



*Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingeniería de Sistemas Informáticos
Máster en Software de Sistemas Distribuidos y Empotrados
Proyecto Fin de Máster*

VIMS – Vehicle IoT Metrics System

Javier Alonso Silva

Tutor: Norberto Cañas de Paz

Madrid, 10 de julio de 2022

Índice general

Índice de figuras	III
Índice de cuadros	v
Definiciones, siglas, y abreviaturas	xI
1 Introducción	3
1.1 Estado del arte	3
1.1.1 Herramientas de monitorización y control del automóvil	8
1.2 Objetivos del desarrollo del proyecto	11
1.3 Metodología	12
2 Estructura del proyecto	14
2.1 Estudio de mercado	15
2.2 Estudio matemático	25
2.3 Diseño <i>software</i>	33
2.4 Diseño <i>hardware</i>	34
2.5 Análisis de planificabilidad	38
3 Especificación de requisitos	40
3.1 Introducción	40
3.1.1 Propósito	40
3.1.2 Alcance	41
3.1.3 Estructura del documento	42
3.2 Descripción general del producto	42
3.2.1 Perspectiva del producto	42
3.2.2 Características de los usuarios finales	44
3.2.3 Restricciones generales	44
3.2.4 Suposiciones y dependencias	45
3.3 Requisitos específicos	46
3.3.1 Requisitos de usuario	46
3.3.2 Requisitos funcionales	50
3.3.3 Requisitos no funcionales	53
3.3.4 Requisitos de interfaces externas	55
3.3.5 Restricciones de desarrollo	55

3.3.6	Requisitos de entorno físico	55
4	Diagramas y diseño	56
4.1	Diagramas que modelan el sistema	56
4.1.1	Casos de uso	56
4.1.2	Diagramas de bloques	69
4.2	Diseño <i>hardware</i> del sistema	80
4.3	Diseño 3D del sistema	80
4.4	Planificación del sistema de tiempo real	80
4.5	Diseño <i>software</i> del sistema	82
5	Planificación, costes y tiempo empleado	83
5.1	Diagramas de Gantt	83
5.2	Coste de los materiales	83
5.3	Sueldos propuestos y costes obtenidos	83
5.4	Contratiempos y tiempo de desarrollo final	83
6	Conclusiones	84
6.1	Conclusiones técnicas	84
6.2	Conocimientos adquiridos y nuevas competencias	84
6.3	Reflexión final	84
7	Trabajo pendiente y futuras líneas de trabajo	85
	Bibliografía	86

Índice de figuras

1.1	Ford modelo T del 1927 – de Rmhermen [11]	4
1.2	Volkswagen Tipo III, modelo de inyección de 1969 – de OSX - Trabajo propio [12].	5
1.3	Conecotor <i>Computerized Assembly Line Diagnostics Link</i> (ALDL), creado por <i>General Motors</i> antes de estandarizar OBD-II [14].	6
1.4	Controlador ELM327 que cuenta con antenas Wi-Fi y Bluetooth para un acceso remoto [15].	7
1.5	Esquema seguido para determinar los hábitos de conducción usando el <i>On-Board Diagnostics</i> (OBD)-II [1].	9
1.6	Flujo de análisis para determinar el comportamiento al volante de un conductor [18].	10
1.7	Esquema general que modela el modo de funcionamiento del sistema.	11
2.1	Gráfico de tarta que muestra quiénes de los encuestados son conductores (72,7 %) y quiénes no (27,3 %).	16
2.2	Histograma que muestra los años de carnet de los encuestados.	17
2.3	Gráfico de tarta que muestra la distribución de edades (valor previo al paréntesis) y su frecuencia en porcentaje.	18
2.4	Distribución de las características de los vehículos de los encuestados.	20
2.5	Placa de desarrollo LILYGO TTGO T-SIM7000G usada en el proyecto [20].	36
2.6	Conecotor OBD-II hembra con posibilidad de definir los pines de salida [21].	37
2.7	Conecotor OBD-II hembra con cables ya incluidos [22].	38
3.1	Representación de los datos en la interfaz propuesta – fuente: https://grafana.csselectronics.stellarhosted.com/d/6qvL90vMz/css-playground-obd2?orgId=1	50
4.1	Caso de uso 01 – <i>autenticación</i>	57

4.2	Casos de uso <i>02 – generación y transmisión de datos</i>	58
4.3	Caso de uso <i>03 – generación de estadísticas</i>	60
4.4	Caso de uso <i>04 – envío de notificaciones</i>	61
4.5	Caso de uso <i>05 – visualización en tiempo real</i>	63
4.6	Caso de uso <i>06 – generación de eventos</i>	65
4.7	Casos de uso <i>07.* – envío, almacenamiento y visualización de geolocalización</i> . .	66
4.8	Diagrama de bloques que modela el conjunto de la placa <i>Vehicle IoT Metrics System</i> (VIMS) y sus componentes.	70
4.9	Subsistema que define las comunicaciones usando el <i>Global Positioning System</i> (GPS) para obtener la ubicación.	71
4.10	Subsistema que define las comunicaciones con el vehículo usando OBD-II. . . .	72
4.11	Subsistema que define las comunicaciones de red con el exterior.	73
4.12	Subsistema de conexionado Bluetooth para la transmisión en vivo de los datos. .	74
4.13	Componente VIMS dentro del sistema embebido.	75
4.14	Diagrama de bloques que modela al servidor y sus componentes.	76
4.15	Subsistema del servidor que conforma la arquitectura web del mismo.	78
4.16	Subsistema del servidor que conforma la gestión de datos y usuarios.	79
4.17	Diagrama global del sistema. Como se puede ver, se sigue una arquitectura cliente–servidor.	79
4.18	Diagrama esquemático del sistema con las tareas y objetos protegidos que lo conforman.	81

Índice de cuadros

2.1	Tabla de distribución de las características de los vehículos, preguntado en el cuestionario.	19
2.2	Opciones ofrecidas a los encuestados. Se han escogido diversas opciones que se encuentran entre los datos habituales generados por un vehículo.	21
2.3	Puntuaciones obtenidas de forma general, por la población al completo (conductores y no conductores).	22
2.4	Puntuaciones obtenidas de forma general, por la población (excluidos los no conductores).	22
2.5	Vector de <i>bytes</i> que representa los datos recibidos del conector OBD-II [7].	25
2.6	Lista de modos de funcionamiento del estándar OBD-II [7].	26
2.7	Obtención de los <i>Parameter ID</i> (PID)s soportados según el modo [7].	26
2.8	PID 04 – carga del motor, en %.	27
2.9	PID 05 – temperatura del refrigerante del motor, en °C.	27
2.10	PID 0C – velocidad del motor, en <i>RPM</i>	27
2.11	PID 0D – velocidad del vehículo, en <i>km/h</i>	28
2.12	PID 11 – posición del acelerador, en %.	28
2.13	PID 2F – nivel del tanque del combustible, en %.	28
2.14	PID 46 – temperatura ambiente, en °C.	29
2.15	PID 5B – tiempo restante de la batería híbrida, en %.	29
2.16	PID 5C – temperatura del aceite, en °C.	29
2.17	PID 5E – consumo actual del motor, en <i>L/h</i>	30
2.18	PID 61 – torque demandado por el conductor, en %.	30
2.19	PID 62 – torque actual del motor, en %.	30
2.20	PID 63 – torque de referencia del motor, en <i>N · M</i>	31

2.21	PID A4 – marcha actual del vehículo, en ratio.	31
2.22	PID A6 – odómetro, en <i>km</i> .	31
2.23	PID 02 – <i>Vehicle Identification Number</i> (VIN).	33
2.24	PID 0A – nombre de la <i>Engine Control Unit</i> (ECU).	33
3.1	Opciones ofrecidas a los encuestados. Se han escogido diversas opciones que se encuentran entre los datos habituales generados por un vehículo.	47
3.2	Puntuaciones obtenidas de forma general, por la población al completo (conductores y no conductores).	48
3.3	Puntuaciones obtenidas de forma general, por la población (excluídos los no conductores).	48

Definiciones, siglas, y abreviaturas

ADAS *Advanced Driver Assistance Systems*

ADC *Analog–Digital Converter*

ALDL *Computerized Assembly Line Diagnostics Link*

API *Application Programming Interface*

BLE *Bluetooth Low Energy*

BR/EDR *Basic Rate/Enhaced Data Rate*

CAN *Controller Area Network*

CARB *California Air Resources Board*

CRUD *Create Read Update Delete*

DTC *Diagnostic Trouble Codes*

ECU *Engine Control Unit*

EOBD *European On-Board Diagnostics*

GB *gigabyte*

GPIO *General Purpose Input/Output*

GPS *Global Positioning System*

GUI *Graphical User Interface*

IoT *Internet of Things*

JOBD *Japanese On-Board Diagnostics*

LTE *Long-Term Evolution*

MIL *Malfunction Indicator Lamp*

MSB *Most Significant Bit*

NTP *Network Time Protocol*

OBD *On-Board Diagnostics*

P2P *Point-to-Point*

PAN *Personal Area Network*

PFM Proyecto Fin de Máster

PID *Parameter ID*

PWM *Pulse-Width Modulation*

REST *REpresentational State Transfer*

RPM Revoluciones Por Minuto

SAE *Society of Automotive Engineers*

SoC *System on Chip*

UART *Universal Asynchronous Receiver–Transmitter*

ULP *Ultra Low Power*

UTC *Coordinated Universal Time*

VIMS *Vehicle IoT Metrics System*

VIN *Vehicle Identification Number*

- *Advanced Driver Assistance Systems* (ADAS) – sistema de control de un vehículo que utiliza sensores del entorno (radar, láser, visión por computador) para mejorar la seguridad al volante y la seguridad del entorno ayudando a los conductores a reconocer y reaccionar ante eventos o situaciones de tráfico potencialmente peligrosas [1].
- *Application Programming Interface* (API) – conjunto de definiciones, subrutinas y protocolos que ofrecen ciertos *softwares* para ser usados por otras aplicaciones como capa de abstracción sobre el original [2].
- *Bluetooth Low Energy* (BLE) – tecnología *Personal Area Network* (PAN) que permite la comunicación entre dispositivos con un rango similar a Bluetooth pero un menor consumo de energía.
- *Basic Rate/Enhaced Data Rate* (BR/EDR) – terminología asociada al estándar Bluetooth que define el modo de funcionamiento clásico del mismo. Se caracteriza por contar con 79 canales de 1 *MHz* de ancho de banda, operar en la red de 2,4 *GHz*, ser *Point-to-Point* (P2P), transmitir a una velocidad máxima de 3 *Mbps* y tener un consumo medio de 1 *W* [3].
- Bus *Controller Area Network* (CAN) – protocolo de comunicaciones de tiempo real para el envío de mensajes en entornos distribuidos, permitiendo la comunicación entre múltiples CPUs.
- *Create Read Update Delete* (CRUD) – funciones básicas que se realizan sobre una base de datos y que conforman el conjunto mínimo de operaciones que se deben poder realizar sobre una.
- *Diagnostic Trouble Codes* (DTC) – códigos de diagnóstico almacenados en un vehículo que identifican un error que tiene el vehículo o que ha tenido.
- ECU – centralita electrónica conectado a todos los sensores y sistemas del vehículo que

recibe la información de los mismos y ejecuta comandos o acciones contra ellos.

- *gigabyte* (GB) – unidad de almacenamiento de información equivalente a 10^9 bytes.
- *General Purpose Input/Output* (GPIO) – pin genérico cuyo comportamiento puede ser controlado en tiempo de ejecución.
- GPS – sistema que permite posicionar cualquier objeto con una precisión de hasta centímetros usando cuatro o más satélites y trilateración [4].
- *Graphical User Interface* (GUI) – conjunto de componentes visuales que permiten a los usuarios interactuar con una aplicación de manera visual e intuitiva.
- *Internet of Things* (IoT) – concepto que se refiere a la interconexión digital de objetos cotidianos con Internet [5].
- *jitter* – retardo relativo que se produce en las comunicaciones y que afecta directamente a la saturación de la red y a la capacidad de transmisión de la misma.
- *Long-Term Evolution* (LTE) – también conocido como “4G”, es la cuarta generación del estándar de conexión móvil a Internet que mejora a su predecesor, el 3G.
- *Most Significant Bit* (MSB) – bit, de acuerdo con su posición, tiene mayor valor (extremo izquierdo).
- *Network Time Protocol* (NTP) – protocolo de sincronización temporal que funciona sobre UDP usando para mantener los relojes de los sistemas informáticos en base al reloj atómico universal, que es *stratum 1*.
- OBD – sistema de diagnóstico a bordo de vehículos que cuenta con múltiples estándares según la región de uso. Estos sistemas ofrecen una monitorización activa y control completo sobre el motor y otros dispositivos del vehículo [6].
- P2P – protocolo de comunicación entre dos nodos en donde sendos nodos se interconectan entre sí para el intercambio de información sin necesidad de usar un intermediario.
- PAN – redes destinadas a la comunicación entre dispositivos en una misma red o malla. Tiene un alcance muy limitado, de unos pocos metros por lo general.
- PID – parámetro utilizado en la comunicación mediante OBD-II que identifica el tipo de petición que el mecánico/usuario quiera hacer al sistema. Son valores enteros de 16 bits que funcionan en formato clave–valor. La información devuelta por el conector es también un conjunto de bytes de 32 bits en total que representan la información solicitada por el usuario. Por lo general, se deben realizar ciertas operaciones aritméticas con ese conjunto de bytes para obtener la representación decimal del valor [7].
- *Pulse–Width Modulation* (PWM) – señal cuadrada de periodo habitualmente constante, entre flancos de subida, en la que se modula el tiempo a nivel alto.
- *REpresentational State Transfer* (REST) – estilo de arquitectura *software* para sistemas

hipermedia distribuidos, como la World Wide Web. En la actualidad se usa en sentido más amplio para describir cualquier interfaz entre sistemas que usa directamente HTTP para obtener datos o indicar una operación sobre un conjunto de datos en cualquier formato (aunque habitualmente se usa JSON o XML) [8].

- *Smog* – nube baja formada por dióxido de carbono, hollín, humo y polvo en suspensión que se forma sobre las grandes ciudades o núcleos industriales.
- *System on Chip* (SoC) – denominación asociada a los sistemas (generalmente embebidos) que aúnan todos los módulos que requieren para funcionar (y otros tantos para añadir más funcionalidades) en un único *chip* que expone al exterior dichas funcionalidades mediante una interfaz *hardware* como si fuese un único circuito integrado.
- trilateración – método matemático que permite determinar las posiciones relativas de objetos usando la geometría de los triángulos [9].
- *Coordinated Universal Time* (UTC) – principal estándar de tiempo por el cual el mundo regula los relojes y el tiempo. Entre otras características, es un reloj monotónico (no puede volver al pasado).
- VIN – código único, incluyendo el número de serie, que identifica individualmente los vehículos a motor, a remolque, motocicletas, *scooters* y *mopeds*, tal como se define en las ISO 3779 (contenido y estructura) e ISO 4030 (ubicación y acoplamiento) [10].

En este documento se va a tratar el desarrollo del pArm, un proyecto integral de ingeniería en el que se modela, diseña y construye un brazo robótico utilizando tecnología de impresión 3D como base. El objetivo principal es ofrecer una forma asequible y sencilla para que otra persona pueda replicar el proyecto y adentrarse en el mundo de la robótica por su cuenta.

Para ello, primero se elicitarán los requisitos que permitirán posteriormente modelar y diseñar el sistema de forma fiel. A su vez, se estudiarán las características del sistema *hardware* lo que permitirá desarrollar y construir una placa de control que será la encargada de gestionar los movimientos del brazo robótico.

Además, las fases de diseño anteriores simplifican el proceso de desarrollo del *software* que ejecutarán los sistemas y permitirán abordar el modelo matemático que rige la estructura pantográfica del brazo robótico atendiendo a las limitaciones tanto físicas como del sistema propuesto en sí.

Por otro lado, se modelarán y diseñarán nuevas piezas que permitirán construir el brazo robótico con otros tipos de componentes distintos a los del brazo original así como con la nueva placa de control.

Por último, se proponen futuras líneas de mejora que se consideran interesantes a la hora de completar el proyecto. Se incluyen además en los anexos el código fuente de las aplicaciones desarrolladas ya que se referencia directamente a lo largo del presente documento.

The pArm development, an integral engineering project which models, designs and builds a robotic arm using 3D printing technology as a basis is going. The main objective is to offer an affordable and easy way for everyone to replicate this project so they can introduce themselves within robotics community.

Firstly, the requirements will be elicited which, will allow the modeling and the design of the system in further steps of the development. Concurrently, the hardware characteristics will be studied in order to allow the development and construction of the board that will handle the movements of the robotic arm.

In addition, the mentioned design steps simplify the software development process alongside the mathematical model, which is defined by both the physical structure itself and the proposed system.

On the other hand, new pieces will be designed in order to make the robotic manipulator compatible with new external componentes, compared to the original ones used in the μ Arm, and the new designed board.

Finally, new improvements that the team considers interesting to complete will be proposed. The annexes are included with the source code developed for the applications, as they are directly referenced in the present document.

1 Introducción

En un mundo cada vez más interconectado, hay ciertas tecnologías que se quedan por detrás en unos campos mientras que siguen progresando en otros. Esto se ve directamente reflejado en la industria automovilística en donde los vehículos cada vez cuentan con mayor y mejor tecnología (como cámaras, sensores, actuadores, etc.) pero no es directamente accesible por el usuario: mediante pantallas e interfaces se ofrecen métodos sencillos que facilitan su uso.

VIMS pretende ser un sistema que facilite el acceso a todos los datos que ofrece un vehículo para generar estadísticas, descubrir patrones en la conducción y detectar errores. De esta forma, el conductor tendrá información de primera mano sobre el estado de su vehículo, eficiencia de su conducción así como obtener información en tiempo real complementaria a la ya propiciada por el vehículo.

1.1. Estado del arte

La historia de la automoción comienza estrictamente en el siglo XIX. Un automóvil es, por definición, un vehículo que se mueve a sí mismo (del griego, *αὐτός* “a sí mismo” y del latín *mobilis*, “que se mueve”).

Desde los primeros modelos como la serie T, de Ford, hasta el inicio de la fabricación de vehículos por parte de Mercedes Benz, la historia del automovilismo ha estado llena de grandes logros y avances en un intervalo de tiempo relativamente pequeño (figura 1.1):



Figura 1.1: Ford modelo T del 1927 – de Rmhermen [11].

Durante aquella época, el “mejor” mecanismo de descubrimiento de problemas era con algunos medidores y, sobre todo, por intuición: sonidos del motor, olores extraños, ...

No fue hasta los años 60 en donde los vehículos empezaron a incorporar distintas interfaces con métricas que podían informar sobre el estado del vehículo. El gran “bum” llegó con la expansión de la computadora, en donde por primera vez se vio factible introducir un pequeño ordenador de abordo en el sistema.

En estos primeros sistemas, se incluyen indicadores del nivel de combustible, sistema de refrigeración, presión del aceite, velocidad del motor, temperatura del motor y otra información relativa al combustible. El primer modelo que se conoce que incluye estos sistemas de cara a la población en general es el Volkswagen Tipo III, en 1969 (figura 1.2):



Figura 1.2: Volkswagen Tipo III, modelo de inyección de 1969 – de OSX - Trabajo propio [12].

Pese a que supusieron un gran avance, estos primeros sistemas de diagnóstico daban una información muy valiosa pero limitada, ya que muchos diagnósticos seguirían siendo mediante los sentidos y las sensaciones que le transmitiese el vehículo al mecánico. No fue hasta 1980 en donde se implementó de forma estándar en los vehículos de *General Motors* el ALDL, un lector de errores del coche que funcionó inicialmente a 160 baudios. Años más tarde, el sistema se refinaría usando el estándar *Universal Asynchronous Receiver–Transmitter* (UART) *half-duplex*, es decir, transmisión en los dos sentidos pero no de forma simultánea (figura 1.3). La principal motivación de incluir estos sistemas no fue otra sino intentar reducir la contaminación de los vehículos teniendo acceso a esta información [13].

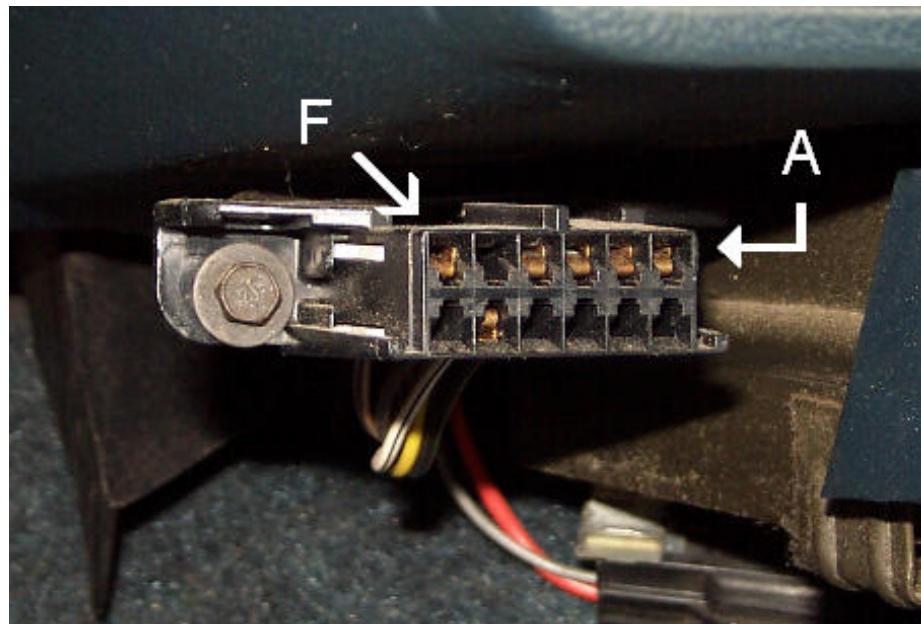


Figura 1.3: Conector ALDL, creado por *General Motors* antes de estandarizar OBD-II [14].

No es hasta 1979 en que la *Society of Automotive Engineers* (SAE) recomienda crear un conector de diagnóstico estandarizado en el mercado así como un conjunto de señales de prueba. Finalmente, en 1991 la *California Air Resources Board* (CARB) require que todos sus vehículos tengan lo que sería el OBD-I. Esta primera versión informaría al conductor de un mal funcionamiento de alguno de los elementos del vehículo mediante un *Malfunction Indicator Lamp* (MIL), un indicador luminoso de fallos. Sin embargo, solo se monitorizaban ciertos componentes relacionados con las emisiones y no estaban calibrados [13].

Por ende, impulsado por las alertas *smog* y por una fuerte regulación, en 1994 la CARB obligó a todos los vehículos fabricados y vendidos a partir de 1996 incluir el OBD-II. Esto supuso un gran impulso en los mecanismos de monitorización de los vehículos ya que además se estandarizó tanto el conector como los protocolos recomendados por la SAE. Tiempo más tarde, el Gobierno de los Estados Unidos aplicó la misma medida a todos los vehículos del país [13].

Esta medida llegaría a Europa en 1998, según la Directiva 98/69EG, que obligaba a todos los vehículos europeos a incluir dicho conector. Específicamente, los automóviles de gasolina debían empezar a equiparlo en los modelos del año 2000; en el año 2003 para vehículos diésel; y en el año 2005 para camiones [13].

OBD-II

OBD-II es la segunda generación del sistema de diagnósticos de abordo, sucesor de OBD-I y su principal función es la de avisar al conductor cuando las emisiones del vehículo son en torno

a 1,5 veces mayores de las diseñadas. A diferencia de OBD-I, OBD-II también detecta fallos eléctricos, químicos y mecánicos que puedan afectar al nivel de emisiones del vehículo (un caso típico era un fallo químico del catalizador, indetectable por OBD-I pero sí por la segunda generación).

Este tipo de conector, al ser el primer estándar, cuenta con múltiples interfaces que permiten conexiones mediante redes Wi-Fi, USB, Bluetooth, etc. (cayendo pues en desuso el protocolo de conexión por puerto serie – RS232). Esto se ha conseguido gracias al rápido avance de los sistemas embebidos, en donde la combinación de *software* y *hardware* embebidos ha permitido que cualquier usuario tenga acceso a este tipo de datos de forma relativamente simple (actualmente, el conector más estandarizado es el controlador ELM327 [13], figura 1.4):



Figura 1.4: Controlador ELM327 que cuenta con antenas Wi-Fi y Bluetooth para un acceso remoto [15].

El sistema verifica todos los sensores directamente involucrados con las emisiones del vehículo, como por ejemplo la inyección de aire al motor. Cuando algún sensor detecta un fallo, se activa el MIL indicando el fallo que sucede (o una combinación de indicadores para notificar la existencia de un fallo, sin especificar exactamente cuál).

El vehículo que incorpora un conector OBD-II almacena la información sobre el fallo del vehículo para que el mecánico que deba revisar el automóvil disponga de todos los datos posibles. Por otra parte, este estándar permite una comunicación directa con el vehículo mediante el envío de órdenes según un PID. Por defecto, hay una serie de PIDs estándar que la gran mayoría de vehículos deben incluir¹, pero también los fabricantes de vehículos incluyen una serie de PIDs propios (conocidos como *PIDs propietarios*) que ofrecen información adaptada a cada vehículo en particular y que, en principio, son privados y cerrados al público en general.

