



*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, 2 de julio de 2022



# Índice general

<b>Índice de figuras</b>	<b>III</b>
<b>Índice de cuadros</b>	<b>v</b>
<b>Definiciones, siglas, y abreviaturas</b>	<b>VIII</b>
<b>1. Introducción</b>	<b>3</b>
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
<b>2. Estructura del proyecto</b>	<b>14</b>
2.1. Estudio de mercado . . . . .	15
2.2. Estudio matemático . . . . .	15
2.3. Diseño <i>software</i> . . . . .	15
2.4. Diseño <i>hardware</i> . . . . .	15
2.5. Análisis de planificabilidad . . . . .	15
<b>3. Especificación de requisitos</b>	<b>16</b>
3.1. Introducción . . . . .	16
3.1.1. Propósito . . . . .	16
3.1.2. Alcance . . . . .	17
3.1.3. Estructura del documento . . . . .	18
3.2. Descripción general del producto . . . . .	18
3.2.1. Perspectiva del producto . . . . .	18
3.2.2. Características de los usuarios finales . . . . .	20
3.2.3. Restricciones generales . . . . .	20
3.2.4. Suposiciones y dependencias . . . . .	21
3.3. Requisitos específicos . . . . .	22
3.3.1. Requisitos de usuario . . . . .	22
3.3.2. Requisitos funcionales . . . . .	26
3.3.3. Requisitos no funcionales . . . . .	29
3.3.4. Requisitos de interfaces externas . . . . .	31
3.3.5. Restricciones de desarrollo . . . . .	31

3.3.6. Requisitos de entorno físico . . . . .	31
<b>4. Diagramas y diseño</b>	<b>32</b>
4.1. Diagramas que modelan el sistema . . . . .	32
4.2. Diseño <i>hardware</i> del sistema . . . . .	32
4.3. Diseño 3D del sistema . . . . .	32
4.4. Diseño <i>software</i> del sistema . . . . .	32
4.5. Planificación del sistema de tiempo real . . . . .	32
<b>5. Planificación, costes y tiempo empleado</b>	<b>33</b>
5.1. Diagramas de Gantt . . . . .	33
5.2. Coste de los materiales . . . . .	33
5.3. Sueldos propuestos y costes obtenidos . . . . .	33
5.4. Contratiempos y tiempo de desarrollo final . . . . .	33
<b>6. Conclusiones</b>	<b>34</b>
6.1. Conclusiones técnicas . . . . .	34
6.2. Conocimientos adquiridos y nuevas competencias . . . . .	34
6.3. Reflexión final . . . . .	34
<b>7. Trabajo pendiente y futuras líneas de trabajo</b>	<b>35</b>
<b>Bibliografía</b>	<b>36</b>
<b>A. Validación de requisitos</b>	<b>37</b>
<b>B. Diagramas que modelan el sistema</b>	<b>38</b>
B.1. Casos de uso . . . . .	38
B.2. Diagramas de bloques . . . . .	50

# Índice de figuras

1.1.	Ford modelo T del 1927 – de Rmhermen [8]. . . . .	4
1.2.	Volkswagen Tipo III, modelo de inyección de 1969 – de OSX - Trabajo propio [9]. . . . .	5
1.3.	Conector <i>Computerized Assembly Line Diagnostics Link</i> (ALDL), creado por <i>General Motors</i> antes de estandarizar OBD-II [11]. . . . .	6
1.4.	Controlador ELM327 que cuenta con antenas Wi-Fi y Bluetooth para un acceso remoto [12]. . . . .	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 [15]. . . . .	10
1.7.	Esquema general que modela el modo de funcionamiento del sistema. . . . .	11
3.1.	Representación de los datos en la interfaz propuesta – fuente: <a href="https://grafana.csselectronics.stellarhosted.com/d/6qvL90vMz/css-playground-obd2?orgId=1">https://grafana.csselectronics.stellarhosted.com/d/6qvL90vMz/css-playground-obd2?orgId=1</a> . . . . .	26
B.1.	Caso de uso 01 – <i>autenticación</i> . . . . .	38
B.2.	Casos de uso 02 – <i>generación y transmisión de datos</i> . . . . .	39
B.3.	Caso de uso 03 – <i>generación de estadísticas</i> . . . . .	41
B.4.	Caso de uso 04 – <i>envío de notificaciones</i> . . . . .	42
B.5.	Caso de uso 05 – <i>visualización en tiempo real</i> . . . . .	44
B.6.	Caso de uso 06 – <i>generación de eventos</i> . . . . .	46
B.7.	Casos de uso 07* – <i>envío, almacenamiento y visualización de geolocalización</i> . . . . .	47
B.8.	Diagrama de bloques que modela el conjunto de la placa <i>Vehicle IoT Metrics System</i> (VIMS) y sus componentes. . . . .	50
B.9.	Diagrama de bloques que modela al servidor y sus components. . . . .	51

B.10. Diagrama global del sistema. Como se puede ver, se sigue una arquitectura cliente-servidor. . . . .	51
---	----

# **Índice de cuadros**

3.1. Opciones ofrecidas a los encuestados. Se han escogido diversas opciones que se encuentran entre los datos habituales generados por un vehículo. . . . .	23
3.2. Puntuaciones obtenidas de forma general, por la población al completo (conductores y no conductores). . . . .	24
3.3. Puntuaciones obtenidas de forma general, por la población (excluídos los no conductores). . . . .	24

# Definiciones, siglas, y abreviaturas

**ADAS** *Advanced Driver Assistance Systems*

**ALDL** *Computerized Assembly Line Diagnostics Link*

**API** *Application Programming Interface*

**BLE** *Bluetooth Low Energy*

**CAN** *Controller Area Network*

**CARB** *California Air Resources Board*

**EOBD** *European On-Board Diagnostics*

**GB** *gigabyte*

**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*

**OBD** *On-Board Diagnostics*

**PAN** *Personal Area Network*

**PFM** Proyecto Fin de Máster

**PID** *Parameter ID*

**RPM** Revoluciones Por Minuto

**SAE** *Society of Automotive Engineers*

**UART** *Universal Asynchronous Receiver–Transmitter*

**VIMS** *Vehicle IoT Metrics System*

- *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.
- 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. misma red o malla. Tiene un alcance muy limitado, de unos pocos metros por lo general.
- *gigabyte* (GB) – unidad de almacenamiento de información equivalente a  $10^9$  bytes.
- *Global Positioning System* (GPS) – sistema que permite posicionar cualquier objeto con una precisión de hasta centímetros usando cuatro o más satélites y trilateración [3].
- *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 [4].
- *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.
- OBD – sistema de diagnóstico a bordo de vehículos que
- PAN – redes destinadas a la comunicación entre dispositivos en una 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 [5].
- *Parameter ID* (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 [6].
- *Smog* – nube baja formada por dióxido de carbono, hollín, humo y polvo en suspensión que se forma sobre las grandes ciudades o núcleos industriales.
- trilateración – método matemático que permite determinar las posiciones relativas de objetos usando la geometría de los triángulos [7].

