



*Universidad Tecnológica Nacional*  
*Facultad Regional Villa María*  
*Ingeniería Electrónica*

## **Gestión de flota por GPS y GPRS**

Autor/es:

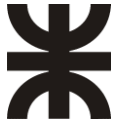
Vidal, Matias Mariano Oscar

Tutor:                   Ing. Fabián Marcelo Sensini

Director:               Ing. José Luis Catalano

Co-Director:          Esp. Ing. Héctor Diego Ferrari

**Fecha de Defensa: 8 de marzo de 2024**



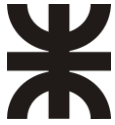


## Dedicatorias

*A mis padres y hermana, por su apoyo y ánimo incansables a lo largo de mi carrera, brindándome la fortaleza y la inspiración necesarias para superar cada desafío.*

*A mi pareja, por su amor incondicional, paciencia infinita y por estar siempre a mi lado, ofreciendo apoyo y fuerza en cada etapa de este desafío.*

*A los amigos con los que compartí esta etapa, por su invaluable compañía y aliento en los momentos más desafiantes y por las risas compartidas que iluminaron el camino.*



## Agradecimientos

*A mi familia, quienes con esfuerzo me permitieron tener esta posibilidad. Por su apoyo inquebrantable y motivación durante este proceso, siendo mi fuente de ánimo y fuerza en cada etapa.*

*A la Universidad Tecnológica Nacional Facultad Regional de Villa María, por brindar un entorno enriquecedor y las herramientas necesarias que han facilitado mi aprendizaje y desarrollo académico.*

*A los docentes de esta carrera, por su dedicación y la ayuda en la realización del proyecto, y por compartir conocimientos que han sido clave en mi formación.*



## Resumen

En el presente proyecto, se desarrolló un sistema de rastreo satelital de vehículos, el cual utiliza GPS y GPRS. Para obtener la ubicación actual y la información de diversos sensores del vehículo, se diseñó y creó un dispositivo que, mediante comunicación GPS, obtiene las coordenadas actuales y, mediante un microcontrolador, obtiene las mediciones de los sensores. Estos datos son enviados usando GPRS hacia una base de datos y, a través de comunicación CAN BUS, a otro dispositivo ubicado dentro del vehículo, que permite monitorear los parámetros en tiempo real en una pantalla LCD.

Para visualizar la información previamente mencionada, se desarrolló un dispositivo que recibe estos datos mediante comunicación CAN BUS y muestra en pantalla los últimos datos obtenidos, entre las que se encuentran las revoluciones por minutos del motor, la velocidad actual, y datos estadísticos respecto a distancia recorrida y tiempo de viaje.

Se implementó una base de datos en un servidor web, donde se almacenan los datos recibidos. Además, se desarrollaron aplicaciones web y móvil para observar la ubicación actual de los vehículos y recibir alertas en caso de detectarse algún incidente.

**Palabras Claves:** *Rastreo vehicular, Aplicación móvil, GPS, GPRS, CAN BUS.*

## Abstract

In this project, a vehicle satellite tracking system was developed, which uses GPS and GPRS. To obtain the current location, and values from several vehicle sensors, a device was designed and created that acquires the current coordinates through GPS communication and, through a microcontroller collects sensor measurements. These data are sent via GPRS to a database and, through CAN BUS to another device inside the vehicle, which allows real-time monitoring of parameters on an LCD screen.

To display the previously mentioned information, a device was developed that receives this data through CAN BUS communication, and shows the latest data available, including engine revolutions per minute, current speed, and statistical data with respect to distance traveled and travel time.

A database was implemented in a web server, where the received data is stored. In addition, web and mobile applications were developed to observe the current location of the vehicles and to receive alerts in case of any incident detected.

**Keywords:** *Vehicle tracking, Mobile application, GPS, GPRS, CAN BUS.*



## ÍNDICE

<b>Título</b>	<b>Página</b>
Dedicatorias.....	3
Agradecimientos.....	4
Resumen.....	5
Abstract.....	5
Introducción.....	7
Análisis del problema.....	7
Análisis de sistemas existentes.....	7
Descripción de las actividades del proyecto.....	7
Objetivos.....	8
Diseño del Proyecto.....	9
Revisión de requerimientos y parámetros de operación.....	9
Selección de componentes y dispositivos.....	9
Dispositivo de rastreo.....	9
Dispositivo de visualización.....	11
Diagrama en bloques del dispositivo.....	14
Diagramas de flujo.....	16
Evaluación Final del Sistema.....	22
Preparación del prototipo con materiales finales.....	22
Dispositivo de rastreo.....	22
Dispositivo de visualización.....	30
Aplicativo dispositivo de visualización.....	33
Aplicación web.....	34
Aplicación Android.....	37
Montaje y ensayo real del prototipo.....	40
Análisis del sistema en campo.....	44
Sistema GPS.....	44
Sistema GPRS.....	44
Sensores.....	44
Dispositivo de visualización.....	46
Aplicaciones web y móvil.....	46
Presentación final del prototipo.....	48
Conclusiones.....	50
Anexo I – esquemático dispositivo de rastreo.....	51
Bibliografía.....	52



## Introducción

### **Análisis del problema**

Para las empresas de transporte, ya sea de carga o de pasajeros, llevar un control de sus unidades es fundamental para poder brindar un buen servicio, mejorar sus tiempos, detectar mal funcionamientos para poder realizar mantenimientos en tiempo y forma, lo cual evitaría gastos futuros. Poder tener una rápida respuesta en caso de inconvenientes también es fundamental, para poder así brindar rápida respuesta a sus pasajeros o cargas.

Este proyecto surge de esta necesidad, proponiendo así un sistema que permita realizar estos monitoreos en tiempo real, mediante el desarrollo de un dispositivo que sea capaz de enviar remotamente esta información de utilidad hacia una base de datos, y poder observar esta información desde un dispositivo móvil o PC.

### **Análisis de sistemas existentes**

A fecha de presentación de este proyecto, existen distintos tipos de rastreadores GPS disponibles en el mercado, donde la mayoría realiza únicamente un seguimiento satelital de las coordenadas, sin tener en cuenta otra información que podría ser de utilidad para realizar un mejor control, incluyendo parámetros como temperatura del motor y nivel de batería.

### **Descripción de las actividades del proyecto**

Para el desarrollo del proyecto, se partió por el planteo de los requerimientos de operación, lo que permitió escoger los componentes necesarios que cumplan con lo planteado. Se diseñó el circuito esquemático, el circuito impreso, y se creó un prototipo que consta de dos partes: dispositivo de rastreo y dispositivo de visualización.

Se planteó e implementó el algoritmo que deben ejecutar estas partes, creando un firmware para el dispositivo de rastreo utilizando lenguaje C. Este firmware permite obtener información de diversos sensores, así también obtener información mediante GPS, procesar la información y enviarla mediante GPRS a un servidor, y mediante CAN BUS al dispositivo de visualización. Para observar la información en este dispositivo, se escribió un script en Python, el cual permite recibir información mediante CAN BUS y mostrarla en pantalla.

Se implementó una base de datos SQL en un servidor web, que recibe la información del dispositivo de rastreo y la almacena.

Para poder observar los datos, se desarrolló una aplicación web usando PHP, HTML y JavaScript, y una aplicación Android utilizando Android Studio, las cuales toman los datos del servidor, y permiten visualizar la información más reciente.

Se procedió a realizar una prueba de funcionamiento del dispositivo en un vehículo, y para el montaje final, se montó los dispositivos en carcasas, diseñándose mediante software de modelado en 3D una carcasa para el dispositivo de visualización.



## Objetivos

- Desarrollar un dispositivo de bajo costo que permita realizar el monitoreo de un vehículo.
- Implementar una base de datos que pueda almacenar la última información disponible.
- Crear una interfaz gráfica simple y accesible para una aplicación web y una aplicación móvil.
- Obtener una rápida alerta ante caso de imprevistos.
- Desarrollar un sistema estable.
- Diseñar e implementar una interfaz gráfica simple y amigable para el conductor del vehículo.



## Diseño del Proyecto

### **Revisión de requerimientos y parámetros de operación**

Para cumplir con los objetivos previamente planteados, se llevó a cabo un análisis de los requisitos necesarios para lograr un sistema económico y versátil, apto para ser empleado en una amplia gama de vehículos. Se estableció que el dispositivo debería ser compatible tanto con vehículos de alimentación 12V como de 24V. Además, se optó por la comunicación GPRS para la transmisión de datos, atribuyendo su elección a su bajo costo de implementación y operación.

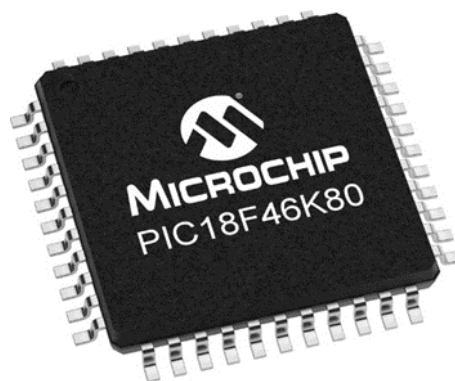
Para facilitar la comunicación entre dispositivos dentro del vehículo, se seleccionó la comunicación CAN BUS, debido a su alta robustez y su alta inmunidad al ruido eléctrico característico en los entornos vehiculares.

Para obtener un dispositivo compacto, se prefirió que los componentes fueran de montaje superficial, optimizando así el espacio y simplicidad del montaje.

### **Selección de componentes y dispositivos**

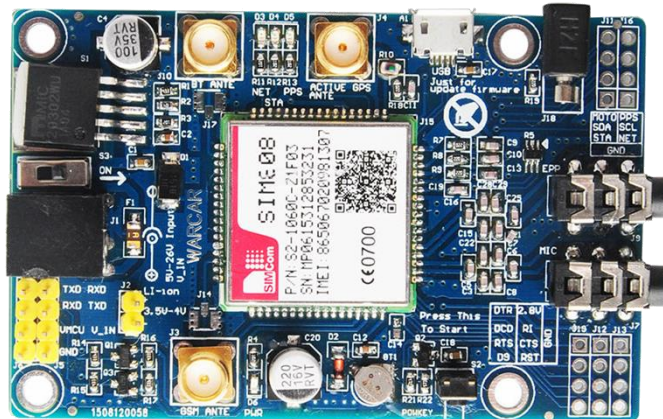
#### **Dispositivo de rastreo**

Microcontrolador PIC18F46K80: Fue seleccionado como procesador central debido a que cumple con los requisitos de poseer puerto de comunicación CAN BUS, ser de bajo costo y tener disponibilidad en el mercado. Este microcontrolador ofrece comunicación UART e I<sup>2</sup>C que es de importancia al momento de interconectar otros dispositivos o módulos. Adicionalmente, dispone de 11 entradas analógicas que permiten el monitoreo de diversos sensores externos. Una ventaja de este dispositivo es que cuenta con oscilador interno, que descarta la necesidad de utilizar osciladores o cristales externos, resultando en una economía de espacio en el montaje.



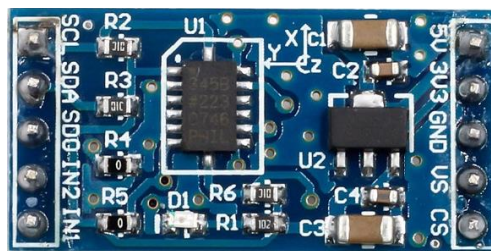
**Fig. 1** - Microcontrolador PIC18F46K80.

SIM808: La elección del módulo de evaluación SIM808 se fundamenta en su integración dual de capacidades, albergando tanto un receptor GPS como una comunicación GPRS en un único módulo, optimizando así el espacio y el funcionamiento. Su flexibilidad en la tensión de alimentación lo hace apto para diferentes rangos de tensión de alimentación, admitiendo un máximo de 26V de tensión de entrada. Para su operación, se emplean comandos AT a través de una única comunicación UART, simplificando así su interfaz y facilitando la interacción con el microcontrolador.



**Fig. 2 - Módulo GPS/GPRS SIM808.**

Módulo sensor acelerómetro ADXL345: Este módulo fue seleccionado por su capacidad de medir aceleraciones en 3 ejes (X, Y, Z) con una resolución de 10 bits, lo que permite una detección precisa de movimientos bruscos en el vehículo. Su rango configurable de medición ( $\pm 2\text{ g}/\pm 4\text{ g}/\pm 8\text{ g}/\pm 16\text{ g}$ ) permite adaptarse a las diferentes condiciones de operación. Adicionalmente cuenta con comunicación I<sup>2</sup>C lo que ofrece una interfaz sencilla y eficaz con el microcontrolador.



**Fig. 3 - Módulo acelerómetro ADXL345.**

Batería de litio y módulo de carga HT4928S: Se optó por utilizar este conjunto de batería con módulo de carga para proporcionar una fuente de alimentación interna del dispositivo ante una eventual pérdida de energía externa. Este módulo fue elegido debido a que permite gestionar la carga de la batería de litio de 3.7V, además de contar con un convertor STEP-UP integrado. Tanto la tensión de entrada como la tensión de salida del módulo es de 5V y utiliza una única conexión de entrada y salida, optimizando el espacio y asegurando una operación continua ante interrupciones de energía.

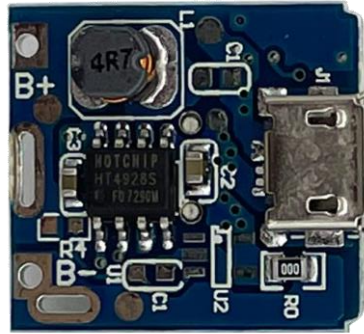


Fig. 4 - Módulo HT4928S.



Fig. 5 - Batería de litio recargable.