¹Se dice “la mayoría de vehículos” porque hay ciertos PIDs que dependen directamente del tipo de vehículo (a combustión, eléctrico, ...) o del combustible utilizado – un vehículo eléctrico no ofrece información sobre las Revoluciones Por Minuto (RPM) al igual que un vehículo diésel no ofrece información sobre las bujías.

En Europa se implantó el *European On-Board Diagnostics* (EOBD), la variación europea del estándar OBD-II implantada en el año 2000 en general. Si bien en apariencia es semejante al OBD-II, las diferencias radican en el *software*. Por ejemplo, el estándar europeo no monitoriza las evaporaciones del depósito de combustible; sin embargo, es más sofisticado ya que usa “mapas” en las entradas de los sensores que obligan a que el sensor se calibre empíricamente al sistema según las condiciones de operación del motor (lo cual se traduce en que los sensores son mucho mejores pero más caros) [13]. Otra característica innovadora es que el sistema europeo registra cuántos kilómetros se han recorrido desde que ha aparecido un defecto [16].

Finalmente, pero no menos importante, Japón tiene también su propio estándar denominado *Japanese On-Board Diagnostics* (JOBD).

OBD-III

El OBD-III se espera que sea la siguiente versión del sistema que ya implementan los coches actualmente. La principal diferencia con respecto a la versión anterior será que el vehículo estará conectado de forma continua y emitiendo datos referentes a las emisiones. De esta forma, se puede saber casi en el momento acerca de modificaciones ilegales, un aumento en la contaminación del coche (signo de deterioro) y demás. No se espera igualmente que sea un salto cualitativo ya que se sigue buscando que sea altamente compatible con las herramientas que ya existen. Actualmente, se están realizando pruebas en EE.UU. pero no hay cerrada ninguna fecha de estandarización oficial por parte de los distintos continentes.

1.1.1. Herramientas de monitorización y control del automóvil

Pese al tiempo que lleva OBD-II disponible, las herramientas existentes para la actuación sobre un vehículo son relativamente escasas. La mayoría de modelos presentes hoy en día en el mercado se basan directa o indirectamente en el ELM327, un dispositivo de diagnóstico OBD que cuenta con conexión WiFi, Bluetooth y serie para la lectura local.

Por lo general, las herramientas que hay se utilizan por mecánicos o fanáticos del sector para acceder a la información del estado del vehículo y ver los errores que pudiera tener. Sin embargo, tras una breve documentación sobre el tema, la mayoría de los casos buscaban directamente monitorizar en el momento el estado del vehículo para obtener información relativa a los consumos, contaminación, etc.

Por ejemplo, en el trabajo de Rimpas *et al.* [17], se utiliza un sensor ELM327 para verificar que la información proporcionada por el puerto del vehículo y la presentada por la telemetría presente en el mismo (velocímetro y tacómetro) son coherentes entre sí (previa adaptación de los valores en *bytes* presentados por el conector a valores legibles). En dicha investigación se llega a la conclusión de que el conector OBD-II obtiene valores fiables y consistentes tanto con los mostrados por el propio vehículo como los proporcionados por el fabricante.

Otro tipo de investigaciones llevadas a cabo gracias a la presencia de este conector en los automóviles es la de la caracterización de conductores y hábitos de conducción según la telemetría reportada por el vehículo. En el estudio realizado por Galih Hermawan y Emir Husni [1] se estudia la combinación de la lectura de los sensores mediante el OBD-II con vehículos que presentan el sistema ADAS. El estudio busca identificar hábitos de conducción según la lectura de los diversos sensores que hay en el sistema. También persigue detectar quién es el conductor que está llevando el vehículo actualmente. En el estudio, el uso de OBD-II junto con los algoritmos de los *k*-Nearest Neighbor (*k*-NN) y *Naive Bayes* consiguieron una precisión en la identificación del 100 % (para un conjunto de datos de 10 conductores). Por otra parte, el uso de inteligencia artificial junto con técnicas de *clustering* permitieron identificar comportamientos de los conductores al volante y relacionarlo además con situaciones de riesgo y peligro. Además, se ha aplicado a otras características también interesantes como detectar el tipo de calzada, predecir el tiempo de viaje, analizar el consumo del vehículo y demás. El esquema seguido en la investigación es el que se presenta en la figura 1.5:

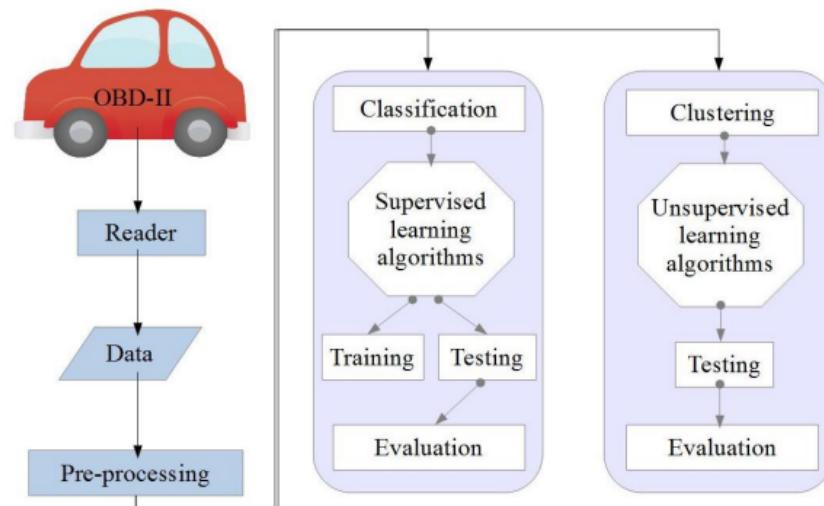


Figura 1.5: Esquema seguido para determinar los hábitos de conducción usando el OBD-II [1].

Por último, uno de los tipos de investigación bastante interesante realizada en los últimos años es la de la generación de perfiles de conducción y de consumo. En el artículo realizado por Ameen *et al.* [18] se define un sistema de clasificación del comportamiento del conductor al volante (que es además el que se propone usar en este proyecto) el cual combina los datos recibidos por el OBD-II y del GPS (figura 1.6):

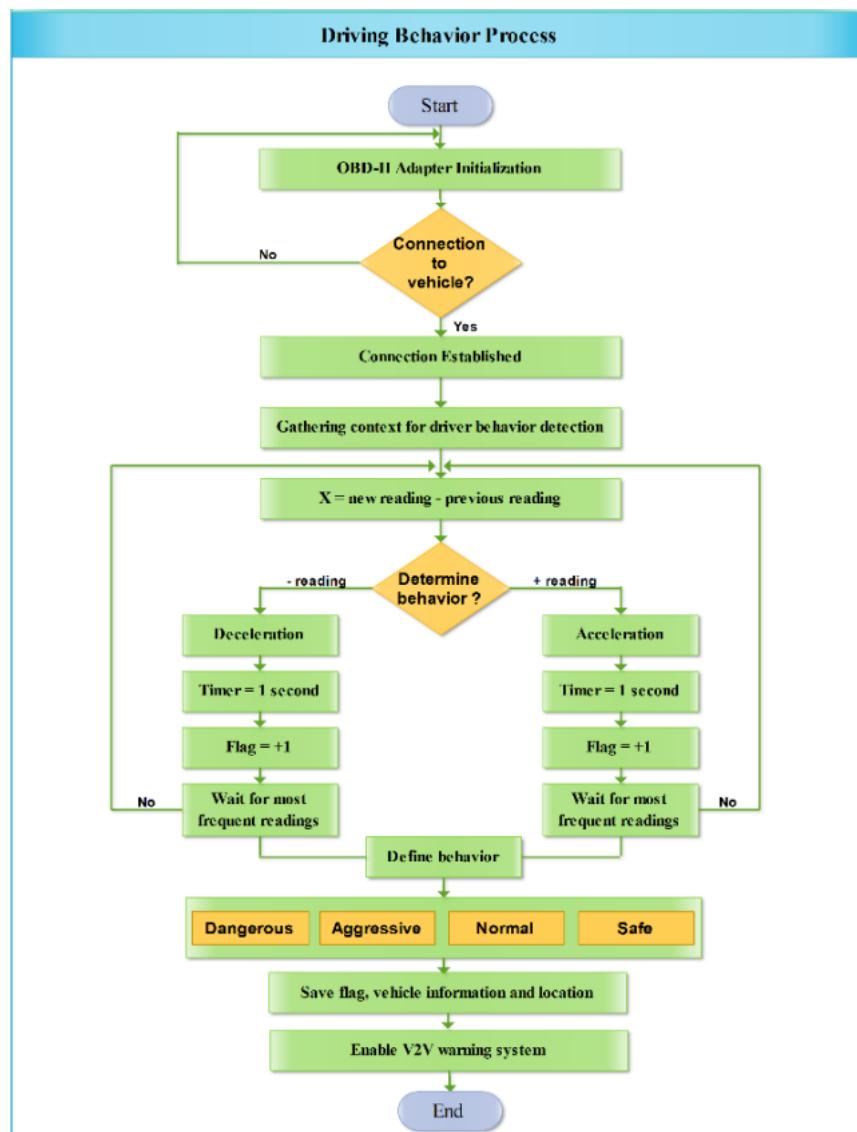


Figura 1.6: Flujo de análisis para determinar el comportamiento al volante de un conductor [18].

Al final, el estudio concluía con los siguientes perfiles de conducción:

- **Peligroso**, para una aceleración en general superior a 7 m/s^2 .
- **Agresivo**, para una aceleración entre $[4 \text{ m/s}^2, 7 \text{ m/s}^2]$.
- **Normal**, para una aceleración entre $[2 \text{ m/s}^2, 4 \text{ m/s}^2]$.
- **Seguro**, para una aceleración entre $[0 \text{ m/s}^2, 2 \text{ m/s}^2]$.

1.2. Objetivos del desarrollo del proyecto

Como se ha podido apreciar, existen multitud de aplicaciones relacionadas directa o indirectamente con el OBD-II, en parte por la longevidad del conector en el mercado.

Sin embargo, todas o la gran mayoría de aplicaciones están destinadas a los profesionales del sector, e incluso se ha aprovechado este conector para dificultar el acceso a los datos del vehículo, habiendo de ir a un taller oficial para que puedan hacer las reparaciones pertinentes.

Por otra parte, el parque de vehículos español es cada año más viejo debido a diversos factores que no se van a analizar en este trabajo. Esto implica que cada año más y más vehículos pierden el soporte por parte del fabricante y se vuelven cada vez más costosos y complejos de mantener.

Para los no eruditos, el mundo del automóvil es el gran desconocido en donde una serie de personas cualificadas se encargan del mantenimiento y correcto funcionamiento del mecanismo que nos transporta por el mundo. Si bien es cierto que es necesaria esta figura, hay una serie de buenas prácticas y actuaciones que pueden prevenir tener que ir al mecánico de forma recurrente. Solo hace falta acceso a la información de manera accesible.

Es por esto que nace VIMS, un proyecto que pretende desarrollar un sistema completo que consta de varias partes: un dispositivo embebido, un servidor y el usuario en sí (figura 1.7):

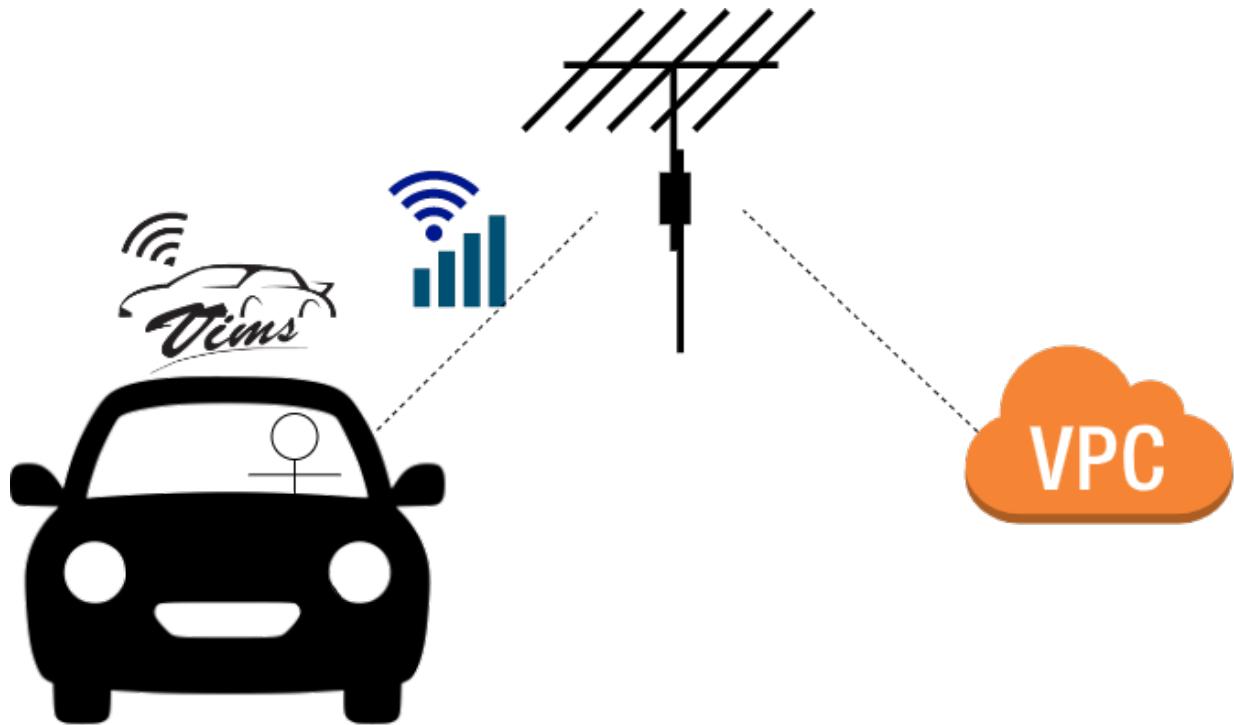


Figura 1.7: Esquema general que modela el modo de funcionamiento del sistema.

La idea fundamental detrás de este proyecto es la de devolverle a los usuarios el control

sobre su vehículo, ser conscientes de cómo funciona o, al menos, entender mejor qué pueden hacer para mejorar tanto su estilo de conducción como la seguridad al volante. De esta forma, con el dispositivo se espera también prevenir riesgos ya que los propietarios y usuarios de los vehículos estarán informados en todo momento de qué error pueda tener.

Como se prevé que este dispositivo sea utilizado por una gran variedad de usuarios es crucial que sea accesible en términos de sencillez de manejo, entendimiento y uso: evitar datos excesivamente técnicos, presentar la información más relevante primero, etc.

Para ello, se hará uso de plataformas en la nube para gestionar, almacenar y presentar la información al usuario y se contará con una aplicación móvil que permita un fácil acceso a los datos del vehículo, tanto históricos como generados en el momento.

Al igual que otros trabajos previamente realizados, el proyecto se realiza sobre las filosofías del código libre y del *hardware* libre, que se traduce en que todos los recursos (tanto físicos como *software*) estarán disponibles enteramente para cualquier persona interesada en ver cómo funciona, replicar el proyecto por su cuenta y contar con plena potestad para mejorarlo, redistribuirlo y trabajar con él (siempre bajo un prisma de reconocimiento al autor original del trabajo regido por *copyright*). Para ello, se ha decidido hacer uso de la licencia MIT².

1.3. Metodología

El proyecto que se pretende desarrollar es un proyecto de ingeniería. Esto se traduce en que la metodología y la forma de trabajo son pilares fundamentales en el desarrollo del mismo.

El primer paso realizado fue el de recopilar recursos e información de qué necesitaban específicamente los usuarios. Para ello, se elaboró un cuestionario en donde de forma general se preguntaba a conductores y no conductores qué querían tener en su vehículo. Los primeros sirvieron como fuente de información principal, debido a sus características *per se*; los segundos para aumentar la entropía de los datos obtenidos y usando sus respuestas como grupo de control para validar la de los primeros. El primer análisis realizado se detalla en la sección 3.3.1, de la especificación de requisitos. Posteriormente, en el punto ?? se analiza en mayor profundidad los datos obtenidos y se extraerán conclusiones.

A continuación, se realizó un estudio sobre qué plataformas y dispositivos están accesibles de forma global para la generación y transmisión de datos. Esta fase se centró principalmente en “descubrir” variantes del modelo ESP32 que incluyesen ciertas antenas para permitir una mayor conectividad. Tras valorar diversas opciones, se decidió usar el LILYGO T-SIM7000G ESP32 que incluye soporte de forma nativa para tarjetas microSD, antena LTE, alimentación externa por batería, antena GPS, WiFi y Bluetooth.

Una vez se decidió que dispositivo físico se iba a utilizar, se comenzó con el desarrollo de los

²Se pueden obtener más detalles sobre la licencia MIT en la siguiente URL: <https://choosealicense.com/licenses/mit/>

distintos diagramas que modelan el sistema, tanto lógicos como de diseño. Esta parte fue crucial para asentar las bases de lo que será el proyecto y ha permitido seguir el avance del mismo.

Por último, se realizó el serigrafiado de la placa y se comenzó la implementación física de los diseños realizados. Por diversos contratiempos que han surgido, el proyecto se ha planteado finalmente como proyecto de especificación y diseño, con una batería de ensayos específicos reales que facilitarán la construcción que se va a realizar posteriormente.

2 Estructura del proyecto

El desarrollo del sistema VIMS es un proceso multidisciplinar en el que se deben desarrollar varias áreas de conocimiento. Este proyecto se ha postulado como un desarrollo integral de ingeniería y es por eso por lo que está dividido en varios bloques que conforman un factor clave en el desarrollo del mismo.

En este proyecto existen diversos bloques diferenciados: un estudio de mercado y de las características de los usuarios, un estudio matemático asociado a la lectura de valores y adecuación del *hardware*, el proceso de diseño *hardware* en sí, el diseño *software* del sistema y el análisis de planificación del mismo.

- El estudio de mercado pretende averiguar y formalizar las necesidades de los conductores y usuarios de la vía. Es la primera aproximación y facilita la delimitación del producto y, sobre todo, ofrecerle al usuario final algo de utilidad y que pueda necesitar.
- El estudio matemático se encarga de investigar la “traducción” de los valores recibidos por el vehículo (según los datos asociados al estudio de mercado realizado con anterioridad).
- El diseño *software* modela principalmente cómo se va a estructurar el sistema y cómo debe comportarse ante los distintos eventos que puede recibir. Esta fase conlleva realizar diagramas lógicos y de diseño del sistema en su conjunto.
- El diseño *hardware* conlleva tanto el estudio de los componentes del sistema así como de las restricciones físicas del mismo. Además, en esta sección también se introduce el diseño 3D de la caja que alojará la placa.
- El análisis de planificabilidad complementa el diseño *software* y estudia si el sistema es planificable. En los requisitos no se define VIMS como un sistema en tiempo real, pero la cantidad de componentes que contiene y las acciones que tiene que realizar requieren del uso de subrutinas y de una planificación previa para asegurar un correcto funcionamiento del mismo.

Es importante detallar que pese a que el sistema se compone de varios componentes, son dos los principales que lo caracterizan:

1. La placa, VIMS, que va embebida en los vehículos del sistema. Se encarga de toda la lectura, adaptación y emisión de datos. Además, cuenta con soporte para poder realizar una transmisión de la información a un dispositivo asociado mediante redes PAN.
2. El servidor *cloud*, el “cerebro” encargado de recibir las tramas, los datos y la información

relativa a las placas VIMS, los dispositivos de usuario y demás componentes. Además, tiene la responsabilidad de ofrecer a los usuarios una GUI, generar información relevante a partir de los datos (como estadísticas), gestionar las suscripciones y enviar periódicamente la información al usuario.

2.1. Estudio de mercado

Para el desarrollo de este proyecto, se hizo un estudio de mercado tanto de los consumidores como de sus características, además de una evaluación exhaustiva de qué les gustaría tener en su vehículo.

Este proyecto pretende en un futuro salir a mercado y suplir características que los usuarios echan en falta en sus correspondientes medios de transporte. Como en principio funciona con cualquier vehículo que cuente con OBD-II, las respuestas no se han limitado a aquellos conductores que condujeseen turismos sino cualquier tipo de automóvil: motocicleta, camión, etc.

Es importante destacar que el estudio tiene varios sesgos que han restringido y delimitado las respuestas que se han registrado:

1. Se ha realizado un cuestionario usando Google Forms, una plataforma de Google que permite preparar una serie de preguntas y respuestas y aplicar ciertos filtros sobre ellas. Por ejemplo, para aquellos que dijeron ser conductores, se hicieron preguntas diferentes frente a quienes no lo fueran.

Esto permite obtener datos más fidedignos y acotados según la población que respondiera. Sin embargo, tiene una limitación implícita: restringe el acceso a aquellos con conocimientos “suficientes” acerca de la plataforma. Pese a que el producto pretende ser lo más accesible posible, no hay que olvidar que este tipo de tecnologías permanecen desconocidas para una gran parte de la población con escasos conocimientos acerca de Internet o de las nuevas tecnologías. Se comentará más adelante, pero esto se ve reflejado principalmente en la edad media de quienes respondieron el cuestionario.

Por otra parte, al ser un cuestionario aparece otra limitación implícita y es la validación y verificación de las respuestas: se confía en la buena fe de los participantes y en la calidad de sus respuestas. Igualmente, se desarrolló el cuestionario junto con una psicóloga que ayudó a definir preguntas cerradas e incluir ciertas respuestas de control. Por otra parte, se usó el propio mecanismo que ofrece este servicio de Google para ordenar aleatoriamente las respuestas del cuestionario (y que eso sirviese también como control).

2. Los encuestados fueron contactados principalmente por la red social Twitter, mediante la difusión con “me gusta” y “retweet”. También se usaron otros medios de comunicación (como el correo UPM y Telegram/WhatsApp), pero el mayoritario fue el ya mencionado. Nuevamente, esto introduce un sesgo tanto por edad como por accesibilidad.

3. Junto con la encuesta, se realizó un sorteo entre aquellos que respondiesen a la misma de un cheque regalo de Amazon valorado en 20 €. Si bien este incentivo pudiera resultar interesante, puede resultar también en un nuevo sesgo en donde personas que o bien no compran en Amazon o bien no sepan lo que es no quisieran hacer el cuestionario.

Además, se plantea la casuística en que ciertas personas quisieran responder al cuestionario solo por el cheque de Amazon, sin importar la calidad de las respuestas, “ensuciando” los resultados obtenidos y quitándole credibilidad al cuestionario en sí.

Esto se analizará posteriormente junto con las respuestas recibidas y las preguntas de control introducidas.

4. El cuestionario, al ser relativamente exhaustivo, pudo echar para atrás a muchos posibles encuestados ya que se estima que el tiempo medio para realizarlo es del orden de 10/15 minutos. A parte del daño evidente de tener menos muestra con la que trabajar, este posible suceso reduciría también la variedad de la población y la calidad del estudio realizado.

La primera pregunta que se realizó a los encuestados era si eran conductores o no. Los datos revelan que el 72,7% (16) de los encuestados son conductores, mientras que el 27,3% (6) restantes no, del total que fueron 22 (figura 2.1):

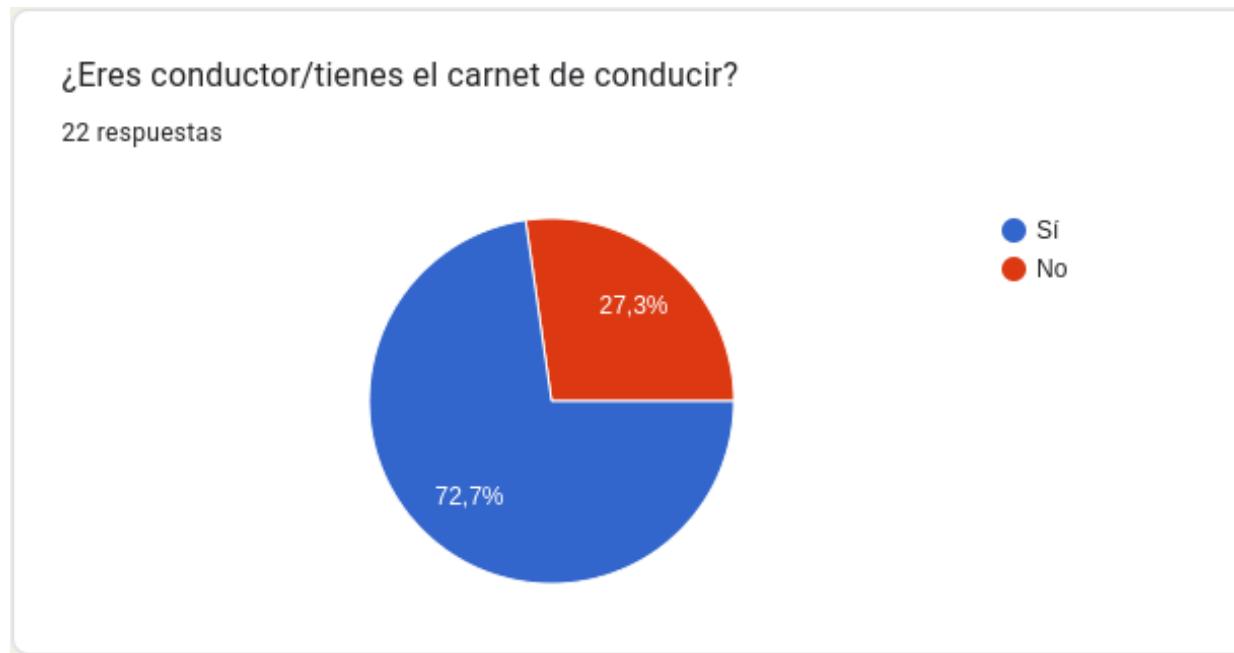


Figura 2.1: Gráfico de tarta que muestra quiénes de los encuestados son conductores (72,7%) y quiénes no (27,3%).

