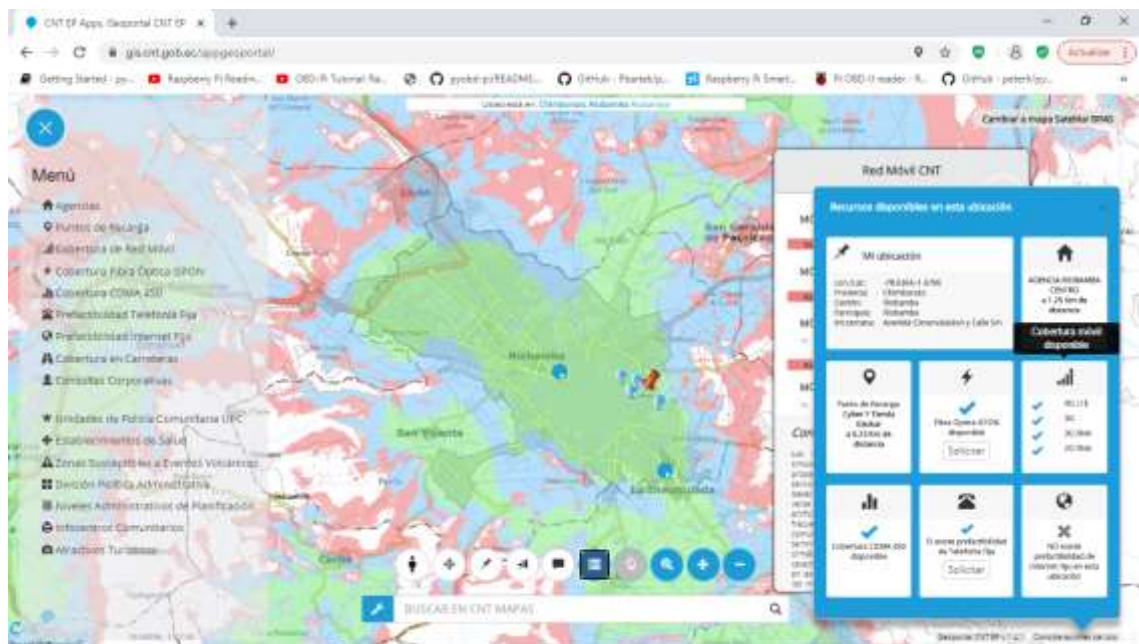


Figura 37-4. Cobertura operadora CNT en la ciudad de Riobamba
Realizado por: CARDOSO Cristhian, 2021.

El paquete generado según **Tabla 5-4** en 2,5 segundos de lectura genera un tamaño de 6840 bytes/minuto por la trama de 27 bytes de información a ser enviados, considerando escenarios en donde la lectura sea de 12 horas diarias en un mes, el consumo de datos se incrementa a:

$$6840 \text{ bytes/min} * 60 \text{ min} * 12 \text{ horas} = 4,992 \text{ Mbyte al día} \quad \text{Ecuación 1.4}$$

Los costos del servicio ofrecidos por las operadoras móviles incluyen servicios de voz, datos, streaming y redes sociales, además de servicios roaming en el caso de un plan premium, este depende de la cantidad de minutos y megabytes de consumo que seleccione el usuario, entre los planes disponibles por las operadoras se tiene:

Plan Básico

Tabla 8-4. Costo plan básico
Servicios

	Claro	Movistar	CNT
<i>Voz (Minutos)</i>	100 min	150 min	50 min
<i>Datos (GB)</i>	7 GB	9 GB	2,5 GB
<i>Datos para Streaming (GB)</i>	5 GB	3 GB	0,5 GB
<i>Redes Sociales</i>	Ilimitado	Ilimitado	Ilimitado
<i>Costo (mes)</i>	\$12,32	\$14,55	\$12,90

Realizado por: CARDOSO Cristhian, 2021.

Plan Premium

Tabla 9-4. Costo plan premium
Servicios

	Claro	Movistar	CNT
<i>Voz (Minutos)</i>	Ilimitado	Ilimitado	Ilimitado
<i>Llamadas Internacionales (Min)</i>	100 min	100 min	100 min
<i>Datos (GB)</i>	135 GB	80 GB	70 GB
<i>Datos para Streaming (GB)</i>	Ilimitado	Ilimitado	Ilimitado
<i>Redes Sociales</i>	Ilimitado	Ilimitado	Ilimitado
<i>Roaming</i>	No disponible	Ilimitado	No disponible
<i>Costo (mes)</i>	\$98,56	\$55,99	\$89,90

Realizado por: CARDOSO Cristhian, 2021.

Para brindar conectividad al prototipo se le debe asignar un chip que permite la conexión a la red de telefonía móvil y a la vez este brinde la conectividad a internet, lo cual es necesario para la transmisión de la información.

Según el cálculo en Ecuación 1.4 el consumo del dispositivo será de 4,697 Mbyte por lo que se selecciona el plan básico ya que dependiendo la operadora este varía de (2,5 GB a 9 GB) el cual abastece la cantidad de datos necesarios para el sistema.

Para el caso de no disponer de un plan de datos móviles, se recomienda utilizar el servicio prepago que ofrecen las operadoras, cuya contratación debe abastecer el mínimo valor cálculo de consumo del prototipo, previamente verificando los costos de por mega consumido, debido que en prepago el costo de consumo por Mbyte es mayor a un pospago.

No se recomienda el uso de la tecnología inalámbrica wifi debido a que el vehículo al encontrarse en continuo movimiento necesita conexión constante y la disponibilidad de una red wifi en toda la ciudad no es viable.

4.3 *Guía de usuario*

4.3.1 *Especificaciones del terminal móvil.*

El dispositivo móvil debe ser alimentado por una fuente de 5V y un amperaje de mínimo 2A para que tenga un buen funcionamiento, el dispositivo tiene un tamaño de 95 x 65 x 30 milímetros. Este dispositivo no contiene batería.



Figura 38-4. Dispositivo móvil
Realizado por: CARDOSO Cristhian, 2021.

4.3.2 *Modo de Instalación*

En la **Figura 38-4** se puede observar el dispositivo móvil dentro del case y listo para su instalación en el vehículo a ser monitoreado.

- Previo a la instalación se verifica que la tarjeta SIM se encuentre insertada en el módulo GPRS.
- Conectar a la alimentación del vehículo ya sea mediante adaptadores o baterías, verificando su voltaje y amperaje máximo.
- Al momento de conectar el dispositivo, se encenderán dos leds uno de color rojo que indica que el dispositivo está conectado a la fuente de alimentación y el de color verde intermitente indica que el dispositivo está arrancando.
- Al iniciar los módulos y el adaptador Bluetooth, el módulo GPS muestra un led color rojo que indica que está conectado a la fuente de alimentación y un led verde intermitente indicando que está recibiendo los datos como se muestra en la **Figura 39-4**, el módulo GPRS con un led azul encendido permanentemente encendido indicando que indica que está conectado a la red como se muestra en la **Figura 40-4** y el adaptador Bluetooth con una luz azul que indica que el dispositivo ELM327 se encuentra conectado tal como se puede apreciar en la **Figura 41-4**.



Figura 39-4. Módulo GPS en marcha
Realizado por: CARDOSO Cristhian, 2021.



Figura 40-4. Módulo GPRS en marcha
Realizado por: CARDOSO Cristhian, 2021.



Figura 41-4. Módulo Bluetooth en marcha
Realizado por: CARDOSO Cristhian, 2021.

CONCLUSIONES

- ✓ Se determinó que el protocolo de comunicación OBD-II no se encuentra disponible en todas las marcas, además que algunos vehículos no cuentan con todos los PIDs de los sensores especificados en los modos de operación, motivo por el cual se utilizó vehículos que posean el protocolo de comunicación compatible con el ELM327, así este dispositivo es capaz de comunicarse con cualquier vehículo dentro de la lista ya especificada.
- ✓ El sistema de extracción y almacenamiento diseñado permite leer, recolectar, enviar remotamente, almacenar e interpretar los datos de la ECU, estas funciones permitirán que los usuarios tengan acceso a estos datos con mayor facilidad y diagnosticar el vehículo en cualquier parte que este se encuentre.
- ✓ Se integró diferentes tecnologías de adquisición, procesamiento y transmisión de datos disponibles para desarrollar el dispositivo móvil, el cual cumple con los requerimientos de extracción y monitoreo básico de vehículos que soportar el estándar OBD-II.
- ✓ Se diseñó una página web en el cual se visualizan los datos del monitoreo del vehículo en cuestión donde se encuentra instalado el dispositivo móvil, esta página posee una interfaz gráfica bastante amigable y de uso sencillo que incluye funciones básicas de monitoreo en tiempo real, con la particularidad de ser gratuita pudiendo acceder a la misma desde cualquier lugar.
- ✓ Se extrajo datos de posicionamiento como latitud, longitud, altura, datos de los sensores del vehículo como las revoluciones del motor, velocidad del vehículo, temperatura del aire, consumo de combustible a corto plazo y el consumo de combustible a largo plazo, adicionalmente la fecha y hora de extracción del dato, las trayectorias realizadas fueron verificadas mediante la herramienta Google Earth Pro.
- ✓ Para la extracción de datos de los sensores del automóvil en el primer escenario se tuvo un tiempo de carga de hasta 7 segundos, lo cual fue reconfigurado hasta lograr un tiempo de respuesta del ELM327 así mejorando el tiempo de carga de datos a un máximo de 2,5 segundos, lo cual lo hace mucho más eficiente en la toma de datos del vehículo, este tiempo de respuesta está directamente relacionado con la versión del lector ELM327, es decir, que independientemente del protocolo que posea el vehículo, el tiempo de respuesta será el mismo, puesto que la versión del lector ELM327 baja (v2.1).

- ✓ El cliente posee a su disposición todos los datos almacenados ya sea de fechas anteriores y en tiempo real, se encuentran ordenadas de acuerdo con la fecha y hora, con opción de descargar el reporte de los datos del monitoreo con información de: fecha hora, latitud, longitud, altitud, rpm, velocidad del vehículo, temperatura del aire, stf y ltf; toda la información almacenada se encuentra disponible en archivos de diferentes formatos como: PDF, EXCEL, CSV.