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  $\mu$ 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 [8].

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 pantallas y 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 [9].

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, una única transmisión en un sentido (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 [10].

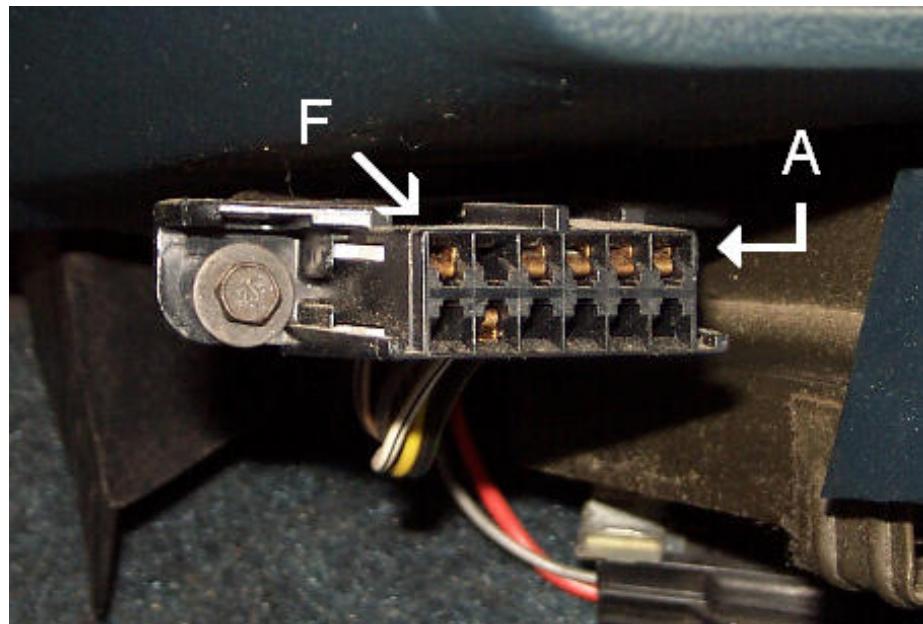


Figura 1.3.: Conector ALDL, creado por *General Motors* antes de estandarizar OBD-II [11].

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 [10].

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 [10].

Esta medida llegaría a Europa en 1998, según la Directiva 98/69EG, que obligaba a todos los vehículos europeos a incluir su propio conector. Específicamente, los automóviles de gasolina debían empezar a equiparlo en los modelos del año 2000; año 2003 para vehículos diésel; y año 2005 para camiones [10].

## 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 [10], figura 1.4):



Figura 1.4.: Controlador ELM327 que cuenta con antenas Wi-Fi y Bluetooth para un acceso remoto [12].

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 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 incluir<sup>1</sup>, 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.

<sup>1</sup>Se 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) [10]. Otra característica innovadora es que el sistema europeo registra cuántos kilómetros se han recorrido desde que ha aparecido un defecto [13].

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/presentar. Sin embargo, tras una breve investigación, la mayoría de las casuísticas buscaba directamente monitorizar en el momento el estado del vehículo para obtener información relativa a los consumos, contaminación, etc.

Por ejemplo, en la investigación llevada a cabo por Rimpas *et al.* [14], 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:

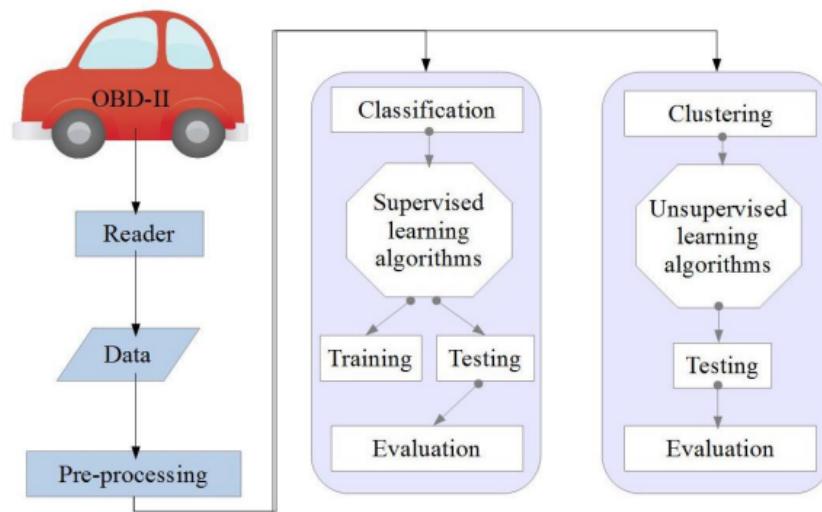


Figura 1.5.: Esquema seguido para determinar los hábitos de conducción usando el OBD-II [1].

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.* [15] 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):

