

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA:

COMPUTACIÓN

Trabajo de titulación previo a la obtención del título de:

INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

TEMA:

**“SISTEMA EMBEBIDO PARA ALERTAR ROBOS Y SECUESTROS DE
CONDUCTORES EN AUTOMÓVILES.”**

AUTOR:

JORGE FABIAN ZAPATA CAISAGUANO

TUTOR:

MANUEL RAFAEL JAYA DUCHE

Quito, noviembre de 2021

CESIÓN DE DERECHOS DE AUTOR

Yo, Jorge Fabian Zapata Caisaguano, con documento de identificación N°172642434-2, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado: “SISTEMA EMBEBIDO PARA ALERTAR ROBOS Y SECUESTROS DE CONDUCTORES EN AUTOMÓVILES”, mismo que ha sido desarrollado para optar por el título de: INGENIERO EN CIENCIAS DE LA COMPUTACIÓN, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....

Jorge Fabian Zapata Caisaguano

C.I: 1726424342

Quito, noviembre de 2021

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “SISTEMA EMBEBIDO PARA ALERTAR ROBOS Y SECUESTROS DE CONDUCTORES EN AUTOMÓVILES”, realizado por Jorge Fabian Zapata Caisaguano, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

A handwritten signature in blue ink, appearing to be 'Manuel Rafael Jaya Duche', written over a horizontal dotted line.

Magister Manuel Rafael Jaya Duche

C.I: 1710631035

Quito, noviembre de 2021

ÍNDICE

CAPÍTULO 1	1
Introducción	1
1.1 Antecedentes	1
1.2 Problema.....	5
1.3 Justificación.....	6
1.4 Objetivos	7
1.5 Metodología	7
CAPÍTULO 2	8
Marco teórico	8
2. ANALISIS DE HARDWARE Y SOFTWARE	8
2.1 HARDWARE.....	13
2.2 SOFTWARE	22
CAPÍTULO 3	25
3. DISEÑO E IMPLEMENTACIÓN	25
3.1 Construcción del prototipo	27
3.2 Diseño del programa de arduino	31
3.3 Diseño de la aplicación móvil	33
CAPÍTULO 4	36
4. PRUEBAS Y RESULTADOS	36
4.1 Pruebas del dispositivo.....	36
4.2 Pruebas de la aplicación	42
4.3 Resultados	50
4.4 Encuesta	57
CONCLUSIONES.....	61
RECOMENDACIONES	62
REFERENCIAS	63

ÍNDICE DE FIGURAS

Figura 1 Escala cualitativa para los componentes	8
Figura 2 Comparativa GPS/GSM.....	9
Figura 3 Calificación GPS/GSM	10
Figura 4 Resultados GPS/GSM	10
Figura 5 Comparativa Arduino.....	11
Figura 6 Calificación Arduino.....	12
Figura 7 Resultados Arduino.....	12
Figura 8. Diagrama de pines.....	17
Figura 9. Descripción general placa SIM808	20
Figura 10. Mini pulsador rojo.....	21
Figura 11. Entorno del SW Arduino.....	23
Figura 12. Interfaces de App inventor	24
Figura 13. Componentes.....	26
Figura 14. Vista lateral izquierda del circuito ensamblado	28
Figura 15. Vista lateral derecha del circuito ensamblado.....	28
Figura 16. Vista superior del circuito ensamblado	29
Figura 17. Comunicación serie	30
Figura 18. Funcionamiento a través del comando AT.....	30
Figura 19. Llamada telefónica a través de comandos AT	31
Figura 20. Inicio del programa Arduino.....	32
Figura 21. Adquisición de datos del módulo SIM808.....	32
Figura 22. Envío de mensajes texto.....	33
Figura 23. Comando AT para llamadas.....	33
Figura 24. Diseño de la aplicación en MIT App Inventor.....	34
Figura 25. Inicialización para SMS.	35
Figura 26. Proceso de uso de datos recibidos.....	35
Figura 27. Pulsador y protoboard.	36
Figura 28. Dispositivo de pruebas.	37
Figura 29. Prueba de recepción de SMS.....	38
Figura 30. Prueba de recepción de SMS.....	39
Figura 31. Código para recolección de datos.	40
Figura 32. Prueba de recepción de SMS.....	41

Figura 33. Prueba de recepción de llamadas.	42
Figura 34. Coordenadas de Google maps y del dispositivo SIM808.	43
Figura 35. Resultados.	44
Figura 36. Comparativa Coordenadas.	44
Figura 37. Instalación de la aplicación.	45
Figura 38. Pruebas de coordenadas.	46
Figura 39. Pruebas de SMS.	46
Figura 40. Llamada del botón de pánico.	48
Figura 41. Mensaje del botón de pánico y ubicación automática de coordenadas.	48
Figura 42. Notificación de la aplicación.	50
Figura 43. Ubicación de la placa de soporte del Arduino.	51
Figura 44. Sistema alimentado desde la batería del automóvil.	51
Figura 45. Botón de pánico ubicado en la palanca de cambios.	52
Figura 46. Gráfico de dispersión.	53
Figura 47. Datos aplicando la ecuación de ajuste.	54
Figura 48. Comparativa de los resultados.	54
Figura 49. Tiempo de recepción de SMS.	55
Figura 50. Tiempo de los mensajes de texto.	56
Figura 51. Tiempo de las llamadas.	56
Figura 52. Tiempos de envío de SMS a dos teléfonos.	56
Figura 53. Resultados pregunta 1	57
Figura 54. Resultados pregunta 2	58
Figura 55. Resultados pregunta 3	58
Figura 56. Resultados pregunta 4	59
Figura 57. Resultados pregunta 5	59
Figura 58. Resultados pregunta 6.	60

ÍNDICE DE TABLAS

Tabla 1. Especificaciones técnicas Arduino UNO R3.....	15
Tabla 2. Especificaciones GSM/GPRS/GPS SIM808 Shield.....	19
Tabla 3. Características Mini Pulsador SPST.....	21
Tabla 4. Interpretación de datos recibidos.....	42
Tabla 5. Resultados de los cálculos.....	43

RESUMEN

El proyecto está orientado a realizar un sistema embebido que alerta robos y secuestros de conductores en automóviles para ayudar a mejorar la seguridad de los usuarios. Se hace uso de una metodología de cuestionario, para determinar la satisfacción de los usuarios y posibles mejoras y el modelo en cascada, determinando las fases de construcción para luego realizarlas. Este sistema brindará una alerta por vía SMS y llamada telefónica, realizada por un módulo SIM808 con un dispositivo Arduino UNO, desencadenando una indicación de alerta para los dispositivos Smartphone de los familiares o amigos de su posición usando GPS, se expondrá el funcionamiento conjunto de los módulos con la aplicación móvil, que al momento de activar el botón se envía mensajes de texto y al recibirse estos en el dispositivo smartphone, con la aplicación abierta, ubica de manera automática las coordenadas en el mapa.

ABSTRACT

The project is aimed at creating an embedded system that alerts thefts and kidnappings of drivers in cars to help improve user safety. A questionnaire methodology is used to determine user satisfaction and possible improvements and the cascade model, determining the construction phases, and then carrying them out. This system will provide an alert via SMS and telephone call, carried out by a SIM808 module with an Arduino UNO device, triggering an alert indication for the Smartphone devices of relatives or friends of their position using GPS, the joint operation of the devices will be exposed. modules with the mobile application, which when the button is activated, text messages are sent and when these are received on the smartphone device, with the application open, it automatically locates the coordinates on the map.

CAPÍTULO 1

Introducción

1.1 Antecedentes

(Sancho Chilet, 2017) en su estudio tuvo como objetivo general la construcción y programación de un sistema de localización de vehículos mediante GPS basado en un dispositivo, al que se ha nombrado UBIC, y una aplicación Android, a la que se ha llamado UBICCAPP. La metodología utilizada es cuantitativa donde el tipo de estudio es no experimental, ya que no se manipulará las variables, sino que se usaran directamente esos datos para la aplicación. En los resultados, se ha obtenido un sistema de fácil utilización para el usuario con una aplicación con interfaz sencilla y con una curva de aprendizaje corta y un dispositivo que solo se necesita de encender un interruptor para que funcione. Concluyendo que tras la comparativa de otros localizadores se ha conseguido realizar un sistema de comunicación entre dispositivo y aplicación basado en Wifi que ofrece mayor rango de comunicación que la mayoría de conexión Bluetooth de los localizadores encontrados en el mercado. También cabe destacar la gran cantidad de trabajo que ha supuesto el desarrollo de este sistema de comunicación hasta conseguir su funcionamiento.

(Martínez Orozco & Callejas Piñeros, 2016) tuvo como objetivo general diseñar e implementar un sistema electrónico de geoposicionamiento y control sobre el sistema de ignición de una motocicleta utilizando un APK para celulares con sistema operativo Android. La metodología utilizada es cuantitativa donde el tipo de estudio es no experimental, ya que no se manipulará las variables, sino que se usaran directamente esos datos para la aplicación. En los resultados se demostró que Arduino puede ser una excelente plataforma para desarrollar diversidad de prototipos electrónicos, gracias a su

amplia gama de componentes que se desarrollan bajo esta plataforma, así como componentes externos comunes, dejando atrás el concepto de ser una herramienta estudiantil. Concluyendo que tienen un excelente rendimiento en cuanto a transmisión y recepción de datos, manejando una comunicación de comandos rápidos con nuestra motocicleta con un alcance de transmisión de 10 metros para el Bluetooth y unos 55 metros de alcance con el llavero RF, teniendo en cuenta que el entorno sea libre de obstáculos como paredes o vitrinas, además que el funcionamiento del software en línea AppInventor, es apto para el desarrollo de programas que trabajen con manejo de datos a distancia, ya sea por Bluetooth o GSM, contando con una gran cantidad de herramientas que posee un celular. (Cámara, sensor de inclinación, almacenamiento interno, reproductor de imágenes y video, entre otras.)

(Salas Robles, 2017) tuvo como objetivo general desarrollar un prototipo de un sistema de seguridad basado en tecnología GSM que monitoree los eventos que se den en un automóvil y para los cuales se programará una serie de acciones a realizar si es que se dan, por ejemplo: el que se abran puertas o el cofre. La metodología utilizada es cuantitativa donde el tipo de estudio es no experimental, ya que no se manipulará las variables, sino que se usaran directamente esos datos para la aplicación. En los resultados se tiene que la tarjeta AVR GSM se comunica eficientemente con el teléfono móvil a través de mensajes SMS que toman aproximadamente de 5 a 6 segundos en ser recibidos e interpretados por el mismo teléfono móvil del usuario final. Dichos mensajes SMS, contienen los estados o eventos del cofre y puertas del prototipo, además el tiempo aproximado de respuesta al enviar el mensaje SMS GETSTATUS del teléfono móvil a la tarjeta AVR GSM, también es similar al anterior, de 4 a 5 segundos en ser recibido por el teléfono móvil del usuario final. Concluyendo que este prototipo de sistema de seguridad para automóviles con tecnología GSM es muy confiable y eficiente. Demostrando que el

uso de la tarjeta AVR GSM ha logrado mediante sus múltiples funciones que este proyecto sea factible, al interpretar la variación de voltaje en sus dos entradas y ejecutar las instrucciones previamente guardadas en el microcontrolador al enviar los mensajes SMS de notificación al usuario final sobre el estado de las puertas y cofre de su automóvil.

(Lima León & Valdez Bermejo, 2019) tuvo como objetivo general diseñar un prototipo de botón de pánico móvil con ubicación GPS para el monitoreo de adultos mayores en el cantón Cuenca, por medio del sistema ECU 911. La metodología utilizada es cuantitativa donde el tipo de estudio es no experimental, ya que no se manipulará las variables, sino que se usaran directamente esos datos para la aplicación. En los resultados se comprobó a través de pruebas los tiempos de respuesta y la cobertura, siendo aceptable y satisfactorio para su propósito. Concluyendo que el hardware y software empleado para el desarrollo de este sistema cumple con los requerimientos y características que se pretende obtener para la transmisión de información de eventos de emergencia y geolocalización en tiempo real y que este evento se transmita de forma automática. El hardware utilizado como protocolo de comunicación logro cumplir con el objetivo de comunicar el prototipo con el sistema de gestión.

(Guayaquil Gonzabay, 2018) tuvo como objetivo desarrollar un modelo de sistema para mejorar la seguridad vehicular a través de redes móviles. La metodología utilizada es cuantitativa donde el tipo de estudio es no experimental, ya que no se manipulará las variables, sino que se usaran directamente esos datos para la aplicación además de que se hizo uso de la metodología MPR “Metodología de Prototipado rápido” el cual permitirá que el propio usuario debe comprobar que el prototipo es idóneo de informar dicho evento en el momento preciso de darse una anomalía. En los resultados la funcionalidad del sensor de movimiento debe ser regulada según las especificaciones del auto, así se tendrá una cobertura más amplia del rango de validación, teniendo en cuenta que el voltaje para

que la sirena funcione perfectamente debe superar los 10 voltios, esto dará un sonido envolvente hacia el exterior del vehículo que podrá ser escuchado a una distancia considerable por la persona encargada del mismo. Concluyendo que los objetivos propuestos al principio fueron alcanzados mediante la concepción de un sistema de seguridad vehicular por redes GSM, brindando seguridad y confiabilidad al dejar estacionada su vehículo en sitios inseguros.