- ✓ Debido a que la cobertura de la plataforma de extracción está definida por la cobertura que ofrece la operadora móvil celular usada en el dispositivo, esta no enviara ningún tipo de información mientras no exista cobertura en el módulo.

- ✓ El tiempo mínimo de toma de los datos de cada sensor del vehículo mediante el lector OBD-II es de 100 milisegundos de petición y respuesta, un valor menor al indicado genera un error de lectura, ya que este es el rango máximo permitido por la versión del lector OBD-II v2.1, es decir con un valor menor la petición no podrá ser procesada por lo tanto el dato no podrá ser extraído mediante el lector OBD-II.

- ✓ La cantidad de datos que se consumirían mensualmente (en los procesos de lectura de las medidas realizadas) es considerablemente bajo, este depende al tiempo de extracción de los parámetros, una cantidad mayor de información por minuto generará un mayor consumo y a su vez costo, como el caso de la toma de información cada 2,5 segundos que genera 281,80 Mb de consumo de datos como máximo mensual.

- ✓ El consumo del almacenamiento en la base de datos (de la información extraída mediante el sistema) es aceptable, el consumo máximo mensual es de 32,630 Mbyte de almacenamiento con la toma de datos cada 2,5 segundos, este consumo podría disminuir considerando el tiempo de extracción de cada dato, mientras mayor sea el tiempo de extracción, menor será el consumo del almacenamiento de la base de datos, esto no generará inconvenientes ya que no existirá una sobre carga de datos gracias a los 20 Gb de almacenamiento gratuito en la base de datos que ofrece AWS.

- ✓ Existen varios trabajos en el país basados en el estándar OBD-II, así como la extracción de datos mediante otro tipo de tarjetas de desarrollo, o simplemente simulaciones con programas disponibles para todo público, a comparación con el sistema diseñado que permite la extracción y visualización de los parámetros en tiempo real, además de su fácil uso y eficiencia, con el uso de dispositivos reducidos y de bajo costo con alta fiabilidad en su funcionamiento.

RECOMENDACIONES

- ✓ Al momento de la instalación del dispositivo móvil, se debe tomar en cuenta la fuente de alimentación, en este caso usar cargadores vehiculares para celular o adaptadores que generen un amperaje de mínimo 2A.
- ✓ La ubicación del dispositivo móvil o a su vez la antena GPS debe tener vista libre sin obstáculos hacia el espacio, de esta manera recibir mejor las señales provenientes de los satélites disponibles, además con esto aumentar la precisión del posicionamiento del vehículo.
- ✓ El dispositivo móvil debe ubicarse a una distancia no mayor a 1 metro, debido que se utiliza el módulo Bluetooth el cual es necesario que se encuentre lo más cerca posible del lector ELM327 para que la comunicación sea más óptima y eficaz.
- ✓ Se debe considerar la aplicación de las fórmulas de cada PID requerido, puesto que la respuesta de la ECU a la petición del ELM327 son datos hexadecimales los cuales deben ser procesados para ser posteriormente enviados y almacenados en la base de datos.
- ✓ Se recomienda utilizar un adaptador Bluetooth compatible con el lector ELM327 o a su vez instalar los módulos necesarios dentro la tarjeta de desarrollo.
- ✓ Se debe tomar en cuenta el retardo que existe al momento de la extracción de los datos de los sensores del vehículo, ya que debido a la versión del ELM327 utilizada en el presente proyecto este tiempo posee un retraso de hasta 2,5 segundos, lo cual podría ser mejorado utilizando una versión mayor del lector ELM327 y una versión de Bluetooth mayor para que su respuesta sea más eficiente.
- ✓ Para evitar un consumo de datos de la red móvil y que este genere costos elevados, considerar el tiempo de extracción y almacenamiento de información, se recomienda realizar un menor muestreo por minuto o a su vez realizar tomas cada minuto, lo cual generará un menor consumo de datos y un menor consumo del almacenamiento de la base de datos.
- ✓ Para trabajos futuros se pueden integrar otras tecnologías y módulos más actualizados, para así mejorar el procesamiento de extracción.

- ✓ Para tener una gran escalabilidad del sistema, este deberá ser un prototipo universal, es decir que abarque todos los protocolos de comunicación vehicular y además que este sea compatible, con esto podría tener un gran alcance dentro de la industria automotriz.

BIBLIOGRAFÍA

- ACOSTA CURIPALLO, E.A.,** *Sistema Prototipo Para El Monitoreo Inalámbrico De La Cantidad De Desechos De Un Contenedor De Basura Para Ciudades Inteligentes* [en línea]. S.l.: Escuela Politécnica Nacional, Ecuador. 2020. Disponible en: https://bibdigital.epn.edu.ec/bitstream/15000/20627/1/CD_10130.pdf.
- ALVAREZ, M. y ET AL,** *Tutorial de FTP*. 2019. S.l.: Desarrollador Web. 2019.
- AMARASINGHE, M., KOTTEGODA, S., ISHANKA, A., MURAMUDALIGE, S., BANDARA, D. y AZEEZ, A.,** *Cloud-based driver monitoring and vehicle diagnostic with OBD2 telematics*. [en línea], 2015. Disponible en: https://www.researchgate.net/publication/304291123_Cloud-based_driver_monitoring_and_vehicle_diagnostic_with_OBD2_telematics.
- AWS,** Amazon Web Services - Información general acerca de los procesos de seguridad. *Amazon Web Services*, 2017.
- BELTRÁN, J.,** *Desarrollo de un simulador electrónico de una ECU y su diagnóstico sobre CAN y OBD-II* [en línea]. S.l.: Universidad de Sevilla. 2015. Disponible en: http://bibing.us.es/proyectos/abreproy/90361/fichero/TFG_JOSE_BELTRAN_ZAMBRA_NO.pdf.
- BERNIAS, G.,** *Aplicación distribuida para la monitorización y diagnosis de automóviles* [en línea]. S.l.: Universidad Politécnica de Madrid, España. 2015. Disponible en: http://oa.upm.es/38445/1/PFG_GONZALO_BERNIAS_VAQUERO.pdf.
- BLOGDEAUTOMOVILES.COM,** *Sensores en los automóviles* - BlogdeAutomóviles.com. [en línea]. [sin fecha]. Disponible en: <http://blogdeautomoviles.com/sensores-en-los-automoviles/>.
- CABALA, M. y GAMEC, J.,** *Wireless Real-Time Vehicle Monitoring Based on Android Mobile Device*. [en línea], 2012. Disponible en: https://www.researchgate.net/publication/273311789_Wireless_Real-Time_Vehicle_Monitoring_Based_on_Android_Mobile_Device.
- CAIZATO, M. y XIMENA, M.,** *Diseño e implementación de un prototipo de monitoreo de automóviles empleando el estándar OBD-II* [en línea]. S.l.: Universidad de las Fuerzas Armadas, Ecuador. 2014. Disponible en: <http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/8517/T-ESPE-047842.pdf?sequence=1&isAllowed=y>.
- CARPIO, C.,** *Universidad del Azuay Universidad del Azuay* - [en línea]. S.l.: Universidad del Azuay, Ecuador. 2013. Disponible en: <http://dspace.uazuay.edu.ec/bitstream/datos/2210/1/09672.pdf>.

- CORONEL, E.J.**, *Desarrollo de un algoritmo para la estimación del consumo de combustible en motores ciclo Otto mediante un dispositivo de adquisición de datos a través de OBD2*. S.l.: Universidad Politécnica Salesiana, Ecuador. 2018.
- CUELLO, M.M. y SARDOTH, A.K.**, *PRINCIPIO DE PROPORCIONALIDAD Y TEST DE PONDERACIÓN COMO TECNICA PARA DAR SOLUCIÓN A DERECHOS FUNDAMENTALES EN CONFLICTO EN DERECHO ADMINISTRATIVO EN EL TIEMPO POSMODERNO*. [en línea]. S.l.: s.n. 1981. Disponible en: <http://repository.usta.edu.co/bitstream/handle/11634/10756/2018Cuellomelba.pdf?sequence=1>.
- DIMATÉ CÁCERES, M.J. y CASTILLO, P.M.G.**, *Diseño de una Interfaz Gráfica en Labview para el Diagnóstico de Vehículos por medio de OBD2*. [en línea]. S.l.: Universidad Pontificia Bolivariana. 2010. Disponible en: https://repository.upb.edu.co/bitstream/handle/20.500.11912/911/digital_19592.pdf?sequence=1&isAllowed=y.
- DOLAN, D., TAUFİK, T. y MATTHEW, D.**, *Characterizing energy usage of Chevrolet Volt versus speed*. [en línea], 2013. Disponible en: https://www.researchgate.net/publication/261040409_Characterizing_energy_usage_of_Chevrolet_Volt_versus_speed.
- ELM ELECTRONICS INC.**, *ELM327 OBD to RS232 Interpreter*. [en línea], 2014. Disponible en: <https://www.elmelectronics.com/wp-content/uploads/2016/07/ELM327DS.pdf>.
- FOROUZAN, B.A.**, *Transmisión de Datos y Redes de Comunicación* [en línea]. Segunda. Madrid-España: s.n. 2007. Disponible en: https://www.academia.edu/30187265/Transmision_de_Datos_y_Redde_de_Comunicacion_es_2da_Edicion_Behrouz_A_Forouzan_FREELIBROS_ORG.
- GARCÍA, R.**, *Mecánica Automotriz (Scanner)* [en línea]. Mexico: s.n. 2004. Disponible en: http://www.ptolomeo.unam.mx:8080/jspui/bitstream/132.248.52.100/12762/1/decd_3999.pdf.
- GÓMEZ, E.**, *Diseño e implementación de un sistema de captura CAN / OBDII* [en línea]. S.l.: Universidad Politécnica de Cartagena. 2016. Disponible en: <http://repositorio.upct.es/bitstream/handle/10317/6537/tfg-dis-gom.pdf?sequence=1&isAllowed=y>.
- MALEKIAN, R., MOLOISANE, N.R., NAIR, L., MAHARAJ, B.T. y CHUDEOKONKWO, U.A.K.**, *Design and Implementation of a Wireless OBD II Fleet Management System*. S.l.: s.n. 2017.
- MANSILLA, O.R., SELLAR, J. y DE, D.E.J.**, *Diseño Y Realización De Un Sistema on Board Diagnostics (Obd-Ii)* [en línea]. S.l.: Universidad Politécnica de Cataluña, España. 2009. Disponible en: <http://bauma.upc.es/pfc/oscar/ProyectoOscar-memoria.pdf>.