Sobre aquellos que dijeron ser conductores, se preguntó acerca de los años que llevaban con carné de conducir, así como los tipos de carné de conducir que tenían los encuestados.

Se vio que un 68,8 % tenía el carnet desde hace 3 años o más (11 encuestados en particular); un 12,5 % tenía el carnet desde hace solo un año (2 encuestados) y el restante, en su mayoría, tenía el carnet desde hace menos de un año. Esto se ve reflejado en el histograma 2.2:

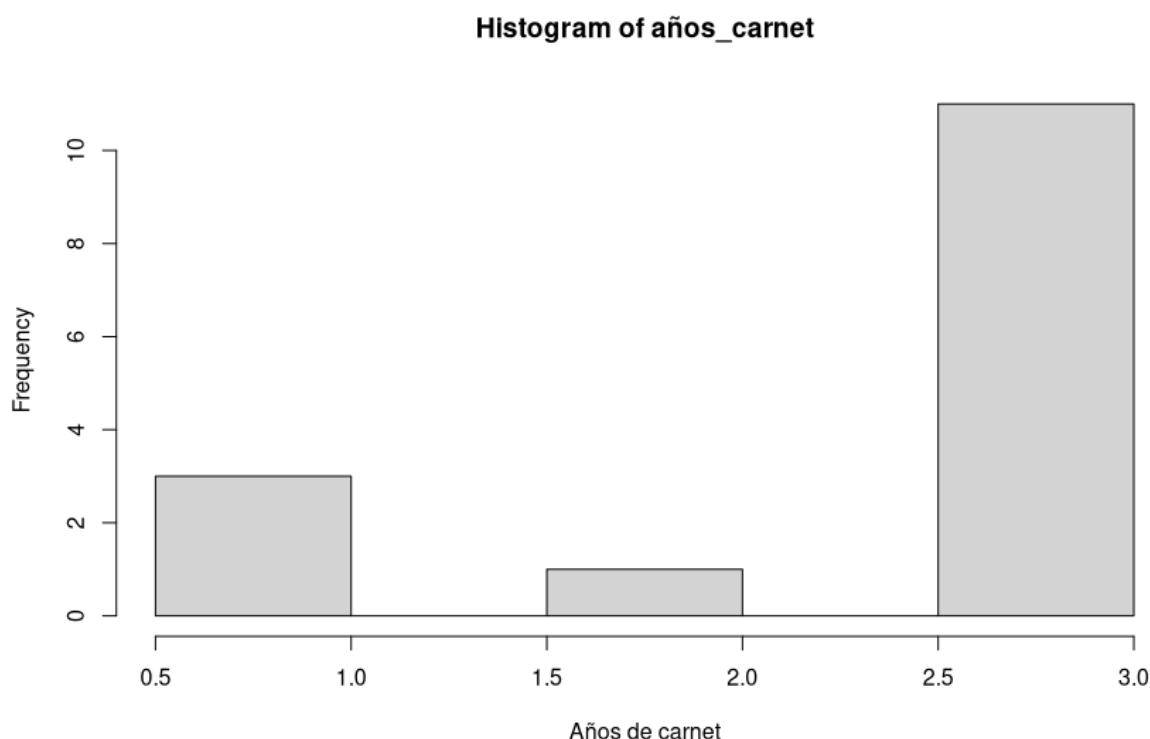


Figura 2.2: Histograma que muestra los años de carnet de los encuestados.

Con respecto a los tipos de carnet, el 100 % de los encuestados (que dijeron ser conductores) tiene el carnet tipo B. Se vio además que el 18,8 % tiene además los carnets relativos a las motocicletas (muy posiblemente, los encuestados tienen el carnet tipo “A” que les habilita automáticamente para aquellos de menor nivel, como el AM, A1 y A2); y únicamente un encuestado tiene el carnet tipo C, que permite conducir camiones.

Una restricción que se comentó con anterioridad era la edad media de los participantes. Esta pregunta se realizó por dos motivos:

1. Definir estadísticamente la edad media de la población para una posterior evaluación de su longevidad y experiencia tanto en la conducción como en la posible compra-venta de vehículos.
2. Diferenciar, definir y clasificar los encuestados por grupos de edad y descubrir posibles sesgos y restricciones en las respuestas para un posterior análisis sobre la causa de dichos sesgos y restricciones.

Es necesario decir que solo se preguntó por la edad a aquellas personas que respondieron afirmativamente a ser conductores. Esto se hizo así debido a que sus respuestas han conformado el dato más relativo a la hora de realizar la investigación, y se hace así también estadísticamente.

Se tiene pues que:

$$\left\{ \begin{array}{l} X_{min} = 19 \\ \bar{X} = 30 \\ \text{Mediana}(X) = 26,5 \\ S(X) \approx \pm 10,564 \\ X_{max} = 55 \end{array} \right. \quad (2.1)$$

Se construyó además un gráfico de tarta (figura 2.3) que muestra cómo quedan distribuidas las edades de los participantes:

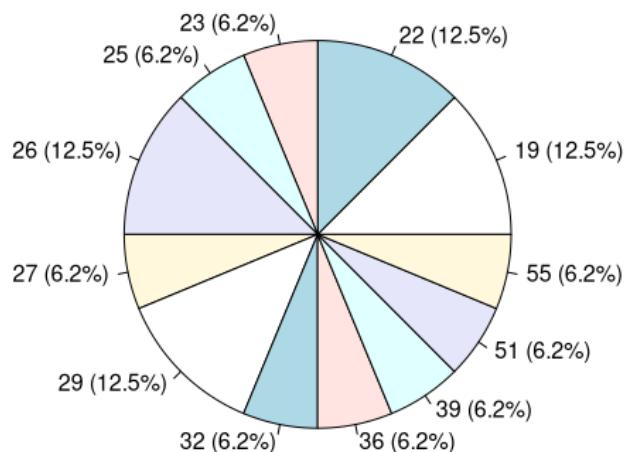


Figura 2.3: Gráfico de tarta que muestra la distribución de edades (valor previo al paréntesis) y su frecuencia en porcentaje.

Como se puede ver en la ecuación 2.1, la distancia entre los valores mínimo y máximo es de 36 puntos. Sin embargo, los valores de la media (\bar{X}) y la mediana ($\text{Mediana}(X)$) muestran que

la distribución está bastante centrada en torno a la media, con una desviación estándar ($S(X)$) de $\approx \pm 10,564$.

Es interesante notar los tres grandes bloques presentes en la figura 2.3 que evidencian lo que se venía indicando anteriormente: en proporción, a la encuesta ha accedido más población joven que adulta. Mismamente, solo el porcentaje de personas encuestadas con edad por debajo de los 26 años es del 49,9 %, casi la mitad de los encuestados. Ampliando dicho margen hasta los 36 años, el porcentaje crece hasta el 81 %.

Este dato se puede ver también reflejado en la distribución de los cuartiles, en donde se tiene que:

$$\begin{cases} Q_1 = 22,75 \\ Q_3 = 33,00 \end{cases} \quad (2.2)$$

Como se puede apreciar en la ecuación 2.2, la distribución de cuartiles está en un rango de edad por debajo de los 33 años para el 75 % de la muestra, indicativo nuevamente de una población encuestada joven.

Destacan dos datos sobre los demás en donde los encuestados tienen 51 y 55 años respectivamente. De este caso en particular se hablará posteriormente, pero cabe destacar que sus respuestas fueron las más pobres en cuanto a contenido (sobre todo en aquellas que sirvieron de control), seguramente debido al formato del cuestionario, fatiga tras responder las secciones anteriores, etc.

Una vez se indagó acerca de la información que identifica a la muestra, se preguntó directamente por el vehículo con el que contaban así como las características del mismo. Para esta sección, se han dividido las preguntas en tres categorías: *básico*, *habitual* y *premium*. Dichas categorías se crean según el porcentaje de uso habitual mundial de las características que se enumeran en la tabla 2.1:

Básico	Habitual	Premium
Control de crucero	Pantalla táctil	Asistente virtual
Limitador de velocidad	GPS	Aplicación móvil
Cámara de visión trasera	Detección de ángulos muertos	Cámara <i>on-board</i>
Botón de arranque	Android Auto	

Cuadro 2.1: Tabla de distribución de las características de los vehículos, preguntado en el cuestionario.

Tras el cuestionario, las frecuencias obtenidas fueron:

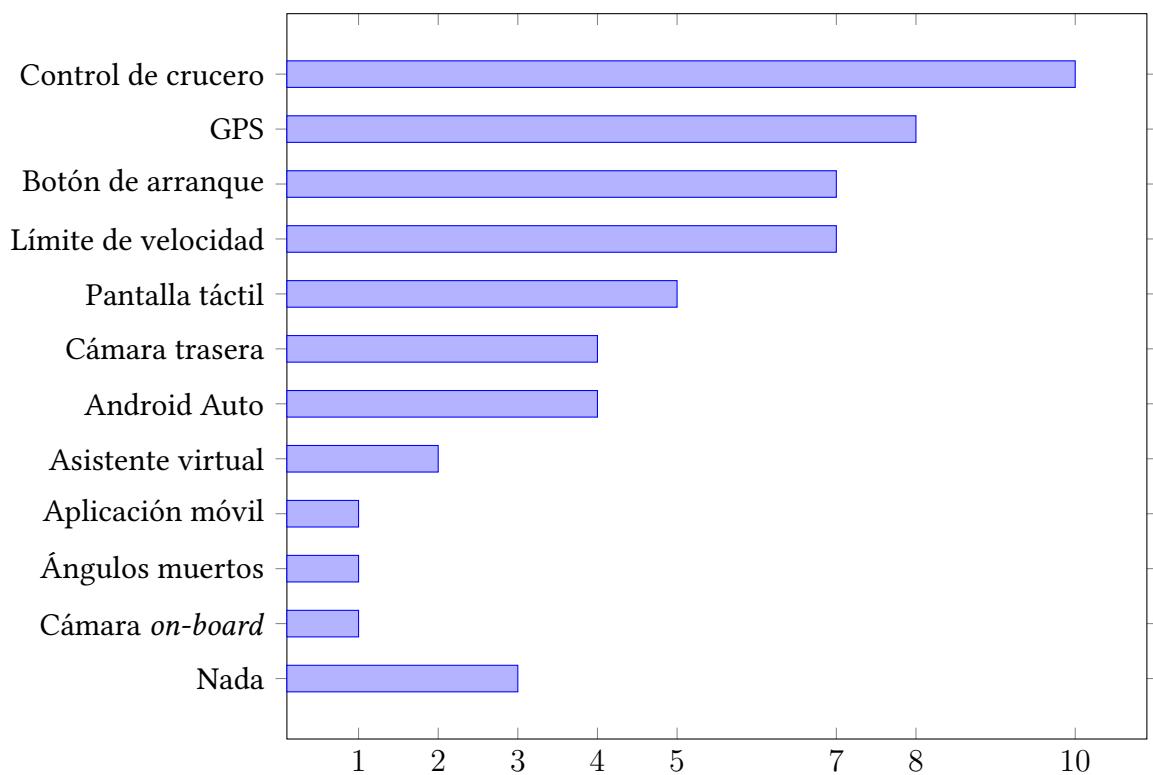


Figura 2.4: Distribución de las características de los vehículos de los encuestados.

Como se puede observar en la imagen 2.4, una gran mayoría de vehículos tienen lo que se consideran “características básicas”, según la tabla 2.1.

Es interesante destacar que cada vez más vehículos dentro del parque español cuenta con diversos elementos de seguridad y confort, como pantalla táctil y navegador GPS, incluso por encima de ciertas características básicas. Si bien esto es positivo para el sector, puede suponer un problema a la hora de sacar a la venta este proyecto ya que estaría, según los datos recogidos anteriormente, mayoritariamente orientado a vehículos “antiguos” o no tan modernos.

Es por ello por lo que se realizó también una encuesta sobre edad aproximada de los vehículos. De la muestra total de 16 conductores, se obtuvieron los siguientes datos sobre la edad del parque de vehículos encuestados (ecuación 2.3):

$$\left\{ \begin{array}{l} X_{min} = 2 \\ \bar{X} = 10,12 \\ \text{Mediana}(X) = 10,00 \\ S(X) \approx \pm 6,839 \\ X_{max} = 21 \end{array} \right. \quad (2.3)$$

En este caso, es muy interesante ver la variedad en las respuestas: entre el mínimo y el máximo

hay casi 20 años; sin embargo, la media y la mediana son casi las mismas, centradas en torno a los 10 años, con una desviación estándar de aproximadamente $\pm 6,839$ años.

Para esta estadística es fundamental observar los cuartiles, ya que van a reflejar datos muy interesantes que darán ciertas pistas sobre las especificaciones de los vehículos, que se habló anteriormente en la figura 2.4:

$$\begin{cases} Q_1 = 3,5 \\ Q_3 = 14,5 \end{cases} \quad (2.4)$$

Como se puede observar en los cuartiles 2.4, el 25 % de la muestra se localiza en los primeros $[0, 3,5]$ años de antigüedad, y el 75 % de la muestra es abarcada para todos los vehículos de hasta 14,5 años de antigüedad.

Esta información es muy relevante ya que gran parte de la muestra se corresponde con vehículos bastante nuevos, y eso explica en parte el porqué de los resultados obtenidos en la encuesta, representada por la figura 2.4. Cabe destacar que la moda de la distribución es:

$$M(X) = 2 \quad (2.5)$$

con 4 de los 16 encuestados teniendo un vehículo de 2 años de edad, lo que se traduce en un parque muy moderno. Sin embargo, esto también ha servido para desvelar nuevas necesidades que no existían antes y que los nuevos vehículos no son capaces de suplir.

Con la encuesta “personal” ya completada, se dio paso en el cuestionario a una evaluación de características que VIMS debería incluir. La primera pregunta era un listado de 20 características que pueden ser medidas mediante el OBD-II y un sistema de puntuaciones escalado del 1 al 20. Cuanto más pequeño fuese el valor, más importante se considera que es esa característica para el encuestado. Las preguntas realizadas se muestran en la tabla 2.2:

Visualización en tiempo real	Información detallada de errores	Velocímetro	Cuentarrevoluciones
Marcha actual ($1^a, 2^a, \dots$)	Temperatura del aceite	Presión del aceite	Temperatura exterior
Intensidad del acelerador (%)	Consumo actual	Presión de las ruedas	Presión de los inyectores
Nivel de combustible	Distancia recorrida	Nivel de batería	Nivel de carga en valor absoluto del motor
Presión atmosférica	Temperatura de la toma de aire	Temperatura del refrigerante	Temperatura del motor

Cuadro 2.2: Opciones ofrecidas a los encuestados. Se han escogido diversas opciones que se encuentran entre los datos habituales generados por un vehículo.

Es importante destacar que la tabla anterior contiene valores puramente técnicos, en donde solo aquellos con un conocimiento más específico entenderán y sabrán interpretar; y otros más genéricos que en principio todos los conductores conocen (como velocidad actual, temperatura del aceite, ...).

Las respuestas a las preguntas anteriores se han estructurado en dos tablas: una primera que representa toda la población en general, y una segunda en donde solo están las respuestas de los conductores (tabla 2.3 y tabla 2.4):

Puntuación	Opción
1	Velocímetro
1	Nivel de combustible
4	Marcha actual
5	Distancia recorrida
5	Nivel de batería
6	Cuentarrevoluciones
8	Intensidad del acelerador (%)
8	Presión de las ruedas
10	Temperatura del aceite
10	Nivel de carga en valor absoluto del motor
12	Visionado en tiempo real
12	Consumo actual
12	Temperatura del motor
14	Presión del aceite
14	Temperatura del refrigerante
16	Presión atmosférica
17	Temperatura exterior
19	Información detallada de errores
19	Presión de los inyectores
20	Temperatura de la toma de aire

Cuadro 2.3: Puntuaciones obtenidas de forma general, por la población al completo (conductores y no conductores).

Puntuación	Opción
1	Velocímetro
1	Nivel de combustible
3	Temperatura del motor
4	Temperatura del aceite
4	Nivel de batería
5	Distancia recorrida
6	Cuentarrevoluciones
7	Presión de los inyectores
8	Presión de las ruedas
10	Presión del aceite
11	Marcha actual
11	Presión atmosférica
14	Temperatura del refrigerante
15	Temperatura exterior
17	Intensidad del acelerador (%)
18	Información detallada de errores
18	Consumo actual
19	Temperatura de la toma de aire
20	Visionado en tiempo real
20	Nivel de carga absoluta del motor

Cuadro 2.4: Puntuaciones obtenidas de forma general, por la población (excluidos los no conductores).

De los datos anteriores es interesante ver cómo ciertos campos cambian completamente de ubicación según ha respondido la población general o solo los conductores, como por ejemplo

la temperatura del motor. Otras, por el contrario, es interesante que se mantengan en la misma posición según las tablas, como el velocímetro. Este caso en particular resulta especialmente interesante ya que es un elemento que tienen todos los vehículos. Sin embargo, tanto la población en general como la comunidad de conductores lo ha puntuado como el elemento de más interés sobre el que obtener datos.

De las características puntuadas en las tablas 2.3 y 2.4 se extraerán una serie de componentes que se detallan en mayor profundidad en el punto 3.3.1, la obtención de los requisitos de usuario.

A continuación, se realizaron una serie de preguntas de desarrollo, en donde el encuestado tenía libertad para poder exponer ideas y opiniones sobre ciertas características del coche. Esta sección sirvió en parte como sección de control y también ha permitido definir con mayor certeza ciertas restricciones del cuestionario, como su accesibilidad. También ha permitido descubrir el porqué de ciertas puntuaciones especialmente elevadas.

La primera pregunta que se realizó fue: “**¿Qué te gustaría poder medir de tu coche?**”. Ante dicha pregunta, hay ciertas ideas interesantes:

- El velocímetro en digital, que explica el porqué tiene una puntuación tan alta.
- Información relativa al estado del coche: nivel de aceite, presión de las ruedas, líquido de frenos, ...
- Nivel de gasto de combustible con respecto al histórico de otros meses.
- Consumo de combustible actual y estimación de kilómetros.
- Información referente al estado mecánico del vehículo que pueda ayudar a prevenir averías.
- Ubicación de aparcamiento.
- Vida útil de la batería del motor eléctrico.
- Recorrido seguido durante un trayecto.

Como se puede apreciar, gran parte de las respuestas van orientadas a prevenir fallos mecánicos, a estimaciones de combustible y generación de información de los viajes (la enumeración anterior es un resumen de las respuestas totales – hay repeticiones entre ellas). Es interesante que ciertas respuestas ya se habían contemplado en la primera etapa de diseño del proyecto, y reafirman lo que se busca desde el comienzo: que el usuario recupere el control sobre su vehículo.

A continuación, se realiza una segunda pregunta muy parecida a la primera que se usó como control (**¿Qué información echas en falta en tu uso habitual del coche?**). Esta pregunta se ha ignorado ya que se utilizó principalmente para verificar la coherencia entre las respuestas. Algunos usuarios aportaron algunas ideas nuevas de elementos que querrían medir, otros copiaron la respuesta anterior (que se consideró válido dada la índole de la pregunta) y otros hicieron referencia a características específicas de la pregunta anterior.

Otra pregunta que se hizo fue “**Si tuvieras que añadirle una característica al coche, ¿cuál sería?**”, en donde el usuario se podía poner “imaginativo” y comentar qué querría que tuviese su vehículo. De las respuestas, las hubieron muy variadas y distintas, y solo se muestran aquí algunas que podrían ser implementadas con VIMS:

- Presión de las ruedas.
- Aviso de errores.
- Desgaste de las ruedas.
- Calidad del aire del habitáculo.
- Emisiones de CO_2 y advertencia ante fuertes temperaturas.

Finalmente, se preguntó por ciertas acciones que resultarían de interés el automatizarlas, también sin restricción en las respuestas. Algunas respuestas fueron:

- Reset del odómetro.
- Aviso de mantenimientos recomendados según kilometraje – ruedas, aceite, frenos, etc.
- Generación de estadísticas del viaje para trayectos de más de 30 minutos según las características mencionadas en las preguntas anteriores.

Estas respuestas fueron muy útiles para generar y definir los requisitos de usuario: a fin de cuentas, el usuario es el *stakeholder* principal y satisfacer sus necesidades debe ser la principal prioridad (y desde aquí, muchas gracias a todos ellos por responder). Se detalla más de qué se implementa finalmente en el punto 3.3.1.

Como se ha mencionado anteriormente, el cuestionario presenta de partida ciertas restricciones para aquellos que quieran participar. Por una parte, al ser un cuestionario puramente *online*, el acceso ya está restringido de base a aquellos con conocimiento suficiente o un uso hábil de la tecnología (esto se ha podido ver en la ecuación 2.1 y en la imagen 2.3 con respecto a la medición de la edad). Por otro lado, otra restricción existente es el conocimiento de los participantes con respecto al tema que se preguntaba.

Para evaluar las restricciones, se introdujo una última pregunta en la que se solicitaba a los encuestados añadir alguna sugerencia o comentario que tuviesen con respecto al cuestionario. Además, se evaluó el lenguaje utilizado para responder las preguntas y se realizó un análisis sobre dichas respuestas.

Por una parte, se indicó que responder y puntuar las características no resultó cómodo, sobre todo en dispositivos móviles donde la pantalla es mucho más pequeña. Se usó el formato matricial ofrecido por Google Forms, pero el problema resultó de que la matriz fuese 20×20 , excesivamente grande en según qué dispositivo se respondiese al cuestionario.

Otra sugerencia que se repitió con más o menos frecuencia fue la de la accesibilidad y fluidez del producto final. Un encuestado con un vehículo relativamente nuevo mencionaba lo molesto

que le resulta interactuar con la pantalla de su vehículo y notar retardo en las acciones, que se quedase bloqueado abriendo aplicaciones, etc., lo cual le transmitía la sensación de que contaba con un vehículo antiguo y de mala calidad.

Siguiendo la misma línea, otro usuario mencionaba que querría que el producto final fuese simple y fácil de entender a primera vista, evitando el exceso de datos y la complejidad de la interfaz de usuario final.

Por otro lado, finalizando con lo que se estudió al inicio de este apartado, se estudió la “calidad” de las respuestas, en especial en los sujetos cuya edad era bastante superior a la media. Es notorio que dichos sujetos, en las preguntas de desarrollo, indicasen en todas las ocasiones “Ns/nc” (*No Sabe/No Contesta*). Tras evaluarlo con cierto detenimiento (teniendo en cuenta la buena fe de los participantes y sabiendo quiénes son), se deduce que el cuestionario resultó excesivamente largo y complejo para los que lo respondieron, en especial el sector más longevo de la población. Esto pudo dificultar el responder al mismo debido a la fatiga, el tiempo que llevaba o lo técnico que pudiera resultar el lenguaje, afectando directamente a la calidad de las respuestas.

Se vio cómo además ciertos usuarios tuvieron dificultad al responder al cuestionario debido a los tecnicismos o palabras propias del sector que pudieran usarse. Un ejemplo es la palabra “odómetro” (también conocido como cuentakilómetros, aunque nunca se uso ese sinónimo ni se mencionó), que resultó ser ajena para varios encuestados.

Estas restricciones se comentarán más adelante, en el punto 6, sobre cómo abordarlas si se volviese a repetir el estudio, junto con ciertas mejoras en el planteamiento del cuestionario.

2.2. Estudio matemático

El estudio matemático va muy ligado al punto anterior (2.1) ya que se analizan los parámetros de OBD-II y su ecuación matemática para obtener un valor en \mathbb{R} entendible por las personas.

Antes de dar paso a las ecuaciones en sí, es importante entender cómo funciona el conector OBD-II en esta situación. Los datos enviados y recibidos por el coche están siempre codificados en un valor binario, recogido en un vector de cuatro elementos en donde cada elemento tiene un *byte* de tamaño. De esta forma, se define al valor obtenido tras leer el conector OBD-II como:

<i>A</i>								<i>B</i>								<i>C</i>								<i>D</i>							
<i>A</i> ₇	<i>A</i> ₆	<i>A</i> ₅	<i>A</i> ₄	<i>A</i> ₃	<i>A</i> ₂	<i>A</i> ₁	<i>A</i> ₀	<i>B</i> ₇	<i>B</i> ₆	<i>B</i> ₅	<i>B</i> ₄	<i>B</i> ₃	<i>B</i> ₂	<i>B</i> ₁	<i>B</i> ₀	<i>C</i> ₇	<i>C</i> ₆	<i>C</i> ₅	<i>C</i> ₄	<i>C</i> ₃	<i>C</i> ₂	<i>C</i> ₁	<i>C</i> ₀	<i>D</i> ₇	<i>D</i> ₆	<i>D</i> ₅	<i>D</i> ₄	<i>D</i> ₃	<i>D</i> ₂	<i>D</i> ₁	<i>D</i> ₀

Cuadro 2.5: Vector de *bytes* que representa los datos recibidos del conector OBD-II [7].

De esta forma, cuando se escribe *A*₄ se hace referencia al cuarto bit del vector *A*. Los bits están ordenados según MSB, de forma que *A*₇ es el bit más significativo y *A*₀ el menor.