(Cajas Malave & Raymondi Lomas, 2018) tuvo como objetivo implementar un prototipo de alerta en situaciones de emergencia utilizando herramientas de geocalización y sensores biométricos. La metodología utilizada es el estudio exploratorio ya que tiene como primer nivel de conocimiento científico de un problema de investigación se logra a través de estudios de tipo exploratorio; este tiene por objetivo, la formulación de un problema para posibilitar una investigación más precisa o el desarrollo de una hipótesis, permitiendo al investigador formular hipótesis de primer y segundo grado. En los resultados el proyecto elaborado responde de manera positiva a los objetivos expuestos al principio del proyecto de investigación. Concluyendo que con el prototipo se busca que las reacciones psicológicas naturales como el nerviosismo, miedo o ansiedad que tenemos al momento de una situación de emergencia, ayudando a monitorear su frecuencia cardiaca ante una posible situación de peligro para poder brindar la ayuda adecuada en un periodo de tiempo corto.

1.2 Problema

En Ecuador el robo de vehículos y el daño a los conductores no es algo nuevo, en las noticias tanto transmisiones de radio y televisión como impresas es prácticamente diario los reportes de robos o secuestros exprés en diferentes provincias del país como Guayas.

En Quito, se ha reportado el mayor índice de estos casos, llegando a superar el de ciudades donde la violencia es muy elevada como Guayaquil (Medina, 2021), haciendo evidente la necesidad de tener una ayuda para lo cual se busca desarrollar un sistema de seguridad contra robos o secuestros, a través de un botón de pánico vinculado a un aplicativo móvil, a fin de reducir estos actos violentos contra los conductores y sus automóviles (El Comercio, 2021).

Entre las causas que pueden estar originando este problema, hemos detectado la inseguridad que es algo que siempre se ha mantenido dentro de la sociedad por lo que se busca una herramienta para ayudar a reducirla.

Los efectos que pueden traer este problema son diversos, entre los que están perdida de automóviles y secuestro exprés que representan perdidas grandes, no solo de bienes sino también de vidas humanas en el peor de los casos.

Por ello, la presente investigación pretende ayudar a este problema a través del desarrollo de un sistema embebido, con botón de pánico oculto para automóviles que envíe señales de alerta y ubicación a dispositivos Android/IOS para su rastreo inmediato y en tiempo real, esto brindará una oportunidad de alertar el peligro de manera inmediata para mejorar la respuesta de las personas ante estos eventos, el desarrollo del proyecto se expone en cinco capítulos que sintetizan el trabajo realizado para conseguir la temática propuesta.

1.3 Justificación

En la actualidad en el Ecuador ha aumentado los delitos en automóviles, desde robos hasta secuestros, lo que hace necesario nuevas formas en la cual proteger la integridad de las personas y vehículos.

Se busca diseñar un sistema embebido que alerte, enviando la ubicación en tiempo real del vehículo cuando el mismo este expuesto a robo o secuestro exprés. Los datos enviados serán recibidos en dispositivos de familiares programados con la aplicación para recibir las alertas y la ubicación en tiempo real del vehículo. Este sistema contará con módulos GSM y GPS.

Este trabajo tiene como finalidad brindar un soporte para los usuarios de automóviles al momento de sufrir este tipo de incidentes, dando una mayor sensación de seguridad y se espera que al implementarse pueda reducir los casos antes mencionados.

El incentivo de este es, como ya se mencionó, brindar un soporte para los usuarios de automóviles, ya que, al momento de ocurrir estos percances, no solo es una preocupación para los afectados directos sino también para sus familiares, que sienten la perdida como propia.

Para finalizar este trabajo está dirigido a todas las personas que usen o tengan un vehículo automotor, además de las que simplemente usen uno como parte de su trabajo o para uso personal.

1.4 Objetivos

1.4.1 Objetivo general:

Elaborar un sistema embebido para alertar robos y secuestros de conductores en automóviles.

1.4.2 Objetivos específicos:

- Construir un botón de pánico, buscando que sus dimensiones sean las menores posibles.
- Construir la aplicación móvil, usando una interfaz amigable con el usuario y de fácil usabilidad.
- Evaluar los resultados del sistema embebido y la satisfacción de los usuarios.

1.5 Metodología

En el proyecto se usará el método cualitativo, ya que, aunque se maneje datos, estos son puramente para su uso en la aplicación y estadísticos sin necesidad de manipularlos, además es del tipo encuesta y cuestionario para conseguir la opinión del usuario sobre la implementación, funcionamiento y mejoras del sistema (García Córdoba, 2005). Por último, la metodología de administración es waterfall o cascada, buscando planificar por completo para luego construir cada fase.

CAPÍTULO 2

Marco teórico

2. ANALISIS DE HARDWARE Y SOFTWARE

Para este proyecto se utilizó una combinación de software y hardware el cual se vincula a través de mensajes de texto SMS en una sola dirección, del dispositivo hacia la aplicación, donde se usa información en un formato predeterminado el cual posteriormente utiliza la aplicación para su funcionamiento.

Para el análisis de los componentes de hardware se utilizó una escala cualitativa de sus características, con los siguientes valores (CHAMBA SALAVARRÍA, 2017):

Figura 1 Escala cualitativa para los componentes

5	4	3	2	1
Muy adecuado	Adecuado	Poco adecuado	Inadecuado	Nada adecuado

Fuente: Realizado por el Autor.

Primero se revisó la placa GSM/GPS, se hizo una comparación entre dos modelos.

Figura 2 Comparativa GPS/GSM

Características	SIM 808	SIM 900
Tipos de Bandas	850/900/1800/1900 MHz	850/900/1800/1900 MHz
Estación Móvil	Clase B	Clase B
Compatibilidad	Arduino Uno y Mega	Arduino Uno y Mega
Antena GPS externa	Si	No
Reloj interno	Tiempo Real	Tiempo Real
Comandos	AT	AT
Consumo de energía	1 mA	1,5 mA
Tamaño	5x7,8 cm	5x4,8 cm
Precisión GPS	2,5 m	3 m
Costo	\$50.00	\$35.04

Fuente: Realizado por el Autor.

Luego se procede a calificar estas características:

Figura 3 Calificación GPS/GSM

Características	SIM 808	SIM 900
Tipos de Bandas	5	5
Estación Móvil	4	4
Compatibilidad	4	4
Antena GPS externa	4	1
Reloj interno	4	4
Comandos	4	4
Consumo de energía	4	3
Tamaño	3	4
Precisión GPS	4	3
Costo	4	5
Total	40	37

Fuente: Realizado por el Autor.

Lo que da como resultado lo siguiente:

Figura 4 Resultados GPS/GSM

Máximo	SIM 808	SIM 900
50	40	37
100%	80%	74%

Fuente: Realizado por el Autor.

Se escogió el SIM808, ya que después de realizar la comparativa de los mismos para el proyecto tiene un porcentaje más alto entre los 2.

Una vez realizado esto, se procede a la elección de la placa Arduino que se va a utilizar, para ello, y teniendo en cuenta la tabla 2, en la parte de compatibilidad, se realizó la comparación de las siguientes placas:

Figura 5 Comparativa Arduino

Características	UNO	MEGA
Microcontrolador	ATmega328	ATmega2560
Voltaje de Entrada	7-12V	7-12V
I/O Digitales	14	54
Entradas Analógicas	6	16
Memoria SRAM	2KB	8KB
Memoria EEPROM	1KB	4KB
Frecuencia de Reloj	16MHz	16MHz
Interfaz de Programación	USB	USB
Dimensiones	68.6 x 53.4 mm	101.52 x 53.3 mm
Costo	\$ 34.82	\$ 59.82

Fuente: Realizado por el Autor.

Luego se procede a calificar estas características:

Figura 6 Calificación Arduino

Características	UNO	MEGA
Microcontrolador	5	1
Voltaje de Entrada	4	4
I/O Digitales	3	5
Entradas Analógicas	3	4
Memoria SRAM	3	4
Memoria EEPROM	3	4
Frecuencia de Reloj	3	3
Interfaz de Programación	5	5
Dimensiones	4	2
Costo	5	2
Total	38	34

Fuente: Realizado por el Autor.

Lo que da como resultado lo siguiente:

Figura 7 Resultados Arduino

Máximo	UNO	MEGA
50	38	34
100%	76%	68%

Fuente: Realizado por el Autor.

De manera similar a lo anterior, se procedió a elegir la placa Arduino UNO, ya que al realizar la respectiva comparación de estas esta dio un mejor resultado para ser utilizada en el proyecto.

Por otro lado, para la elección del software donde se realizara el prototipo de la aplicación para el proyecto se eligió MIT App Inventor por dos razones, la primera, y más importante, el conocimiento y la experiencia que ya se tiene con las misma en comparativa con otros programas de creación de aplicaciones móviles (como Android estudio) y la segunda porque App Inventor es un entorno amigable y práctico para la creación y experimentación de aplicaciones, además el vínculo entre el dispositivo y la aplicación es a través de SMS, como ya menciono anteriormente, por lo que no se necesita enfocar en la parte de la aplicación.

2.1 HARDWARE

2.1.1 ARDUINO

Es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida, activando un motor, encendiendo un LED, publicando algo en línea. Puede decirle a su tablero qué hacer enviando un conjunto de instrucciones al microcontrolador en el tablero. Para hacerlo, utiliza el lenguaje de programación Arduino (basado en Wiring) y el Software Arduino (IDE), basado en Processing.

A lo largo de los años, Arduino ha sido el cerebro de miles de proyectos, desde objetos cotidianos hasta complejos instrumentos científicos. Una comunidad mundial de creadores (estudiantes, aficionados, artistas, programadores y profesionales) se ha reunido en torno a esta plataforma de código abierto, sus contribuciones se han sumado a

una increíble cantidad de conocimiento accesible que puede ser de gran ayuda tanto para principiantes como para expertos (ARDUINO, 2021).

Arduino nació en el Ivrea Interaction Design Institute como una herramienta fácil para la creación rápida de prototipos, dirigida a estudiantes sin experiencia en electrónica y programación. Tan pronto como llegó a una comunidad más amplia, la placa Arduino comenzó a cambiar para adaptarse a las nuevas necesidades y desafíos, diferenciando su oferta desde placas simples de 8 bits hasta productos para aplicaciones de IoT, wearable, impresión 3D y entornos integrados. Todas las placas Arduino son completamente de código abierto, lo que permite a los usuarios construirlas de forma independiente y eventualmente adaptarlas a sus necesidades particulares.

El software también es de código abierto y está creciendo gracias a las contribuciones de los usuarios de todo el mundo. (ARDUINO, 2021)

2.1.2 ARDUINO UNO R3

Es una placa de microcontrolador basada en ATmega328P. Tiene 14 pines de entrada / salida digital (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un resonador cerámico de 16 MHz (CSTCE16M0V53-R0), una conexión USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio.

Contiene todo lo necesario para soportar el microcontrolador; simplemente conéctelo a una computadora con un cable USB o enciéndalo con un adaptador de CA a CC o una batería para comenzar. Puede jugar con su Uno sin preocuparse demasiado por hacer algo mal, en el peor de los casos, puede reemplazar el chip por unos pocos dólares y empezar de nuevo (ARDUINO, 2021).

"Uno" significa uno en italiano y fue elegido para marcar el lanzamiento de Arduino Software (IDE) 1.0. La placa Uno y la versión 1.0 del software Arduino (IDE) fueron las versiones de referencia de Arduino, ahora evolucionadas a versiones más recientes.

La placa Uno es la primera de una serie de placas USB Arduino y el modelo de referencia para la plataforma Arduino; Para obtener una lista extensa de placas actuales, pasadas o desactualizadas, consulte el índice de placas Arduino (ARDUINO, 2021).