- MERCADO LIBRE**, Scanner Automotriz Obd2 Elm327 Wifi, Android iPhone IOS. [en línea]. 2020. Disponible en: https://articulo.mercadolibre.cl/MLC-530507710-scanner-automotriz-obd2-elm327-wifi-android-iphone-ios-_JM.
- MESEGUER, J.E.**, *Caracterización De Los Estilos De Conducción Mediante Smartphones , Dispositivos Obd-Ii Y Redes* [en línea]. S.l.: Universidad Politécnica de Valencia, España. 2012. Disponible en: https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwjA-vSoj5DOAhUEkh4KHevHBdUQFggiMAE&url=https%3A%2F%2Friunet.upv.es%2Fbits-tream%2Fhandle%2F10251%2F21025%2FJavierMeseguerAnastasioTesinaMaster_MIC.pdf%3Fsequence%3D.
- MIFSUF, E.**, Introducción a Apache. [en línea]. S.l.: 2017. Disponible en: <http://descargas.pntic.mec.es/mentor/visitas/Apache.pdf>.
- MILLET, P.B.**, Puesta en marcha de un SGBD y un servidor web local. *FUOC. Fundación para la Universitat Oberta de Catalunya* [en línea], 2013. p. 9–10. Disponible en: https://www.academia.edu/34844922/Puesta_en_marcha_de_un_SGBD_y_un_servidor_web_local.
- MOROCHO, D.P.**, *SISTEMA ELECTRÓNICO DE ALERTA Y MONITOREO PARA EL MANTENIMIENTO DE LOS VEHICULOS DE LA CRUZ ROJA JUNTA PROVINCIAL DE NAPO* [en línea]. S.l.: UNIVERSIDAD TECNICA DE AMBATO, Ecuador. 2017. Disponible en: https://repositorio.uta.edu.ec/bitstream/123456789/26939/1/Tesis_t1342ec.pdf.
- NARANJO, J. y CHUQUITARCO, L.**, *Diseño e instalación de un Sistema De Rastreo Satelital Mediante Gps Y Gprs Para El Vehículo Chevrolet-Aveo De La Escuela De Conducción Espe-Latacunga* [en línea]. S.l.: ESPE-Latacunga. 2012. Disponible en: <http://repositorio.espe.edu.ec/handle/21000/5879>.
- OLCINA VALERO, A.**, *Desarrollo de aplicaciones web con el API de Google Cloud* [en línea]. S.l.: UNIVERSIDAD POLITÉCNICA DE VALENCIA, España. 2017. Disponible en: <https://riunet.upv.es/handle/10251/91213>.
- ORTIZ, C.**, Diseño De Escáner Automotriz Obdii Multiprotocolo. [en línea], 2014. Disponible en: https://s3.amazonaws.com/academia.edu/documents/43589213/08_0365_EO.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1531982135&Signature=4Zedm4yKCYYYJhXDRwYn3d3M%2FARM%3D&response-content-disposition=inline%3Bfilename%3DDISENO_DE_ESCANER_AUTOMOTRIZ_OBDII_.
- SABERES Y CIENCIAS**, La importancia de la electrónica en el desarrollo del automóvil. [en línea]. abril 2017. 2017. Disponible en: <https://saberesciencias.com.mx/2017/06/04/la->

importancia-de-la-electronica-en-el-desarrollo-del-automovil/.

- SALAS, R.,** EVOLUCION DE LOS PROTOCOLOS DE DIAGNOSTICO OBD 2 y OBD3 DIAGNOSTICO A BORDO – (parte2). [en línea]. 2018. Disponible en: <https://mecanicaparatodosblog.wordpress.com/2018/10/02/historia-de-los-protocolos-de-diagnostico-parte2-obd-2-diagnostico-a-bordo/>.
- SÁNCHEZ, J.,** *Simulador de una ECU y diagnóstico mediante CAN y OBD-II* [en línea]. S.l.: Escuela Politécnica CUENCA. 2017. Disponible en: <https://docplayer.es/72078651-Resumen-1-abstract-3-parte-i-memoria-5-introduccion-6-fundamentos-teoricos-obdii-7.html>.
- SIMBAÑA, W., CAIZA, J., CHÁVEZ, D. y LOPEZ, G.,** Diseño e Implementación de un Sistema de Monitoreo Remoto del Motor de un Vehículo basado en Obd-II y la plataforma Arduino. *Revista Politécnica*, 2016.
- SIMBAÑA, W.A.,** *Diseño e Implementación de una Solución Telemática Basada en OBD-II (ON-BOARD DIAGNOSTIC) que Permita Obtener y Procesar la Información de los Sensores del Motor de una Automóvil* [en línea]. S.l.: Escuela Politécnica Nacional. 2015. Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/10659/1/CD-6287.pdf>.
- SISLEMA, O.E.,** *DISEÑO DE UN SISTEMA DE RASTREO BÁSICO EN TIEMPO REAL CON TARJETAS DE DESARROLLO DE BAJO COSTO PARA OBJETOS RASTREABLES* [en línea]. S.l.: ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO, Ecuador. 2018. Disponible en: <http://dspace.espace.edu.ec/bitstream/123456789/9265/5/98T00221.pdf>.
- VELASCO, N.,** *Introducción al estándar GSM* [en línea]. S.l.: Universidad de Sevilla. 2005. Disponible en: <https://bibing.us.es/proyectos/abreproy/11141/fichero/PFC%252F4+Red+GSM.pdf>.
- VELÁZQUEZ, Á.M.,** *Android OBD-II* [en línea]. S.l.: Universidad de Valladolid. 2013. Disponible en: <http://uvadoc.uva.es/bitstream/handle/10324/4084/TFG-B.359.pdf?sequence=1&isAllowed=y>.
- VILLÉN MACÍAS, J.P.,** *Simulador de la ECU de un vehículo con protocolo ISO 9141-2* [en línea]. S.l.: Universidad de Sevilla, Sevilla. 2016. Disponible en: [https://idus.us.es/bitstream/handle/11441/48672/Simulador de la ECU de un vehículo sobre OBDII con protocolo ISO 9141-2.pdf?sequence=1&isAllowed=y](https://idus.us.es/bitstream/handle/11441/48672/Simulador%20de%20la%20ECU%20de%20un%20veh%C3%ADculo%20sobre%20OBDII%20con%20protocolo%20ISO%209141-2.pdf?sequence=1&isAllowed=y).
- XIE, Y., SU, X., HE, Y., CHEN, X., CAI, G., XU, B. y YE, W.,** STM32-based vehicle data acquisition system for Internet-of-Vehicles. [en línea], 2017. Disponible en: https://www.researchgate.net/publication/318035668_STM32-based_vehicle_data_acquisition_system_for_Internet-of-Vehicles.

ANEXOS

ANEXO A:

PIDs de los modos de operación de OBD-II

Modo 01

PIDs (hex)	PID (Dec)	Data bytes returned	Description	Min value	Max value	Units	Formula
00	0	4	PIDs supported [01 - 20]				Bit encoded [A7..D0] == [PID \$01..PID \$20]
01	1	4	Monitor status since DTCs cleared. (Includes malfunction indicator lamp (MIL) status and number of DTCs.)				Bit encoded. See below
02	2	2	Freeze DTC				
03	3	2	Fuel system status				Bit encoded. See below
04	4	1	Calculated engine load	0	100	%	$\frac{100}{255} A$ (or $\frac{100}{2.55}$)
05	5	1	Engine coolant temperature	-40	215	°C	A-40
06	6	1	Short term fuel trim— Bank 1	-100 (Reduce Fuel: Too Rich)	99.2 (Add Fuel: Too Lean)	%	$\frac{100}{128} A - 100$ Or $(\frac{A}{1.28} - 100)$
07	7	1	Long term fuel trim— Bank 1	-100 (Reduce Fuel: Too Rich)	99.2 (Add Fuel: Too Lean)	%	$\frac{100}{128} A - 100$ Or $(\frac{A}{1.28} - 100)$
08	8	1	Long term fuel trim— Bank 1	-100 (Reduce Fuel: Too Rich)	99.2 (Add Fuel: Too Lean)	%	$\frac{100}{128} A - 100$ Or $(\frac{A}{1.28} - 100)$
09	9	1	Long term fuel trim— Bank 2	-100 (Reduce Fuel: Too Rich)	99.2 (Add Fuel: Too Lean)	%	$\frac{100}{128} A - 100$ Or $(\frac{A}{1.28} - 100)$

Continuación Modo 01

0A	10	1	Fuel pressure (gauge pressure)	0	765	kPa	3A
0B	11	1	Intake manifold absolute pressure	0	255	kPa	A
0C	12	1	Engine speed	0	16,383.75	rpm	$\frac{256A + B}{4}$
0D	13	1	Vehicle speed	0	255	km/h	A
0E	14	1	Timing advance	-64	63.5	° before TDC	$\frac{A}{2} - 64$
0F	15	1	Intake air temperature	-40	215	°C	A-40
10	16	2	Mass air flow sensor (MAF) air flow rate	0	655.35	grams/s	$\frac{256A + B}{100}$
11	17	1	Throttle position	0	100	%	$\frac{100}{255} A$
12	18	1	Commanded secondary air status				Bit encoded. See below
13	19	1	Oxygen sensors present (in 2 banks)				A0..A3] == Bank 1, Sensors 1-4. [A4..A7] == Bank 2...