Los datos se obtienen del OBD-II utilizando un lenguaje estándar llamado PID. El PID se codifica como un número entero de 16 bits en donde los 8 primeros bits identifican el servicio/modo

y los 8 restantes la operación a realizar. Actualmente, están registrados los siguientes modos de funcionamiento (tabla 2.6):

Modo (hex)	Descripción
01	Muestra los datos actuales del vehículo
02	Muestra los datos almacenados del cuadro del vehículo
03	Muestra los códigos DTC
04	Elimina los códigos DTC almacenados
05	Resultados del test del sensor de oxígeno (sin CAN)
06	Resultados del test del sensor de oxígeno (con CAN)
07	Muestra los códigos DTC pendientes (eliminados durante el último ciclo de conducción)
08	Operaciones de control sobre los componentes del sistema
09	Petición de información sobre el vehículo
0A	Códigos DTC eliminados

Cuadro 2.6: Lista de modos de funcionamiento del estándar OBD-II [7].

Es importante destacar que no todos los fabricantes tienen por qué soportar todos los modos, y que además ciertos fabricantes pueden definir sus modos propios por encima del valor 09.

Todos los modos definidos anteriormente tienen un conjunto de órdenes de soporte en donde el sistema indica qué PIDs están soportados y cuáles no, por cada 32 PIDs. Por ejemplo, enviar la orden $0x0100$ (*modo 1, PID 0*) devolverá un valor que representará si los siguientes 32 PIDs están soportados. Si el valor fuese, por ejemplo, BE1FA813 se tendría que:

Hexadecimal	B	E	1	F	A	8	1	3
Binario	1 0 1 1	1 1 1 0	0 0 0 1	1 1 1 1	1 0 1 0	1 0 0 0	0 0 0 1	0 0 1 1
¿Soportado?	✓ ✗ ✓ ✓	✓ ✗ ✓ ✗	✗ ✗ ✗ ✗	✓ ✗ ✗ ✗	✓ ✗ ✗ ✗	✓ ✗ ✗ ✗	✓ ✗ ✗ ✗	✓ ✗ ✗ ✗
PID	01 02 03 04	05 06 07 08	09 0A 0B 0C	0D 0E 0F 10	11 12 13 14	15 16 17 18	19 1A 1B 1C	1D 1E 1F 20

Cuadro 2.7: Obtención de los PIDs soportados según el modo [7].

A continuación, se dejan un conjunto de PIDs que se van a implementar en el proyecto así como el código de acceso a ellos y la ecuación que permite obtener el valor real.

Modo 01

Este modo permite acceder a la información en tiempo real del vehículo, según se está en marcha. Los datos a los que se accede son:

PID (hex)	04
Bytes devueltos	1
Descripción	Carga del motor, en porcentaje
Valor mínimo	0 %
Valor máximo	100 %
Fórmula	$\frac{A}{2,55}$

Cuadro 2.8: PID 04 – carga del motor, en %.

PID (hex)	05
Bytes devueltos	1
Descripción	Temperatura del refrigerante del motor
Valor mínimo	-40 °C
Valor máximo	215 °C
Fórmula	$A - 40$

Cuadro 2.9: PID 05 – temperatura del refrigerante del motor, en °C.

PID (hex)	0C
Bytes devueltos	2
Descripción	Velocidad del motor
Valor mínimo	0 RPM
Valor máximo	16383,75 RPM
Fórmula	$\frac{256A + B}{4}$

Cuadro 2.10: PID 0C – velocidad del motor, en RPM.

PID (hex)	0D
Bytes devueltos	1
Descripción	Velocidad del vehículo
Valor mínimo	0 <i>km/h</i>
Valor máximo	255 <i>km/h</i>
Fórmula	A

Cuadro 2.11: PID 0D – velocidad del vehículo, en *km/h*.

PID (hex)	11
Bytes devueltos	1
Descripción	Posición del acelerador
Valor mínimo	0 %
Valor máximo	100 %
Fórmula	$\frac{A}{2,55}$

Cuadro 2.12: PID 11 – posición del acelerador, en %.

PID (hex)	2F
Bytes devueltos	1
Descripción	Nivel del tanque del combustible
Valor mínimo	0 %
Valor máximo	100 %
Fórmula	$\frac{A}{2,55}$

Cuadro 2.13: PID 2F – nivel del tanque del combustible, en %.

PID (hex)	46
Bytes devueltos	1
Descripción	Temperatura ambiente
Valor mínimo	-40 °C
Valor máximo	215 °C
Fórmula	$A - 40$

Cuadro 2.14: PID 46 – temperatura ambiente, en °C.

PID (hex)	5B
Bytes devueltos	1
Descripción	Tiempo restante de la batería híbrida
Valor mínimo	0 %
Valor máximo	100 %
Fórmula	$\frac{A}{2,55}$

Cuadro 2.15: PID 5B – tiempo restante de la batería híbrida, en %.

PID (hex)	5C
Bytes devueltos	1
Descripción	Temperatura del aceite
Valor mínimo	-40 °C
Valor máximo	215 °C
Fórmula	$A - 40$

Cuadro 2.16: PID 5C – temperatura del aceite, en °C.

PID (hex)	5E
Bytes devueltos	2
Descripción	Consumo actual del motor
Valor mínimo	0 L/h
Valor máximo	3212,75 L/h
Fórmula	$\frac{256A + B}{20}$

Cuadro 2.17: PID 5E – consumo actual del motor, en L/h.

PID (hex)	61
Bytes devueltos	1
Descripción	Torque demandado por el conductor
Valor mínimo	-125 %
Valor máximo	130 %
Fórmula	$A - 125$

Cuadro 2.18: PID 61 – torque demandado por el conductor, en %.

PID (hex)	62
Bytes devueltos	1
Descripción	Torque actual del motor
Valor mínimo	-125 %
Valor máximo	130 %
Fórmula	$A - 125$

Cuadro 2.19: PID 62 – torque actual del motor, en %.

PID (hex)	63
Bytes devueltos	2
Descripción	Torque de referencia del motor
Valor mínimo	0 N · M
Valor máximo	65535 N · M
Fórmula	$256A + B$

Cuadro 2.20: PID 63 – torque de referencia del motor, en $N \cdot M$.

PID (hex)	A4
Bytes devueltos	4
Descripción	Marcha actual del vehículo
Valor mínimo	0 ratio
Valor máximo	65535 ratio
Fórmula	$A_1 = 1 \Rightarrow \text{soportado}$ $R = \frac{256C + D}{1000}$

Cuadro 2.21: PID A4 – marcha actual del vehículo, en ratio.

PID (hex)	A6
Bytes devueltos	4
Descripción	Odómetro
Valor mínimo	0 km
Valor máximo	429496729,5 km
Fórmula	$\frac{A(2^{24}) + B(2^{16}) + C(2^8) + D}{10}$

Cuadro 2.22: PID A6 – odómetro, en km.

Modo 03

El modo 03 devuelve los DTC guardados de la sesión actual. Estos códigos de diagnóstico representan los distintos errores que hay en el vehículo, con un conjunto de bytes que los identifican.

Este modo, a diferencia de los otros, no requiere de un parámetro PID sino que solo se envía el servicio como identificador. Una petición al modo 03 devolverá una lista de n elementos en donde cada elemento ocupa 2 bytes (por ende, el tamaño esperable de la trama es $2n$).

Los códigos de error se definen como un conjunto de 5 caracteres de la forma: "U0158". El valor de los caracteres define así:

$A_7 - A_6$	Primer carácter DTC	$A_5 - A_4$	Segundo carácter DTC	$A_3 - A_0$	Tercer carácter DTC
00	P – sistema de propulsión (<i>powertrain</i>)	00	0	0000	0
01	C – chassis	01	1	0001	1
10	B – cuerpo (<i>body</i>)	10	2	0010	2
11	U – comunicaciones (<i>network</i>)	11	3	0011	3
				0100	4
				0101	5
				0110	6
				0111	7
				1000	8
				1001	9
				1010	A
				1011	B
				1100	C
				1101	D
				1110	E
				1111	F

Los caracteres cuarto y quinto se corresponden a los bits $B_7 - B_4$ y $B_3 - B_0$ respectivamente, y siguen la notación hexadecimal (al igual que los bits $A_3 - A_0$).

De esta forma, con el código ya extraído, se necesita mirar en una tabla de valores DTC para saber exactamente a qué error se corresponde. Una web muy interesante es la de “OBD-Codes.com” [19], en donde hay información tanto de códigos DTC estándar como de códigos propietarios. Mirando en la propia web, el código U0158 se correspondería a: “*Lost communication with head-up display*”¹.

¹En la propia web dan muchos detalles e información extendida sobre el error en cuestión, al igual que procedimientos para poder solucionar el problema – <https://www.obd-codes.com/u0158>

Modo 09

El modo 09 devuelve información referente al vehículo en sí, no al estado de los sensores o del propio vehículo. Algunos datos interesantes son:

PID (hex)	02
Bytes devueltos	17
Descripción	VIN
Fórmula	VIN de 17 caracteres ASCII con <i>padding</i> a la izquierda de caracteres nulos (0x00) si hace falta

Cuadro 2.23: PID 02 – VIN.

PID (hex)	0A
Bytes devueltos	20
Descripción	Nombre de la ECU
Fórmula	20 caracteres ASCII con <i>padding</i> a la derecha de caracteres nulos (0x00)

Cuadro 2.24: PID 0A – nombre de la ECU.

2.3. Diseño *software*

El proyecto cuenta con una parte *software* muy importante, ya que está conformado por múltiples sistemas que deben cooperar entre sí. Por una parte, el desarrollo del *software* del proyecto aborda los siguientes aspectos:

- Desarrollo de la aplicación embebida que irá en el dispositivo enganchado al coche en sí. En dicho dispositivo se programará, entre otros, la planificación del sistema según el análisis de planificabilidad llevado a cabo en el punto 4.4, la gestión de los datos y de la información y la transmisión al servidor remoto.
- Desarrollo de la infraestructura en la nube que alojará la lógica de almacenamiento de los datos y de la información generada a partir de ellos para los usuarios de dispositivos VIMS.
- Desarrollo de la aplicación web que servirá de pasarela entre los usuarios y sus datos alojados en el servidor en la nube, mediante una interfaz de usuario que facilitará ciertas tareas de gestión y la observación de los datos.
- Desarrollo de la aplicación móvil que permitirá la conexión directa con la placa VIMS para su configuración inicial y la visualización de la información en el momento.

El sistema sigue el prototipo de una arquitectura cliente–servidor, con uno o varios clientes que se conectan a un servidor en la nube que realiza tareas de gestión de la información, entre otros.

Uno de los requisitos principales de la aplicación (que se vio además en el estudio del mercado) era que esta fuese accesible, simple y fácil de entender. Mediante la definición de tanto la aplicación web como la aplicación móvil se pretende unificar el acceso a la información de los usuarios y sus vehículos. Con ambas aplicaciones, se pretende que de un vistazo se tenga acceso a los últimos viajes, estadísticas, perfiles de conducción y demás.

Además, la arquitectura del servidor permite generar notificaciones personalizadas cuando sucedan ciertos eventos. Por ejemplo, se pueden enviar correos electrónicos con información sobre el último viaje, cuánto ha durado el depósito de gasolina, fecha recomendada para el siguiente mantenimiento, revisión del estado de los neumáticos, etc. De esta manera, se pretende que el usuario no esté directamente pendiente del estado mecánico de su vehículo sino que delegue esa tarea en el sistema VIMS.

Por su lado, la programación del dispositivo VIMS en sí es una parte crítica y fundamental en el desarrollo del proyecto. El dispositivo va a estar en un entorno no controlado con condiciones cambiantes, sobre todo en lo referente a las comunicaciones. Es fundamental que el desarrollo del *software* del dispositivo embebido sea resiliente y esté diseñado para soportar situaciones adversas en lo referente a la transmisión de la información.

Esto se detalla más adelante en los requisitos, y algunas de las características que se necesitan tener en el dispositivo en sí son el almacenamiento provisional de los datos, la retransmisión de la información en caso de fallo, modos de bajo consumo cuando el vehículo no está en marcha, etc.

2.4. Diseño *hardware*

Los elementos *hardware* hacen referencia directa al dispositivo VIMS que irá embebido en el vehículo y al resto de componentes que son necesarios para un correcto funcionamiento del mismo. En esta sección también se incluye el diseño 3D de la caja contenedora del dispositivo.

En términos generales, el *hardware* en que se descompone el proyecto es:

- Dispositivo controlador SoC ESP32, que incluye de fábrica radios WiFi y Bluetooth.
- Dispositivo de control GPS.
- Dispositivo de comunicaciones de red 4G.
- Dispositivo de almacenamiento de datos usando tarjetas microSD.
- Desarrollo de la placa de circuito impreso de control que engloba los elementos necesarios para el correcto desempeño del sistema.

- Diseño de la estructura 3D que albergará el dispositivo final y los componentes necesarios para su funcionamiento.
- Diseño de dispositivo que permita la adaptación del estándar OBD-II a un protocolo entendible por el dispositivo embebido en sí, como el CAN.

En primer lugar, se ha escogido ese SoC por las características que ofrece y el gran soporte (tanto oficial como de la comunidad) que tiene por detrás. El SoC cuenta con un microprocesador Xtensa LX6 de 32 bits con dos núcleos que opera a 240 MHz. Además, cuenta con un co-procesador *Ultra Low Power* (ULP) que permite realizar ciertas operaciones con un consumo extremadamente bajo (del orden de los μA).

Además de lo anterior, tiene directamente integradas las antenas WiFi y Bluetooth (estándares 802.11 b/g/n y v4.2 BR/EDR + BLE respectivamente), controlador PWM integrado, controlador CAN integrado, controlador para tarjetas SD integrado, múltiples *Analog-Digital Converter* (ADC)s de 18 canales y 12 bits, criptografía acelerada por *hardware*, hasta 20 pines GPIO y demás.

Junto con las características físicas anteriores, es importante destacar que el ESP32 tiene soporte para ser programado usando el *framework* de Arduino. Esto facilita la labor de programación y evita tener que entrar en excesivo detalle sobre cómo se puede configurar la placa internamente (por ejemplo, ajuste de registros, esperar a la carga de los condensadores, ...) pero abriendo la posibilidad a ello si hace falta. Además, el SoC tiene soporte nativo para FreeRTOS, que será lo que se use para programar las tareas en tiempo real del controlador.

Por otra parte, el resto de conectores que se necesitan en el dispositivo son muy variados y cada uno tiene sus particularidades. Es por esto que se ha decidido buscar una PCB que ya aúne la lógica de diseño necesaria para que el SoC tenga soporte para ellos. Se estuvo realizando una investigación sobre las distintas alternativas existentes y finalmente se tomó la decisión de usar la placa LILYGO TTGO T-SIM7000G (figura 2.5) que tiene soporte nativo para tarjeta SIM con conexión LTE, antena GPS, tarjeta microSD, batería externa y carga solar.

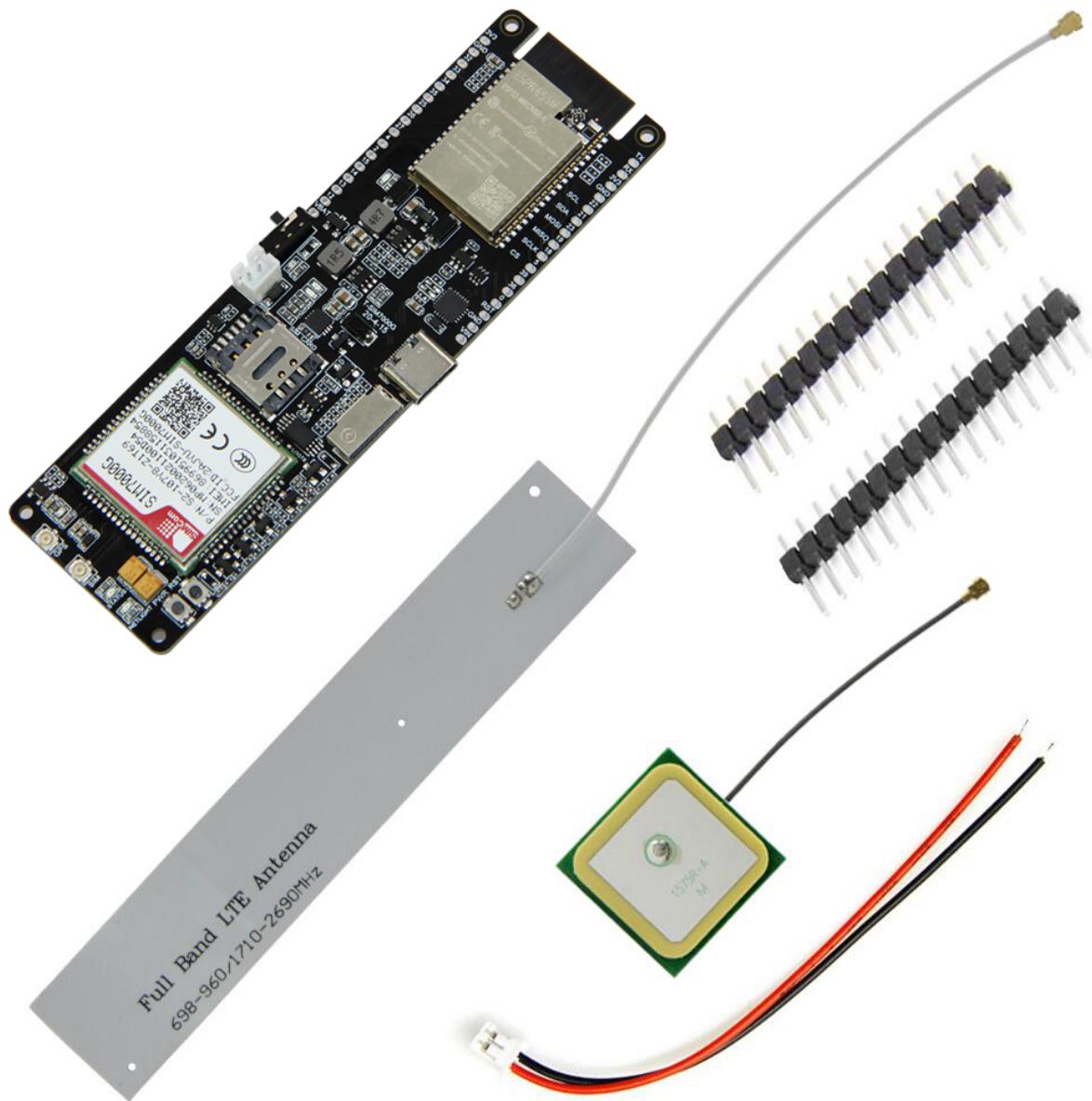


Figura 2.5: Placa de desarrollo LILYGO TTGO T-SIM7000G usada en el proyecto [20].

En lo referente al conexionado con el coche, y en estrecha relación con el diseño 3D, se buscó un conector hembra de OBD-II que permitiese definir el *pinout* hacia la placa en este caso. Se valoraron distintas opciones, y una de las más interesantes resultó la siguiente:



Figura 2.6: Conector OBD-II hembra con posibilidad de definir los pines de salida [21].

También existen otros modelos que vienen ya configurados y que son más simples de implementar, como:



Figura 2.7: Conector OBD-II hembra con cables ya incluidos [22].

Finalmente, para gestionar los sensores y conectores que necesita tener la placa para funcionar correctamente, se diseñará una PCB que definirá la lógica de conexión para el correcto funcionamiento de todo el sistema al conjunto.

2.5. Análisis de planificabilidad

A raíz del apartado anterior y de la cantidad de datos que se pretenden recoger durante el funcionamiento del sistema, una parte fundamental es la planificación de las tareas del mismo.

En principio, VIMS no se define como un sistema en tiempo real. Sin embargo, la envergadura del proyecto y la cantidad de acciones a realizar de forma coordinada y concurrente requiere de una planificación del proyecto a la hora de definir tareas, plazos y períodos.

En un primer análisis, se detectan las siguientes casuísticas que llevarán asociada la tarea pertinente:

1. Lectura de datos desde el conector OBD-II. Esta tarea almacenará la información en crudo del conector en una estructura de datos propia, preparada para que otra tarea los adapte y transforme en información entendible. Además, establecerá una marca temporal para los datos de manera que se sepa exactamente en qué momento se obtuvieron.
2. Conversión de los datos leídos desde el OBD-II. Esta tarea deberá leer y convertir los datos almacenados en crudo por el OBD-II en datos legibles y entendibles por el ser humano.

Se encargará además de definir las unidades de medida asociadas al dato en sí.

3. Transmisión de los datos al servidor en la nube. Esta tarea deberá enviar los datos ya preparados con su unidad correspondiente y su marca temporal usando el protocolo MQTT al servidor en la nube. Según la tecnología de recepción presente en el *backend*, se pueden enviar los datos individualmente a un *endpoint* específico de MQTT (lo cual simplificaría la gestión de la información por parte del servidor) o se deberán enviar todos en conjunto como un JSON al servidor en backend.
4. Persistencia de los datos en la tarjeta microSD. Esta tarea almacenará los datos provisionalmente en la tarjeta microSD si no han podido ser enviados al servidor remoto (por ejemplo, por no disponibilidad de la red). Se deberán serializar los datos junto con su marca temporal para una posterior retransmisión cuando haya disponibilidad de red.
5. Estado de la conectividad de red. Esta tarea se encargará de comprobar periódicamente que la conexión de red está disponible y es accesible desde el dispositivo, simplificando la lógica de comunicación del resto de tareas.
6. Sincronización del reloj con la red. Esta tarea se encargará, mediante el protocolo NTP, de mantener el reloj interno de la placa sincronizado con el tiempo UTC, de forma que todas las marcas temporales asociadas están siempre con la hora adecuada.
7. Ubicación del dispositivo. Esta tarea se encargará de actualizar la ubicación del dispositivo y almacenar los datos para su posterior envío al servidor remoto.
8. Estado eléctrico del sistema. Esta tarea se encargará de comprobar periódicamente la tensión de entrada del sistema para detectar cambios drásticos en dicha tensión (de $\approx \pm 1,1V$) para saber cuándo el sistema debe entrar en modo de bajo consumo.
9. Transmisión en vivo de los datos mediante Bluetooth. Si hay un dispositivo conectado al sistema, se transmitirán los datos generados por el sistema en vivo al dispositivo que esté conectado.
10. Desactivar el sistema y entrar en modo de bajo consumo cuando sea necesario.

Las tareas anteriores deberán planificarse siguiendo un esquema de planificación de tiempo real que contemple frecuencia de activación, retardos relativos producidos por la interdependencia de las tareas, tiempo de cómputo estimado, prioridades según frecuencia, acceso a recursos compartidos y herencia de prioridad por acceso a dichos recursos (prioridad de techo).

3 Especificación de requisitos

3.1. Introducción

En un mundo cada vez más interconectado, hay ciertas tecnologías que se quedan por detrás en unos campos mientras que siguen progresando en otros. Esto se ve directamente reflejado en la industria automovilística en donde los vehículos cada vez cuentan con mayor y mejor tecnología (como cámaras, sensores, actuadores, etc.) pero no es directamente accesible por el usuario: mediante pantallas e interfaces se ofrecen métodos sencillos que facilitan su uso.

VIMS pretende ser un sistema que facilite el acceso a todos los datos que ofrece un vehículo para generar estadísticas, descubrir patrones en la conducción y detectar errores. De esta forma, el conductor tendrá información de primera mano sobre el estado de su vehículo, eficiencia de su conducción así como obtener información en tiempo real complementaria a la ya propiciada por el vehículo.

En este documento se realiza la especificación de requisitos y se acota el alcance del proyecto, en donde se define de forma clara y concisa qué va a hacer el sistema y qué no va a hacer.

3.1.1. Propósito

El propósito de este documento es múltiple: por una parte, el de establecer un punto de partida claro y conciso en el desarrollo del proyecto. Una correcta especificación acompañada de sus correspondientes diagramas permite iniciar el diseño, desarrollo y verificación del proyecto con ideas claras y evaluadas con anterioridad, afrontando los posibles fallos o problemas que puedan surgir y previendo situaciones complejas o errores en el diseño.

Por otra parte, el documento se redacta como especificación de requisitos de usuario, funcionales, no funcionales, restricciones en el desarrollo, requisitos de interfaces externas y de entorno físico del proyecto. Esto sirve como “contrato” con aquello que debe aparecer y existir en una implementación final del proyecto (*requisitos funcionales*), otros requisitos que son importantes pero que, en un momento dado, no aportan funcionalidad al producto (*requisitos no funcionales*), las características de los usuarios que quieran usar el producto y otras restricciones o información relevante que se haya de tener en cuenta a la hora de diseñar, desarrollar y probar el producto antes de darlo por concluido.

Finalmente, este documento se redacta orientado a otros ingenieros que tengan curiosidad o

interés en cómo se ha desarrollado el proyecto, las necesidades que se han de suplir, qué características conforman el producto final o mismamente replicar el proyecto para hacer una implementación propia o añadir alguna característica nueva. Sin embargo, con intención de facilitar la accesibilidad del documento, se buscará ser claro y conciso en la especificación, usando un lenguaje que sea preciso. Por ello, se incluye un apartado de definiciones, acrónimos y abreviaturas que pretenden servir de orientación en el lenguaje técnico que se usará en el documento.