2.1.2.1 Especificaciones técnicas

Tabla 1. Especificaciones técnicas Arduino UNO R3

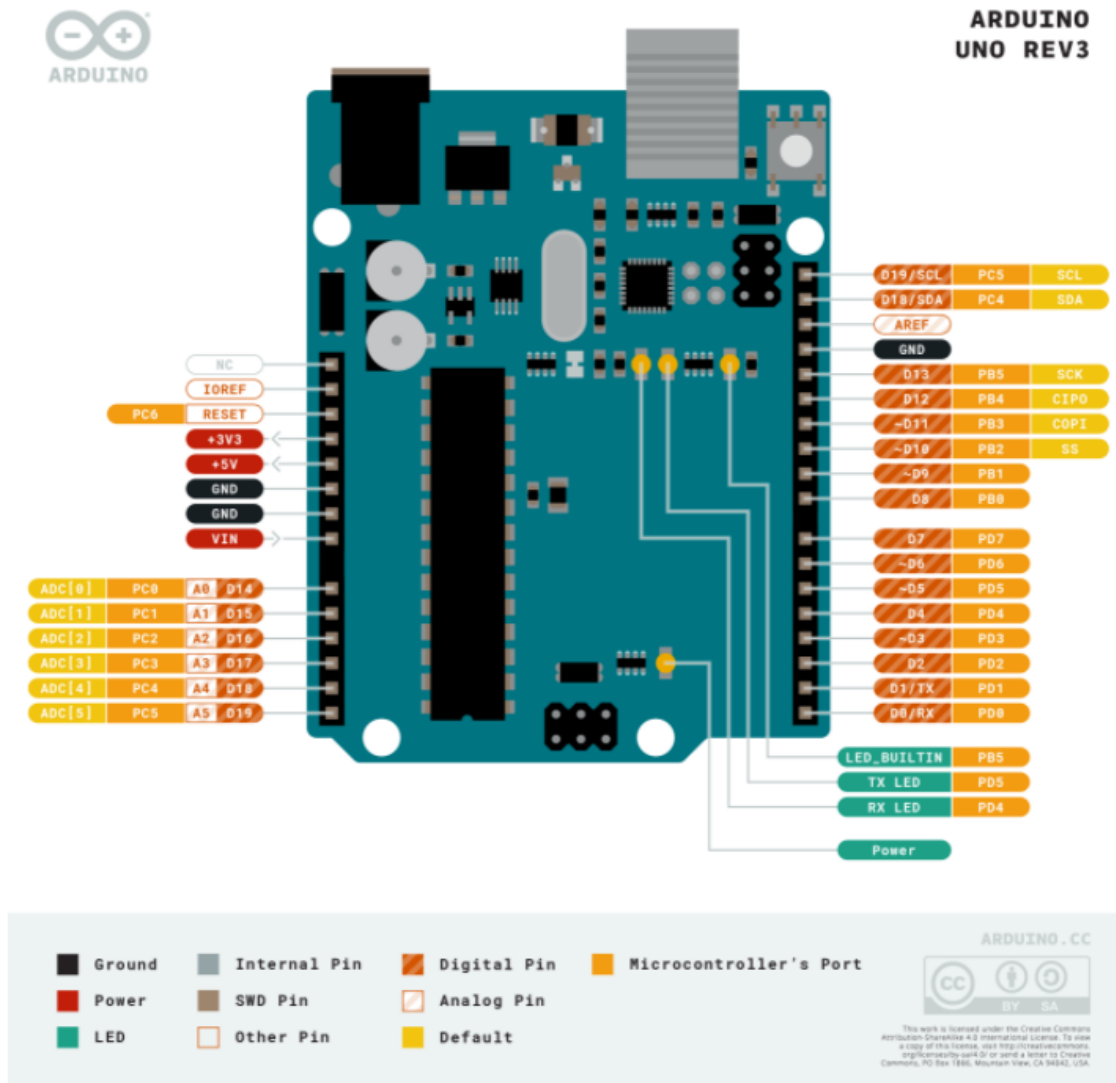
Microcontrolador	ATmega328P
Voltaje de funcionamiento	5 V
Voltaje de entrada (recomendado)	7-12 V
Voltaje de entrada (límite)	6-20 V
Pines de E / S digitales	14 (de los cuales 6 proporcionan salida PWM)
Pines de E / S digitales PWM	6
Pines de entrada analógica	6
Corriente CC por pin de E / S	20 mA
Corriente CC para pin de 3.3V	50 mA
Memoria Flash	32 KB (ATmega328P) de los cuales 0,5 KB utilizados por el gestor de arranque

SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz
LED_BUILTIN	13
Longitud	68,6 mm
Ancho	53,4 mm
Peso	25 g

Fuente: (ARDUINO, 2021)

2.1.2.2 Diagrama de pines

Figura 8. Diagrama de pines



Fuente: (ARDUINO, 2021)

2.1.3 GSM/GPRS/GPS SIM808 Shield

Es una tecnología de navegación integrada de cuatro bandas GSM / GPRS y GPS Arduino con escudos de expansión. Solo el tamaño de una tarjeta de crédito, de acuerdo con el empaque estándar del pin Arduino, compatible con Arduino UNO, arduino Leonardo, arduino Mega y otras placas base arduino. En comparación con la generación anterior SIM908, SIM808 realizó algunas mejoras en el rendimiento y la estabilidad. Además de

las funciones normales de SMS y teléfono, el escudo también admite MMS, DTMF, FTP y otras funciones. Puede lograr la adquisición de datos, el transceptor de datos inalámbrico, la aplicación IoT y la orientación GPS. Debería integrar un micrófono integrado y un conector para auriculares, lo que ahorra costos y hace que su proyecto sea fácil. también se puede conectar directamente a la antena GSM y GPS mediante un conector de antena externo.

SIM808 GPS / GPRS / GSM Arduino Shield V1.0 utiliza la última versión del módulo Simcom SIM808, en comparación con el primer módulo SIM808 disponible en el mercado, el nuevo módulo tiene mejor estabilidad. (DFROBOT, 2021)

2.1.3.1 Especificaciones

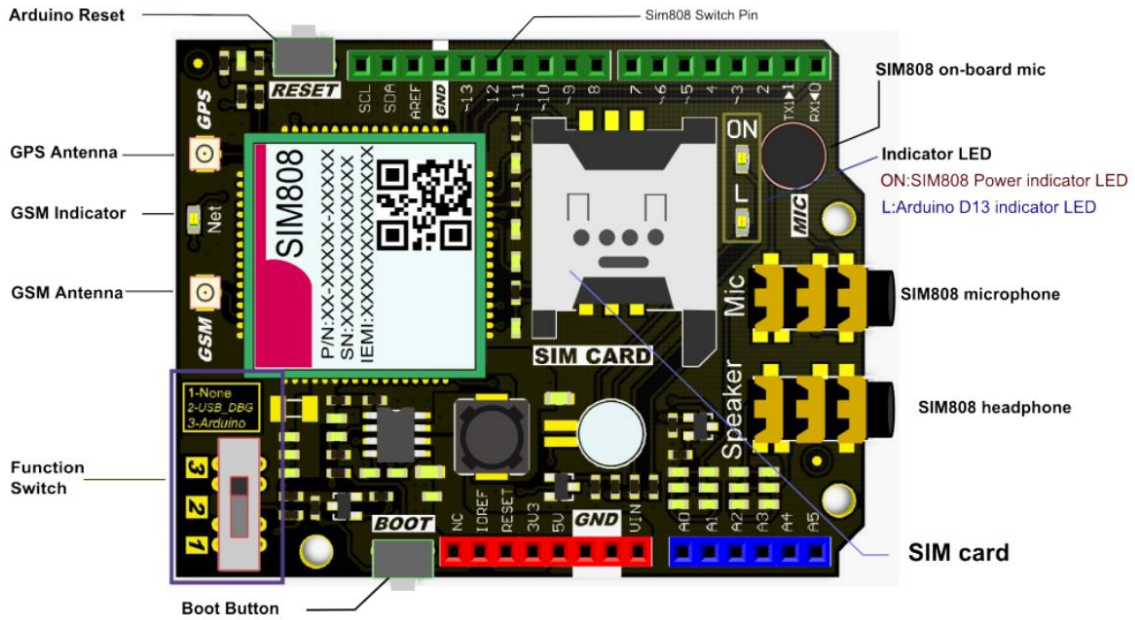
Tabla 2. Especificaciones GSM/GPRS/GPS SIM808 Shield

Voltaje de funcionamiento	5 V
Potencia de entrada	7-23 V
Quad-band	850/900/1800/1900 MHz
Clase de ranuras múltiples GPRS 12/10x	
Estación móvil GPRS clase B	
Cumplir con GSM fase 2/2 +	Clase 4 (2 W @ 850 / 900MHz) Clase 1 (1 W a 1800/1900 MHz)
Admite modo de bajo consumo de energía	Modo 100mA @ 7V-GSM
Admite control de comando AT (3GPP TS 27.007,27.005 y comandos AT mejorados de SIMCOM)	
Admite tecnología de navegación por satélite GPS	
Indicador de estado LED de soporte	Estado de la fuente de alimentación, estado de la red y modos de funcionamiento
Entorno de trabajo	-40 °C ~ 85 °C
Tamaño	69 * 54 mm / 2,71 * 2,12 pulgadas

Fuente: (DFROBOT, 2021)

2.1.3.2 Descripción general de la placa

Figura 9. Descripción general placa SIM808



Fuente: (DFROBOT, 2021)

2.1.4 Mini Pulsador SPST

Pequeño botón pulsador sin enclave normalmente abierto, con diámetro de 7mm para montura. (Electronics, 2021)

Figura 10. Mini pulsador rojo



Fuente: (Electronics, 2021)

2.1.4.1 Características

Tabla 3. Características Mini Pulsador SPST

Tipo	Sin enclave.
Estructura de contacto	Normalmente abierto (rojo) y Normalmente cerrado (negro).
Voltaje máximo	125V.
Corriente máxima	3A.
Diámetro	7mm.
Número de pines	2.
Vida útil	Aproximadamente 10,000 operaciones mecánicas.
Modelo	PBS-110

Fuente: (Electronics, 2021)

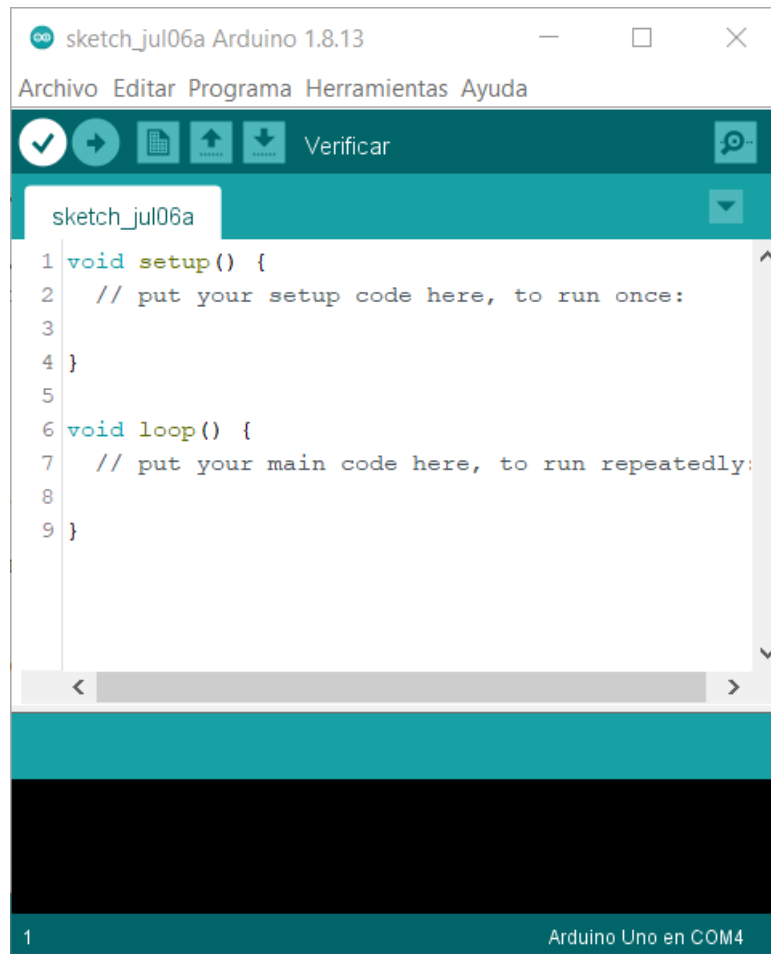
2.2 SOFTWARE

2.2.1 SW ARDUINO

El software de Arduino es un IDE, entorno de desarrollo integrado (siglas en inglés de Integrated Development Environment). Es un programa informático compuesto por un conjunto de herramientas de programación.

El IDE de Arduino es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además, incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware. (Aprendiendo Arduino, 2017)

Figura 11. Entorno del SW Arduino



Fuente: Realizado por el Autor.

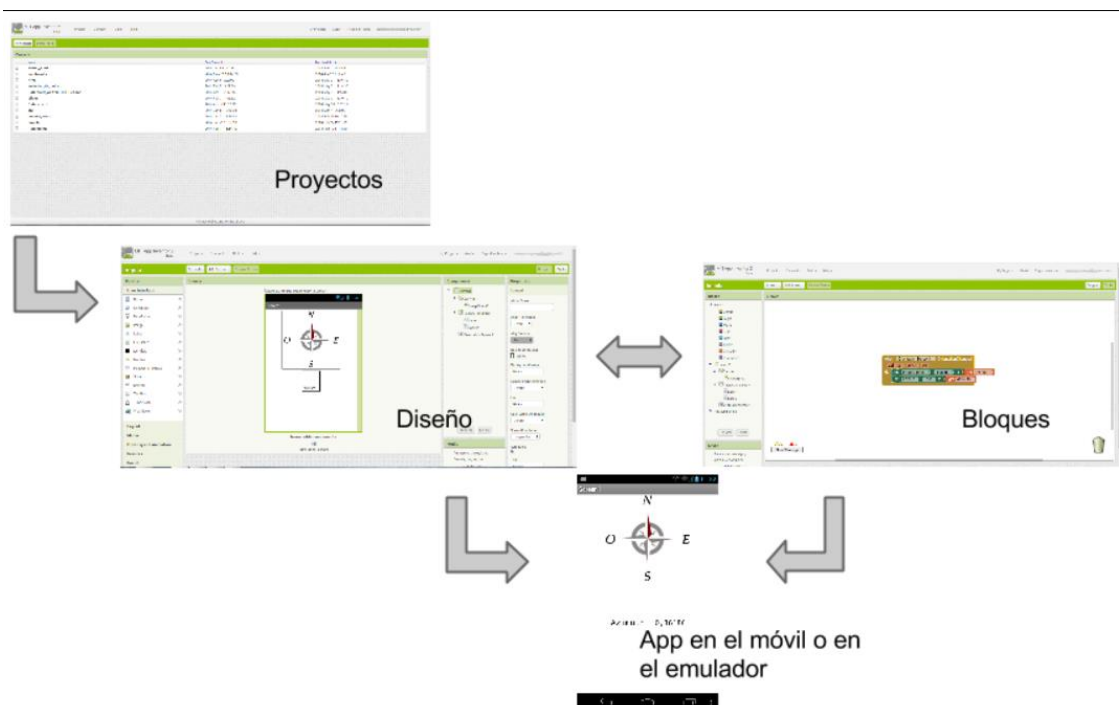
2.2.2 MIT APP INVENTOR

App Inventor es un entorno de desarrollo de aplicaciones para dispositivos Android. Para desarrollar aplicaciones con App Inventor sólo necesitas un navegador web y un teléfono o Tablet Android (si no lo tienes podrás probar tus aplicaciones en un emulador). App Inventor se basa en un servicio web que te permitirá almacenar tu trabajo y te ayudará a realizar un seguimiento de sus proyectos.