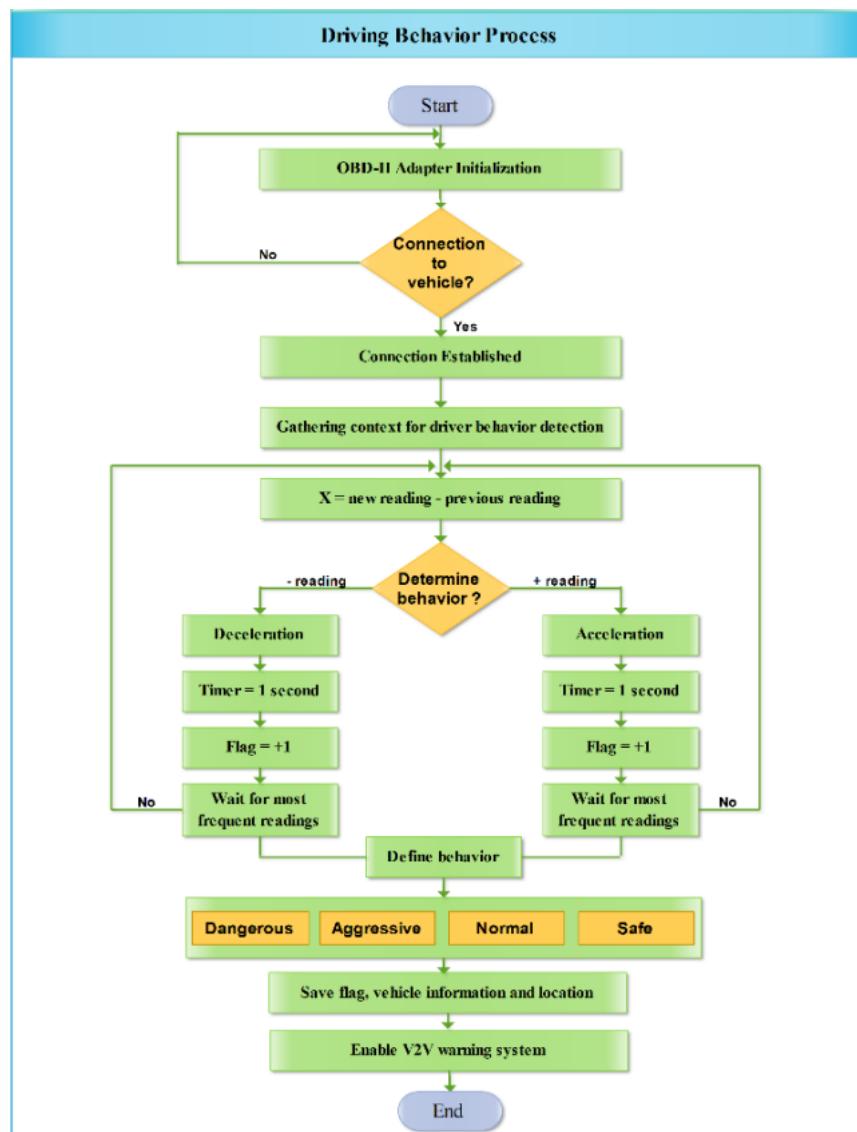


Figura 1.6.: Flujo de análisis para determinar el comportamiento al volante de un conductor [15].

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

- **Peligroso**, para una aceleración en general superior a  $7 \text{ m/s}^2$ .
- **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):

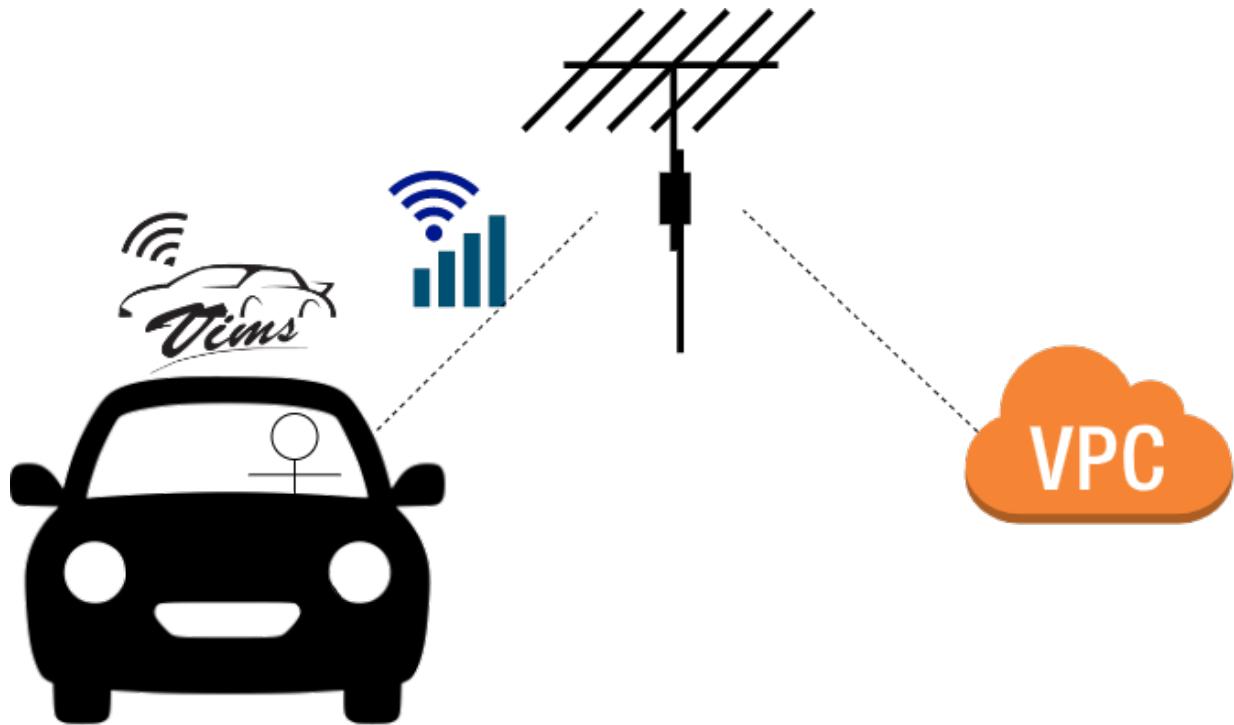


Figura 1.7.: Esquema general que modela el modo de funcionamiento del sistema.

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 MIT<sup>2</sup>.

### 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 querrían tener en su vehículo. Los primeros sirvieron de grupo de control, los segundos para aumentar la entropía de los datos obtenidos. El primer análisis realizado se detalla en la sección 3.3.1, de la especificación de requisitos. Posteriormente, en el punto 2.1 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 distintos diagramas que modelan el sistema, tanto lógicos como de diseño. Esta parte fue crucial

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

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. Sin embargo, esta etapa no se ha podido completar por distintos contratiempos que se comentan en más detalle en el punto 5.

## 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 de a quienes no lo fueran.

Esto permite obtener datos más fidedignos y acotados según la población que respondiera.

## 2.2. Estudio matemático

### 2.3. Diseño *software*

### 2.4. Diseño *hardware*

### 2.5. Análisis de planificabilidad

# 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 geográfico<sup>1</sup> 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 [16]) 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).

---

<sup>1</sup>Dentro 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. (B).

## 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 <sup>a</sup> , 2 <sup>a</sup> , ...)	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 [6]. 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:

#### Conectividad

- 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 [15].