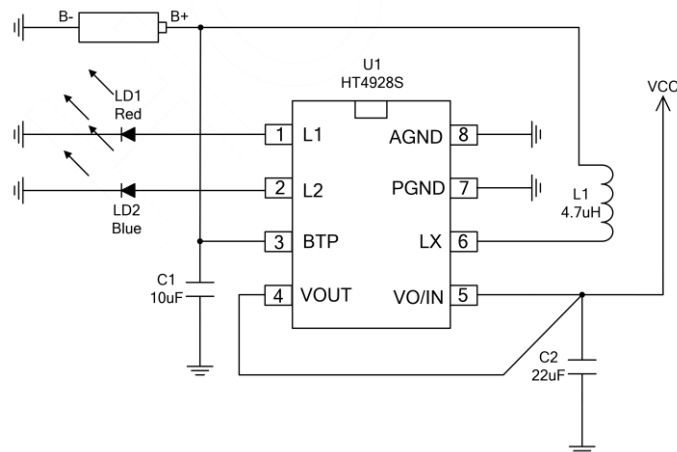


Fig. 6 - Circuito de carga HT4928S.

### Dispositivo de visualización

Raspberry PI + pantalla LCD: La elección de utilizar un microordenador Raspberry PI en combinación con una pantalla LCD, responde a la facilidad de crear y visualizar una interfaz gráfica, que será utilizada como dispositivo de visualización. Cuenta con puertos de comunicación UART y SPI que son de utilidad para la conexión e interacción con otros componentes del sistema.



Fig. 7 - Raspberry PI.



Fig. 8 - Pantalla LCD táctil 7".

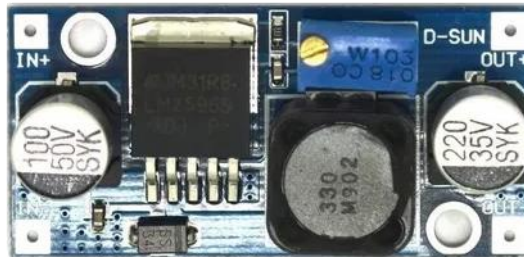
Módulo MCP2515 CAN BUS: Debido a que la Raspberry PI no dispone de una interfaz de comunicación CAN propia, se seleccionó este módulo para poder incorporar esta funcionalidad. Este módulo requiere únicamente de la comunicación SPI disponible en la Raspberry PI, con lo que ahora permite al microordenador enviar y recibir mensajes por protocolo CAN.



Fig. 9 - Módulo CAN BUS MCP2515.



Módulo STEP-DOWN LM2596: Para poder proveer la tensión necesaria para la Raspberry PI y para el módulo CAN BUS, se escogió el módulo de conversor STEP-DOWN LM2596 debido a su bajo coste, disponibilidad del mercado. Entre sus características principales se destaca que su tensión de salida es regulable, posee una baja disipación de energía comparado a un conversor lineal, además puede entregar hasta 3A de corriente de salida, con una tensión máxima de entrada de 40V, lo que cumple con los parámetros de tensión vehicular previamente establecidos.



**Fig. 10** - Conversor STEP-DOWN LM2596.



## Diagrama en bloques del dispositivo

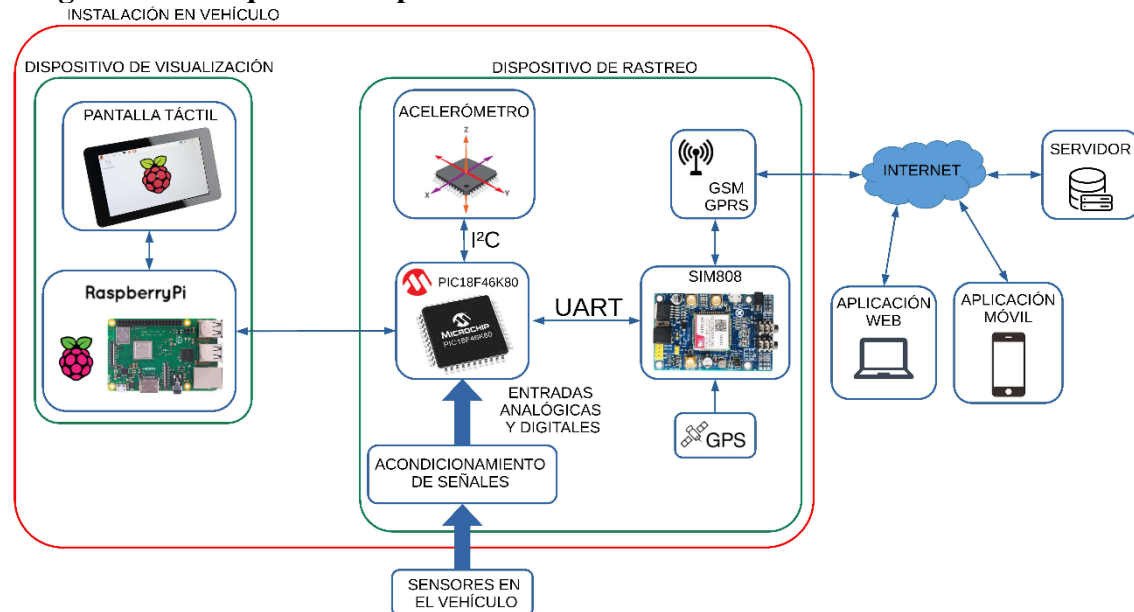


Fig. 11 - Diagrama en bloques.

Como se observa en el diagrama en bloques de la **Fig. 11**, el proyecto consta de una instalación en vehículo, un servidor web y aplicaciones tanto web como Android. La instalación en el vehículo se subdivide en dos dispositivos principales: el dispositivo de rastreo, que posee conexiones GPS y GPRS, y el dispositivo de visualización que permite monitorear datos a través de una pantalla.

Dentro del dispositivo de rastreo, se encuentra un microcontrolador PIC18F46K80 como componente central, el cual cuenta con puertos de comunicaciones UART, CAN, I²C que se usan para comunicarse con el resto de los componentes. Además, dispone de entradas analógicas para poder medir los diferentes sensores propios de un vehículo, como ser temperatura del motor o presión de aceite.

Mediante una comunicación I²C, el microcontrolador recibe información de un sensor acelerómetro ADXL345 de 3 ejes, es decir, que puede medir aceleración en 3 direcciones diferentes. Esto se aprovecha para detectar cambios bruscos de aceleración, indicativo de una conducción agresiva o eventos inusuales.

El módulo SIM808 se encuentra conectado al microcontrolador a través de una comunicación UART. Mediante antenas externas, este dispositivo puede recibir coordenadas GPS de su ubicación actual, además de contar con comunicación GPRS donde haya cobertura. Las coordenadas GPS, la información de los sensores leídos y los eventos detectados por el sensor acelerómetro son enviados mediante GPRS a una base de datos alojada en un servidor web.

Para poder monitorear desde el vehículo la información de los sensores, se utiliza un microordenador Raspberry PI con una pantalla táctil de 7 pulgadas, donde se visualiza en tiempo real las variables obtenidas de los sensores, como así también información obtenida por GPS incluyendo la hora, velocidad, tiempo de viaje y otros datos estadísticos. La comunicación desde el dispositivo de rastreo y el dispositivo de visualización es mediante CAN BUS, utilizando una tasa de baudios de 250kbps.

Con respecto a la visualización de datos en dispositivos externos, se utiliza tanto la aplicación



web como la aplicación Android, las cuales se conectan a la base de datos alojada en el servidor web para obtener la ubicación del vehículo e información de los sensores, y ser observada en tiempo real en un mapa. Además, en caso de ser detectado algún incidente en el vehículo, se visualizará una alerta.

### Diagramas de flujo

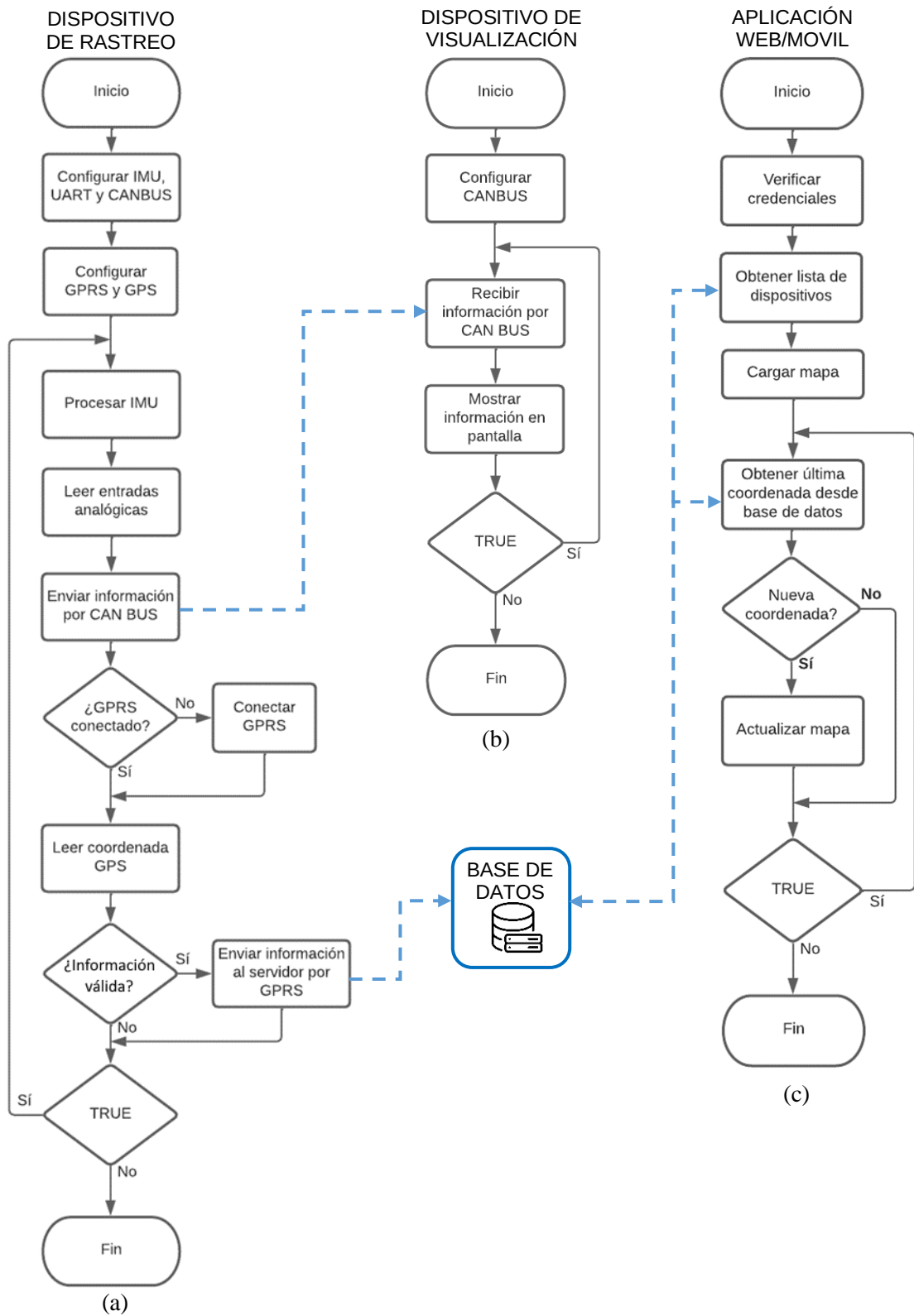


Fig. 12 - Diagramas de flujo.





En los diagramas de flujo de la **Fig. 12** se observan los algoritmos utilizados en dispositivo de rastreo (**a**), dispositivo de visualización (**b**) y, aplicación web y móvil (**c**).

#### Dispositivo de rastreo:

Dentro del dispositivo de rastreo, al iniciar, se deben configurar el puerto UART con una tasa de transmisión de 115200 bps para comunicarse con el módulo SIM808, la comunicación CAN BUS con el estándar CAN 2.0A a una tasa de 250 kbps que será utilizada para comunicarse con el microordenador Raspberry PI, y la comunicación I<sup>2</sup>C a una velocidad de 400 kbps para comunicarse con el sensor acelerómetro.

Luego, se configura la comunicación GPS y GPRS del módulo SIM808, mediante comandos AT a través de la comunicación UART, definiendo una tasa de actualización de las coordenadas GPS cada un segundo y proveedor para utilizar los servicios GPRS.

Dentro del bucle principal se encuentra el algoritmo que se repite periódicamente, donde primero se obtiene información de los 3 ejes del sensor acelerómetro conectado por I<sup>2</sup>C, y se procesa para detectar si existe algún evento inusual como una frenada brusca o una posición incoherente del vehículo, lo que puede ser indicador de algún vuelco o accidente. Luego, mediante las entradas analógicas del microcontrolador se realiza la medición de los sensores del vehículo. Esta información es enviada mediante CAN BUS para ser visualizada en tiempo real en la pantalla del microordenador a la vista del conductor.

Para el envío de información al servidor, primero se debe verificar mediante comandos AT que la comunicación GPRS está establecida, en caso contrario, se debe establecer conexión GPRS. Luego, cuando se tiene conexión, se leen las coordenadas GPS provistas por el módulo SIM808, y si la información es válida, se envía al servidor web mediante una solicitud POST, que a su vez almacena los datos en la base de datos MySQL alojada en el servidor.

#### Dispositivo de visualización:

Se trata del microordenador Raspberry PI, donde se visualizan los parámetros obtenidos de los sensores, e información obtenida mediante GPS.

Para esto, al iniciar el dispositivo, se deben configurar la comunicación CAN BUS para poder comunicarse con el dispositivo de rastreo, usando el estándar CAN 2.0A a una tasa de 250 kbps.

Una vez configurado, se entra en el bucle principal, el cual recibe información mediante CAN BUS, y muestra dicha información en pantalla.

