

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN PROTOTIPO DE MONITOREO VEHICULAR CONTROLADO MEDIANTE UNA APLICACIÓN CELULAR

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES**

Byron Javier Freire Ramírez

byron.freire01@epn.edu.ec

Edison David Armijos Chamba

edison.armijos@epn.edu.ec

DIRECTOR: ING. LEANDRO ANTONIO PAZMIÑO ORTIZ, MSC.

leandro.pazmino@epn.edu.ec

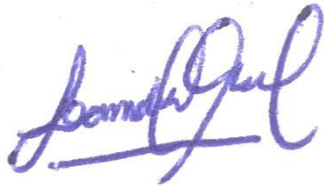
CODIRECTOR: ING. MONICA DE LOURDES VINUEZA RHOR, MSC.

monica.vinueza@epn.edu.ec

Quito, julio 2021

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por los Sres. Freire Ramírez Byron Javier y Armijos Chamba Edison David, como requerimiento parcial a la obtención del título de TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES, bajo nuestra supervisión:



MSc. Leandro Pazmiño

DIRECTOR DEL PROYECTO

MSc. Mónica Vinuesa


CODIRECTORA DEL PROYECTO

DECLARACIÓN

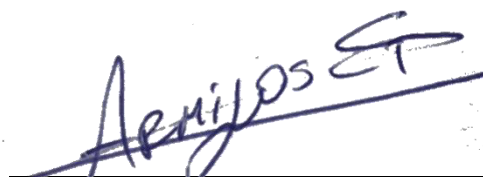
Nosotros Freire Ramírez Byron Javier con CI: 1727015222 y Armijos Chamba Edison David con CI: 1725300790 declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, somos titulares de la obra en mención y otorgamos una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entregamos toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.



Freire Ramírez Byron Javier



Armijos Chamba Edison David

DEDICATORIA

Este documento se lo quiero dedicar especialmente a mis padres, ya que, gracias a sus aportes, a su amor, y su inmensa bondad he podido llegar a estas instancias, también quiero dedicarlo a todas las personas cercanas que me apoyaron durante todo este proceso de desarrollo de la tesis. Gracias por toda su ayuda.

Byron Freire

AGRADECIMIENTO

Un agradecimiento especial a mi familia por haber estado siempre conmigo y haberme apoyado durante todo el ciclo universitario, gracias por permitirme cumplir con esta gran meta. también agradezco la ayuda de mis maestros por haberme transferido todos sus conocimientos, también agradezco a mis compañeros que hicieron más agradable la vida en la universidad.

Byron Freire

DEDICATORIA

Dedicado este proyecto con mucho afecto a:

Mis padres Alonso y Maryuri quienes me apoyaron de forma incondicional y tuvieron la paciencia de estar ahí durante este largo proceso de formación.

Mis hermanos Javier y Lizeth y toda mi familia que siempre me han brindado su apoyo y han estado pendientes de todo este proceso académico y me han alentado a seguir adelante.

Mis amigos quienes me ayudaron en momentos difíciles y me brindaron la mano para estar en este punto de mi vida, cumpliendo más propósitos planteados y tener otra visión del mundo.

Edison Armijos

AGRADECIMIENTO

Primeramente, quiero agradecer a Dios por permitirme cumplir una meta más en mi vida. Por estar ahí protegiéndome y cuidándome en cada momento.

Agradezco a mis padres Alonso y Maryuri, por ser el pilar para seguir con mis metas y propósitos, por guiarme y motivarme en los momentos de flaqueza, enseñándome a ser perseverante y ver que con esfuerzo todo es posible.

Gracias a mis hermanos Javier y Lizeth, por apoyarme en los momentos de tristeza, brindándome su mano para levantarme y con sus ocurrencias sacarme una sonrisa olvidándome así de los problemas por un momento.

Gracias a mis compañeros y amigos(as) por hacer más tolerable la vida universitaria y acompañarme durante ese largo proceso, buscando el lado bueno a cada problema que se nos presentaba e incluso ver el lado gracioso.

También quiero agradecer a los ingenieros que tuve el agrado de conocer y compartir aula, ya que me enseñaron innumerables cosas y no solo del ámbito de mi carrera sino de experiencias vividas y me compartieron una noción de lo que en verdad es la vida, preparándome mejor y teniendo otra visión de los que es un profesional.

Finalmente agradecer a la ESFOT, por formarme como profesional en electrónica y telecomunicaciones.

Edison Armijos

ÍNDICE DE CONTENIDOS

1	Introducción	1
1.1	Objetivo general	2
1.2	Objetivos específicos	2
1.3	Fundamentos.....	2
	Sistema GPS	2
	Plataforma Arduino	2
	Protocolo de comunicación UART en Arduino	4
	Protocolo de comunicación I2C en Arduino	4
	Comandos AT.....	5
	Módulo GSM SIM900.....	5
2	Metodología.....	7
2.1	Descripción de la metodología usada	7
3	Resultados y Discusión.....	8
3.1	Requerimientos del prototipo	8
	Localización del vehículo	8
	Interfaz de usuario	9
	Orientación del vehículo	9
	Detección de apertura de puertas del vehículo	9
	Tecnología de comunicación	9
	Alimentación del prototipo.....	10
	Automatización del proceso	10
	Estructura del proyecto.....	10
3.2	Determinación del <i>hardware</i> y <i>software</i> necesario para el prototipo	10
	Localización del vehículo	11
	Interfaz de usuario	12
	Orientación del vehículo	13
	Detección de aperturas de las puertas del vehículo.....	14

Tecnología de comunicación	15
Alimentación del prototipo.....	16
Automatización del proceso	17
Estructura del prototipo.....	19
3.3 Diseño e implementación del prototipo de monitoreo vehicular.....	20
Esquema general del proyecto	20
Diagramas de conexiones de todos los elementos	21
Diseño de la programación	26
Implementación del prototipo	41
3.4 Desarrollo de la aplicación móvil.....	45
Pantalla de inicio.....	45
Pantalla principal	47
Pantalla del Mapa.....	48
Pantalla de la orientación.....	50
Pantalla de las alarmas.....	52
3.5 Pruebas y Análisis de Resultados.....	56
Prueba del GPS.....	56
Prueba del modo alarma del prototipo	57
Prueba de orientación.....	59
3.6 Manual de Uso y Mantenimiento.....	62
4 Conclusiones y Recomendaciones	63
4.1 Conclusiones	63
4.2 Recomendaciones	64
5 REFERENCIAS BIBLIOGRÁFICAS.....	65

ÍNDICE DE FIGURAS

Figura 1.1 IDE Arduino.....	3
Figura 1.2 Ranura para colocar la tarjeta SIM.....	6
Figura 1.3 LEDs de monitoreo del módulo SIM900	6
Figura 3.1 Módulo GPS GY-GPS6MV2	12
Figura 3.2 Sensor magnetómetro HMC5883L.....	13
Figura 3.3 Interruptor magnético	14
Figura 3.4 Módulo SIM 900.....	15
Figura 3.5 Baterías de litio	16
Figura 3.6 Relevador.....	17
Figura 3.7 Módulo de carga para baterías de litio	17
Figura 3.8 Arduino Mega2560.....	19
Figura 3.9 Esquema general del funcionamiento del proyecto	21
Figura 3.10 Conexiones del módulo GPS con el microcontrolador.....	22
Figura 3.11 Conexiones entre el microcontrolador y el módulo SIM900.....	23
Figura 3.12 Conexión entre el microcontrolador y el sensor magnetómetro	24
Figura 3.13 Conexiones entre el Arduino y los interruptores magnéticos	24
Figura 3.14 Circuito de conmutación de las fuentes de alimentación	25
Figura 3.15 Conexiones entre módulo de carga y las baterías de litio.....	25
Figura 3.16 Diagrama de flujo general	26
Figura 3.17 Función Serial.begin().....	27
Figura 3.18 Comando AT+IPR.....	28
Figura 3.19 Comando AT+CMGF	28
Figura 3.20 Comando AT+CMGR.....	29
Figura 3.21 Comando AT+CNMI.....	29
Figura 3.22 Función para configurar los puertos	30
Figura 3.23 Diagrama de flujo de lectura de un SMS	31
Figura 3.24 Selección del proceso a realizar.....	33
Figura 3.25 Diagrama de flujo para solicitar ubicación	34
Figura 3.26 Función para borrar el buffer del puerto GPS.....	34
Figura 3.27 Función para estabilizar los datos del GPS	35
Figura 3.28 Función para enviar coordenadas a la interfaz.....	36
Figura 3.29 Diagrama de flujo de la orientación del vehículo	37
Figura 3.30 Diagrama de flujo activación de modo alarma.....	39
Figura 3.31 Diagrama de flujo de lectura de los sensores magnéticos.....	40

Figura 3.32	Montaje de los elementos electrónicos.....	41
Figura 3.33	Conexiones de los componentes electrónicos del prototipo	42
Figura 3.34	Construcción de la maqueta del vehículo	42
Figura 3.35	Colocación de los sensores magnéticos en las puertas del vehículo.....	43
Figura 3.36	Armado del circuito de conmutación de baterías	43
Figura 3.37	Armado del circuito de carga de las baterías.....	44
Figura 3.38	Implementación final del prototipo	44
Figura 3.39	Diagrama de flujo general de la aplicación	45
Figura 3.40	Pantalla de inicio	46
Figura 3.41	Diagrama de flujo pantalla inicio.....	47
Figura 3.42	Pantalla principal	47
Figura 3.43	Diagrama de flujo pantalla principal.....	48
Figura 3.44	Pantalla "Mapa".....	49
Figura 3.45	Diagrama de flujo de la pantalla "mapa".....	50
Figura 3.46	Pantalla orientación.....	51
Figura 3.47	Diagrama de flujo pantalla orientación	52
Figura 3.48	Pantalla alarmas	53
Figura 3.49	Diagrama de flujo pantalla alarmas 1/3	54
Figura 3.50	Diagrama de flujo pantalla alarmas 2/3	55
Figura 3.51	Diagrama de flujo pantalla de alarma 3/3	56
Figura 3.52	Información de las puertas del vehículo recibidas en la aplicación.....	58
Figura 3.53	Llamada de alerta de movimiento.....	59
Figura 3.54	Datos obtenidos de la posición 1	60
Figura 3.55	Datos obtenidos de la posición 2.....	60
Figura 3.56	Datos obtenidos de la posición 3.....	61
Figura 3.57	Video del funcionamiento del prototipo.....	62
Figura 3.58	Video del mantenimiento del prototipo	62

ÍNDICE DE TABLAS

Tabla 1.1	Descripción de botones del ID Arduino	3
Tabla 1.2	Comandos AT más frecuentes	5
Tabla 3.1	Comparativa de los módulos de localización	11
Tabla 3.2	Características técnicas GPS GY-GPS6MV2	12
Tabla 3.3	Características técnicas magnetómetro HMC5883L	13
Tabla 3.4	Características del Switch-MC38	14
Tabla 3.5	Características Técnicas GSM SIM900	15
Tabla 3.6	Consumo del prototipo	16
Tabla 3.7	Comparación Raspberry Pi Vs Arduino	18
Tabla 3.8	Características del Arduino Mega2560	19
Tabla 3.9	Asignación de puertos	27
Tabla 3.10	Configuración de puertos de salida.....	30
Tabla 3.11	Configuración de puertos de entrada.....	30
Tabla 3.12	Asignación de valores a los puntos cardinales.....	38
Tabla 3.13	Comparación de coordenadas	56
Tabla 3.14	Rango de valores que representan los puntos cardinales.....	59
Tabla 3.15	Resultados de las pruebas del sensor de orientación	61

RESUMEN

El presente proyecto de titulación se planteó con la finalidad de proponer una solución al problema de robos de automotores que actualmente existe en el país, para lo cual se ha propuesto el diseño, construcción e implementación de un prototipo de monitoreo vehicular, el cual está compuesto de tres etapas, que son: un sistema de rastreo, un sistema de alarma de puerta abierta y un sistema de orientación basado en los puntos cardinales, todas las etapas podrán ser controladas a través de una interfaz que estará a disposición única del dueño del automotor.

El sistema emplea componentes con muy buenas características, las cuales le permitirán acceder a las tecnologías requeridas para el óptimo funcionamiento del proyecto, tales como la red de telefonía celular empleada para la comunicación del sistema, la red satelital para la obtención de las coordenadas geográficas, y la implementación de sensores que ayudarán a complementar el monitoreo del vehículo.

PALABRAS CLAVE: módulo GSM, placa Arduino, módulo GPS, App inventor.

ABSTRACT

The purpose of this degree project was to propose a solution to the problem of vehicle theft that currently exists in the country, therefore the design, construction and implementation of a vehicle monitoring prototype has been proposed. It is composed of three stages, which are: a tracking system, an open-door alarm system and an orientation system based on the cardinal points, all stages can be controlled through an interface that will be at the sole disposal of the owner of the automotive.

The system uses components with very good characteristics, which allow to access to the technologies required for the optimal functioning of the project, such as the cellular network employed for the communication of the system, the satellite network for obtaining the geographical coordinates, and the implementation of sensors for monitoring the vehicle.

KEYWORDS: *GSM module, Arduino board, GPS module, App inventor.*

1 INTRODUCCIÓN

Se han registrado 1196 robos de vehículos solo en la ciudad de Quito en el 2018 según datos de la Policía Nacional y en el 2019 esa cifra se incrementó un 23%, lo que evidencia un grave problema en lo que se refiere a seguridad vehicular en el país y las pocas acciones que toma la policía y el gobierno para frenar o interrumpir el crecimiento de estos actos ilícitos [1].

El robo de vehículos aumenta diariamente en Quito y en todo el país en general. El incremento de la delincuencia que se refleja exige tomar medidas para reducir el riesgo de perder los vehículos a manos de los antisociales. En la actualidad, existen múltiples sistemas que buscan mitigar los problemas de inseguridad vehicular que se tiene en el país, ya sea sistemas mecánicos como lo son seguros en los frenos o volantes, así como sistemas más avanzados como alarmas. Pero en muchos casos estos sistemas ya no son seguros debido a que los delincuentes cambian los métodos de robo y exige contar con un sistema de seguridad más robusto [2].

Con el aumento de la inseguridad a nivel nacional y la falta de control en los robos de los vehículos se ve la necesidad de realizar un sistema de monitoreo vehicular con el objetivo de tener un mejor control del automotor y así visualizar aspectos claves de este, a partir de una aplicación móvil, que se usará para conocer la ubicación actual o conocer si las puertas del automotor se abrieron, incluso saber su orientación.

El diseño de este sistema nace para proteger los vehículos y así mitigar en gran medida que sean robados tan fácilmente. Para ello se evaluó qué aspecto se podría obtener de un vehículo, para saber que está siendo intervenido, como lo sería si las puertas se abren o si el vehículo se mueve sin autorización, lo cual ocasionaría que se presenten alertas en una aplicación de manera rápida.

1.1 Objetivo general

Implementar un prototipo de monitoreo vehicular controlado mediante una aplicación celular.

1.2 Objetivos específicos

- Identificar los requerimientos necesarios para la implementación del prototipo
- Analizar el *hardware* adecuado según los requerimientos necesarios para el prototipo.
- Diseñar e implementar el prototipo de monitoreo vehicular.
- Desarrollar la aplicación móvil que permita enlazarse al prototipo de monitoreo vehicular.
- Realizar las pruebas necesarias de funcionamiento.

1.3 Fundamentos

Sistema GPS

El sistema de posicionamiento global (GPS) es una tecnología que permite saber con precisión la ubicación de una persona o vehículo sobre la Tierra. Su sistema está basado en satélites artificiales que se encuentran alrededor del planeta. Para hacer uso de esta tecnología se necesita adquirir un receptor GPS, el cual para funcionar busca establecer conexión con al menos tres satélites. El tiempo que tarde en establecer la conexión dependerá si el receptor GPS se encuentra en un lugar cerrado o en un lugar abierto, ya que en los lugares cerrados tomará más tiempo debido a los obstáculos, pero menos tiempo en lugares abiertos por la línea de vista [2] .

Los datos que proporciona un receptor GPS son: latitud y longitud. Estos datos son conocidos como coordenadas geográficas y se usan para ubicar un punto cualquiera en la superficie de la Tierra, para lo cual se utiliza en combinación con programas informáticos basados en mapas, por ejemplo, Google Maps [2].

Plataforma Arduino

Arduino es una plataforma de *hardware* de código abierto, el cual está basado en una simple placa de circuito impreso que incorpora un microcontrolador reprogramable de la marca Atmel, este dispositivo une el mundo analógico con el digital, lo cual hace posible

conectar a sus pines de entrada toda una variedad de sensores. Su enfoque se especializa en la creación de entornos interactivos [3].

Para controlar las placas Arduino se utiliza un conjunto de instrucciones, las cuales se cargan a su microcontrolador a través de un programa llamado entorno de desarrollo integrado (IDE), mismo en donde se desarrolla la programación. Las instrucciones en las plataformas Arduino están basadas en el lenguaje C++, además de instrucciones propias del mismo programa. Para cargar la programación se necesita conectar el Arduino a una computadora mediante un cable USB, el cual servirá de medio para transferir las instrucciones al microcontrolador. En la Figura 1.1 se observa la pantalla principal del IDE de Arduino [3].

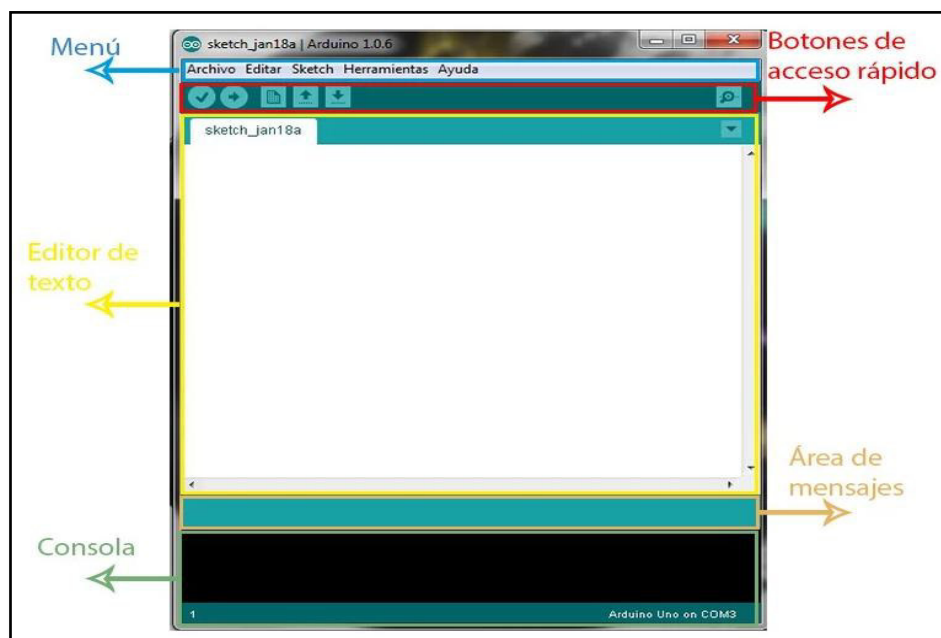








Figura 1.1 IDE Arduino

Las descripciones de los botones de acceso rápido se observan en la Tabla 1.1.

Tabla 1.1 Descripción de botones del ID Arduino

Iconos	Descripción
	Verifica si la sintaxis de las instrucciones es correcta
	Carga el programa en la placa Arduino, tras haber compilado el programa.

Iconos	Descripción
	Crea un programa nuevo
	Abre un programa
	Guarda el programa en el ordenador
	Abre el monitor serial

Protocolo de comunicación UART en Arduino

UART (Transmisor Receptor Asíncrona Universal), es un protocolo de comunicación que transmite los datos en serie. Prácticamente todos los modelos del Arduino disponen al menos de una unidad UART, por cada una de estas con la que se cuente, se tendrá en la placa del Arduino un par de pines denominados como Rx (Recepción) y Tx (Transmisión), los cuales usa el Arduino para comunicarse con otros dispositivos externos, siempre y cuando esos dispositivos manejen el mismo protocolo de comunicación [4].

Los pines Rx y TX de las placas Arduino pueden comunicarse máximo con un dispositivo externo a la vez, es decir, si se tuviera más de un elemento que necesite establecer comunicación con el Arduino, se debe tener más de una unidad UART. Por otra parte, para que la comunicación se realice correctamente, las velocidades de transmisión deben estar configuradas de igual forma; es decir, tanto en el lado del transmisor como en el del receptor deben tener los mismos valores. En cuanto a las conexiones de los dispositivos que utilizan esta comunicación, se lo realiza de forma cruzada, es decir, el pin Tx del dispositivo 1 debe conectarse al pin Rx del dispositivo 2, mientras que el pin Tx del dispositivo 2 debe conectarse al pin Rx del dispositivo 1 [4].

Protocolo de comunicación I2C en Arduino

A más de la comunicación UART existen otros tipos de protocolos, como I2C (Circuito Inter Integrado). En las placas Arduino se pueden identificar fácilmente los pines que manejan esta comunicación, ya que están marcados con las nomenclaturas SDA (*System Data*) y SCL (*System Clock*), a diferencia de la comunicación UART en I2C se pueden conectar varios dispositivos a la vez, los cuales se deben configurar como

esclavos o maestros. Todos los dispositivos que se comunican con el protocolo I2C tienen una dirección hexadecimal única asignada, que es utilizada por el maestro para identificar al dispositivo con el cual se comunicará [5].

La manera en la que se conectan los dispositivos que manejan este protocolo de comunicación es de forma directa; es decir, el SCL del dispositivo 1 con el SCL del dispositivo 2. El SDA del dispositivo 1 con el SDA del dispositivo 2 [5].