## 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**

**4.1. Diagramas que modelan el sistema**

**4.2. Diseño *hardware* del sistema**

**4.3. Diseño 3D del sistema**

**4.4. Diseño *software* del sistema**

**4.5. Planificación del sistema de tiempo real**

## **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] «GPS», *Wikipedia, la enciclopedia libre*, jun. de 2021.
- [4] «Internet de las cosas», *Wikipedia, la enciclopedia libre*, jun. de 2021.
- [5] «OBD», *Wikipedia, la enciclopedia libre*, mar. de 2021.
- [6] «OBD-II PIDs», *Wikipedia*, jun. de 2021.
- [7] «Trilateración», *Wikipedia, la enciclopedia libre*, feb. de 2021.
- [8] «Ford T», *Wikipedia, la enciclopedia libre*, mayo de 2022.
- [9] «Volkswagen Tipo 3», *Wikipedia, la enciclopedia libre*, ene. de 2021.
- [10] *Sistema OBD2 - Historia, Descripción y Futuro*. <https://obd2-elm327.es/sistema-obd2-historia-descripcion-futuro>.
- [11] *Reference Manual: Chapter 10: Emissions Equipment: ECM ALDL*, [http://tech.oldsgmail.com/ecm\\_aldl.pdf](http://tech.oldsgmail.com/ecm_aldl.pdf)
- [12] *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%A9stico-original-compatible/dp/B07L4926C1>.
- [13] *EOBD - OBD2*, <https://obd2-elm327.es/informacion-sobre-eobd>.
- [14] 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.
- [15] 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.
- [16] 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.

## **A. Validación de requisitos**

## B. Diagramas que modelan el sistema

### B.1. Casos de uso

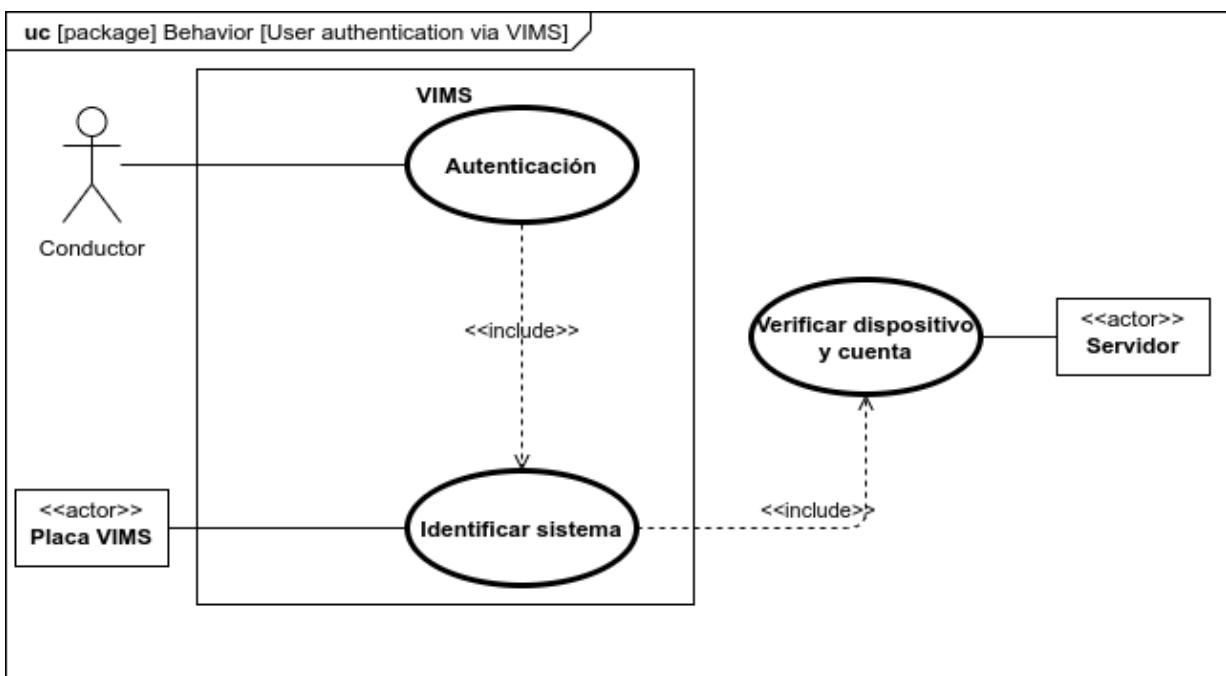


Figura B.1.: Caso de uso 01 – autenticación.

01	<i>Autenticación</i>	
<b>Descripción</b>	La placa identificará de forma inequívoca al conductor (usuario) y a sí misma frente al servidor.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	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.
	4	La placa almacena la información del usuario y finaliza el proceso de inicio de sesión.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	La placa no cuenta con conexión a la red o el servidor no está disponible.
	3	La cuenta del usuario no existe.
<b>Resolución</b>	La placa funciona en modo desconectado, es decir, no transmite datos al servidor.	

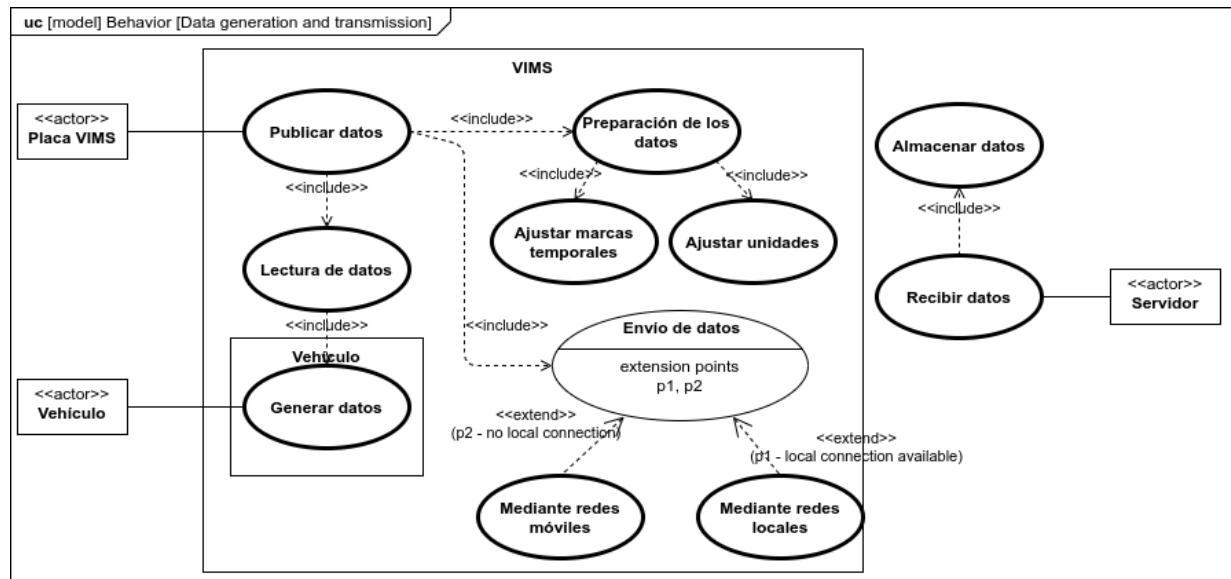


Figura B.2.: Casos de uso 02 – generación y transmisión de datos.