#### Aplicación web y móvil:

Para las aplicaciones web y móvil, primero se deben verificar las credenciales del usuario antes de poder acceder al mapa. Para esto, se muestra una pantalla de inicio de sesión en aplicación web (**Fig. 13**) y en aplicación móvil (**Fig. 14**), donde se verifican las credenciales del usuario en el servidor web. Una vez autorizado el acceso, se obtiene la lista de dispositivos desde la base de datos, y se carga el mapa mostrando la ubicación del dispositivo, como se observa en los diseños de **Fig. 15** y **Fig. 16**. La aplicación tiene un bucle principal donde constantemente se envía una solicitud POST al servidor web para consultar la última coordenada del dispositivo en la base de datos MySQL. Si el dispositivo cambió de ubicación, se actualiza su posición en el mapa, y dependiendo del estado del dispositivo (Disponibilidad GPS, sin conexión GPS, alimentación desconectada, entre otros) se actualiza el ícono del marcador en el mapa para indicar la condición actual, tal como se describe en la **Tabla 1**.

---

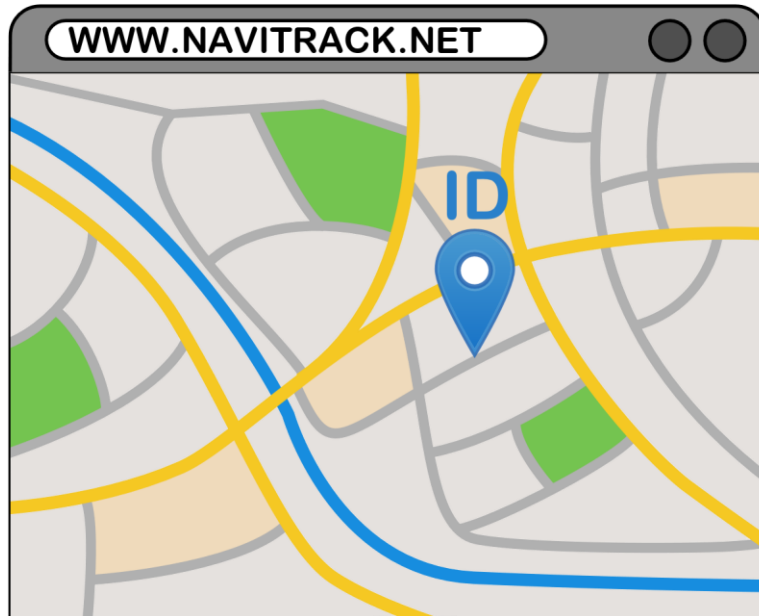


The image shows a web browser window with the address bar containing "WWW.NAVITRACK.NET". The main content area features the "NaviTrack" logo in a large, bold, black font. Below the logo, there are two input fields: the first is labeled "Correo electrónico" and the second is labeled "Contraseña". At the bottom of the form is a blue button with the text "Iniciar sesión".

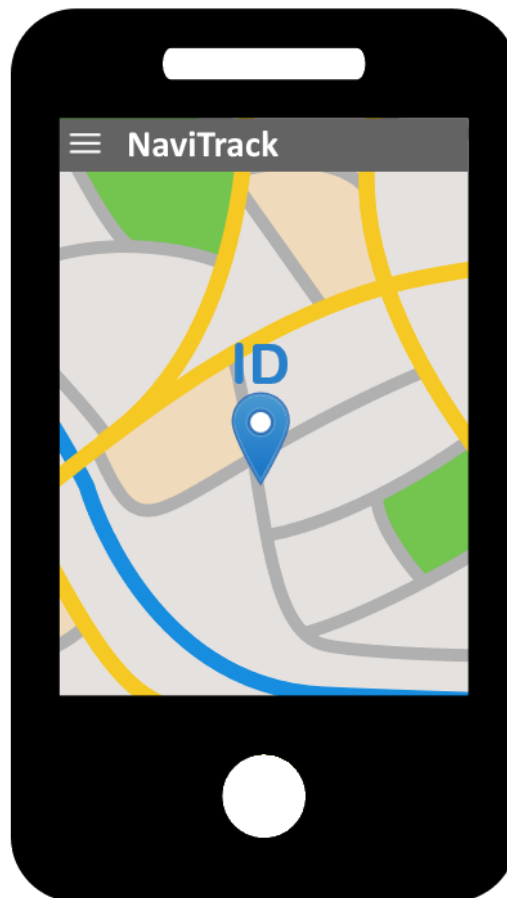
**Fig. 13** - Diseño inicio de sesión web.

The image shows a black smartphone with a white screen. The screen displays the "NaviTrack" login form, which is a scaled-down version of the web form shown in Figure 13. It includes the "NaviTrack" logo, the "Correo electrónico" and "Contraseña" input fields, and the "Iniciar sesión" button.

**Fig. 14** - Diseño inicio de sesión Android.







**Fig. 15** - Diseño aplicación web.




**Fig. 16** - Diseño aplicación Android.



Icono	Descripción
 Icono normal	Indicativo de un funcionamiento normal. El dispositivo posee señal de GPS disponible y comunicación GPRS activa.
 Icono sin señal GPS	El dispositivo posee comunicación GPRS activa, pero sin señal GPS disponible. Esto puede deberse a encontrarse en una zona urbana con una gran cantidad de edificios, en zona de arboleda o bajo techo. En el mapa se muestra la última ubicación conocida.
 Icono sin comunicación GPRS.	El dispositivo no posee comunicación GPRS activa debido a que se encuentra fuera del área de cobertura, o a falta de energía. En el mapa se muestra la última ubicación conocida.
 Icono de advertencia	Representa una advertencia debido a un manejo brusco, o algún parámetro fuera de su rango normal de funcionamiento, como ser una alta temperatura del agua del motor o una velocidad excesiva.
 Icono de alerta grave	Indica una alerta grave, indicando un accidente o un vuelco del vehículo detectado por los acelerómetros.



---

 <p>Icono de falta de energía</p>	<p>Indica que el dispositivo no está recibiendo energía del vehículo debido a una desconexión de la batería o extracción del mismo. En este caso el dispositivo se encuentra alimentado por su batería interna.</p>
--	---

**Tabla 1** – Iconos.

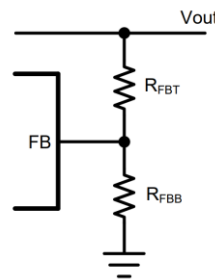
## Evaluación Final del Sistema

### Preparación del prototipo con materiales finales

#### Dispositivo de rastreo

Para este dispositivo se requiere una tensión de entrada de 5V para alimentar el microcontrolador, el sensor acelerómetro y el módulo de comunicación SIM808. Con el objetivo de asegurar compatibilidad de vehículos que operan en un rango de 12V a 24V, se empleó un convertor step-down o Buck en el circuito de alimentación. Este convertor, mediante conmutación permite una alta eficiencia en la reducción de la tensión de entrada al nivel necesario, minimizando así la disipación térmica asociada al uso de un regulador lineal.

El regulador LM22673MRE-ADJ empleado en este diseño permite ajustar la tensión de salida mediante un divisor resistivo como se observa en la **Fig. 17**. La relación entre las resistencias del divisor y la tensión de salida se describe en la *Ecu. 1*, donde  $V_{FB}$  es la tensión de referencia y tiene un valor de 1.285V. A partir de la *Ecu. 1*, se deriva la *Ecu. 2* que nos permite calcular el valor de  $R_{FBT}$  definiendo un valor de  $R_{FBB}$ . Definiendo un valor de  $R_{FBB} = 1k\Omega$  se obtiene que  $R_{FBT} = 2.87k\Omega$  (*Ecu. 3*).



**Fig. 17** - Divisor resistivo de realimentación.

$$V_{out} = V_{FB} \left( 1 + \frac{R_{FBT}}{R_{FBB}} \right) \quad (Ecu. 1)$$

$$R_{FBT} = \left[ \frac{V_{OUT}}{V_{FB}} - 1 \right] R_{FBB} \quad (Ecu. 2)$$

$$R_{FBT} = \left[ \frac{5V}{1.285V} - 1 \right] 1k\Omega = 2.87k\Omega \quad (Ecu. 3)$$

Con referencia al diseño del circuito de alimentación para este dispositivo, ilustrado en la **Fig. 18**, se determina que las resistencias del divisor resistivo del regulador deben ser  $R_{10} = 2.87k\Omega$  y  $R_{11} = 1k\Omega$ , conforme a los cálculos previos.

Para mitigar la interferencia de ruido eléctrico generados en el entorno del vehículo, se incorpora un diodo TVS (supresor de voltaje transitorio) modelo SMJ30CA en la entrada de alimentación, permitiendo así una tensión de entrada más limpia y estable.

Adicionalmente, en el circuito de alimentación se integra el módulo de carga de batería

HT4982S junto con una batería de litio de 3.7V. Este módulo, al recibir una tensión de alimentación de 5V, carga la batería, y en ausencia de alimentación externa, provee una salida de 5V a partir de la batería con un convertor step-up o boost integrado elevando la tensión de la batería de 3.7V a 5V, todo a través del mismo pin. Esta configuración asegura una fuente de energía continua ante una eventual interrupción de la alimentación externa.

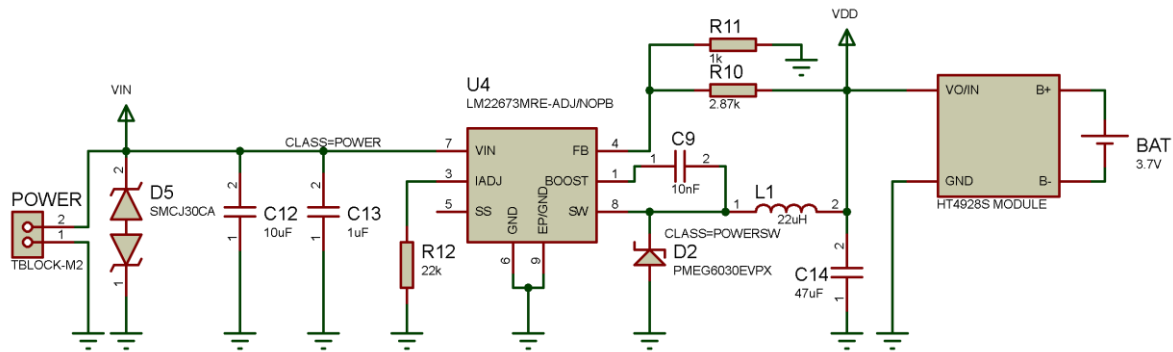


Fig. 18 - Circuito de alimentación.

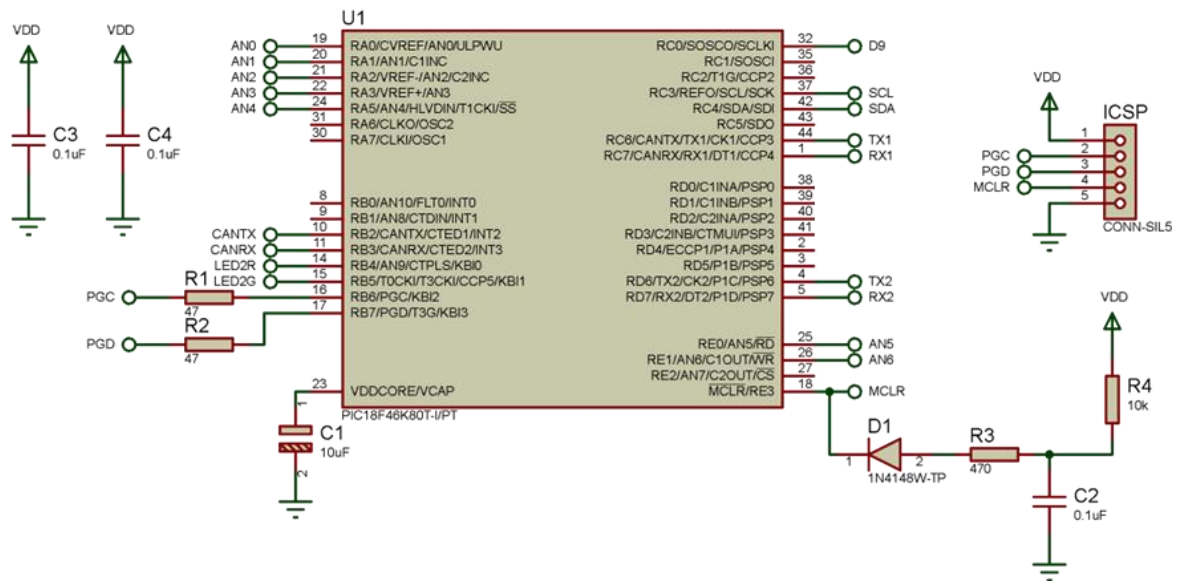


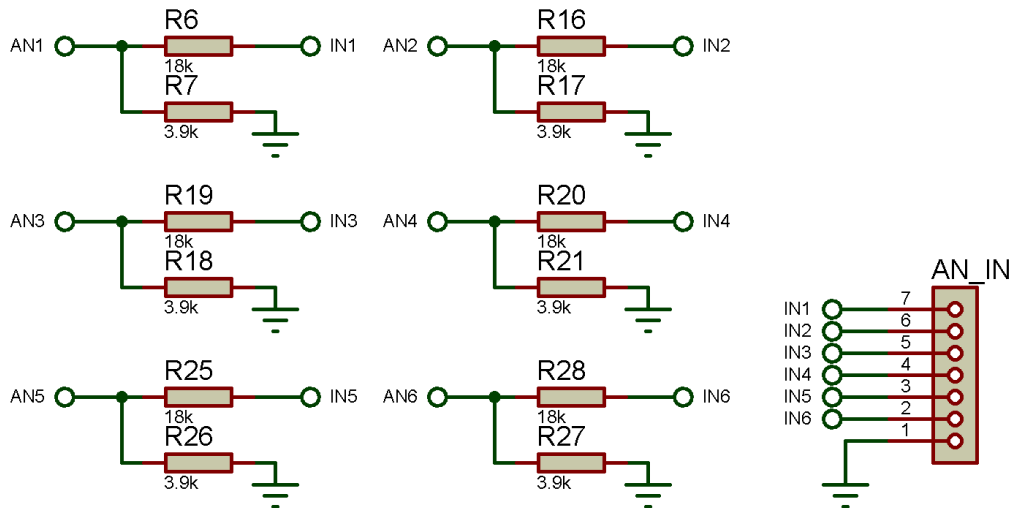
Fig. 19 – Circuito microcontrolador.