Continuación Modo 01

14	20	2	Oxygen Sensor 1 A: Voltage B: Short term fuel trim	0 -100	1.275 99.2	volts %	$\frac{A}{200}$ $\frac{100}{128}B-100$ (if B==\$FF, sensor is not used in trim calculation)
15	21	2	Oxygen Sensor 2 A: Voltage B: Short term fuel trim				
16	22	2	Oxygen Sensor 3 A: Voltage B: Short term fuel trim				
17	23	2	Oxygen Sensor 4 A: Voltage B: Short term fuel trim				
18	24	2	Oxygen Sensor 5 A: Voltage B: Short term fuel trim				
19	25	2	Oxygen Sensor 6 A: Voltage B: Short term fuel trim				
1A	26	2	Oxygen Sensor 7 A: Voltage B: Short term fuel trim				
1B	27	2	Oxygen Sensor 8 A: Voltage B: Short term fuel trim				
1C	28	1	OBD standards this vehicle conforms to	1	250	-	<u>enumerated. See below</u>
1D	29	1	Oxygen sensors present (in 4 banks)				Similar to PID 13, but [A0..A7] == [B1S1, B1S2, B2S1, B2S2, B3S1, B3S2, B4S1, B4S2]
1E	30	1	Auxiliary input status				A0 == <u>Power Take Off</u> (PTO) status (1 ==

Continuación Modo 01

1F	31	2	Run time since engine start	0	65,535	seconds	$256A + B$
20	32	4	PIDs supported [21 - 40]				Bit encoded [A7..D0] == [PID \$21..PID \$40] See below
21	33	2	Distance traveled with malfunction indicator lamp (MIL) on	0	65,535	km	$256A + B$
22	34	2	Fuel Rail Pressure (relative to manifold vacuum)	0	5177.265	kPa	$0.079(256A + B)$
23	35	2	Fuel Rail Gauge Pressure (diesel, or gasoline direct injection)	0	655,350	kPa	$10(256A+B)$
24	36	4	Oxygen Sensor 1 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Voltage				
25	37	4	Oxygen Sensor 2 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Voltage				$\frac{2}{65536}(256A+B)$
26	38	4	Oxygen Sensor 3 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Voltage	0 0	< 2 < 8	ratio V	$\frac{8}{65536}(256C+D)$
27	39	4	Oxygen Sensor 4 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Voltage				
28	40	4	Oxygen Sensor 5 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Voltage				
29	41	4	Oxygen Sensor 6 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Voltage				$\frac{2}{65536}(256A+B)$
2A	42	4	Oxygen Sensor 7 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Voltage	0 0	< 2 < 8	ratio V	$\frac{8}{65536}(256C+D)$

Continuación Modo 01

2B	43	4	Oxygen Sensor 8 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Voltage				
2C	44	1	Commanded EGR	0	100	%	$\frac{100}{255}A$
2D	45	1	EGR Error	-100	99.2	%	$\frac{100}{128}A - 100$
2E	46	1	Commanded evaporative purge	0	100	%	$\frac{100}{255}A$
2F	47	1	Fuel Tank Level Input	0	100	%	$\frac{100}{255}A$
30	48	1	Warm-ups since codes cleared	0	255	count	A
31	49	2	Distance traveled since codes cleared	0	65,535	km	256A + B
32	50	2	Evap. System Vapor Pressure	- 8,192	8191.75	Pa	$\frac{256A + B}{4}$ (AB is two's complement signed)
33	51	1	Absolute Barometric Pressure	0	255	kPa	A
34	52	4	Oxygen Sensor 1 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Current				
35	53	4	Oxygen Sensor 2 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Current				
36	54	4	Oxygen Sensor 3 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Current				
37	55	4	Oxygen Sensor 4 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Current				
38	56	4	Oxygen Sensor 5 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Current	0 -128	2 <128	ratio mA	$\frac{2}{65536}(256A+B)$ $\frac{256C+D}{256}-128$ Or $C + \frac{D}{256} - 128$
39	57	4	Oxygen Sensor 6 AB: Air-Fuel Equivalence Ratio				

Continuación Modo 01

3A	58	4	Oxygen Sensor 7 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Current				
3B	59	4	Oxygen Sensor 8 AB: Air-Fuel Equivalence Ratio (lambda,λ) CD: Current				
3C	60	2	Catalyst Temperature: Bank 1, Sensor 1				
3D	61	2	Catalyst Temperature: Bank 2, Sensor 1	-40	6,513.5	°C	$\frac{256A+B}{10}-40$
3E	62	2	Catalyst Temperature: Bank 1, Sensor 2				
3F	63	2	Catalyst Temperature: Bank 2, Sensor 2	-40	6,513.5	°C	$\frac{256A+B}{10}-40$
40	64	4	PIDs supported [41 - 60]				Bit encoded [A7..D0] == [PID \$41..PID \$60] See below
41	65	4	Monitor status this drive cycle				Bit encoded. See below
42	66	2	Control module voltage	0	65.535	V	$\frac{256A + B}{1000}$
43	67	2	Absolute load value	0	25,700	%	$\frac{100}{255}(256A + B)$
44	68	2	Commanded Air-Fuel Equivalence Ratio (lambda,λ)	0	< 2	ratio	$\frac{2}{65536}(256A + B)$
45	69	1	Relative throttle position	0	100	%	$\frac{100}{255}A$
46	70	1	Ambient air temperature	-40	215	°C	A - 40
47	71	1	Absolute throttle position B				

Continuación Modo 01

48	72	1	Absolute throttle position C				
49	73	1	Accelerator pedal position D				
4A	74	1	Accelerator pedal position E				
4B	75	1	Accelerator pedal position F	0	100	%	$\frac{100}{255}A$
4C	76	1	Commanded throttle actuator				
4D	77	2	Time run with MIL on				
4E	78	2	Time since trouble codes cleared	0	65,535	minutes	256A + B
4F	79	4	Maximum value for Fuel–Air equivalence ratio, oxygen sensor voltage, oxygen sensor current, and intake manifold absolute pressure	0,0,0,0	255, 255, 255, 2550	ratio, V, mA, kPa	A, B, C, D * 10
50	80	4	Maximum value for air flow rate from mass air flow sensor	0	2550	g/s	A*10, B, C, and D are reserved for future use
51	81	1	Fuel Type				From fuel type table see below
52	82	1	Ethanol fuel %	0	100	%	$\frac{100}{255}A$
53	83	2	Absolute Evap system Vapor Pressure	0	327.675	kPa	$\frac{256A + B}{200}$
54	84	2	Evap system vapor pressure	- 32,767	32,768	Pa	$((A * 256) + B) - 32767$
55	85	2	Short term secondary oxygen sensor trim, A: bank 1, B: bank 3				
56	86	2	Long term secondary oxygen sensor				

Continuación Modo 01

57	87	2	Short term secondary oxygen sensor trim, A: bank 2, B: bank 4				
58	88	2	Long term secondary oxygen sensor trim, A: bank 2, B: bank 4				
59	89	2	<u>Fuel rail</u> absolute pressure	0	655,350	kPa	10(256A + B)
5A	90	1	Relative throttle pedal position	0	100	%	$\frac{100}{255}A$
5B	91	1	Hybrid battery pack remaining life	0	100	%	$\frac{100}{255}A$
5C	92	1	Engine oil temperature	-40	210	°C	A - 40
5D	93	2	Fuel injection timing	-210.00	301.992	°	$\frac{256A + B}{128} - 210$
5E	94	2	Engine fuel rate	0	3212.75	L/h	$\frac{256A + B}{20}$
5F	95	1	Emission requirements to which vehicle is designed				Bit Encoded
60	96	4	PIDs supported [61 - 80]				Bit encoded [A7..D0] == [PID \$61..PID \$80] See below
61	97	1	Driver's demand engine - percent torque	-125	130	%	A - 125
62	98	1	Actual engine - percent torque	-125	130	%	A - 125
63	99	2	Engine reference torque	0	65,535	Nm	256A + B
64	100	5	Engine percent torque data	-125	130	%	A-125 Idle B-125 Engine point 1 C-125 Engine point 2 D-125 Engine point 3 E-125 Engine point 4

Continuación Modo 01

65	101	2	Auxiliary input / output supported				Bit Encoded
66	102	5	Mass air flow sensor				
67	103	3	Engine coolant temperature			°C	
68	104	7	Intake air surface sensor				
69	105	7	Commanded EGR and EGR Error				
6A	106	5	Commanded Diesel intake air flow control and relative intake air flow position				
6B	107	5	Exhaust gas recirculation temperature				
6C	108	5	Commanded throttle actuator control and relative throttle position				
6D	109	6	Fuel pressure control system				
6E	110	5	Injection pressure control system				
6F	111	3	Turbocharger compressor inlet pressure				
70	112	9	Boost pressure control				
71	113	5	Variable Geometry turbo (VGT) control				
72	114	5	Wastegate control				
73	115	5	Exhaust pressure				
74	116	5	Turbocharger RPM				
75	117	7	Turbocharger temperature				
76	118	7	Turbocharger temperature				
77	119	5	Charge air cooler				

Continuación Modo 01

78	120	9	Exhaust Gas surface (EGT) Bank 1				Special PID. See below
79	121	9	Exhaust Gas surface (EGT) Bank 2				Special PID. See below
7 ^a	122	7	Diesel particulate filter (DPF)				
7B	123	7	Diesel particulate filter (DPF)				
7C	124	9	Diesel Particulate filter (DPF) temperature			°C	$\frac{256A + B}{10} - 40$
7D	125	1	Nox NTE (Not- To-Exceed) control surf status				
7E	126	1	PM NTE (Not- To-Exceed) control surf status				
7F	127	13	Engine run time			seconds	
80	128	4	PIDs supported [81 – A0]				Bit encoded [A7..D0] == [PID \$81..PID \$A0] See below
81	129	21	Engine run time for Auxiliary Emissions Control Device(AECD)				
82	130	21	Engine run time for Auxiliary Emissions Control Device(AECD)				
83	131	5	Nox sensor				
84	132	1	Manifold surface temperature				
85	133	10	Nox reagent system				
86	134	5	Particulate matter (PM) sensor				