Comandos AT

Los comandos AT (*Attention*) no son más que instrucciones que permiten una comunicación entre usuarios y terminales mediante la utilización de un lenguaje que tiene instrucciones codificadas. El principal objetivo que tienen los comandos AT es la comunicación con módems y terminales de telefonía celular. Mediante las instrucciones codificadas se pueden realizar diferentes acciones tales como: hacer llamadas de voz, enviar mensajes de texto, configurar velocidades de transmisión, entre otros. En la Tabla 1.2 se puede observar algunos de los comandos AT más usados [6].

Tabla 1.2 Comandos AT más frecuentes [6].

Comando	Descripción
ATD	Realizar una llamada
ATH	Descolgar llamada
AT+IPR	Seleccionar la velocidad de transmisión
AT+CMGF	Activar modo texto (para el envío de SMS)
AT+CMGS	Enviar el mensaje al destinatario
AT+CMGR	Leer mensajes
AT+CSQ	Nivel de fuerza de la señal
AT+CGMM	Identificador de modelo

Módulo GSM SIM900

El SIM900 es un módulo que puede realizar las mismas funciones que un teléfono celular normal, como: enviar y recibir SMS (Servicio de Mensajes Cortos), también hacer o recibir llamadas telefónicas. Para que el módulo SIM900 adquiera las funciones de un celular, es necesario colocarle un chip de celular de cualquier operadora. Tiene una ranura SIM (*Subscriber Identity Module*) en la parte posterior del módulo, en donde se debe colocar la tarjeta SIM, ver Figura 1.2 [7].



Figura 1.2 Ranura para colocar la tarjeta SIM [7]

El módulo GSM cuenta con tres LEDs (ver Figura 1.3) que indican el estado de la energía o la conectividad. Al observar estos LEDs se determina, si el módulo está trabajando correctamente. A continuación, se detalla la función de cada LED [7].

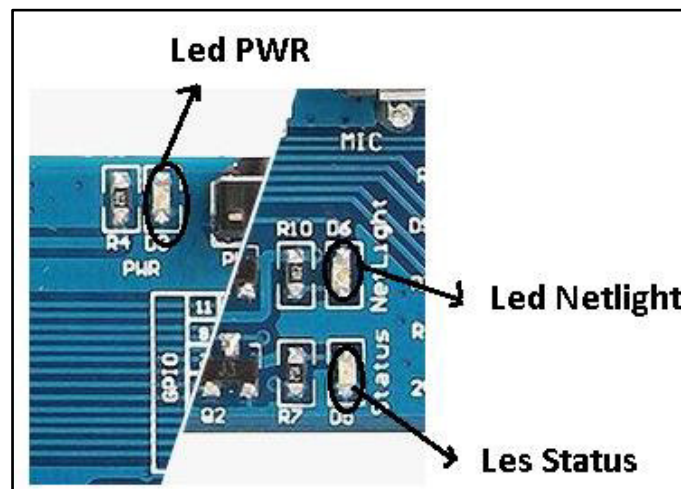


Figura 1.3 LEDs de monitoreo del módulo SIM900 [7]

- **LED PWR:** este LED está conectado a la línea de alimentación del módulo; es decir, si se encuentra encendido, entonces el módulo está recibiendo energía [7].
- **LED estado:** indica el estado de funcionamiento del módulo. Si este se mantiene encendido, el módulo está operativo y listo para trabajar [7].

- **LED *Netlight*:** si parpadea cada tres segundos el módulo se ha conectado a la red celular, pero si parpadea cada segundo significa que aún no ha podido conectarse a la red celular [7].

2 METODOLOGÍA

2.1 Descripción de la metodología usada

Para la alineación del prototipo fue necesario realizar una investigación de las tecnologías que existen en el país con el propósito de identificar aquellas que faciliten obtener coordenadas geográficas como también las que permitan establecer una comunicación a distancia eficiente. Entre las tecnologías de localización se analizaron las siguientes: GPS, Galileo y GLONASS. Para la elección de alguna de estas tecnologías se consideraron algunas características importantes como, qué tan accesibles son, la precisión de localización y su sensibilidad, por otra parte, para la tecnología de comunicación se procedió de igual forma, se consideraron algunas opciones como: GSM y GPRS. Y para elegir alguna de ellas se analizaron los mismos aspectos considerados en la elección de la técnica de localización.

Una vez que fue seleccionado el tipo de tecnología a usar tanto en la localización como en la comunicación del prototipo se realizó una investigación adicional para identificar a aquellos componentes electrónicos que puedan hacer uso de estas tecnologías, actualmente se encontró una gran variedad de opciones en el mercado para lo cual se realizó una comparación y así seleccionar al que posea mejores características como, consumo adecuado de voltaje, precio accesible y que los protocolos de comunicación se adapten al microcontrolador usado. De esta forma fue cómo se procedió a elegir los módulos de localización y de comunicación.

Considerando las demás funciones del prototipo como el sistema de alarma, fue necesario recabar información para identificar qué tipo de sensores pueden generar una señal al momento que una de las puertas del vehículo sea abierta, para lo cual se consideró emplear sensores basados en el efecto hall, los cuales generan una señal cuando tienen cerca un campo magnético como el de un imán y generan otra señal diferente cuando no hay presencia de campo magnético.

Considerando el último aspecto del prototipo, que es la orientación, se requirió hallar sensores que detecten el campo magnético de la Tierra para que así puedan hacer

lectura del mismo, comportándose como una brújula digital, de esta forma al utilizar estos tipos de sensores se pudo reconocer la dirección de los puntos cardinales, por lo cual se agregaron al prototipo.

Para la visualización de la información del auto se desarrolló un programa, para lo cual se usó una plataforma que permite la creación de aplicaciones móviles la cual es llamada *App inventor*, en dicha plataforma se creó la aplicación móvil para mostrar de forma clara la información del vehículo cuando el usuario la solicite, como su ubicación, orientación e incluso a través de esta aplicación puede el usuario activar o desactivar la alarma del auto.

3 RESULTADOS Y DISCUSIÓN

El prototipo implementado en el interior del automóvil es controlado mediante una interfaz, la cual está instalada en el teléfono móvil del usuario, mediante dicha interfaz el usuario puede solicitar información de los sensores y módulos que conforman el prototipo, estos componentes son los que proporcionan los datos del estado del automóvil como: la ubicación actual, la orientación tomando como referencia los puntos cardinales y el estado de las puertas del vehículo.

3.1 Requerimientos del prototipo

Para determinar los requerimientos del prototipo se realizó un análisis en base a la observación, de esta forma se logró identificar las necesidades del proyecto, las cuales son de gran utilidad para la selección de los componentes. Los requerimientos que se identificaron fueron los siguientes:

- Localización del vehículo.
- Interfaz del usuario.
- Orientación del vehículo.
- Detección de apertura de las puertas del vehículo.
- Tecnología de comunicación.
- Alimentación del prototipo.
- Automatización del proceso.
- Estructura del proyecto.

Localización del vehículo

El prototipo debe contar con un módulo de rastreo; es decir, un dispositivo que permita obtener datos de latitud y longitud. Además, dicho dispositivo debe poseer una buena

sensibilidad para que la conexión no se pierda aún en condiciones poco favorables, también debe tener la posibilidad de colocar una antena lo suficientemente larga para colocarla sobre el techo del vehículo y así poseer línea de vistas con los satélites y por tanto mejore la recepción de la señal. Además, el dispositivo debe ser capaz de trabajar en conjunto con un microcontrolador para automatizar el proceso.

Interfaz de usuario

Para que el usuario pueda acceder a las funciones del prototipo se necesita crear una plataforma, en donde se pueda ver e interactuar con cada una de las funciones del prototipo y de igual manera, permita observar la información de los sensores de forma clara y sencilla. Para realizar lo mencionado se necesita hacer uso de algún *software* gratuito que permita crear aplicaciones para los sistemas operativos de los teléfonos móviles y considerando que la mayoría de celulares (El 86%) usan el sistema operativo Android, el *software* a usar debe estar orientado a la creación de aplicaciones en Android [8].

Orientación del vehículo

Para la orientación del vehículo es necesario integrar al prototipo un sensor que sea capaz de detectar y leer el campo magnético de la Tierra; para que actúe como una brújula digital. Además, el mismo debe ser manejado desde un microcontrolador para que realice las mediciones solo cuando se lo solicite, de esta manera el proceso se automatizará.

Detección de apertura de puertas del vehículo

Para detectar en qué instante las puertas del vehículo son abiertas, se necesita emplear un sensor que proporcione una señal o aviso cuando detecte que un objeto se ha movido de su lugar habitual, que en este caso el objeto es la puerta del vehículo. Dicho sensor debe trabajar en conjunto con un microcontrolador para gestionar de mejor manera la información.

Tecnología de comunicación

Para que el prototipo pueda establecer comunicación con la interfaz, se requiere incorporar un elemento que sea capaz de utilizar alguna de las tecnologías empleadas en las redes móviles. Considerando que la información que se enviará y recibirá es pequeña, por lo tanto, no es un factor preocupante la velocidad de la transmisión, pero si lo es el área de cobertura. La red celular será empleada para hacer uso de los mensajes de texto y así enviar y recibir la información, también se empleará el servicio

de voz para alertar al usuario mediante una llamada cuando el vehículo se esté moviendo.

Alimentación del prototipo

Para la alimentación del prototipo se emplea la batería del vehículo, pero también se agrega una batería adicional que se usa como respaldo de energía, cuando el vehículo se apague. Por tanto, para realizar esta conmutación entre las baterías se necesita integrar un circuito adicional que permita detectar el apagado de la batería del vehículo, para que entre en acción la batería de respaldo y así el prototipo se mantenga operativo en todo momento.

Otro requerimiento por considerar es la clase de batería de respaldo que se empleará, ya que debe tener un alto amperaje hora, para que el circuito permanezca el mayor tiempo posible encendido, también se debe emplear elementos que permitan gestionar la recargar de las baterías de forma automática, cuando la fuente principal (batería del vehículo) esté conectada.

Automatización del proceso

El prototipo requiere de un dispositivo centralizado para que controle y gestione las acciones de los demás elementos electrónicos, por lo cual, se necesita integrar un tipo de microcontrolador que sea compatible con los demás elementos que se seleccionarán.

Estructura del proyecto

Para la simulación del vehículo convencional, se construye una maqueta que lo representará, por tanto, se requiere hacer uso de un material maleable, fácil de manipular. Otro requerimiento en la construcción de la maqueta es que las puertas se puedan abrir y cerrar para simular los sistemas de alarma que se implementarán, ya una vez construida la maqueta, se procede a instalar el prototipo en el interior de la misma para posteriormente realizar las pruebas de funcionamiento y las correcciones que se requieran.

3.2 Determinación del *hardware* y *software* necesario para el prototipo

Una vez terminado con el análisis de los requerimientos del proyecto, se procede con la selección de los componentes para la construcción del prototipo.

Localización del vehículo

Para la selección del dispositivo de localización se ha considerado elegir una de las tres opciones que se muestran en la Tabla 3.1. Estas opciones se consideraron, ya que son las más comerciales y todas ellas trabajan en conjunto con un microcontrolador, el cual es uno de los requerimientos para el funcionamiento del proyecto.

Tabla 3.1 Comparativa de los módulos de localización [9] [10] [11]

Características	Módulo GPS Skylab SKM53	GPS Neo 6M	Módulo GPS Skylab SKM53
Voltaje de funcionamiento	5 (V)	5 (V)	5 (V)
Antena	Soldada a la placa	Separada de la placa	Soldada a la placa
Tipo de Comunicación con microcontroladores	UART	UART	UART
Sensibilidad de rastreo	-165 (dBm)	-161 (dBm)	-162 (dBm)
Consumo de energía en modo de rastreo	40 (mA)	45 (mA)	37 (mA)
Costo	Alto	Bajo	Medio

Si se observa en la Tabla 3.1, se nota claramente que el módulo SKM53 tiene la mejor sensibilidad, sin embargo, su antena se encuentra ligada a la placa, por lo tanto queda descartado, ya que otro de los requerimientos es que el dispositivo a elegir tenga la opción de extraer su antena que viene por defecto para remplazarla por una de mayor ganancia y así tener la libertad de colocarla en el lugar más estratégico del vehículo, sin que la longitud del cable sea un impedimento, de esta forma se podrá colocar la antena sobre el techo del vehículo y tener línea de vista con los satélites, dicho esto, también se descartaría el módulo GP-735, ya que su antena viene soldada a su placa, dejando únicamente al módulo GPS Neo 6m.

El módulo GPS Neo 6m (ver Figura 3.1) tiene la ventaja de contar con un conector de tipo U. FL que sirve para conectar cualquier tipo de antena que trabaje con 3 (V), además tiene una buena sensibilidad, sin mencionar que su consumo de energía es muy bajo, ya que está en el orden de los (mA). Las principales características se describen en la Tabla 3.2, teniendo en cuenta estas características se seleccionó este módulo, ya que cumple con los requerimientos identificados para el proyecto [9].

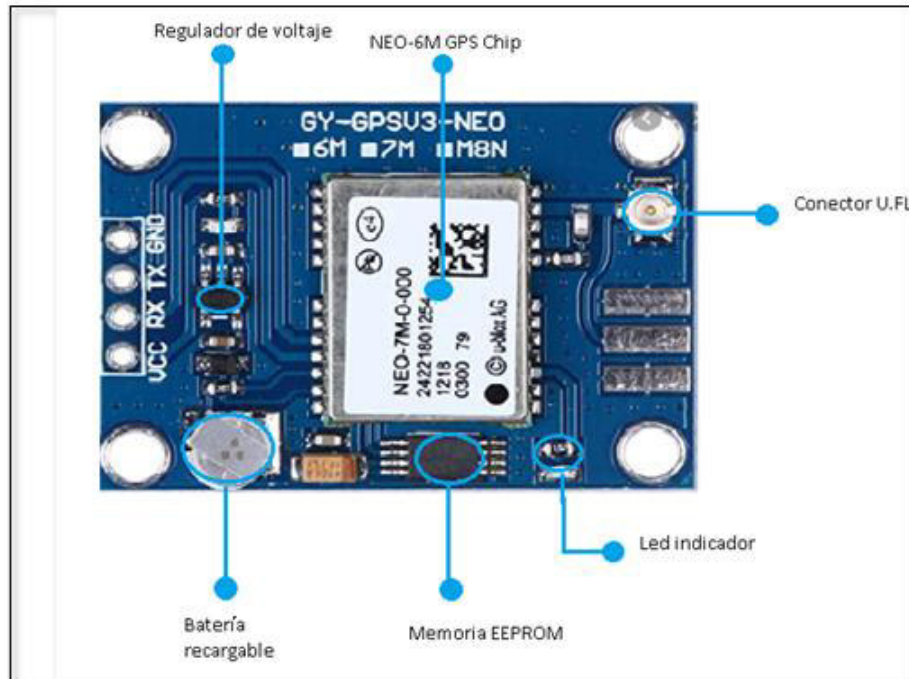


Figura 3.1 Módulo GPS GY-GPS6MV2 [12]

Tabla 3.2 Características técnicas GPS GY-GPS6MV2 [9]

Modelo	GY-GPS6MV2
Voltaje de funcionamiento	3.3 a 5 (V)
Consumo de corriente	17 (mA)
Sensibilidad de rastreo	-161 (dBm)
Interfaz de comunicación	UART (puerto serie)
Velocidad de transmisión	9600 baudios por defecto
Microprocesador	Neo-6M-0-001
Memoria de almacenamiento	EEPROM
Dimensiones	25 (mm) * 35 (mm)
Tiempo de inicio por primera vez	38 (seg) en promedio

Interfaz de usuario

Para el desarrollo de la interfaz gráfica se optó por utilizar el *software App inventor*, ya que está especializado para el desarrollo de aplicaciones orientadas a los sistemas operativos Android, el cual es uno de los requerimientos principales para el desarrollo del proyecto.

El *software* App inventor trabaja con bloques para realizar la programación, la cual la hace una herramienta muy intuitiva de usar y por tanto poco complicada a comparación de muchos otros programas que requieren de un nivel de conocimientos más alto para usarlos. Además, con esta plataforma se pueden desarrollar aplicaciones de forma gráfica e interactivas, por estos motivos se ha decidido trabajar con esta plataforma [13].

Orientación del vehículo

Una de las etapas del sistema es la capacidad de detectar la orientación del vehículo, para lo cual se ha decidido emplear el sensor HMC5883L (ver Figura 3.2). La razón por la que se eligió este modelo es porque incorpora la electrónica necesaria para trabajar en conjunto con un microcontrolador, además puede detectar y leer el campo magnético de la Tierra actuando como una brújula digital [14].

El sensor HMC5883L cuenta con la característica de trabajar en dos modos distintos que son: el modo continuo que realiza mediciones constantemente y el modo simple en donde el magnetómetro solo hará una medición cuando el microcontrolador lo ordene. estas funciones resultan muy útiles al momento de trabajar con un microcontrolador. Estas han sido las razones por las cuales se ha considerado emplear este tipo de sensor en el proyecto. Se observa otras características del sensor en la Tabla 3.3 [14].

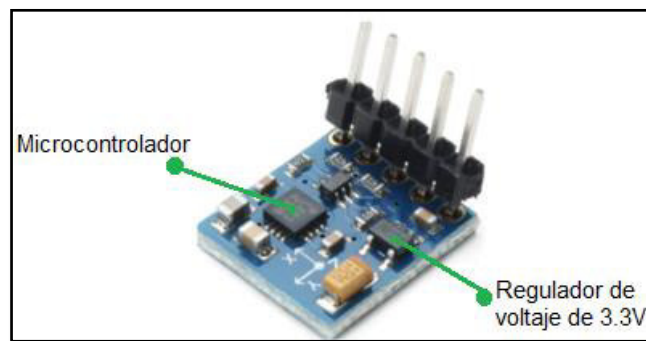


Figura 3.2 Sensor magnetómetro HMC5883L [15]

Tabla 3.3 Características técnicas magnetómetro HMC5883L [16]

Modelo	GY-GPS6MV2
Alimentación	3 a 5 (V)
Interfaz de comunicación	I2C
Dirección del registro I2C	0X1E
Modos de funcionamiento	Medición continua o simple
Medidas	14 (mm) * 15 (mm)

Detección de aperturas de las puertas del vehículo

Para cumplir con este requerimiento se optó por el uso de un interruptor de tipo magnético (ver Figura 3.3). Se ha elegido este dispositivo, porque está diseñado especialmente para alertar cuando una puerta ha sido abierta. Estos interruptores son utilizados principalmente en las casas como sistemas de seguridad, pero se los puede adaptar fácilmente en cualquier tipo de puerta como las de un vehículo. Además, tienen la ventaja de que pueden trabajar en conjunto a un microcontrolador lo cual automatizará esta etapa. En la Tabla 3.4 se puede observar algunas características más del dispositivo seleccionado.

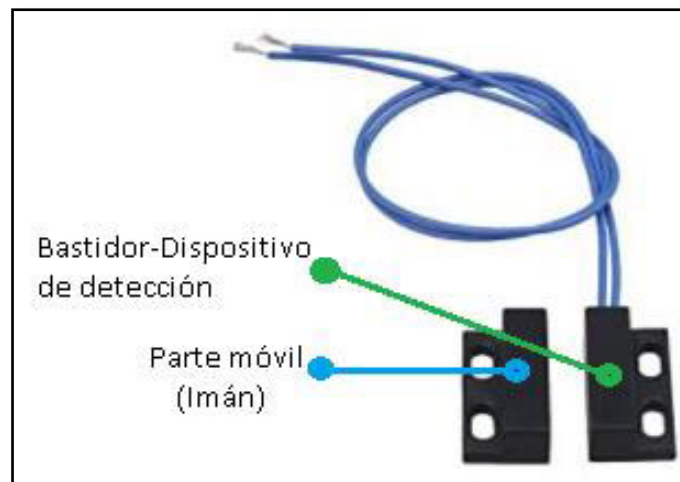


Figura 3.3 Interruptor magnético

Tabla 3.4 Características del Switch-MC38 [17]

Tipo de sensor	Switch-MC38
Paso de voltaje máximo	12 (V)
Corriente máxima	1 (Am)
Máximo consumo	10 (W)
Separación de brecha	15 (mm)
Modo de contacto	NC (normalmente cerrado)
Dimensiones	30 (mm)*15 (mm)

Tecnología de comunicación

Para solventar la necesidad de crear un enlace de comunicación entre la interfaz y el prototipo se optó por utilizar la tecnología 2.5G, ya que a pesar de no ser la más actual sí abarca una amplia cobertura y considerando que la información que se manejará son de unos pocos caracteres la velocidad de transmisión no será un factor importante, por estos motivos se ha elegido trabajar con dicha tecnología.

El módulo que se ha elegido para trabajar es el SIM900 (ver Figura 3.4). El cual se lo ha considerado por el hecho de usar la tecnología GSM (Sistema Global para las Comunicaciones Móviles) y GPRS (Servicio General de Paquetes vía Radio) que corresponden a 2G y 2.5G. Estas tecnologías entre los servicios que ofrecen están el envío de SMS y el de servicio de voz, los cuales son servicios necesarios para el funcionamiento del prototipo, por tales razones se ha elegido el módulo. Se muestran otras características importantes en la Tabla 3.5 [7].