En la Fig. 19 se muestra el microcontrolador PIC18F46K80, el cual es alimentado por los 5V suministrados por el circuito de alimentación previamente descrito. Para mejorar la estabilidad y minimizar el ruido en la alimentación del microcontrolador, se han colocado capacitores de desacoplamiento (C3 y C4) próximos a sus pines de alimentación.

Dado que se necesita adaptar las señales de entrada provenientes de los sensores hacia un nivel seguro del microcontrolador, se diseñó un divisor resistivo, para el cual fue tomada en cuenta la tensión de alimentación de los vehículos de 24V, añadiéndose un margen adicional de tolerancia del 15%, definiendo así una tensión máxima de entrada de 27.6V. Este divisor asegura que, para este nivel de tensión máxima de entrada, no se supere la tensión máxima soportada por las

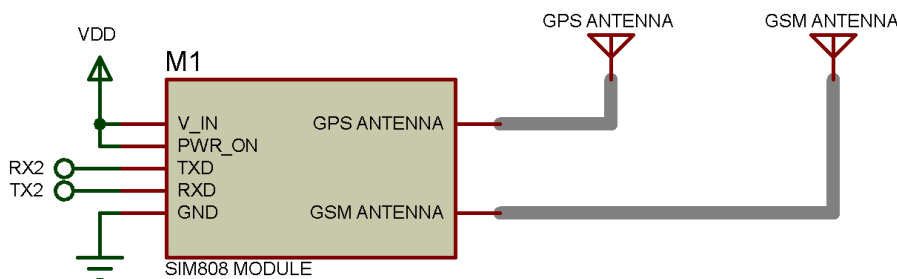
entradas analógicas del microcontrolador, etiquetadas como AN1 hasta AN6. La relación entre la tensión máxima de entrada del divisor y la tensión máxima en la entrada analógica del microcontrolador se describe mediante la Ecu. 4. En ella se observa que para la tensión máxima de 27.6V, el microcontrolador recibirá un máximo de 4.92V, lo cual es un nivel seguro.

$$V_{AN1} = V_{IN1} \left( \frac{R7}{R6 + R7} \right) = 27.6V \left( \frac{3.9k\Omega}{18k\Omega + 3.9k\Omega} \right) = 4.92V \quad (\text{Ecu. 4})$$



**Fig. 20 - Entradas analógicas.**

Para facilitar la depuración y desarrollo, se han designado los pines de comunicación UART TX1 y RX1 del microcontrolador, mientras que para la comunicación UART destinada al módulo SIM808 se han utilizado los pines TX2 y RX2. Como se observa en la Fig. 21, este módulo se alimenta con VDD 5V, posee conexión hacia una antena GSM, y a una antena GPS externa conectada mediante cable coaxial.



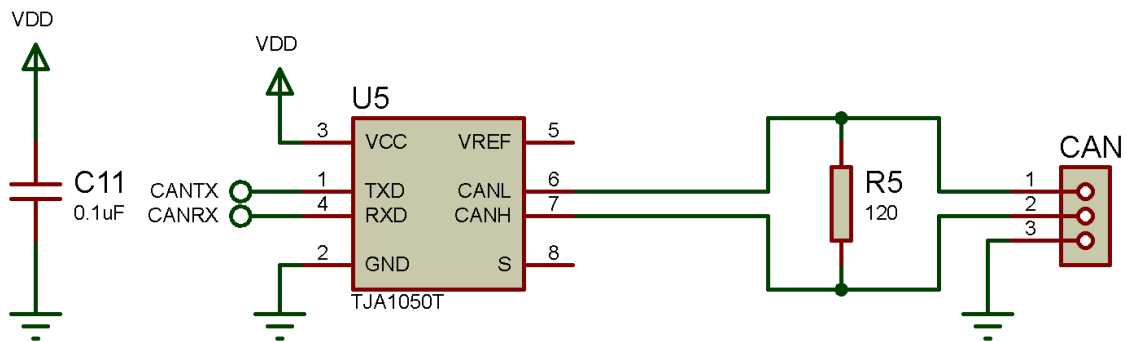
**Fig. 21 - Módulo SIM808.**

Para adicionar una conectividad con el dispositivo de visualización, se agregó una comunicación CAN BUS, que se caracteriza por su robustez ante interferencias, las cuales son comunes en el ámbito vehicular. Para esta comunicación se utilizan los pines CANTX y CANRX del microcontrolador, los cuales trabajan a niveles TTL. Debido a esto, no puede conectarse directamente al bus, por lo que se necesita de un transceptor, que es el encargado de convertir de niveles TTL que maneja el microcontrolador, a la comunicación en formato diferencial que utiliza



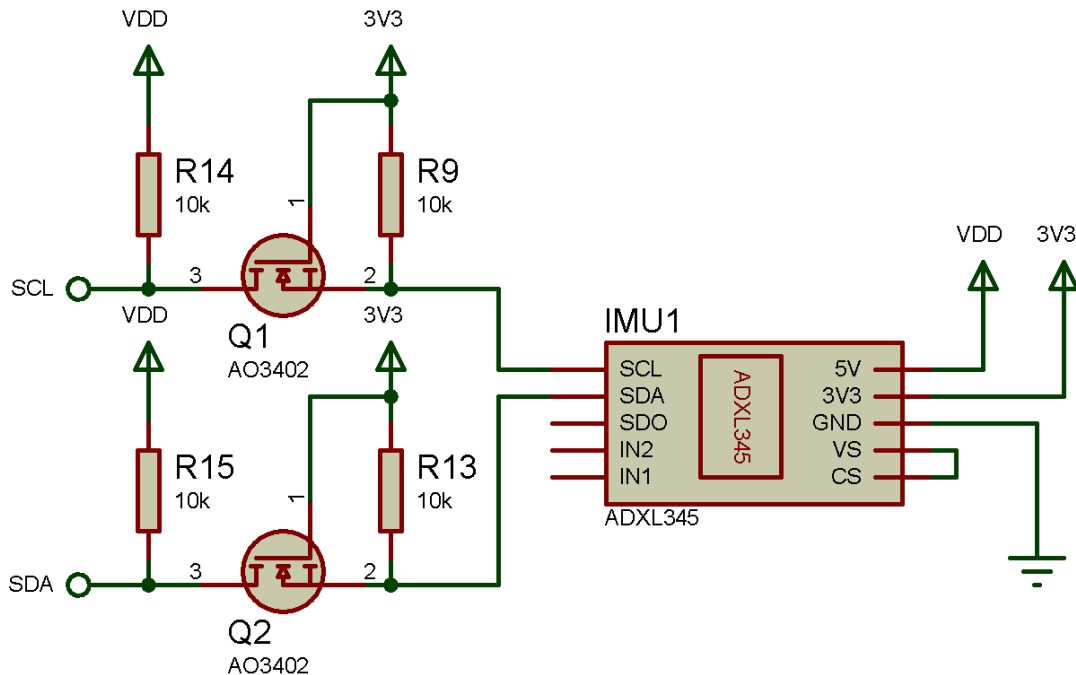
en la comunicación CAN y viceversa, permitiendo así el envío y recepción de mensajes o tramas CAN.

Se utiliza un transceptor CAN de alta velocidad modelo TJA1050T mostrado en la **Fig. 22**, el cual se alimenta con 5V y es compatible con el estándar ISO 11898, que admite una tasa de transmisión de hasta 1Mbps. Este protocolo requiere la colocación de una resistencia de terminación de 120ohm en cada extremo del bus para asegurar la integridad de la señal y la eliminación de reflexiones. En este diseño particular, dado que se trata de una red compuesta únicamente por dos dispositivos, se ha provisto una resistencia de terminación próxima a cada transceptor, convirtiéndolos así en los extremos del bus.



**Fig. 22** - Transceptor CAN BUS.

Para poder medir aceleraciones, se utiliza el módulo acelerómetro de 3 ejes ADXL345, que utiliza la comunicación I<sup>2</sup>C configurada en modo FAST, es decir, a una velocidad de 400kbps. Este sensor se alimenta con VDD 5V para su funcionamiento, mientras que genera internamente 3.3V para usar como nivel lógico en su bus de comunicación. La comunicación I<sup>2</sup>C se realiza a través de los pines SDA (data) y SCL (clock) del microcontrolador, los cuales operan a niveles lógicos de 5V. Dada la diferencia de niveles lógicos del bus entre el sensor y el microcontrolador, se diseñó un convertor de nivel bidireccional que permite que la señal de 3.3V del sensor se convierta en 5V hacia el microcontrolador, y viceversa. Para este convertor se utilizó un transistor MOSFET canal-N modelo AO3402 para cada una de las líneas de este bus de comunicación.



**Fig. 23** - Módulo acelerómetro ADXL345.

La explicación previa ha abordado el diseño del esquemático del dispositivo, detallando cada componente y su función en el sistema. Este análisis es una descomposición del esquemático completo, que se presenta en el **Anexo I – esquemático dispositivo de rastreo**.

En base a este esquemático, se procedió a diseñar la placa de circuito impreso (PCB), donde se optó por emplear componentes de montaje superficial (SMD) cuando fuese posible, con el fin de optimizar el tamaño y mejorar la eficiencia en el montaje. El diseño resultante es una placa de doble cara o doble faz, cuyas vistas se observan en la **Fig. 24** y **Fig. 25**.

Posteriormente, conforme a lo diseñado, se encomendó la fabricación de la placa a un fabricante externo, cuyo resultado se puede apreciar en la **Fig. 26** y **Fig. 27**. En esta placa se soldaron todos los componentes presentes en el esquemático, a excepción del módulo SIM808. Este resultado de la soldadura de los componentes se aprecia en la **Fig. 28**.

En cuanto al montaje del módulo SIM808, este se colocó encima de la placa, asegurándose mediante tornillos a la misma. Para la conexión eléctrica se emplearon cables que permiten la alimentación del módulo y la transmisión de datos. Este montaje final de los componentes, con el módulo de comunicación incluido, se observa en la **Fig. 29**.

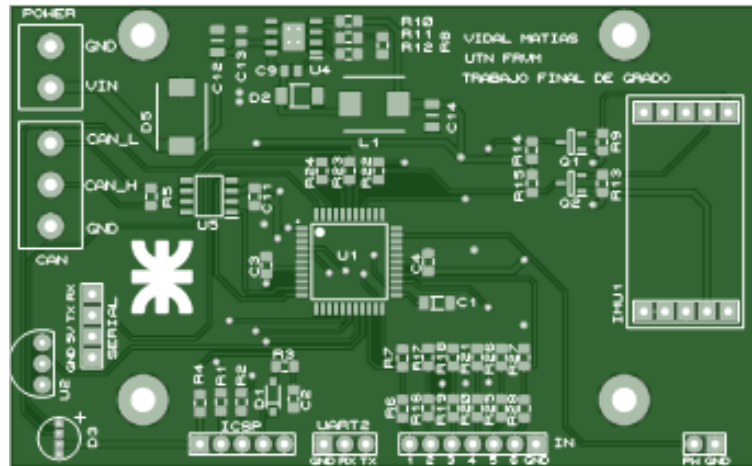


Fig. 24 – Diseño de circuito impreso cara superior.

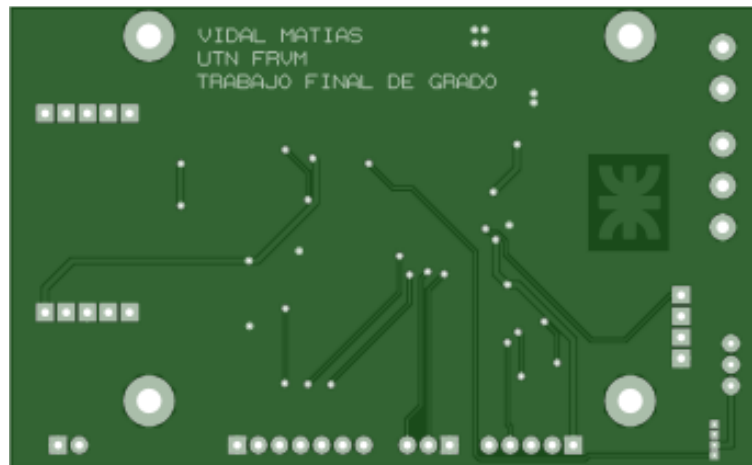


Fig. 25 - Diseño de circuito impreso cara inferior.

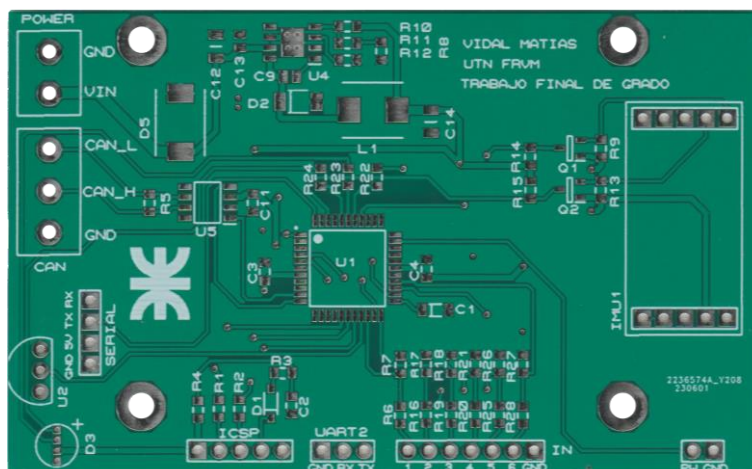


Fig. 26 - Circuito impreso cara superior.