Continuación Modo 01

87	135	5	Intake manifold absolute pressure				
88	136	13	SCR Induce System				
89	137	41	Run Time for AECD #11-#15				
8 ^a	138	41	Run Time for AECD #16-#20				
8B	139	7	Diesel Aftertreatment				
8C	140	16	O2 Sensor (Wide Range)				
8D	141	1	Throttle Position G	0	100	%	
8E	142	1	Engine Friction – Percent Torque	-125	130	%	A – 125
8F	143	5	PM Sensor Bank 1 & 2				
90	144	3	WWH-OBD Vehicle OBD System Information			hours	
91	145	5	WWH-OBD Vehicle OBD System Information			hours	
92	146	2	Fuel System Control				
93	147	3	WWH-OBD Vehicle OBD Counters support			hours	
94	148	12	Nox Warning And Inducement System				
95	152	9	Exhaust Gas Temperature Sensor				
96	153	9	Exhaust Gas Temperature Sensor				
9 ^a	154	6	Hybrid/EV Vehicle System Data, Battery, Voltage				
9B	155	4	Diesel Exhaust Fluid Sensor Data				
9C	156	17	O2 Sensor Data				

Continuación Modo 01

9D	157	4	Engine Fuel Rate			g/s	
9E	158	2	Engine Exhaust Flow Rate			kg/h	
9F	159	9	Fuel System Percentage Use				
A0	160	4	PIDs supported [A1 – C0]				Bit encoded [A7..D0] == [PID \$A1..PID \$C0] See below
A1	161		Nox Sensor Corrected Data			ppm	
A2	162	2	Cylinder Fuel Rate			mg/stroke	
A3	163	9	Evap System Vapor Pressure			Pa	
A4	164	4	Transmission Actual Gear				
A5	165	4	Diesel Exhaust Fluid Dosing				
A6	166	4	Odometer	0	429,496,729.5		$\frac{A(2^{24}) + B(2^{16}) + C(2^8) + D}{10}$
C0	192	4	PIDs supported [C1 – E0]	0x0	0xffffffff		Bit encoded [A7..D0] == [PID \$C1..PID \$E0] See below
C3	195	i	i	i	i	i	Returns numerous data, including Drive Condition ID and Engine Speed*
C4	196	i	i	i	?	?	B5 is Engine Idle Request B6 is Engine Stop Request*

Modo 02

PID (hex)	Data bytes returned	Description	Min value	Max value	Units	Formula
2	2	DTC that caused freeze frame to be stored.				BCD encoded. Decoded as in service 3

Modo 03

PID (hex)	Data bytes returned	Description	Min value	Max value	Units	Formula
N/A	n*6	Request trouble codes				3 codes per message frame. See below

Modo 04

PID (hex)	Data bytes returned	Description	Min value	Max value	Units	Formula
N/A	0	Clear trouble codes / Malfunction indicator lamp (MIL) / Check engine light				Clears all stored trouble codes and turns the MIL off

Modo 05

PID (hex)	Data bytes returned	Description	Min value	Max value	Units	Formula
0100	4	OBD Monitor IDs supported (\$01 – \$20)	0x0	0xffffffff		
0101	2	O2 Sensor Monitor Bank 1 Sensor 1	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0102		O2 Sensor Monitor Bank 1 Sensor 2	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0103		O2 Sensor Monitor Bank 1 Sensor 3	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0104		O2 Sensor Monitor Bank 1 Sensor 4	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0105		O2 Sensor Monitor Bank 2 Sensor 1	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0106		O2 Sensor Monitor Bank 2 Sensor 2	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage

Continuación Modo 05

0107		O2 Sensor Monitor Bank 2 Sensor 3	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0108		O2 Sensor Monitor Bank 2 Sensor 4	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0109		O2 Sensor Monitor Bank 3 Sensor 1	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
010A		O2 Sensor Monitor Bank 3 Sensor 2	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
010B		O2 Sensor Monitor Bank 3 Sensor 3	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
010C		O2 Sensor Monitor Bank 3 Sensor 4	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
010D		O2 Sensor Monitor Bank 4 Sensor 1	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
010E		O2 Sensor Monitor Bank 4 Sensor 2	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
010F		O2 Sensor Monitor Bank 4 Sensor 3	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0110		O2 Sensor Monitor Bank 4 Sensor 4	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0201		O2 Sensor Monitor Bank 1 Sensor 1	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0202		O2 Sensor Monitor Bank 1 Sensor 2	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0203		O2 Sensor Monitor Bank 1 Sensor 3	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0204		O2 Sensor Monitor Bank 1 Sensor 4	0.00	1.275	Volts	0.005 Rich to lean sensor threshold voltage
0205		O2 Sensor Monitor Bank 2 Sensor 1	0.00	1.275	Volts	0.005 Rich to lean sensor threshold voltaje
0206		O2 Sensor Monitor Bank 2 Sensor 2	0.00	1.275	Volts	0.005 Rich to lean sensor threshold voltaje
0207		O2 Sensor Monitor Bank 2 Sensor 2	0.00	1.275	Volts	0.005 Rich to lean sensor threshold voltaje

Continuación Modo 05

0208		O2 Sensor Monitor Bank 2 Sensor 4	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
0209		O2 Sensor Monitor Bank 3 Sensor 1	0.00	1.275	Volts	0.005 Rich to lean sensor threshold voltage
020A		O2 Sensor Monitor Bank 3 Sensor 2	0.00	1.275	Volts	0.005 Rich to lean sensor threshold voltage
020B		O2 Sensor Monitor Bank 3 Sensor 3	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage
020C		O2 Sensor Monitor Bank 3 Sensor 4	0.00	1.275	Volts	0.005 Rich to lean sensor threshold voltage
020D		O2 Sensor Monitor Bank 4 Sensor 1	0.00	1.275	Volts	0.005 Rich to lean sensor threshold voltage
020E		O2 Sensor Monitor Bank 4 Sensor 2	0.00	1.275	Volts	0.005 Rich to lean sensor threshold voltage
020F		O2 Sensor Monitor Bank 4 Sensor 3	0.00	1.275	Volts	0.005 Rich to lean sensor threshold voltage
0210		O2 Sensor Monitor Bank 4 Sensor 4	0.00	1.275	volts	0.005 Rich to lean sensor threshold voltage

Modo 09

PID (hex)	Data bytes returned	Description	Min value	Max value	Units	Formula
00	4	Service 9 supported PIDs (01 to 20)				Bit encoded. [A7..D0] = [PID \$01..PID \$20] See below
01	1	VIN Message Count in PID 02. Only for ISO 9141-2, ISO 14230-4 and SAE J1850.				Usually the value will be 5.
02	17	Vehicle Identification Number (VIN)				17-char VIN, ASCII-encoded and left-padded with null chars (0x00) if needed to.

Continuación Modo 09

03	1	Calibration ID message count for PID 04. Only for ISO 9141-2, ISO 14230-4 and SAE J1850.				It will be a multiple of 4 (4 messages are needed for each ID).
04	16,32,48,64..	Calibration ID				Up to 16 ASCII chars. Data bytes not used will be reported as null bytes (0x00). Several CALID can be outputted (16 bytes each)
05	1	Calibration verification numbers (CVN) message count for PID 06. Only for ISO 9141-2, ISO 14230-4 and SAE J1850.				
06	4,8,12,16	Calibration Verification Numbers (CVN) Several CVN can be output (4 bytes each) the number of CVN and CALID must match				Raw data left-padded with null characters (0x00). Usually displayed as hex string.
07	1	In-use performance tracking message count for PID 08 and 0B. Only for ISO 9141-2, ISO 14230-4 and SAE J1850.	8	10		8 if sixteen (16) values are required to be reported, 9 if eighteen (18) values are required to be reported, and 10 if twenty (20) values are required to be reported (one message reports two values, each one consisting in two bytes).
08	4	In-use performance tracking for spark ignition vehicles				4 or 5 messages, each one containing 4 bytes (two values). See below
09	1	ECU name message count for PID 0A				

ANEXO B:

Comandos AT de OBD-II



ELM327

AT Command Summary (continued)

OBD Commands

AL	Allow Long (>7 byte) messages
AMC	display Activity Monitor Count
AMT hh	set the Activity Mon Timeout to hh
AR	Automatically Receive
AT0, 1, 2	Adaptive Timing off, auto1*, auto2
BD	perform a Buffer Dump
BI	Bypass the Initialization sequence
DP	Describe the current Protocol
DPN	Describe the Protocol by Number
FT	Filter for Transmitter off*
FT hh	Filter for Transmitter = hh
H0, H1	Headers off*, or on
IA	Is the protocol Active?
MA	Monitor All
MR hh	Monitor for Receiver = hh
MT hh	Monitor for Transmitter = hh
NL	Normal Length messages*
PC	Protocol Close
R0, R1	Responses off, or on*
RA hh	set the Receive Address to hh
S0, S1	printing of Spaces off, or on*
SH xyz	Set Header to xyz
SH xxyyzz	Set Header to xxyyzz
SH wwxyyzz	Set Header to wwxyyzz
SP h	Set Protocol to h and save it
SP Ah	Set Protocol to Auto, h and save it
SP 00	Erase stored protocol
SR hh	Set the Receive address to hh
SS	use Standard Search order (J1978)
ST hh	Set Timeout to hh x 4 msec
TA hh	set Tester Address to hh
TP h	Try Protocol h
TP Ah	Try Protocol h with Auto search

J1850 Specific Commands (protocols 1 and 2)

IFR0, 1, 2	IFRs off, auto*, or on, if not monitoring
IFR4, 5, 6	IFRs off, auto, or on, at all times
IFRH, S	IFR value from Header* or Source

ISO Specific Commands (protocols 3 to 5)

FI	perform a Fast Initiation
IB10	set the ISO Baud rate to 10400*
IB12	set the ISO Baud rate to 12500
IB15	set the ISO Baud rate to 15625
IB48	set the ISO Baud rate to 4800
IB96	set the ISO Baud rate to 9600
IIA hh	set ISO (slow) Init Address to hh
KW	display the Key Words
KW0, KW1	Key Word checking off, or on*
SI	perform a Slow (5 baud) Initiation
SW hh	Set Wakeup interval to hh x 20 msec
SW 00	Stop sending Wakeup messages
WM [1 - 6 bytes]	set the Wakeup Message