02	<i>Generación y transmisión de datos</i>	
<b>Descripción</b>	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.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	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.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	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.
<b>Resolución</b>	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.	

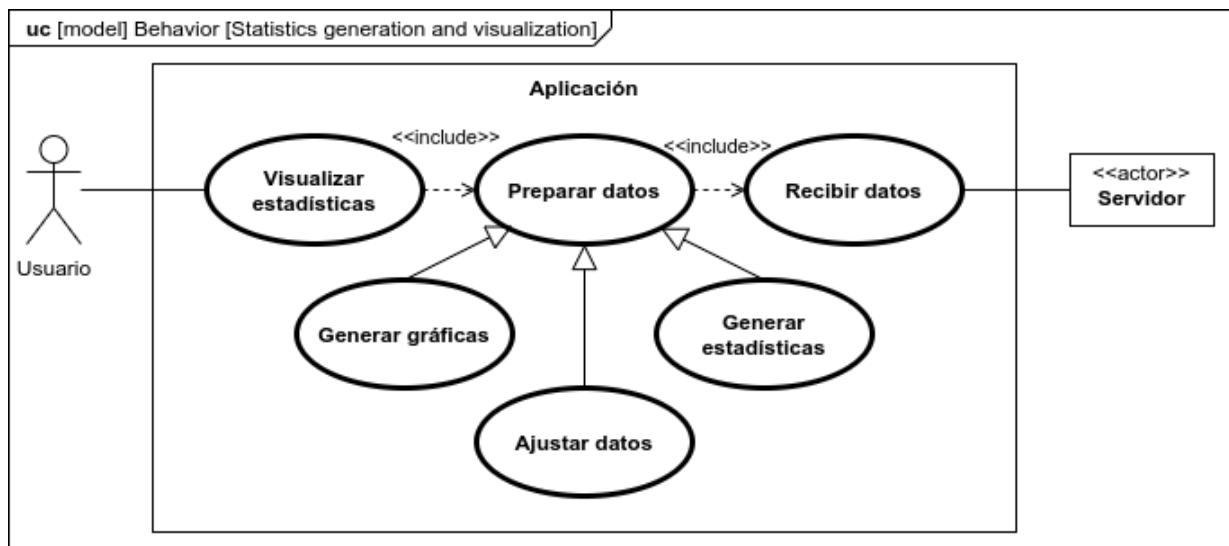


Figura B.3.: Caso de uso 03 – generación de estadísticas.

03	<i>Generación de estadísticas</i>	
<b>Descripción</b>	El servidor, en conjunción con el resto de elementos del sistema, preparará los datos para generar información estadística útil para el usuario.	
<b>Secuencia normal</b>	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.
<b>Excepciones</b>	Paso	Acción
	1.1	El usuario no está autenticado.
	1.2	El usuario no cuenta con ningún dispositivo VIMS asociado.
<b>Resolución</b>	2 Todavía no se ha registrado ningún dato.	
	Se notifica al usuario de este suceso y se le sugiere crear una cuenta.	