3.1.2. Alcance

El objetivo principal de este Proyecto Fin de Máster (PFM) es el de diseñar un sistema de recolección de métricas orientado a vehículos en el ámbito del IoT (VIMS, de ahora en adelante) que permita, utilizando los conectores estándar del vehículo, generar y almacenar información relevante del vehículo, la conducción y demás factores de interés que se den mientras se interactúa con el vehículo.

El sistema VIMS deberá poder conectarse a Internet desde cualquier punto geográfico¹ usando el automóvil como sistema de alimentación y fuente de información. De esta manera, podrá enviar todo tipo de datos provistos a un servidor en donde se gestionarán, almacenarán y procesarán para una posterior visualización y generación de información.

Además, el sistema deberá poder integrarse con cualquier vehículo del mercado que utilice las conexiones estándar reguladas y que trabaje con tramas e información estandarizada. En otro caso, el sistema no funcionará correctamente y puede comportarse de manera impredecible.

La infraestructura del servidor por su parte deberá poder recibir una gran cantidad de datos (según estimaciones del mundo IoT, se pueden recibir del orden de varios GB diarios [23]) y gestionarlos debidamente. Cada dispositivo emisor se considerará único, por lo que los datos recibidos deberán ser clasificados acorde a quién los emite.

Sin embargo, el sistema no modificará parámetros del vehículo ni realizará modificaciones sobre la configuración del mismo: se limitará a ser un “espía” y no emitirá ningún dato hacia el automóvil.

Finalmente, desde el propio vehículo con un dispositivo externo (como un *smartphone*) se podrán acceder a los datos en tiempo real que ofrece el vehículo mediante una interfaz hacia el sistema VIMS. De esta manera, se podrá saber rápidamente el estado del automóvil y detectar fallos en el mismo.

Así, el producto está orientado para su implantación en cualquier vehículo y que, haciendo uso de las características de conectividad inalámbrica que presentará, pueda adaptarse a nuevos automóviles y nuevas tramas estándar. Su venta está dirigida principalmente a conductores que cuenten con un vehículo con algún conector estándar compatible.

Este documento sigue la taxonomía de especificación descrita en la sección anterior (3.1.1).

¹Dentro de las restricciones y limitaciones físicas y geográficas del entorno.

3.1.3. Estructura del documento

Este documento se estructura de la siguiente manera:

1. En la sección 3.2 se define la descripción general del producto. Allí, se abordan aspectos directamente relacionados con el producto en sí como son definirlo brevemente junto con el contexto de desarrollo y la perspectiva del mismo (a quién va orientado, qué funcionalidades tendrá) (3.2.1), qué características deben tener o se contemplan en los usuarios finales (3.2.2), qué restricciones de diseño, técnicas o regulaciones se aplican sobre el producto (3.2.3) y, finalmente, aquellas suposiciones y dependencias sobre el producto que se asumirán a lo largo del proyecto.
2. En la sección 3.3 se especifican formalmente los requisitos del proyecto. En primer lugar, se definen aquellos requisitos de usuario que especifican las necesidades finales de las personas con respecto del producto (3.3.1). En segundo lugar, se exponen los requisitos funcionales a un nivel más técnico e interno. Estos requisitos definen la base del producto y aquellas funcionalidades que deben existir en el resultado final (3.3.2). Por otra parte, se continúa detallando los requisitos no funcionales. Estos requisitos si bien no añaden funcionalidad al sistema directamente, limitan y acotan el alcance del mismo (3.3.3). Allí se incluye la especificación de las interfaces externas del sistema, es decir, aquellos elementos externos con los que interactuará (3.3.4); restricciones a aplicar durante el desarrollo del producto, en donde se tienen en cuenta estándares, limitaciones físicas, etc (3.3.5); y por último, los requisitos que se imponen del entorno físico del producto, que limitan y definen bajo qué circunstancias debe funcionar y bajo cuáles no (3.3.6).
3. Al final de la especificación, en los anexos, se incluyen los distintos diagramas que complementan la especificación, como casos de uso, bloques, etc. (??).

3.2. Descripción general del producto

A lo largo de esta sección se va a presentar el producto en sí, presentando por una parte la perspectiva del mismo (3.2.1), las características de los usuarios finales (3.2.2), restricciones generales que se aplican sobre el producto (3.2.3) y supuestos y dependencias sobre el proyecto que afectan directamente al desarrollo (3.2.4).

3.2.1. Perspectiva del producto

VIMS se constituirá de un módulo independiente diseñado desde cero aprovechando las tecnologías que ofrecen los vehículos de forma estándar (como el conector OBD).

La intención principal es ofrecer un sistema de recolección de métricas autónomo, automático y lo más simple posible para el usuario, que siga la idea de “conectar y funcionar”, con la

intención de salvar la diferencia tecnológica existente entre vehículos y otorgarle al conductor el control total.

Al igual que se narró en el alcance (3.1.2), el objetivo principal del proyecto es el de desarrollar un sistema autónomo que pueda funcionar con cualquier vehículo del mercado (que cumpla con las condiciones especificadas) y que permita generar, recolectar, procesar y mostrar cientos de datos relativos al coche y su estado actual y estado pasado.

Para ello, se necesitará de una conexión permanente y activa a Internet (siempre y cuando sea posible) por la cual se enviarán en flujo los datos del vehículo. En posteriores etapas de diseño se valorará la cantidad de datos a enviar según la red en uso, disponibilidad de los recursos, saturación local, *jitter* en las comunicaciones, etc.

La transmisión de los datos recibidos constituyen la calidad característica del sistema. Sin embargo, como es posible que por factores del entorno ciertos valores no se puedan transmitir en el momento, estos se almacenarán en memoria persistente (mínimo 1 Mb) hasta que haya una conexión de red por la que enviarlos.

Por otra parte, se ofrece la posibilidad de ver los datos con retardo mínimo: dado que no siempre es adecuado observar los datos *a posteriori* sino que puede ser necesario evaluarlos en el momento, se podrá observar información en el momento del estado del vehículo, información de sensores, etc. Esta visualización se realizará mediante un dispositivo externo al sistema como puede ser un *smartphone*. Para facilitar la integración, se desarrollará en paralelo una aplicación móvil específicamente diseñada para VIMS.

Entre las demás características del producto, se contempla una más que es la geolocalización del vehículo. Para ello, se usarán las distintas redes con las que contará el producto así como el GPS. De esta forma, al usuario final no solo se le mostrarán estadísticas e información sobre sus desplazamientos sino que también sabrá en qué puntos ha estado y así obtener más información con respecto a su conducción. Esto permite también usar VIMS como medida de seguridad, en caso de hurto del vehículo, para saber su ubicación precisa. También puede ser útil como recordatorio de dónde estaba aparcado el coche, ya que solo habrá que revisar la última ubicación.

Finalmente, el sistema deberá integrarse de forma fácil y sencilla, y así debe ser también su utilización. Esto se traduce en que tanto el montaje como el desmontaje debe ser sencillo, permitiendo que si se necesita acceso a los puertos estándar del vehículo el sistema no será un impedimento.

En definitiva, para conductores de vehículos que quieran conocer más información sobre su automóvil, que necesiten hacer diagnósticos o que quieran estadísticas/datos, VIMS es un sistema autónomo integrado que ayuda al conductor a generar y obtener los datos mencionados anteriormente. A diferencia de otras soluciones presentes en el mercado, nuestro producto permitirá integrarse sin dificultades en cualquier vehículo que cumpla con los estándares y sin ninguna configuración adicional para el usuario. Además, será personalizable y podrá recibir actualizaciones y mejoras que aumenten su funcionalidad y amplíen su compatibilidad con nue-

vos vehículos.

3.2.2. Características de los usuarios finales

Dado que el sistema se integra sobre un vehículo, se espera los usuarios finales puedan tener una o varias de las siguientes características:

- El usuario contará con un carnet de conducir.
- El usuario dispondrá de un vehículo que cuente con alguno de los conectores estándar empleados por VIMS.
- El usuario trabajará en temas relacionados con la mecánica y automovilismo.
- El usuario tendrá interés en la automatización de tareas y generación de datos.
- El usuario tendrá interés en el análisis de datos y generación de información a partir de los mismos.
- El usuario usará de forma habitual su vehículo.
- El usuario vivirá en una zona en donde haya cobertura de alguna de las tecnologías inalámbricas empleadas por el dispositivo para comunicarse.
- El usuario contará con un *smartphone* o dispositivo inteligente que le permita conectarse con su vehículo.

3.2.3. Restricciones generales

El producto en principio no está restringido en términos de legalidad, ya que las actividades de monitorización que realiza en nada deberían alterar la funcionalidad del vehículo al que se conectan (el producto, como se especificó en el alcance – 3.1.2, no está diseñado para modificar parámetros relativos al vehículo).

Por otra parte, el sistema no solo se compone del dispositivo que va sobre el automóvil en sí, sino también de la aplicación de visualización. Por consiguiente, el desarrollo al completo deberá realizarse teniendo en cuenta todos los componentes para que sea compatible desde el primer momento.

En lo referente a la presentación gráfica, los usuarios solicitaron que fuese simple y accesible, así como personalizable. Se comentará más sobre esto en la sección de requisitos de usuario (3.3.1), pero se comenta aquí ya que es una limitación en el desarrollo en cuanto a que define una característica que se considera necesaria.

Además, como el sistema se integra con los automóviles, la única fuente de alimentación será el vehículo en sí. Existe una casuística en la que el conductor apaga el coche pero existen

datos que todavía no se han podido enviar. Ante esta situación, como el conector OBD sigue ofreciendo alimentación, se tiene que detectar este evento y decidir si o bien se envían los datos pendientes si se cuenta con conexión de red o bien se almacenan en memoria persistente hasta que el vehículo se encienda de nuevo. De esta forma, no se compromete la batería del automóvil.

Como se ha contemplado anteriormente, el sistema será geolocalizable. Esto conlleva tener en cuenta el consumo adicional de los módulos de geolocalización, los cuales suelen tener asociados un elevado gasto energético, para evitar un desgaste prematuro de la batería. Por otra parte, como la información recogida del vehículo se puede considerar información sensible, debe permanecer privada y accesible únicamente al usuario poseedor de VIMS.

Si el sistema funciona correctamente, se espera una implantación en el mercado elevada y que se empiece a redistribuir de forma nacional. Esto se produciría por la necesidad del mercado de este producto, la facilidad en su instalación y su correcto funcionamiento final.

3.2.4. Suposiciones y dependencias

Dependencias

- DEP-1** El sistema necesitará siempre de una conexión a Internet para funcionar. En otro caso, almacenará los datos en la memoria persistente que se estima de al menos 1 Mb.
- DEP-2** El sistema dependerá del área geográfica en la que se encuentre para enviar información, ya que pueden existir zonas en donde no haya ningún tipo de conectividad.
- DEP-3** El sistema deberá localizarse al aire libre en una zona sin apantallar la señal de los satélites para ofrecer los servicios de geolocalización al completo.
- DEP-4** Los dispositivos de visualización deberán encontrarse en un rango suficientemente cercano para realizar la transmisión de la información. Dependerá de la tecnología de red utilizada: WiFi, Bluetooth o BLE.
- DEP-5** Las redes que se usen para transmitir datos pueden variar con el tiempo así como su disponibilidad y velocidad. El sistema debe estar preparado para este escenario y adecuarse correctamente.
- DEP-6** El método de transmisión y envío de datos por Internet hacia el servidor de gestión y almacenamiento debe ser lo más eficiente posible, ya que la calidad de la conexión puede ser mala (debido a **DEP-5**).

Supuestos

- SUP-1** Se supone que el vehículo en que se implante el sistema contará con un conector estándar que permita las comunicaciones y la alimentación del módulo, como el OBD.

- SUP-2** Se supone que el servidor de recolección de datos será capaz de aguantar la demanda de los dispositivos destinados a pruebas y de una cantidad considerable de dispositivos en entorno de producción.
- SUP-3** Se supone que cada VIMS contará con un identificador único que permitirá identificar al dispositivo inequívocamente del resto.
- SUP-4** Se supone que cada usuario poseedor de un VIMS contará con acceso a Internet recurrente y dispondrá de una cuenta en el servicio de estadísticas que le permita vincular su(s) dispositivo(s) VIMS a su cuenta, según el **SUP-3**.
- SUP-5** Se supone que las tramas transmitidas por el vehículo son las estándar definidas de forma global para los automóviles del mercado. Este supuesto guarda una estrecha relación con **SUP-1**.

3.3. Requisitos específicos

A continuación se definen los requisitos específicos en sí del proyecto. En esta sección primero se habla de las necesidades de los usuarios (3.3.1) y a continuación se prosigue con las funciones, dependencias y restricciones que se aplican al proyecto durante el desarrollo en los puntos pendientes.

Se pretende que esta sección sea auto contenida, esto es, sirva por sí sola para diseñar el sistema. Sin embargo, se recomienda una lectura acompañada de las secciones anteriores para una mayor comprensión.

3.3.1. Requisitos de usuario

La especificación de requisitos de usuario pretende recoger las necesidades de los usuarios finales del producto para tenerlas en cuenta a lo largo del desarrollo del mismo.

Para esta labor, se ha realizado un cuestionario voluntario donde la comunidad de conductores y no conductores informaban de sus preferencias, gustos, necesidades y otra información que se pudiera considerar relevante.

Como los encuestados se mezclaban, se ha decidido hacer una separación por conductores y no conductores teniendo en cuenta sus preferencias. La distribución de las preguntas quedaba de esta forma:

1. Se le pregunta al encuestado datos básicos y si es conductor.
2. En caso afirmativo, se recoge esta información:
 - Años de carnet.

- Tipo de carnet.
 - Edad del conductor.
 - Tecnologías que presenta el vehículo de uso habitual.
 - Tipo de vehículo de uso habitual.
 - Años del vehículo de uso habitual.
3. A continuación, se le muestra al usuario una matriz de selección en donde, de una escala del 1 al 20, debe priorizar los distintos elementos que aparecen (en total, 20). Solo se permite una única prioridad por elemento, de forma que múltiples elementos tengan la misma prioridad.

Los elementos que se han propuesto en este cuestionario son (tabla 3.1):

Visualización en tiempo real	Información detallada de errores	Velocímetro	Cuentarrevoluciones
Marcha actual (1 ^a , 2 ^a , ...)	Temperatura del aceite	Presión del aceite	Temperatura exterior
Intensidad del acelerador (%)	Consumo actual	Presión de las ruedas	Presión de los inyectores
Nivel de combustible	Distancia recorrida	Nivel de batería	Nivel de carga en valor absoluto del motor
Presión atmosférica	Temperatura de la toma de aire	Temperatura del refrigerante	Temperatura del motor

Cuadro 3.1: Opciones ofrecidas a los encuestados. Se han escogido diversas opciones que se encuentran entre los datos habituales generados por un vehículo.

4. Finalmente, de forma libre, se le pide al usuario que indique qué sensores añadiría a su vehículo (si tiene), qué datos querría poder medir y qué tareas querría automatizar.

Las opciones anteriores se componen de elementos que se pueden obtener mediante el OBD estándar, usando una lista de *pids* comunes públicos que todos los vehículos deberían usar [7]. Dado que la encuesta se podía realizar tanto por conductores como por no conductores, se van a separar las respuestas en donde se evaluará la población global (tabla 3.2) y luego a la población únicamente de conductores (tabla 3.3). Todo el desglose y el análisis se encuentra disponible en: <https://s.javinator9889.com/vims-analysis>.

Puntuación	Opción
1	Velocímetro
1	Nivel de combustible
4	Marcha actual
5	Distancia recorrida
5	Nivel de batería
6	Cuentarrevoluciones
8	Intensidad del acelerador (%)
8	Presión de las ruedas
10	Temperatura del aceite
10	Nivel de carga en valor absoluto del motor
12	Visionado en tiempo real
12	Consumo actual
12	Temperatura del motor
14	Presión del aceite
14	Temperatura del refrigerante
16	Presión atmosférica
17	Temperatura exterior
19	Información detallada de errores
19	Presión de los inyectores
20	Temperatura de la toma de aire

Cuadro 3.2: Puntuaciones obtenidas de forma general, por la población al completo (conductores y no conductores).

Puntuación	Opción
1	Velocímetro
1	Nivel de combustible
3	Temperatura del motor
4	Temperatura del aceite
4	Nivel de batería
5	Distancia recorrida
6	Cuentarrevoluciones
7	Presión de los inyectores
8	Presión de las ruedas
10	Presión del aceite
11	Marcha actual
11	Presión atmosférica
14	Temperatura del refrigerante
15	Temperatura exterior
17	Intensidad del acelerador (%)
18	Información detallada de errores
18	Consumo actual
19	Temperatura de la toma de aire
20	Visionado en tiempo real
20	Nivel de carga absoluta del motor

Cuadro 3.3: Puntuaciones obtenidas de forma general, por la población (excluidos los no conductores).

Resulta interesante ver cómo varios elementos coinciden en puntuación y posición, como el *Velocímetro* o el *Nivel de combustible* mientras que otros se mueven radicalmente de posición, como la *Marcha actual*, la *Temperatura del motor*, etc.

En general, se observa cómo los usuarios quieren tener más información sobre distintos elementos del vehículo y no necesariamente en “tiempo real” sino *a posteriori*, como información estadística. En particular, entre la comunidad de conductores prima información mecánica y relativa al estado del vehículo más allá de información sobre el viaje.

Además, hay bastante interés en la automatización de ciertas acciones como el cierre de las puertas automáticamente, *reset* del odómetro cuando se repuesta, notificaciones cuando haya

que realizar labores de mantenimiento, etc. Sin embargo, el diseño inicial del proyecto no contempla el realizar acciones sobre el automóvil principalmente porque los conectores estándar permiten la lectura de datos pero no la escritura. Para conseguir enviar información al vehículo sería necesario realizar un montaje directamente sobre los cables de los buses CAN que tiene, escuchar de forma activa las tramas y generar la información a enviar. En principio, esto se encuentra fuera del alcance del proyecto, por lo que se ignoran todas las modificaciones sobre el automóvil en sí.

Finalmente, se observa que hay un interés generalizado en que la interfaz sea simple, en que se pueda saber dónde se ha aparcado el vehículo y en que el sistema sea fluido y rápido.

A partir de lo anterior, se definen los siguientes requisitos de usuario:

- RU-1** Como usuario no necesito que la información se muestre en tiempo real sino que se almacene para una posterior consulta, o bien en forma de estadísticas o bien como datos en crudo. Dichas estadísticas se conformarían de la media, varianza, tendencia y moda de los datos de interés recogidos individualmente por cada conductor durante el uso del vehículo, destacando la velocidad, carga del motor, consumo, tiempo acelerando, metros ascendidos, metros descendidos, etc.
- RU-2** Como conductor me gustaría conocer de primera mano el estado del motor, a niveles tanto de temperatura de los elementos principales como de estado de los distintos elementos mecánicos del mismo.
- RU-3** Como usuario, me gustaría conocer fácilmente información relativa al mantenimiento del vehículo, como el desgaste de las ruedas, nivel de aceite, líquido de frenos, etc.
- RU-4** Como usuario, y en estrecha relación con **RU-3**, me gustaría recibir una notificación o alerta sobre cuándo se debe realizar el mantenimiento del vehículo.
- RU-5** Como usuario también querría ser notificado si se detecta algún error o problema en el vehículo.
- RU-6** Como conductor, me gustaría saber dónde he aparcado el vehículo.
- RU-7** Como conductor, me gustaría que tras realizar un viaje o tras realizar un repostaje se generasen estadísticas que me sirvieran para analizar mi perfil de conducción.
- RU-8** Como usuario, me gustaría recibir recomendaciones sobre mi estilo de conducción para optimizar el consumo o entender mejor cómo manejar el vehículo.
- RU-9** Como usuario, me gustaría que el sistema fuese fluido y funcionase bien, con tiempos de carga pequeños y sin ir a trompicones.
- RU-10** Como usuario, me gustaría que el sistema fuese accesible e intuitivo, simple a primera vista pero personalizable. Se propone una interfaz como la que se muestra en la figura 3.1 que permite seleccionar el rango temporal sobre el cual evaluar los datos y la información estadística definida en **RU-1**:



Figura 3.1: Representación de los datos en la interfaz propuesta – fuente: <https://grafana.csselectronics.stellarhosted.com/d/6qvL90vMz/css-playground-obd2?orgId=1>.

3.3.2. Requisitos funcionales

Analizando las características del producto y las necesidades del mercado, se proponen los siguientes requisitos funcionales con respecto a diversas temáticas y características de VIMS:

Conecividad

- RF-1** El sistema se conectará a cualquier vehículo que use un conector estándar, como el OBD.
- RF-2** El sistema usará redes móviles para intentar estar conectado siempre a la red. De esta manera, se intenta cubrir la dependencia **DEP-1** (pero se sigue sujeto a **DEP-2**).
Para esta labor, se propone realizar un uso combinado de redes móviles 4G/3G/2G y WiFi (si está disponible).
- RF-3** El sistema deberá ser geolocalizable bien mediante el uso de tecnologías GPS o bien mediante el uso de redes móviles.
- RF-4** El sistema usará redes PAN para permitir la comunicación con dispositivos cercanos y mostrar información en tiempo casi real.
- RF-5** El sistema, usando la conectividad de red, podrá actualizarse de forma remota para ofrecer nuevas funcionalidades.

Datos

- RF-6** El sistema recogerá todo tipo de datos emitidos por el vehículo y los clasificará según corresponda, con sus unidades de medida e información que representa.
- RF-7** En relación con el requisito **RF-6**, el sistema emitirá esos datos a un servicio en la red para su gestión. Los datos deberán ir acompañados de una marca temporal que permita identificar el instante origen, como se especifica posteriormente en **RF-10**.
- RF-8** Además de los datos del vehículo definidos en **RF-6**, el sistema deberá enviar información sobre la geolocalización del vehículo en sí, según lo estipulado en **RF-7**.
- RF-9** El sistema almacenará los datos en memoria si no se han podido transmitir hasta que o bien no quepan más datos o bien se hayan podido enviar. En caso de que la memoria estuviese completa, mediante indicadores visuales se le notificaría al usuario de este evento, dándole a entender que las entradas más antiguas serán eliminadas (ver **RNF-19**).
- RF-10** El sistema deberá llevar un control del tiempo para asociar el dato con la marca temporal en la que se obtuvo.
- RF-11** El sistema ofrecerá una interfaz desde la cual se podrá configurar la cuenta asociada y el dispositivo en sí.
- RF-12** El sistema, una vez se apague el vehículo, deberá notificar su última posición a modo de recordatorio de dónde se ha aparcado.
- RF-13** Las estadísticas de los viajes se generan tras un periodo de tiempo con el motor apagado. Dicho periodo se establece por defecto en 1 hora pero podrá ser configurable por el usuario desde la aplicación. Pasado ese tiempo, se generará la información relacionada con el trayecto realizado y se pondrá a disposición del usuario en forma de estadísticas y datos en sí. Opcionalmente, el usuario recibiría una notificación una vez estuviesen disponibles.
- RF-14** Además de las estadísticas de los viajes, se generarán estadísticas temporales: cada semana, cada mes y cada año. En ellas, el usuario conocerá de primera mano el uso que ha hecho del coche, gasto estimado, desgastes, etc. y se le notificarán cuando estén disponibles.

Servidor

- RF-15** El servicio web recibirá los datos transmitidos por los múltiples sistemas VIMS que existan.
- RF-16** El servicio web clasificará la información recibida por cada dispositivo y la asociará a la cuenta del usuario correspondiente.
- RF-17** El servicio web ofrecerá una interfaz en donde el usuario podrá ver información relacionada con sus últimos viajes, estadísticas e información del vehículo.

- RF-18** El servicio web ofrecerá una API para acceder desde otras máquinas a los datos almacenados e información estadística.
- RF-19** El servicio web mostrará información sobre la ubicación actual del coche, trazando un mapa con la ruta realizada.
- RF-20** El servicio web mostrará al usuario la última ubicación conocida del vehículo, es decir, dónde está aparcado.
- RF-21** El servidor almacenará los datos en principio de forma indefinida. Como todavía no se conoce la cantidad de información que se almacenará, en un futuro este requisito puede ser modificado y se puede establecer una temporalidad de los datos.

Usuario

- RF-22** El usuario se dará de alta en la plataforma y se asociará un dispositivo VIMS a su cuenta.
- RF-23** El usuario podrá usar su *smartphone* para visualizar información en tiempo casi real de su vehículo.
- RF-24** El usuario podrá definir información básica sobre el estado del vehículo actualmente. De esta forma, tras el tiempo estipulado por el fabricante o tras cierto kilometraje este recibirá una notificación indicándole los mantenimientos a realizar.
- RF-25** El sistema usará la información recogida según los requisitos **RF-6** y **RF-8** para ofrecerle al usuario un perfil personalizado sobre su conducción. En dicho perfil se tendrán en cuenta los datos de revoluciones del motor, intensidad del acelerador, desnivel del terreno, consumo del vehículo, distancia recorrida y demás para adecuarlo bajo un perfil propio según diversos modelos existentes [18].

Aplicación móvil

- RF-26** La aplicación deberá poder conectarse de forma inalámbrica al sistema VIMS en cuestión.
- RF-27** La aplicación deberá leer toda la información obtenida por VIMS con retardo mínimo de hasta 200 ms.
- RF-28** La aplicación deberá poder mostrar información estadística en base a los datos almacenados en el servidor. Para ello, se hará uso del requisito **RF-18**.
- RF-29** La aplicación podrá mostrar la última ubicación conocida del vehículo, es decir, dónde está aparcado.