CAN Specific Commands (protocols 6 to C)

C0, C1	send Confirmation off, or on*
CAF0, CAF1	Automatic Formatting off, or on*
CEA	CAN Extended Addressing off
CEA hh	use CAN Extended Address hh
CER hh	set CAN Extended Rx address to hh
CF hhh	set the ID Filter to hhh
CF hhhhhhhh	set the ID Filter to hhhhhhhh
CFC0, CFC1	Flow Controls off, or on*
CM hhh	set the ID Mask to hhh
CM hhhhhhhh	set the ID Mask to hhhhhhhh
CP hh	set CAN Priority to hh (29 bit)
CRA	reset the Receive Address filters
CRA hhh	set CAN Receive Address to hhh

**AT Command Summary (continued)****CAN Specific Commands (continued)**

CRA hhhhhhhh	set the Rx Address to hhhhhhhh
CS	show CAN Status counts & freq
CSM0, CSM1	Silent Monitoring off, or on*
CTM1	set Timer Multiplier to 1*
CTM5	set Timer Multiplier to 5
D0, D1	display of the DLC off*, or on
FC SM h	Flow Control, Set the Mode to h
FC SH hhh	FC, Set the Header to hhh
FC SH hhhhhhhh	Set the Header to hhhhhhhh
FC SD [1 - 5 bytes]	FC, Set Data to [...]
PB xx yy	Protocol B options and baud rate
RTR	send an RTR message
V0, V1	use of Variable DLC off*, or on

J1939 CAN Specific Commands (protocols A to C)

DM1	monitor for DM1 messages
JE	use J1939 Elm data format*
JHF0, JHF1	Header Formatting off, or on*
JS	use J1939 SAE data format
JTM1	set Timer Multiplier to 1*
JTM5	set Timer Multiplier to 5
MP hhhh	Monitor for PGN 0hhhh
MP hhhh n	" " and get n messages
MP hhhhhh	Monitor for PGN hhhhhh
MP hhhhhh n	" " and get n messages

* Settings which are shown with an asterisk (*) are the default values.

ANEXO C:

Hoja de especificaciones técnica tarjeta Raspberry Pi 3 model B

Raspberry Pi 3 Model B+

2

Specifications

Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	<ul style="list-style-type: none">■ 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)■ 4 × USB 2.0 ports
Access:	Extended 40-pin GPIO header
Video & sound:	<ul style="list-style-type: none">■ 1 × full size HDMI■ MIPI DSI display port■ MIPI CSI camera port■ 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	<ul style="list-style-type: none">■ 5V/2.5A DC via micro USB connector■ 5V DC via GPIO header■ Power over Ethernet (PoE)-enabled (requires separate PoE HAT)
Environment:	Operating temperature, 0–50°C
Compliance:	For a full list of local and regional product approvals, please visit www.raspberrypi.org/products/raspberry-pi-3-model-b+
Production lifetime:	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.



ANEXO D:

Hoja de especificaciones técnicas módulo GPS UBLOX



NEO-M8 - Data sheet

1.3 Performance

Parameter	Specification					
Receiver type	72-channel u-blox M8 engine GPS L1C/A, SBAS L1C/A, QZSS L1C/A, QZSS L1 SAIF, GLONASS L1OF, BeiDou B1I, Galileo E1B/C					
Accuracy of time	RMS	30 ns				
pulse signal	99%	60 ns				
Frequency of time pulse signal	0.25 Hz...10 MHz (configurable)					
Operational limits ¹	Dynamics	≤ 4 g				
	Altitude	50,000 m				
	Velocity	500 m/s				
Velocity accuracy ²	0.05 m/s					
Heading accuracy ²	0.3 degrees					
GNSS	GPS & GLONASS	GPS	GLONASS	BeiDou	Galileo	
Horizontal position accuracy ³	2.5 m	2.5 m	4 m	3 m	TBC ⁴	
NEO-M8N/Q						
Max navigation update rate	NEO-M8N	5 Hz	10 Hz	10 Hz	10 Hz	10 Hz
	NEO-M8Q	10 Hz	18 Hz	18 Hz	18 Hz	18 Hz
Time-To-First-Fix ⁵	Cold start	26 s	29 s	30 s	34 s	45 s
	Hot start	1 s	1 s	1 s	1 s	1 s
	Aided starts ⁶	2 s	2 s	2 s	3 s	7 s
Sensitivity ⁷	Tracking & Navigation	-167 dBm	-166 dBm	-166 dBm	-160 dBm	-159 dBm
	Reacquisition	-160 dBm	-160 dBm	-156 dBm	-157 dBm	-153 dBm
	Cold start	-148 dBm	-148 dBm	-145 dBm	-143 dBm	-138 dBm
	Hot start	-157 dBm	-157 dBm	-156 dBm	-155 dBm	-151 dBm
NEO-M8M						
Max navigation update rate	10 Hz	18 Hz	18 Hz	18 Hz	18 Hz	
Time-To-First-Fix ⁵	Cold start	26 s	30 s	33 s	39 s	57 s
	Hot start	1 s	1 s	1 s	1 s	1 s
	Aided starts ⁶	3 s	3 s	3 s	7 s	7 s
Sensitivity ⁷	Tracking & Navigation	-164 dBm	-164 dBm	-163 dBm	-160 dBm	-154 dBm
	Reacquisition	-160 dBm	-159 dBm	-156 dBm	-155 dBm	-152 dBm
	Cold start	-148 dBm	-147 dBm	-145 dBm	-143 dBm	-133 dBm
	Hot start	-157 dBm	-156 dBm	-155 dBm	-155 dBm	-151 dBm

Table 1: NEO-M8 performance in different GNSS modes (default: concurrent reception of GPS and GLONASS including QZSS, SBAS)

ANEXO E:

Configuración en PHP de las páginas web de la plataforma de extracción de datos.

conexion.php

```
<?php
//Mantener la sesion activa
if(!isset($_SESSION))
    session_start();
//CONEXIÓN A LA
BASE DE DATOS
$hostname_db="localhost";
$database_db="db_obd2";
$username_db = "root";
$password_db = "Tobd2";
$conexion =
mysqli_connect($hostname_db,
                $username_db,
                $password_db);
mysqli_select_db($conexion,$database_db) or die
("Ninguna DB
seleccionada");
?>
```

login.php

```
<?php
session_start();
if(isset($_SESSION['usuario'])) {
    header('location:
index.php');
}
$error = "";
if($_SERVER['REQUEST_METHOD'] ==
'POST'){
    $usuario = $_POST['usuario'];
    $clave = $_POST['clave'];
    $statement =
    mysqli_prepare($conexion,
    "SELECT * FROM
    login WHERE usuario =
    :usuario AND clave =
    :clave");
    mysqli_stmt_bind_param($statement,
    "ss");
    mysqli_stmt_bind_value($statement,
    $usuario);
    mysqli_stmt_bind_value($statement,
    $clave);
    mysqli_stmt_execute($statement);
    $resultado =
    mysqli_stmt_fetch($statement);
    if ($resultado !=
    false){

```

```

$_SESSION['usuario'] =           }else{           }
$usuario;
header('location:           $error .= '<i>Este           require 'frontend/login-
principal.php');           usuario no existe</i>';           vista.php';
?>

```

principal.php

```

<?php           body{           <center><h2>Sistema
require_once('conexion.p           background-           OBD2</h2></center>
hp');           image:           <?php
$conexion=conexion();           url(image/bg2.jpeg);           $sql="SELECT *
?>           }           from login";
<!DOCTYPE html>           .welcome a:hover{           $result=mysqli_query($co
<html lang="en">           color: black;           nexion,$sql);
<head>           background: white;           while($ver=mysqli_fetch_
<meta charset="UTF-           }           row($result)){
8">           </style>           $datos=
<title>Sesion           </head>           $ver[0]."|" .
iniciada</title>           <body>           $ver[1]."|" .
<meta           <div class="container-           $ver[2];
name="viewport"           form">           ?>
content="width=device-           <div class="header">           <div>
width, user-scalable=no,           <div class="logo-           <h2><?php
initial-scale=1.0,           title">           echo $ver[2] ?></h2>
maximum-scale=1.0,           <?php           </div>
minimum-scale=1.0">           }           <?php
<link rel="stylesheet"           <img           ?>
href="css/style.css">           src="image/logo_magtimu           </div>
<style>           s.png" alt="">

```

```

    <div
class="menu">
    <a
href="cerrar.php"><li
class="module-register
active">Cerrar
sesión</li></a>
</div>

```

```

</div>
<div
class="welcome">
    <h1>Bienvenido,
has logrado iniciar sesion
en el Sistema OBD2</h1>

```

```

    <a
href="tabla.php">Continu
ar</a>
</div>
</div>
</body>
</html>

```

tabla.php

```

<?php
require_once
"conexion.php";
$conexion=conexion();
?>
<!doctype html>
<html lang="en">
    <head>
        <!-- Required meta tags
-->
        <meta charset="utf-8">
        <meta
name="viewport"
content="width=device-
width, user-scalable=no,
initial-scale=1.0,
maximum-scale=1.0,
minimum-scale=1.0">

```

```

<link rel="shortcut
icon" href="#" />
<title>Datos
Registrados</title>
<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="bootstrap/css/bootst
rap.min.css">
<!-- CSS personalizado
-->
<link rel="stylesheet"
href="main.css">
<link rel="stylesheet"
href="css/style.css">
<!--datables CSS
básico-->
<link rel="stylesheet"
type="text/css"
href="datatables/datatable
s.min.css"/>

```

```

<!--datables estilo
bootstrap 4 CSS-->
<link rel="stylesheet"
type="text/css"
href="datatables/DataTabl
es-
1.10.18/css/dataTables.bo
otstrap4.min.css">
<link rel="stylesheet"
type="text/css"
href="librerias/alertifyjs/c
ss/alertify.css">
<link rel="stylesheet"
type="text/css"
href="librerias/alertifyjs/c
ss/themes/default.css">
<link rel="stylesheet"
type="text/css"
href="librerias/datatables/d
ataTables.bootstrap.min.cs
s">