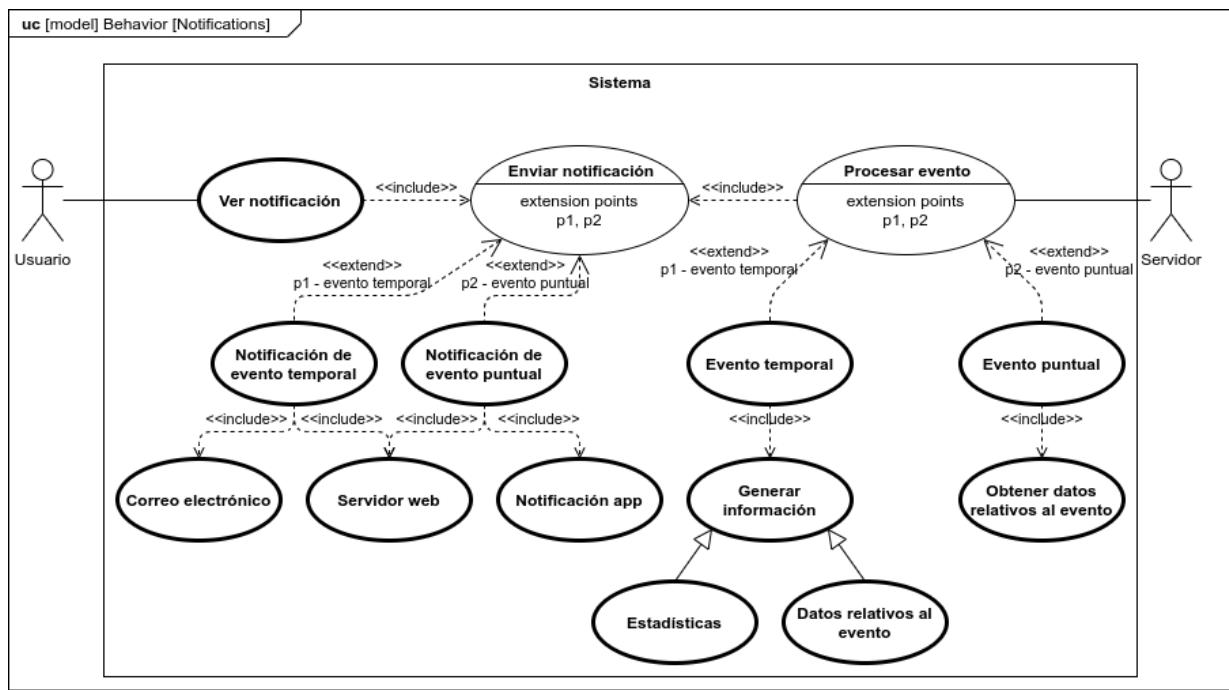
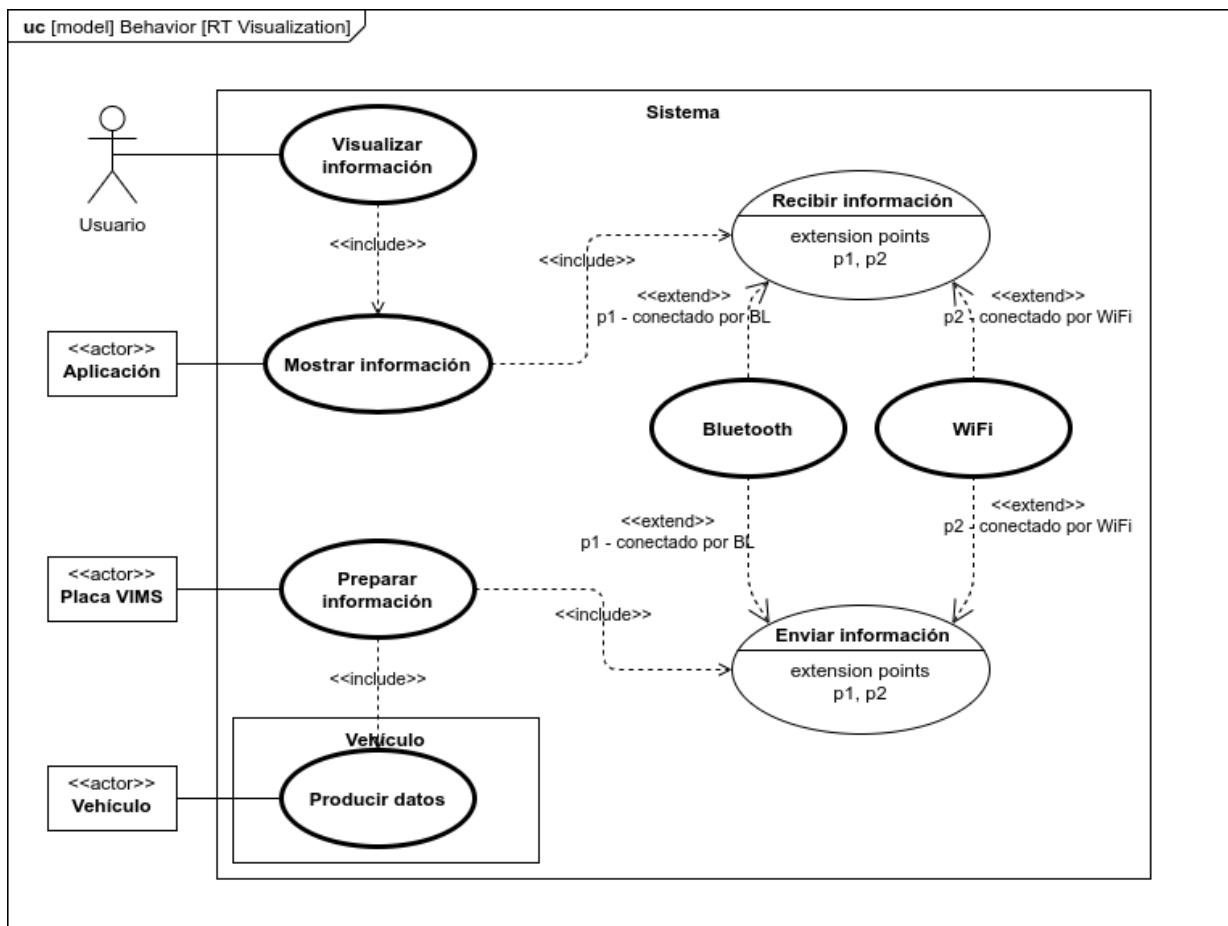


Figura B.4.: Caso de uso 04 – envío de notificaciones.

04	<i>Envío de notificaciones</i>	
<b>Descripción</b>	El servidor, en conjunción con el resto de elementos del sistema, detecta ciertos eventos y actúa generando una notificación que envía al usuario.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	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.

---

Figura B.5.: Caso de uso 05 – *visualización en tiempo real*

05	<i>Visualización en tiempo real</i>	
<b>Descripción</b>	Un usuario podrá visualizar información en tiempo real sobre su vehículo mediante la aplicación.	
<b>Secuencia normal</b>	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.
<b>Excepciones</b>	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.
<b>Resolución</b>	La placa no realiza ninguna transmisión de información hasta que no haya una conexión disponible.	