Se trata de una herramienta de desarrollo visual muy fácil de usar, con la que incluso los no programadores podrán desarrollar sus aplicaciones.

Al construir las aplicaciones para Android trabajarás con dos herramientas: App Inventor Designer y App Inventor Blocks Editor. En Designer construirás el Interfaz de Usuario, eligiendo y situando los elementos con los que interactuará el usuario y los componentes que utilizará la aplicación. En el Blocks Editor definirás el comportamiento de los componentes de tu aplicación. (Ricoy Riego, 2020)

Figura 12. Interfaces de App inventor



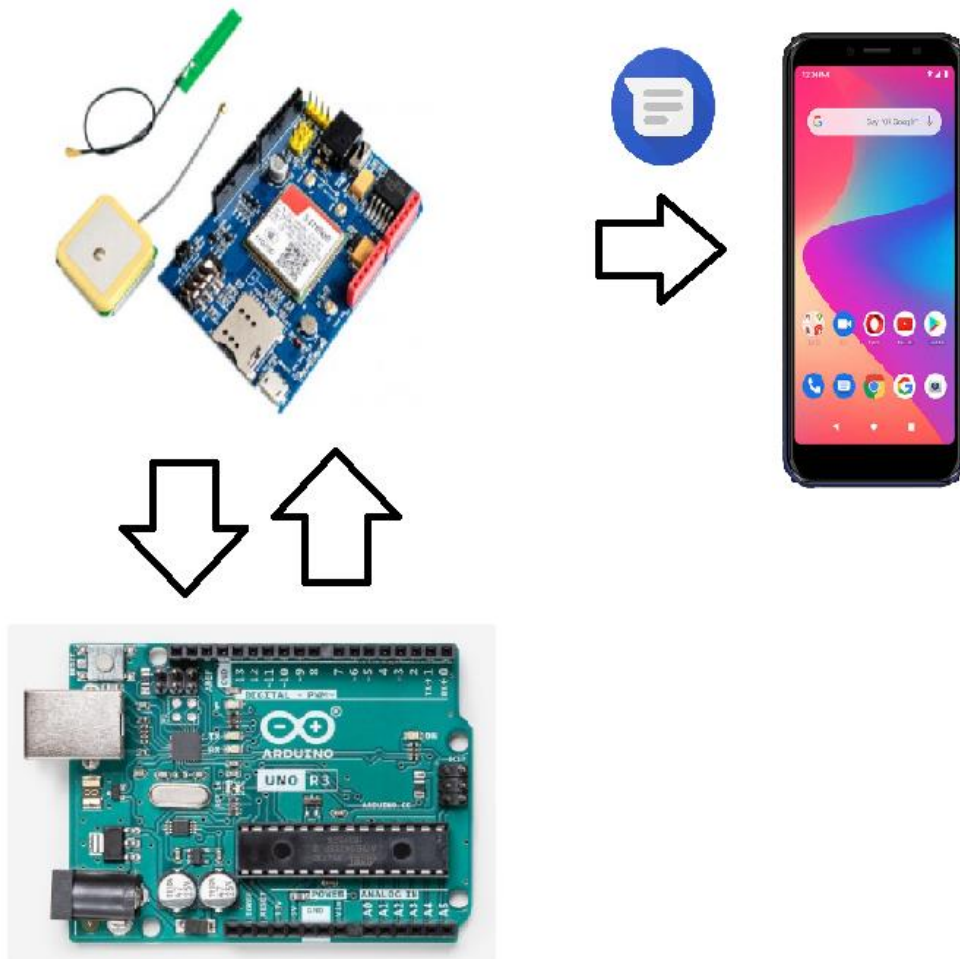
Fuente: (Ricoy Riego, 2020).

CAPÍTULO 3

3. DISEÑO E IMPLEMENTACIÓN

El diseño es sencillo, al ensamblar directamente el GSM/GPRS/GPS SIM808 Shield sobre la placa Arduino R3, de tal manera que se encajen los pines sin dejar ninguno fuera, esto también optimiza el espacio a utilizar ya que el objetivo final de este es implementarlo dentro de un automóvil.

Figura 13. Componentes



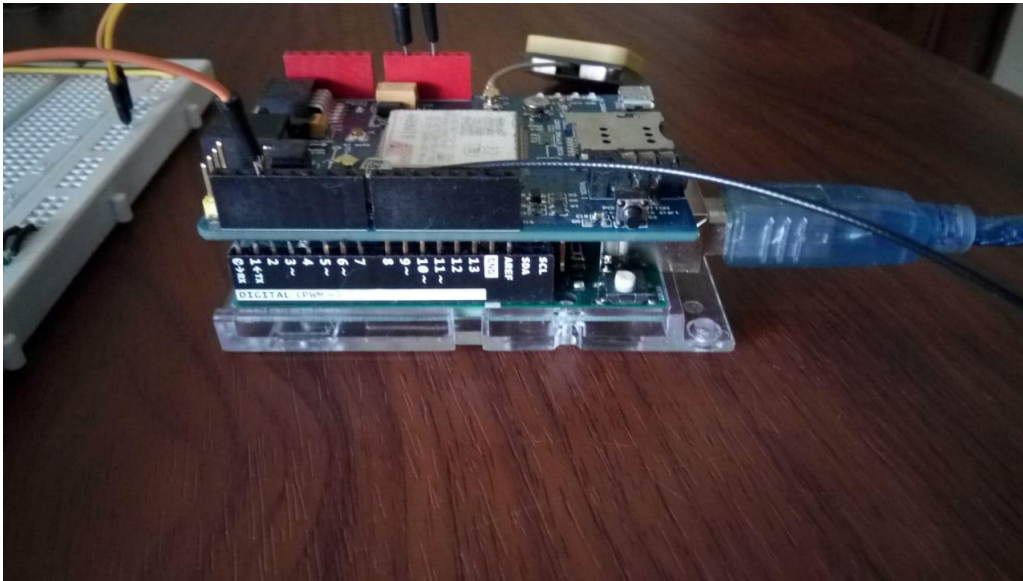
Fuente: Realizado por el Autor.

El funcionamiento esperado de los elementos, como se indica en la Figura 6, es que el SIM808 y el Arduino trabajen conjuntamente para recibir los datos de GPS y luego enviarlos vía SMS al dispositivo móvil.

3.1 Construcción del prototipo

Como se mencionó anteriormente, se optó por un ensamblaje directo para optimizar el espacio de manera modular.

Figura 14. Vista lateral izquierda del circuito ensamblado



Fuente: Realizado por el Autor.

Figura 15. Vista lateral derecha del circuito ensamblado



Fuente: Realizado por el Autor.

Figura 16. Vista superior del circuito ensamblado



Fuente: Realizado por el Autor.

Luego se procedió a comprobar la conexión de ambos a través del programa Arduino y comando AT.

3.1.1 Comandos AT

Para comprobar el funcionamiento de ambas placas se usó primero el siguiente programa (Macho, 2021):

Figura 17. Comunicación serie



```
Comunicacion_serie Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

Comunicacion_serie
1 #include <SoftwareSerial.h>
2 SoftwareSerial SIM808(7, 8); //Seleccionamos los pines 7 como Rx y 8 como Tx
3
4 void setup()
5 {
6   SIM808.begin(19200);
7   Serial.begin(19200);
8   delay(100);
9 }
10
11 void loop()
12 {
13   //Enviamos y recibimos datos
14   if (Serial.available() > 0)
15     SIM808.write(Serial.read());
16   if (SIM808.available() > 0)
17     Serial.write(SIM808.read());
18 }
```

Fuente: (Uriarte, 2019)

El código de la Figura 10 es la base para la lectura de los comandos AT.

Luego en el monitor serie se comprueba su funcionamiento con el comando “AT”, lo que nos dará como respuesta un “OK”, si todo está funcionando correctamente.

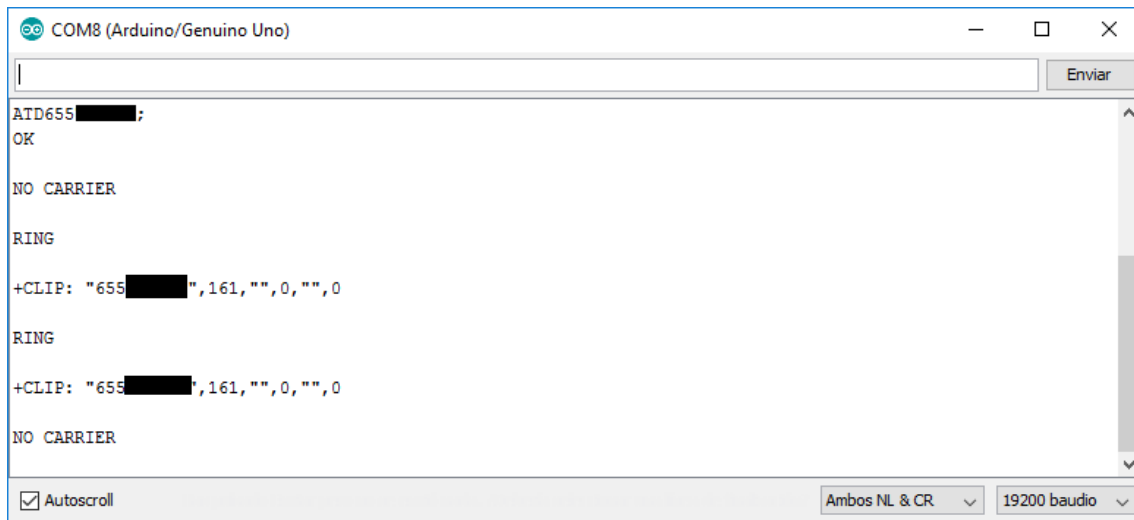
Figura 18. Funcionamiento a través del comando AT



Fuente: (Uriarte, 2019)

Luego se procedió a comprobar el funcionamiento de los SMS y las llamadas.

Figura 19. Llamada telefónica a través de comandos AT



Fuente: (Uriarte, 2019)

3.2 Diseño del programa de arduino

Para el programa en arduino se dividió en tres partes, primero la parte de inicio donde se define la velocidad del monitor serie, convierte el módulo en modo mensajes de texto (recibir y enviar SMS) y activa el GPS (Figura 13).

Figura 20. Inicio del programa Arduino

```
void inicio() { //inicializacion
  SIM808.println("AT+IPR=19200");
  delay(500);
  SIM808.println("AT+CMGF=1");
  delay(500);
  SIM808.println("AT+CMGR=?");
  delay(500);
  SIM808.println("AT+CNMI=2,2,0,0,0");
  delay(500);
  SIM808.println("AT+CGNSPWR=1");
  delay(500);
  while (SIM808.available() > 0)
    Serial.write(SIM808.read());
}
```

Fuente: Realizado por el Autor.

La segunda parte es la adquisición de datos, donde antes de enviar un nuevo mensaje vuelve a tomar y guardar la posición (Figura 14).

Figura 21. Adquisición de datos del módulo SIM808

```
while (cont2==1) {
  dato = "";
  SIM808.println("AT+CGPSINF=0"); //solicitud de posicion
```

Fuente: Realizado por el Autor.

La tercera y última es la parte del envío, la cual en estos momentos esta comentada para evitar el gasto incensario de saldo, lo que hace es leer los números guardados previamente y dependiendo del mismo se repite en un bucle for con todos los números (Figura 15).

Figura 22. Envío de mensajes texto

```
for (int n = 0; n < Cantidad_Movil; n++) { //for para enviar n mensajes
  SIM808.print("AT+CMGF=1\r");
  delay(1000);
  SIM808.println("AT+CMGS=\"" + Num_Movil[n] + "\""); // Envia mensaje
  delay(100);
  SIM808.println(dato);
  delay(100);
  SIM808.println((char)26);
  delay(20000);
  SIM808.println();
  Serial.println();
  Serial.print("Mensaje numero: ");
  Serial.print(n+1);
  Serial.print(" enviado");
}
```

Fuente: Realizado por el Autor.