3.3.3. Requisitos no funcionales

Precisión

RNF-1 El módulo de geolocalización ofrecerá una precisión que vendrá limitada directamente por el sistema utilizado. Así, con el GPS se puede llegar a obtener una precisión media de 5 metros mientras que con redes móviles la precisión media está en torno a los 60 metros. Por ello, no se considera algo crítico.

Rendimiento

RNF-2 El sistema deberá leer y recibir los datos del conector estándar a la máxima velocidad que permita este.

RNF-3 El sistema enviará los datos a medida los vaya recibiendo. Si hay congestión en la red y los datos no se pueden enviar tan rápido como se generan, se almacenarán en un *buffer* hasta poder transmitirlos.

RNF-4 La aplicación móvil deberá recibir directamente los datos desde el dispositivo VIMS con un retardo mínimo de 200 ms desde su petición, indiferentemente de la tasa de refresco.

Eficiencia

RNF-5 Según la sección 3.2.3, se prevé que el dispositivo pueda funcionar sin alimentación directa del coche cuando queden datos por enviar. Por razones de seguridad, el tiempo máximo que el sistema permanecerá activo será de hasta una hora. Así se evita drenar la batería del coche. Pasado ese periodo, los datos que no se pudieron transmitir se almacenarán de forma definitiva en memoria persistente hasta que el vehículo se encienda de nuevo.

RNF-6 A razón del requisito **RNF-5**, se deberá escoger la red priorizando aquella que suponga un menor consumo. Este criterio se puede basar en usar la que ya esté activa (ya que el inicio de una interfaz de red conlleva un alto consumo), usar una red de menor potencia o usar la red que esté disponible. En otro caso, se almacenarán los datos pendientes de envío en VIMS.

Mantenibilidad

RNF-7 Los componentes *software* estarán estructurados siguiendo patrones de código limpio (como los principios S.O.L.I.D.) de forma que permitan una fácil mantenibilidad del mismo.

RNF-8 Las actualizaciones OTA no deberán tardar más de 5 minutos en instalarse (se obvia el tiempo de descarga debido a que se depende directamente de la red en la que se esté conectado), para evitar un alto periodo de *downtime*.

Portabilidad

RNF-9 Como se mencionó en la perspectiva del producto (3.2.1), el sistema debe ser fácilmente montable y desmontable, permitiendo el uso directo del conector estándar para otras labores.

Recuperabilidad

RNF-10 En caso de fallo inesperado del sistema, este deberá reiniciarse en no más de 30 segundos, si el vehículo está activo. En otro caso, permanecerá desactivado.

Seguridad

RNF-11 Si no se detecta ningún tipo de red, el sistema avisará al usuario de dicho evento y se almacenarán los datos temporalmente hasta que esté disponible.

RNF-12 Si la aplicación no detecta conexión con el servidor, notificará al usuario de dicho evento.

RNF-13 Dependiendo de la precisión de la ubicación del GPS, el sistema informará al usuario de la calidad del mismo y de las “consecuencias” (falta de precisión en el recorrido, en el aparcamiento, etc.).

Integridad

RNF-14 Los mensajes enviados hacia y desde el servidor estarán siempre cifrados y debidamente protegidos.

RNF-15 Las actualizaciones de la placa deberán estar verificadas y protegidas, para evitar instalar *malware* de un atacante.

RNF-16 Las comunicaciones directas con la aplicación deberán ser previamente verificadas, para evitar enviar información a otro(s) dispositivo(s) no autorizados.

RNF-17 Antes de acceder a ningún tipo de dato producido por VIMS, el usuario deberá autenticarse apropiadamente.

Escalabilidad

RNF-18 El servidor web puede recibir cientos (o miles) de peticiones cada minuto, por lo que deben tomarse medidas específicas que faciliten la escalabilidad del mismo.

Usabilidad

- RNF-19** El sistema indicará distintos estados y mensajes mediante indicadores visuales, como luces LED.
- RNF-20** Los indicadores visuales que se integren podrán ser ajustables por el usuario mediante un potenciómetro, que permitirá regular la intensidad lumínica a gusto del conductor.

3.3.4. Requisitos de interfaces externas

Interfaces de usuario

- RNF-21** Los datos mostrados en las interfaces de usuario deberán ser simples, claros y accesibles.
- RNF-22** La interfaz de usuario estará distribuida en varios dispositivos: el servidor, la aplicación y VIMS en sí. Desde las dos primeras, se ofrecen los datos recibidos, estadísticas, perfiles de conducción, etc. Desde el sistema, se muestra una interfaz básica de registro, configuración de la misma y gestiones básicas.
- RNF-23** La configuración inicial de VIMS se realizará o bien desde la placa en sí o bien desde la aplicación. Ambas interfaces serán similares.
- RNF-24** Las notificaciones al usuario en principio se realizan única y exclusivamente desde la aplicación.

3.3.5. Restricciones de desarrollo

Limitaciones hardware

- RNF-25** El sistema debe ser lo más pequeño posible y adaptarse a los vehículos en general. Para ello, se encapsulará el *hardware* en una carcasa exterior que lo proteja.
- RNF-26** El circuito del sistema debe diseñarse teniendo en cuenta que ciertos pines pueden ser expuestos hacia el exterior para añadir módulos, en una versión futura.

3.3.6. Requisitos de entorno físico

- RNF-27** El sistema en principio estará alojado siempre dentro del vehículo. Sin embargo, se estima que en épocas de especial calor o según la zona geográfica, la temperatura del interior del habitáculo pueda alcanzar los 60 °C. Por ello, debe estar preparado para soportar dichas temperaturas.
- RNF-28** El sistema sin embargo no estará preparado para resistir al agua, por lo que nunca debe entrar en contacto con la misma. Se sugiere pues que se sitúe en un entorno aislado.

4 Diagramas y diseño

En esta sección se detalla cómo se ha diseñado en general el sistema. Por una parte, en la sección 4.1 se introducen los diagramas de casos de uso que dan una visión general de cómo se va a organizar todo el sistema. Además, se comentan también los diagramas de bloques que dan forma a la arquitectura final que se va a implementar. A continuación, en la sección 4.2 se introduce la PCB diseñada y las distintas decisiones de diseño que se han considerado a la hora de la fabricación. Después, en la sección 4.3 se habla del proceso de diseño 3D y de cómo el sistema se imagina estará embebido dentro del vehículo. Luego en la sección 4.4 se estructurarán las tareas definidas en la sección 2.5 y se realizará la planificación en tiempo real del sistema. Finalmente, se comenta en más detalle el proceso de diseño *software*, en la sección 4.5, en donde se comentarán los casos de uso del sistema, cómo se ha diseñado el código que va en VIMS y diseños y patrones para las interfaces gráficas de la implementación final.

4.1. Diagramas que modelan el sistema

Para facilitar el entendimiento del sistema al completo se han desarrollado dos tipos de diagramas: los diagramas de casos de uso con sus correspondientes explicaciones (punto 4.1.1) y los diagramas de bloques que modelan al sistema en su conjunto, sus relaciones y operaciones (punto 4.1.2).

Es importante destacar que los diagramas se quedan en una capa de abstracción por encima de la definición de comportamiento del mismo, es decir, no se entra en excesivo detalle sobre cómo va a hacer algo, solo especifica el qué.

4.1.1. Casos de uso

A continuación se presentan los distintos diagramas de casos de uso que modelan el comportamiento del sistema ante ciertos eventos. Se incluye además, por cada caso de uso, una tabla adjunta que describe el caso de uso, los pasos esperados en la secuencia normal de ejecución, las excepciones previstas durante la ejecución normal y la resolución de las mismas.

Caso de uso 01 – autenticación

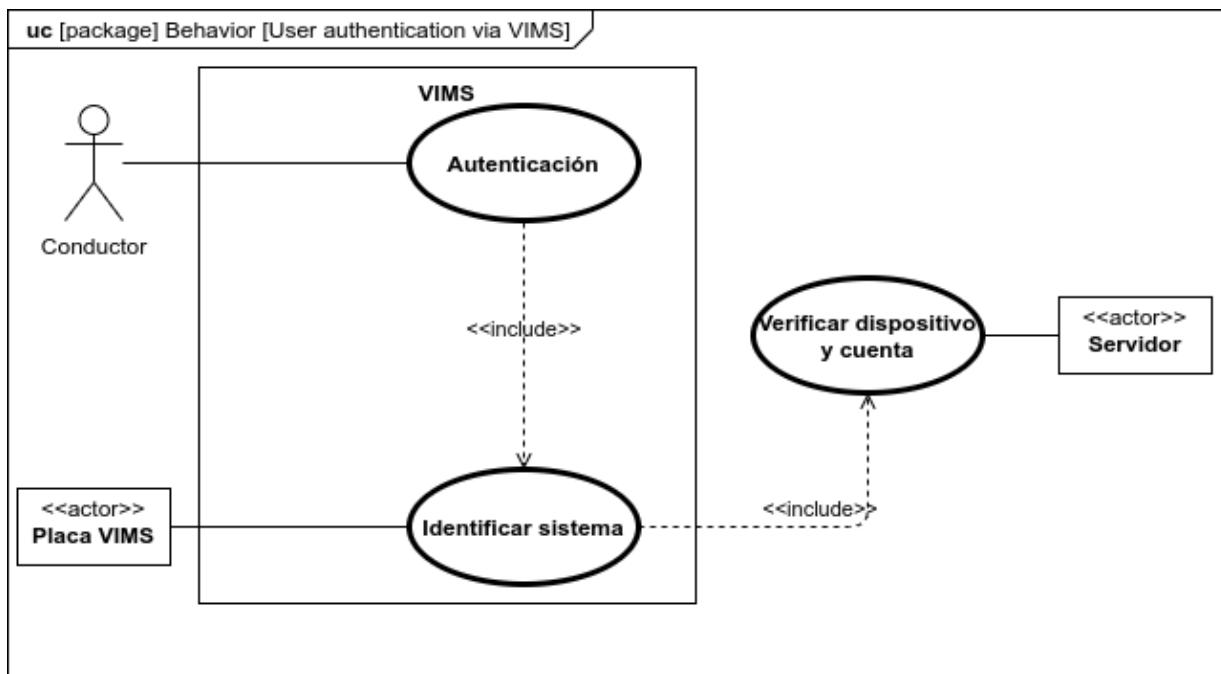


Figura 4.1: Caso de uso 01 – autenticación.

01	<i>Autenticación</i>	
Descripción	La placa identificará de forma inequívoca al conductor (usuario) y a sí misma frente al servidor.	
Secuencia normal	Paso	Acción
	1	El usuario se autentica contra la placa con su cuenta personal ya creada.
	2	La placa VIMS recoge la información del usuario y la envía al servidor junto con su identificador único.
	3	El servidor verifica que la cuenta del usuario existe y se asocia la información al dispositivo.
Excepciones	Paso	Acción
	2	La placa no cuenta con conexión a la red o el servidor no está disponible.
	3	La cuenta del usuario no existe.
Resolución	La placa funciona en modo desconectado, es decir, no transmite datos al servidor.	

Caso de uso 02 – generación y transmisión de datos

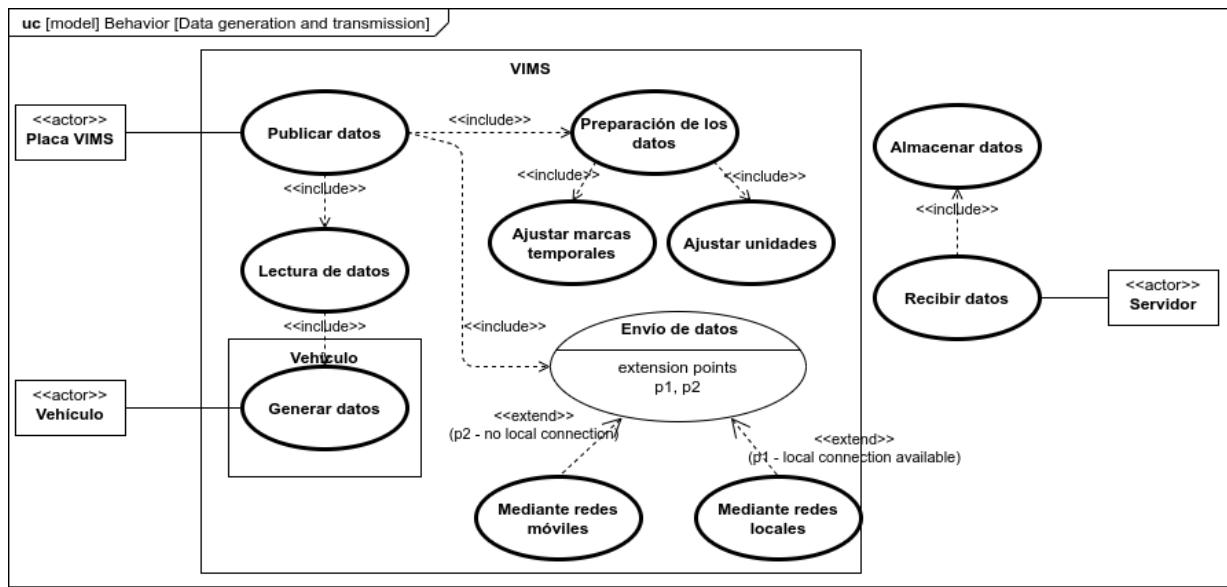


Figura 4.2: Casos de uso 02 – generación y transmisión de datos.

02	<i>Generación y transmisión de datos</i>	
Descripción	El dispositivo VIMS recibirá los datos del vehículo al que está conectado y los preparará para una posterior transmisión al servidor remoto de almacenamiento y gestión.	
Secuencia normal	Paso	Acción
	1	La placa VIMS recibe los datos que el vehículo está generando de forma continuada.
	2	Los datos recibidos se preparan para el envío, ajustando cierta información y añadiendo valores como la cuenta asociada a dichos datos.
	3	Mediante el uso de redes móviles o locales, según disponibilidad, se envían los datos al servidor.
	4	El servidor recibe la información transmitida por el sistema y la almacena para una posterior visualización y tratamiento.
Excepciones	Paso	Acción
	1	El vehículo no está conectado o no transmite datos.
	2	Todavía no hay ninguna cuenta asociada a la placa VIMS.
	3.1	No hay redes móviles disponibles, se intenta transmitir por redes locales.
	3.2	No hay redes locales disponibles, se intenta transmitir por redes móviles.
	3.3	No hay redes disponibles, se almacenan los datos para su posterior transmisión.
Resolución	La placa funciona en modo desconectado, es decir, no transmite datos al servidor. Si existe una cuenta asociada pero no hay conexión, los datos se almacenan en memoria hasta que se puedan transmitir.	

Caso de uso 03 – generación de estadísticas

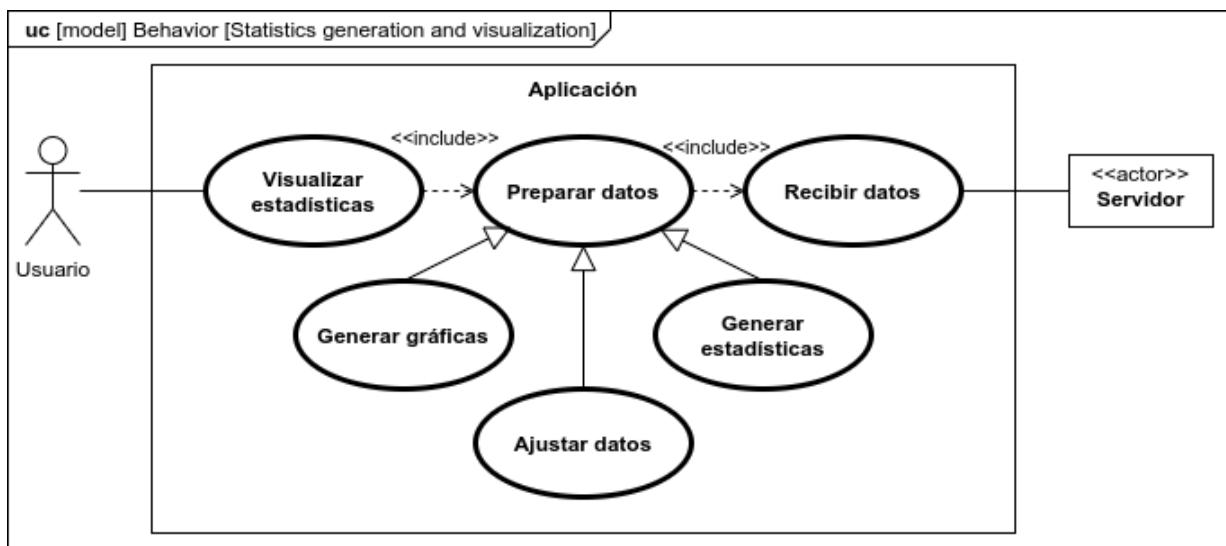
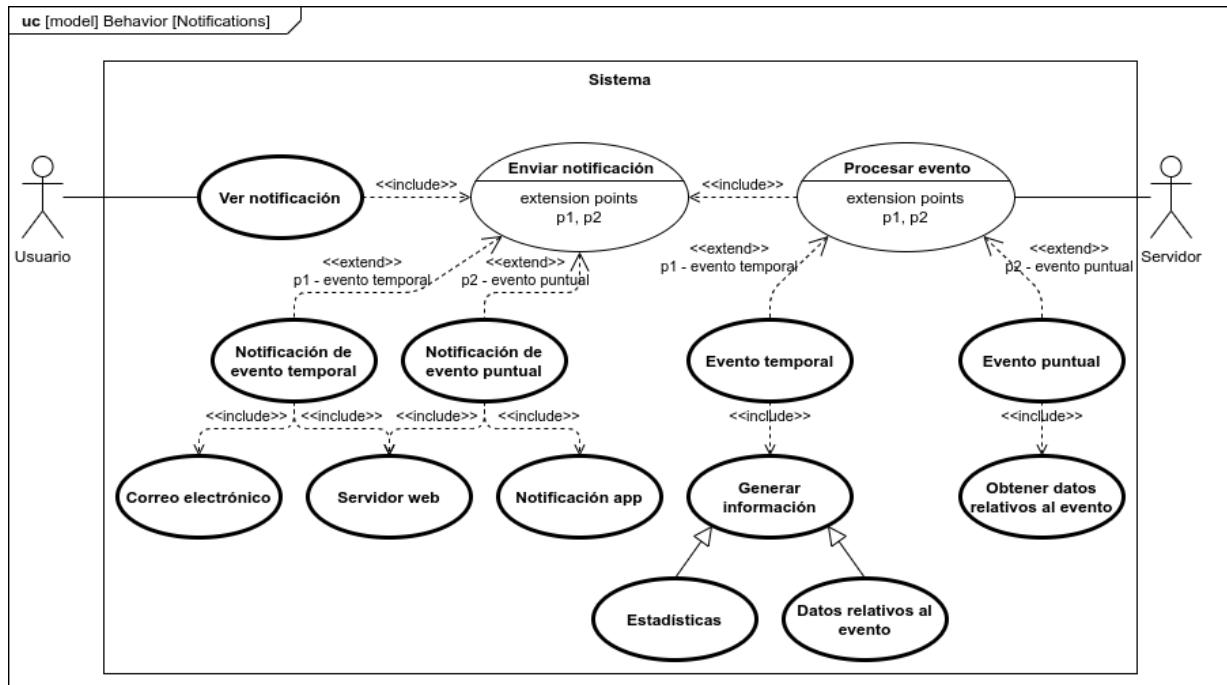
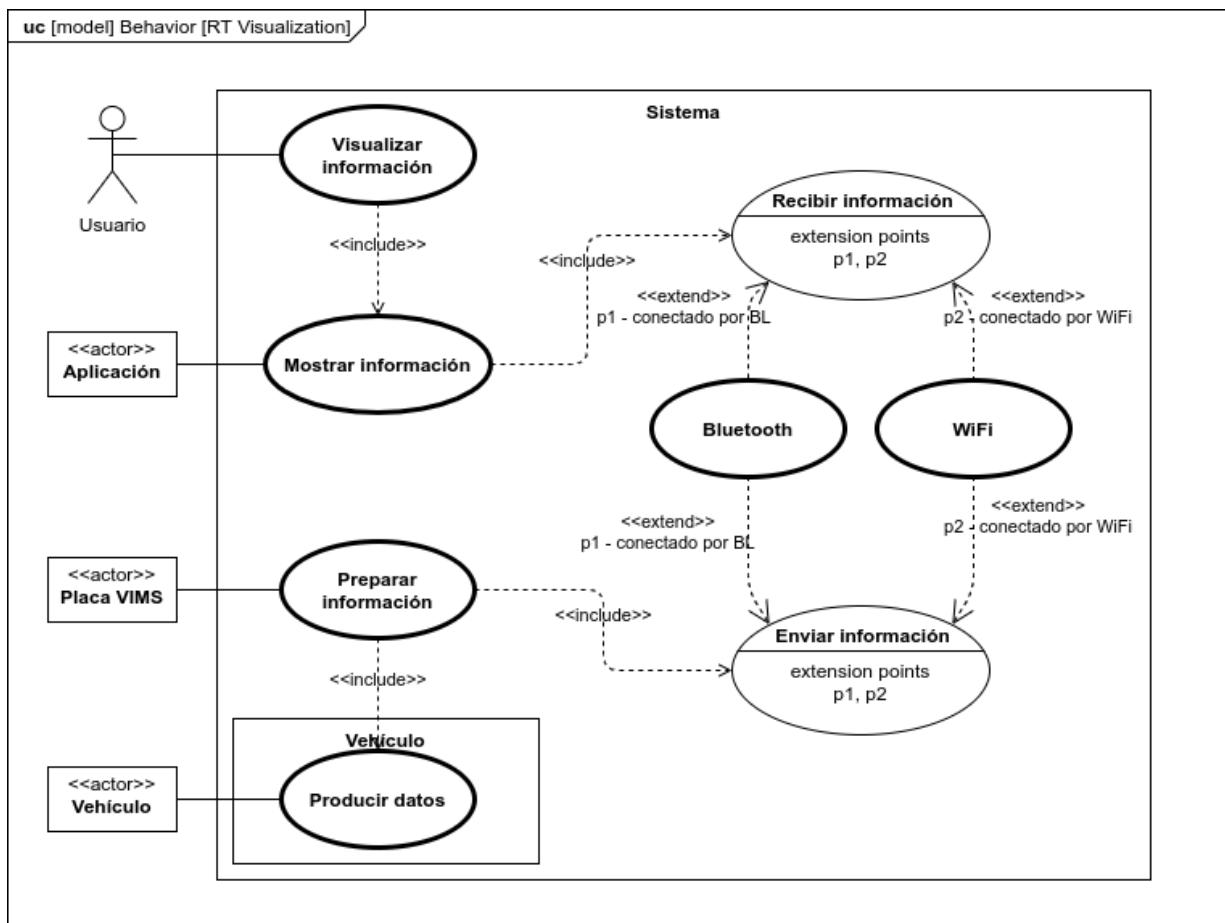


Figura 4.3: Caso de uso 03 – *generación de estadísticas*.

03	<i>Generación de estadísticas</i>	
Descripción	El servidor, en conjunción con el resto de elementos del sistema, preparará los datos para generar información estadística útil para el usuario.	
Secuencia normal	Paso	Acción
	1	El usuario solicita al servidor visualizar estadísticas con respecto a sus vehículos.
	2	El servidor prepara los datos y genera distintos tipos de información estadística visual basados en tablas, gráficos, etc.
Excepciones	Paso	Acción
	1.1	El usuario no está autenticado.
	1.2	El usuario no cuenta con ningún dispositivo VIMS asociado.
Resolución	Todavía no se ha registrado ningún dato.	
	Se notifica al usuario de este suceso y se le sugiere crear una cuenta.	

Caso de uso 04 – envío de notificacionesFigura 4.4: Caso de uso 04 – *envío de notificaciones*.

04	<i>Envío de notificaciones</i>	
Descripción	El servidor, en conjunción con el resto de elementos del sistema, detecta ciertos eventos y actúa generando una notificación que envía al usuario.	
Secuencia normal	Paso	Acción
	1	El servidor en un instante puntual produce y procesa un evento.
	1.1	Si el evento se ha producido por un hecho (p.e.: repostar, finalizar un viaje, ...), es un evento puntual sobre el cual se envía información relativa al mismo y al contexto (estadísticas del depósito, información del viaje, etc.).
	1.2	Si el evento se ha producido porque ha pasado un lapso de tiempo, es un evento temporal. El servidor generará estadísticas relativas a ese lapso de tiempo y mandará esa notificación.
	2	Se envía una notificación al usuario con los datos relativos al evento.
	2.1	Si es un evento puntual, la notificación se envía a todos los medios: correo electrónico, servidor web y aplicación.
	2.2	Si es un evento temporal, la notificación se envía solo al correo electrónico y al servidor web.
	3	El usuario recibe la notificación en alguno de los tres medios.