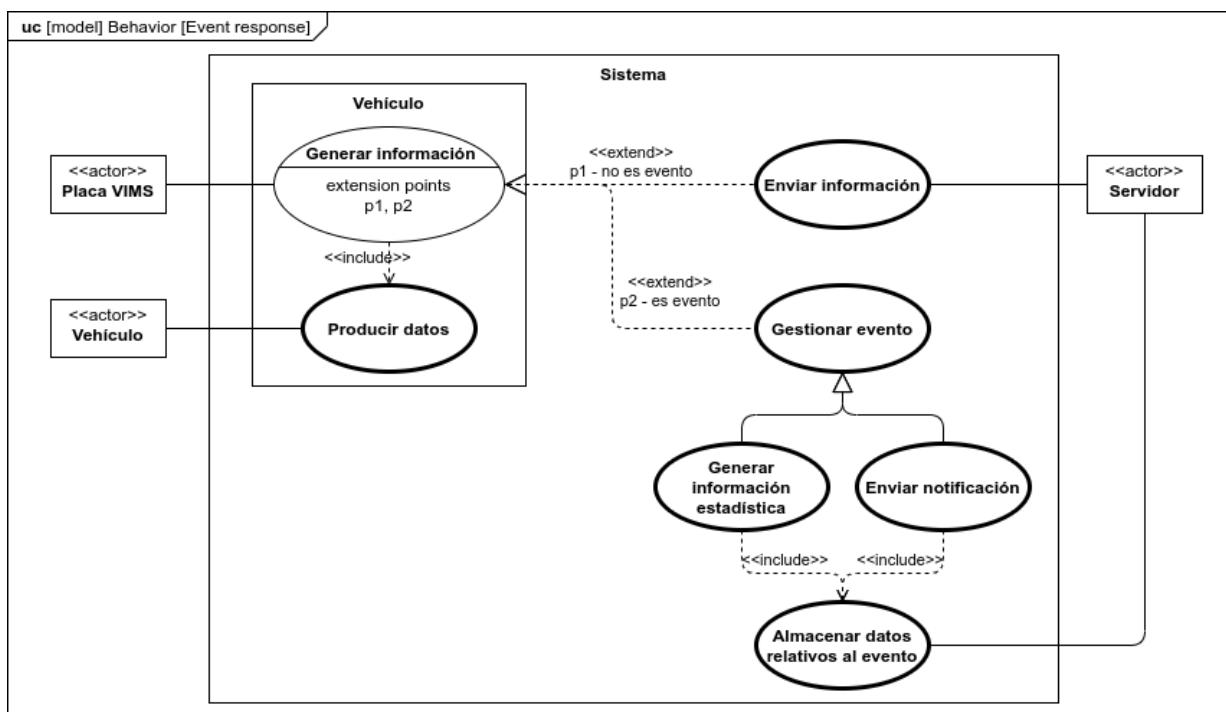


Figura B.6.: Caso de uso 06 – generación de eventos

	<b>06</b>	<i>Generación de eventos</i>	
<b>Descripción</b>	El dispositivo VIMS, a la hora de enviar datos, podrá producir eventos según el tipo de información que haya de enviar.		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	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.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	No hay ninguna cuenta asociada al dispositivo.	
	1.1	No hay conexión a Internet por parte del dispositivo VIMS.	
<b>Resolución</b>	La placa funciona en modo desconectado, es decir, no transmite datos al servidor.		

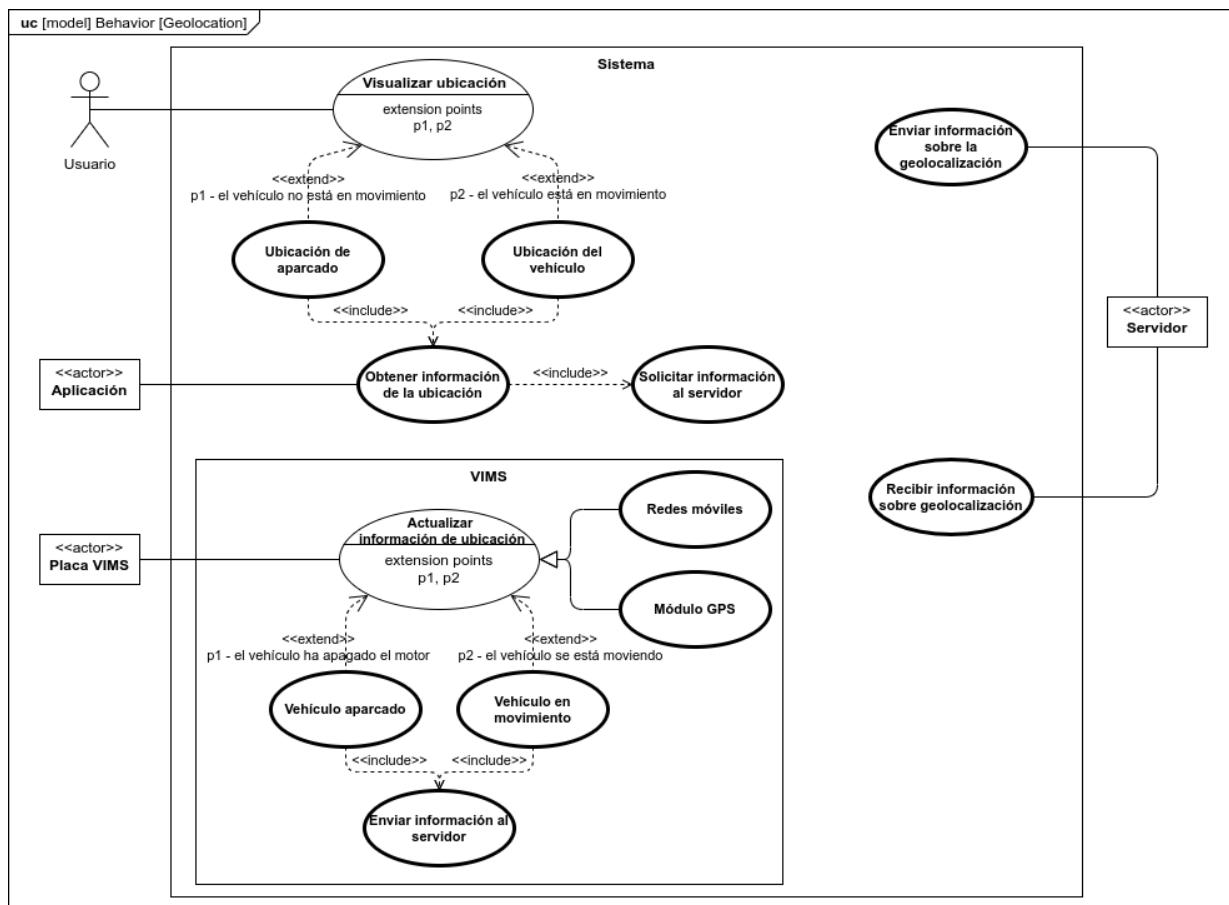


Figura B.7.: Casos de uso 07\* – envío, almacenamiento y visualización de geolocalización

<b>07.1</b>	<i>Visualización de ubicación</i>	
<b>Descripción</b>	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.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	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.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	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.
<b>Resolución</b>	La placa funciona en modo desconectado, es decir, no transmite datos al servidor.	

<b>07.2</b>	<i>Envío de ubicación</i>	
<b>Descripción</b>	La placa VIMS enviará la información relativa al vehículo al servidor.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	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.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	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.
<b>Resolución</b>	La placa funciona en modo desconectado, es decir, no transmite datos al servidor.	

07.3	<i>Almacenamiento de los datos de ubicación</i>	
<b>Descripción</b>	El servidor almacenará y gestionará los datos de ubicación recibidos por los múltiples dispositivos VIMS.	
<b>Secuencia normal</b>	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.
	2	El servidor ofrece los datos de ubicación a través de la API.
<b>Excepciones</b>	Paso	Acción
	1	No hay ninguna cuenta asociada al dispositivo.
<b>Resolución</b>	La placa funciona en modo desconectado, es decir, no transmite datos al servidor.	