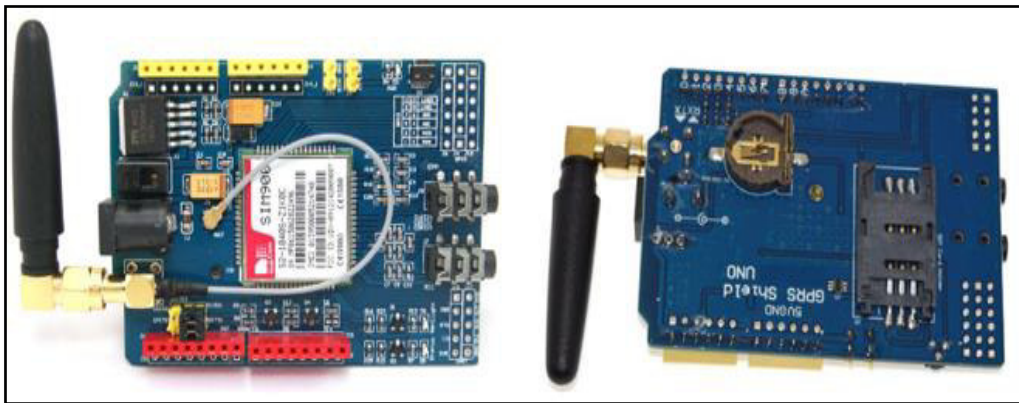


Figura 3.4 Módulo SIM 900 [7]

Tabla 3.5 Características Técnicas GSM SIM900 [7]

Características	Módulo SIM 900
Voltaje de alimentación chip SIM900	3.2 a 4.8 (V)
Voltaje de alimentación usando Jack de CC	5 a 12 (V)
Corriente de consumo	1 (mA) en modo ahorro
Corriente máxima	2 (mA)
Control vía comandos	AT
Tecnologías	GSM/GPRS
Control de encendido	Se controlar por <i>software</i> o por <i>hardware</i>

Alimentación del prototipo

Para la elección de las baterías se considera el consumo de corriente total del prototipo en la Tabla 3.6 se puede observar el consumo de cada componente.

Tabla 3.6 Consumo del prototipo [7] [9] [16] [18] [19]

Componentes	Corriente
Arduino Mega 2560	600 (mA)
Módulo GPS	17 (mA)
Sensor magnetómetro	26 (mA)
Interruptores magnéticos x4	400 (mA)
Módulo SIM900	30(mA)
Zumbador	15 (mA)
TOTAL	1088 (mA)

Teniendo en cuenta el nivel de consumo total del prototipo, se optó por el uso de baterías de litio (ver Figura 3.5), tomando en cuenta que cada batería de litio tiene un voltaje de 3.7 (V) y una corriente de 3 000 (mAh), por lo que se conectaron dos baterías en paralelo y dos baterías en serie dando como resultado un voltaje de 7.4 (V), con una corriente de 6 000 (mAh), lo cual satisface la exigencia de energía de los distintos componentes del prototipo.



Figura 3.5 Baterías de litio

Para el control de la alimentación se emplea un relevador, mismo que ayuda a conmutar entre la fuente principal y las baterías de respaldo para que así el circuito se mantenga activo cuando la fuente principal se desconecte. El relé que se usará se muestra en la Figura 3.6



Figura 3.6 Relevador

Para recargar las baterías cuando estas se agoten, se utilizará el módulo que se muestra en la Figura 3.7, dicho módulo cuenta con la ventaja de tener protección contra sobre cargas y cortocircuitos, lo cual evitará daños cuando existan anomalías, tanto del circuito que lo alimenta como de las mismas baterías, por estas razones se decidió trabajar con este módulo [20].

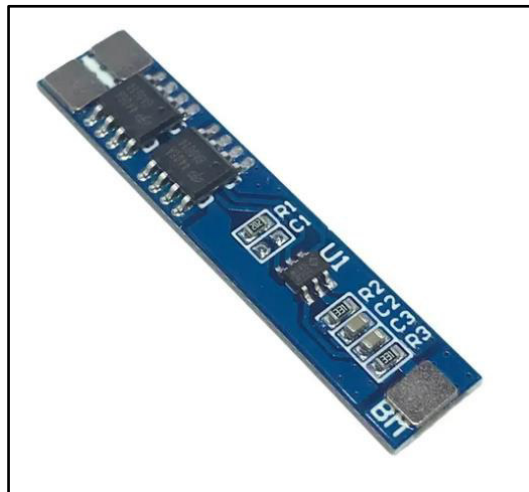


Figura 3.7 Módulo de carga para baterías de litio [20]

Automatización del proceso

Para automatizar el proyecto se requiere hacer uso de un microcontrolador, el cual se encargará de gestionar y manejar toda la información de los sensores y módulos del prototipo convirtiéndose en autónomo. Para la selección del microcontrolador, se realizó una comparación entre dos ecosistemas diferentes (Tabla 3.7) que son: *Raspberry Pi* y *Arduino*. La razón por la que se eligió estos dos elementos para compararlos es porque son los más usados en el mundo de la electrónica y en aplicaciones de automatización como domótica.

Tabla 3.7 Comparación Raspberry Pi Vs Arduino [21]

Raspberry Pi	Arduino
Funciona con un voltaje estándar de 5 (v)	Funciona en un rango de voltaje de 5 a 12 (V)
Utiliza procesadores de la familia ARM	Su procesador es de la familia AVR
Utiliza lenguaje de programación <i>Python</i> , pero se puede usar <i>C</i> , <i>C++</i> y <i>Ruby</i>	Utiliza lenguaje de programación <i>C++</i> , Arduino IDE
Acceso a internet	Requiere de módulos de expansión para acceder a internet
Puede ejecutar múltiples programas a la vez	Se concentra en ejecutar un programa a la vez
Cuenta con una tarjeta micro SD para el almacenamiento	Utiliza una memoria EEPROM para el almacenamiento
Requiere de instalación de <i>software</i> y librerías adicionales para interactuar con sensores	Sus componentes y sensores funcionan de manera integrada
Tiene un costo más elevado a comparación de un Arduino	Tiene un costo más accesible a comparación de un Raspberry Pi
Cuenta con un sistema operativo Linux	No tiene sistema operativo

Como resultado de la comparación, se puede deducir que el *Raspberry Pi* tiene mejores características, como por ejemplo su capacidad de ejecutar varios programas a la vez, sin embargo, esto no es un requerimiento del todo necesario, ya que en el proyecto solo se ejecutará un programa, tampoco necesitará de un sistema operativo complejo como lo es Linux, por tanto, se optó por el Arduino por su simplicidad y bajo costo, además que trabaja con cualquier tipo de sensor, siendo la mejor opción para el proyecto.

Una vez seleccionada la plataforma, se procedió a elegir el modelo de la placa Arduino, en función a la cantidad de puertos seriales que tenga, ya que para el proyecto se necesitará 3 puertos seriales, para lo cual la mejor opción resulto el Arduino Mega2560, el cual se observa en la Figura 3.8.

Las características de la placa Arduino Mega2560 se muestran en la Tabla 3.8, el cual a diferencia de otros modelos posee un número mayor de puertos tanto digitales como analógicos e incluso cuenta con varios puertos de comunicación seriales, siendo muy

importante para el funcionamiento del proyecto. Cada módulo y sensor hará uso de uno de los puertos serial, para recibir y enviar la información correspondiente.

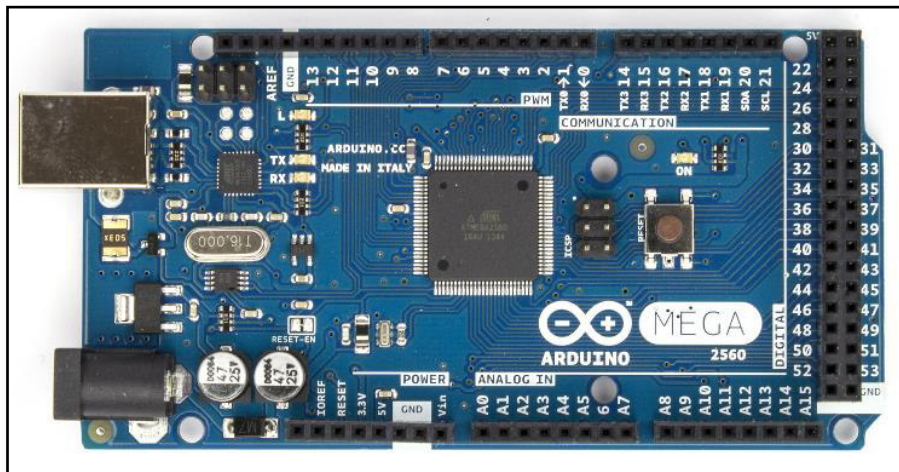


Figura 3.8 Arduino Mega2560 [18]

Tabla 3.8 Características del Arduino Mega2560 [18]

Microcontrolador	ATmega2560
Voltaje de funcionamiento	5 a 12 (V)
Pines digitales E/S	54 (15 proveen salidas PWM)
Voltaje de entrada	7 – 12 (V)
Corriente pines de entrada y salida	40 (mA)
Corriente del pin de 3,3 (V)	50 (mA)
Puertos seriales	4
Velocidad de reloj	16 (MHz)
Memoria flash	256KB
Pines analógicos de entrada	16 desde A0 - A15

Estructura del prototipo

El material que se usó para la construcción de la maqueta del vehículo es madera, ya que es maleable y fácil de manejar. Para el diseño de la estructura de la maqueta se optó por emplear plantillas.

3.3 Diseño e implementación del prototipo de monitoreo vehicular

Esquema general del proyecto

En la Figura 3.9 se puede observar el esquema general del proyecto. El microcontrolador Atmel (contenido en el módulo Arduino) es el que actúa como el núcleo de todo el mecanismo, este microcontrolador se encargará de coordinar y controlar todas las comunicaciones dentro del sistema.

El proceso inicia cuando el usuario utiliza la interfaz para enviar una petición al prototipo, dicha información es recibida en primera instancia por el módulo de comunicación que es el SIM900, luego la información es pasada a Arduino a través de uno de los puertos seriales, luego este procesará la información recibida, para saber qué datos está solicitando el usuario, por ejemplo si desea saber la ubicación de su vehículo o si quiera activar o desactivar el modo alarma (modo en donde se detecta si las puertas son abiertas o si se mueve el vehículo) o si es que desea saber la orientación de su vehículo.

Cabe señalar que cada vez que el usuario solicite alguna petición al prototipo, el Arduino usará el módulo SIM900 para enviar un mensaje de vuelta a la interfaz para señalar que la petición se recibió correctamente y que al cabo de unos pocos segundos obtendrá la información solicitada.

Si la información que solicitó el usuario son los datos del GPS, entonces el Arduino procederá a acceder al puerto serial que está conectado el GPS, ya que ahí podrá obtener los datos de latitud y longitud, los cuales serán enviados en un mensaje de texto a la interfaz del usuario. Cuando reciba los datos, la interfaz colocará automáticamente los datos de latitud y longitud en *Google Maps* y el usuario podrá observar inmediatamente la ubicación de su vehículo.

Cuando se solicite la petición de la orientación, el Arduino solicitará la información al magnetómetro a través del puerto I2C, ya que es la comunicación que maneja ese sensor. Una vez que el microcontrolador obtenga la información, lo enviará al módulo SIM900, para que este lo envíe a la interfaz del usuario, ya en la interfaz el usuario podrá visualizar en qué orientación está su vehículo e incluso la interfaz contará con una pequeña brújula que le servirá al usuario para saber en qué dirección está apuntando el punto cardinal que acabó de recibir.

Cuando el usuario solicite la petición de activar el modo alarma, el microcontrolador estará pendiente tanto de los sensores magnéticos que están en las puertas del

vehículo, como del magnetómetro que indica la orientación del vehículo. Si los datos de estos dos sensores se ven alterados; es decir, si se abren las puertas o si el vehículo de repente cambia de orientación, el microcontrolador usará el SIM900 para notificarle al usuario. Si una de las puertas del vehículo se abre se notificará con un mensaje de texto señalando qué puerta fue abierta, si el vehículo se mueve se notificará mediante una llamada telefónica.

Cabe señalar que cuando está activado el modo alarma, no se podrán hacer peticiones al prototipo como solicitar la ubicación, hasta que se desactive nuevamente el modo alarma, el cual se lo hace con presionar un simple botón desde la interfaz del usuario.

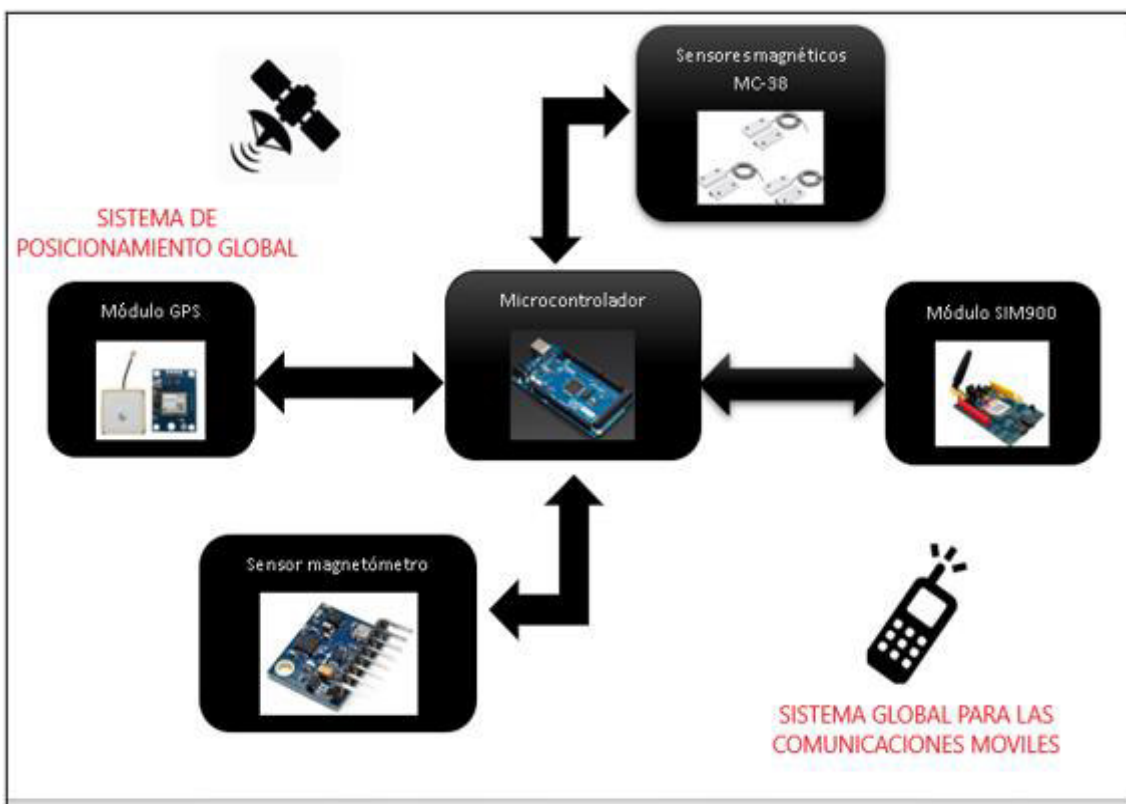


Figura 3.9 Esquema general del funcionamiento del proyecto

Diagramas de conexiones de todos los elementos

En la siguiente sección se muestra las conexiones de cada uno de los elementos electrónicos con el microcontrolador. Todos los componentes están montados sobre una baquelita perforada.

Conexiones del microcontrolador con el GPS

Para la comunicación del módulo GPS con Arduino se usa el puerto serial número 2 (pin 16 y 17), ya en la programación se establecerá la velocidad de transmisión con la que se trabajará en este puerto serial. El GPS necesita energizarse con un voltaje de 3.3 a 5 (V) para trabajar correctamente, por lo tanto, se usa un pin digital del Arduino (Pin 8) que proveerá de 5 (V) y finalmente se une los GND de ambos dispositivos. En la Figura 3.10 se pueden observar las conexiones de mejor manera.

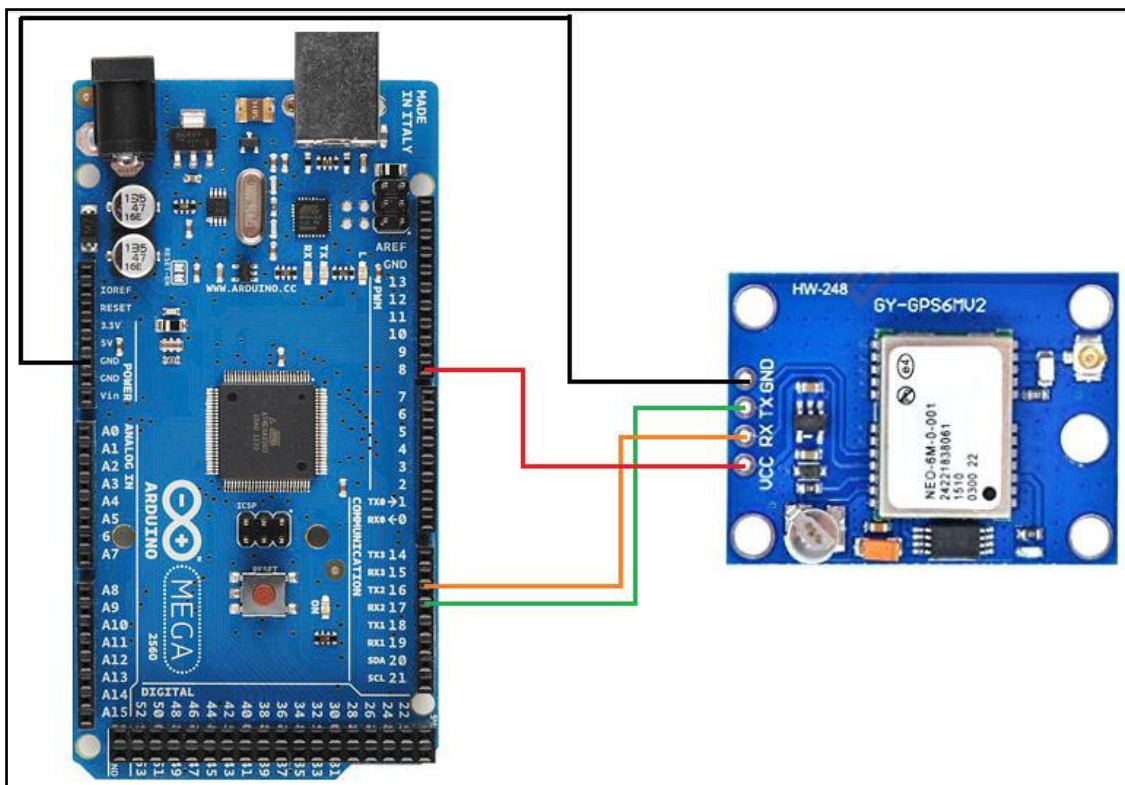


Figura 3.10 Conexiones del módulo GPS con el microcontrolador

Conexiones del microcontrolador con el módulo SIM900

Para la comunicación entre el Arduino y el SIM900 se usa el puerto serial número 1 (Pin 18 y 19), más adelante en la programación se establecerá una velocidad de transmisión para este puerto. Para la alimentación del SIM900 se utiliza el pin (Vcc) del Arduino, el cual proveerá al SIM900 de 5 (V). Para el encendido automático por programación del módulo SIM900 se usará el pin digital número dos de la placa Arduino, por último, se unen los GND de ambos elementos. En la Figura 3.11 se puede observar las conexiones realizadas.

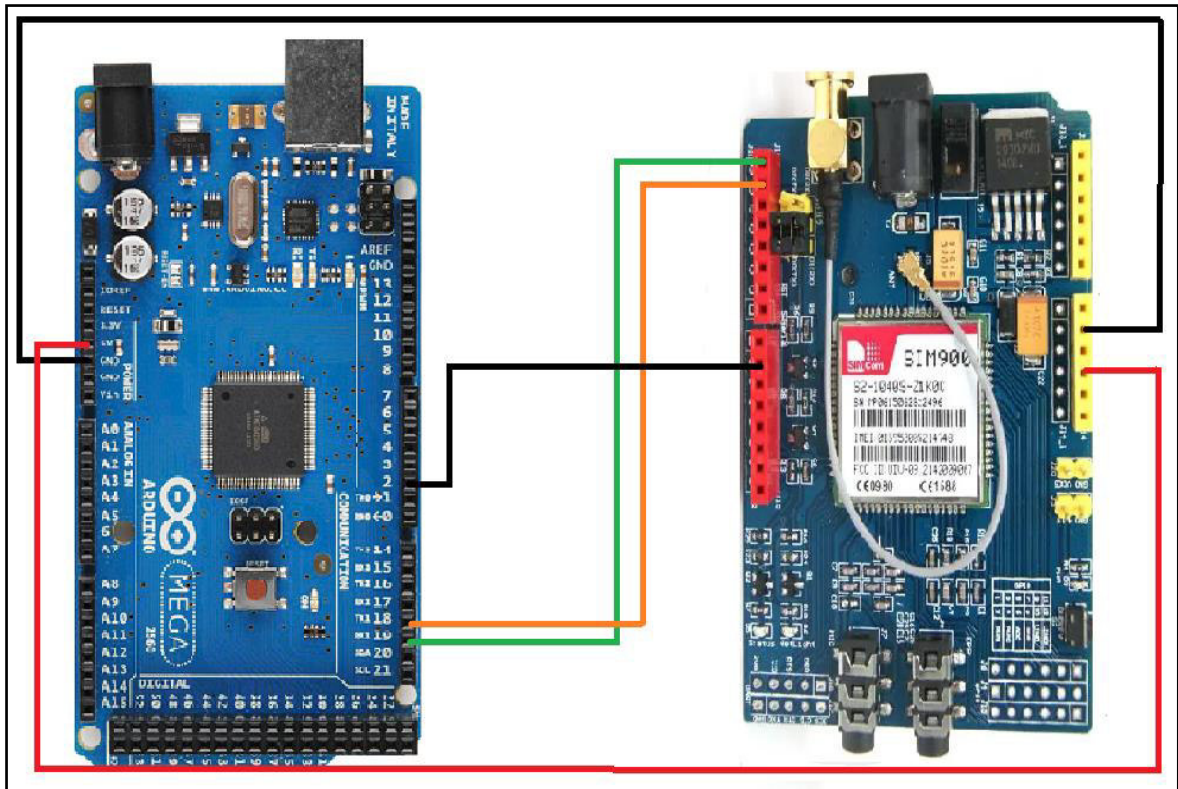


Figura 3.11 Conexiones entre el microcontrolador y el módulo SIM900

Conexiones del microcontrolador con el sensor magnetómetro

Para la alimentación del sensor se utilizará el pin de 3.3 (V) del Arduino y el pin de GND. Hay que considerar que el protocolo de comunicación que maneja el sensor es de tipo I2C, por lo tanto, para que el microcontrolador y el sensor establezcan la comunicación se usará los pines de SDA y SCK del Arduino que corresponden a los pines 20 y 21. En la Figura 3.12 se puede observar las conexiones de mejor manera.