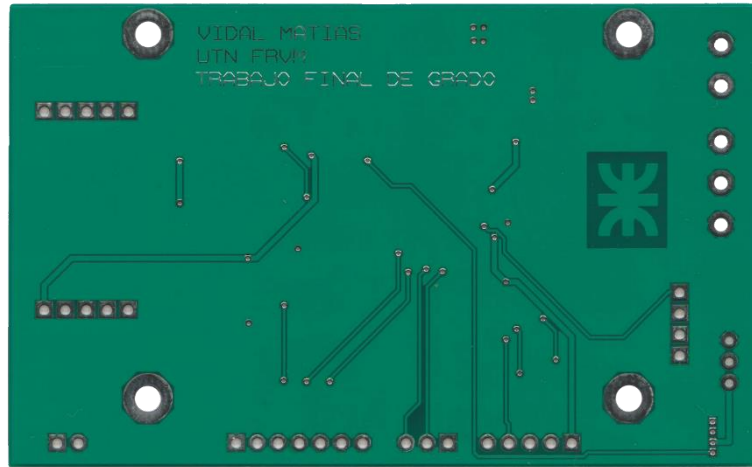


Fig. 27 - Circuito impreso cara inferior.

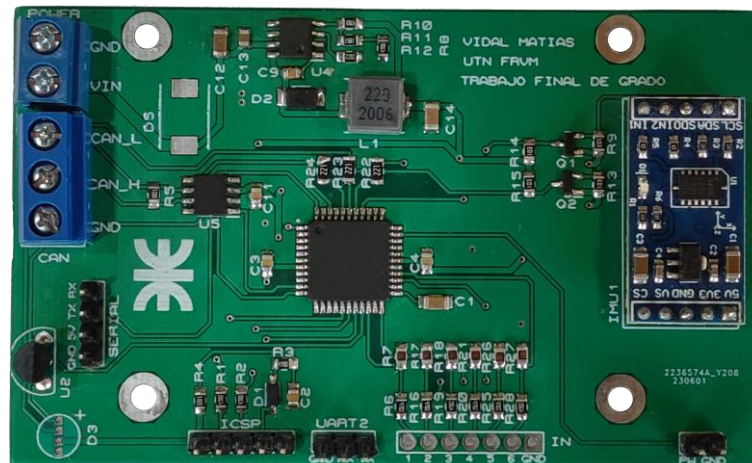


Fig. 28 - Placa con componentes montados.

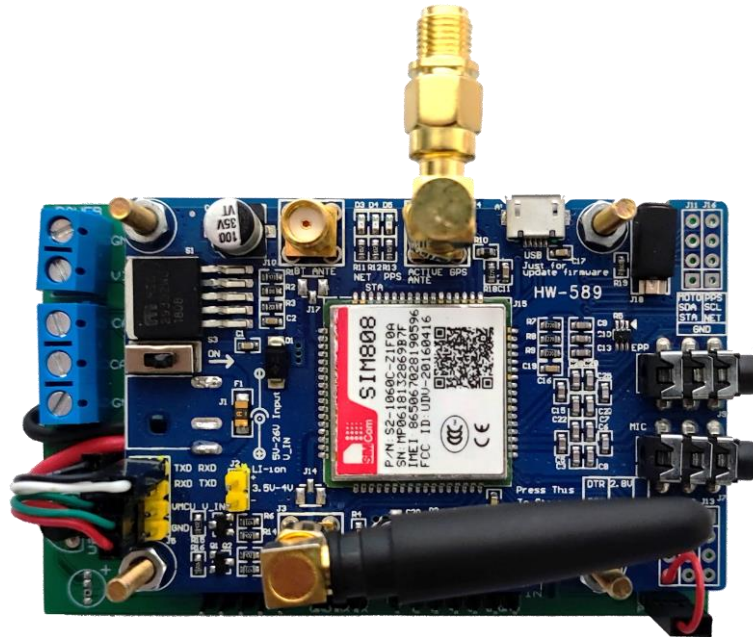


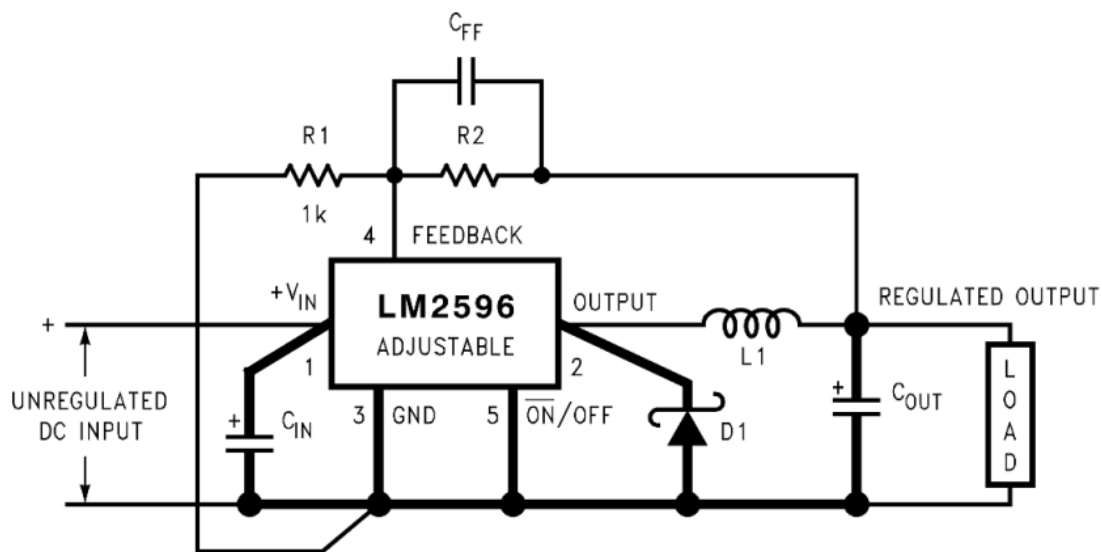
Fig. 29 - Placa completa.

### Dispositivo de visualización

Este dispositivo se trata del microordenador Raspberry PI modelo 3B+ con su pantalla de 7", al cual se le añadió comunicación CAN BUS para comunicarse con el dispositivo de rastreo.

Para la alimentación del dispositivo se utilizó el módulo de regulador de voltaje STEP-DOWN LM2596, que se alimenta con la tensión disponible en el vehículo. Este módulo permite, mediante un preset resistivo, ajustar la tensión de salida a 5V, necesaria para el funcionamiento de los componentes. El circuito interno del módulo se muestra en la **Fig. 30**, donde se puede apreciar que se trata de un circuito de un convertor Buck.

De acuerdo a la hoja de datos del LM2596, la tensión de salida se determina por la relación de resistencias  $R_1$  y  $R_2$ , con una tensión de referencia ( $V_{REF}$ ) de 1.23V, tal cómo se expresa en la *Ecu. 5*. En este módulo,  $R_1$  y  $R_2$  forman parte del preset.



**Fig. 30** - Diagrama interno módulo LM2596.

$$V_{OUT} = V_{REF} \left( 1 + \frac{R_2}{R_1} \right) \quad (Ecu. 5)$$

Para adicionar comunicación CAN BUS a este dispositivo, se utilizó el módulo MCP2515 operando con una tensión de alimentación de 5V. Este módulo provee de una comunicación CAN BUS a dispositivos que no cuentan con este puerto, mediante el uso de un puerto SPI, comúnmente utilizado para la conexión entre dispositivos. El esquemático de este módulo se observa en la **Fig. 31**, donde se aprecia el componente MCP2515 conectado a los pines SPI, siendo este componente capaz de convertir mensajes SPI a mensajes CAN a nivel TTL (TXCAN y RXCAN) y viceversa. Para poder conectarse al bus se requiere de una comunicación diferencial que este componente no provee, por lo que se utiliza un transceptor TJA1050 para convertir entre niveles TTL y una comunicación diferencial (CANH y CANL). Además, se aprecia la presencia de una resistencia de 120Ω entre los pines CANH y CANL, sirviendo como resistencia de terminación del bus, lo que es necesario para evitar reflexiones de señal, debido a que este dispositivo constituye un extremo del bus.



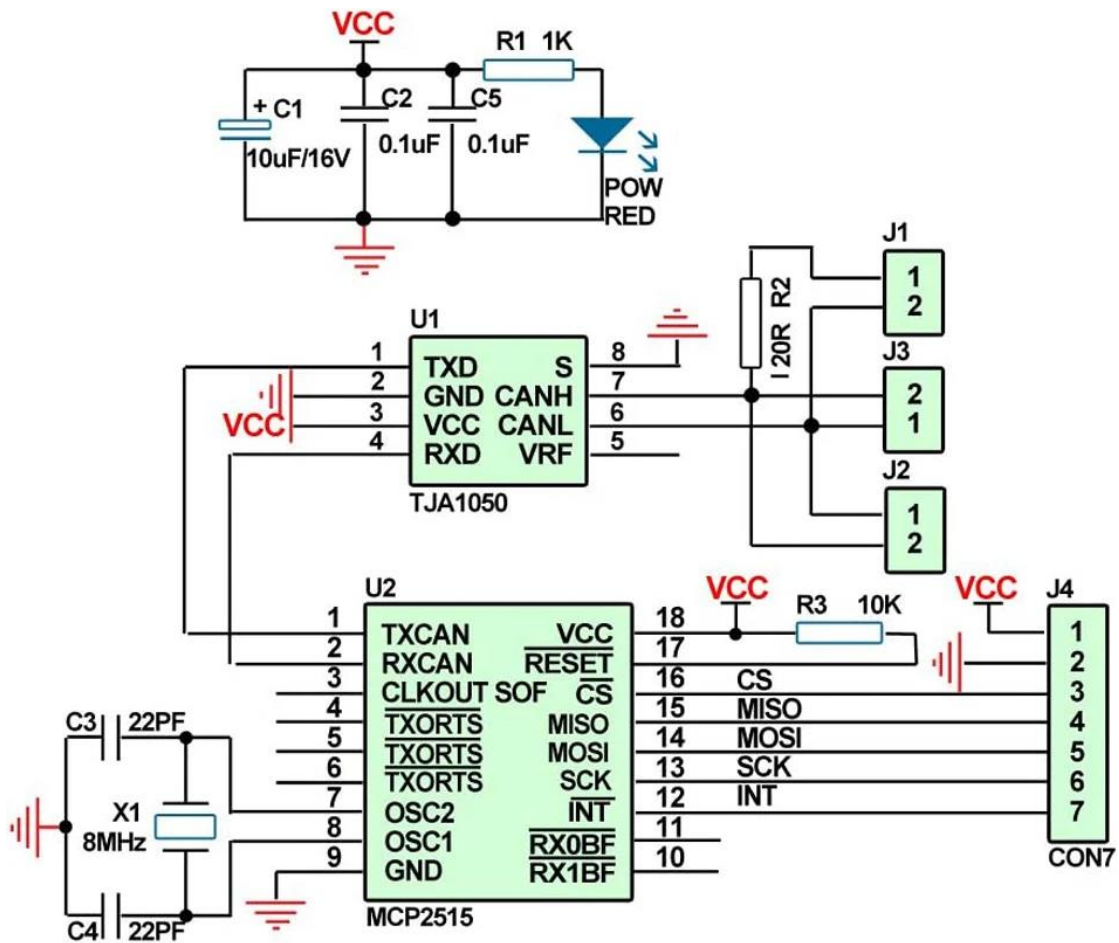


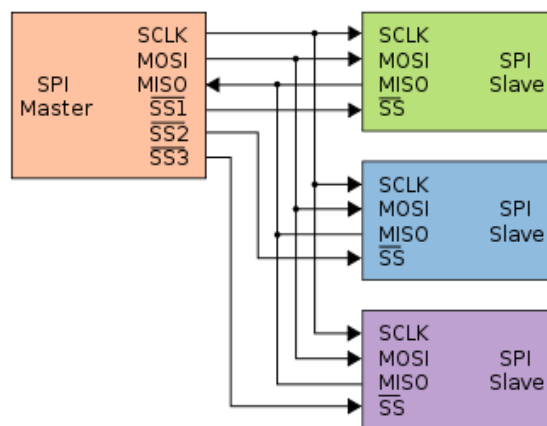
Fig. 31 - Circuito módulo CAN MCP2515.

Para conectar los componentes previamente mencionados a la Raspberry pi, se utiliza su conector GPIO (General Purpose Input/Output), cuya distribución de pines se muestra en la Fig. 32. La tensión de alimentación proporcionada por el módulo LM2596 se conecta a los pines **Ground** y **5V** de la Raspberry Pi. Para la comunicación con el módulo CAN MCP2515, se emplean los pines designados para la comunicación SPI: **SPI\_MOSI**, **SPI\_MISO**, **SPI\_CLK**. La comunicación SPI, como se ilustra en la Fig. 33, opera en un esquema maestro-esclavo, donde varios dispositivos esclavos comparten las líneas de comunicación SPI, pero cada uno necesita de una línea individual SS (slave select) para indicar el dispositivo que hará uso de la comunicación. En este caso, se configuró el pin **GPIO22** como salida, para actuar como el pin SS y así habilitar el módulo CAN MCP2515 a utilizar la comunicación SPI.

En la Fig. 34 se observa el esquema de conexión entre los pines de conexión GPIO de la Raspberry PI, el módulo regulador de tensión LM2596 y el módulo CAN MCP2515. Para las conexiones al exterior, se adiciona un conector de 4 pines, de los cuales 2 son para alimentación (VDD y GND) y 2 para comunicación CAN (CANH y CANL).

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

**Fig. 32 -** Distribución de pines GPIO Raspberry PI 3.



**Fig. 33 -** Esquema comunicación SPI.



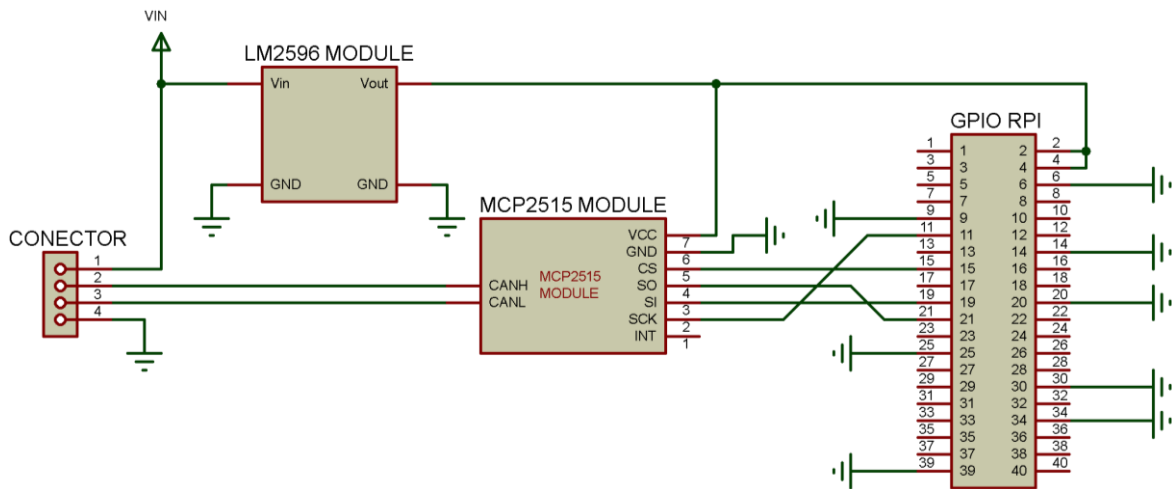


Fig. 34 - Esquema de conexión.

### Aplicativo dispositivo de visualización

Para poder observar los parámetros del vehículo, se diseñó una interfaz gráfica observada en Fig. 35, la cual muestra los parámetros recibidos mediante comunicación CAN BUS desde el dispositivo de rastreo, donde se incluyen las mediciones tomadas de los sensores del vehículo (revoluciones y temperatura del motor, temperatura exterior e interior, presión de aire) y parámetros obtenidos del módulo GPS y otros valores estadísticos (Velocidad actual, tiempo de viaje, velocidad máxima y media, coordenadas actuales).



Fig. 35- Diseño pantalla de monitoreo.

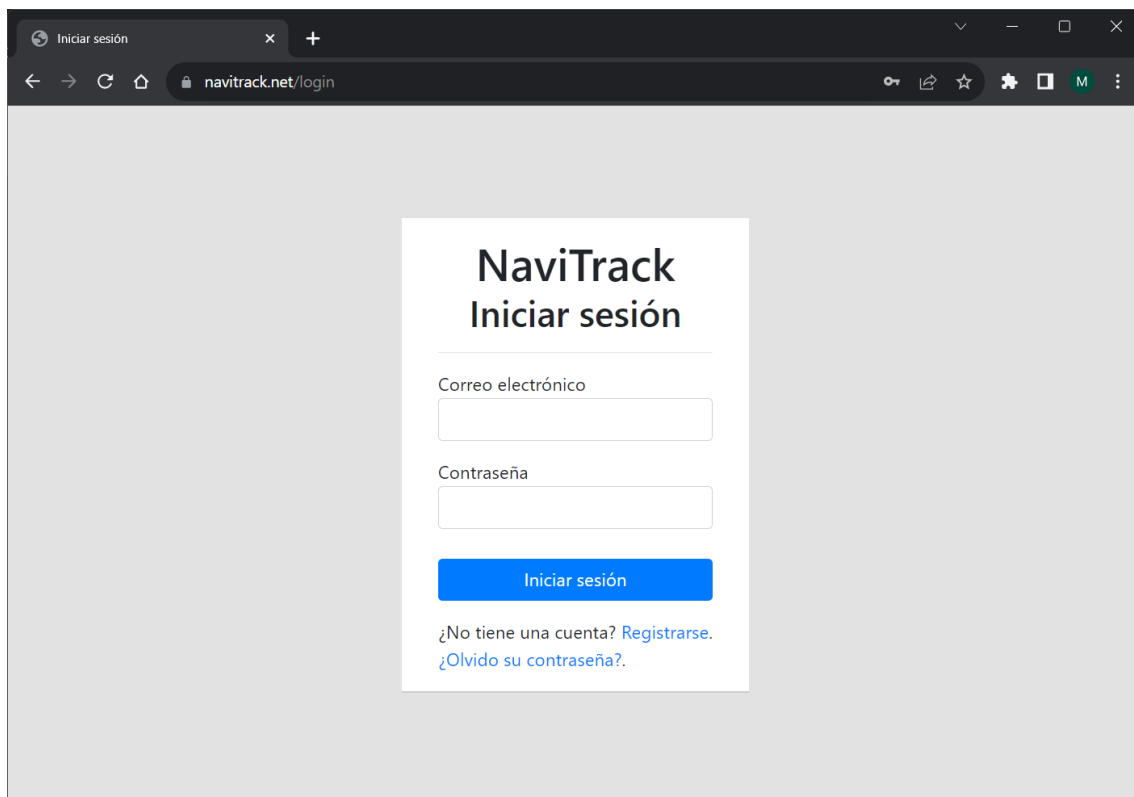
Para desarrollar esta interfaz gráfica, se creó un script Python, usando librería **tkinter** que se encarga de la parte gráfica, y la librería **python-can** que permite recibir tramas CAN BUS. Una vez recibidos los mensajes, se obtienen los valores de cada parámetro y se actualiza la vista en pantalla.

### Aplicación web

Para la aplicación web, se creó un sitio usando PHP para poder leer los elementos de la base de datos, HTML para el diseño del sitio, CSS para los estilos y JavaScript para actualizar en tiempo real la información en pantalla. Para el nombre del aplicativo se utilizó el nombre **NaviTrack**, y se accede mediante la dirección [navitrack.net](http://navitrack.net).

La aplicación web consta de 2 partes: la verificación de credenciales, y la vista de dispositivos. Para verificar las credenciales del usuario, se utiliza la página de inicio de sesión observada en la **Fig. 36**, donde se permite el acceso al mapa de dispositivos utilizando un correo electrónico y una contraseña válida. Una vez autorizado el acceso, se pasa la vista de dispositivos de la **Fig. 37**, donde se muestra un mapa, en el que se encuentran marcadores para cada dispositivo vinculado al usuario. En la barra lateral izquierda se muestra la lista de dispositivos, con un pequeño círculo que indica si el dispositivo está actualmente conectado a la red, colocándose verde si el dispositivo posee comunicación GPRS activa, y gris si no posee comunicación.

De acuerdo a lo propuesto en unidades anteriores, el ícono del dispositivo cambia de acuerdo a su estado actual, pudiendo indicar condiciones como falta de señal GPS, pérdida de energía, entre otros, tal como se propuso en la **Tabla 1**.



**Fig. 36** - Inicio de sesión web.

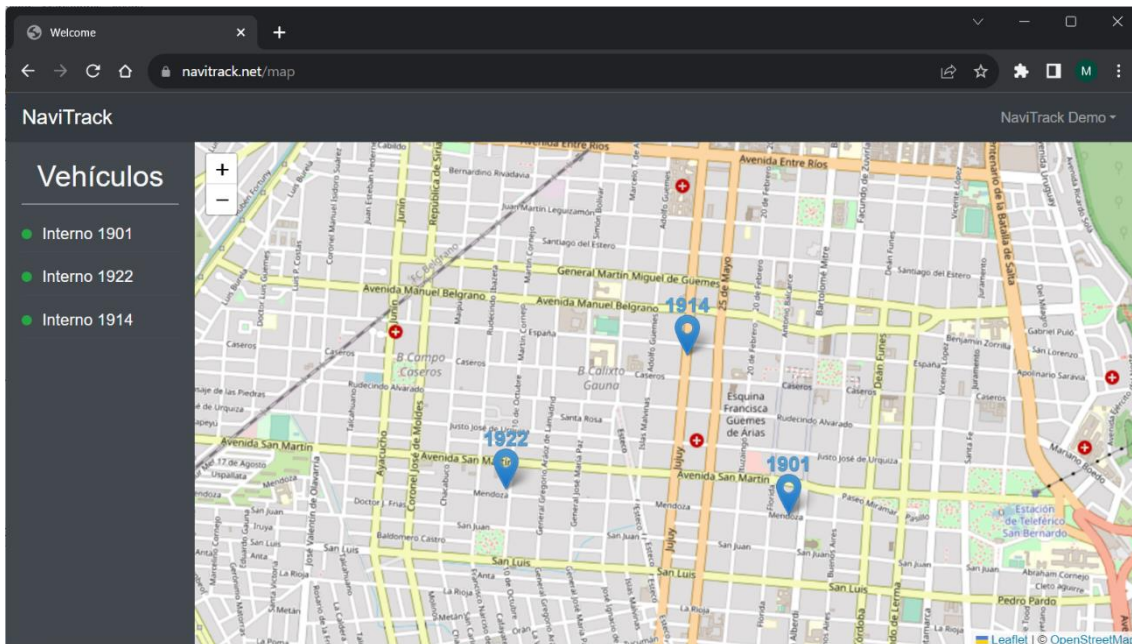


Fig. 37 – Vista aplicación web.

Para poder acceder a más información sobre algún vehículo, debe seleccionarse de la lista, o clicando en el marcador correspondiente. Esto abrirá una barra lateral derecha, como se aprecia en Fig. 38 y Fig. 39. Esta barra muestra la última información disponible sobre el vehículo, indicando hace cuanto fue la última vez que se comunicó con el servidor.

Dentro de los parámetros que se pueden obtener del vehículo, hay información estadística y de GPS, indicando la velocidad actual, la velocidad máxima detectada, velocidad promedio, tiempo de viaje, temperatura del motor y nivel de batería.

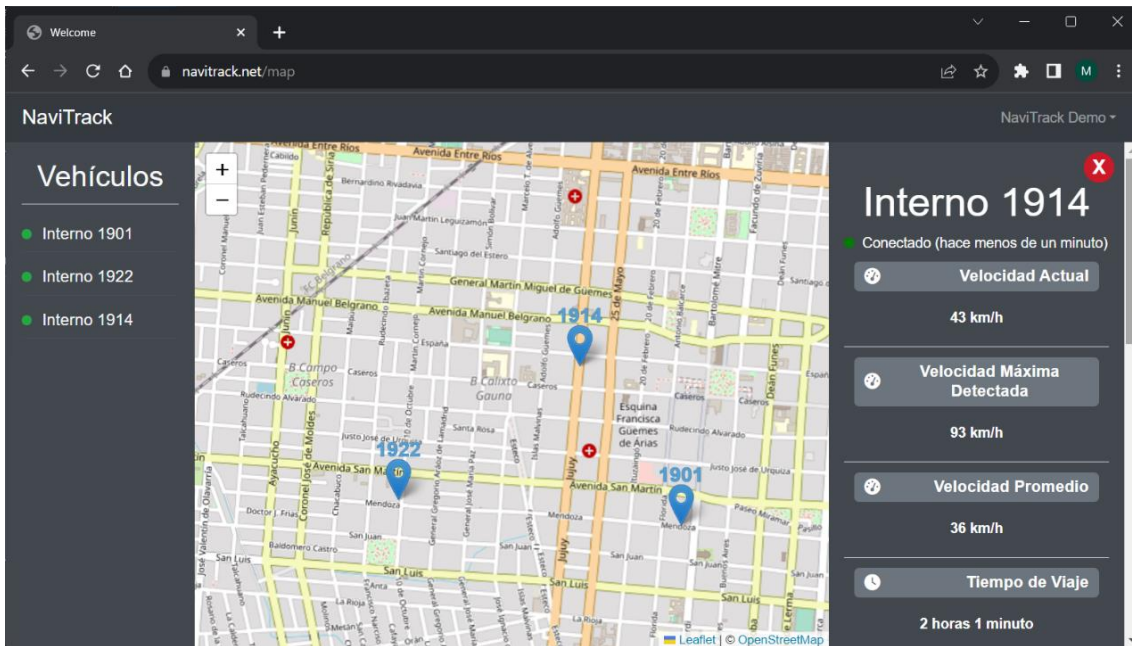
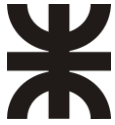


Fig. 38 - Vista aplicación web.

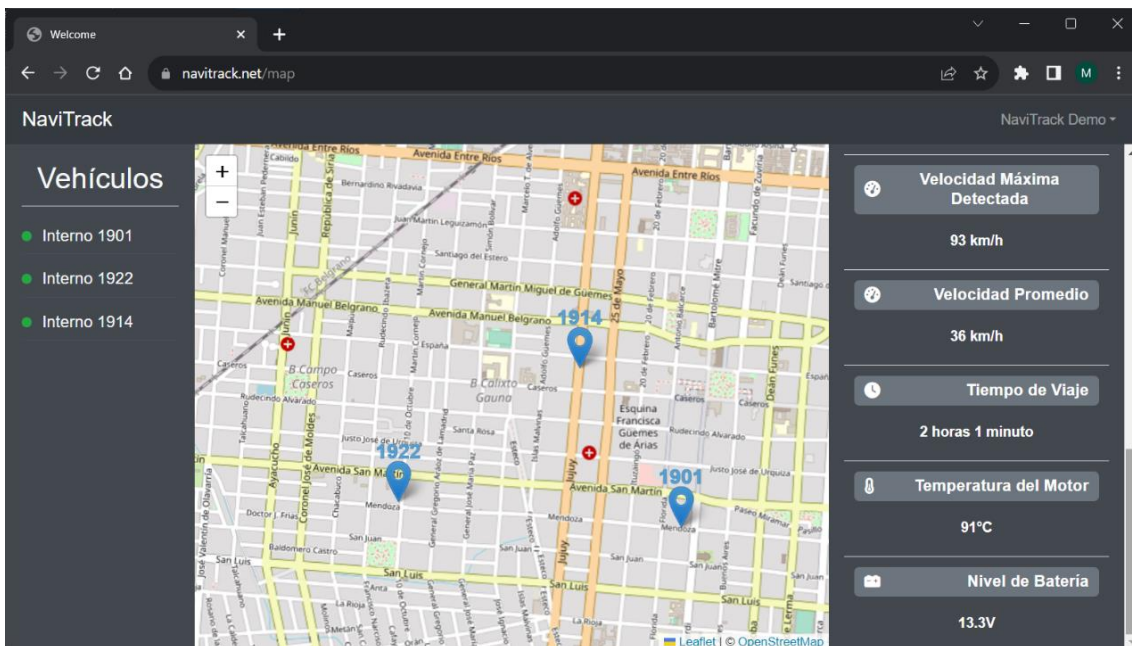


Fig. 39 - Vista aplicación web.



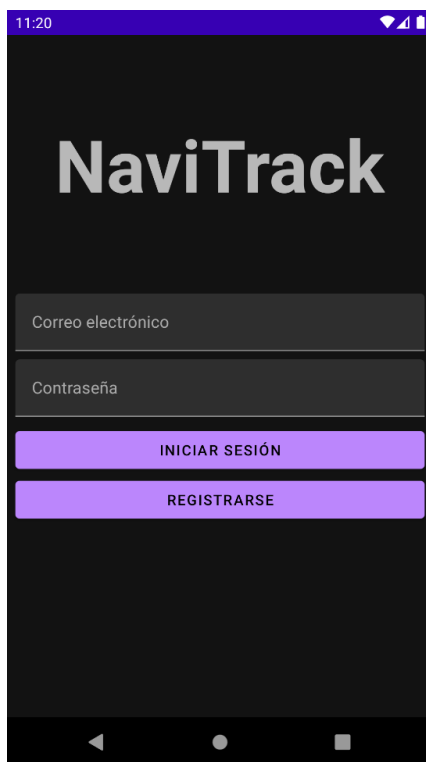
### **Aplicación Android**

De similar funcionamiento a la aplicación web, se desarrolló una aplicación Android, utilizando la interfaz de desarrollo Android Studio. El nombre de esta aplicación mantiene el mismo nombre **NaviTrack**.

Para verificar las credenciales, se diseñó una pantalla de inicio de sesión (**Fig. 40**) donde el usuario ingresa mediante correo electrónico y contraseña. Una vez autorizado el ingreso, se muestra la pantalla principal de la **Fig. 41**, que se trata de una vista de mapa, mostrando los marcadores correspondientes a cada vehículo.

Dentro de la vista principal, se puede acceder al menú lateral, donde se mostrará la lista de vehículos que el usuario tiene registrado, tal como se aprecia en la **Fig. 42**. Al seleccionar un dispositivo de esta lista, se centrará su marcador en el centro de pantalla, mostrando así su última ubicación conocida.

Para acceder a las últimas informaciones disponibles del vehículo, se debe seleccionar el marcador del vehículo deseado, y esto abrirá una barra inferior como se ve en la **Fig. 43**, donde se muestra la misma información que la aplicación web: velocidad actual, velocidad máxima, velocidad promedio, tiempo de viaje, temperatura de motor y nivel de batería.



**Fig. 40** - Inicio de sesión móvil.



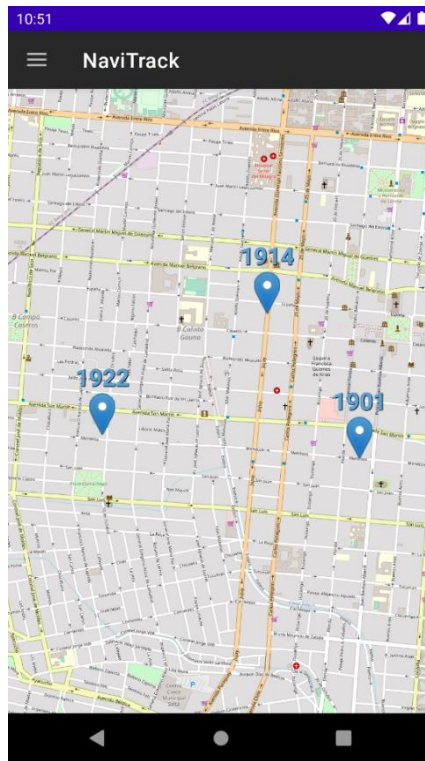


Fig. 41 - Aplicación móvil.

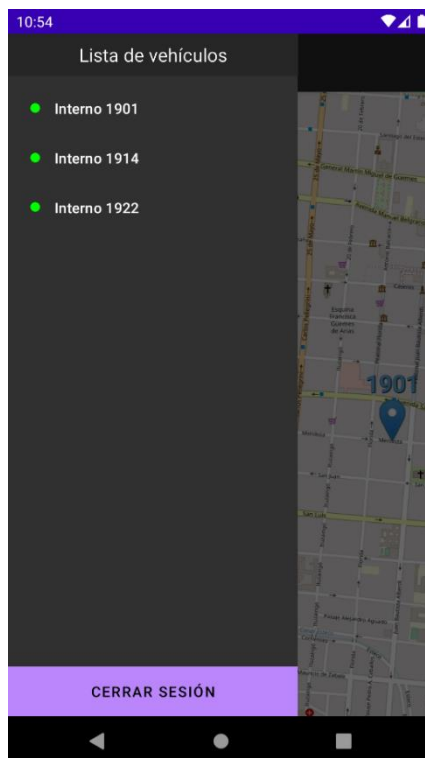


Fig. 42 - Menú lateral aplicación móvil.

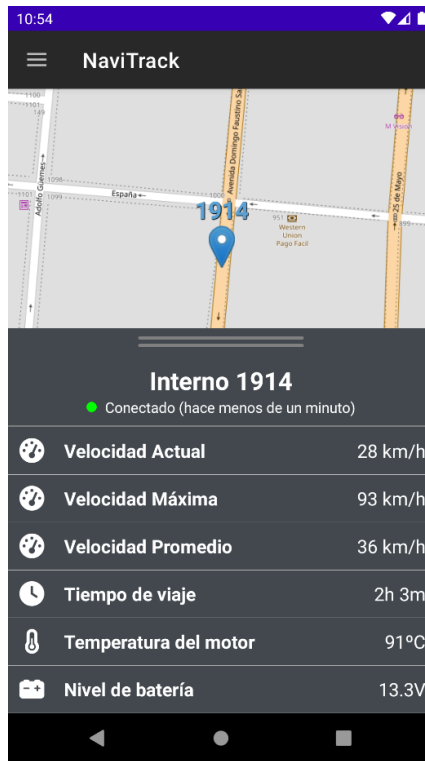


Fig. 43 - Barra de información aplicación móvil.

### Montaje y ensayo real del prototipo

Una vez ensamblada la placa del dispositivo de rastreo, se procedió a montarla en un gabinete plástico, adicionando conectores necesarios para la comunicación con el exterior, los cuales son un conector GX16 de 4 pines (**Fig. 44**) para alimentación y CAN BUS, y un conector GX16 de 7 pines (**Fig. 45**) para las entradas analógicas. Para la comunicación en la red móvil celular se colocó una antena GSM con conector SMA (**Fig. 46**), la cual permanecerá interna dentro del gabinete, mientras que para la recepción de señal GPS se colocó un conector SMA a 90° con conexión al exterior, lo que permite conectar una antena externa mediante cable coaxial, como la observada en la **Fig. 47**.



**Fig. 44** - Conector GX16 4 pines.



**Fig. 45** - Conector GX16 7 pines.



**Fig. 46** - Antena GSM con conector SMA.



**Fig. 47** - Antena GPS con conector SMA.



El montaje del dispositivo, con sus conectores colocados y cableados se aprecia en la **Fig. 48**, destacándose además la presencia de su batería de litio recargable con su módulo de carga.



**Fig. 48** - Montaje dispositivo de rastreo.

En cuanto al dispositivo de visualización, se diseñó y se utilizó impresión en 3D para crear un gabinete que sostiene la pantalla mediante tornillos de fijación. Este diseño se puede ver en **Fig. 49** y **Fig. 50**, donde se aprecian los 4 puntos de fijación de los tornillos y el orificio para colocar un conector GX16 de 4 pines, el cual se utiliza para alimentación y CAN BUS. En la **Fig. 51** se observa el montaje de los componentes, cableados a la Raspberry PI que forma un conjunto con la pantalla.



**Fig. 49** - Diseño gabinete dispositivo de visualización.



**Fig. 50** - Diseño gabinete dispositivo de visualización.



**Fig. 51** - Montaje dispositivo de visualización.



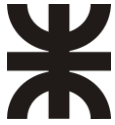
**Fig. 52** - Vista frontal de dispositivo de visualización.



**Fig. 53** – Vista lateral de dispositivo de visualización.

Una vez montados los dispositivos, se procedió a realizar una prueba de funcionamiento, donde se comprobó el correcto funcionamiento de los siguientes ítems:

- Correcto funcionamiento del circuito de alimentación.
- Correcto funcionamiento del sistema de carga de batería.
- Correcto funcionamiento del sensor acelerómetro.



- Correcto funcionamiento de comunicación CAN BUS.
- Correcto funcionamiento de las comunicaciones GPS y GPRS.
- Correcto funcionamiento de las entradas analógicas.
- Correcto almacenamiento en base de datos del servidor.

### **Análisis del sistema en campo**

Una vez montado el prototipo, se realizó la prueba con un vehículo, analizándose cada una de sus partes.

#### **Sistema GPS**

Se analizó la disponibilidad y precisión de la antena GPS en base a distintas condiciones, una de ellas fue al encontrarse en cielo abierto, es decir, sin obstáculos entre la antena GPS y los satélites, donde se obtuvo una precisión en el rango de los 5 metros, siendo aceptable para el rastreo vehicular. En cuanto a la velocidad obtenida por el sistema GPS, representa correctamente la velocidad observada en el velocímetro del vehículo.

Un caso opuesto a esto se da en zonas densamente pobladas, donde los edificios de elevada altura representan un obstáculo para la recepción satelital, apreciándose una notable disminución de la precisión de la ubicación actual, aumentando la variación con la posición real hasta 100 metros. Esto afecta también a la información de velocidad, que se aprecia distorsionada respecto a la velocidad real. En casos donde la cantidad de edificios es elevada, formándose así un cañón urbano, se apreció una pérdida señal GPS, no pudiendo actualizar la ubicación ni velocidad actual hasta encontrarse en una zona con menos obstáculos.

Cuando el vehículo se encuentra con la señal GPS totalmente bloqueada, ya sea por estar bajo techo, circular por un túnel, o encontrarse debajo de una densa arboleda, no fue posible obtener la ubicación actual.

#### **Sistema GPRS**

Al momento de enviar datos al servidor, se observó una diferencia en la frecuencia de actualización de datos en distintas condiciones de red móvil.

En zonas urbanas, en su mayoría con buena cobertura móvil, se actualizan los datos al servidor cada un promedio de 5 segundos, mientras que en zonas donde la cobertura es deficiente, se observó que esa frecuencia de actualización aumentó hasta 20 segundos.

Cabe destacar que en zonas donde no hay cobertura móvil, no es posible realizar el envío de información al servidor.

#### **Sensores**

El sistema fue correctamente probado en vehículos que poseen sensores y relojes analógicos, que permiten obtener la información correcta mediante la lectura analógica de estos sensores, con la excepción de las revoluciones del motor, cuyo funcionamiento se trata de una señal senoidal o de pulsos que entrega el alternador del vehículo, y cuya relación entre la frecuencia de la señal, y la frecuencia de las revoluciones del motor dependen de cada vehículo. En el caso del vehículo donde se realizaron las pruebas, la relación es de 1:2, es decir, que las revoluciones por minuto del motor (RPM) será el doble que la frecuencia obtenida del alternador. En **Fig. 54** se observa la señal de salida del alternador, con una frecuencia de 310Hz, lo que representa 620 RPM, como se aprecia en la pantalla de visualización y el reloj analógico de RPM de la **Fig. 55**.

---

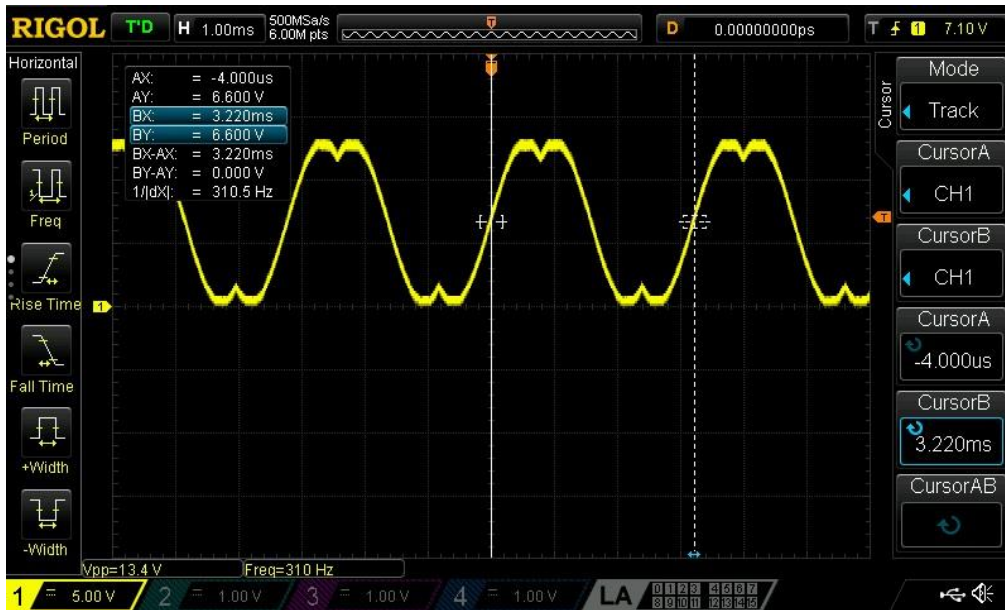


Fig. 54 - Señal de revoluciones de un alternador.



Fig. 55 - Relojes analógicos y pantalla de visualización.



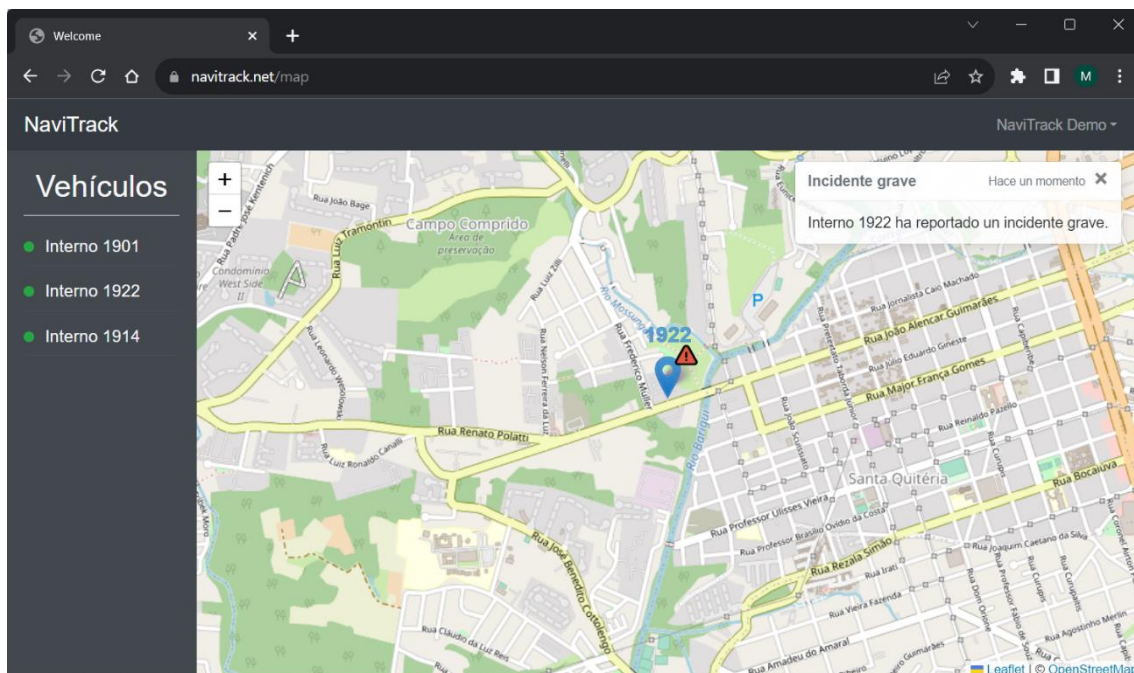
## Dispositivo de visualización

Al analizar la frecuencia de actualización de parámetros en la pantalla de la Raspberry PI, se observó que la información en pantalla se actualiza con una frecuencia de 5 veces por segundo, obteniendo así una rápida respuesta hacia el conductor, siendo esta rápida actualización de importancia para los parámetros de RPM y velocidad.

## Aplicaciones web y móvil

Para probar el correcto funcionamiento de las aplicaciones, se procedió a observar en tiempo real como el marcador de ubicación realmente representaba la ubicación actual, observando cómo cambia su posición, y como sus parámetros en pantalla se actualizan representando lo observado por los relojes indicadores del vehículo.

Respecto al sistema de alertas, no se realizó la prueba en vehículos, sino con potenciómetros conectados al dispositivo rastreador a manera de simular los sensores analógicos. Para simular un incidente grave, se procedió a invertir la posición del dispositivo, de manera de representar un vuelco del vehículo. La alerta de incidente grave se puede observar en la **Fig. 56**, apreciándose una notificación y un cambio de ícono. De igual manera, en la aplicación móvil se observa el cambio de ícono en la **Fig. 57**, y la notificación recibida en la **Fig. 58**.



**Fig. 56** - Aplicación web con alerta de incidente grave.

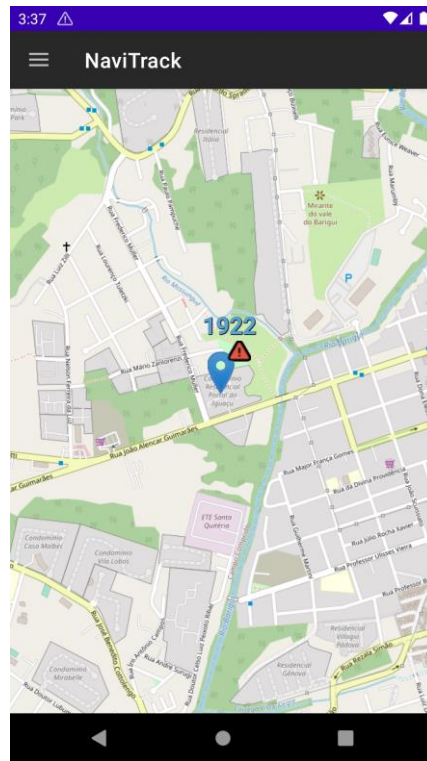


Fig. 57 - Marcador con alerta en aplicación móvil.

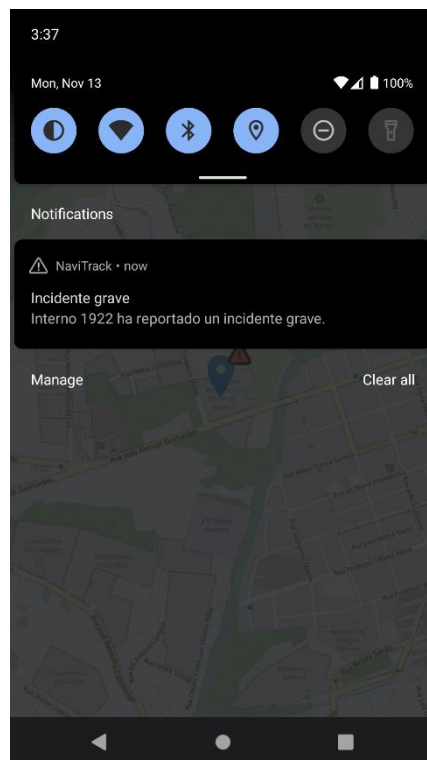


Fig. 58 - Notificación de incidente grave.

### Presentación final del prototipo

Una vez concluidos los análisis y pruebas, se procedió a cerrar los gabinetes de los montajes, obteniendo así el prototipo final, siendo una de sus partes el dispositivo de rastreo de la **Fig. 59**, donde se aprecian los conectores GX16 para alimentación, CAN BUS y entradas analógicas, y además, el conector SMA para la antena GPS externa.

Otra de las partes del prototipo final es el dispositivo de visualización, el cual se observa en funcionamiento en la **Fig. 60**, donde se aprecia su único conector para alimentación y CAN BUS.

El prototipo final completo se observa en la **Fig. 61**, encontrándose al dispositivo de rastreo, al cual se encuentra conectada una antena GPS externa, además de su cable de alimentación y comunicación conectado. De igual manera, se encuentra conectado y en funcionamiento el dispositivo de visualización.



**Fig. 59** - Dispositivo de rastreo.



**Fig. 60** - Dispositivo de visualización.





**Fig. 61** - Prototipo final.



## Conclusiones

Mediante el desarrollo del presente proyecto, se cumplieron los objetivos planteados, incluyendo el desarrollo de un dispositivo de bajo coste, y adaptable a diferentes tipos de vehículos, brindándose así la solución al problema de control de unidades para transportes de carga y pasajeros.

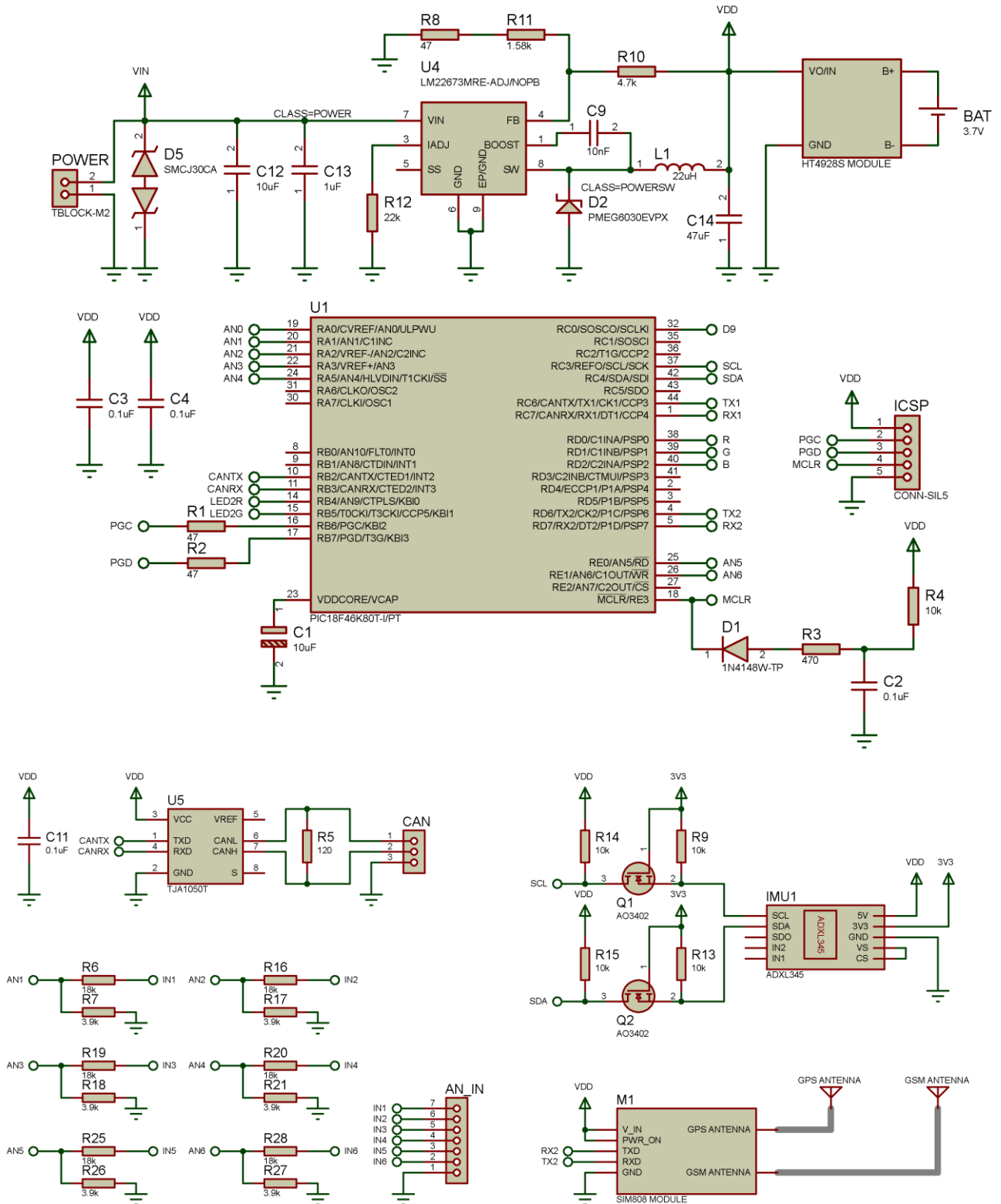
El desarrollo de las aplicaciones web y móvil, permite el fácil rastreo de vehículos en cualquier momento, cumpliendo con los objetivos de ser simple e intuitiva.

Actualmente, la utilización de sensores analógicos en los vehículos se encuentra en disminución, debido a que un número mayor de fabricantes de vehículos optan por utilizar sensores digitales, los cuales se comunican a la computadora central del vehículo mediante protocolos que utilizan la red física del CAN BUS, por lo que, con una modificación de firmware, este prototipo desarrollado podrá tomar la información necesaria directamente desde este bus de comunicación, evitándose así la necesidad de un nuevo desarrollo de hardware para adaptarse a las nuevas tecnologías.

Se encontraron desventajas en cuanto a la exactitud del sistema GPS en diferentes condiciones de disponibilidad satelital, apreciándose una dificultad en realizar un rastreo vehicular en zonas densamente pobladas. Otro punto débil es debido a la falta de cobertura GSM en las rutas y zonas rurales, lo que limita el rastreo a zonas urbanas. Estas desventajas son producto de un dispositivo de bajo coste, ya que podría utilizarse comunicación satelital



## Anexo I – esquemático dispositivo de rastreo





## Bibliografía

- [1] Eduardo García Breijo, *Compilador CCS y Simulador Proteus para Microcontroladores PIC*, 2da edición, Marcombo, 2009
- [2] José Manuel Huidobro, *Comunicaciones Móviles. Sistemas GSM, UTMS y LTE*, 1ra edición, RA-MA S.A., 2012
- [3] Microchip, *PIC18F66K80 Family Enhanced Flash Microcontrollers with ECAN XLP Technology Data Sheet*, (DS30009977G, 2017)
- [4] Microchip, *MCP2515 Family Data Sheet*, (DS20001801K, 2021)
- [5] Texas Instruments, *LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator datasheet (Rev. G)*, (SNVS124G, 2023)
- [6] Texas Instruments, *LM22673/-Q1 42 V, 3 A SIMPLE SWITCHER® , Step-Down Voltage Regulator with Features (Rev. O)*, (SNVS586O, 2014)
- [7] Philips Semiconductors, *TJA1050 High speed CAN transceiver*, (9397 750 12157, 2003)
- [8] Texas Instruments, *Introduction to the Controller Area Network (CAN)*, (SLOA101B, 2016)
- [9] SIMCOM, *SIM800 Series AT Command Manual*, (SIM800 Series\_AT Command Manual\_V1.09, 2015)
- [10] *ADXL345 Data Sheet*. (2021). *ADXL345 (Rev. G)*. Recuperado 22 de octubre de 2023, de <https://www.analog.com/media/en/technical-documentation/data-sheets/adxl345.pdf>
- [11] *Raspberry Pi Documentation - Raspberry Pi Hardware*. (2023). Recuperado 23 de octubre de 2023, de <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>