## B.2. Diagramas de bloques

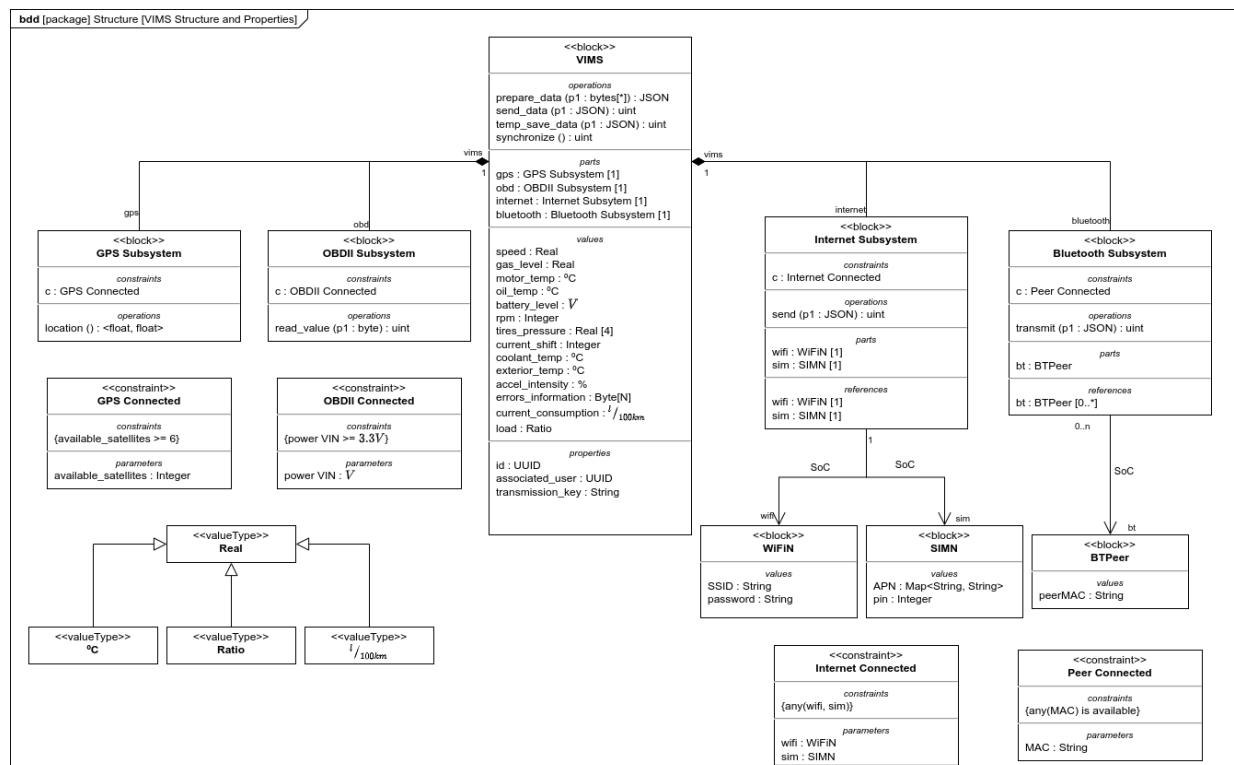


Figura B.8.: Diagrama de bloques que modela el conjunto de la placa VIMS y sus componentes.

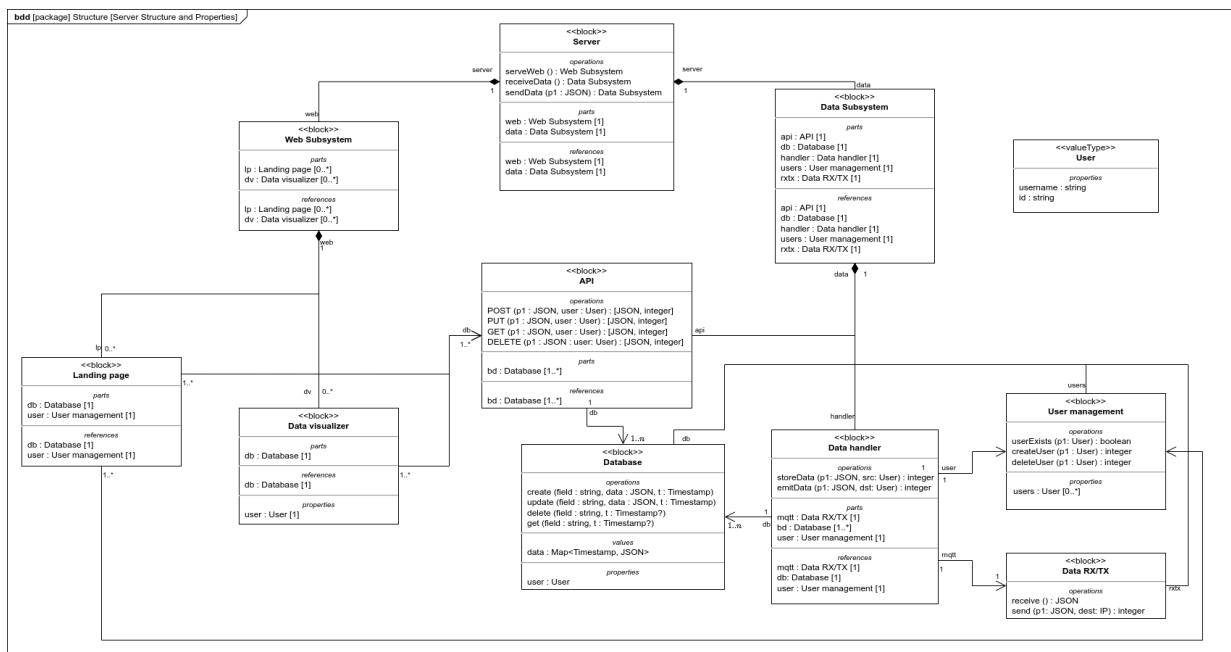


Figura B.9.: Diagrama de bloques que modela al servidor y sus componentes.

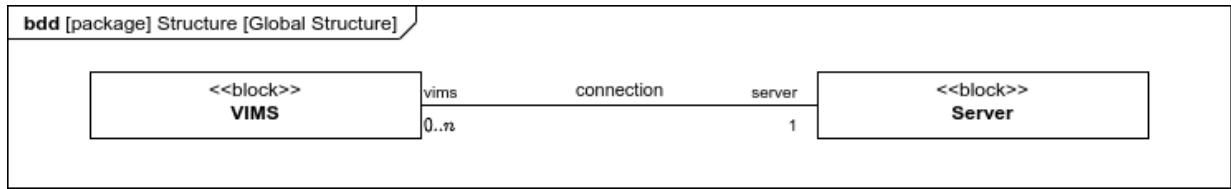


Figura B.10.: Diagrama global del sistema. Como se puede ver, se sigue una arquitectura cliente-servidor.