También cabe mencionar la parte de la llamada, la cual está ubicada fuera de este bucle.

Figura 23. Comando AT para llamadas.

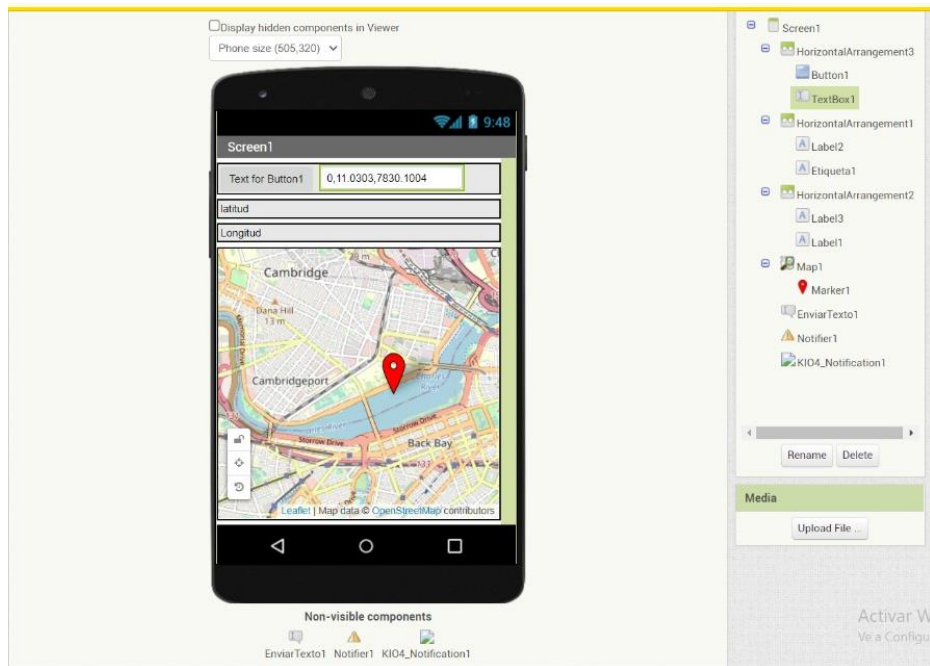
```
//llamada
delay(10000);
SIM808.print("AT+CMGF=1\r");
SIM808.println("ATD0987997454;");
```

Fuente: Realizado por el Autor.

3.3 Diseño de la aplicación móvil

La aplicación se diseñó en MIT App Inventor, esta cuenta con etiquetas, botones, cajas de texto, también módulos de mapa y enviar texto. Este diseño es preliminar para las pruebas.

Figura 24. Diseño de la aplicación en MIT App Inventor.



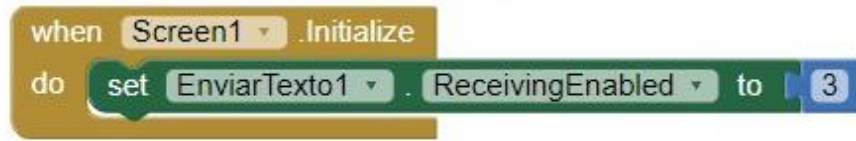
Fuente: Realizado por el Autor.

La finalidad de este diseño es separar las pruebas en la aplicación, de manera que se pueda comprobar que los datos de coordenadas se muestren en el mapa.

En la parte de bloques, tanto para las pruebas como para el funcionamiento normal, se divide en dos partes.

Primero permite el acceso de la aplican a el envío y recepción de mensaje SMS

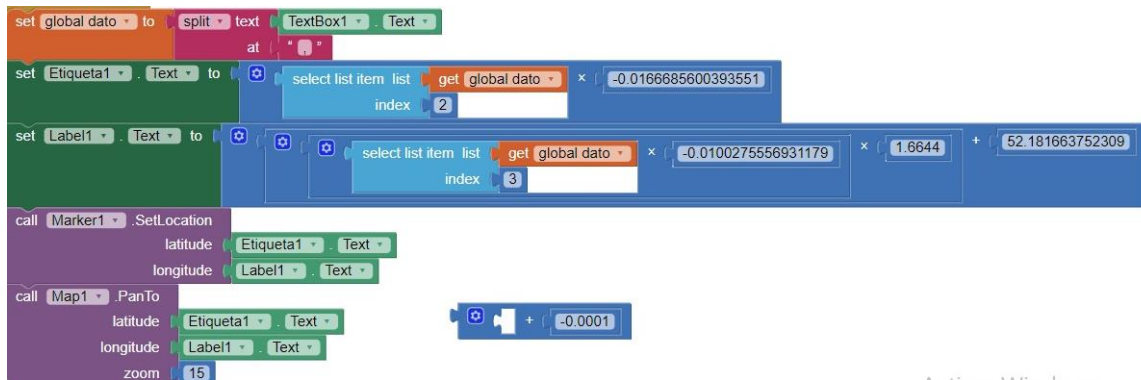
Figura 25. Inicialización para SMS.



Fuente: Realizado por el Autor.

La segunda es el almacenamiento, transformación y uso de la información recibida del mensaje de texto en la aplicación.

Figura 26. Proceso de uso de datos recibidos.



Fuente: Realizado por el Autor.

CAPÍTULO 4

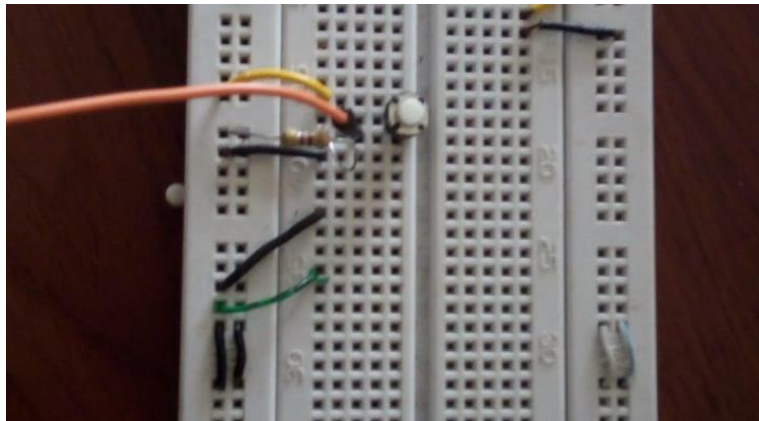
4. PRUEBAS Y RESULTADOS

Una vez armado se procedió a realizar las correspondientes pruebas tanto del hardware como el software, primero se comprobó el funcionamiento del envío de mensajes de texto y llamadas.

4.1 Pruebas del dispositivo

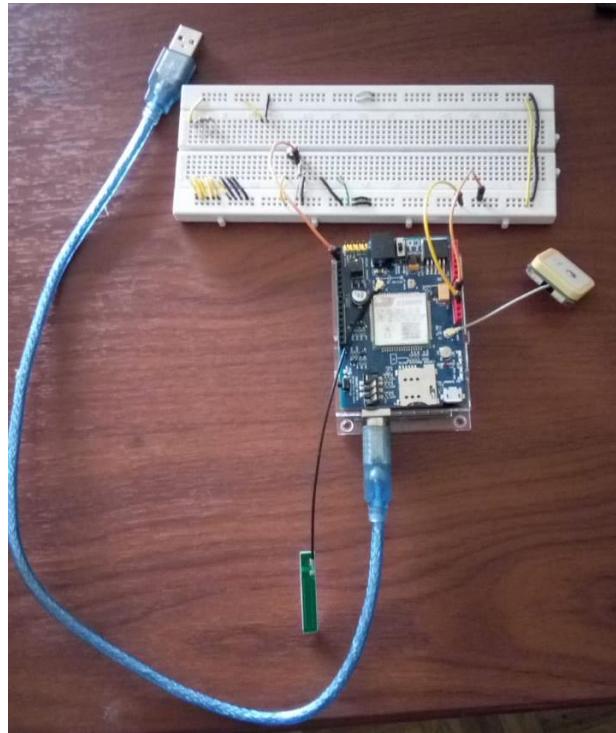
Para ello, se implementó en el dispositivo junto a un pulsador en un protoboard, para simular el momento de presionar el botón de pánico y enviar el mensaje.

Figura 27. Pulsador y protoboard.



Fuente: Realizado por el Autor.

Figura 28. Dispositivo de pruebas.

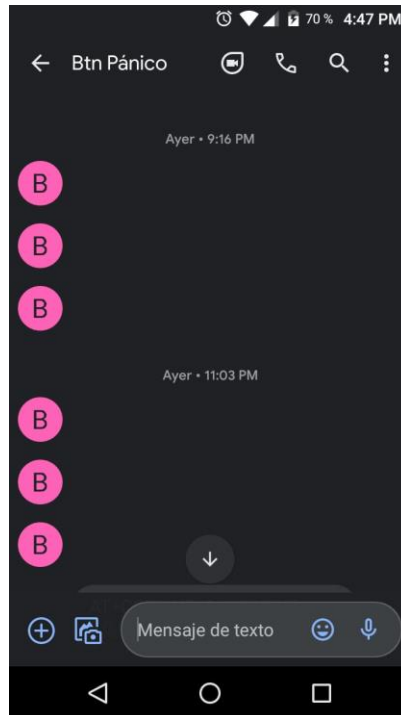


Fuente: Realizado por el Autor.

En la Figura 21 se puede apreciar el prototipo listo para sus pruebas, simulando el momento de pulsar el botón de pánico iniciando todo el proceso por parte del Arduino y el módulo SIM808.

Una vez revisado el funcionamiento del pulsador se procedió a enviar mensajes de texto a un celular, el contenido de este mensaje es la respuesta del comando AT “AT+CGPSINF=0” lo cual nos da el resumen de coordenadas GPS.

Figura 29. Prueba de recepción de SMS

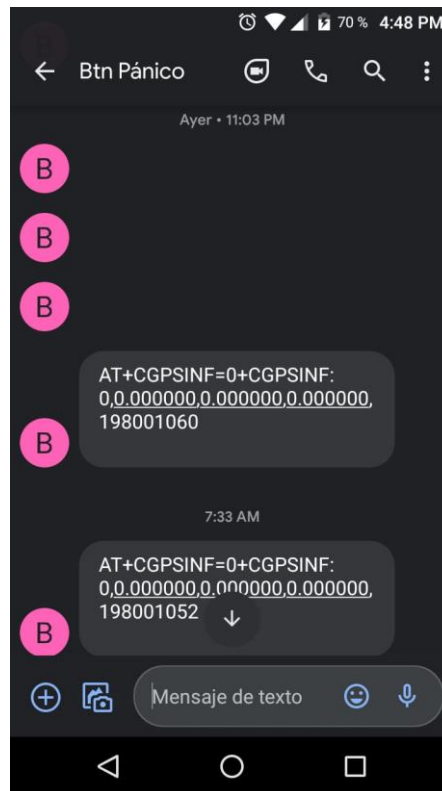


Fuente: Realizado por el Autor.

Como se observa en la Figura 22, se muestra inconvenientes, aunque se enviaba el mensaje, este llegaba vacío.

Luego se reconfiguro el código del Arduino, dando como resultado lo siguiente:

Figura 30. Prueba de recepción de SMS.



Fuente: Realizado por el Autor.

En la Figura 23 se muestra como ya están llegando datos desde el módulo, estos se muestran en “ceros” porque aún no tiene señal el GPS, y hasta que esto suceda seguirá mandando datos en blanco.

Figura 31. Código para recolección de datos.

```
void loop()
{
  mensaje = "";
  if(digitalRead(boton)==1 && bloqueo == 0){
    Serial.println("boton");
    SIM808.println("AT+CGPSINF=0");
    bloqueo = 1;
    delay(200);
    if(SIM808.available() > 0){
      while(SIM808.available() > 0 ){
        a= (char)SIM808.read();
        if(a!='\n' && a!='\r')mensaje += a;
      }
      Serial.print("aquiii:..");
      Serial.println(mensaje);
      enviar_mensaje(mensaje);
    }
    Serial.println("fin boton");
  }
  if(digitalRead(boton)==0)bloqueo = 0;

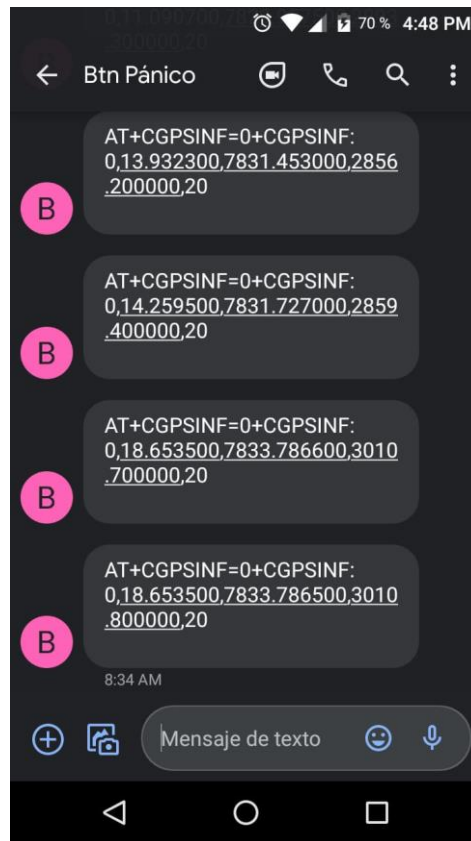
  if(SIM808.available() > 0){
    Serial.write(SIM808.read());
  }
}
```

Fuente: Realizado por el Autor.

En la Figura 24 está el código donde se almacena la información del SIM808 dentro de una variable interna del programa Arduino, donde se implementa un “bloqueo” para que al momento de presionar el botón solo llegue un pulso al programa.

Al final, cuando el módulo GPS pudo establecer una conexión estable y los datos aparecieron de manera correcta, como se muestra en la Figura 25.

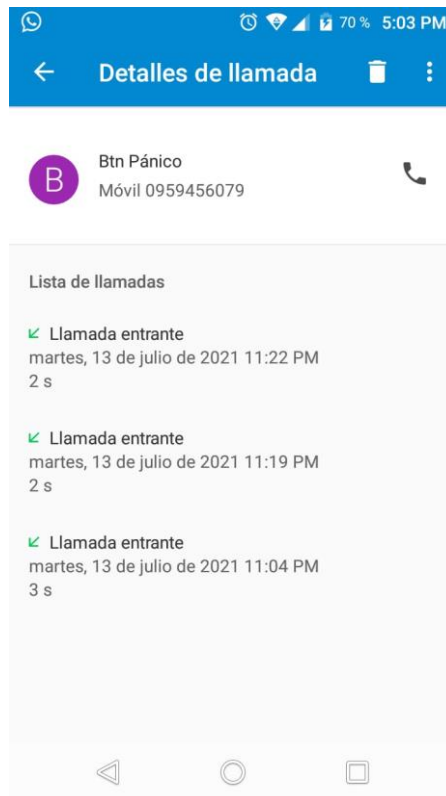
Figura 32. Prueba de recepción de SMS.



Fuente: Realizado por el Autor.

Para las llamadas de igual manera se tuvo inconvenientes, principalmente con la señal y los obstáculos (paredes, techos), pero termino funcionando correctamente.

Figura 33. Prueba de recepción de llamadas.



Fuente: Realizado por el Autor.

4.2 Pruebas de la aplicación

Antes de las pruebas de ubicación, fue necesario realizar una calibración de los datos de GPS, ya que los recibidos por el módulo era de tipo geográficos (SW), por lo que se hizo una transformación.

Tabla 4. Interpretación de datos recibidos.

Datos del modulo	Interpretación
11,0303	0°11'03.03"S
7830,1004	78°30'10.04"W

Fuente: Realizado por el Autor.

Para esta transformación se hizo una división entre los datos llegados del módulo y los datos adquiridos por Google maps.

Figura 34. Coordenadas de Google maps y del dispositivo SIM808.

	A	B	C	D	E	F
1	maps		arduino		Calculo N.1	
2	-0,183974	-78,501608	11,0303	7830,1004	-0,0166789661205951	-0,010025619594865
3	-0,245168	-78,533911	14,7134	7832,0384	-0,0166629059224924	-0,010027263272867
4	-0,2588	-78,530025	15,5308	7831,7986	-0,0166636618847709	-0,010027074112963
5	-0,263349	-78,531753	15,8	7831,8997	-0,0166676582278481	-0,010027165312140
6	-0,268772	-78,533073	16,1229	7831,981	-0,0166702020108045	-0,010027229764730
7	-0,272606	-78,538085	16,3542	7832,28	-0,0166688679360654	-0,010027486887599
8	-0,278928	-78,54286	16,7361	7832,5709	-0,0166662484091276	-0,010027724102695
9	-0,282283	-78,543553	16,9368	7832,6141	-0,0166668437957583	-0,010027757271994
10	-0,297458	-78,55013	17,8473	7833,0128	-0,0166668347593193	-0,010028086510978
11	-0,306875	-78,551873	18,4055	7833,1052	-0,0166730053516612	-0,010028190735904
12	-0,323843	-78,557878	19,4291	7833,4703	-0,0166679362399699	-0,010028489927383
13	-0,328636	-78,559632	19,7176	7833,5742	-0,0166671400170406	-0,010028580823298

Fuente: Realizado por el Autor.

Luego se hizo un promedio de los resultados para multiplicar este resultado con los datos recibidos por el módulo.

Tabla 5. Resultados de los cálculos.

-0,0166683558896211	0,0100275556931179
---------------------	--------------------

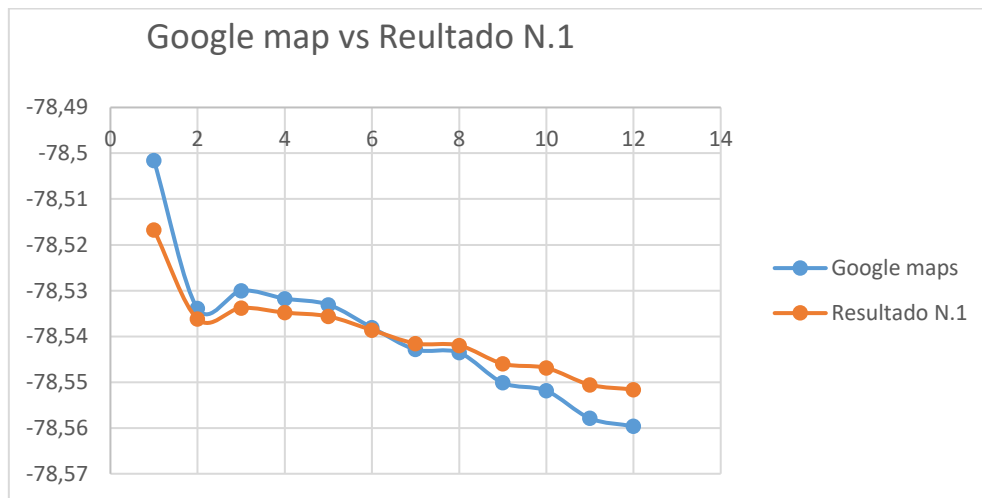
Fuente: Realizado por el Autor.

Figura 35. Resultados.

Resultado N.1	
-0,183856966	-78,5167678
-0,245248188	-78,5362012
-0,258872902	-78,5337966
-0,263360023	-78,5348104
-0,268742235	-78,5356257
-0,272597626	-78,5386239
-0,278963271	-78,5415409
-0,28230861	-78,5419741
-0,297485148	-78,5459721
-0,306789424	-78,5468986
-0,323851153	-78,5505597
-0,328659974	-78,5516016

Fuente: Realizado por el Autor.

Figura 36. Comparativa Coordenadas.

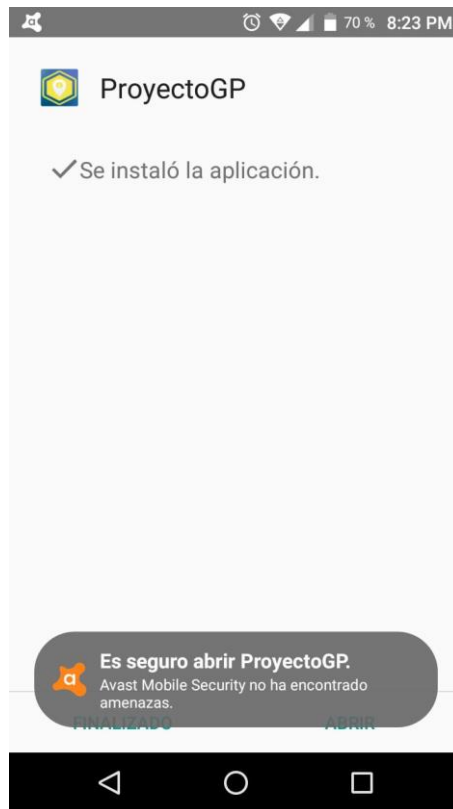


Fuente: Realizado por el Autor.

En la Figura 29 se distingue el error entre los cálculos del módulo y los datos que provienen de la aplicación Google maps.

Una vez culminado esto se procede a la instalación de la aplicación en un teléfono smartphone.

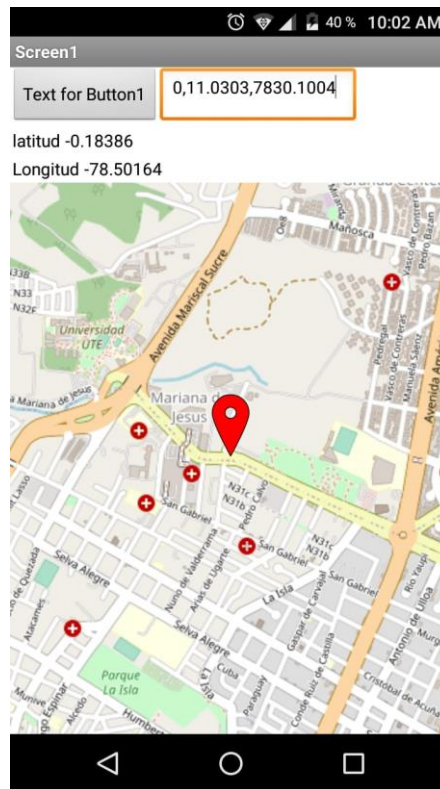
Figura 37. Instalación de la aplicación.



Fuente: Realizado por el Autor.

Luego se verifico el funcionamiento del mapa con coordinas ingresadas en la caja de texto, al momento de ingresarlas con un formato igual al que se va a recibir desde el módulo vía SMS, la aplicación opera de manera correcta ubicando en el mapa esta posición, comprobando el funcionamiento de la aplicación que se utilizara al final.

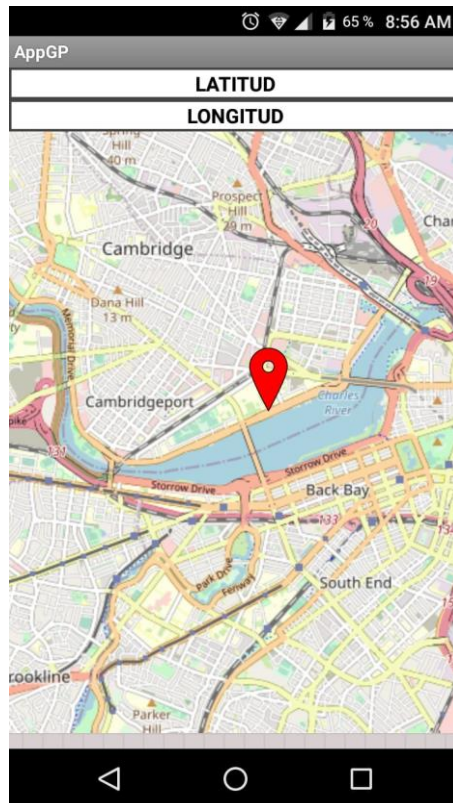
Figura 38. Pruebas de coordenadas.



Fuente: Realizado por el Autor.

Después se procede a instalar la aplicación de prueba de SMS, donde al momento de recibir este mensaje de texto se espera la ubicación automática de las coordenadas en el mapa.

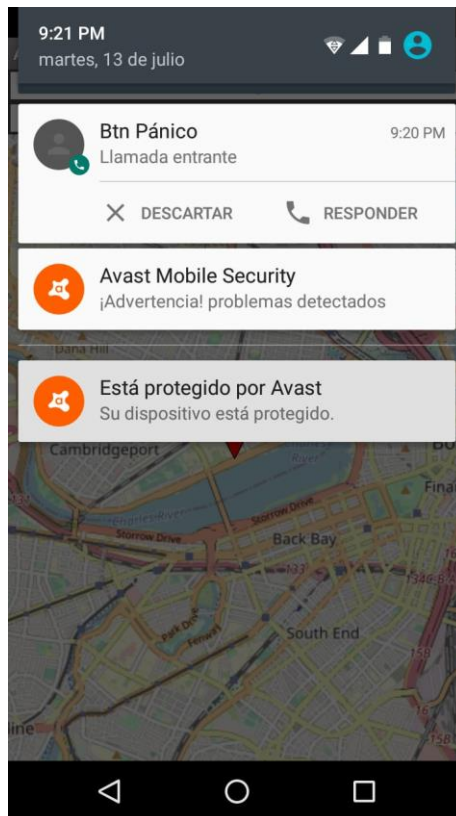
Figura 39. Pruebas de SMS.



Fuente: Realizado por el Autor.

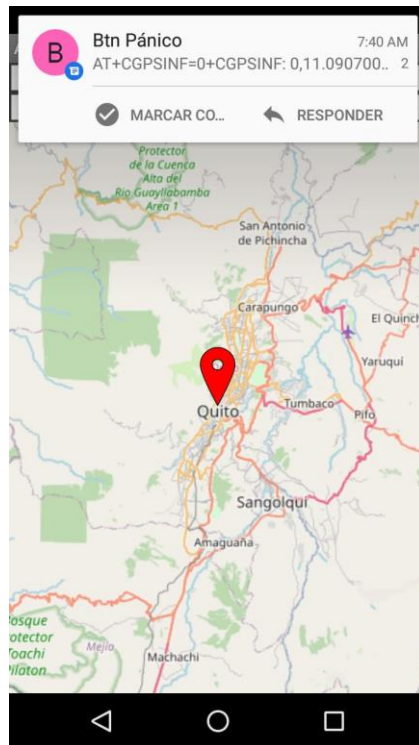
Una vez instalada se probó la recepción de mensajes y llamadas desde el dispositivo, el cual se denominó como Btn Pánico, después de recibir la llamada sigue con el mensaje de texto, el cual utilizara ara ubicar la posición en el mapa.

Figura 40. Llamada del botón de pánico.



Fuente: Realizado por el Autor.

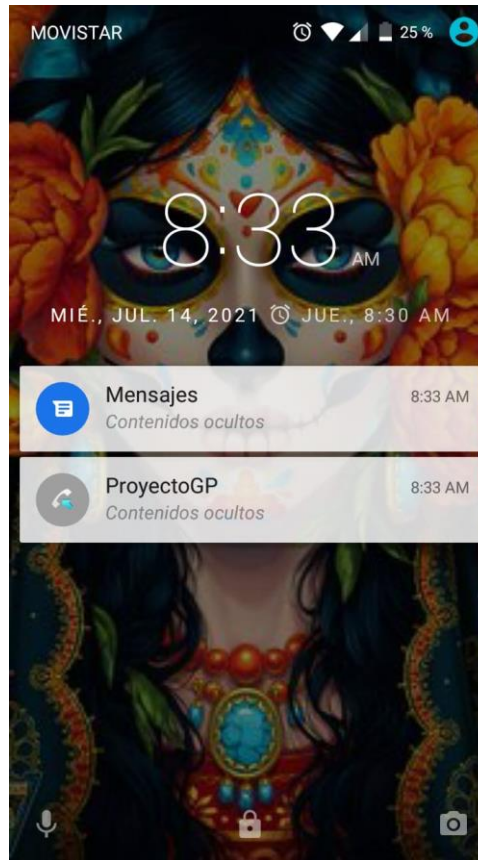
Figura 41. Mensaje del botón de pánico y ubicación automática de coordenadas.



Fuente: Realizado por el Autor.

Además, al momento de recibir un mensaje nuevo, aparte de la aplicación determinada, se reciben notificaciones por parte del aplicativo del proyecto.

Figura 42. Notificación de la aplicación.



Fuente: Realizado por el Autor.

4.3 Resultados

4.3.1 Implementación en el automóvil.

Primero se realizó la comprobación del funcionamiento del sistema en un automóvil bajo tres aspectos, primero la ubicación, la accesibilidad y el funcionamiento al conectarse con la batería del auto.

Figura 43. Ubicación de la placa de soporte del Arduino.



Fuente: Realizado por el Autor.

En la Figura 36 se observa la ubicación de la placa base del Arduino donde se ensamblará todo el dispositivo, procurando que no estorbe la palanca de cambios del automóvil.

Figura 44. Sistema alimentado desde la batería del automóvil.



Fuente: Realizado por el Autor.

Después se comprueba el encendido y posterior funcionamiento del sistema (Figura 37).

Figura 45. Botón de pánico ubicado en la palanca de cambios.



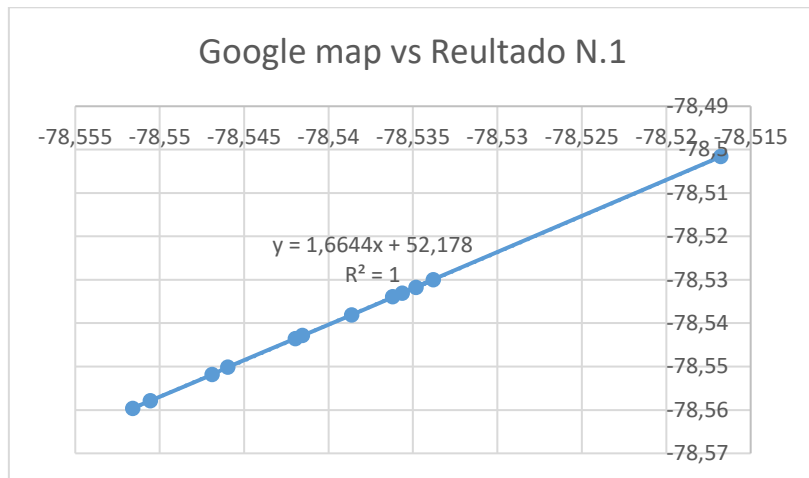
Fuente: Realizado por el Autor.

Por último, Figura 38, la ubicación del botón, el cual se encuentra en la parte baja del pomo de la palanca de cambios, sobresaliendo del guardapolvo, esta ubicación se determinó a través de una encuesta, posteriormente detallada.

4.3.2 Ajustes de coordenadas.

Como se observa en la Figura 29, hay cierto margen de error en las coordenadas, por lo que se realizó una ecuación lineal a través de un gráfico de dispersión (Figura 39) donde nos dio la fórmula para ajustar estos datos.

Figura 46. Gráfico de dispersión.



Fuente: Realizado por el Autor.

Usando esta fórmula se hizo el ajuste correspondiente, comprobando que los resultados sean más parecidos a las coordenadas de Google maps.

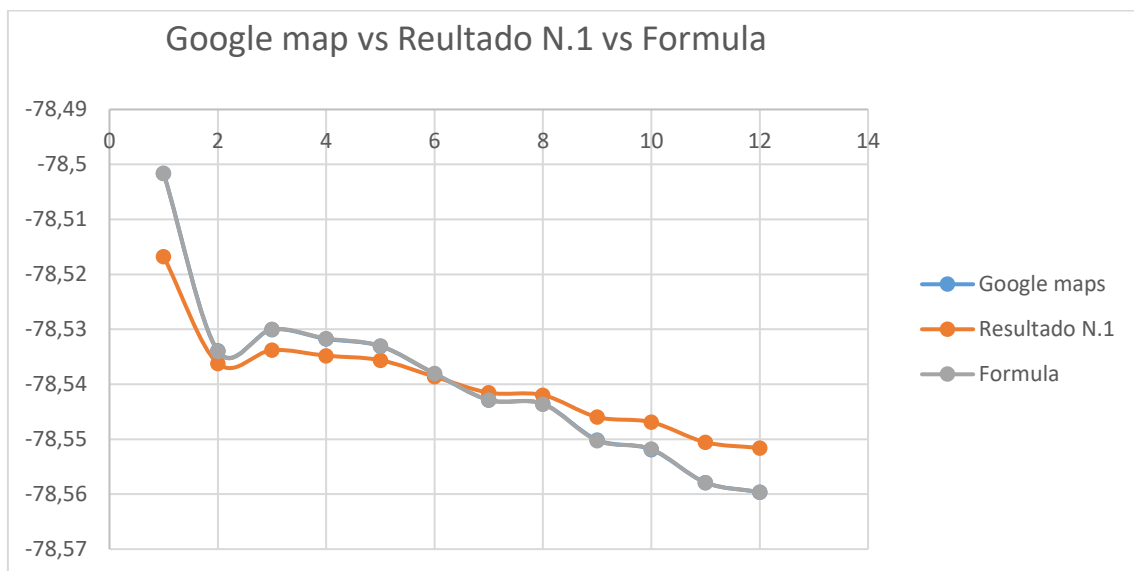
Figura 47. Datos aplicando la ecuación de ajuste.

Formula	
-0,18430601	-78,5016446
-0,24581378	-78,5339896
-0,25946436	-78,5299874
-0,26396	-78,5316747
-0,26935243	-78,5330316
-0,27321514	-78,5380219
-0,27959287	-78,542877
-0,28294456	-78,543598
-0,29814991	-78,5502522
-0,30747185	-78,5517943
-0,32456597	-78,5578878
-0,32938392	-78,5596219

Fuente: Realizado por el Autor.

Esta comparativa se visualiza de mejor manera en el siguiente mapa (Figura 41), donde se puede contrastar los resultados de las coordenadas.

Figura 48. Comparativa de los resultados.



Fuente: Realizado por el Autor.

4.3.3 Tiempo de recepción de mensajes de texto.

Al momento de realizar las pruebas se realizó la medición del tiempo de llegada del mensaje de texto desde el momento en que se pulso el botón de pánico, estos tiempos se tomaron desde el interior del automóvil en la ciudad de Quito a partir de las 8:00 am. La máxima variación de estos tiempos es de un segundo.

Figura 49. Tiempo de recepción de SMS.

Tiempo de recepcion del mensaje de texto (segundos)
1
1
2
1
1
2
2
1
1
2

Fuente: Realizado por el Autor.

Luego se hizo lo mismo, pero con dos teléfonos, se debe tomar en cuenta que en el código arduino hay un tiempo entre los procesos de envió de mensajes a múltiples números el cual afecta este resultado, además del de llamada (Figuras 43 y 44).

Figura 50. Tiempo de los mensajes de texto.

```
delay(1000);  
SIM808.println("AT+CMGS=\"" + Num_Movil[n] + "\""); // Envía mensaje  
delay(100);  
SIM808.println(dato);  
delay(100);  
SIM808.println((char)26);  
delay(20000);
```

Fuente: Realizado por el Autor.

Figura 51. Tiempo de las llamadas.

```
//llamada  
delay(10000);
```

Fuente: Realizado por el Autor.

Figura 52. Tiempos de envío de SMS a dos teléfonos.

Bucle	Tiempo de recepción del mensaje de texto (segundos)	
	Teléfono 1	Teléfono 2
1	31,2	31,2
2	32,2	52,4
3	54,4	74,6
4	75,6	95,8
5	96,8	117
6	119	139,2
7	141,2	161,4
8	162,4	182,6
9	183,6	203,8
10	205,8	226

Fuente: Realizado por el Autor.

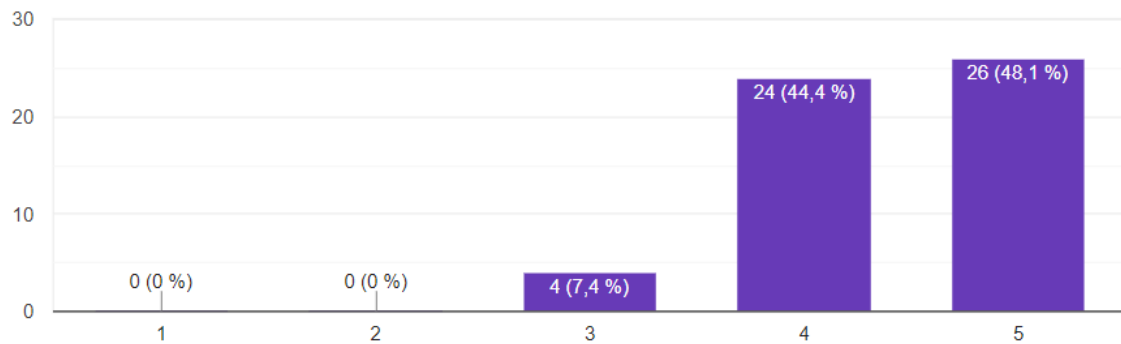
Los resultados, Figura 45, son obtenidos de la suma de los tiempos entre la primera llamada y los mensajes a dos teléfonos móviles, en caso de agregar un tercer número telefónico es necesario tener en cuenta que estos tiempos aumentarían.

4.4 Encuesta

Concluido con la construcción e implementación del sistema, se realizó una encuesta donde se determina la opinión de usuarios, tanto poseedores de automóviles como los que no los tienen, para determinar cuáles son las características que se podría mejorar en un futuro.

1. Diseño de la aplicación (1 a 5).

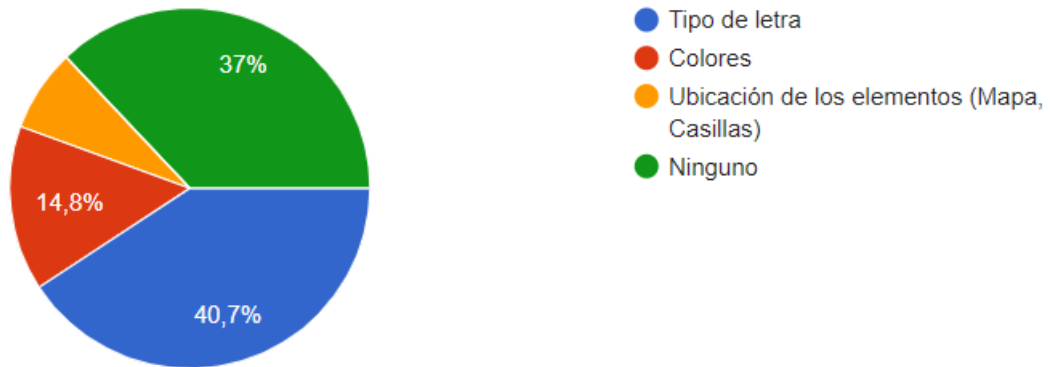
Figura 53. Resultados pregunta 1



Fuente: (Google Forms, 2021)

2. Cambios que se podrían realizar a la aplicación.

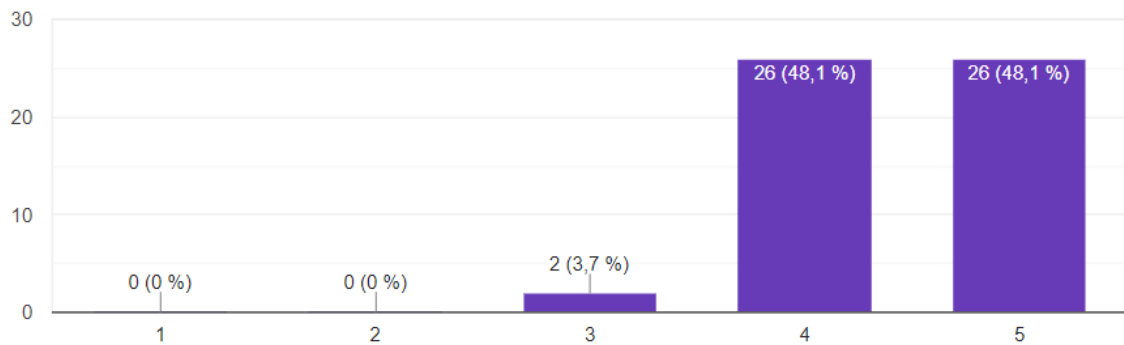
Figura 54. Resultados pregunta 2



Fuente: (Google Forms, 2021)

3. Descripción del funcionamiento de la aplicación (1 a 5).

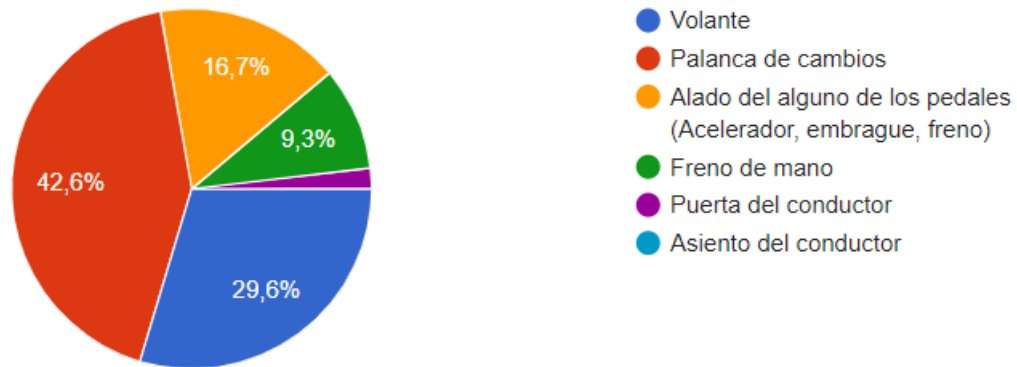
Figura 55. Resultados pregunta 3



Fuente: (Google Forms, 2021)

4. Ubicación aproximada del botón de pánico dentro del automóvil.

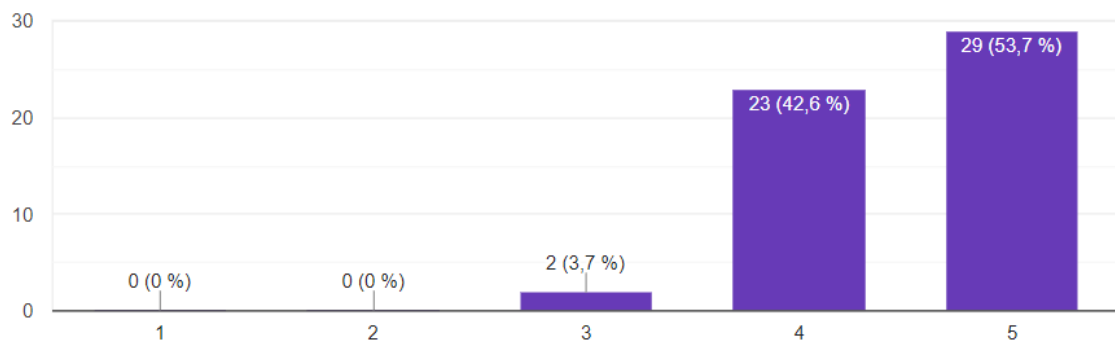
Figura 56. Resultados pregunta 4



Fuente: (Google Forms, 2021)

5. Descripción del funcionamiento del botón de pánico (1 a 5).

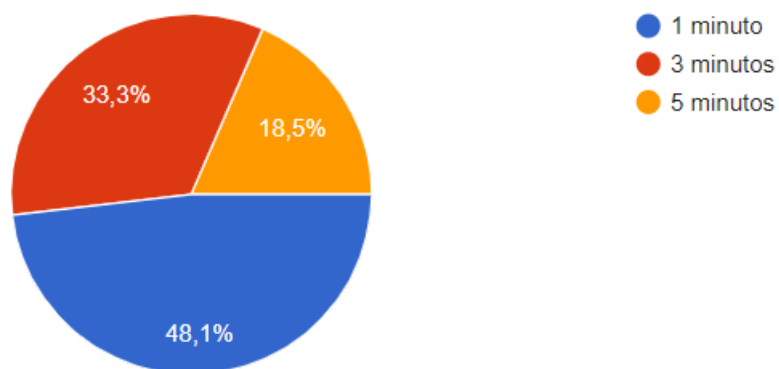
Figura 57. Resultados pregunta 5



Fuente: (Google Forms, 2021)

6. Intervalos de repetición del proceso de envió de mensajes.

Figura 58. Resultados pregunta 6.



Fuente: (Google Forms, 2021)

CONCLUSIONES

- Se realizó un proyecto donde el resultado final fue satisfactorio, tomando en cuenta los objetivos propuestos, se demostró el funcionamiento conjunto de los módulos SIM808 y Arduino con la aplicación móvil, que al momento de activar el botón se envían mensajes de texto y al recibirse estos en el dispositivo smartphone, con la aplicación abierta, ubica de manera automática las coordenadas en el mapa.
- Las dimensiones del módulo completo fueron óptimas para su instalación en áreas pequeñas dentro del automóvil, utilizando elementos que cumplieran con esta necesidad, ocupando poco espacio debajo de la cubierta de la palanca de cambios, como se ve en las Figuras 36 y 37, también el botón de pánico está ubicado con cierto grado de disimulación (Figura 38), ante la vista de cualquier persona que desconozca la existencia de dicho sistema.
- Para la aplicación móvil, se dejó de lado la intervención del usuario para que solo fuera una herramienta de observación de las coordenadas en un mapa, permitiendo solo acercar o alejar esta vista, se actualiza de manera automática al recibir un nuevo mensaje de texto con las características de los datos adquiridos por el módulo SIM808.
- Por último, en los resultados se mostró la funcionalidad completa del sistema, siendo esto puesto bajo el análisis de usuarios, tanto poseedores de vehículos como gente que no tiene automóviles, para que se califique la estética, ubicación y funcionalidad del sistema, usando esta información para una futura referencia de mejoras de este.

RECOMENDACIONES

- Realizar las pruebas de módulos SIM en espacios abiertos, ya que hay ciertas circunstancias, como el tiempo en el cual se establecerá la conexión con los satélites y las condiciones climáticas las cuales afectan al GPS y GSM disminuyendo su efectividad especialmente al momento de recibir datos.
- Al igual que otras aplicaciones y dispositivos similares, este prototipo se puede mejorar de distintas maneras, aumentando o disminuyendo elementos, por lo cual es recomendable partir desde una base con la cual se pueda reducir el tiempo de ensamblaje y usarlo en actualizaciones.

REFERENCIAS

TESIS

Cajas Malave, R., & Raymondi Lomas, W. (2018). *DISEÑO Y DESARROLLO DE UN PROTOTIPO DE ALERTA EN SITUACIONES DE EMERGENCIA*

UTILIZANDO HERRAMIENTAS DE GEOLOCALIZACION Y SENSORES

BIOMETRICOS. Obtenido de

<http://repositorio.ug.edu.ec/bitstream/redug/32880/1/B-CINT-PTG->

[N.316%20Cajas%20Malave%20Ronny%20Patricio%20.%20Raymondi%20Lomas%20William%20Rafael.pdf](http://repositorio.ug.edu.ec/bitstream/redug/32880/1/B-CINT-PTG-N.316%20Cajas%20Malave%20Ronny%20Patricio%20.%20Raymondi%20Lomas%20William%20Rafael.pdf)

CHAMBA SALAVARRÍA, C. A. (2017). *IMPLEMENTACIÓN DE UN PROTOTIPO*

DE SISTEMA DE. Obtenido de *IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE*:

<http://dspace.esPOCH.edu.ec/bitstream/123456789/6367/3/98T00136.pdf>

Guayaquil Gonzabay, C. (2018). *Sistema móvil vehicular para mejorar la seguridad a través de la red GSM*. Obtenido de

<http://dspace.utb.edu.ec/bitstream/handle/49000/4814/-PT-UTB-FAFI-SIST-00028.pdf?sequence=1&isAllowed=y>

Lima León, J., & Valdez Bermejo, C. (2019). *DISEÑO DE UN BOTÓN DE PÁNICO*

MÓVIL PARA ADULTOS MAYORES PARA EL MONITOREO A TRAVÉS DEL SISTEMA ECU 911. Obtenido de

<https://dspace.ups.edu.ec/bitstream/123456789/17180/1/UPS-CT008213.pdf>

Martínez Orozco, F., & Callejas Piñeros, J. (2016). *SISTEMA DE MONITOREO PARA*

MOTOCICLETAS CON TECNOLOGÍA ARDUINO Y ANDROID. Obtenido de

<https://repository.unad.edu.co/bitstream/handle/10596/7918/1110448165.pdf?sequence=3&isAllowed=y>

Salas Robles, L. (2017). *SISTEMA DE SEGURIDAD PARA AUTOMOVIL*

UTILIZANDO. Obtenido de

<http://erecursos.uacj.mx/bitstream/handle/20.500.11961/3005/Documento%20Final%20Final%2089373%20210513.pdf?sequence=1&isAllowed=y>

Sancho Chilet, H. (2017). *DESARROLLO DE UN SISTEMA DE LOCALIZACIÓN Y*

APLICACIÓN MÓVIL PARA VEHÍCULOS EN APARCAMIENTOS. Obtenido

de

https://riunet.upv.es/bitstream/handle/10251/88241/73655831_TFG_15046301276554283957125751135335.pdf?sequence=2

PÁGINAS WEB

Aprendiendo Arduino. (2017). Obtenido de

<https://aprendiendoarduino.wordpress.com/tag/software/>

ARDUINO. (2021). Obtenido de <https://www.arduino.cc/en/Guide/Introduction>

ARDUINO. (2021). Obtenido de <https://store.arduino.cc/usa/arduino-uno-rev3>

Bertoletti, P. (2019). *Proyectos com ESP32 y LoRa*. NCB.

DFROBOT. (2021). Obtenido de

https://wiki.dfrobot.com/SIM808_GPS_GPRS_GSM_Shield_SKU__TEL0097

El Comercio. (2021). *El Comercio*. Obtenido de El Comercio:

<https://www.elcomercio.com/tag/secuestro-expres>.

El País. (2016). *El País*. Obtenido de El País:

https://elpais.com/internacional/2016/08/17/actualidad/1471453528_996707.htm

1.

Electronics, A. (2021). Obtenido de <https://avelectronics.cc/producto/mini-pulsador-spst/>

Flores, C. (2019). *La evolución de la tecnología GPS*. Obtenido de UBITEC:

<https://ubitec.mx/la-evolucion-de-la-tecnologia-gps/>

García Córdoba, F. (2005). *El cuestionario: recomendaciones metodológicas para el diseño de cuestionarios*. Limusa.

Google Forms. (2021). *Google Forms*. Obtenido de Google Forms:

<https://forms.gle/fiQMUaaosATYb3Em9>

Guimerá Orozco, A. (2018). *Iniciación a Android en Kotlin. Casos prácticos*. Paraninfo, S.A.

Huidobro Moya, J. (2016). *Comunicaciones Móviles. Sistemas GSM, UMTS y LTE*. RA-MA.

Letham, L. (2001). *GPS fácil. Uso del sistema de posicionamiento global*. Paidotribo.

Lissardy, G. (2019). Por qué América Latina es la región más violenta del mundo. *BBC News Mundo*.

Macho, J. (2021). *Prometec*. Obtenido de Prometec: <https://www.prometec.net/sim808/>

Malachovska, M. (2021). *Historia de los sistemas de alerta temprana y notificación de emergencias*. Obtenido de Telegrafía:

<http://www.sirenaselectronicas.com/historia-de-los-sistemas-de-alerta-previa-y-notificacion-de-emergencias/>

Medina, F. (2021). *El Comercio*. Obtenido de El Comercio:

<https://www.elcomercio.com/actualidad/presencia-policias-guayaquil-violencia-ruales.html>

Programacion. (2018). *LA EVOLUCIÓN E IMPLICACIONES DEL RASTREO*

SATELITAL. Obtenido de DECA: <https://www.grupodeca.com.mx/la-evolucion-e-implicaciones-del-rastreo-satelital/>

Ricoy Riego, A. (2020). *App Inventor en Español*. Obtenido de App Inventor en

Español: <https://sites.google.com/site/appinventormegusta/primeros-pasos>

Staff USERS. (2014). *ELECTRÓNICA - Plataformas Arduino y Raspberry Pi*. USERS.

Uriarte, I. (2019). *Prometec*. Obtenido de Prometec: <https://www.prometec.net/sim808/>