Caso de uso 05 – visualización en tiempo realFigura 4.5: Caso de uso 05 – *visualización en tiempo real*

05	<i>Visualización en tiempo real</i>	
Descripción	Un usuario podrá visualizar información en tiempo real sobre su vehículo mediante la aplicación.	
Secuencia normal	Paso	Acción
	1	El usuario solicita visualizar información sobre el vehículo desde la aplicación.
	2	La aplicación recibe la información de la placa VIMS usando redes PAN.
	3	La placa VIMS prepara la información que recibe del vehículo y la transmite hacia la aplicación.
Excepciones	Paso	Acción
	1.1	El usuario no está autenticado.
	1.2	El usuario no cuenta con ningún dispositivo VIMS asociado.
	2.1	No hay conexión mediante Bluetooth, se realiza la comunicación por WiFi.
	2.2	No hay conexión mediante WiFi, se realiza la comunicación por Bluetooth.
	2.3	La placa VIMS está desconectada.
Resolución	La placa no realiza ninguna transmisión de información hasta que no haya una conexión disponible.	

Caso de uso 06 – generación de eventos

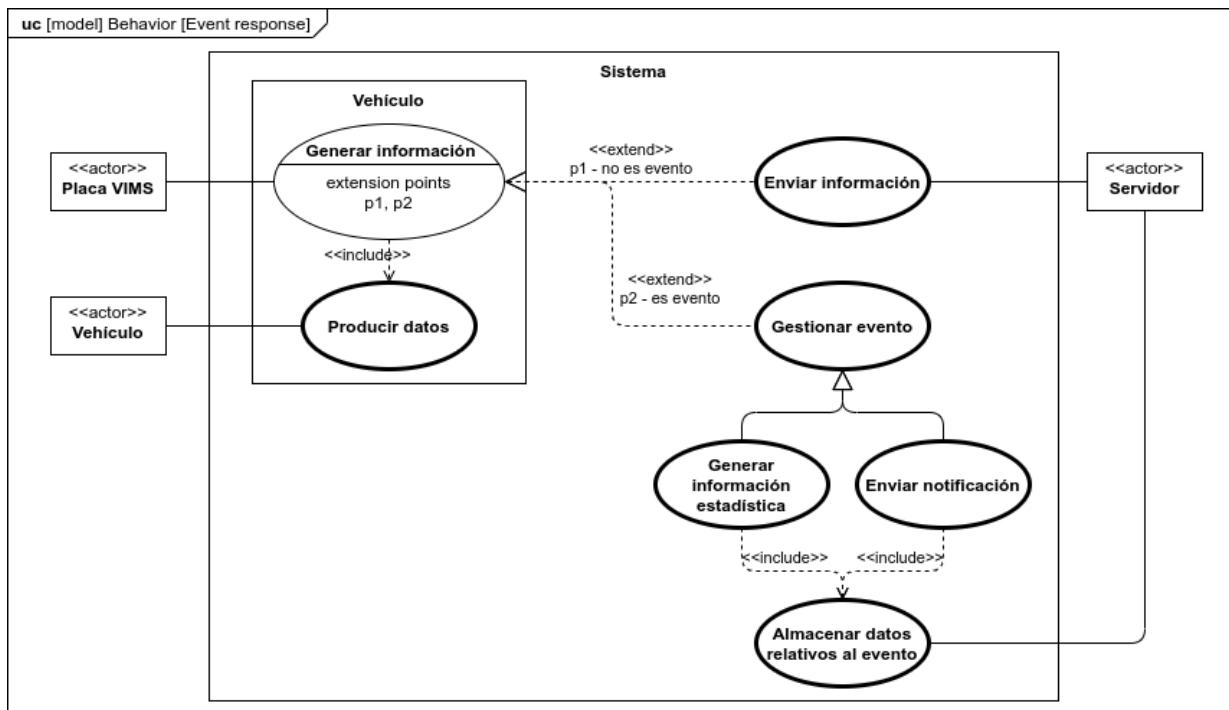


Figura 4.6: Caso de uso 06 – generación de eventos

06	<i>Generación de eventos</i>	
Descripción	El dispositivo VIMS, a la hora de enviar datos, podrá producir eventos según el tipo de información que haya de enviar.	
Secuencia normal	Paso	Acción
	1	El dispositivo VIMS genera información y la transmite hacia el servidor.
	1.1	Si la información a transmitir es “normal”, se envía directamente al servidor.
	1.2	Si la información a transmitir es eventual, se procesa el evento generando información estadística referente al mismo o mandando una notificación.
	2	El servidor recibe el evento y se encarga de gestionarlo, como se vio en el UC-04.
Excepciones	Paso	Acción
	1	No hay ninguna cuenta asociada al dispositivo.
	1.1	No hay conexión a Internet por parte del dispositivo VIMS.
Resolución	La placa funciona en modo desconectado, es decir, no transmite datos al servidor.	

Caso de uso 07.* – envío, almacenamiento y visualización de geolocalización

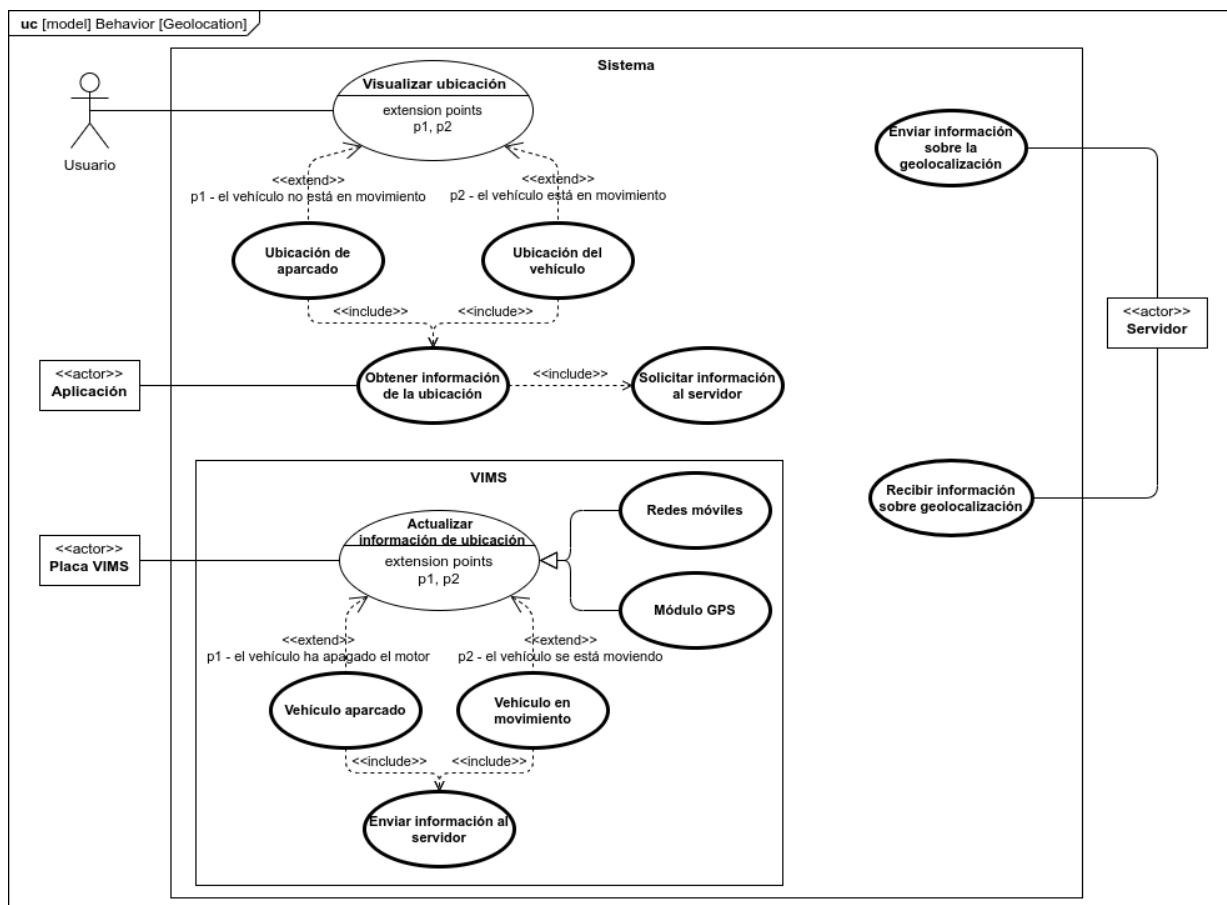


Figura 4.7: Casos de uso 07.* – envío, almacenamiento y visualización de geolocalización

07.1	<i>Visualización de ubicación</i>	
Descripción	El usuario mediante una aplicación podrá visualizar la ubicación del vehículo en tiempo real. Si el vehículo está apagado, se visualiza la ubicación del aparcamiento.	
Secuencia normal	Paso	Acción
	1	El usuario inicia una aplicación para visualizar la ubicación de su vehículo.
	1.1	Si el vehículo se encuentra en movimiento, se visualiza la ubicación en tiempo real según se va actualizando.
	1.2	Si el vehículo está apagado se considera que está aparcado y se visualiza la última ubicación conocida, correspondiente con el aparcamiento.
	2	Se realizan peticiones al servidor para obtener la información de la geolocalización.
Excepciones	Paso	Acción
	1	No hay ningún dispositivo asociado a la cuenta.
	1.1	El vehículo está desconectado de la red, por lo que no se pueden transmitir los datos de geolocalización.
	1.2	No hay ningún dato de geolocalización almacenado.
	2	La aplicación no cuenta con conexión a la red.
Resolución	La placa funciona en modo desconectado, es decir, no transmite datos al servidor.	

07.2	<i>Envío de ubicación</i>	
Descripción	La placa VIMS enviará la información relativa al vehículo al servidor.	
Secuencia normal	Paso	Acción
	1	La placa actualiza la información sobre la ubicación del vehículo.
	1.1	Si se ha perdido la conexión con el vehículo (motor apagado), se considera que está aparcado y se envía la ubicación actual como “ubicación de aparcado”.
	1.2	Si el vehículo está activo (motor encendido), se considera que está en movimiento y envía de forma periódica la ubicación actual del vehículo.
	2	Según la precisión de las redes y la disponibilidad de las mismas, la ubicación se obtiene mediante dos métodos.
	2.1	Se obtiene la ubicación mediante el uso de redes móviles cuando el módulo GPS no se encuentre disponible o no se tengan suficientes satélites.
	2.2	Se obtiene la ubicación mediante el módulo GPS como primera alternativa, y se usan las redes móviles cuando estén disponibles para mejorar la precisión de la ubicación.
	3	Se manda la ubicación obtenida al servidor y se enlaza a la cuenta asociada.
Excepciones	Paso	Acción
	1	No hay ninguna cuenta asociada al dispositivo.
	1.1	No hay ninguna conectividad de red disponible para realizar el envío de la ubicación.
	3	La conexión de red no está disponible para realizar la transmisión de la información.
Resolución	La placa funciona en modo desconectado, es decir, no transmite datos al servidor.	

07.3	<i>Almacenamiento de los datos de ubicación</i>	
Descripción	El servidor almacenará y gestionará los datos de ubicación recibidos por los múltiples dispositivos VIMS.	
Secuencia normal	Paso	Acción
	1	El servidor recibe la información de la ubicación de un dispositivo VIMS.
	1.1	Los datos de ubicación se almacenan en una línea temporal para poder ver el histórico de ubicaciones del vehículo.
	1.2	El último valor de ubicación se almacena para una posterior visualización.
Excepciones	Paso	Acción
	1	No hay ninguna cuenta asociada al dispositivo.
Resolución	La placa funciona en modo desconectado, es decir, no transmite datos al servidor.	

4.1.2. Diagramas de bloques

Los diagramas de bloques ofrecen una visión genérica del sistema en su conjunto. Para este sistema, se han definido tres diagramas de bloques: el primero define a *grosso modo* cómo se compone el sistema embebido que irá en el vehículo, con sus respectivos componentes y condiciones; el segundo, define la arquitectura del servidor, las conexiones con la base de datos y la API para consulta de datos externa. Finalmente, el tercer diagrama muestra cómo es todo el sistema en su conjunto, uniendo VIMS con el servidor.

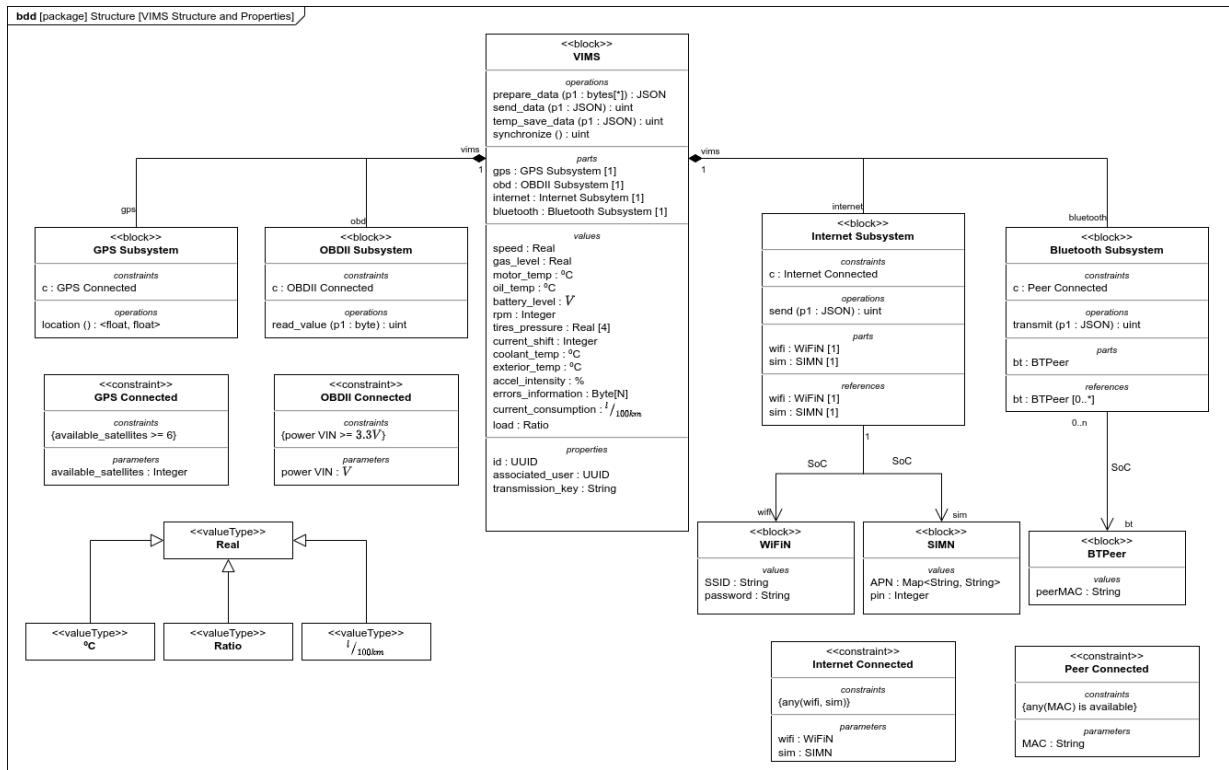


Figura 4.8: Diagrama de bloques que modela el conjunto de la placa VIMS y sus componentes.

Por una parte, se puede observar en la figura 4.8 cómo está compuesta la placa. Se distinguen los siguientes componentes principales:

- El dispositivo VIMS en sí, con sus funciones básicas y los datos que almacenará.
- El subsistema de conexiónado GPS, encargado de devolver la ubicación del sistema al que se conecta.
- El subsistema de conexiónado OBD-II, encargado de obtener en bytes los datos generados por el vehículo.
- El subsistema de conexiónado a Internet, encargado de escoger adecuadamente el módulo de conexión a la red adecuado según las circunstancias del sistema y de realizar las comunicaciones de red.
- El subsistema de conexiónado Bluetooth, cuya funcionalidad es la de conectarse a un dispositivo externo y realizar una emisión en el momento de los datos enviados por el vehículo.

Se va a empezar comentando cada uno de los subsistemas. Por su parte, el subsistema GPS (figura 4.9) cuenta con una restricción “*GPS Connected*” que define cuándo se considera que hay conexión con GPS. Este valor se establece en 6 satélites para resolver el problema del eje

Z que existe en trilateración con 3 satélites. En principio, con 4 satélites se podría considerar que hay conexión con el GPS pero con una redundancia del doble se obtiene una precisión de aproximadamente 5 m.

Este subsistema devuelve la latitud y la longitud, expresadas como números en coma flotante de 32 bits (que dota al sistema de una precisión máxima de siete dígitos decimales).

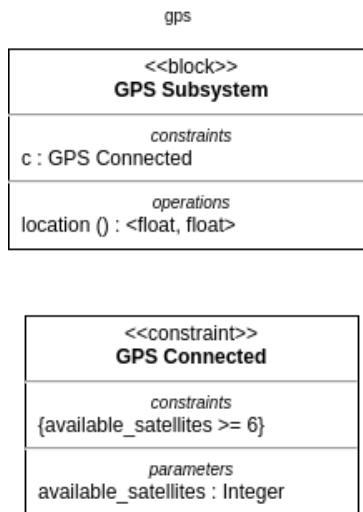


Figura 4.9: Subsistema que define las comunicaciones usando el GPS para obtener la ubicación.

El subsistema de OBD-II (figura 4.10) se encarga de la conexión con el vehículo y remite los datos al sistema principal. La función `read_value` recibe un byte que se correspondería al PID y devuelve los bytes con los datos asociados. Al igual que el subsistema anterior, se plantea la restricción de que el OBD-II esté desconectado. Esto se hará confirmando que el $V_{in} \geq 3,3V$ aunque, en la vida real, esta restricción no existe dado que el sistema OBD-II del vehículo siempre está proveyendo de 12V al conector (sin embargo, se ha contemplado porque existe la posibilidad de que la placa esté encendida mediante USB sin conexión al vehículo).

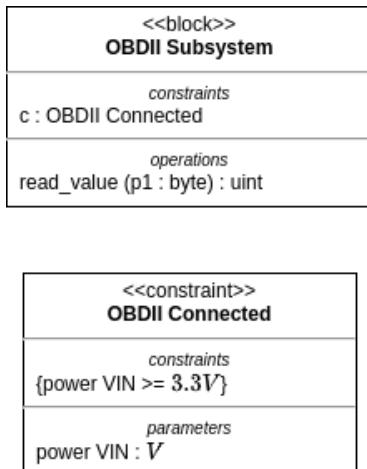


Figura 4.10: Subsistema que define las comunicaciones con el vehículo usando OBD-II.

El siguiente subsistema se encarga de la conectividad de red (figura 4.11). Contiene dos elementos “WiFiN” y “SIMN” en donde el primero provee de conectividad WiFi y el segundo de conectividad LTE. Este subsistema cuenta con un único método que le permite enviar datos en forma JSON al exterior, ya que es lo más compatible. Sin embargo, lo más importante es que se encarga de escoger según disponibilidad un sistema u otro. Esto se realiza comprobando que tienan acceso a la red según orden de prioridad: primero WiFi y luego SIM.

Individualmente, cada componente tiene las propiedades necesarias para conectarse. Por su parte, la conexión WiFi contiene información sobre el SSID y la contraseña. Mientras, la conexión LTE (SIM) almacena los datos del APN y del pin de desbloqueo de la SIM.

La restricción para que este subsistema se considere activo es que disponga de al menos una conexión activa y funcional.

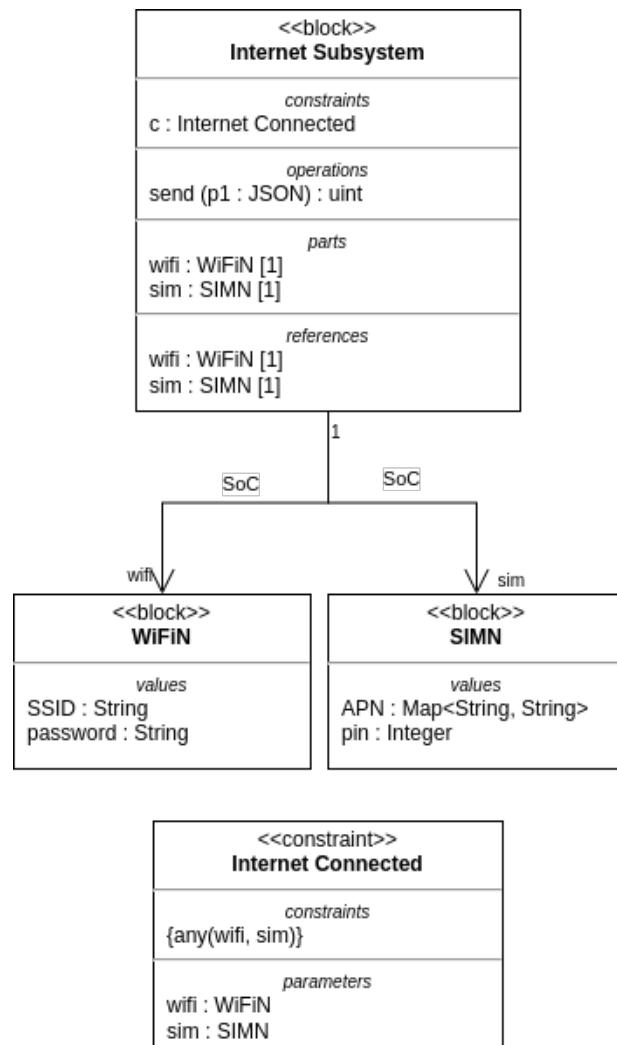


Figura 4.11: Subsistema que define las comunicaciones de red con el exterior.

El último subsistema que forma parte de VIMS es aquél que provee de conectividad Bluetooth (figura 4.12). Este subsistema funciona en principio usando BLE por lo que es posible conectar múltiples dispositivos a la vez. Sin embargo, por restricciones del proyecto se limita a una única conexión simultánea. Por cada elemento emparejado, se almacena su dirección MAC para realizar la conexión de nuevo.

Además, este subsistema tiene una restricción que define si está conectado si alguna MAC almacenada está disponible y aceptando conexiones.

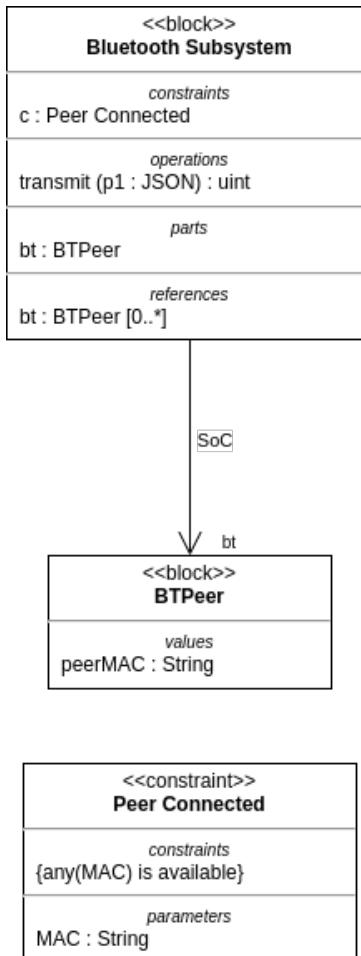


Figura 4.12: Subsistema de conexionado Bluetooth para la transmisión en vivo de los datos.

Finalmente, queda el sistema principal: VIMS que congrega al resto de subsistemas y hace uso de las funcionalidades que ofrecen (figura 4.13).

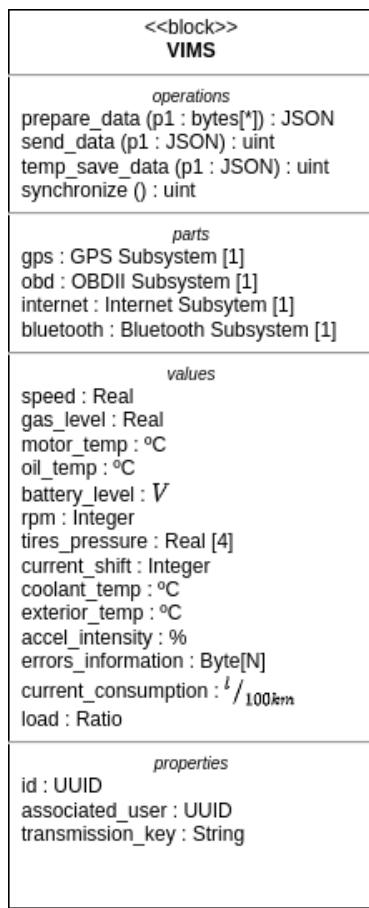


Figura 4.13: Componente VIMS dentro del sistema embebido.

Este sistema se congrega de los subsistemas definidos anteriormente y define múltiples métodos que realizan una serie de operaciones con los mismos. Por una parte, `prepare_data` prepara los valores recibidos en crudo por el subsistema OBD para su posterior transmisión. Esta operación realiza las conversiones adecuadas de los valores en bytes, ajusta las unidades de los mismos y lo guarda en los campos correspondientes definidos en “*values*”.

Por otra parte, la función `send_data` envía los valores previamente preparados al servidor remoto de gestión. El valor de retorno de esta operación es un código entero que identifica la respuesta del servidor ante la transmisión.

Si por casualidad el subsistema de conexión a Internet no se encuentra disponible (es decir, no hay ni conexión WiFi ni SIM), existe la función `temp_save_data` que almacenará los datos temporalmente en memoria persistente, hasta que se cuente con conexión a la red.

Finalmente, existe la función `synchronize` que se encargará de mantener actualizados los datos que se muestran en el dispositivo del usuario cuando dicho dispositivo está conectado por Bluetooth y está en modo de monitorización.

En lo referente a los valores del bloque, se corresponden a los distintos datos que serán accedidos usando el subsistema OBD y que se guardan como referencia para su posterior transmisión. No se va a entrar en detalle en explicar qué dato es cada uno de los mostrados. Referirse a la sección 2.4 para más detalles.