```

```

<script src="js/funciones.js"></script>
<!--font awesome con CDN-->
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.2/css/all.css" integrity="sha384-oS3vJWv+0UjzBfQzYUh tDYW+Pj2yciDJxpsK1O YPAYjqT085Qq/1cq5FL XAZQ7Ay" crossorigin="anonymous" >
<style>
body{
background-image: url(image/bg11.jpg);
}
table th {
background-color: #6f4988 !important;
color: white;
}
table>tbody>tr>td {
color: white;
}
}
table>tbody>tr>td {
vertical-align: middle !important;
}
/*para alinear los botones y cuadro de busqueda*/
.btn-group, .btn-group-vertical {
position: absolute !important;
}
</style>
</head>
<body>
<header>
<h1 class="text-center text-light"></h1>
<h2 class="text-center text-light"> <span class="badge badge-info">LECTURA DE DATOS EN TIEMPO REAL CON OBD-II</span></h2>
</header>
<div style="height:50px"></div>
<!--Ejemplo tabla con DataTables-->
<div class="container">
<div class="row">
<div class="col-lg-12">
<nav class="navbar navbar-expand-sm bg-basic ">
<!-- Brand/logo -->
<a class="navbar-brand" href="principal.php">

</a>
<button class="navbar-toggler bg-dark navbar-dark" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">

```

```

        <span
class="navbar-toggler-
icon"></span>

    </button>

    <div
class="collapse navbar-
collapse justify-content-
end"
id="collapsibleNavbar">

        <ul
class="navbar-nav">

            <li
class="nav-item">

                <a
class="nav-link"
href="tabla.php">Actualiz
ar Tabla</a>

            </li>

            <li
class="nav-item">

                <a
class="nav-link"
href="principal.php">Reg
resar</a>

            </li>

            <li
class="nav-item
dropdown">

                <a
class="nav-link

```

```

dropdown-toggle"
href="#" id="navbardrop"
data-toggle="dropdown">

    Graficas

    </a>

    <div
class="dropdown-menu
bg-basic">

        <a
class="dropdown-item"
href="grafica.php">Grafic
a de datos</a>

        <a
class="dropdown-item"
href="grafreal.php">Lectu
ra de Datos en Tiempo
real</a>

        <a
class="dropdown-item"
href="medidores.php">Le
ctura de datos con
Medidores</a>

    </div>

    </li>

    </ul>

    </div>

    <div
class="table-responsive">

```

```

    <table
id="example" class="table
table-striped table-
bordered" cellspacing="0"
width="100%">

        <thead>

            <tr>

                <th>Fecha

                <th>-Hora</th>

                <th>Latitud</th>

                <th>Longitud</th>

                <th>Altitud [m]</th>

                <th>RPM</th>

                <th>Velocidad
[Km/h]</th>

                <th>Temperatura
[°C]</th>

                <th>Short Term Fuel
[%]</th>

                <th>Short Term Fuel
[%]</th>

            </tr>

        </thead>

        <tbody>

            <?php

            $sql="SELECT
fecha,lat,lng,alt,rpm,vel,te
mp_air,sft,lft

            from
lec_obd";

            $result=mysqli_query($co
nexion,$sql);

            while($ver=mysqli_fetch_

```



```

row($result)){
$datos= $ver[0]."||".
$ver[1]."||".
$ver[2]."||".
$ver[3]."||".
$ver[4]."||".
$ver[5]."||".
$ver[6]."||".
$ver[7]."||".
$ver[8];
<tr>
<td><?php echo $ver[0]
?></td>
<td><?php echo $ver[1]
?></td>
<td><?php echo $ver[2]
?></td>
<td><?php echo $ver[3]
?></td>
<td><?php echo $ver[4]
?></td>
<td><?php echo $ver[5]
?></td>
<td><?php echo $ver[6]
?></td>
<td><?php echo $ver[7]
?></td>
<td><?php echo $ver[8]
?></td>
</tr>
<?php
}
?>
</tbody>
</table>
</div>
</div>
</div>
<!-- jQuery, Popper.js,
Bootstrap JS -->
<script
src="librerias/jquery-
3.3.1.min.js"></script>
<script
src="librerias/popper.min.
js"></script>
<script
src="bootstrap/js/bootstra
p.min.js"></script>
<script
src="librerias/alertifyjs/ale
rtify.js"></script>
<!-- datatables JS -->
<script
type="text/javascript"
src="datatables/datatables.
min.js"></script>
<!-- para usar botones
en datatables JS -->
<script
src="datatables/Buttons-
1.5.6/js/dataTables.button
s.min.js"></script>
<script
src="datatables/JSZip-
2.5.0/jszip.min.js"></scrip
t>
<script
src="datatables/pdfmake-
0.1.36/pdfmake.min.js"><
/script>
<script
src="datatables/pdfmake-
0.1.36/vfs_fonts.js"></scri
pt>
<script
src="datatables/Buttons-
1.5.6/js/buttons.html5.min
.js"></script>
<!-- código JS propio-->
<script
type="text/javascript"
src="main.js"></script>
</body>
</html>

```

grafica.php

```
<!DOCTYPE html>                                .0/jquery.min.js"></script
>
<html lang="en">                                .welcome a:hover{
<script
<head runat="server">                          color: black;
<meta                                           background: white;
  charset="utf-8">                                }
<meta                                           </style>
  name="viewport"                                </head>
  content="width=device-                          <body>
  width, user-scalable=no,                       <div
  initial-scale=1.0,                              class="container">
  maximum-scale=1.0,                              <div
  minimum-scale=1.0">                            class="row">
<link      rel="shortcut                         <div class="col-
icon" href="#" />                               lg-12">
<title>Graficos                                <div class="panel
OBD-II</title>                                  panel-primary">
<link                                           <div
rel="stylesheet"                                class="panel
type="text/css"                                  panel-
href="librerias/bootstrap/c                    heading">
ss/bootstrap.css">
<script
src="https://cdn.plot.ly/pl                    <button class="btn btn-
otly-                                           info">
latest.min.js"></script>                          <a
<script                                          href="principal.php">
src="https://ajax.googleapis.com/aj             
  background-                                     </a>
  image: url(image/bg9.jpg);                       </button>
```

<p style="text-align: center;">Graficas de Datos</p> <p style="text-align: center;"><button class="btn btn-info"></p> <p style="text-align: center;"></p> <p style="text-align: center;">Regresa r</p> <p style="text-align: center;"></button></p> <p style="text-align: center;"><button class="btn btn-info"></p> <p style="text-align: center;">Actua lizar Grafica</p> <p style="text-align: center;"></p> <p style="text-align: center;"></button></p> <p style="text-align: center;"></div></p> <p style="text-align: center;"><div class="panel panel-body"></p> <p style="text-align: center;"><div class="row"></p>	<p style="text-align: center;"><div class="col-1- sm-12"></p> <p style="text-align: center;"><center><div id="graficaGps_alt"></div ></center></p> <p style="text-align: center;"></div></p> <p style="text-align: center;"><div class="col-2- sm-12"></p> <p style="text-align: center;"><center><div id="graficaRpm_vel"></d iv></center></p> <p style="text-align: center;"></div></p> <p style="text-align: center;"><div class="col-3- sm-3"></p> <p style="text-align: center;"><center><div id="graficaTemp"></div> </center></p> <p style="text-align: center;"></div></p> <p style="text-align: center;"><div class="col-4- sm-3"></p> <p style="text-align: center;"><center><div id="graficaStf"></div></c enter></p> <p style="text-align: center;"></div></p> <p style="text-align: center;"></div></p> <p style="text-align: center;">grafreal.php</p>	<p style="text-align: center;"></div></p> <p style="text-align: center;"></div></p> <p style="text-align: center;"></div></p> <p style="text-align: center;"></div></p> <p style="text-align: center;"></body></p> <p style="text-align: center;"></html></p> <p style="text-align: center;"><script type="text/javascript"></p> <p style="text-align: center;">\$(document).read y(function){</p> <p style="text-align: center;">\$('#graficaGps_alt<br ').load('php="" gps_alt.php');<="" p=""/> <p style="text-align: center;">\$('#graficaRpm_v el').load('php/rpm_vel.php<br ');<="" p=""/> <p style="text-align: center;">\$('#graficaTemp'). load('php/graftemp.php');</p> <p style="text-align: center;">\$('#graficaStf').loa d('php/grafstf.php');</p> <p style="text-align: center;">});</p> <p style="text-align: center;"></script></p> <p style="text-align: center;"><head runat="server"></p> </p></p>
<p style="text-align: center;"><!DOCTYPE html></p>	<p style="text-align: center;"><html lang="en"></p>	

```

        <meta
charset="utf-8">

        <meta
name="viewport"
content="width=device-
width, user-scalable=no,
initial-scale=1.0,
maximum-scale=1.0,
minimum-scale=1.0">

        <title>Graficos
Tiempo Real</title>

        <link
rel="stylesheet"
type="text/css"
href="librerias/bootstrap/c
ss/bootstrap.css">

        <script
src="https://cdn.plot.ly/pl
otly-
latest.min.js"></script>

        <script
src="https://ajax.googleap
is.com/ajax/libs/jquery/2.2
.0/jquery.min.js"></script
>

        <script
type="text/javascript"
src="https://www.gstatic.c
om/charts/loader.js"></scr
ipt>

        <script
type="text/javascript">

```

```

        <script
src="librerias/jquery-
3.3.1.min.js"></script>

        <script
src="librerias/plotly-
latest.min.js"></script>

        <script
src="librerias/highcharts.j
s"></script>

        <script
src="librerias/esporting.js
"></script>

</head>

<body>

        <div
class="container">

                <div
class="row">

                        <div class="col-
sm-12">

                                <div class="panel
panel-primary">

                                        <div class="panel
panel-heading">

                                                <button class="btn btn-
info">

                                                        <a

                                                                </div>

</div>

                                <div
class="panel
panel-body">