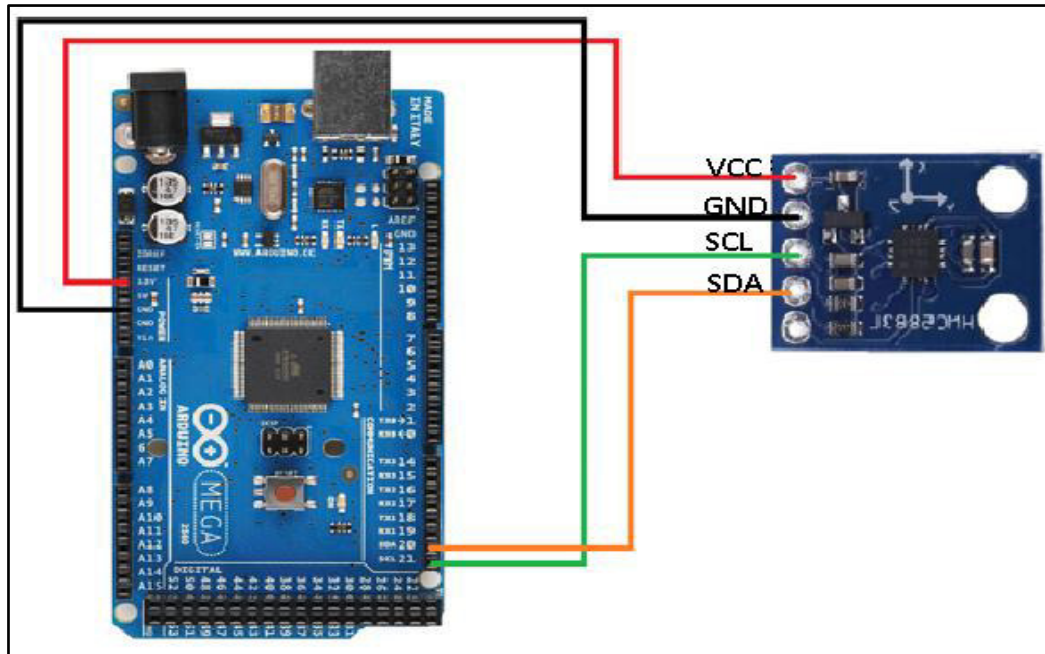


Figura 3.12 Conexión entre el microcontrolador y el sensor magnetómetro

Conexiones del microcontrolador con los sensores magnéticos

Para las conexiones de los sensores magnéticos se utiliza adicionalmente 4 resistencias de 1 (Kohm), las cuales sirven para realizar configuraciones de tipo *pull down*. Además, se utilizarán 4 pines digitales para comprobar el estado de los sensores (Pines: 4, 5, 6, 7). En la Figura 3.13 se puede observar las conexiones realizadas.

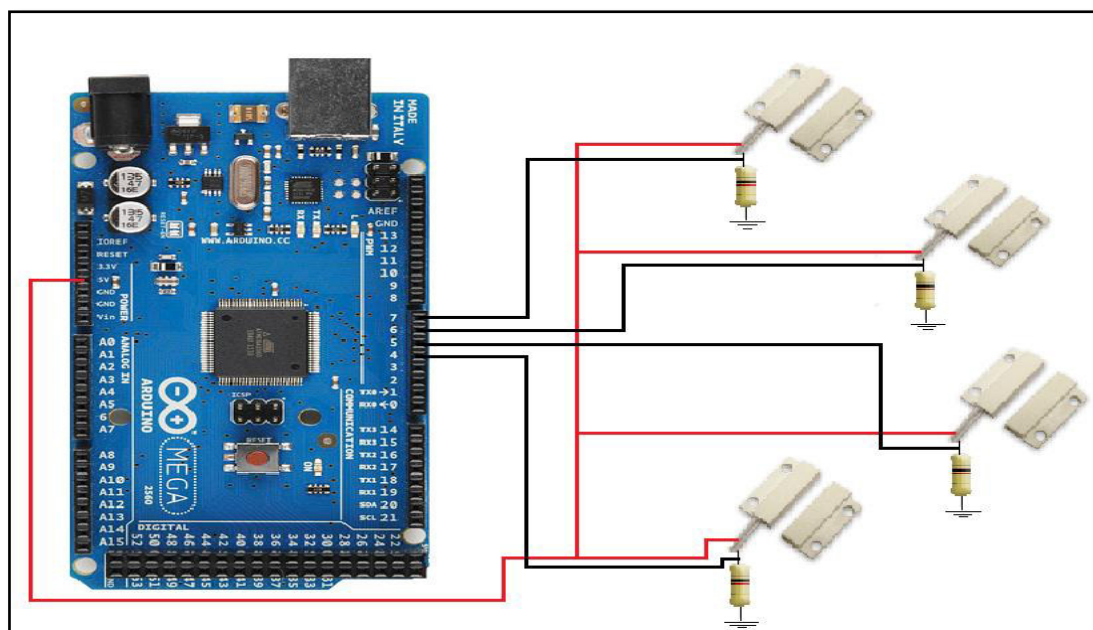


Figura 3.13 Conexiones entre el Arduino y los interruptores magnéticos

Conexiones del circuito de alimentación del prototipo

En la Figura 3.14 se puede observar las conexiones que se realiza entre el relevador y las dos fuentes de alimentación (batería principal y batería de respaldo), de tal forma que cuando la fuente principal se apague la batería de respaldo entre de inmediato y alimente el circuito.

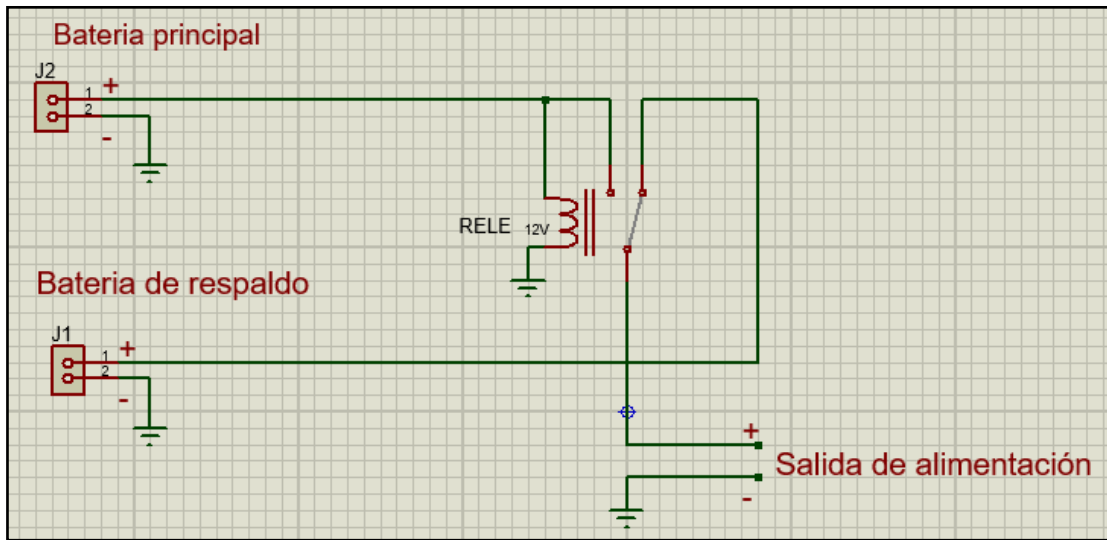


Figura 3.14 Circuito de conmutación de las fuentes de alimentación

El diagrama que se muestra en la Figura 3.15 corresponde a las conexiones entre las baterías de respaldo y el módulo de carga. Las conexiones se han hecho de tal modo para obtener un voltaje de 7.4 (V) y una corriente de 6 000 (mAh).

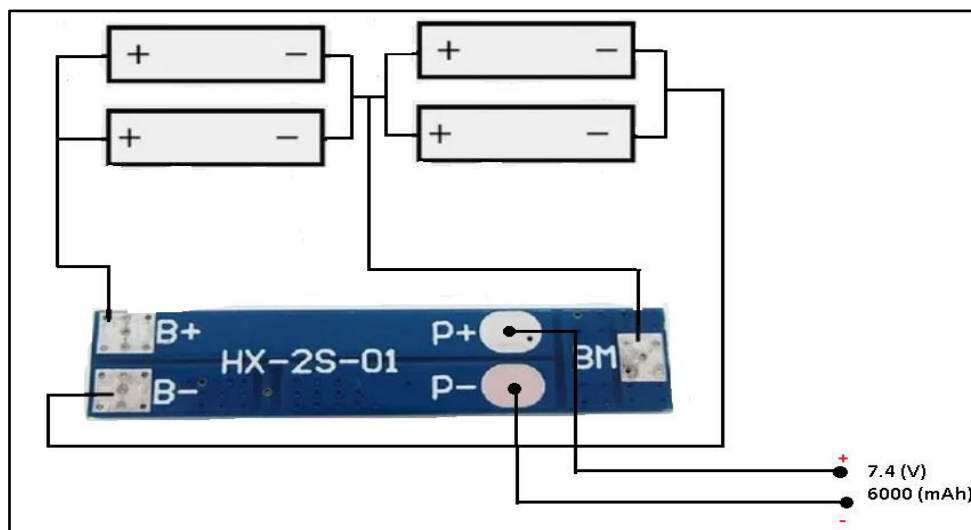


Figura 3.15 Conexiones entre módulo de carga y las baterías de litio

Diseño de la programación

En las siguientes secciones se explica el funcionamiento de la programación a través de los diagramas de flujo. Además, se mencionan los aspectos importantes que se tuvieron en cuenta para el desarrollo del código y su buen funcionamiento.

El diagrama de flujo de la Figura 3.16 muestra de forma general el proceso de funcionamiento que sigue la programación. El proceso comienza con las configuraciones iniciales de cada elemento que conforma el sistema: Arduino, módulos GPS y GSM. Una vez con los dispositivos inicializados, el prototipo espera la llegada del primer mensaje, el cual señala la acción que se debe ejecutar.

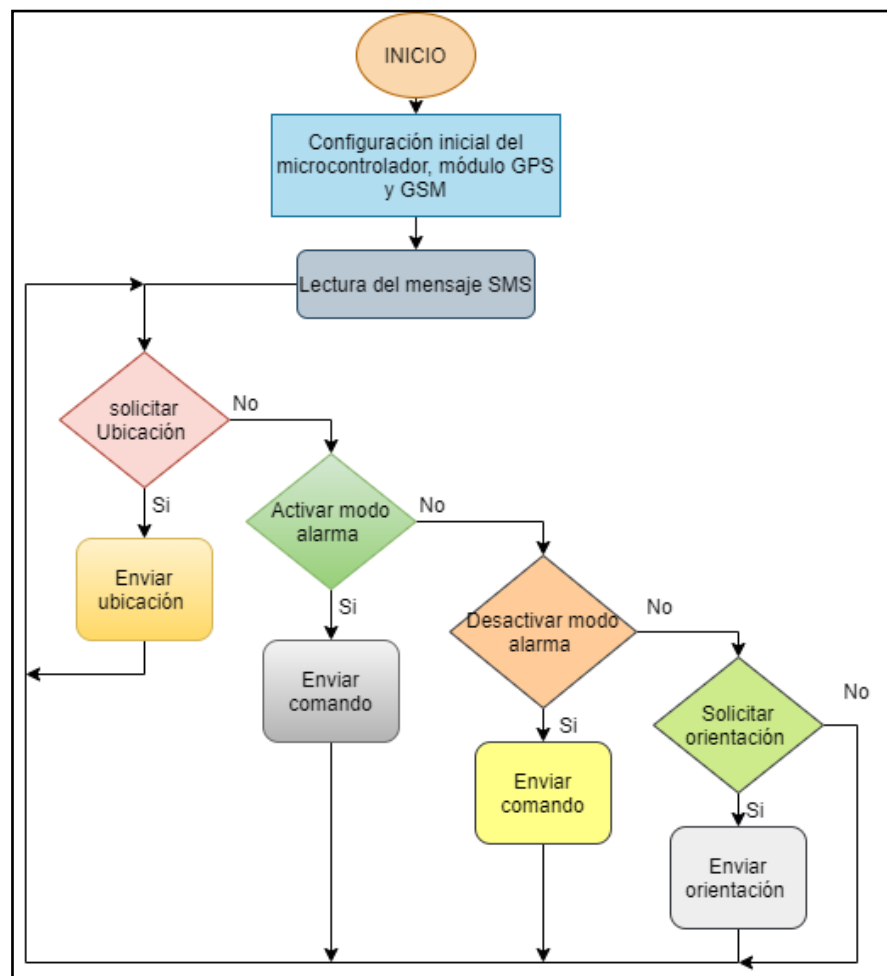


Figura 3.16 Diagrama de flujo general

Configuraciones iniciales

En primera instancia se procede con las configuraciones iniciales de los siguientes elementos: Arduino, módulo GSM y GPS. En las configuraciones iniciales se realizaron

las siguientes tareas: se procede con la activación de los puertos seriales, se configuran los pines respectivos como entradas y salidas según corresponda el caso, se establecen las velocidades de transmisión de los puertos seriales y por último se configura el módulo GSM mediante los comandos AT. A continuación, se explica a más detalle las tareas realizadas.

Configuración de los puertos seriales de los módulos GPS y GSM

Para configurar los módulos se establecen las velocidades de transmisión de los puertos seriales en donde están conectados cada uno de los dispositivos. Los módulos son capaces de trabajar a distintas velocidades, los valores que se pueden elegir son: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 o 115200 baudios. El valor que se selecciona entre todos es de 9600 baudios, ya que es la velocidad de transmisión con el que trabaja el puerto USB de una computadora es de 9600 baudios, lo que permite visualizar los datos en el monitor serial del ID del Arduino. Para colocar los valores se utiliza la función que se muestra en la Figura 3.17.

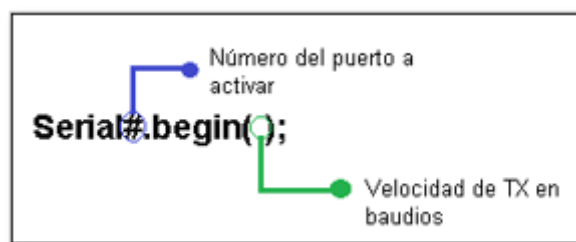


Figura 3.17 Función Serial.begin().

La función mostrada permite abrir el puerto serial para que pueda iniciarse la comunicación. Considerando que la placa Arduino Mega tiene más de un puerto serial es necesario especificar en la función el número del puerto que se desea activar, en este caso se utiliza el puerto serial uno para el SIM900, el dos para el GPS y el cero para visualizar los datos de los procesos. El valor a configurar se coloca dentro del paréntesis de la función. En la Tabla 3.9 se observa la asignación de los puertos con sus velocidades de transmisión asignadas.

Tabla 3.9 Asignación de puertos

Dispositivo	Función
Módulo GSM	Serial1.begin(9600);
Módulo GPS	Serial2.begin(9600);
Monitor Serial	Serial.begin(9600);

Configuraciones iniciales del módulo GSM

Una vez que los puertos seriales hayan sido activados se procede con la configuración del SIM900, para lo cual se utiliza los comandos AT especificados por el fabricante de la placa, los comandos son enviados a través del puerto serial número uno que corresponde al módulo GSM. A continuación, se detallan los comandos usados para la configuración del mismo.

El primer comando que se utiliza es para configurar la velocidad de transmisión del módulo GSM, para lo cual se procede a enviar por el puerto serial el comando que se muestra en la Figura 3.18.



Figura 3.18 Comando AT+IPR

El segundo comando que se utiliza es para indicar al módulo GSM cómo se van a recibir y enviar los mensajes, ya que existen dos opciones como texto o PDU. La seleccionada fue en modo texto, para lo cual se envía por el puerto serial el comando que se muestra en la Figura 3.19.

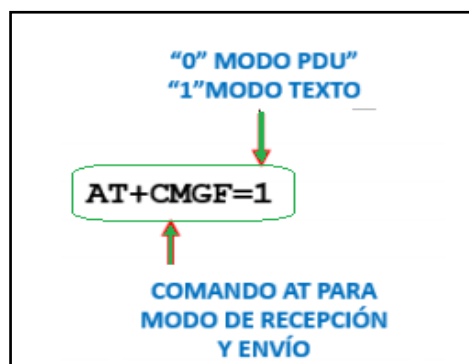


Figura 3.19 Comando AT+CMGF

Ahora usando el comando que se muestra en la Figura 3.20 se indica la manera en que el dispositivo lee el mensaje. En el comando se observa que se incluye al final un signo de interrogación el cual señala que el módulo leerá el último mensaje recibido.



Figura 3.20 Comando AT+CMGR

Finalmente se activa la opción de que los mensajes que se reciban vayan directamente al puerto serial del dispositivo, para que puedan ser leídos y guardados en las variables definidas y tomar las decisiones en base a ello. Para activar esta opción se envía el comando que se muestra en la Figura 3.21.



Figura 3.21 Comando AT+CNMI

Configuración de los puertos de entrada y salida del microcontrolador

En este bloque se configuran los pines digitales del microcontrolador, los cuales son usados para diversas tareas, tales como: alimentar al módulo GPS, también para encender el SIM900 mediante un pulso en estado alto, y para las lecturas de los sensores de alarma. La función utilizada para la configuración de los puertos se muestra en la Figura 3.22.

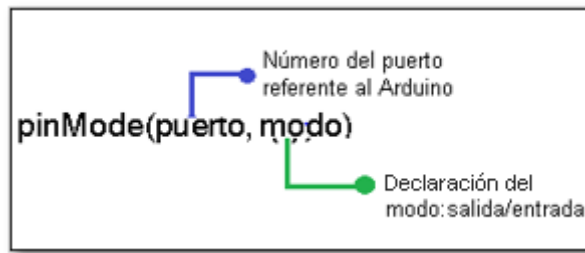


Figura 3.22 Función para configurar los puertos

En la función se debe especificar el número del puerto que se va a configurar y también si va a ser una entrada o una salida. Para configurar como salida se escribe “*OUTPUT*” y para configurar como entrada se escribe “*INPUT*”. El pin que alimentará el módulo GPS se configura como salida al igual que el pin que controla el encendido del SIM900. Los únicos pines que se configuran como entradas serán los que leerán el estado de los interruptores magnéticos.

Para el encendido del módulo GSM se utiliza el puerto digital 2 del Arduino y para la alimentación del GPS el puerto digital 8, se han definido algunas variables para hacer referencia a los puertos mencionados, las cuales se observan en la Tabla 3.10.

Tabla 3.10 Configuración de puertos de salida

Variables de referencia	Configuración de puertos digitales
int VccGPS = 8;	pinMode (OnGSM, OUTPUT);
int OnGSM = 2;	pinMode (VccGPS, OUTPUT);

Para las lecturas de los interruptores magnéticos se usan los puertos 4, 5, 6, 7, también se han definido algunas variables para referenciar los puertos, las cuales se observan en la Tabla 3.11.

Tabla 3.11 Configuración de puertos de entrada

Variables de referencia	Configuración de puertos digitales
int lecturaP1 = 4;	pinMode (LecturaP1, INPUT);
int LecturaP2 = 5;	pinMode (LecturaP2, INPUT);
int LecturaP3 = 6;	pinMode (LecturaP3, INPUT);
int LecturaP4 = 7;	pinMode (LecturaP4, INPUT);

Lectura y almacenamiento SMS

Luego de haber realizado todas configuraciones iniciales, el prototipo está listo para recibir el primer mensaje. En la programación se creó una función llamada “LeerMensaje”, cuyo propósito es detectar el arribo de un mensaje de texto, para posteriormente leerlo y guardarlo en una variable específica. El proceso que realiza la función se muestra en la Figura 3.23.

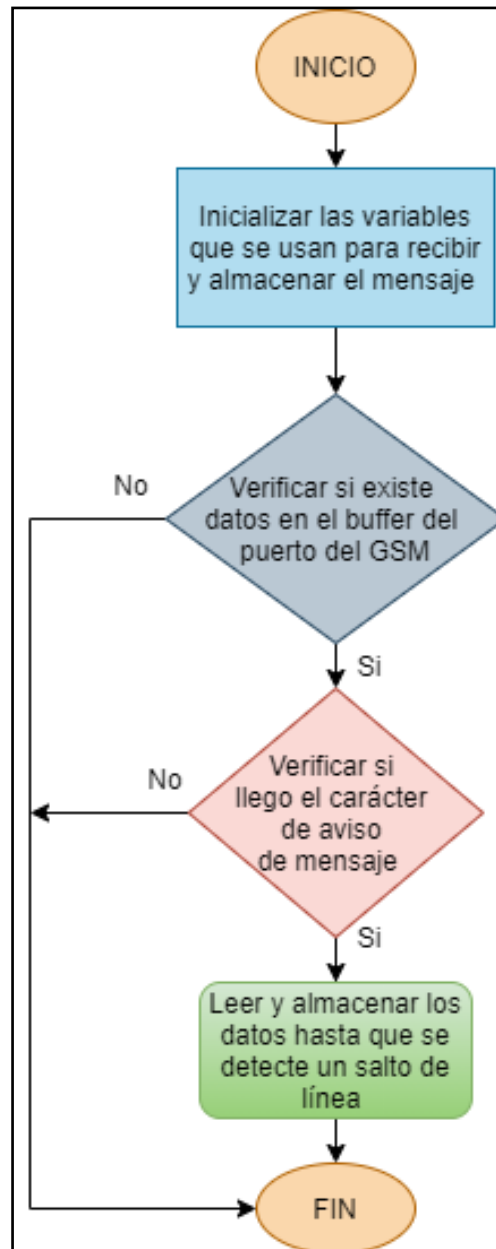


Figura 3.23 Diagrama de flujo de lectura de un SMS

Entre las variables que se definieron para recibir el mensaje están dos, la primera de ellas es de tipo *char*, cuya función es detectar el carácter “@” que indica el inicio de un

mensaje. La segunda variable es una matriz de tipo *char* que se usa para almacenar los caracteres del mensaje recibido.

El proceso inicia con la revisión del *buffer* del puerto serial uno donde está conectado el módulo GSM, para hacer esta revisión se utiliza la función que se muestra a continuación.

If (Serial1.available(>0));

La función toma un valor de cero cuando no hay datos en el *buffer* del puerto serial y toma un valor diferente de cero cuando sí lo hay; es decir, cuando llega un mensaje. Al momento que llega el mensaje se procede con la lectura del mismo, para lo cual se usa la siguiente función.

“serail1.read();”

La función procede con la lectura del mensaje hasta encontrar el carácter “@”, en caso de no encontrar el carácter significa que el mensaje no provino de la interfaz y por tanto es desechado. Cuando el carácter es detectado se procede a guardar los siguientes datos después del “@”, ya que corresponden al mensaje enviado desde la aplicación móvil, se usa la matriz *char* que se creó previamente para guardar los caracteres del mensaje y así termina el proceso de lectura y guardado del mensaje.

Elección del proceso a realizar

El diagrama de flujo de la Figura 3.24 indica el proceso que sigue el programa una vez se recibe el mensaje. En la estructura de la programación, se crea una función que realiza este proceso y se llama “SeleccionTarea”.

Cuando el mensaje es guardado en la matriz, se procede a compararlo con la información establecida en la programación y así el programa elige ejecutar alguna de las acciones mostradas en el diagrama de flujo de la Figura 3.24.

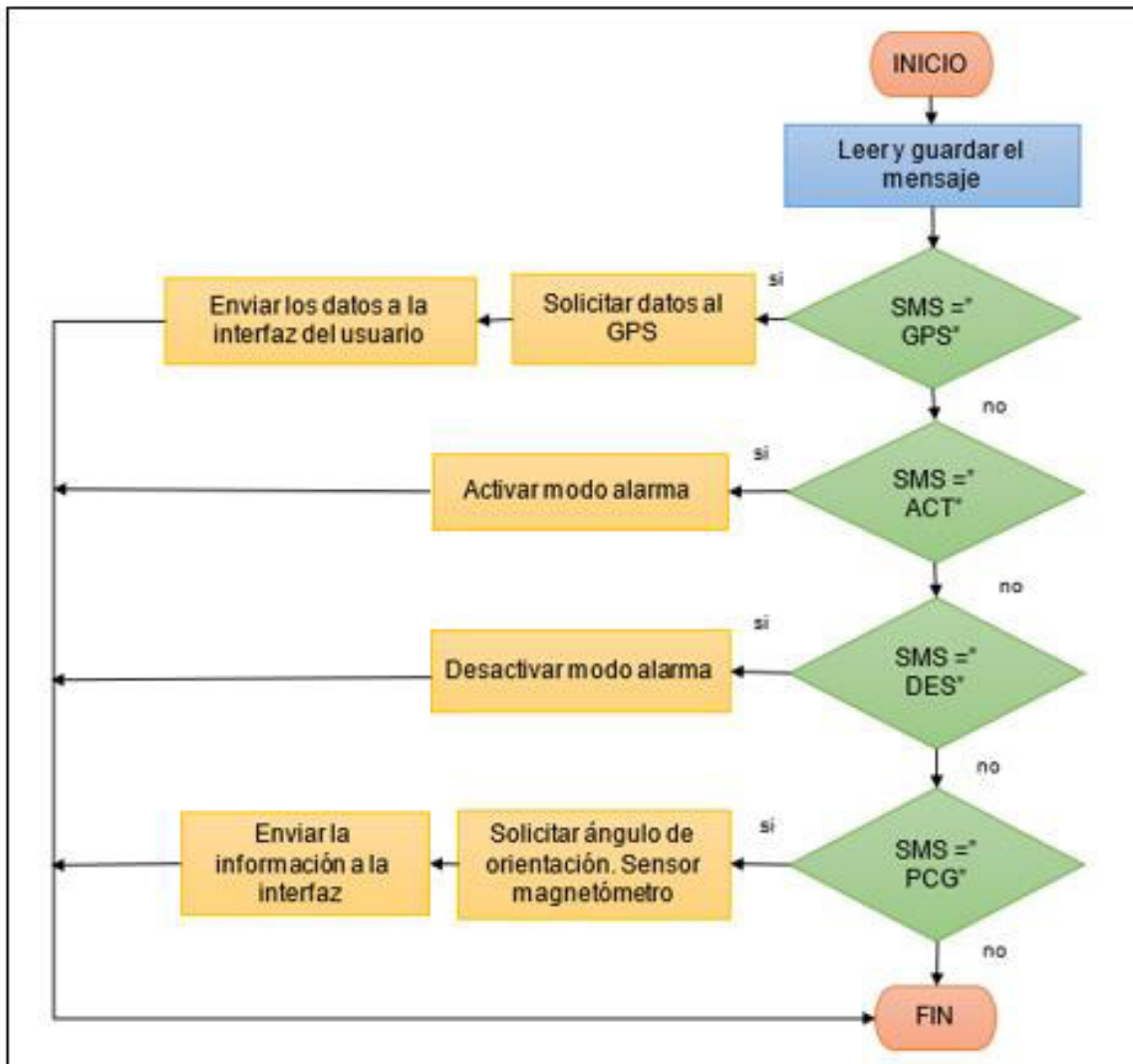


Figura 3.24 Selección del proceso a realizar

Solicitar ubicación al módulo GPS.

En el diagrama de flujo de la Figura 3.25 se muestra el proceso que se realiza cuando el usuario solicita la ubicación de su vehículo. Cabe recalcar que para este proceso se lo realiza con la ayuda de la librería “TinyGPS”, la cual permite extraer los datos del GPS de forma más fácil sin emplear tantas líneas de programación.

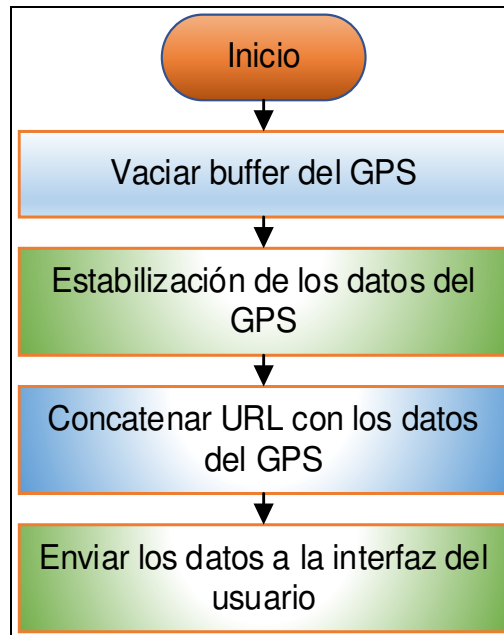


Figura 3.25 Diagrama de flujo para solicitar ubicación

Vaciar *buffer* del GPS

El proceso inicia vaciando los datos del *buffer* del puerto GPS, ya que, una vez encendido el prototipo, el módulo GPS empezará a llenar de información al *buffer*, sin que se lo solicite, la capacidad máxima del *buffer* es de 64 bytes, una vez que se llena de información los demás datos que vayan a llegar no podrán ser almacenados y serán desechados.

El problema que se presenta al no vaciar el *buffer* es que al momento de que el usuario solicite los datos del GPS se podría estar enviando la información de una ubicación pasada y no la que se encuentra actualmente el prototipo, por esta razón el proceso del envío de la ubicación empieza con este bloque, para así tener los datos de la ubicación actualizados dentro del *buffer* para después leerlos en los siguientes bloques. Para realizar el vaciado del *buffer* se utiliza función que se muestra en la Figura 3.26.

```

while (Serial2.available()>0) {
  Serial2.read();}
  
```

Figura 3.26 Función para borrar el *buffer* del puerto GPS

La función mostrada va leyendo y desechando los datos del *buffer*, una vez vaciado se procede con el siguiente bloque.

Estabilización de datos del GPS

Al momento que el módulo GPS empieza a enviar su información al Arduino a través del puerto serial, las primeras lecturas suelen tener valores de cero grados hasta que el módulo GPS se estabilice. Para dejar pasar este tiempo, en el cual se envía datos erróneos, se crea una función dentro de la programación llamada “estabilizacionDatosGPS” (ver Figura 3.27), la cual deja pasar las primeras lecturas que tienen valores de cero grados.

```
void EstabilizacionDatosGPS () {  
  for(int contador=0; contador<50; contador++){  
    lecturasGPS();} //función que lee el GPS  
  int contador=0;  
}
```

Figura 3.27 Función para estabilizar los datos del GPS

El funcionamiento es simple, se crea una variable llamada “contador” la cual va incrementando su valor en una unidad por cada lectura del GPS que se realiza, de esta forma se deja pasar dichas lecturas por un tiempo corto hasta que la variable llega a un valor determinado en este caso “50” y hasta entonces ya se estabiliza los datos del GPS.

Concatenar datos y enviar al usuario

Los datos del GPS a enviar tienen que estar acompañados del URL de Google Maps, para que en el lado de la aplicación se pueda visualizar de forma inmediata, sin la necesidad de que el usuario escriba las coordenadas de forma manual.

Para el envío de los datos concatenados a la interfaz del usuario, se ha creado una función dentro de la programación llamada “EnviarCordenadas”. En la función primero se le indica al módulo SIM900 el número celular al cual se transmite el mensaje, para lo cual se envía el siguiente comando al puerto serial del módulo de comunicación:

AT + CMGS = \"numeroCelular\"

Cabe señalar que para enviar el comando al puerto serial del módulo de comunicación y este lo procese, se utiliza el comando “Serial1.print()”, dentro del paréntesis se coloca la información que se desea enviar al módulo. Una vez enviado el comando especificado, el módulo SIM900 está listo para enviar el mensaje, así que se procede a indicar cuál es el mensaje a enviar.

Primero se envía el URL de Google Maps al puerto del módulo GPS, luego los datos de la latitud y longitud del GPS, para obtener dichos valores se utilizan los comandos que se muestra a continuación:

gps.location.lat() y **gps.location.lng()**

Dichos comandos son propios de la librería TinyGPS, basta con escribirlos en la programación para que retorne los valores de la latitud y longitud del GPS.

Una vez cargado el mensaje en el puerto serial del módulo de comunicación, se procede a enviarlo a la aplicación. Para lo cual se envía el carácter “char 26” al SIM900 y automáticamente el mensaje es enviado. En la Figura 3.28 se observa todo el proceso mencionado.

```
void EnviarCordenadas () {
Serial1.println("AT + CMGS = \"0983113518\"); //Número de teléfono de destino.
delay(200);
Serial1.print("https://maps.google.com/maps?q="); //se envia el url de google maps al modulo GSM
Serial1.print(gps.location.lat(),6); //Obtemos los datos de latitud
Serial1.print(","); // separador
Serial1.print(gps.location.lng(),6); //Obtemos los datos de longitud
delay(200);
Serial1.println((char)26); //para enviar el mensaje
```

Figura 3.28 Función para enviar coordenadas a la interfaz

Solicitud de datos de orientación

El diagrama de flujo de la Figura 3.29 muestra el proceso de lectura y envío de los datos de la orientación del vehículo a la aplicación móvil.

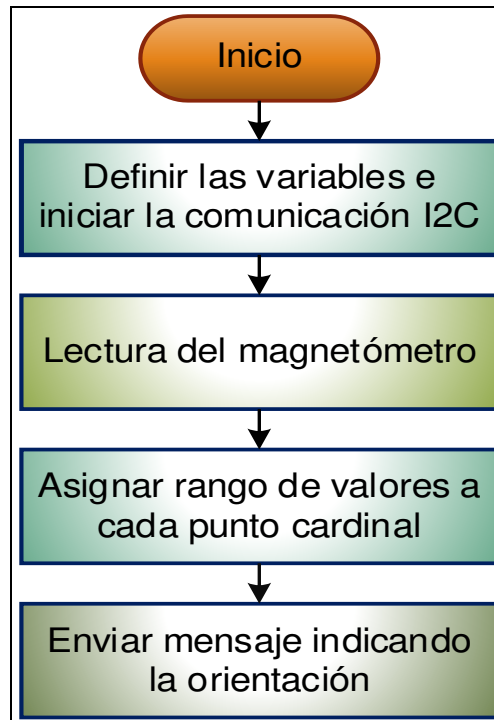


Figura 3.29 Diagrama de flujo de la orientación del vehículo

Definir las variables e iniciar la comunicación I2C

Para manejar la comunicación I2C es necesario incluir la librería “Wire” y de igual forma se necesita una segunda librería para manejar correctamente el sensor magnetómetro, cabe señalar que todas las líneas de la programación están comentadas y se las podrá revisar en la sección de anexos. Una vez colocadas las librerías se procede con la activación del puerto I2C, para ello se usa la función “Wire.begin();”. En la estructura de la programación se crea una función con el nombre de “LecturaMagnetometro”, en dicha función se empieza definiendo las variables, cuyos nombres deben ser los que demanda la librería del sensor y no se las podrá modificar, ya que podría ocasionar un mal funcionamiento. Hasta el momento se han definido las variables y activado la comunicación I2C.

Lectura del magnetómetro

Una vez declaradas las variables y activada la comunicación I2C, se comienza con la lectura del sensor magnetómetro, la librería usada realiza todas las operaciones matemáticas con ayuda del microcontrolador incorporado en el sensor, es decir para ver el valor del ángulo del *azimut* geográfico solo se imprime la variable que contiene el

valor en el monitor serial, ahí se podrán observar los valores que va generando el sensor de acuerdo con la dirección que apunte.

Asignar rango de valores a cada punto cardinal

En la programación la variable que muestra el valor del ángulo geográfico de la Tierra se llama “azimut” esta variable puede tomar valores que van desde los 0 (grados) hasta los 360 (grados) según la dirección que esté apuntando el sensor magnetómetro, considerando lo dicho se establece un rango de valores a cada punto cardinal, tal como se muestra en la Tabla 3.12.

Tabla 3.12 Asignación de valores a los puntos cardinales

Punto cardinal	Valores asignados a los puntos cardinales
Norte	0 a 22.5 y 337.7 a 360 (grados)
Sur	157.5 a 202.5 (grados)
Este	67.5 a 112.5 (grados)
Oeste	247.5 a 292,5 (grados)
Puntos laterales	Valores asignados a los puntos laterales
Noroeste	292.5 a 337.5 (grados)
Noreste	22.5 a 67.5 (grados)
Suroeste	202.5 a 247.5 (grados)
Sureste	112.5 a 157.5 (grados)

Enviar mensaje indicando la orientación

Para enviar el mensaje se necesita hacer uso de una variable de tipo “String” la cual se usa para almacenar la frase que se envía al usuario.

Dependiendo del ángulo geográfico generado por el sensor, se envía una frase diferente, por ejemplo, si el valor de la variable “Azimut” es de cero grados, entonces en la variable “String” almacena la siguiente oración “El vehículo está apuntando en dirección: norte” y así sucede con los demás valores de acuerdo con la Tabla 3.12

Una vez guardada la frase en la variable “String” se la envía al puerto serial del módulo de comunicación para que este a la vez la envíe a la aplicación del usuario.

Activar modo alarma

En el diagrama de flujo de la Figura 3.30 se muestra el proceso que se realiza al activarse el modo alarma.

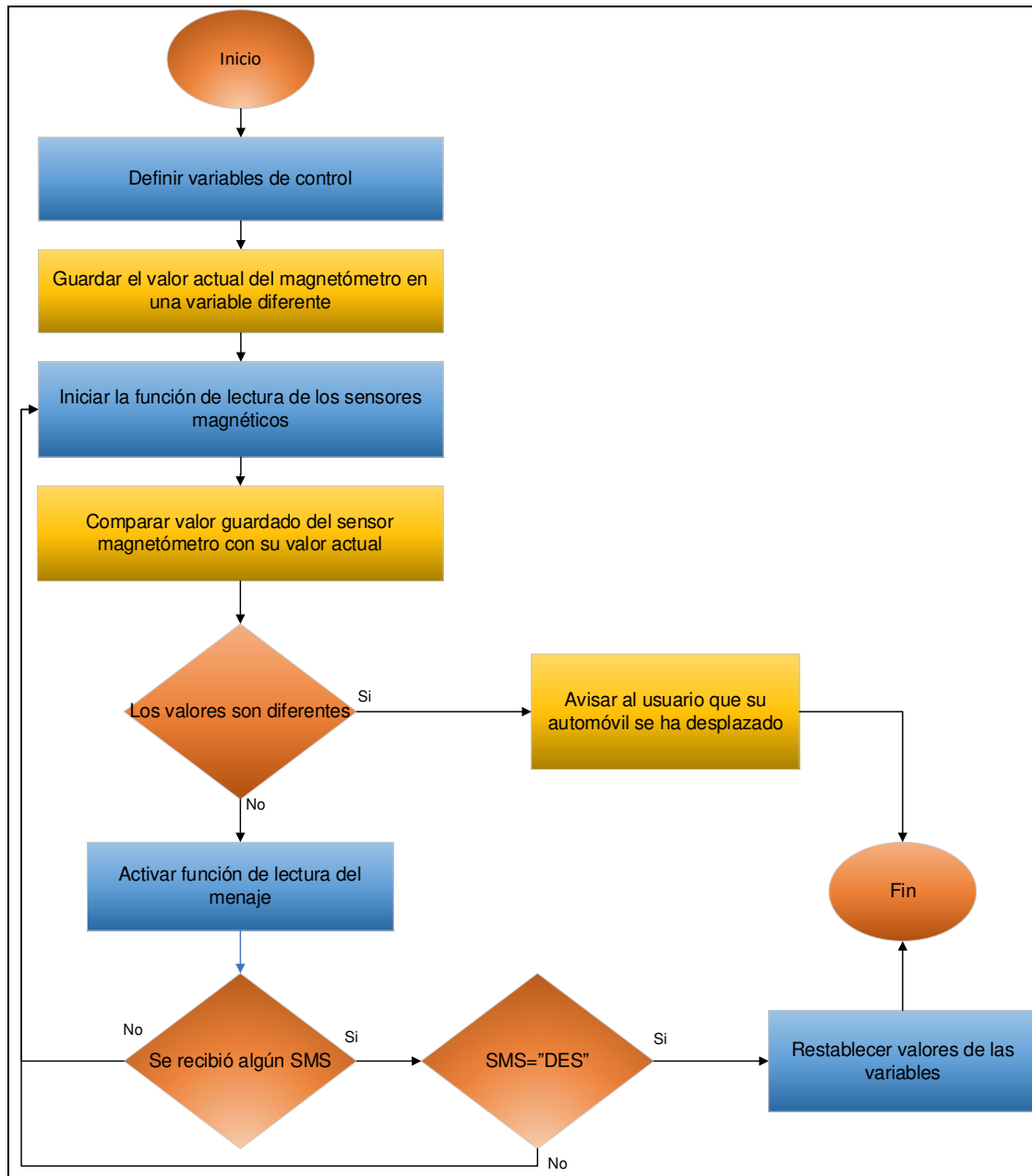


Figura 3.30 Diagrama de flujo activación de modo alarma

El proceso empieza definiendo dos variables la primera de ellas es de tipo “String” y se la usa para guardar el valor del sensor magnetómetro, que indica la orientación del vehículo. La segunda variable es de tipo entera y se la usa para entrar a un ciclo *while*, ya que al momento que el usuario solicite activar el modo alarma, esta variable tomará el valor de “1” y mientras se mantenga en este valor no saldrá del ciclo *while*, en este

ciclo se estarán ejecutando varias tareas constantemente, la única forma de salir del ciclo *while*, es que el usuario solicite desactivar el modo alarma, cuando lo haga el valor de la variable tipo entera cambiará a “0” y por lo tanto el ciclo *while* terminará.

Entre las tareas que se ejecutan en el ciclo *while* se tienen tres, la primera de ellas es una función, que en la estructura de la programación se lo ha denominado como “Sistema Alarma”. Esta función se encarga de detectar cuando las puertas del vehículo son abiertas. En la Figura 3.31 se puede observar el proceso que realiza esta función en particular.

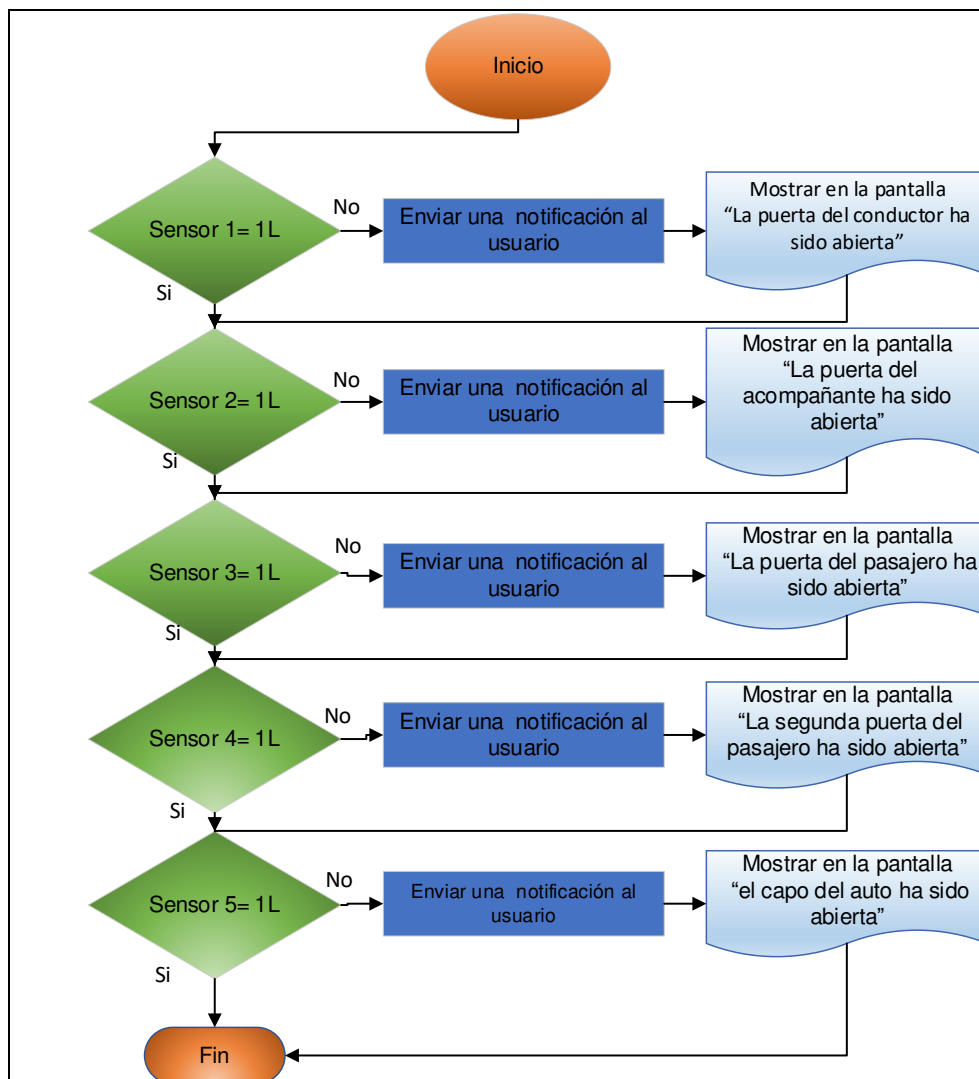


Figura 3.31 Diagrama de flujo de lectura de los sensores magnéticos

En el diagrama de flujo se observa que se realiza un barrido de los interruptores magnéticos, para así ir verificando el estado de cada uno de ellos. El estado normal de cada interruptor es cerrado con un valor alto; es decir, 5 (V), y cuando estos se abren cambian su estado a bajo con 0 (V). En caso de que uno de ellos llegara a abrirse, se

avisará inmediatamente al usuario señalando qué puerta fue abierta. Para enviar el mensaje se necesita declarar una variable de tipo "String" que almacena una frase a enviarse según el caso que corresponda, dicha variable se ha definido en la programación con el nombre de "EstadoPuerta".

La segunda tarea que se ejecuta constantemente en el ciclo *while* es la acción de comparar dos variables con el propósito de verificar si el auto se ha desplazado. La primera variable contiene la orientación del vehículo que se guarda justo en el momento de activar el modo alarma y la segunda variable contiene los valores que constantemente se están generando en el magnetómetro. Una vez que estos dos valores sean diferentes, la función realiza una llamada para alertar al usuario y posteriormente se envía una notificación indicando que su auto está en movimiento.

La tercera y última tarea que se ejecuta en el ciclo *while* es la comprobación de llegada del mensaje que desactiva el modo alarma, en caso de que no se haya detectado ningún mensaje la función *while* se repite constantemente.

Implementación del prototipo

Finalizado el proceso de diseño y programación, se procede a la implementación del prototipo. Los dispositivos electrónicos que se han seleccionado se han colocado sobre una placa de baquelita perforada, ver Figura 3.32. Una vez acoplados los componentes se dio paso a las conexiones respectivas, tal como se muestra en la Figura 3.33.

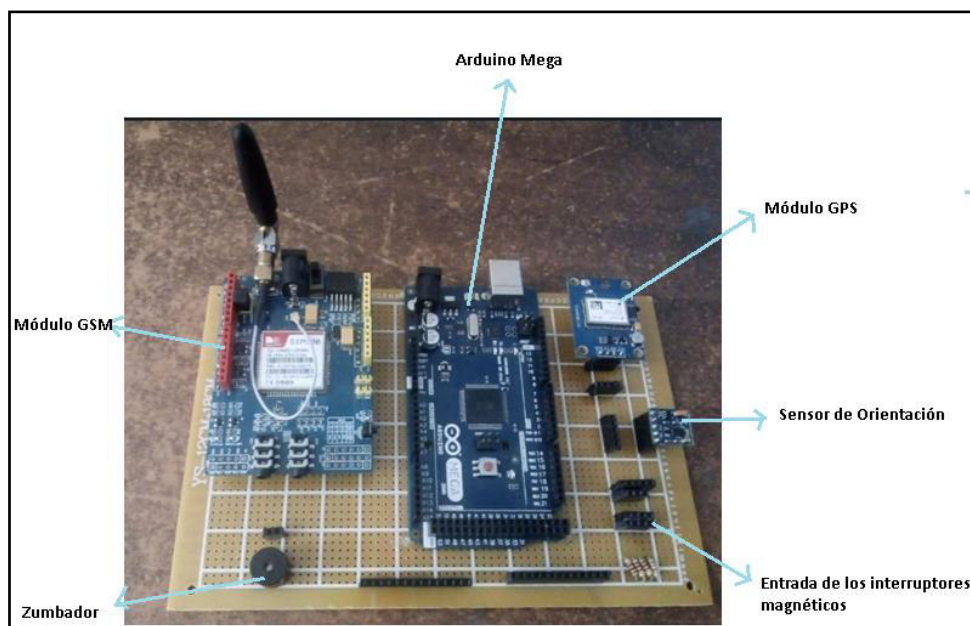


Figura 3.32 Montaje de los elementos electrónicos

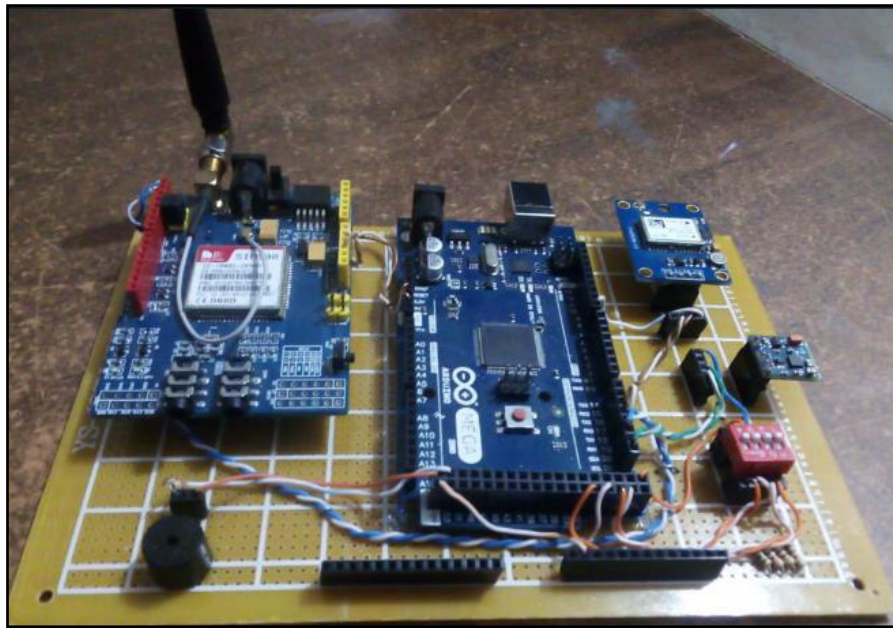


Figura 3.33 Conexiones de los componentes electrónicos del prototipo

Con la ayuda de plantillas se procedió a construir la maqueta, la cual representa al vehículo. En la Figura 3.34 se puede observar el proceso de fabricación.



Figura 3.34 Construcción de la maqueta del vehículo

Una vez construida la maqueta se procede a colocar los interruptores magnéticos en las puertas del vehículo (ver Figura 3.35).

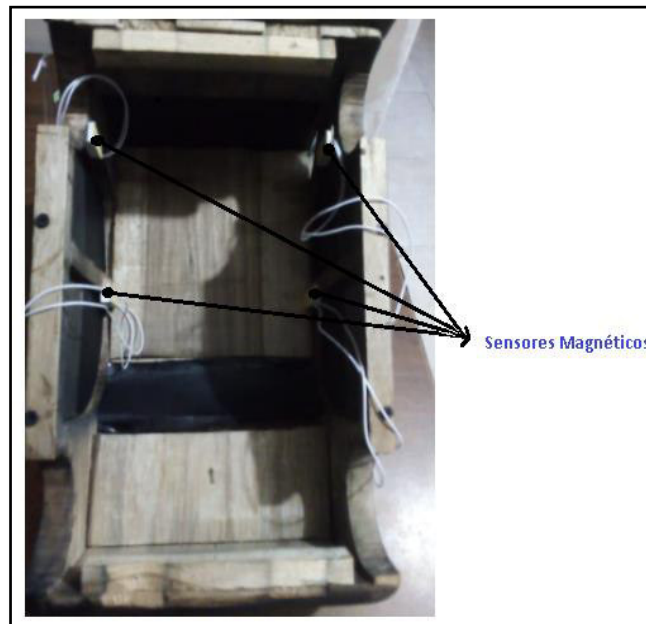


Figura 3.35 Colocación de los sensores magnéticos en las puertas del vehículo

En la Figura 3.36 se muestra el circuito de conmutación de las fuentes de alimentación, el cual consta de tres borneras, una para conectar la fuente principal, otra para conectar la fuente de baterías de respaldo y la última que es la salida para alimentar el prototipo.

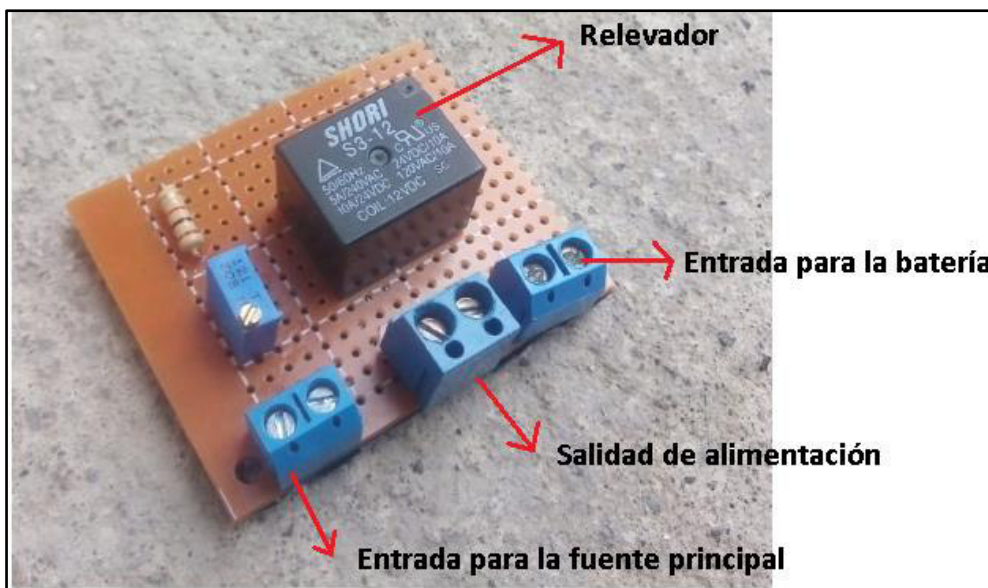


Figura 3.36 Armado del circuito de conmutación de baterías

En la Figura 3.37 se muestra las baterías de litio que se usan, las cuales están sobre una carcasa que facilita las conexiones con el módulo de carga.

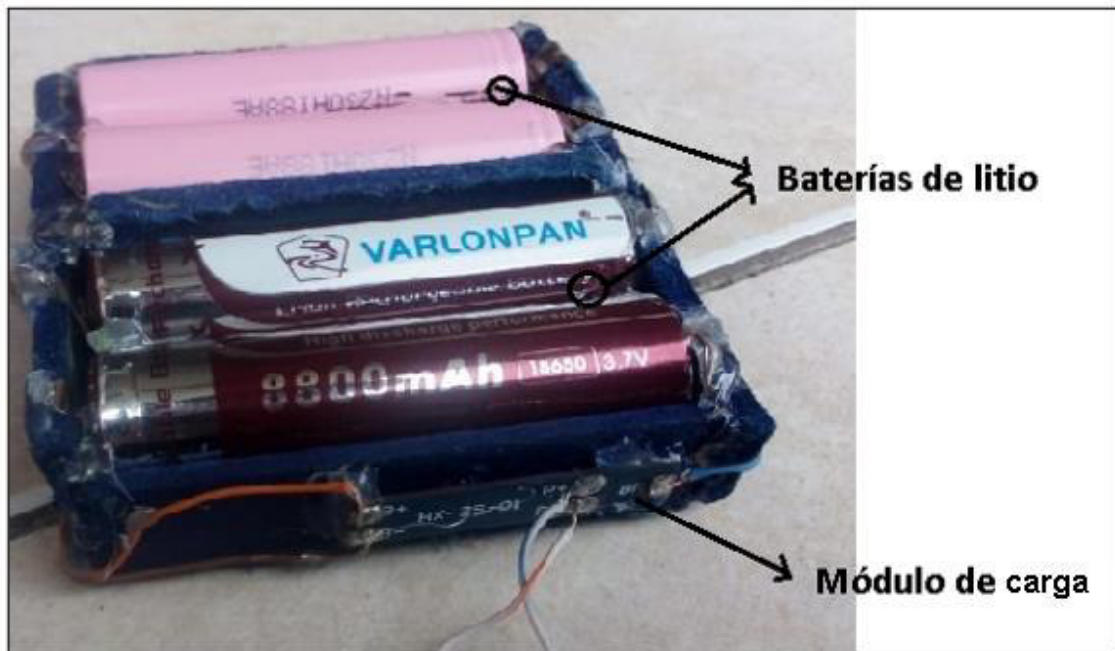


Figura 3.37 Armado del circuito de carga de las baterías

En la Figura 3.38 se observa la implementación final del prototipo, la cual ha sido montada sobre la base de la maqueta que representa al vehículo.



Figura 3.38 Implementación final del prototipo

3.4 Desarrollo de la aplicación móvil

El diagrama de flujo general de la aplicación se observa en la Figura 3.39. La aplicación cuenta con un total de cinco pantallas, las cuales son: Pantalla de inicio, pantalla principal, pantalla del mapa, pantalla de orientación y pantalla de alarma. Cada una cumple una función específica dentro de la aplicación.

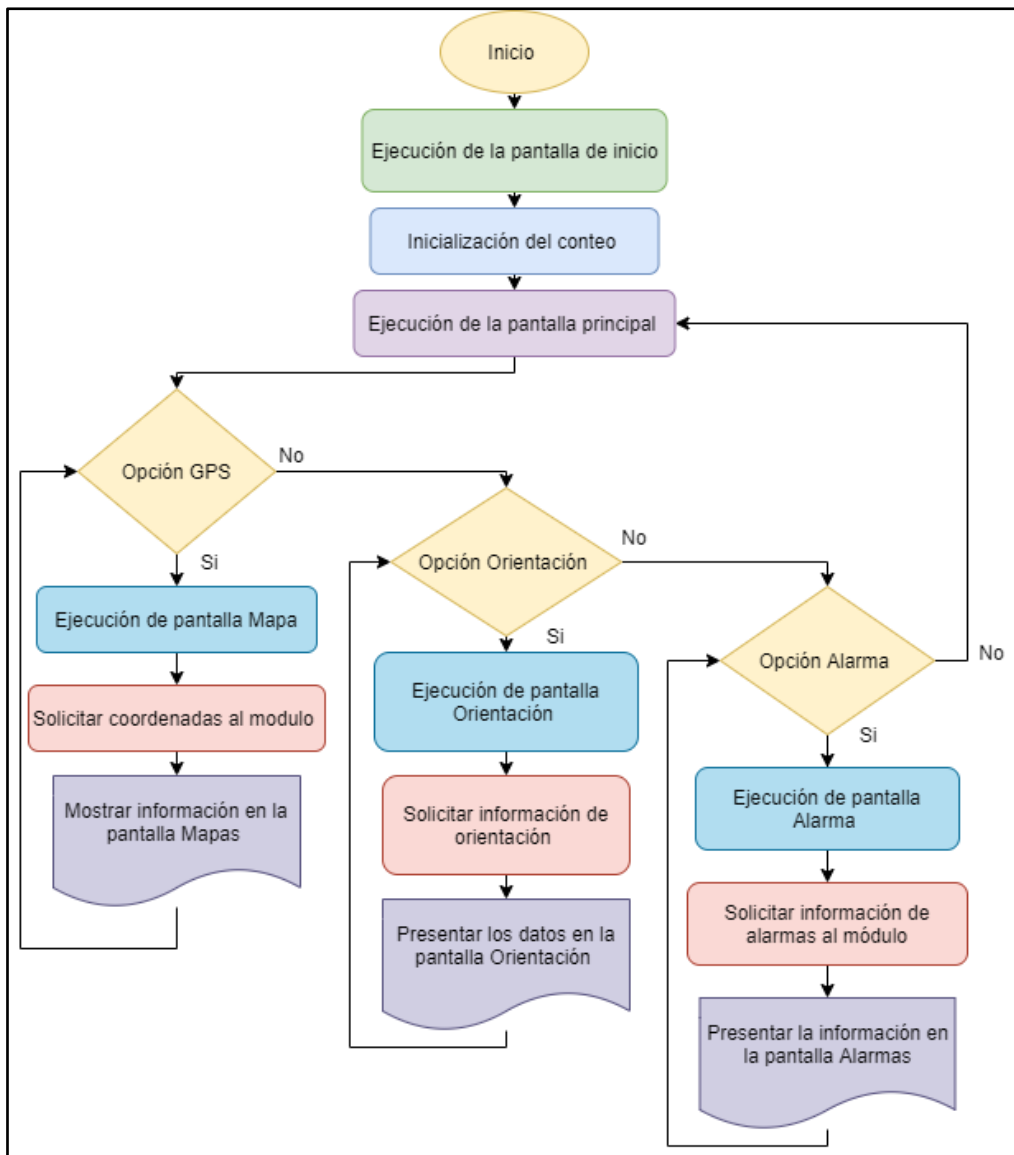


Figura 3.39 Diagrama de flujo general de la aplicación

Pantalla de inicio

Esta pantalla tiene como función la de brindar al usuario una ventana en donde se visualiza un entorno previo a las funciones de control. En esta pantalla lo que se busca es que se observe una barra de conteo que hace alusión a que la aplicación está

inicializándose. La ventana presenta una barra donde comienza un conteo desde cero y se incrementa hasta llegar a 100, posterior a este proceso se abre la pantalla principal.

En la Figura 3.40 se muestra la forma en que se visualizará la pantalla en el celular. Se observa que en la parte superior se presenta una imagen que hace referencia a un vehículo, seguido por el nombre del proyecto “prototipo de monitoreo vehicular”, también se agregó una barra de carga acompañada con una leyenda que señala el porcentaje en la que avanza la barra.



Figura 3.40 Pantalla de inicio

El diagrama de flujo de la pantalla de inicio se la puede observar en la Figura 3.41, primero se crea una variable con el nombre de “lblnumero” con un valor inicial de cero, seguidamente se establece un temporizador que va a ir incrementando el valor de la variable y el avance de la barra, para ello se realiza una comparación si la variable “lblnumero” es inferior a 100 se procede a incrementar +1, mientras que la barra se incrementa x2, este proceso se realiza cada 100 (mseg). Si la comparación en cambio resulta que “lblnumero” es 100 se procede a desactivar el temporizador y dirigirse a la pantalla de nombre “pantalla principal”.

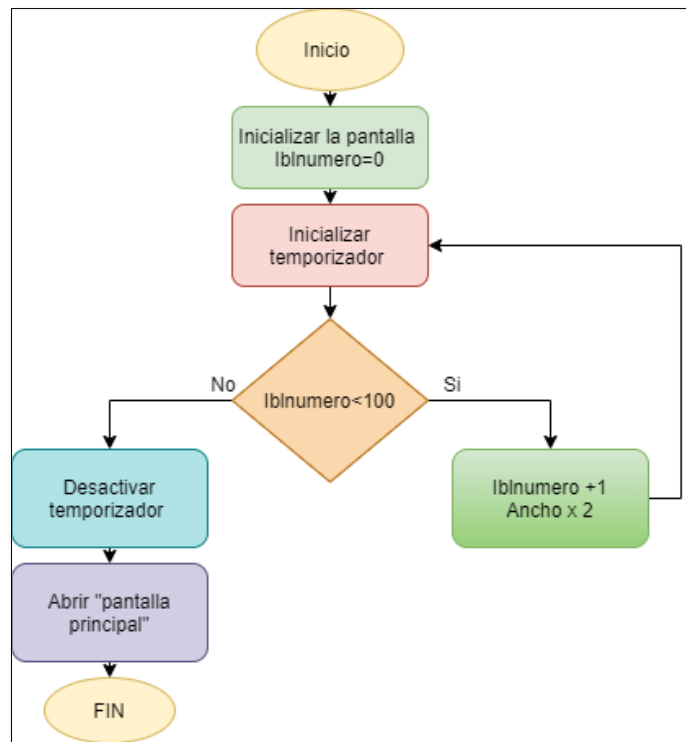


Figura 3.41 Diagrama de flujo pantalla inicio

Pantalla principal

Esta pantalla alberga los botones principales de la aplicación que son: control del GPS, control de orientación y control de alarmas. En la Figura 3.42 se visualiza la pantalla principal.



Figura 3.42 Pantalla principal

En la Figura 3.43 se presenta el diagrama de flujo de la pantalla principal, se observa que se cuenta con tres opciones y dependiendo de la elección se redirige a una tercera pantalla predeterminada, siendo así que, si al presionar el botón de control de GPS se dirige a la pantalla mapa, en cambio si se elige la opción control orientación se dirige a la pantalla orientación y como último botón se tiene al control de alarma que al ser presionado se dirige a la pantalla alarma. Si ningún botón es presionado, la aplicación permanece en ese estado hasta que el usuario realice alguna elección o cierre la aplicación.

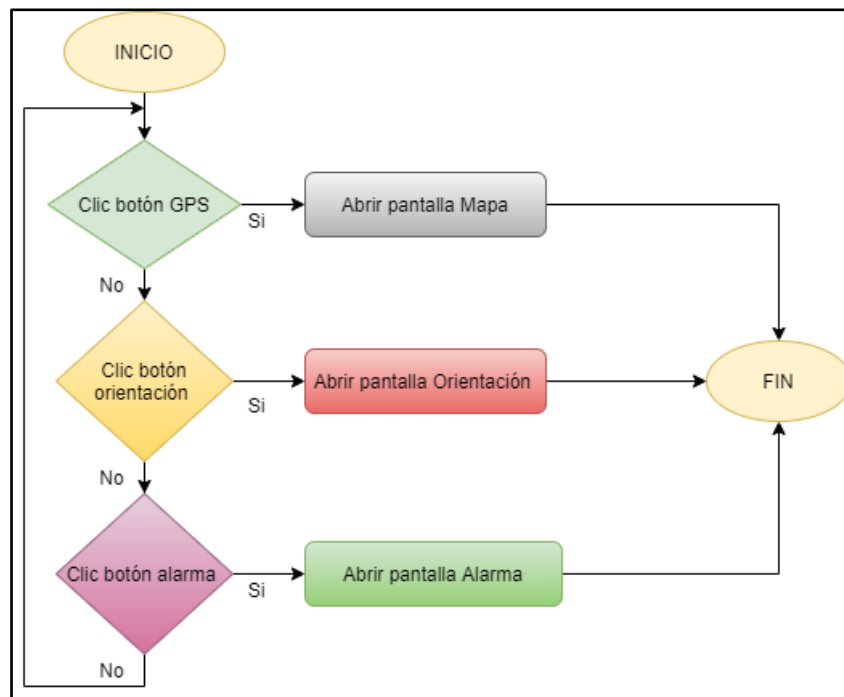


Figura 3.43 Diagrama de flujo pantalla principal

Pantalla del Mapa

En la Figura 3.44 se muestra la parte visual de esta pantalla, se observa que se colocó un título “ubicación de vehículo” y posteriormente se colocan dos botones uno para solicitar las coordenadas del vehículo y otro botón para regresar a la pantalla principal, también se sitúa un visor WEB con la finalidad de abrir el enlace que se recibirá por parte del módulo de comunicación GSM, el cual permite ver la ubicación del vehículo. Otras herramientas que se colocaron en esta pantalla y que no son visibles son la función de enviar y recibir los mensajes, así como también, la función de notificar la llegada de un mensaje.

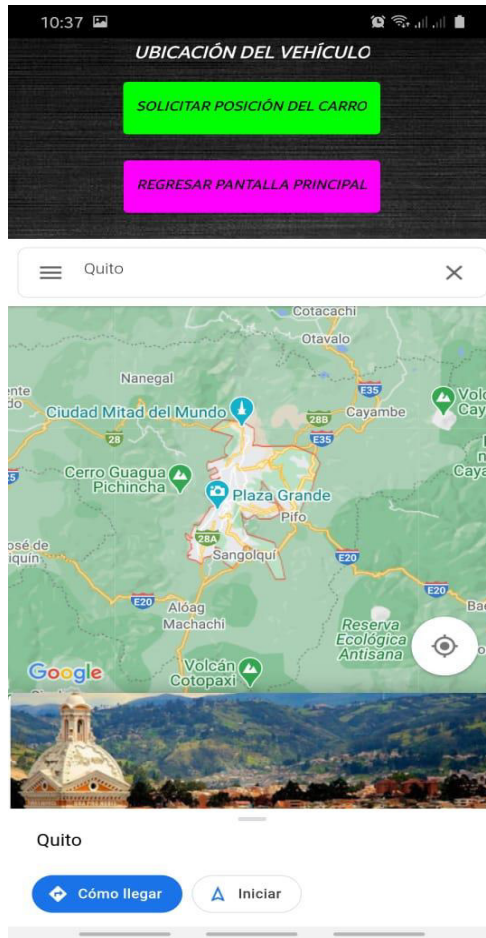


Figura 3.44 Pantalla "Mapa"

En la Figura 3.45 se muestra el diagrama de flujo de esta pantalla, para así entender el proceso que cumple.

Cuando se presiona el botón "SMSGPS" que tiene la leyenda "solicitar posición del carro" automáticamente se carga un mensaje con el texto "@GPS", el número de celular al que se envía el mensaje está definido dentro de la programación y corresponde al chip del módulo GSM. Una vez que el módulo GSM recibe el mensaje, y lo pasa al Arduino, se realiza una comparación con los posibles mensajes que podría estar recibiendo el Arduino, al detectar las siglas "@GPS" se sabe que se requieren los datos del módulo GPS y por tanto se envía a la aplicación dicha información. Cabe señalar, que cuando el módulo GSM recibe un mensaje de la aplicación, el Arduino también envía un mensaje de vuelta para indicarle al usuario que se recibió correctamente la petición, en la aplicación este aviso se muestra mediante una notificación.

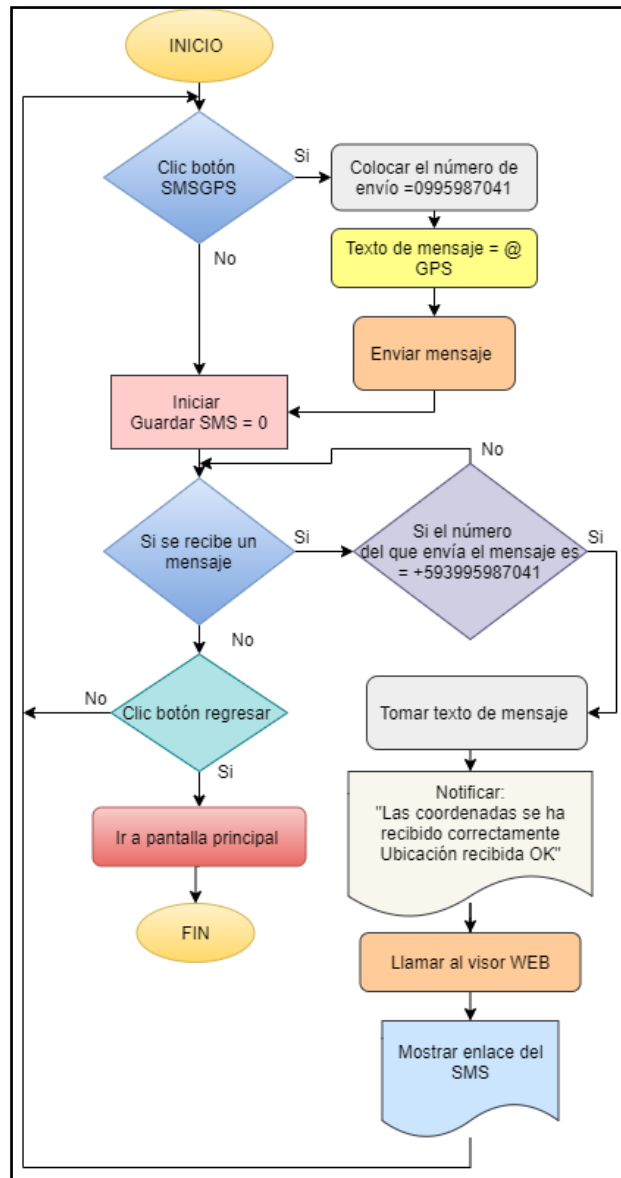


Figura 3.45 Diagrama de flujo de la pantalla "mapa"

Pantalla de la orientación

La pantalla de orientación tiene el propósito de brindar como su nombre lo dice, la orientación del vehículo, lo logra mediante la lectura del magnetómetro ubicado en el prototipo, dicha lectura se envía a la aplicación mediante un mensaje de texto, que la aplicación interpreta y mediante el sistema de orientación que posee el propio celular se logra visualizar en qué dirección va el vehículo.

Esta pantalla cuenta con dos imágenes las cuales una de ella es fija y la otra rota según el ángulo de *azimut* del sensor de orientación del propio celular. Para solicitar los datos de la orientación del vehículo, esta pantalla posee un botón que realiza la petición al

módulo GSM y este responde con un SMS. La respuesta del mensaje se coloca en un recuadro en la parte inferior de la pantalla y con ello permite al usuario saber la orientación que está tomando el automotor. Esta descripción se puede visualizar en la Figura 3.46, la cual corresponde a la de la pantalla orientación.

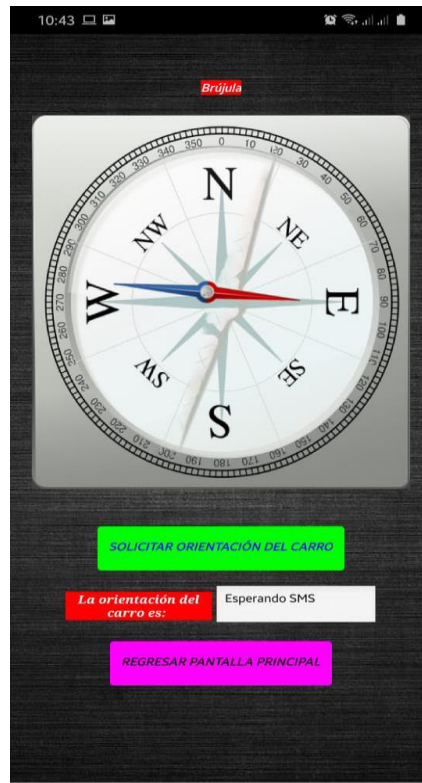


Figura 3.46 Pantalla orientación

Para entender la programación de esta pantalla se puede visualizar el diagrama de flujo que se muestra en la Figura 3.47. Se comienza con la inicialización de la pantalla, donde se muestran dos botones, el primero corresponde a la petición de los datos del magnetómetro, cuando es presionado se envía un mensaje al módulo GSM con el texto @PCG, luego de unos segundos se tienen los datos solicitados. Cuando la aplicación recibe un mensaje, realiza una comparación para saber si el mensaje recibido proviene del módulo GSM, en caso de que así sea se toma el texto del mensaje y este se lo presenta al usuario en el recuadro inferior de la pantalla. El segundo botón que se muestra en la pantalla es para regresar a la ventana principal de la aplicación.

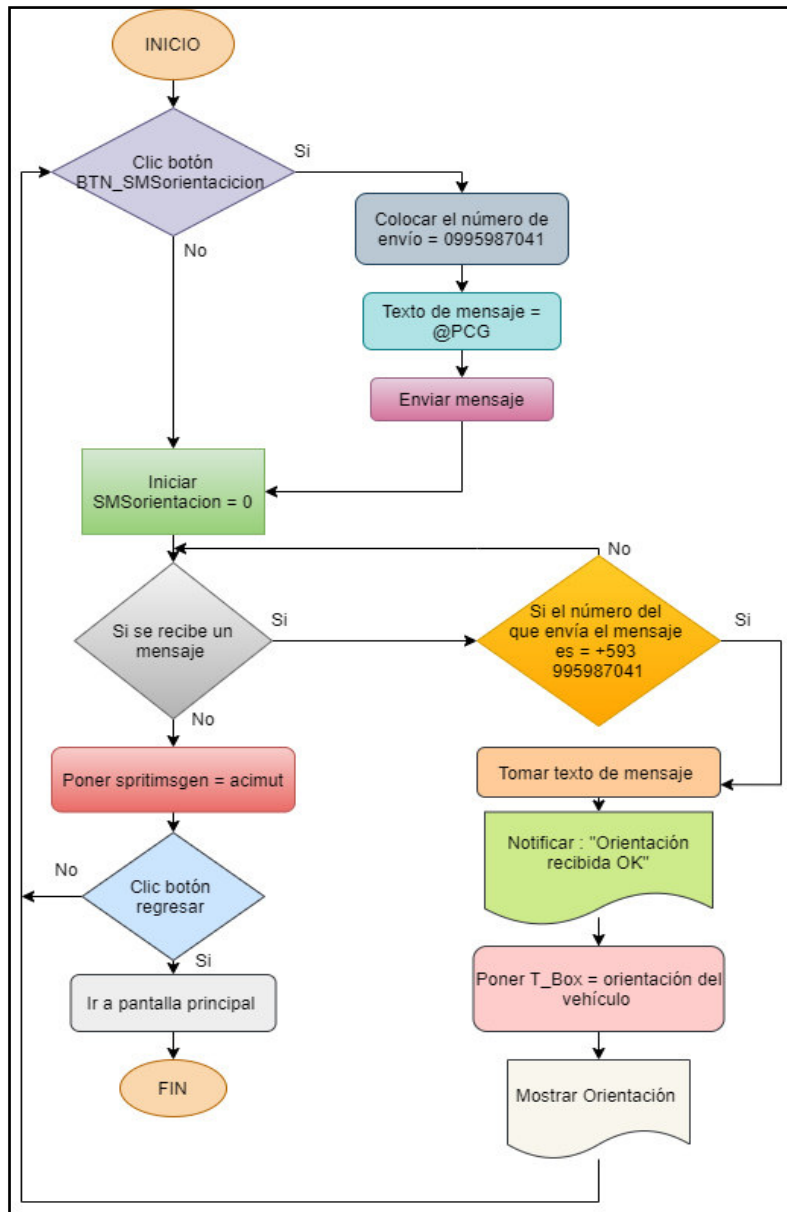


Figura 3.47 Diagrama de flujo pantalla orientación

Pantalla de las alarmas

La pantalla de alarma está configurada para que se pueda activar y desactivar las alarmas que se incorporaron al vehículo. Esta pantalla se muestra en la Figura 3.48, consta de dos botones, de los cuales uno es para activar y el otro es para desactivar la alarma, también cuenta con un indicador que muestra el estado actual de la alarma, es decir, si está activada o no, este indicador incluso cambia de color el texto, siendo de color verde cuando se encuentre activado y rojo cuando no lo está. Otro indicador también es la información del vehículo que de igual manera tiene un campo que indica si alguna de las puertas es abierta o si el vehículo está en movimiento.

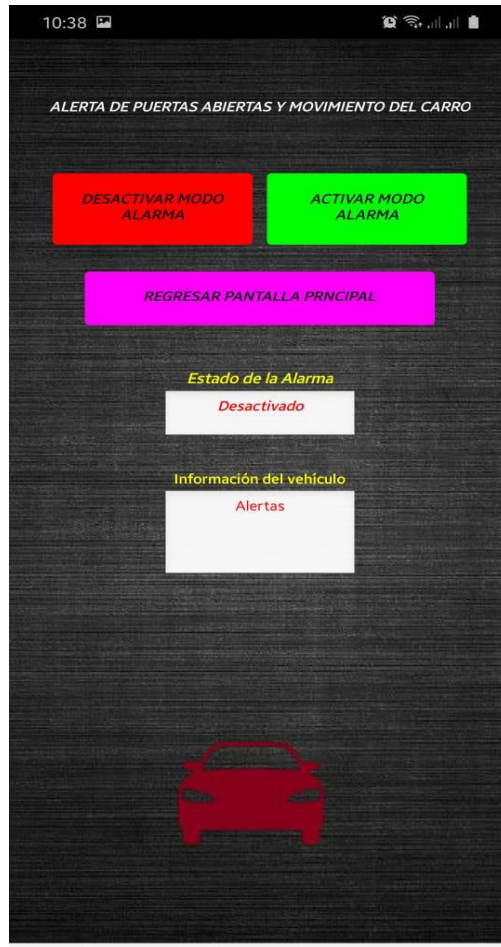


Figura 3.48 Pantalla alarmas

Esta pantalla denominada alarmas se divide en tres diagramas de flujo para entender de mejor manera el funcionamiento y el proceso que cumple como tal.

El primer diagrama se lo observa en la Figura 3.49, se muestra el proceso de enviar las peticiones al módulo GSM. Al momento de presionar alguno de los botones la aplicación completa el número de destino del mensaje con un número previamente guardado, luego se coloca en el campo texto la instrucción de @ACT si se quiere activar las alarmas o de @DES si se requiere desactivarla. Por el contrario, si no se presiona ningún botón la aplicación está recorriendo ese mismo proceso hasta que se realice alguna acción o se presione el botón de regreso que se dirija a la pantalla principal.

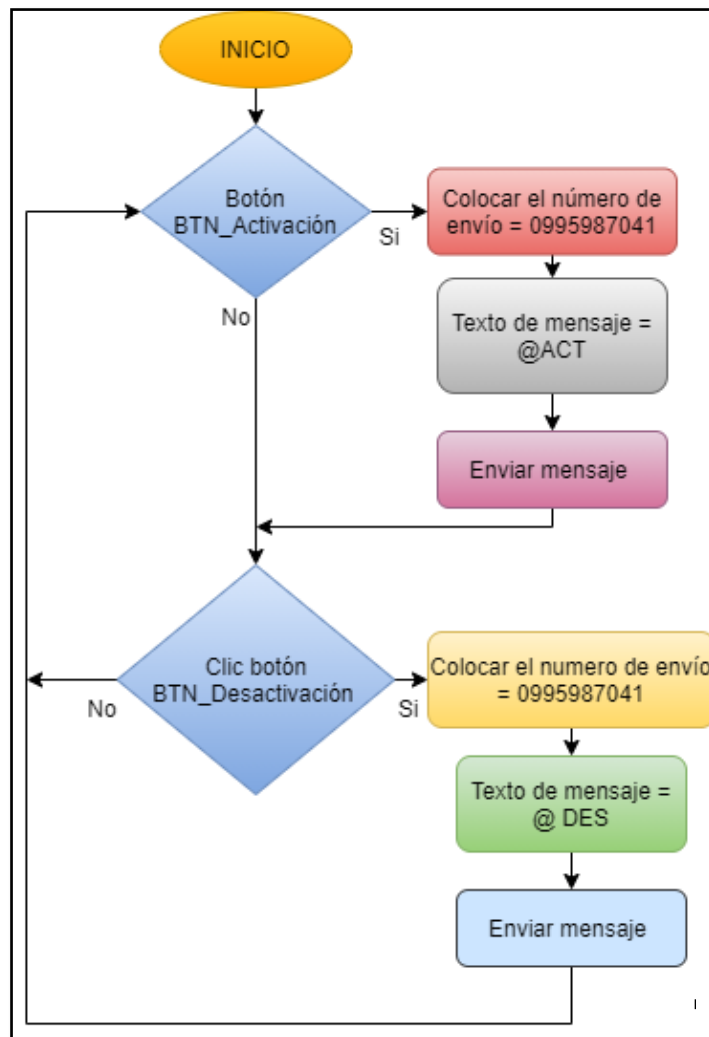


Figura 3.49 Diagrama de flujo pantalla alarmas 1/3

La segunda parte del diagrama de flujo corresponde a la recepción de los mensajes de texto y los diferentes procesos que toma dependiendo del mensaje recibido. De igual forma que en los anteriores bloques, cuando la aplicación recibe un mensaje de texto verifica que el mensaje provenga del prototipo para proceder a guardar los datos en una variable en este caso denominada “SMSpuertas”, luego con el texto que se recibe del prototipo se realiza otra comparativa, si el mensaje tiene la palabra “activado” muestra el mensaje en color verde, en cambio si el mensaje tiene la palabra “desactivado” lo muestra en color rojo.

Cuando el mensaje recibido corresponde a la información de las puertas del vehículo, este texto es mostrado con letras de color rojo en la casilla con la leyenda “alertas”. El proceso se lo puede observar más claramente en la Figura 3.50 y Figura 3.51.

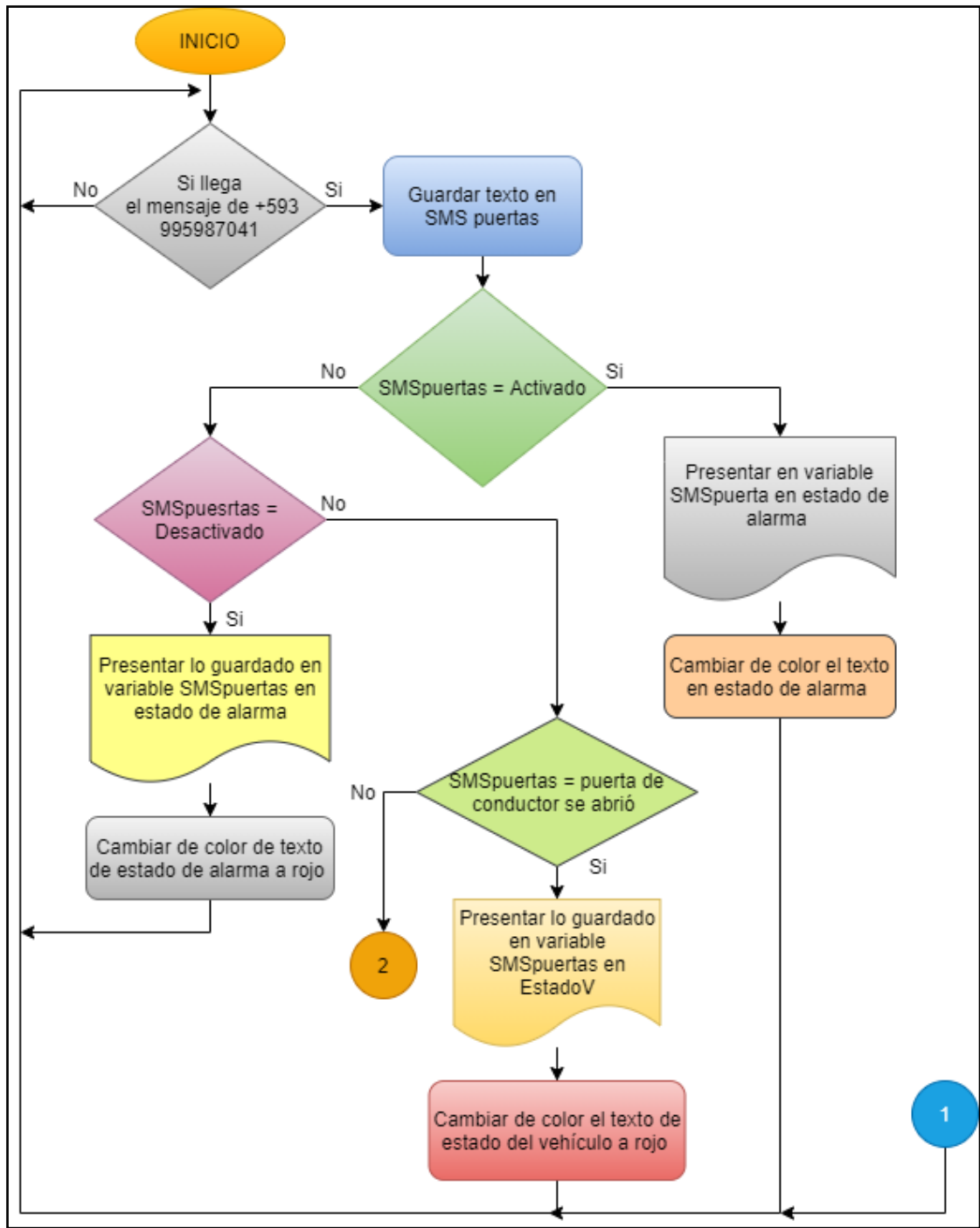


Figura 3.50 Diagrama de flujo pantalla alarmas 2/3

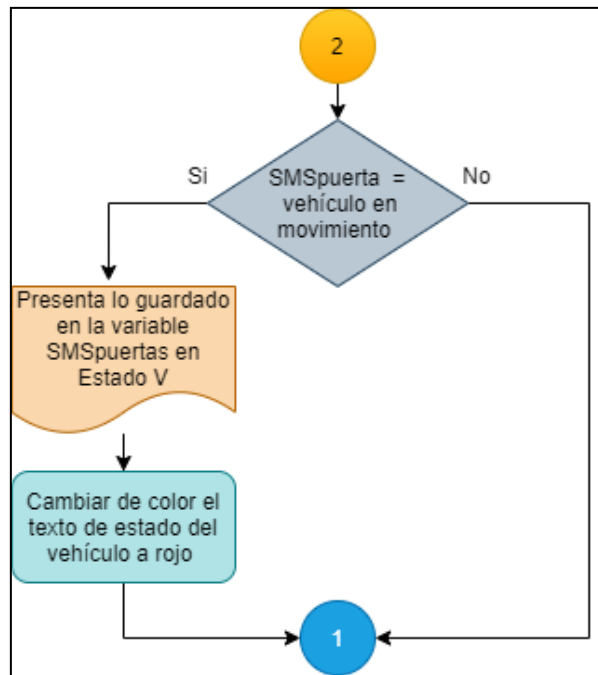


Figura 3.51 Diagrama de flujo pantalla de alarma 3/3

3.5 Pruebas y Análisis de Resultados

Una vez instalados todos los componentes en la maqueta y haber desarrollado la aplicación, se procede con la realización de las pruebas de funcionamiento

Prueba del GPS

La siguiente prueba se realizó con la finalidad de comprobar la precisión del GPS del prototipo, para lo cual se procede a comparar los datos con los de Google Maps.

Para realizar la prueba se coloca el prototipo en un lugar específico de la ciudad, luego con Google Maps se obtiene las coordenadas del lugar, a continuación, usando la aplicación móvil se obtuvieron los datos del GPS del prototipo y por último se procede a comparar ambos resultados, en la

Tabla **3.13** se observa la comparación realizada.

Tabla 3.13 Comparación de coordenadas

Parámetro	Valores de la ubicación usando Google Maps	Valores de la ubicación usando el prototipo	Diferencia en metros entre la ubicación de Google Maps y el GPS del prototipo
Latitud	0°16'01.7"S	0°16'01.2"S	5
Longitud	78°33'04.6"W	78°33'04.4"W	2

Como se puede observar en la

Tabla **3.13**, las diferencias en los valores son de unos pocos metros radiales. La latitud tiene una variación de 5 metros mientras que en la longitud solamente 2 metros, por lo que se puede concluir que el prototipo tiene un grado de fiabilidad alto en cuanto a su localización.

Prueba del modo alarma del prototipo

Esta prueba se realiza con el propósito de verificar que cuando una de las puertas del carro sea abierta, muestre la información pertinente en la aplicación del usuario sin ningún tipo de error y de igual forma comprobar que el prototipo realiza una llamada cuando el auto se mueva.

Para la realización de esta prueba se simula que el vehículo, en este caso la maqueta, estuviese estacionado sobre la acera para posteriormente empleando la aplicación móvil activar el modo alarma, una vez activado este modo se procede a abrir cada una de las puertas del vehículo para así ir verificando que la información que se recibe en la aplicación señale correctamente la puerta que se abrió. El orden en que se abren las puertas del carro es el siguiente, primero la puerta del acompañante luego la puerta del conductor y por último la puerta del pasajero. La información que se obtiene en la aplicación móvil se muestra en la Figura 3.52.

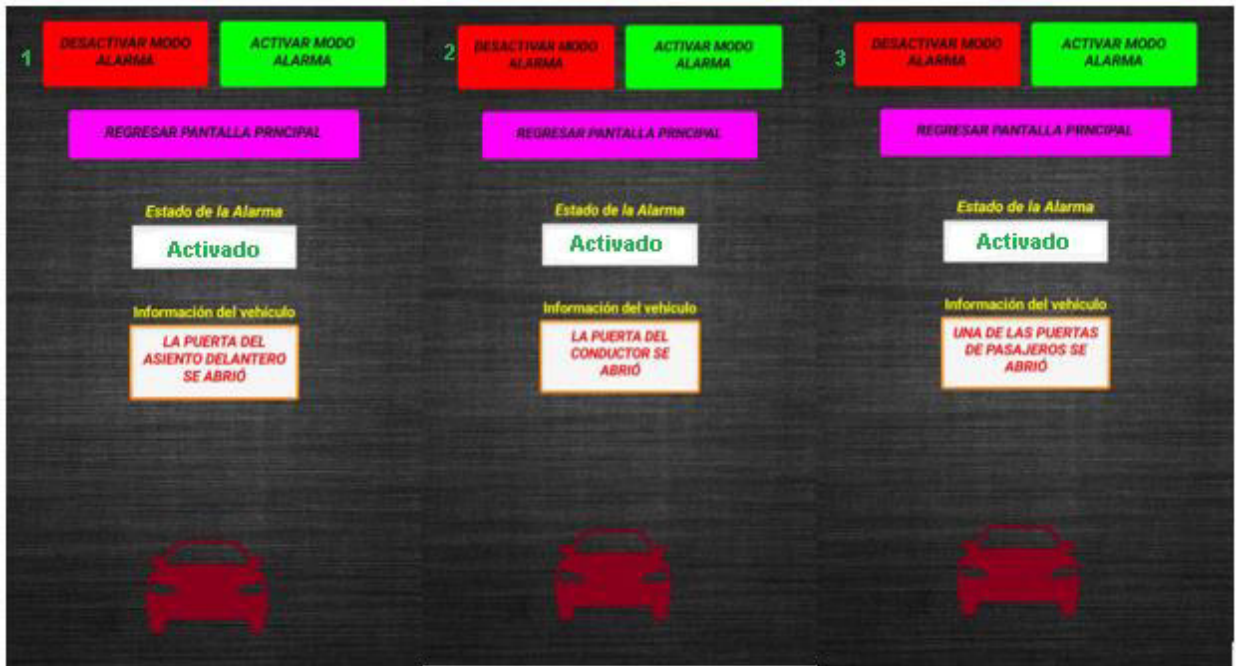


Figura 3.52 Información de las puertas del vehículo recibidas en la aplicación

Como se observa en las imágenes, la información mostrada fue correspondiente al orden de las puertas que se abrieron, es decir, que no existieron inconvenientes y no se presentaron errores. A continuación, en la Figura 3.53 muestra la alerta cuando el vehículo empezó a moverse sin permiso del usuario.



Figura 3.53 Llamada de alerta de movimiento

Como se observa en la Figura 3.53 se recibe una llamada tras solo unos segundos de haber movido el carro, por lo que se puede concluir que el modo alarma del prototipo al estar activado funciona correctamente ya que detecta tanto el abrir de sus puertas como el movimiento del carro y avisa oportunamente al usuario.

Prueba de orientación

Esta prueba se realiza para comprobar que los datos obtenidos del magnetómetro indiquen correctamente los puntos cardinales y esta información se envíe correctamente a la aplicación móvil.

Para realizar la prueba se coloca la maqueta en tres posiciones diferentes y se realiza una lectura en cada posición, una vez obtenido el valor del *azimut* marcado por el sensor se verifica que referencie al punto cardinal de acuerdo a la Tabla 3.14.

Tabla 3.14 Rango de valores que representan los puntos cardinales

Valores del acimut geográfico establecidos en la programación			
Puntos cardinales	Rangos asignados	Puntos laterales	Rangos asignados
Norte	De 0 a 22 grados	Noreste	22 a 67 grados
Este	De 67 a 112 grados	Sureste	112 a 157 grados
Sur	De 157 a 202 grados	Suroeste	202 a 247 grados
Oeste	De 247 a 292 grados	Noroeste	292 a 337 grados

En las siguientes imágenes se observa los resultados obtenidos en la prueba de orientación del vehículo.



Figura 3.54 Datos obtenidos de la posición 1



Figura 3.55 Datos obtenidos de la posición 2

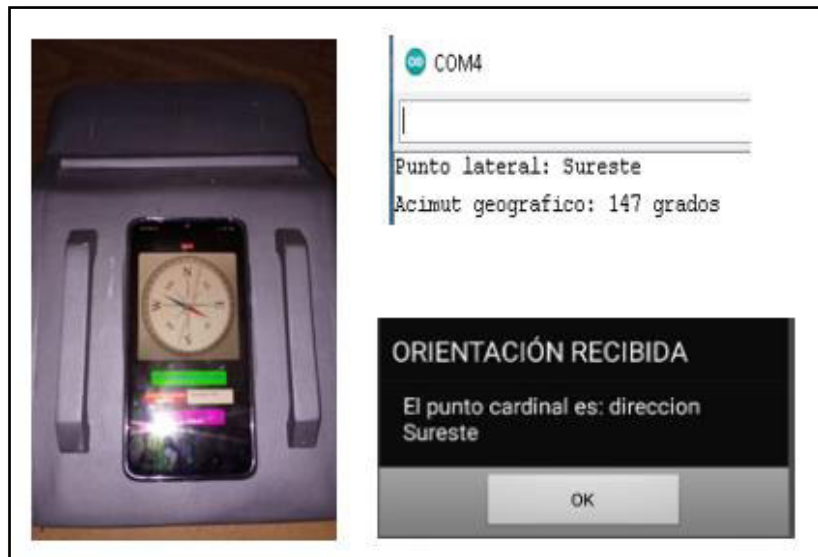


Figura 3.56 Datos obtenidos de la posición 3

Tabla 3.15 Resultados de las pruebas del sensor de orientación

Posiciones de la maqueta	Valor del magnetómetro	Rango en el valor que se encuentra	Punto cardinal teórico referente al rango
1	6 grados	De 0 a 22,5	Norte
2	102 grados	De 67.5 a 112.5	Este
3	147 grados	De 157.5 a 202.5	Sureste

Como se observa en los datos en la Tabla 3.15, los valores marcados por el magnetómetro en cada posición son correctos, ya que indica el punto cardinal de acuerdo al rango de valores que se establecieron en la programación. Además, los valores del sensor fueron ratificados con la ayuda de la brújula creada en la aplicación móvil la cual señala los mismos puntos marcados por el sensor magnetómetro.

3.6 Manual de Uso y Mantenimiento

En los siguientes códigos QR se pueden visualizar los videos que muestran el funcionamiento y el mantenimiento del prototipo.



Figura 3.57 Video del funcionamiento del prototipo



Figura 3.58 Video del mantenimiento del prototipo

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Los objetivos planteados al inicio del trabajo de tesis fueron conseguidos, ya que se logró desarrollar e implementar un prototipo eficiente de monitoreo vehicular, el cual además de saber la posición del automóvil en todo momento también controla otros aspectos importantes como alarmas integradas en las puertas y su posición respecto a los puntos cardinales.
- Los resultados obtenidos en el proyecto fueron satisfactorios debido a que la interfaz creada para controlar el prototipo es sencilla de usar, esto permite que cualquier persona pueda emplearla.
- El prototipo al emplear un módulo GPS con alta precisión y la red GPRS para el envío de la información permite que sea un sistema con una gran cobertura y por ello se concluye que para entornos metropolitanos el sistema funciona de una manera óptima ya que prácticamente la cobertura que tiene es la misma que la de red celular 2G.
- Este sistema está enfocado para automotores pequeños, pero tranquilamente se puede implementar en cualquier automotor incluso de carga pesada, ya que el prototipo no requiere de gran espacio para que sea instalado y puesto en marcha.
- Los componentes del prototipo fueron seleccionados, ya que trabajan con al menos uno de los protocolos de comunicación que utiliza el Arduino haciendo que el manejo de los datos entre los elementos se lo realice de forma coordinada y por tanto que el sistema funcione óptimamente.
- Uno de los aspectos más importantes a la hora de determinar los componentes fue la selección del Arduino, ya que es el elemento centralizado del prototipo, el cual gestiona y maneja toda la información y para su elección fue importante considerar que posea al menos tres puertos seriales, los cuales fueron necesarios para que se lleve a cabo la comunicación entre los elementos electrónicos que conforman el prototipo.
- Gracias a la implementación del módulo de comunicación que trabaja con las tecnologías GSM/GPRS se ha logrado abarcar una amplia cobertura logrando así una comunicación inmediata garantizando que los datos lleguen en el instante que fueron enviados y además libre de errores.

- Con la plataforma App inventor se creó una interfaz capaz de enviar y recibir los datos del prototipo, haciendo que dicha información se visualice de forma clara y todo esto gracias a que la plataforma maneja un lenguaje de programación sencillo, la cual está basada en bloques, lo que no requiere gran conocimiento para usarla.
- Para el desarrollo de la programación del Arduino se lo realizó a base de subrutinas, ya que de esta forma se pueden realizar cambios en el código sin tener que modificar toda la estructura de la programación, además tiene como ventaja de hallar rápidamente errores de funcionamiento sin tener que estar revisando todo el código fuente.

4.2 Recomendaciones

- El sensor magnetómetro debe estar aislado o lejos de los imanes para que estos no interfieran en sus lecturas y se obtengan datos erróneos.
- Para este proyecto es recomendable que la tarjeta SIM empleada en el módulo SIM900 posea un plan de SMS y saldo para enviar la información necesaria para realizar las diferentes peticiones por parte de la aplicación.
- Antes de adquirir los componentes y sensores es recomendable tener en cuenta las características técnicas que posee cada dispositivo y así saber si van a ser compatibles y evitarse gastos innecesarios.
- Es recomendable que se tenga una batería robusta ya que es un sistema que tiene un alto consumo de energía y al ser usado para asegurar que un vehículo no sea sustraído debe estar activo siempre.
- Para optimizar el funcionamiento de la programación de Arduino, dimensionar de mejor manera el microcontrolador para que el tamaño del código no supere el 75% de la memoria dinámica.
- La ubicación de la antena del módulo del GPS debe tener línea de vista con la atmosfera para que tarde lo menos posible en conectarse con los satélites y así poder obtener de forma rápida y eficaz las coordenadas de ubicación del vehículo.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] «El Universo,» 2 Febrero 2020. [En línea]. Available: <https://www.eluniverso.com/noticias/2020/02/02/nota/7719456/delincuencia-usa-receptores-senal-abrir-carros-robar>. [Último acceso: 08 Mayo 2020].
- [2] «coello,» Introduccción al Sistema GPS, [En línea]. Available: http://coello.ujaen.es/asignaturas/mtopo/descargas/GPSBasics_es.pdf. [Último acceso: 02 04 2021].
- [3] Y. FM, «Xataka,» Qué es Arduino, cómo funciona y qué puedes hacer con uno, 08 03 2018. [En línea]. Available: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>. [Último acceso: 07 07 2020].
- [4] E. Gómez, «rinconingenieril,» Cómo funciona el Puerto Serie y la UART, 31 10 2017. [En línea]. Available: <https://www.rinconingenieril.es/funciona-puerto-serie-la-uart/>. [Último acceso: 2 04 2021].
- [5] «Descripción y funcionamiento del Bus I2C,» Robots Didácticos, 07 12 2018. [En línea]. Available: <http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>. [Último acceso: 27 08 2020].
- [6] «Bluehack,» [En línea]. Available: <http://bluehack.elhacker.net/proyectos/comandosat/comandosat.html>. [Último acceso: 10 Julio 2020].
- [7] «lastminuteengineers,» Enviar Recibir SMS y llamar con SIM900 GSM Shield y Arduino, [En línea]. Available: <https://lastminuteengineers.com/sim900-gsm-shield-arduino-tutorial/>. [Último acceso: 15 04 2021].
- [8] M. M. Roa, «atista,» Android e iOS dominan el mercado de los smartphones, 30 07 2020. [En línea]. Available: <https://es.statista.com/grafico/18920/cuota-de-mercado-mundial-de-smartphones-por-sistema-operativo/>. [Último acceso: 30 07 2021].
- [9] u.-b. ublox, «NEO-6 u-blox 6 GPS Modules Data Sheet,» geekfactory, [En línea]. Available: [https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf?utm_source=en%2Fimages%2Fdownloads%2FProduct_Docs%](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf?utm_source=en%2Fimages%2Fdownloads%2FProduct_Docs%2F)

2FNEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf. [Último acceso: 24 09 2020].

- [10] «puntoflotante,» Módulo SkyLab SKM53 para aplicaciones GPS con microcontroladores., [En línea]. Available: <https://www.puntoflotante.net/MODULO-GPS-SKYLAB-SKM53-PARA-MICROCONTROLADORES.htm#:~:text=El%20m%C3%B3dulo%20SKM53%20 cuenta%20con,metros%20la%20especificaci%C3%B3n%20del%20fabricante..> [Último acceso: 2021 04 2021].
- [11] «tecbolivia,» Módulo GPS Skylab SKM53, [En línea]. Available: <http://tecbolivia.com/index.php/venta-de-componentes-electronicos-11/comunicaciones/modulo-gps-skylab-skm53-detail>. [Último acceso: 15 04 2021].
- [12] «makersportal,» NEO-6M GPS MODULE, [En línea]. Available: <https://makersportal.com/shop/neo-6m-gps-module>. [Último acceso: 11 04 2021].
- [13] «App net,» App Inventor: Desarrollador online para Apps en Android, [En línea]. Available: <https://www.tu-app.net/blog/app-inventor/>. [Último acceso: 31 03 2021].
- [14] «naylampmechatronics,» TUTORIAL MAGNETÓMETRO HMC5883L, [En línea]. Available: https://naylampmechatronics.com/blog/49_tutorial-magnetometro-hmc5883l.html. [Último acceso: 31 03 2021].
- [15] «Rambal,» Magnetometro 3 Ejes GY271-HMC5883L, [En línea]. Available: <https://rambal.com/acelerac-inclinac-giroscop/970-magnetometro-compas-3-ejes-gy271-hmc5883l.html>. [Último acceso: 12 04 2021].
- [16] «Afel,» Sensor Magnetometro HMC5883L, [En línea]. Available: <https://afel.cl/producto/sensor-magnetometro-hmc5883l/>. [Último acceso: 16 08 2020].
- [17] «La electronica,» Switch Magnético, [En línea]. Available: <https://laelectronica.com.gt/switch-magnetico-pareja>. [Último acceso: 17 09 2020].

- [18] «complubot,» Guía para la selección de una controladora Arduino, 26 02 2015. [En línea]. Available: <https://complubot.com/guia-para-la-seleccion-de-una-controladora-arduino/>. [Último acceso: 02 04 2021].
- [19] «blogmasterwalkershop,» Switch MC-38, [En línea]. Available: <https://blogmasterwalkershop.com.br/arquivos/datasheet/Datasheet%20MC-38.pdf>. [Último acceso: 14 04 2021].
- [20] «Mercado libre,» Modulo Bms 2s Cargador 2 Celdas 18650 Protec. 7.4v 8.4v 3a, [En línea]. Available: https://articulo.mercadolibre.com.ar/MLA-873673044-modulo-bms-2s-cargador-2-celdas-18650-protec-74v-84v-3a-_JM#position=2&type=item&tracking_id=3255f8db-7a29-4cb4-a0c9-1f1a7e98947c. [Último acceso: 17 04 2021].
- [21] R. Alonso, «hardzone,» Raspberry Pi vs Arduino, ¿en qué se diferencian y para qué se usan?, 03 04 2020. [En línea]. Available: <https://hardzone.es/reportajes/comparativas/raspberry-pi-vs-arduino/>. [Último acceso: 1 04 2021].