Por último, en el apartado de propiedades, se almacenan todos aquellos valores que identifican el dispositivo VIMS en sí y la cuenta de usuario asociada al mismo (junto con su clave de transmisión). Cada dispositivo VIMS deberá poder ser identificado de manera inequívoca y existe la posibilidad de que un mismo usuario tenga múltiples dispositivos VIMS. Como hay cierta información que se puede considerar sensible, se genera una clave de transmisión asociada a la cuenta y al dispositivo que permitirá asegurar la integridad, inmutabilidad y confidencialidad de los datos durante la transmisión.

Estos valores se almacenan durante el primer arrancado del dispositivo. Según los casos de uso (4.1.1), el dispositivo VIMS no hará nada mientras esos campos no hayan sido cumplimentados, impidiendo cualquier operación con el mismo.

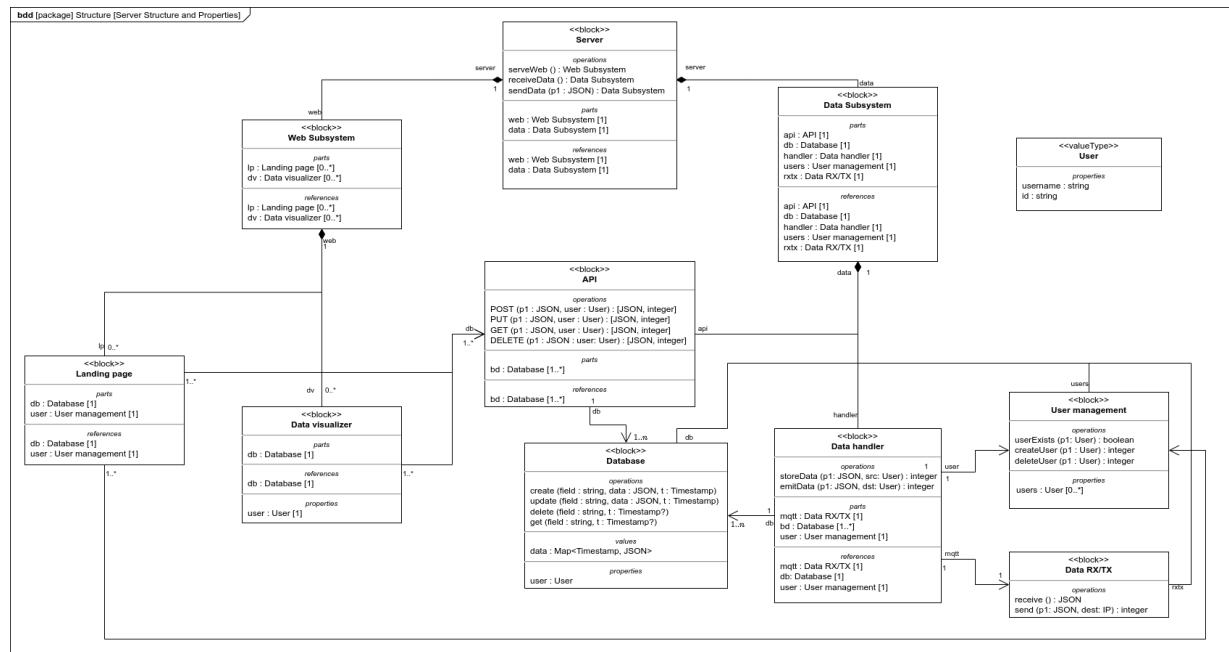


Figura 4.14: Diagrama de bloques que modela al servidor y sus componentes.

La arquitectura definida en el lado del servidor cuenta con menos componentes pero más segregados:

- El subsistema de la web, que define aquellos elementos que conforman la web con la que un usuario interactuará. Por una parte, existe una “*landing page*” que permite que los usuarios se registren y creen sus cuentas. Por otro lado, se cuenta también con un visualizador de datos en donde el usuario podrá acceder a los últimos reportes sobre sus

viajes y dispositivos, además de realizar las gestiones propias sobre su cuenta.

- El subsistema de datos, que engloba toda la gestión de la información por parte del servidor. Este subsistema se compone de una base de datos de almacenamiento de la información de usuario así como de los datos transmitidos por los dispositivos VIMS. Otro bloque que lo conforma es el de gestión de la información el cual relacionará los datos recibidos por los dispositivos VIMS con el usuario que los transmite. Finalmente, está el bloque de la API que relaciona los dos subsistemas: accede a la información de manera segura almacenada en el bloque de los datos y se la ofrece al subsistema de la web y, en un futuro, al sistema de la aplicación móvil.

Toda esta arquitectura se ejecutará en un servicio en la nube que congregará y simplificará las operaciones de mantenimiento y escalabilidad del sistema. Antes de continuar, se va a analizar cada subsistema individualmente.

El subsistema de la web (figura 4.15) se compone de dos bloques principalmente: un “*landing page*” y un visualizador de datos. El primero de ellos, siguiendo el flujo de operación definido en los casos de uso (4.1.1), será lo que un usuario arbitrario se encuentre cuando pretenda acceder al sistema por primera vez. Allí, el usuario tendrá la capacidad de registrarse, almacenar sus datos y registrar algún dispositivo VIMS, si lo tiene. Como el servidor se ejecuta en una nube alquilada, se plantea un modelo de suscripción libre en donde un usuario puede decidir pagar una cuota mensual para acceder a las funcionalidades del servidor.

Por otra parte, el visualizador de datos será la pantalla habitual que se encuentre un usuario ya registrado. Allí se mostrará de forma agrupada y ordenada los datos almacenados hasta el momento de los vehículos asociados a su cuenta (con una estructura similar a la mostrada en la figura 3.1). Además del acceso a los datos, el usuario podrá configurar sus preferencias, modalidad de suscripción, dispositivos asociados, etc.

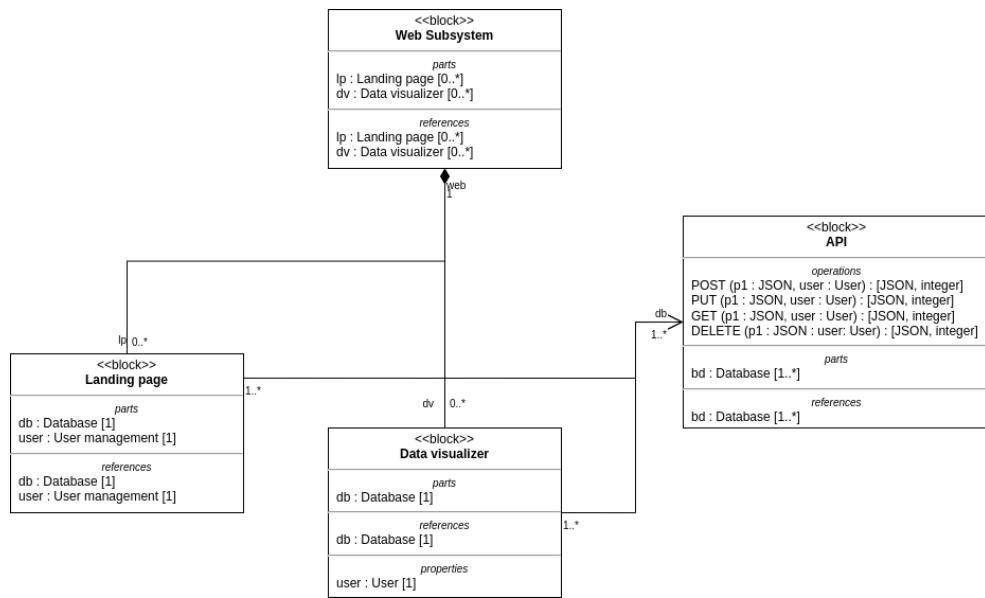


Figura 4.15: Subsistema del servidor que conforma la arquitectura web del mismo.

El subsistema de los datos (figura 4.16) cuenta con múltiples componentes que lo conforman. El más básico de ellos que es usado por los demás es la base de datos. Las operaciones definidas para dicha base de datos son CRUD. La particularidad de la base de datos es que cada “instancia” tiene asociada una cuenta de usuario, de manera que las operaciones a realizar sobre la misma serán siempre con el usuario como clave principal. Debido a los datos que se van a almacenar (que responden a series temporales), el sistema propuesto de base de datos es una mezcla relacional y no relacional. La primera se usará para almacenar toda la información que identifica a los usuarios y sus características, que es un componente que está definido y es limitado. Además, se almacenarán en esa base de datos toda aquella información relacionada con las estadísticas del viaje, ya que es también una característica inmutable. La segunda se usará para almacenar la información transmitida por los dispositivos VIMS dado que es una serie temporal. Se guardarán también datos derivados de dicha transmisión y otra información de utilidad que pueda ser usada posteriormente para generar perfiles de conducción, estadísticas, tendencias, etc.

Por otra parte, existe otro conjunto de bloques que representan las distintas interacciones que puedan existir con los datos. Hay un bloque genérico llamado “*Data handler*” que engloba las operaciones de escritura de los usuarios y de los datos recibidos por los distintos dispositivos VIMS. Este a su vez se compone del bloque “*User management*” que engloba las operaciones de creación, lectura y eliminación de usuarios. El otro bloque es “*Data RX/TX*” que se encarga de recibir los datos de los dispositivos VIMS mediante el protocolo MQTT. Además, también realizará las transmisiones hacia los dispositivos cuando sea necesario (validar el usuario, actualizaciones de *firmware*, ...).

Finalmente, se encuentra el bloque de la API. En particular, es una API REST que permite realizar diversas operaciones sobre los datos asociados a una cuenta de usuario (se especifican más

detalles en la sección 4.5). Al usar un sistema ampliamente estandarizado permite la interacción directa de diversos sistemas como son VIMS, el servidor en sí y la aplicación móvil.

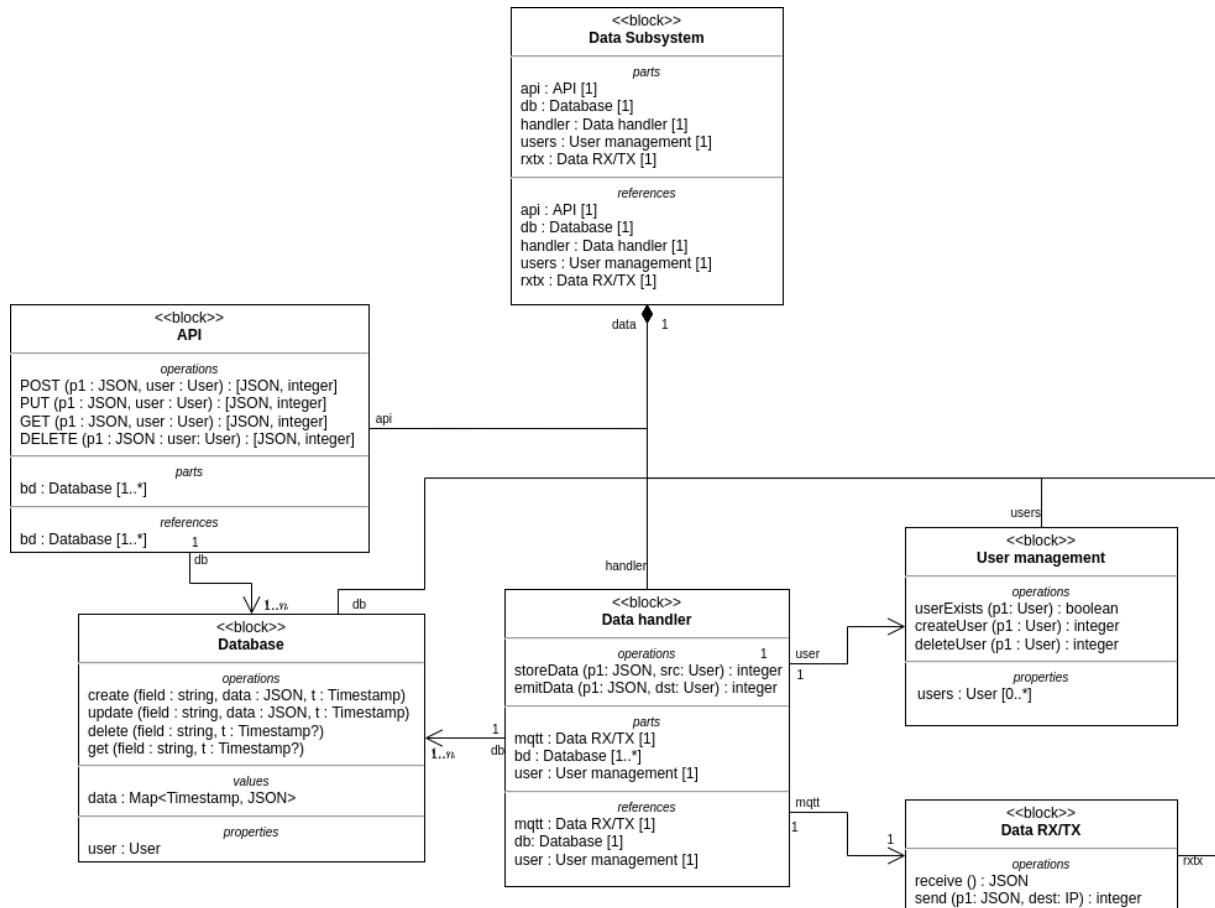


Figura 4.16: Subsistema del servidor que conforma la gestión de datos y usuarios.

Por último, se cuenta con un último diagrama que relaciona cómo funcionan los dos esquemas de bloques anteriores (figura 4.17):

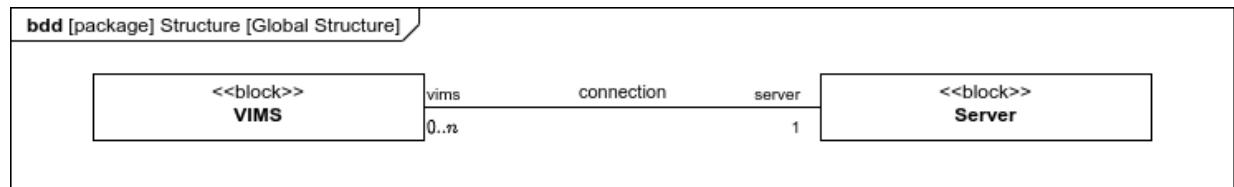


Figura 4.17: Diagrama global del sistema. Como se puede ver, se sigue una arquitectura cliente–servidor.

Como se puede observar, la multiplicidad contempla que hayan múltiples sistemas VIMS

conectados y funcionando contra un único servidor (indiferentemente de que, por razones de escalabilidad, este esté distribuido).

4.2. Diseño *hardware* del sistema

4.3. Diseño 3D del sistema

4.4. Planificación del sistema de tiempo real

La planificación del sistema constituye una fase crítica en el desarrollo del proyecto. Como se mencionó anteriormente en el punto 2.5, se cuentan con múltiples tareas que comparten recursos y que se deben ejecutar con cierta frecuencia para enviar los datos al servidor remoto y detectar los eventos que se produzcan en el sistema.

El diagrama esquemático que representa al sistema con sus tareas es:

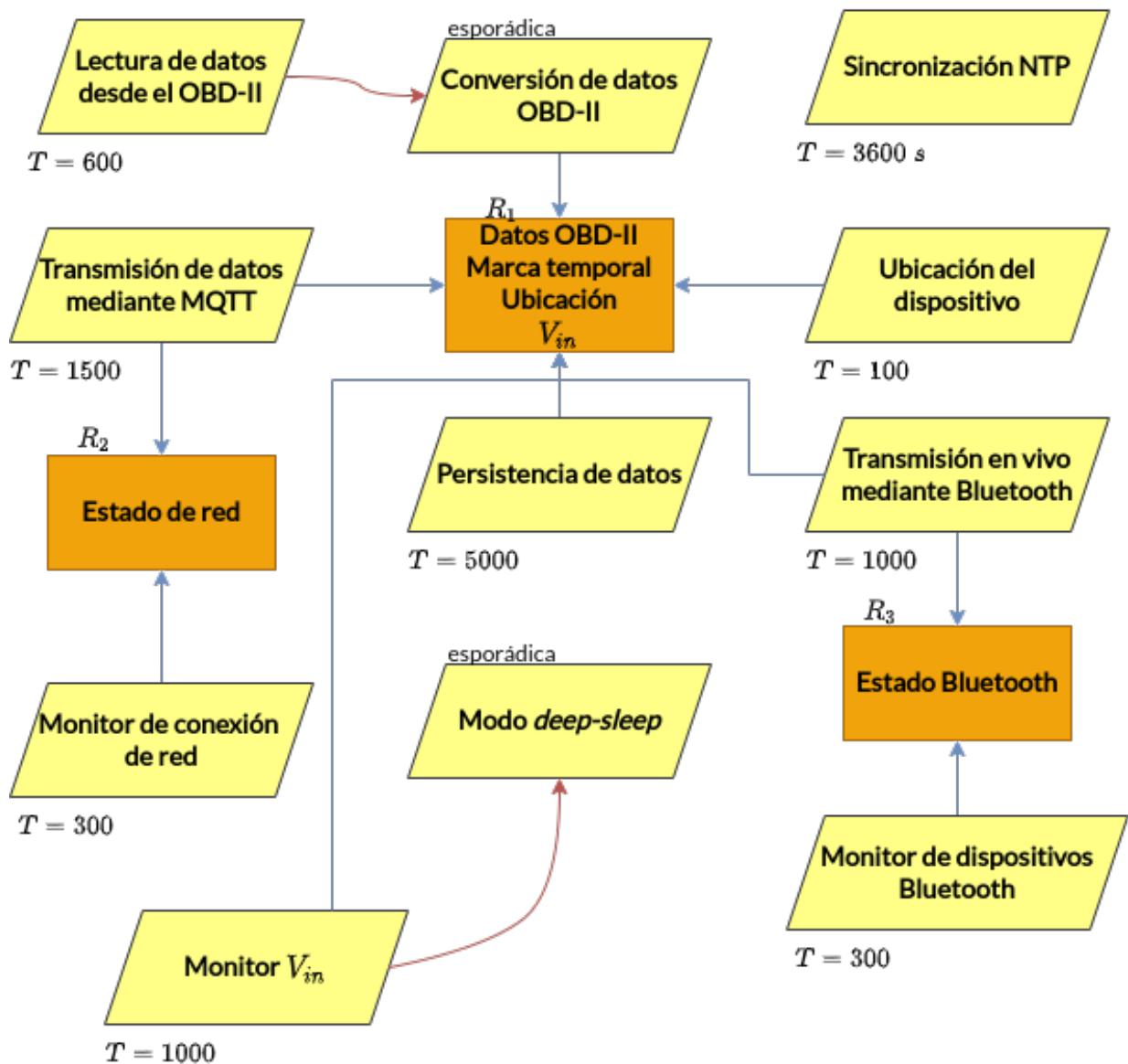


Figura 4.18: Diagrama esquemático del sistema con las tareas y objetos protegidos que lo conforman.

En la figura 4.18 se pueden apreciar las tareas y los objetos protegidos compartidos por las tareas. En particular, las tareas se representan por los trapecios amarillos que van acompañados de sus períodos o se indican si son tareas esporádicas. Los objetos protegidos se representan como rectángulos de color naranja y se indica, mediante flechas, qué tareas acceden a ellos.

Los datos de las tareas vienen representados en la tabla ??:

<i>i</i>	Tarea	Tipo	T_i (ms)	D_i (ms)	C_i	R_1	R_2	R_3
1	Lectura de datos desde OBD-II	<i>P</i>	600	400	C_1			
2	Conversión de datos OBD-II	<i>S</i>	700	200	C_2	r_1^2		
3	Sincronización NTP	<i>P</i>	$3,6 \cdot 10^6$	1000	C_3			
4	Transmisión de datos mediante MQTT	<i>P</i>	1500	1000	C_4	r_1^4	r_2^4	
5	Ubicación del dispositivo	<i>P</i>	100	100	C_5	r_1^5		
6	Persistencia de datos	<i>P</i>	5000	2000	C_6	r_1^6		
7	Transmisión en vivo mediante Bluetooth	<i>P</i>	1000	500	C_7	r_1^7		r_3^7
8	Monitor de conexión de red	<i>P</i>	300	300	C_8		r_2^8	
9	Modo <i>deep-sleep</i>	<i>S</i>	5000	30000	C_9			
10	Monitor V_{in}	<i>P</i>	1000	100	C_{10}	r_1^{10}		
11	Monitor de dispositivos Bluetooth	<i>P</i>	300	300	C_{11}			r_3^{11}

4.5. Diseño *software* del sistema

5 Planificación, costes y tiempo empleado

5.1. Diagramas de Gantt

5.2. Coste de los materiales

5.3. Sueldos propuestos y costes obtenidos

5.4. Contratiempos y tiempo de desarrollo final

6 Conclusiones

6.1. Conclusiones técnicas

6.2. Conocimientos adquiridos y nuevas competencias

6.3. Reflexión final

7 Trabajo pendiente y futuras líneas de trabajo

Bibliografía

- [1] G. Hermawan y E. Husni, «Acquisition, Modeling, and Evaluating Method of Driving Behavior Based on OBD-II: A Literature Survey», *IOP Conference Series: Materials Science and Engineering*, vol. 879, n.º 1, pág. 012 030, jul. de 2020, ISSN: 1757-899X. DOI: 10.1088/1757-899X/879/1/012030.
- [2] «Interfaz de programación de aplicaciones», *Wikipedia, la enciclopedia libre*, jun. de 2021.
- [3] *Comparison of Bluetooth BR/EDR and Bluetooth LE Specifications - MATLAB & Simulink*, <https://www.mathworks.com/help//bluetooth/gs/comparison-of-bluetooth-bredr-and-bluetooth-le.html>.
- [4] «GPS», *Wikipedia, la enciclopedia libre*, jun. de 2021.
- [5] «Internet de las cosas», *Wikipedia, la enciclopedia libre*, jun. de 2021.
- [6] «OBD», *Wikipedia, la enciclopedia libre*, mar. de 2021.
- [7] «OBD-II PIDs», *Wikipedia*, jun. de 2021.
- [8] «Transferencia de Estado Representacional», *Wikipedia, la enciclopedia libre*, ene. de 2022.
- [9] «Trilateración», *Wikipedia, la enciclopedia libre*, feb. de 2021.
- [10] «Vehicle identification number», *Wikipedia*, jul. de 2022.
- [11] «Ford T», *Wikipedia, la enciclopedia libre*, mayo de 2022.
- [12] «Volkswagen Tipo 3», *Wikipedia, la enciclopedia libre*, ene. de 2021.
- [13] *Sistema OBD2 - Historia, Descripción y Futuro*. <https://obd2-elm327.es/sistema-obd2-historia-descripcion-futuro>.
- [14] *Reference Manual: Chapter 10: Emissions Equipment: ECM ALDL*, http://tech.oldsgmail.com/ecm_aldl.pdf
- [15] *Amazon.Com: Elm327 WiFi OBDii Interface OBD2 Puede Bus Escáner Herramienta de Diagnóstico Con Chip Original 25k80 Compatible Con iOS / Android (V2.1) : Automotriz*, <https://www.amazon.es/Interface-herramienta-diagn%C3%B3stico-original-compatible/dp/B07L4926C1>.
- [16] *EOBD - OBD2*, <https://obd2-elm327.es/informacion-sobre-eobd>.
- [17] D. Rimpas, A. Papadakis y M. Samarakou, «OBD-II sensor diagnostics for monitoring vehicle operation and consumption», *Energy Reports, Technologies and Materials for Renewable Energy, Environment and Sustainability*, vol. 6, págs. 55-63, feb. de 2020, ISSN: 2352-4847. DOI: 10.1016/j.egyr.2019.10.018.

- [18] Hussein Ali Ameen, Abd Kadir Mahamad, Sharifah Saon, Mohd Anuaruddin Ahmadon y Shingo Yamaguchi, «Driving Behaviour Identification based on OBD Speed and GPS Data Analysis», *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, n.º 1, págs. 550-569, ene. de 2021, ISSN: 2415-6698.
- [19] *OBD-Codes.Com the Leading OBD-II Trouble Code Resource Site*, <https://www.obd-codes.com/>.
- [20] 42.69€ |LILYGO® TTGO T SIM7000G Placa de Desarrollo SIM T SIM7000G, Módulo Inalámbrico ESP32, WiFi, Bluetooth, Antena GPS, Soporte de Expansión, Carga Solar/Circuitos| - AliExpress, //es.aliexpress.com/item/4000542688096.html?src=ibdm_d03p0558e02r02&sk=&aff_platform
- [21] 1.3€ 10 % de DESCUENTO|Conector Hembra OBD2, Adaptador de Interfaz de Diagnóstico Obd II, Enchufes de Cable Hembra, 16 pines|Conectores| - AliExpress, //es.aliexpress.com/item/4000544285
- [22] Amazon.Com: Aupoko OBD2 OBD-II 16 Pin Hembra Conector Para Abrir Plug Wire Cable de Diagnóstico : Automotriz, <https://www.amazon.com/-/es/aupoko-OBD2-OBD-II-16-Pin-Conector-diagn%C3%A9stico/dp/B06XYP9P91>.
- [23] Vish, *How Much Data Is Created Every Day in 2021? [You'll be shocked!]*, <https://techjury.net/blog/how-much-data-is-created-every-day/>, jun. de 2020.