```

```

alt="logo"
style="width:60px;">

</a>

</button>

        Graficas de Datos
obtenidos en tiempo real

        <button
class="btn btn-info">

                <span
class="glyphicon
glyphicon-arrow-
left"></span>

                <a
href="tabla.php">Regresa
r</a>

                </button>

                <button
class="btn btn-info">

                        <a
href="grafreal.php">Actu
alizar Grafica</a>

                                <span
class="glyphicon
glyphicon-
refresh"></span>

                                        </button>

                                                </div>

</div>

                <div
class="panel
panel-
body">

```

```

        <div
class="row">
        <div class="col-
sm-6">
        <center><div
id="Altitud"
style="height:
300px;"></div></center>
        </div>
        <div
class="col-sm-6">
        <center><div
id="Revoluciones"
style="height:
300px;"></div></center>
        </div>
        <div
class="col-sm-6">
        <center><div
id="Velocidad"
style="height:
300px;"></div></center>
        </div>
        <div
class="col-sm-6">

```

```

        <center><div
id="Temperatura"
style="height:
300px;"></div></center>
        </div>
        <div class="col-
sm-6">
        <center><div
id="Stf" style="height:
300px;"></div></center>
        </div>
        <div class="col-
sm-6">
        <center><div
id="Ltf" style="height:
300px;"></div></center>
        </div>
        </div>
        </div>
        </div>
        </div>

```

```

</body>
</html>
<script
type="text/javascript">
        $(document).read
y(function(){
        $('#Velocidad').lo
ad('php/realtime/vel_real.
php');
        $('#Revoluciones').load('p
hp/realtime/rev_real.php');
        $('#Altitud').load('php/real
time/alt_real.php');
        $('#Temperatura').load('ph
p/realtime/tem_real.php');
        $('#Stf').load('php/
realtime/stf_real.php');
        $('#Ltf').load('php
/realtime/ltf_real.php');
        });
</script>

```

medidores.php

```

<!DOCTYPE html>
<html lang="en">

```

```

<head runat="server">
        <meta
charset="utf-8">

```

```

        <meta
name="viewport"
content="width=device-

```

```
width, user-scalable=no,
initial-scale=1.0,
maximum-scale=1.0,
minimum-scale=1.0">
```

```
<link rel="shortcut
icon" href="#" />
```

```
<title>Graficos
Medidores</title>
```

```
<link
rel="stylesheet"
type="text/css"
href="librerias/bootstrap/c
ss/bootstrap.css">
```

```
<script
src="https://cdn.plot.ly/pl
otly-
latest.min.js"></script>
```

```
<script
src="https://ajax.googleap
is.com/ajax/libs/jquery/2.2
.0/jquery.min.js"></script
>
```

```
<script
type="text/javascript"
src="https://www.gstatic.c
om/charts/loader.js"></scr
ipt>
```

```
<script
type="text/javascript">
```

```
<script
src="librerias/jquery-
3.3.1.min.js"></script>
```

```
<script
src="librerias/plotly-
latest.min.js"></script>
```

```
<script
src="librerias/highcharts.j
s"></script>
```

```
<script
src="librerias/esporting.js
"></script>
```

```
<script
type="text/javascript"
src="datatables/datatables.
min.js"></script>
```

```
<style>
body{
background-
image: url(image/bg6.jpg);
}
```

```
.welcome a:hover{
color: black;
background: white;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div
class="container">
```

```
<div
class="row">
```

```
<div class="col-
lg-12">
```

```
<div class="panel
panel-primary">
```

```
<div
class="panel panel-
heading">
```

```
<button class="btn btn-
info">
```

```
<a
href="principal.php">
```

```

```

```
</a>
```

```
</button>
```

```
Graficos con
Medidores en Tiempo Real
```

```
<button
class="btn btn-info">
```

```
<span
class="glyphicon
glyphicon-arrow-
left"></span>
```

```

        <a
href="tabla.php">Regresa
r</a>
    </button>
    <button
class="btn btn-info">
        <a
href="medidores.php">Ac
tualizar Grafica</a>
        <span
class="glyphicon
glyphicon-
refresh"></span>
    </button>
    </div>
    <div
class="panel
panel-
body">
        <div
class="row">
            <a
href="tabla.php">Regresa
r</a>
            </div>
            <div
class="col-
sm-4">
                <center><div
id="Medidor_rpm"></div
></center>
            </div>
            <div
class="col-
sm-4">
                <center><div
id="Medidor_vel"></div>
</center>
            </div>
            <div
class="col-
sm-4">
                <center><div
id="Medidor_temp"></di
v></center>
            </div>
            </div>
        </div>
    </div>
</body>
</html>
<script
type="text/javascript">
    $(document).read
y(function(){
    $('#Medidor_rpm').load('p
hp/med_rpm.php');
    $('#Medidor_vel').
load('php/med_vel.php');
    $('#Medidor_temp
').load('php/med_temp.ph
p');
});
</script>

```

insertar.php

```

<?php
include("../conexion.php");
if ($_GET['lat']!=0.0' &&
$_GET['rpm']>0.0){
    if(isset($_GET['lat']))
        $lat = $_GET['lat'];
        $fech = $_REQUEST['fech'];
        $lng = $_REQUEST['lng'];
        $alt = $_REQUEST['alt'];
        $rpm = $_REQUEST['rpm'];
        $vel = $_REQUEST['vel'];
}

```

```

    $temp_air      =          $username      =          // use exec()
$_REQUEST['temp_air'];    "cristhian";          because no results are
                          returned
    $sft           =          $password      =
$_REQUEST['sft'];        "tesisobd";          $conn-
                          >exec($sql);
    $lft           =          $dbname        =
$_REQUEST['lft'];        "db_obd2";          echo "Dato
                          insertado";
                          $conn = new
                          PDO("mysql:host=$server
name;dbname=$dbname",
                          $username, $password);
    echo $fech, "\t";
    echo $lat, "\t";          // set the PDO
    echo $lng, "\t";          error mode to exception
    echo $alt, "\t";          $conn-
    echo $rpm, "\t";          >setAttribute(PDO::ATT
    echo $vel, "\t";          R_ERRMODE,
    echo $temp_air, "\t";    PDO::ERRMODE_EXCE
    echo $sft, "\t";          PTION);
    echo $lft, "\t";
                          $sql      =
                          "INSERT INTO lec_obd
                          (fech, lat, lng, alt, rpm, vel,
                          temp_air, sft, lft)
                          VALUES
                          ($fech', '$lat', '$lng', '$alt',
                          '$rpm', '$vel', '$temp_air',
                          '$sft', '$lft)";
    try {
        $servername =
"localhost";

```


ANEXO F:

Configuración del programa principal el Python

```
import serial                                serdata = dat1_int =
import os                                    os.linesep.join([s for s in        int(dat1,16)
serdata.splitlines() if s])
import pynmea2                                sensor_rev =
                                                ((256*dat0_int)+dat1_int)/
import urllib                                  4
import string                                if
import time                                    serdata.find('GGA') > 0:                revoluciones
                                                = str(sensor_rev)
                                                datagga =
pynmea2.parse(serdata)

def goto(linenum):
    global line
    line = linenum
    if
    serdata.find('VTG') > 0:
        datavtg =
        pynmea2.parse(serdata)
        scan.write(b'010D\r')
        #COMANDO AT
        VELOCIDAD
        vel =
        scan.readall()
        dat2 =
        vel[6:8]
        dat2_int =
        int(dat2,16)
        velocidad =
        str(dat2_int)
        scan.write(b'010C\r')
        #COMANDO AT
        REVOLUCIONES POR
        MINUTO
        rev =
        scan.readall()
        dat0 =
        rev[6:8]
        dat1 =
        rev[9:11]
        dat0_int =
        int(dat0,16)
        scan.write(b'010F\r')
        #COMANDO AT
        ENTRADA DE AIRE
        air =
        scan.readall()
        dat3 = air[6:8]
        dat3_int =
        int(dat3,16)

print "iniciando..."
time.sleep(3)
ser=serial.Serial("/dev/tty
AMA0", baudrate = 9600)
scan =
serial.Serial('/dev/rfcomm
0', 115200,
timeout=0.100)
line = 1
while True:
    if
    (ser.inWaiting())>0):
        serdata =
        ser.readline()
```

```

        sensor_a =
dat3_int - 40
        aire =
str(sensor_a)

scan.write(b'0106\r')
#COMANDO
CONSUMO A CORTO
PLAZO

sf =
scan.readall()

dat4 = sf[6:8]
dat4_int =
int(dat4,16)

sensor_sft =
(dat4_int*100/128) - 100

stf =
str(sensor_sft)

        scan.write(b'0107\r')
#COMANDO
CONSUMO A LARGO
PLAZO

        if =
scan.readall()

        dat5 = sf[6:8]
        dat5_int =
int(dat5,16)

        sensor_lft =
(dat5_int*100/128) - 100

        ltf =
str(sensor_lft)

campo='&fech='+str(time.
strftime('%d/%b/%Y
%H:%M:%S'))+'&lat='+st
r(datagga.latitude)+'&lng=
'+str(datagga.longitude)+'

        &alt='+str(datagga.altitud
e)+'&rpm='+str(revolucio
nes)+'&vel='+str(velocida
d)+'&temp_air='+str(aire)
+'&sft='+str(stf)+'&lft='+s
tr(ltf)

url='http://54.207.19.109/s
istema-
obd2/inc/insertar4.php?'+c
ampo

        sitio =
urllib.urlopen(url)

        print
(sitio.read())

time.sleep(0.001)

except:

        print "Error"

        goto (1)

```



ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO



DIRECCIÓN DE BIBLIOTECAS Y RECURSOS
PARA EL APRENDIZAJE Y LA INVESTIGACIÓN

UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 13/04/2021

INFORMACIÓN DEL AUTOR
Nombres – Apellidos: CRISTHIAN JORGE CARDOSO CARRAZCO
INFORMACIÓN INSTITUCIONAL
Facultad: INFORMÁTICA Y ELECTRÓNICA
Carrera: INGENIERÍA EN ELECTRÓNICA TELECOMUNICACIONES Y REDES
Título a optar: INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES
f. Analista de Biblioteca responsable:

