



# **UNIVERSIDAD TÉCNICA DEL NORTE**

## **FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS CARRERA DE INGENIERÍA EN MANTENIMIENTO AUTOMOTRIZ**

### **TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN MANTENIMIENTO AUTOMOTRIZ**

**TEMA: DISEÑO E IMPLEMENTACIÓN DE SISTEMA ELECTRÓNICO DE  
SEGURIDAD PARA MONITOREO EN TIEMPO REAL DEL EXCESO DE  
VELOCIDAD EN EL VEHÍCULO**

**AUTOR: JONATHAN FLAVIO CADENA ROMERO**

**DIRECTOR: ING. JORGE MELO, MSc.**

**Ibarra 2021**

## CERTIFICADO

### ACEPTACIÓN DEL DIRECTOR

En mi calidad de director del plan de trabajo de grado, previo a la obtención del título de Ingeniería en Mantenimiento Automotriz, nombrado por el Honorable Consejo Directivo de la Facultad de Ingeniería en Ciencias Aplicadas.

### CERTIFICO:

Que una vez analizado el plan de grado cuyo título es "Diseño e Implementación de sistema electrónico de seguridad para monitoreo en tiempo real del exceso de velocidad en el vehículo" presentado por el señor: Cadena Romero Jonathan Flavio con número de cédula 172467644-8, doy fe que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a presentación pública y evaluación por parte de los señores integrantes del jurado examinador que se designe.

En la ciudad de Ibarra, a los 18 días del mes de febrero del 2021

Atentamente



Firmado electrónicamente por:  
**JORGE LUIS  
MELO OBANDO**

Ing. Jorge Luis Melo Obando MSc.

**DIRECTOR DEL TRABAJO DE GRADO**



## UNIVERSIDAD TÉCNICA DEL NORTE

### BIBLIOTECA UNIVERSITARIA

#### AUTORIZACIÓN DE USO Y PUBLICACIÓN

#### A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CEDULA DE IDENTIDAD:	172467644-8		
APELLIDOS Y NOMBRES:	Cadena Romero Jonathan Flavio		
DIRECCIÓN:	San Vicente de Guayllabamba – Quito		
EMAIL:	jfcadenar@utn.edu.ec		
TELÉFONO FIJO	2164 260	TELÉFONO MÓVIL:	095 992 7886
DATOS DE LA OBRA			
TÍTULO:	DISEÑO E IMPLEMENTACIÓN DE SISTEMA ELECTRÓNICO DE SEGURIDAD PARA MONITOREO EN TIEMPO REAL DEL EXCESO DE VELOCIDAD EN EL VEHÍCULO		
AUTOR (ES):	Cadena Romero Jonathan Flavio		
FECHA:	18 de febrero del 2021		
SOLO PARA TRABAJOS DE GRADO			
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO	<input type="checkbox"/> POSGRADO	
TÍTULO POR EL QUE OPTA	INGENIERÍA EN MANTENIMIENTO AUTOMOTRIZ		
ASESOR/DIRECTOR	Ing. Jorge Luis Melo Obando MSc.		

## 2. CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá de defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 18 días del mes de febrero del 2021

**AUTOR:**



Cadena Romero Jonathan Flavio  
172467644-8

## **DEDICATORIA**

El presente trabajo quiero dedicar principalmente a mis padres Flavio Cadena y Maria Romero quienes con su amor, esfuerzo y dedicación fueron el pilar fundamental para lograr la meta tan anhelada de ser Ingeniero en Mantenimiento Automotriz, al enseñarme valores y darme todo lo necesario para culminar con mi carrera.

También quiero dedicar este logro a mi esposa Evelyn Parreño quien ha estado conmigo en todo momento, apoyándome con sus palabras, con su presencia y al estar siempre motivándome a ser mejor persona y un excelente profesional. Además dedicar este resultado a mis hermanos que siempre estuvieron presentes, brindándome su ayuda bajo cualquier circunstancia.

**Jonathan Cadena**

## **AGRADECIMIENTO**

Quiero agradecer primero a Dios por darme salud y las fuerzas necesarias para poder alcanzar este objetivo. También un enorme agradecimiento a mis padres quienes con su ejemplo hicieron de mi una persona responsable, honesta y agradecida de la vida además me brindaron su dedicación y recursos para que hoy pueda tener el anhelado título universitario.

Agradezco a mi amada esposa quien a lo largo de la carrera siempre me brindo palabras de animo en los momentos más difíciles y confió en mi. Gracias por todo mi amor.

También quiero agradecer a toda mi familia que se mantuvo apoyándome en toda esta etapa, especialmente a mis hermanos y sobrinos quienes fueron mi motivación para seguir adelante e intentar ser un ejemplo, gracias por sus palabras y consejos acertados en el momento adecuado, fue una etapa difícil pero deben saber que este logro es de todos.

Finalmente un profundo agradecimiento a la Universidad Técnica del Norte y a todos sus docentes con quienes aprendí sobre mi pasión que son los automotores y la responsabilidad de realizar un buen trabajo.

**Jonathan Cadena**

## ÍNDICE DE CONTENIDOS

	PÁGINA
<b>RESUMEN</b>	<b>XIV</b>
<b>ABSTRACT</b>	<b>XV</b>
<b>INTRODUCCIÓN</b>	<b>XVI</b>
<b>1. REVISIÓN BIBLIOGRÁFICA</b>	<b>1</b>
1.1 Objetivos	1
1.1.1 Objetivo General	1
1.1.2 Objetivos Específicos	1
1.2 Justificación	1
1.3 Alcance	2
1.4 Antecedentes	3
1.5 Microcontrolador	4
1.6 Plataforma de Prototipos Electrónica Arduino	5
1.6.1 Historia de Arduino	6
1.6.2 Entorno de Desarrollo Integrado (Integrated Development Environment)	7
1.6.3 Entradas y Salidas	8
1.6.4 Shields	8
1.6.4.1 Sistema Global de Comunicaciones Móviles (GSM Shield)	8
1.6.4.2 Wifi Shield 101	9
1.6.4.3 Ethernet Shield	10
1.6.5 Protoboard	10
1.7 Sistema de Posicionamiento Global (GPS)	11
1.7.1 División Gps	12
1.7.1.1 Espacial	13
1.7.1.2 De Control	13
1.7.1.3 Receptor	14
1.7.2 Funcionamiento del GPS	14

1.8	Módulo GSM /GPRS	15
1.8.1	Servicios GSM	16
1.8.2	Servicio General de Paquetes Vía Radio (GPRS)	17
1.8.3	Módulo De Identificación de Abonado (Subscriber Identity Module SIM)	17
1.9	Almacenamiento de Datos (Nubes de Datos)	17
1.9.1	Computación en la Nube (Cloud Computing)	18
1.9.2	Características de la Computación en la Nube	18
1.9.3	Modelos de Servicios en la Nube	19
1.9.4	Máquina a Máquina (M2M)	19
1.9.5	Bases de Datos	20
1.9.6	Herramientas de Desarrollo	21
1.10	Internet de las Cosas (Internet Of Things)	21
1.11	Sistema de Visualización Remota	23
1.11.1	Mapas Web	23
<b>2.</b>	<b>MATERIALES Y MÉTODOS</b>	<b>24</b>
2.1	Materiales y Equipos	25
2.1.1	Arduino	25
2.1.1.1	Pines Arduino Uno	26
2.1.1.2	Arduino IDE	27
2.1.2	Modulo SIM 808 GSM/GPRS + GPS	28
2.1.2.1	Servicio General de Paquetes Vía Radio	30
2.1.3	Página Web y Base de Datos	31
2.1.4	Fuente de Alimentación DC-DC XL4015	33
2.2	Métodos	34
2.2.1	Desarrollo Interfaz Electrónica Arduino – GPS	34
2.2.1.1	Enlace Arduino – SIM 808 GSM/GPRS + GPS	34
2.2.1.2	Comunicación GPS mediante Comandos AT	36
2.2.1.3	Pruebas de Campo	40
2.2.2	Implementación Módulo de Internet	40
2.2.2.1	Comunicación GPS-Arduino-Internet	40
2.2.3	Nube de Almacenamiento	43
2.2.3.1	Diseño Pagina Web	43

2.2.3.2	Desarrollo Base de Datos en la Nube	47
2.2.3.3	Parametrización de los Límites de Velocidad	50
2.2.4	Implementación del Sistema en el Vehículo	53
2.2.4.1	Pruebas en Tiempo Real	54
<b>3.</b>	<b>RESULTADOS Y DISCUSIÓN</b>	<b>56</b>
3.1	Descripción e Implementación Prototipo	56
3.2	Análisis Financiero	59
3.3	Pruebas de Campo Datos GPS	60
3.4	Envío de Información mediante Datos Móviles	64
3.4.1	Envío Datos HTTP	64
3.4.2	Discusión Envío de Datos	69
3.5	Visualización de Datos en la Página Web	70
3.5.1	Descripción Página Web	72
3.5.2	Discusión Pagina Web	75
3.6	Instalación y Pruebas en Tiempo Real	76
3.6.1	Montaje	76
3.6.2	Prueba De Funcionamiento En Ciudad	78
3.6.3	Prueba De Funcionamiento En Carretera	80
3.6.4	Análisis De Pruebas	82
3.6.5	Resultado De Las Pruebas	85
<b>4.</b>	<b>CONCLUSIONES Y RECOMENDACIONES</b>	<b>87</b>
4.1	Conclusiones	87
4.2	Recomendaciones	88
	<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>90</b>
	<b>ANEXOS</b>	<b>95</b>

## ÍNDICE DE TABLAS

<b>TABLA NÚM.</b>	<b>PÁGINA</b>
<b>Tabla 2.1</b> Características Placa Arduino UNO	26
<b>Tabla 2.2</b> Características técnicas del módulo SIM 808 GSM/GPRS + GPS	29
<b>Tabla 2.3</b> Características GPRS modulo SIM 808	31
<b>Tabla 2.4</b> Características Paquete Gratuito Hosting	32
<b>Tabla 2.5</b> Características Regulador de Voltaje	33
<b>Tabla 3.1</b> Costo Sistema de Seguridad	59
<b>Tabla 3.2</b> Costos adicionales del Proyecto	59
<b>Tabla 3.3</b> Excesos de velocidad almacenados el día 11 de febrero del 2020	79
<b>Tabla 3.4</b> Excesos de velocidad en carretera	81
<b>Tabla 3.5</b> Comparación Pruebas	83

## ÍNDICE DE FIGURAS

<b>FIGURA NÚM.</b>	<b>PÁGINA</b>
1.1 Estructura de un microcontrolador	4
1.2 Arduino Mega	6
1.3 Shield GSM de Arduino	9
1.4 Placa Protoboard	10
1.5 Meridiano de Greenwich y línea Ecuatoriana	12
1.6 Satélites en la Orbits	13
1.7 Módulo receptor GPS PARALLAX	14
1.8 Redes de comunicación móvil	16
1.9 Estructura del IoT	22
2.1 Flujograma	24
2.2 Pines digitales Arduino UNO	26
2.3 Pines Análogos Arduino UNO	27
2.4 Partes del Software Arduino IDE	28
2.5 SIM 808 GSM/GPRS + GPS	29
2.6 Pines Modulo SIM 808	30
2.7 Pantalla principal hosting	31
2.8 Paquetes del hosting	32
2.9 Fuente de alimentación	33
2.10 Enlace Arduino UNO - SIM 808	34
2.11 Conexión entre Arduino Uno y la SIM 808	35
2.12 Conexión física PC – Arduino - SIM 808	36
2.13 Placa Arduino y Puerto COM	37
2.14 Endender GPS comandos AT	38
2.15 Estado GPS comandos AT	38
2.16 Datos GPS comandos AT	39
2.17 Enlace Arduino - GPS - Internet	40
2.18 Ranura tarjeta SIM	41
2.19 Configuración del sistema dentro del IDE de Arduino	42
2.20 Tipo de alojamiento en el Hosting	44
2.21 Registro en el Hosting Web	44
2.22 Nombre del sitio web	45
2.23 Diseño en Visual Studio	45
2.24 Enlace de conexión al dominio	46
2.25 Previsualización en Localhost	46
2.26 Cargar archivos al Dominio	47

<b>2.27</b>	Base de Datos	48
<b>2.28</b>	Tablas de la base de datos	49
<b>2.29</b>	Parametrización Norte de Ibarra	50
<b>2.30</b>	Parametrización Sur de Ibarra	51
<b>2.31</b>	Datos de Parametrización Excel	52
<b>2.32</b>	Datos Parametrización en la Base de Datos	52
<b>2.33</b>	Esquema del dispositivo conectado al vehículo	54
<b>2.34</b>	Esquema completo del sistema	55
<b>3.1</b>	Regulador DC-DC XL4015 con su potenciómetro	57
<b>3.2</b>	Conexión Regulador de Voltaje	57
<b>3.3</b>	Conexión Hardware	58
<b>3.4</b>	Primera conexión del prototipo	60
<b>3.5</b>	Primer prueba de coordenadas, panamericana norte 03/12/2019, 22:59	61
<b>3.6</b>	Primeras coordenadas Google Maps	61
<b>3.7</b>	Segunda prueba de coordenadas, Avenida 17 de Julio 04/12/2019, 01:39	62
<b>3.8</b>	Segundas coordenadas Google Maps	62
<b>3.9</b>	Tercera prueba coordenadas, calle Miguel Oviedo 04/12/2019, 01:50	63
<b>3.10</b>	Terceras coordenadas Google Maps	63
<b>3.11</b>	Conexión Ready	64
<b>3.12</b>	Cobertura de red Claro	65
<b>3.13</b>	Conexión módulo SIM 808 a la red GPRS	66
<b>3.14</b>	Conexión a la APN Claro	66
<b>3.15</b>	Envío de datos mediante HTTP	67
<b>3.16</b>	Comandos para el envío de Datos	68
<b>3.17</b>	Base de datos de la página web	69
<b>3.18</b>	Funcionamiento página web	71
<b>3.19</b>	Inicio de Sesión Pagina web	72
<b>3.20</b>	Pantalla TIEMPO REAL	73
<b>3.21</b>	Reporte por fechas	73
<b>3.22</b>	Reporte de velocidad	74
<b>3.23</b>	Reporte del exceso de velocidad	75
<b>3.24</b>	Vehículo de Instalación	77
<b>3.25</b>	Montaje antena GPS	77
<b>3.26</b>	Instalación dispositivo	78
<b>3.27</b>	Imagen tomada en la ciudad de Ibarra el día 11 de febrero, 07:15	79
<b>3.28</b>	Mapa exceso de velocidad en ciudad	80
<b>3.29</b>	Imagen tomada en carretera el día 11 de febrero a las 07:02:52	81
<b>3.30</b>	Mapa exceso de velocidad en carretera	82

## ÍNDICE DE ANEXOS

<b>ANEXO NÚM.</b>	<b>PÁGINA</b>
1. Programación IDE de arduino	96
2. Datos parametrización	101
3. Programación página web	102

## RESUMEN

El presente trabajo tiene como objetivo principal el diseñar e implementar un sistema electrónico de seguridad para el monitoreo en tiempo real de la velocidad del vehículo, es decir, obtener la información de forma inmediata, así como también almacenar los registros de excesos de velocidad. Para lograrlo primeramente se realizó una investigación bibliográfica sobre el funcionamiento y características de cada uno de los dispositivos usados, así como también sobre su manera de conexión y programación para ponerlo en marcha correctamente. El siguiente paso consistió en dividir el trabajo en cuatro fases, la primera fase fue obtener los datos de coordenadas, velocidad, fecha y hora mediante GPS(Sistema de Posicionamiento Global) y Arduino UNO, seguidamente se procedió con la fase dos donde se realizó la conexión de hardware y software para que el módulo SIM 808 GPS/GPRS mediante el uso de datos móviles pueda enviar la información a un servidor externo en un intervalo de tiempo de 15 segundos, en esta fase se usó un SIM de la Operadora Claro con un paquete activo de Megabyte. En la fase tres se realizó el diseño de una página web con la ayuda del programa Visual Studio que se cargó al *hosting* llamado “Somee” y además se añadió la base de datos creada en Microsoft SQL donde ingresa toda la información que envía el módulo SIM 808. Cabe recalcar que antes de avanzar a la siguiente fase se fueron realizando pruebas de campo para corroborar su funcionamiento, al tener una respuesta positiva la fase cuatro fue implementar el sistema en el vehículo para lo cual se usó un regulador de voltaje que sirve para obtener la corriente constante y requerida (5V y 2A). Finalmente, luego de haber diseñado e implementado el sistema en el vehículo, se realizaron test por rangos de tiempo prolongados pudiendo concluir que el sistema funciona de forma exitosa coincidiendo la información de los registros con los datos reales, además la interacción del usuario con la página web es amigable, quien puede ingresar desde cualquier dispositivo con acceso a internet para realizar el monitoreo y consulta.

## ABSTRACT

The main objective of this work was to design and implement an electronic safety system for real-time monitoring of vehicle speed, in other words, to obtain information immediately, as well as to store records of speeding. In order to achieve this, first a bibliographic investigation was made on the functioning and features of each one of the devices used, as well as on their way of connection and programming to start it up correctly. The next step was to divide the work into four phases, where the first phase was to obtain the coordinate data, speed, date and time by means of GPS (Global Positioning System) and Arduino UNO. In the second phase the connection was made in order for the SIM 808 GPS/GPRS module to send the information to an external server in a 15 second time interval, using mobile data. In this phase, a SIM from operator *Claro* with an active packet of Megabytes was used. In phase three the design of a web page was made with the assistance of the program Visual Studio that was uploaded to the hosting called "*Somee*". In addition, the database created in Microsoft SQL was added, where all the information sent by the SIM 808 module was entered. It is worth mentioning that before moving on to the next phase, field tests were carried out to corroborate its operation. Based on the positive results, phase four implemented the system in the vehicle for which a voltage regulator was used to obtain the required constant current (5V and 2A). Finally, after having designed and implemented the system in the vehicle, tests were carried out for long periods of time, concluding that the system functioned successfully, coinciding the information in the registers with the real data. Furthermore, the user interaction with the web page is friendly, and the user can enter from any device with internet access to carry out the monitoring and consultation.

## INTRODUCCIÓN

En el mundo entero una de las principales causas de accidentes de tránsito es el exceso de velocidad según la OMS (Organización Mundial de la Salud, 2017, pág. 2) y esta problemática es cada vez mayor para la sociedad debido al acelerado e incontrolable incremento del parque automotor. En el Ecuador se han realizado múltiples campañas para concientizar sobre el peligro del exceso de velocidad y sus graves consecuencias, pero al no tener el éxito deseado se han implementado medidas más radicales, como por ejemplo radares en las vías y sanciones económicas-penales que se encuentran estipuladas en el Código Orgánico Integral Penal. Ante esta problemática, el proyecto busca diseñar una herramienta que permita monitorear la velocidad de conducción del vehículo, además de acceder a estos registros desde cualquier ubicación remota en tiempo real mediante la ayuda de dispositivos electrónicos. Para llevar a cabo la elaboración de este sistema es necesario el uso principalmente de la tecnología con la que hoy en día contamos, como un sistema de almacenamiento en la nube mediante la ayuda de un servicio de *hosting*, una plataforma de prototipos electrónica Arduino, un sistema de posicionamiento global y un Módulo SIM 808 GSM/GPRS, pero sobre todo la aplicación de los conocimientos adquiridos y la investigación realizada en artículos científicos, libros, tesis, etc. Al concluir este proyecto se pretende tener un sistema confiable con altos estándares de calidad proporcionando a los organismos de control una herramienta que permita un control exacto. Por ejemplo, al tener instalado el sistema en una flota de vehículos a nivel nacional se lograría captar de forma inmediata los registros de exceso de velocidad que se hayan producido a cualquier hora y lugar, haciendo posible que se pueda realizar sanciones justas y equitativas a todos los conductores que hayan infringido la ley, para de esa manera ayudar a disminuir la cantidad de accidentes de tránsito y por ende también la tasa de mortalidad de nuestro país.

# CAPÍTULO I

## 1. REVISIÓN BIBLIOGRÁFICA

### 1.1 OBJETIVOS

#### 1.1.1 OBJETIVO GENERAL

Diseñar e implementar un sistema electrónico de seguridad para el monitoreo en tiempo real del exceso de velocidad en el vehículo.

#### 1.1.2 OBJETIVOS ESPECÍFICOS

- Desarrollar una interfaz electrónica que permita tomar lectura de la velocidad en el vehículo con el uso de GPS.
- Implementar un módulo *Shield* que permita comunicar al vehículo con un servidor externo mediante datos móviles.
- Emplear una nube de almacenamiento de información que permita registrar los datos de velocidad del vehículo y acceder desde ubicaciones remotas.

### 1.2 JUSTIFICACIÓN

En el Ecuador existen tasas de siniestralidad muy altas, lo que constituye un problema social que se debe intentar resolver. Según la Agencia Nacional de tránsito, en el 2017 ocurrieron aproximadamente 79 accidentes de tránsito al día debido a la imprudencia de los conductores que no acatan las señales de tránsito (Agencia Nacional de Tránsito, 2018). A pesar de existir programas de capacitación y concientización a los conductores tanto profesionales como no profesionales, se nota la necesidad de tener un control o monitoreo para evitar siniestros.

Dentro de América Latina, según estadísticas Ecuador es el segundo país en tener más fallecidos a causa de accidentes de tránsito, además según la OMS (Organización Mundial de la Salud) se

tienen pérdidas económicas de alrededor del 3% del PIB, que lo que para el Ecuador equivaldría a unos 3 093 miles de millones de dólares (eltelégrafo, 2017, pág. 1).

El presente trabajo se enfoca en aportar tecnología que ayude a mitigar el problema de los accidentes de tránsito que se ocasionan por el exceso de velocidad, este inconveniente es una de las principales causas de muerte, lesiones y pérdidas económicas en el país. Al poder monitorear en tiempo real la velocidad del vehículo desde cualquier lugar remoto se puede tener un mayor control, además de conocer la ubicación y parámetros del momento exacto que el vehículo excedió el límite de velocidad.

Mediante la aplicación de este proyecto a nivel nacional se conseguiría disminuir las tasas de siniestralidad y por ende también las tasas de mortalidad en el país, además se podría concientizar a los conductores mediante la implementación de multas significativas al momento que el vehículo rebase los límites de velocidad.

La idea de este proyecto se basa principalmente en el objetivo 1 del Plan Nacional de Desarrollo, donde menciona la importancia de vivir en un ambiente seguro y también que se debe hacer énfasis en prevenir las causas de muerte en el país, como en este caso se pretende prevenir los accidentes de tránsito y sus consecuencias mortales (Secretaría Nacional de Planificación y Desarrollo (Senplades), 2017, pág. 53). Adicional, este proyecto apoya principalmente a las estrategias de transformación de la matriz productiva, donde habla sobre la importancia de fomentar al desarrollo de la tecnología en el país (Secretaría Nacional de Planificación y Desarrollo (Senplades), 2012, págs. 11,12).

### **1.3 ALCANCE**

En este proyecto se diseñará un sistema de interfaz electrónica con GPS y un módulo Arduino que se instalará en un vehículo, la cual permitirá obtener datos de ubicación, fecha, hora y velocidad.

Además, se realizará la comunicación del módulo Arduino a una nube de almacenamiento de información, que permitirá acceder a estos datos desde cualquier ubicación remota para

monitorear la velocidad del vehículo en tiempo real, así como también se tendrá información de las condiciones en las que se excedió los límites de velocidad permitidos.

Este proyecto llegará hasta el diseño e implementación del sistema electrónico de seguridad para el monitoreo de velocidad en tiempo real en un vehículo donde se realizará la prueba piloto y se accederá a esa información desde un dispositivo remoto. Es importante mencionar que no se realizará el estudio sobre los efectos o el índice de reducción de accidentes de tránsito que se tendría al utilizar este dispositivo en varios vehículos del país.

## **1.4 ANTECEDENTES**

Los accidentes de tránsito desde hace varios años se considera una problemática a nivel mundial, de acuerdo con la Organización Mundial de la Salud (OMS) alrededor de 1,25 millones de personas mueren cada año en colisiones debido a causas como: exceso de velocidad, conducción distraída, manejar bajo efectos de alcohol, etc (Organización Mundial de la Salud, 2017, pág. 2). Específicamente, se tiene que la tercera parte de los accidentes de tránsito son provocados por el exceso de velocidad o por conducir a una velocidad inadecuada (Noticias ONU, 2017, pág. 1).

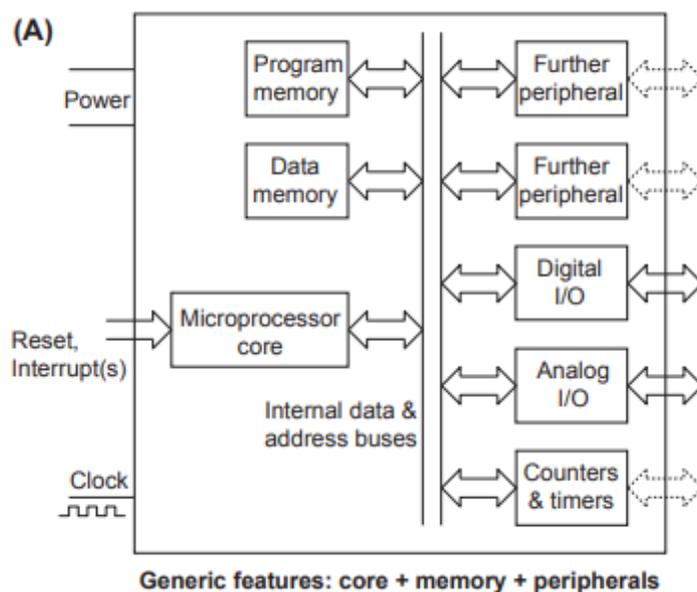
Al mismo grado que se eleva la velocidad de conducción, también se tiene mayor peligro de que ocurran accidentes de tránsito y que sus consecuencias sean catastróficas, por lo que en varios países se ha hecho énfasis en solucionar este problema al establecer límites nacionales de velocidad (Organización Mundial de la Salud, 2015, pág. 6).

En el Ecuador, los límites de velocidad son designados por los GADs (Gobiernos Autónomos Descentralizados) de acuerdo con la situación territorial. Estos, siguen los lineamientos generales que se especifican en el artículo 191 del “Reglamento a ley de transporte terrestre, tránsito y seguridad vial” donde se clasifica de acuerdo con el vehículo ya sea: automotores livianos, de carga y de transporte público o comercial (Agencia Nacional de Tránsito, 2016, pág. 44).

A nivel nacional se ha intentado concientizar a los conductores socializando sobre el exceso de velocidad y sus graves consecuencias que además de daños materiales termina en traumatismos o incluso muertes. En vista de la gran cantidad de accidentes que se registra en la ANT y que se producen por causa de la velocidad se ha notado la necesidad de sancionar, por lo que en el COIP (Código Orgánico Integral Penal) se detallan las sanciones que van desde la reducción de seis puntos y una multa equivalente al treinta por ciento de un salario básico unificado, hasta tres días de cárcel, una multa de un salario básico unificado y la reducción de diez puntos a la licencia (Zúñiga Rocha & González Camacho, 2014, pág. 1).

## 1.5 MICROCONTROLADOR

Por las características esenciales que posee el microcontrolador se asemeja a un computador, ya que contiene una unidad central de procesamiento (CPU), memoria y periféricos tanto de entrada como de salida, además añade rasgos que permiten realizar su trabajo de control sobre dispositivos electrónicos. El microcontrolador puede realizar distintas tareas de control siempre y cuando tenga una alimentación de corriente continua y una señal de reloj estable, ya que esto es indispensable para su correcto funcionamiento (Arenas Mas, 2008, pág. 21).



**Figura 1.1** Estructura de un microcontrolador  
(Toulson & Wilmshurst, 2017, pág. 10)

La estructura del microcontrolador como se puede notar en la Figura 1.1 consta de tres partes fundamentales para su funcionamiento: un núcleo el cual es el CPU y los circuitos destinados al control, además, una memoria donde se almacena todos los datos así como también el programa, es importante señalar que, lo que lo hace diferente de los microprocesadores son los periféricos, es decir, las entradas y salidas tanto análogas como digitales que puede tener, las cuales ayudan a la interacción de los sistemas exteriores con el microcontrolador (Toulson & Wilmshurst, 2017, págs. 10,11).

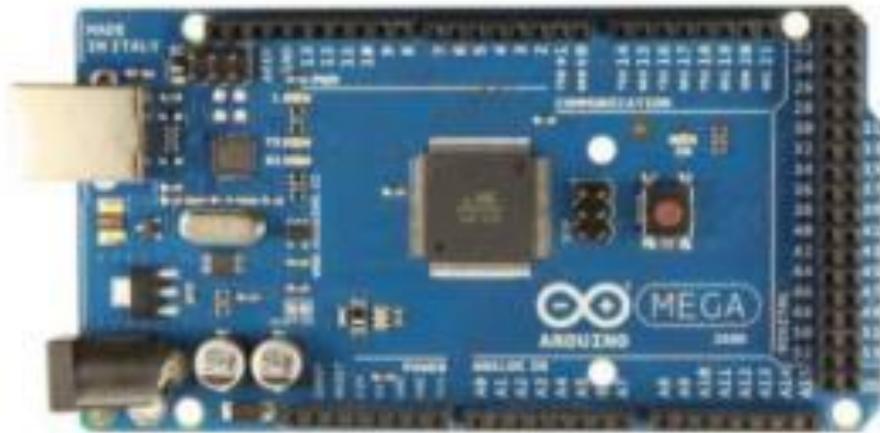
A pesar del pequeño tamaño de un microcontrolador (alrededor de 13 milímetros por lado) este posee una gran complejidad y un alto trabajo de ingeniería, donde se lleva a cabo distintos procesos para el control de sistemas externos, cuya función es que mediante datos de entrada iniciar un algoritmo interno realizado por un programador dando así una señal de salida diferente.

## **1.6 PLATAFORMA DE PROTOTIPOS ELECTRÓNICA ARDUINO**

Arduino es una plataforma de prototipos electrónicos, conformada básicamente por un microcontrolador que contiene tanto entradas como salidas digitales y análogas, el cual permite programar una gran variedad de sistemas mediante una biblioteca que contiene un lenguaje simplificado de la programación C o C++. Esta plataforma cuenta con la parte de hardware, así como de software, cabe indicar que lo que le hace tan atractivo es que su lenguaje de programación es libre y se puede usar para cualquier necesidad (Caicedo Pedrera, 2017, pág. 5).

La placa de Arduino es un sistema de control, el cual mediante sus pines de entrada tanto análogas como digitales se obtiene señales de sensores que interpretan variables externas, los cuales son procesados y a través del lenguaje de programación se puede dar órdenes o señales eléctricas a actuadores conectados a los pines de salida del Arduino. Por lo tanto, este sistema es capaz de transformar los datos en señales eléctricas.

Arduino está basado en un medio *open-source*, es decir, que todos sus códigos están libremente esparcidos, muchas personas divulgan su programación y permiten a cualquier persona modificar, mejorar y adecuar los códigos de programación para su uso. Además, es multiplataforma lo que quiere decir que permite la programación desde diferentes sistemas operativos como Windows, Mac os, entre otras.



**Figura 1.2** Arduino Mega  
(Caicedo Pedrera, 2017, pág. 11)

En la actualidad se tiene algunos tipos de Arduino, entre los más conocidos están: Arduino Nano, Arduino Uno y como se muestra en la Figura 1.2 Arduino Mega. Todo depende del uso que se le vaya a dar, por ejemplo, si para el proyecto se necesita más número de entradas y salidas, se puede utilizar la placa Arduino Mega, si en el caso de necesitar más memoria y/o velocidad se podrá usar las placas Zero o Due, por otra parte si se necesita un menor tamaño de la placa se tiene disponibilidad de incorporar un Arduino Nano o Pro Mini y también existe la posibilidad de utilizar el Arduino Yún o Ethernet que tienen capacidad para conexión a internet (Torrente Artero, 2016, págs. 102-115).

### 1.6.1 HISTORIA DE ARDUINO

Arduino nació en el año 2005 debido a la necesidad de ayudar económicamente con las ganancias al Instituto IVRAE para que no cerrara sus puertas y adicionalmente para reducir

costos en proyectos estudiantiles, por lo que desde el inicio la idea fue crear un sistema de hardware y software libre para el uso académico.

Dentro del equipo fundador está: Massimo Banzi que inicialmente hizo una placa simple conectando un microcontrolador con resistencias, luego David Mellis realizó un arduo trabajo mejorando su interfaz de software, más tarde David Cuartielles colaboró con la mejora del hardware adicionando suficientes microcontroladores, después Tom Igoe hizo que la placa sea más potente y añadió puertos USB, por último Gianluca Martino se encargó de la distribución inicial así como de la venta en masa que se produjo con la buena acogida; su nombre es debido al lugar que frecuentaban ir los fundadores (Reyes Cortés & Cid Monjaraz, 2015, págs. 3-9).

La primera producción de las placas Arduino se realizó para entregar a los estudiantes del Instituto IVRAE quienes empezaron a realizar distintos proyectos académicos, siendo uno de los primeros el reloj-alarma. En la actualidad Arduino es una placa utilizada a nivel mundial para realizar tanto pequeños proyectos de principiantes, así como proyectos más avanzados de investigación.

### **1.6.2 ENTORNO DE DESARROLLO INTEGRADO (INTEGRATED DEVELOPMENT ENVIRONMENT)**

El software IDE (Integrated Development Environment) es el programa que se ejecuta en el computador el cual permite escribir *sketches* para luego cargarlo en la placa Arduino, este software es muy importante ya que es el encargado de transformar de un lenguaje *processing* que es un lenguaje fácil y manejable a un lenguaje C que es más complejo de entender (Banzi & Shiloh, 2016, pág. 22).

Por lo tanto, el proceso que se realiza para que funcione una placa Arduino es:

- Conectar el Arduino al computador mediante un cable USB.
- Desarrollar la programación o el *sketch* que sea necesario.
- Cargar la programación escrita a la placa Arduino.

- La programación cargada se ejecutará en la placa al estar conectada mediante USB o con una fuente de alimentación externa.

### **1.6.3 ENTRADAS Y SALIDAS**

Todo microcontrolador cuenta con entradas y salidas las cuales sirven para conectarse con el mundo exterior, en el caso de la tarjeta Arduino lleva los conectores hembra alrededor de la placa, donde puede tener una conexión rápida y fácil sin necesitar de un proceso de soldadura.

El número de los conectores depende de la tarjeta Arduino, sobre estos también se apilan o conectan fácilmente los módulos *Shield*, los cuales sirven para funciones específicas, estas entradas y salidas tienen características especiales que pueden trabajar con máximo 5v de tensión y 40 mA de corriente eléctrica (Goilav & Loi, 2016, págs. 13,14).

### **1.6.4 SHIELDS**

Los módulos *Shield* son placas que prestan un servicio adicional, las cuales se montan sobre la placa Arduino, teniendo una conexión directa mediante los pines machos que ofrece los *Shields* con los pines hembra de la placa Arduino. Estos módulos de acuerdo con las necesidades pueden apilarse uno sobre otro, claro está, depende de que cuente con pines hembra para su conexión.

#### **1.6.4.1 Sistema Global de comunicaciones Móviles (GSM Shield)**

Global System for Mobile, es un sistema de telefonía móvil el cual permite enviar o recibir mensajes de texto y multimedia, además de realizar llamadas de voz y poder conectarse a internet. A pesar de ser 2G (segunda generación) puede conectarse a redes GPRS (General Packet Radio Service) que son 2.5G, pero no puede conectarse a redes más confiables y mucho más rápidas como son: UMTS (3G), HSPA (3.5G) Y LTE (4G) (Torrente Artero, 2016, págs. 157-202).

Funciona con 5V operativos, mismos que son proporcionados por la placa Arduino al módulo *Shield* GSM, la cual, a su vez se conecta mediante cable USB a un computador o se puede conectar a una fuente de alimentación externa (Ahuja & Bhavsar, 2018, pág. 178).



**Figura 1.3** Shield GSM de Arduino  
(Ahuja & Bhavsar, 2018, pág. 177)

En la Figura 1.3 se muestra un módulo GSM *Shield*, donde se nota los pines y la estructura del hardware que tiene la placa así por ejemplo donde se coloca la tarjeta SIM (Subscriber Identification Module).

#### 1.6.4.2 WiFi Shield 101

El módulo *Shield* 101 está diseñado para el Internet de las cosas, por lo que tiene la capacidad de conectar a la placa Arduino a internet mediante redes WiFi simplemente teniendo las características básicas de la red como el nombre, contraseña y realizando una configuración sencilla de programación.

Este *Shield* tiene la posibilidad de conectarse con las placas Arduino Uno, Mega, Zero o Due y además permite el montaje de otro módulo *Shield* mediante sus pines hembras.

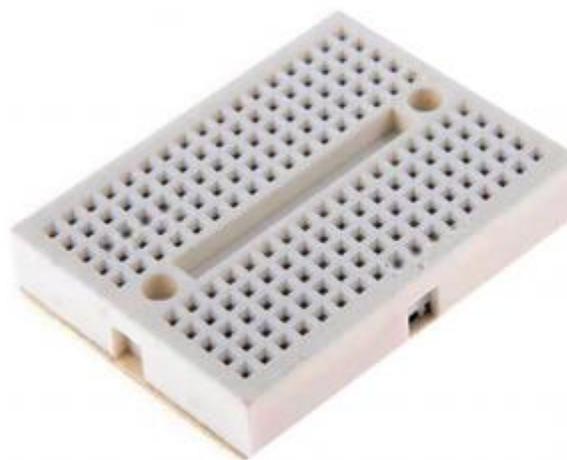
### 1.6.4.3 Ethernet Shield

La placa *Shield* Ethernet es un módulo que permite la conexión a internet mediante cable, es decir mediante un puerto RJ-45 se puede conectar a un router y así conectarse en una red LAN (Local Área Network) donde pueda leer sensores, o activar actuadores que estén conectadas a la misma red por cable.

Este módulo cuenta con una ranura para insertar una microSD por lo que al colocar la tarjeta sirve para guardar ciertos datos o ficheros y compartir a la red; además mediante PoE (Power Over Ethernet) se puede obtener el voltaje necesario para que el sistema se alimente mediante el RJ-45 y así no conectar un cable adicional de alimentación externa.

### 1.6.5 PROTOBOARD

Luego de crear la programación, cargarla en la placa Arduino y ponerla en funcionamiento, se necesita conectar algunos sensores, actuadores y más, los cuales una vez que se hayan realizado las pruebas se conectarán de manera fija. Pero primeramente se necesita una placa donde se pueda conectar y desconectar fácilmente los dispositivos electrónicos para poder realizar los ensayos, por lo que la placa Protoboard es la más indicada para este trabajo.



**Figura 1.4** Placa Protoboard  
(Puerchambud Ladino & Henao Ariza, 2018, pág. 43)

Una placa Protoboard tiene varios conectores hembra como se nota claramente en la Figura 1.4, donde se pueden ir conectando los dispositivos, dentro la placa esta interconectada mediante una serie de planchas metálicas, y depende de la manera en que se conecte para que los componentes interactúen entre sí (Lozano Equisoain, 2017, pág. 7).

## **1.7 SISTEMA DE POSICIONAMIENTO GLOBAL (GPS)**

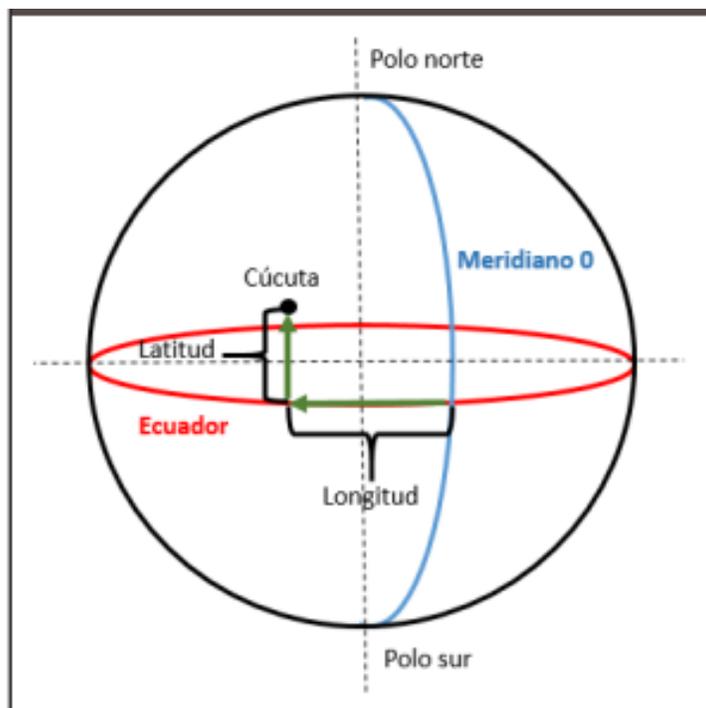
El sistema GPS (Global Positioning System) empezó a tener funcionamiento desde los años 60, donde se tuvo el sistema de posicionamiento TRANSIT que se dio uso en la marina, donde los satélites de este GPS se encontraban a 1100 km de altitud, finalmente se dejó de usarse a final de 1995 (Leick, Rapoport, & Tatarikov, 2015, págs. 225-244).

Los Estados Unidos se encargan de controlar los sistemas de posicionamiento global, los cuales principalmente se usan con fines militares, aunque en las últimas décadas se ha empezado a tener diferentes usos mediante personas civiles.

El uso del GPS se basa en la utilización de satélites que se encuentran en órbita. Desde 1978 hasta 1997 se han lanzado al espacio varios llamados Bloque I, Bloque II, Bloque IIA Y Bloque IIR. Este conjunto de satélites puede medir la ubicación exacta de un sistema en la tierra por lo que al contar con una mayor cantidad de satélites se ha logrado tener mejores prestaciones del sistema de posicionamiento global, esto hace que su uso esté creciendo muy rápidamente.

La implementación de dichos dispositivos GPS hace posible conocer en tiempo real, datos como: longitud, latitud, velocidad y curso, que son elementos necesarios para establecer la ubicación y trayectoria de un objeto. Además, se tiene la posibilidad de aplicar e integrar un software que permita manipular, tratar y mostrar los datos obtenidos.

Los datos de posicionamiento geográfico dentro del globo terrestre necesarios para la localización de un objeto son la longitud y la latitud, las cuales son medidas teniendo en cuenta el meridiano de Greenwich y la línea del Ecuador respectivamente. A su vez, para garantizar el monitoreo en tiempo real del movimiento, posición y trayectoria son necesarios como mínimo el curso, la velocidad, la fecha y la hora en cada lectura realizada.



**Figura 1.5** Meridiano de Greenwich y línea Ecuatorial

(Castro Correa, Sepúlveda Mora, Medina Delgado, Guevara Ibarra, & López Bustamante, 2019, págs. 145-157)

Como se observa en la Figura 1.5 el meridiano de Greenwich divide el planeta en dos hemisferios del polo norte a sur donde se obtiene coordenadas de longitud y mediante la línea Ecuatorial se tiene las coordenadas de latitud, el punto de unión entre las dos se denomina punto 0.

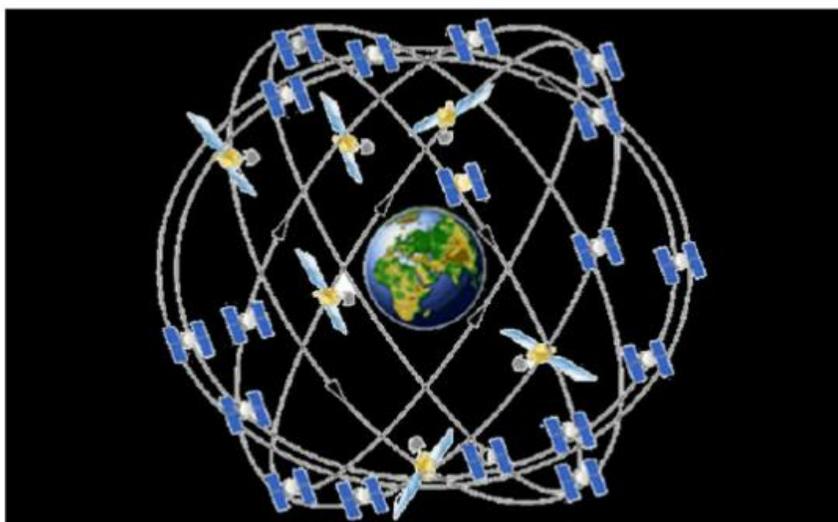
La ventaja principal del GPS es que se puede tener la información en todo tiempo y en cualquier característica climática, pero su desventaja es que se es vulnerable a tener interferencia tanto intencional o no intencional y así perder la información, por lo que una solución viable es la integración o fusión con otros sensores para evitar el problema.

### 1.7.1 DIVISIÓN GPS

El GPS se divide en tres partes: el segmento espacial, de control, y el de equipo de usuario.

### 1.7.1.1 Espacial

En el segmento espacial se cuenta con 24 satélites funcionales más tres que son de repuesto, están colocados en seis planos orbitales, y con cuatro satélites en cada uno de estos planos a una altitud aproximada de 20200 km desde la superficie de la tierra, las orbitas están nombradas por las letras A,B,C,D,E y F; además un satélite realiza dos revoluciones alrededor de la tierra en un día sideral o 23 horas y 56 minutos (Watson, 2016, págs. 38-50).



**Figura 1.6** Satélites en la Orbits  
(Watson, 2016, pág. 42)

En la Figura 1.6 se nota la distribución que tienen los satélites en el espacio.

### 1.7.1.2 De control

Son estaciones terrestres las cuales se encargan de enviar la información para el control de los satélites, además mediante este sistema se realiza el correspondiente mantenimiento mediante el control de las señales de radiodifusión. Se compone de varios centros de control de vigilancia global y de antenas ubicadas en la tierra para la comunicación con los satélites. El segmento terrestre de control se encarga de monitorear los datos que envían los satélites cada cierto tiempo, conociendo sobre el estado de los relojes y los parámetros de rotación terrestre (Berné Valero, Garrido Villén, & Capilla Romá, 2019, págs. 107-109).

### 1.7.1.3 Receptor

Son los dispositivos que se encuentran regados por todo el mundo, los cuales dan a conocer la ubicación exacta de donde se encuentran. La antena de estos aparatos es lo principal debido a que esta se encarga de recoger las señales que emiten los satélites desde el espacio, cuando existe un error de precisión, el receptor analiza la señal de cuatro en cuatro satélites y escoge la posición que menos error tenga.



**Figura 1.7** Módulo receptor GPS PARALLAX

(Vázquez Seisdedos, Palacios Meléndez, Córdova Rivadeneira, & Romero Paz, 2016, pág. 1)

En general los satélites envían una señal que capta el receptor, por lo que este conoce el tiempo en llegar la señal, que satélite le ha enviado y la posición en la que se encuentra, luego por medio de la triangulación entre varios satélites se puede conocer la posición exacta en el planeta, en la Figura 1.7 se muestra un módulo receptor GPS (Pérez Rodríguez, 2015, pág. 166).

## 1.7.2 FUNCIONAMIENTO DEL GPS

Su funcionamiento se basa en que el receptor GPS mide la distancia hasta los satélites, su medición se basa en el tiempo en que tarda en llegar la señal al receptor (conociendo que la señal viaja a la velocidad de la luz se realiza el cálculo de la distancia), al conocer la información de dos satélites se sabe que el receptor se encuentra dentro de una circunferencia que se forma cuando se interceptan las dos esferas, luego al recibir información de un tercer satélite la nueva esfera corta en dos puntos con la circunferencia anterior, teniendo solo un corte que no es

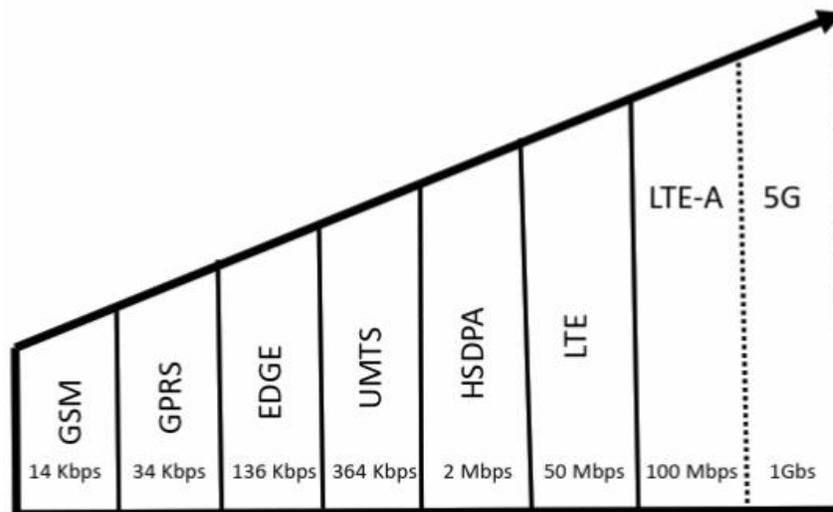
incoherente por lo que ya se tiene un pequeño volumen de la localización (Barra Zapata & Barra Zapata, 2011, pág. 383).

Por último, ya que los relojes entre el receptor y los satélites no están sincronizados, se obtiene la información de un cuarto satélite el cual ayuda a igualar los relojes y de esta manera ya no se tiene un pequeño volumen, sino que se convierte en un solo punto con información como: la altitud, longitud y latitud.

## **1.8 MÓDULO GSM (SISTEMA GLOBAL PARA LAS COMUNICACIONES MÓVILES) /GPRS (SERVICIO GENERAL DE PAQUETES VÍA RADIO)**

La comunicación a largas distancias ha sido muy importante desde hace mucho tiempo atrás, desde que intentaban comunicarse mediante tambores o señales de humo, hasta ahora que se tiene sistemas de comunicación móvil de datos inalámbrica, mediante llamadas de voz, mensajes de texto o mediante internet (Penttinen, The Telecommunications Handbook, 2015, págs. 281-370).

El sistema GSM tiene sus inicios en los años de 1980 al crear un grupo especial móvil en la conferencia Europea de Telecomunicaciones, este grupo tenía el trabajo de crear un sistema para comunicación único a 900 MHz, desde esos años se han creado múltiples modificaciones al sistema para poder satisfacer la gran demanda de usuarios y que sigue en crecimiento (Al Agha, Pujolle, & Ali-Yahiya, 2016, págs. 2-7).



**Figura 1.8** Redes de comunicación móvil  
(Al Agha, Pujolle, & Ali-Yahiya, 2016, pág. 6)

En la Figura 1.8 se tiene las modificaciones que se han ido dando a lo largo del tiempo con su respectiva velocidad en transmisión de datos desde la red GSM, hasta la red de quinta Generación (5G).

### 1.8.1 SERVICIOS GSM

La red GSM brinda tres tipos de servicios a sus usuarios:

- **Básicos:** Ejemplos de funciones básicas de comunicación son: el realizar llamadas de voz, mensajes de texto y hasta comunicarse mediante red.
- **Suplementarios:** Se trata de funciones que complementan a los servicios básicos al poder tener un registro, restringir ciertas llamadas o retener llamadas en espera, entre otras.
- **De Emergencia:** Se puede realizar la comunicación de datos digitales de dos maneras distintas, con un estándar de 9.6 kbps o con velocidades más altas para servicios de emergencia.

### **1.8.2 SERVICIO GENERAL DE PAQUETES VÍA RADIO (GPRS)**

Los sistemas GSM realizan la transmisión de datos por conmutación de circuitos, por lo que se asigna un canal a un solo usuario durante toda la llamada, por lo tanto, este servicio para el uso de internet es ineficiente. De este problema surge el sistema GPRS (General Packet Radio Service) en el cual múltiples usuarios pueden usar un canal físico, conectarse y desconectarse inmediatamente; logrando tiempos de acceso más cortos, mayor transferencia de datos y una simplificación del acceso a las redes de paquetes de datos (Das, 2016, págs. 40-58).

Con el sistema GPRS se tiene aproximadamente una velocidad de 115 kbps el cual no es suficiente para la utilización de aplicaciones de internet, por lo que nace el servicio de GPRS EDGE que mejora su velocidad de transmisión de datos logrando llegar a aproximadamente 250 kbps.

### **1.8.3 MÓDULO DE IDENTIFICACIÓN DE ABONADO (SUBSCRIBER IDENTITY MODULE SIM)**

Cuando inició la red GSM no existía tanto peligro o inseguridad como la de hoy en día que existe gran cantidad de virus maliciosos o malware, para ese tiempo la única forma de tener seguridad fue la de tener una autenticación del usuario para realizar la comunicación de datos. En la actualidad se cuenta con la llamada tarjeta SIM que contiene registro de la identidad del usuario y algoritmos de autenticación, hecha a base de normas ISO, que tiene ciertos contactos y vienen en diferentes tamaños desde la inicial 2FF, con aproximadamente 2,5 x 1,5 cm, 3FF (micro tarjeta) y 4FF (nano tarjeta) (Penttinen, Wireless Communications Security, 2016, págs. 60,61).

## **1.9 ALMACENAMIENTO DE DATOS (NUBES DE DATOS)**

En los últimos tiempos los servicios en la nube se han visto en crecimiento y con una gran demanda, por lo que algunos proveedores como Amazon, Microsoft, Google y Rackspace,

cuentan con una enorme infraestructura informática, de la cual depende para que la prestación de servicios sea confiable, donde se lleva a cabo los servicios como: servidores, matrices de almacenamiento y equipos de red (Lynn, Morris, & Kenny, 2018, pág. 123).

### **1.9.1 COMPUTACIÓN EN LA NUBE (CLOUD COMPUTING)**

A partir de mediados de la década de los años 70 la computación ha pasado por diferentes escalones de mejoramiento, como la disminución del tamaño de las computadoras personales, las aplicaciones y ahora la conexión a la nube mediante internet, donde se puede jugar con los datos o recursos al almacenarlos o compartirlos (Murugesan & Bojanova, 2016, págs. 1-14).

La denominada computación en la nube no tiene una definición como tal, más bien se refiere a un sistema que se relaciona directamente con los servicios que se ofrecen a través de internet, donde se tiene un proveedor que ofrece y controla estos servicios, por lo que se tiene la posibilidad de contratarlos de una forma flexible, en base a las necesidades y requerimientos del consumidor (Alda Montes, y otros, 2016, págs. 47-105).

### **1.9.2 CARACTERÍSTICAS DE LA COMPUTACIÓN EN LA NUBE**

La computación en la nube se ha convertido en uno de los temas más significativos en la actualidad debido a la importancia que tiene en la tecnología, arquitectura e incluso en la parte económica. En los últimos años gran cantidad de empresas han empezado a utilizar esta plataforma en la nube ya que cuenta con beneficios que incluyen estabilidad, escalabilidad, flexibilidad de servicios, entrega a pedido de recursos, eficiencia, ahorro de costos, entre otros (Ravindran & Prakash, 2016, págs. 167-188).

Este sistema ha cambiado por completo la industria y ha hecho que el compartir información o recursos sea mucho más fácil, flexible y sobre todo económico. Ahora los usuarios no se preocupan por invertir muchas cantidades de dinero en un espacio físico para poder realizar sus actividades, más bien contratan el servicio en la nube y así pueden utilizar este sistema en tiempo

real, almacenando información, creando sus propias aplicaciones con la facilidad de ingresar a la plataforma desde cualquier parte del mundo.

### 1.9.3 MODELOS DE SERVICIOS EN LA NUBE

La plataforma *Cloud Computing* tiene tres modelos de servicios disponibles para el uso de los usuarios:

- **Software como un servicio (SaaS):** Es un modelo que permite que las aplicaciones funcionen desde el proveedor de servicios, es decir que los beneficiarios tienen el acceso mediante un navegador o también un dispositivo móvil mediante una aplicación, como por ejemplo Gmail, Google Apps, office 365, entre otros (Yeluri & Castro Leon, 2014, pág. 3).
- **Plataforma como un servicio (PaaS):** Los usuarios pueden tener el acceso a la plataforma en la nube para desarrollar sus propias aplicaciones utilizando bases de datos o servidores web, aunque existen ciertas restricciones para algunas aplicaciones, por ejemplo, se tiene Microsoft Azure y Google App Engine (Alda Montes, y otros, 2016, págs. 183-212).
- **Infraestructura como un servicio (IaaS):** Esta plataforma cuenta con la infraestructura que es compartida, como las redes, los servidores y el almacenamiento, por ejemplo, Amazon, Claro, Strato, HP cloud, entre otras, donde a los clientes se les cobra dependiendo del uso que se da (Joyanes Aguilar, 2015, pág. 173).

### 1.9.4 MÁQUINA A MÁQUINA (M2M)

Máquina a máquina (M2M), es el sistema que permite la conexión entre dispositivos, con esta tecnología se admite el intercambio automatizado de datos, además de poder rastrear o monitorear ciertos dispositivos, realizar operaciones de emergencia, entre otras. Los operadores

de satélites indican que la conectividad en toda la tierra se puede dar en un 90% en maquina a máquina (Minoli, 2015, págs. 222-232).

### 1.9.5 BASES DE DATOS

Existen diferentes bases de datos que pueden ser utilizadas, por ejemplo:

- **SQL:** Se trata de bases de datos relacionales, las cuales son la parte principal de algunas plataformas de computación, por ejemplo, Microsoft SQL server.
- **NoSQL:** Este tipo de bases de datos es no relacional, pero si es un sistema altamente escalable y con un alto rendimiento, en este caso es una base de datos la cual no utiliza el lenguaje SQL como principal; se tiene cuatro diferentes categorías: almacenes clave-valor, almacenes de documentos, almacenes de grafos, y familia de bases de datos columnares (Joyanes Aguilar, 2015, pág. 242).
- **En memoria:** Son bases de datos que a diferencia de los sistemas tradicionales que realizan el almacenamiento en disco, estos realizan todos sus procesos en memoria principal utilizada como almacenamiento de datos, teniendo así una característica principal como es una muy mejorada velocidad de proceso. Existen diversos ejemplos como: Oracle, IBM solidDB, Hackaton de Microsoft, entre otras (Joyanes Aguilar, 2015, pág. 250).
- **Legado:** O plataformas “heredadas”, son bases de datos antiguas, que se han complementado con bases de datos actuales como las relacionales o analíticas, las cuales siguen en funcionamiento hasta el día de hoy.
- **En la nube:** En la actualidad existen diversas bases de datos que funcionan en la nube lo que da lugar a la “Database as a Service”, por ejemplo, se tiene a: Amazon, Google App Engine, Cloudant, entre otras.

### 1.9.6 HERRAMIENTAS DE DESARROLLO

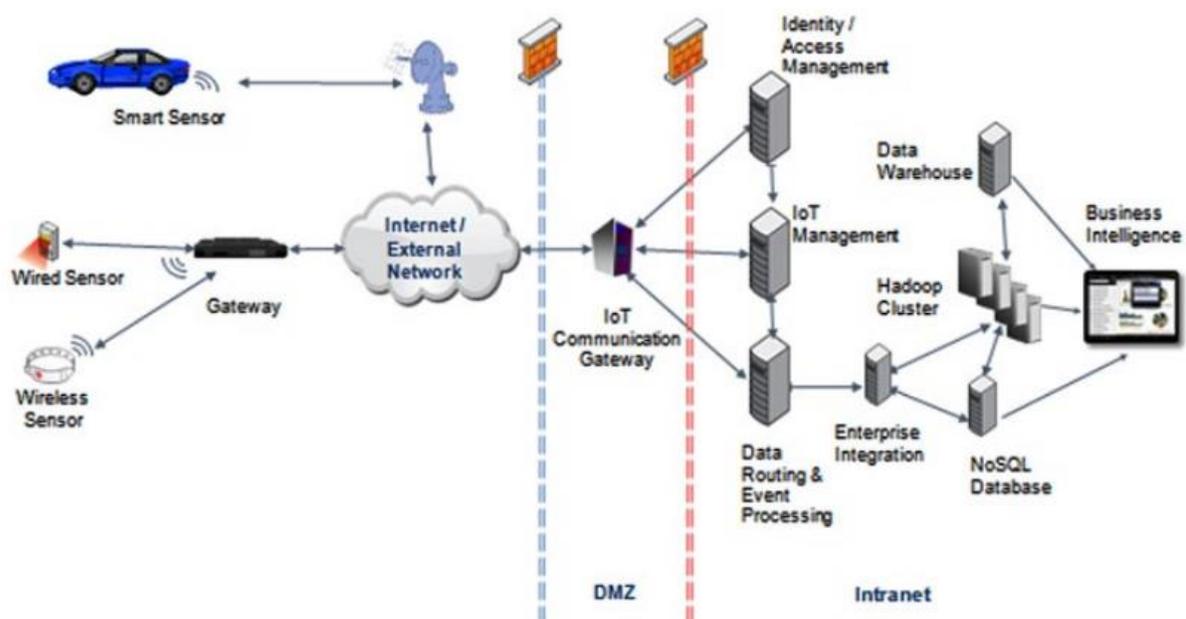
Son herramientas también conocidas como IDE (Integrated Development Environments), las cuales mediante un lenguaje de programación se realiza una escritura, compilación, prueba y mantenimiento, para la creación de una aplicación o página web; entre las principales se tiene tres:

- **Visual Studio:** Se trata del IDE de Microsoft que utiliza lenguajes en C, C++, Javascript, HTML, VB.NET, entre otras; con las que se puede desarrollar sitios web, aplicaciones de escritorio y aplicaciones web. Esta herramienta cuenta con esquemas de bases de datos, diseño y muchas más (Fylaktopoulos, Goumas, Skolarikis, Sotiropoulos, & Maglogiannis, 2016, págs. 7,8).
- **Eclipse:** Se basa principalmente en Java el cual acepta varios lenguajes de programación como Ada, ABAP, C, C ++, JavaScript, Lasso, PHP, Prolog, Python, R , Ruby, Scala, Groovy, y algunos más; esta plataforma es de código abierto y permite la extensión utilizando cualquier lenguaje de programación (Fylaktopoulos, Goumas, Skolarikis, Sotiropoulos, & Maglogiannis, 2016, pág. 7).
- **Netbeans:** Es muy similar a la herramienta Eclipse, que se basa en Java, pero además el Netbeans puede crear aplicaciones en Java swing, así como también actualizar la plataforma usando modelos guardados.

### 1.10 INTERNET DE LAS COSAS (INTERNET OF THINGS)

En los últimos años, gracias al rápido y continuo desarrollo tecnológico se hizo posible el denominado internet de las cosas (IoT), el cual se entiende como la interconexión entre los dispositivos de todo el mundo que cuentan con sensores, actuadores, tecnología de comunicación y el internet (Castro Correa, Sepúlveda Mora, Medina Delgado, Guevara Ibarra, & López Bustamante, 2019, pág. 147).

Debido a que la demanda de los sensores ha ido en crecimiento durante los últimos años, se ha realizado una mayor cantidad de investigación para su progreso, por lo que los sensores cada vez se fabrican de un menor tamaño y son adaptables en espacios más reducidos. Muy pronto se espera que existan cientos de miles de millones de sensores en el mundo, además, NoSQL y Hadoop las dos bases de datos más importantes se pusieron en funcionamiento principalmente para procesar y analizar sitios web o redes sociales, pero resulto que son de gran ayuda para transmitir datos de vehículos, dispositivos móviles, monitoreo para salud, entre otros (Stackowiak, Licht, Mantha, & Nagode, 2015, págs. 20-22).



**Figura 1.9** Estructura del IoT  
(Stackowiak, Licht, Mantha, & Nagode, 2015, pág. 22)

En la Figura 1.9 se muestra la estructura o los componentes principales del Internet de las cosas, donde se recibe datos de sensores en vehículos, dispositivos móviles, entre otras, para luego analizar y tener plataformas de gestión de datos.

La visión del internet de las cosas es la de crear un ambiente totalmente interconectado de donde surja entornos inteligentes, es decir edificios inteligentes, medios de transporte público inteligentes, sanitarios inteligentes, ciudades inteligentes y mucho más; todo esto mediante la tecnología invisible de las redes inalámbricas conectados mediante un IP único para cada objeto (Bibri, 2015, pág. 6).

La manera de funcionamiento del IoT se basa en que cada objeto o dispositivo en el mundo que está conectado a internet tiene un número único de identificación también llamado IP, mediante este se puede acceder para enviar o recibir datos de información y así poder realizar alguna acción específica. El internet de las cosas ha abierto puertas hacia varios trabajos en la tecnología, así como para resolver problemas.

## **1.11 SISTEMA DE VISUALIZACIÓN REMOTA**

En los últimos tiempos la ubicación se ha convertido en un tema principal, y uno de los mayores promotores ha sido Google, debido que ahora la mayoría de los sitios web cuentan con un enlace a *Google Maps* donde se indica la ubicación exacta dentro de un mapa. El uso de *Google Maps* es muy amplio por lo que se puede utilizar para varias tareas debido a su soporte en mapas base o satélites imágenes y a pesar de que no es una herramienta totalmente gratuita hay un límite de 25 000 cargas de mapas por día que es suficiente para los desarrolladores (Dincer & Uraz, 2013, págs. 28-52).

### **1.11.1 MAPAS WEB**

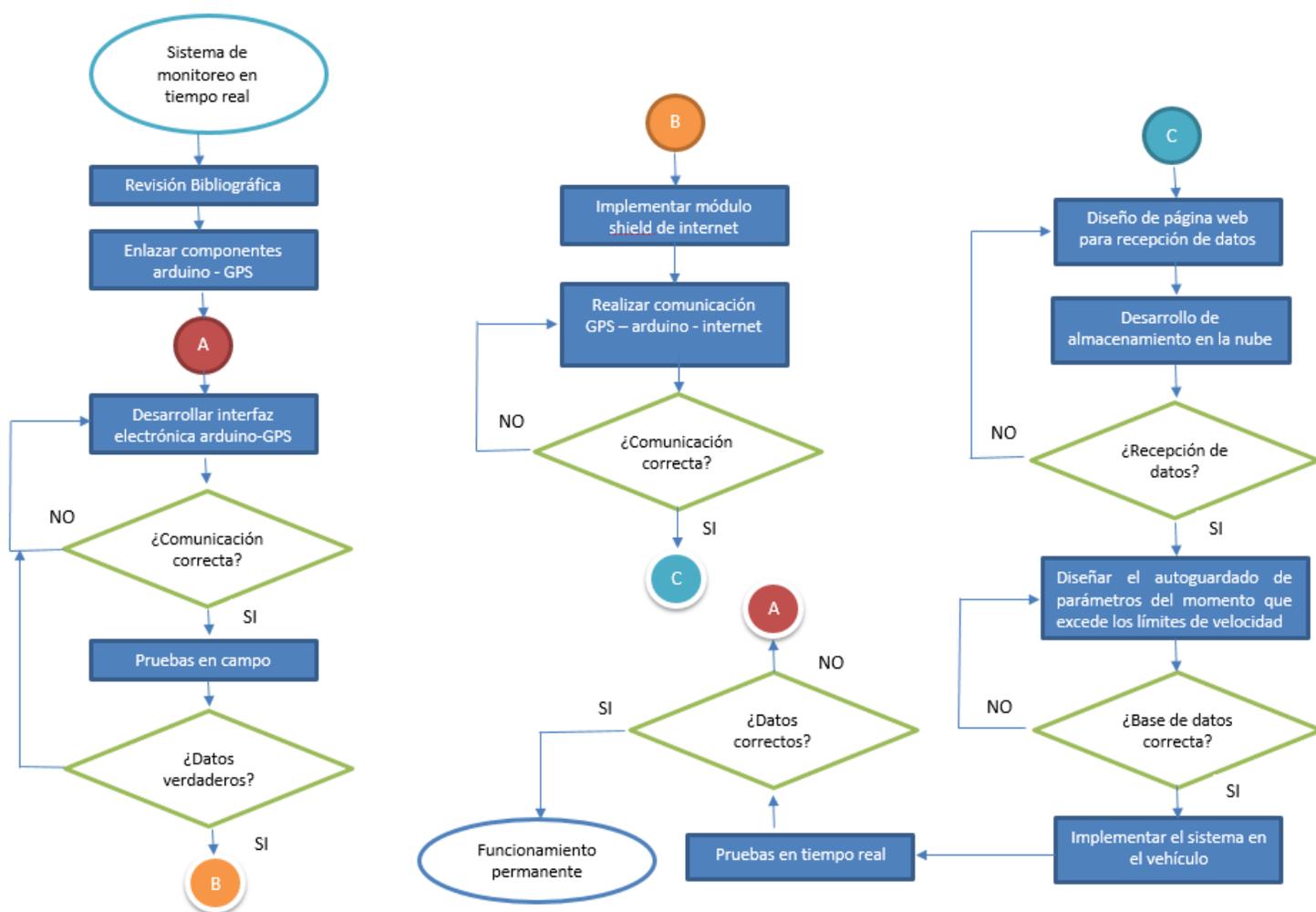
En el año 2005 apareció el primer servicio de mapas en la historia, era la primera vez que cualquier persona con acceso a internet y sin ningún costo adicional, podía ver las imágenes de territorios de cualquier parte del mundo con imágenes satelitales. Los primeros mapas web estuvieron disponibles gracias a *Google Maps* y *Google Earth*, aunque más adelante se sumaron algunas más, todas rápidamente fueron utilizadas por millones de personas para diferentes usos que implican la búsqueda de una ubicación (Reyes Nuñez, 2018, págs. 87,88).

La información relacionada con la ubicación física de una persona o un dispositivo se almacenan como datos, que luego se analizan y observan mediante plataformas web como *Google Earth* o *Google Maps*, aquí los datos se cargan y se puede observar las ubicaciones tanto anteriores que visitó o la ubicación incluso actual, todo esto gracias al sistema de GPS (Harrington & Cross, 2015, págs. 41,42)

## CAPÍTULO II

### 2 MATERIALES Y MÉTODOS

El objetivo del presente estudio es el de diseñar un sistema electrónico de seguridad el cual sirve para realizar un monitoreo en tiempo real de la velocidad de un vehículo, mediante el uso de GPS y una plataforma web, para conocer las condiciones en las que se tiene un exceso de velocidad.



**Figura 2.1** Flujograma

En la Figura 2.1 se presenta un flujograma detallado de los pasos seguidos para lograr el objetivo propuesto donde se tiene que, como primer punto se desarrolló una interfaz electrónica entre

Arduino y GPS para obtener datos como ubicación, fecha, hora, velocidad; luego se implementó un módulo *Shield* para tener comunicación con un servidor externo y por último se utilizó una nube de almacenamiento para registrar los datos donde se puede tener acceso desde ubicaciones remotas.

## **2.1 MATERIALES Y EQUIPOS**

Al diseñar e implementar un sistema de seguridad, es decir un prototipo en un vehículo para realizar las pruebas tanto en tiempo real como en una base de datos, se utilizó algunos materiales y equipos que a continuación se describen detalladamente.

### **2.1.1 ARDUINO**

Arduino es sin duda uno de los microcontroladores más usados a nivel mundial debido a que es una plataforma de prototipos electrónica de código abierto, es decir que se puede encontrar códigos de programación tanto en su propia página web como en otras páginas de internet.

Además, su hardware y software son fáciles de usar, así como también cuenta con módulos *Shield* que son compatibles con las placas Arduino, pudiendo conectarse y ampliar sus funciones.

Es por tales razones que para el presente trabajo se utilizó una placa de Arduino UNO (ATmega328) cuya velocidad de CPU y su memoria es suficiente para realizar el sistema, además, no se necesita tantas entradas ni salidas, más bien su tamaño reducido es una ventaja para crear el prototipo electrónico.

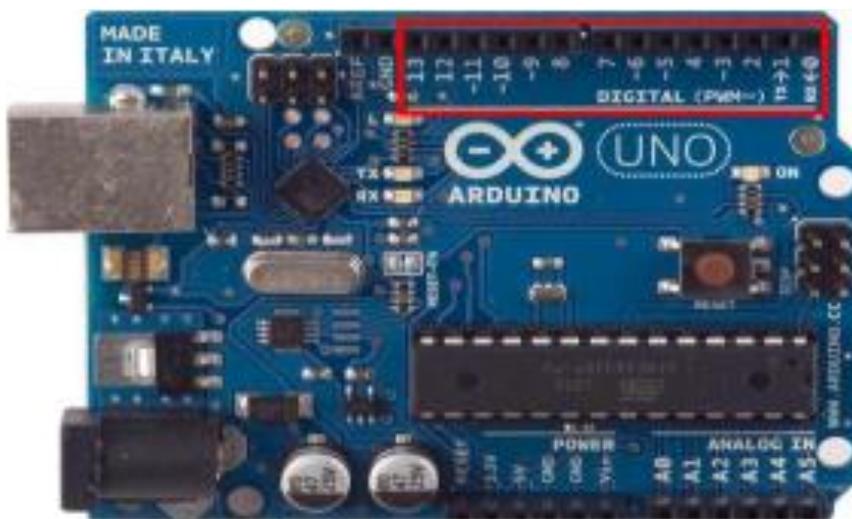
Las características principales que hay que tomar en cuenta se muestra en la siguiente Tabla 2.1:

**Tabla 2.1** Características Placa Arduino UNO

<b>Microcontrolador:</b>	ATmega328
<b>Voltaje de Funcionamiento:</b>	5 V
<b>Voltaje de Entrada:</b>	7-12 V
<b>Velocidad CPU:</b>	16 MHz
<b>Entradas Analógicas:</b>	6
<b>Entradas Digitales:</b>	14 (6 utilizables como PWM)
<b>EEPROM:</b>	1 kB
<b>SRAM:</b>	2 kB
<b>Flash:</b>	32 kB
<b>USB:</b>	Regular
<b>UART:</b>	1

La alimentación de la placa de Arduino UNO se puede realizar mediante conexión USB al computador o mediante una fuente externa.

### 2.1.1.1 Pines Arduino uno

**Figura 2.2** Pines digitales Arduino UNO

(Universidad de Cádiz, 2012, pág. 3)

Como se puede notar en la Figura 2.2 se tiene 14 pines digitales desde el 0 al 13 que pueden recibir un máximo de 40 mA funcionando como entrada o salida, aunque algunos pines tienen funciones especiales, donde:

- **Pin 0.-** o pin RX se utiliza para la recepción de datos
- **Pin 1.-** o pin TX es usado para el envío o transmisión de datos
- **Pin 3,5,6,9,10,11.-** son los 6 pines que se puede utilizar como PWM o modulación por ancho de pulsos.



**Figura 2.3** Pines Análogos Arduino UNO  
(Aprendiendo Arduino, 2019, pág. 1)

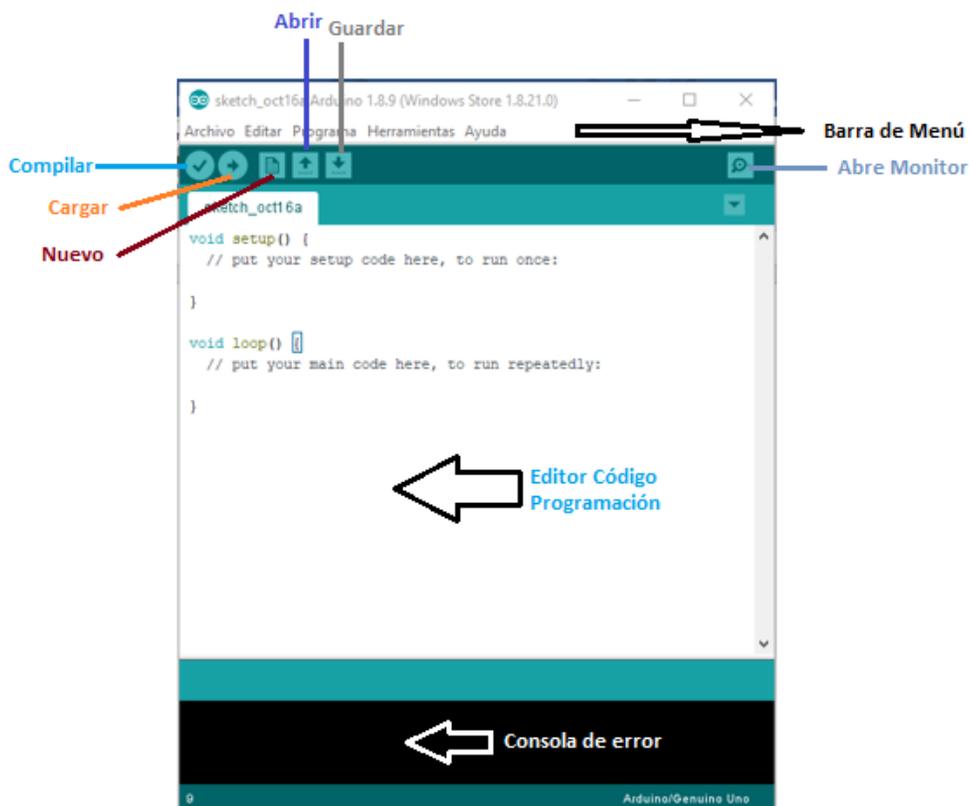
En la Figura 2.3 se nota los pines analógicos del Arduino UNO que van desde el A0 al A5, es decir cuenta con 6 entradas analógicas, las cuales por defecto se tiene una tensión de 5V.

En este caso para comunicar el Arduino UNO con el Módulo GPS se utiliza pines digitales que sirven para el envío y recepción de la información, y adicionalmente el pin GND (tierra) de Arduino.

### 2.1.1.2 Arduino IDE

El IDE es el software oficial de la plataforma Arduino, el mismo que es compatible con casi todas las placas, donde se realiza la programación en un lenguaje *processing* que el software se encarga de convertir en un lenguaje C para enviar a la placa Arduino. En este IDE se hace la petición y se puede obtener los datos del GPS, además es de fácil instalación y manejo ya que

se puede instalar en todos los computadores debido a que viene definido una versión para cada sistema operativo.



**Figura 2.4** Partes del Software Arduino IDE

En la Figura 2.4 se muestra la pantalla principal del software IDE Arduino y sus partes.

### 2.1.2 MODULO SIM 808 GSM/GPRS + GPS

Para realizar el monitoreo de la ubicación y velocidad de un vehículo en tiempo real se usó el módulo SIM 808 GSM/GPRS + GPS que está basado en el chip SIMCOM 808. El cual además de contar con los servicios de enviar y recibir tanto llamadas como mensajes o conectarse a internet utilizando una tarjeta SIM de cualquier operadora, también tiene la función de conocer la ubicación mediante la tecnología GPS, por lo que es un elemento principal para lograr el objetivo del presente trabajo.



**Figura 2.5** SIM 808 GSM/GPRS + GPS

En la Figura 2.5 se ilustra el módulo SIM 808 GSM/GPRS +GPS, donde se puede notar la placa como tal, a su izquierda la antena del sistema GPS y en la parte inferior derecha la antena para el sistema GSM/GPRS.

A continuación, en la Tabla 2.2 se indica las características técnicas más importantes que presenta este módulo *Shield*.

**Tabla 2.2** Características técnicas del módulo SIM 808 GSM/GPRS + GPS

<b>SIM 808 GSM/GPRS + GPS</b>	
Bandas de funcionamiento	Cuádruple 850/900/1800/1900 MHz
Rango de Tensión de Alimentación	3.4 v – 4.4 v
Clase 4	(2 W 850/900 MHz)
Clase 1	(1 W 800/1900 MHz)
Control	Mediante comandos AT
Tipo receptor GPS	22 tracking, 66 canal de adquisición GPS
Sensibilidad	Seguimiento: -157 dBm Arranque en frío: -144 dBm
Tiempo de inicio	Arranque en frío: 100 s Arranque en caliente: 1s
Precisión	Menor a 2,5 m
Velocidad de transmisión	Por defecto 9600bps
GPRS conectividad	Máx. 85.6 kbps

**Fuente:** (Módulo Celular GSM GPRS/GPS – SIM808 con Antenas GPS y GSM, 2019, pág. 1)



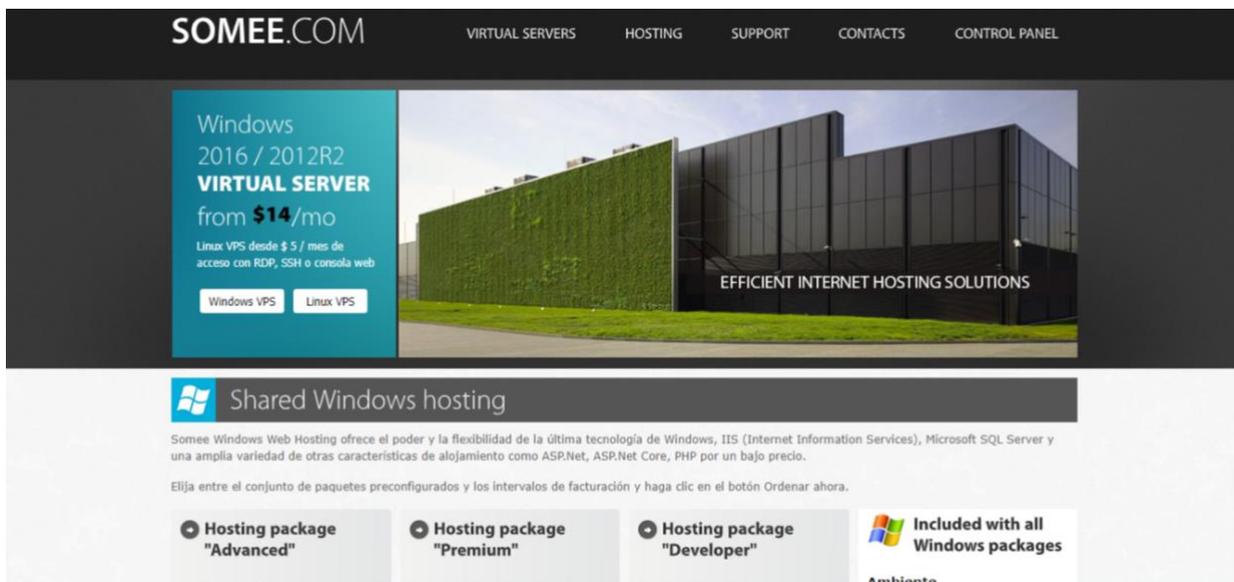
**Tabla 2.3** Características GPRS modulo SIM 808

<b>GPRS MODULO SIM808</b>	
Conectividad	GPRS multi-slot class 12(predeterminado) GPRS multi-slot class 1~12 (opcional)
Transferencia de enlace descendente de datos	máx. 85,6 kbps
Transferencia de enlace ascendente de datos	máx. 85,6 kbps
Esquema de codificación	CS-1, CS-2, CS-3 y CS-4
Tasas de transmisión de CSD	2.4,4.8,9.6,14.4 kbps

**Fuente:** (Módulo SIM808 con antena GSM Y GPS, 2019, pág. 1)

### 2.1.3 PÁGINA WEB Y BASE DE DATOS

Para obtener la información enviada del GPS y guardarla, primeramente, se necesita construir una página web, la cual permitirá monitorear el vehículo en tiempo real conociendo su ubicación y velocidad, además, se necesita almacenar la información del momento en que el vehículo se excedió el límite de velocidad con las características del incidente.



**Figura 2.7** Pantalla principal hosting  
(Somee, 2020, p. 1)

Se utilizó **somee.com** que es un *hosting*, es decir, una página para el alojamiento web con interesantes prestaciones, donde se carga el diseño de la página previamente realizada en el

programa Visual Studio y la base de datos realizada en Microsoft SQL que sirve para el almacenamiento de toda la información recibida, en la Figura 2.7 se indica la pantalla principal del *hosting*.

The image shows four hosting packages from Somee (2020):

- Gratis .Red Hospedaje** (highlighted with a red box):
  - Free ASP.Net web hosting
  - 150MB storage, 5GB transfer
  - ASP.Net 2.0-4.8, ASP.Net Core 2.2
  - 15MB MSSQL 2014/2016/2019
  - Free third level domain
  - FTP access
  - Price: \$ 0.00
- SRA SQL Hospedaje**:
  - MS SQL 2019/2016/2014 - 2008
  - 500MB data, 750MB log size
  - 1500MB backup storage
  - Auto backup every 12 hours
  - Manual backup/restore/attach
  - Remote database access
  - Price: \$ 7.85
- Ventanas hospedaje**:
  - 1000MB premium storage
  - 50GB premium transfer
  - ASP.Net 2.0-4.8 / .Net Core 2.2
  - Dedicated application pool
  - 100MB MSSQL 2019/16/14/12
  - 99.9% Uptime guarantee
  - Price: \$ 7.95
- Virtual Servidores**:
  - Windows or Linux
  - Full Admin/root access
  - RDP/SSH and Web console
  - Automated Setup/Upgrades
  - Daily backups
  - High quality hardware
  - Price: \$ 17.49

**Figura 2.8** Paquetes del hosting  
(Somee, 2020, p. 1)

Para el presente estudio se realizaron las pruebas con el paquete del *hosting* gratuito, el cual tiene prestaciones suficientes para el proyecto, en la Figura 2.8 se presenta los paquetes que ofrece, sus características y costos mensuales.

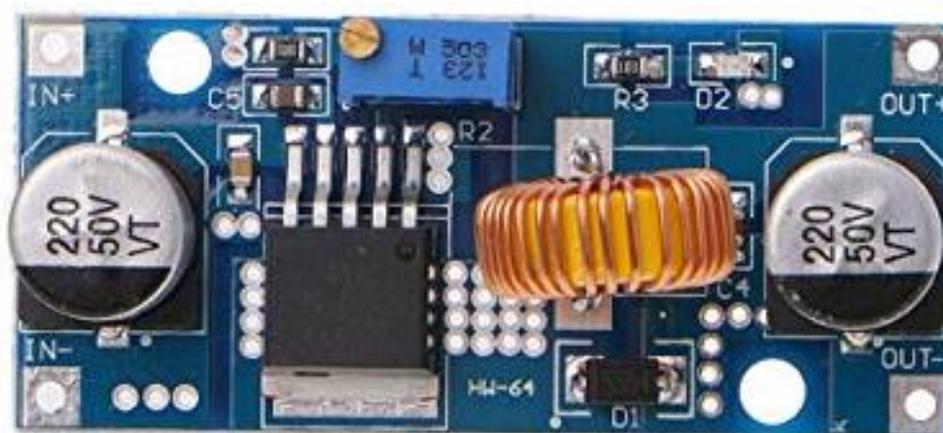
Las principales características que presenta el *hosting* gratuito utilizado son las indicadas en la siguiente Tabla 2.4:

**Tabla 2.4** Características Paquete Gratuito Hosting

HOSTING GRATUITO	
Capacidad de Almacenamiento	150MB
Transferencia Mensual	5GB
Cantidad de Dominios	1
Compatibilidad	ASP.Net, ASP.Net Core, PHP, SQL
Tamaño Base de Datos	15MB
Tamaño de Registro	20MB
Tamaño Almacenamiento de Respaldo	40MB

**Fuente:** (Somee, 2020, p. 1)

### 2.1.4 FUENTE DE ALIMENTACIÓN DC-DC XL4015



**Figura 2.9** Fuente de alimentación

Para realizar la conexión al vehículo, fue necesario contar con una fuente de alimentación tanto para Arduino como para el módulo SIM 808 GSM/GPRS/GPS, por lo que se utilizó la corriente de la batería pasando por un regulador de voltaje (Figura 2.9). El cual sirve para disminuir la corriente que se envía a los equipos, cuenta con 2A y 5V a la salida del regulador, lo necesario para el correcto funcionamiento del sistema realizado.

Las características por las que se utilizó el regulador XL4015 se muestran en la Tabla 2.5 a continuación:

**Tabla 2.5** Características Regulador de Voltaje

<b>REGULADOR DC-DC XL4015</b>	
Voltaje de Entrada	8 - 36V
Voltaje de Salida	1.25 – 32V
Corriente de Salida	0 – 5 A
Potencia de Salida	75W
Eficacia de conversión	96%
Temperatura de Funcionamiento	-40 a +85 °C
Dimensiones	54 x 23 x 15 mm (largo x ancho x alto)
Conexión	Soldadura

**Fuente:** (Probots, 2020, pág. 1)

## 2.2 MÉTODOS

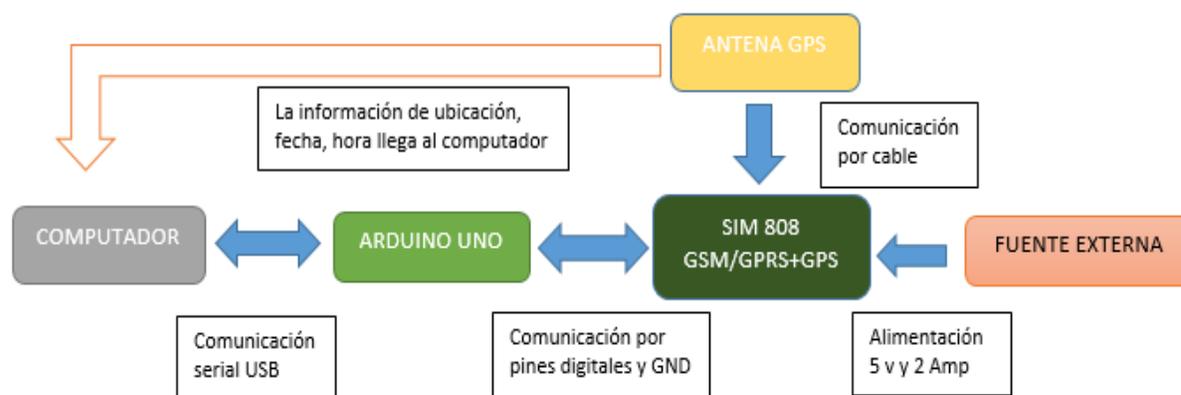
Al iniciar el presente trabajo se aplicó el método de investigación bibliográfica, realizando búsquedas de información tanto en libros, artículos científicos, revistas, páginas web, entre otras; las cuales permitieron conocer a fondo acerca de la problemática del exceso de velocidad en el Ecuador, así como los dispositivos electrónicos que se pueden usar para diseñar el sistema de seguridad.

Además, se empleó un método de enfoque experimental el cual consiste en manipular una o más variables para obtener un resultado, en este caso fue el diseño e implementación en el vehículo de un sistema electrónico de seguridad para el monitoreo del exceso de velocidad. A continuación, se detalla cada paso para cumplir el objetivo.

### 2.2.1 DESARROLLO INTERFAZ ELECTRÓNICA ARDUINO – GPS

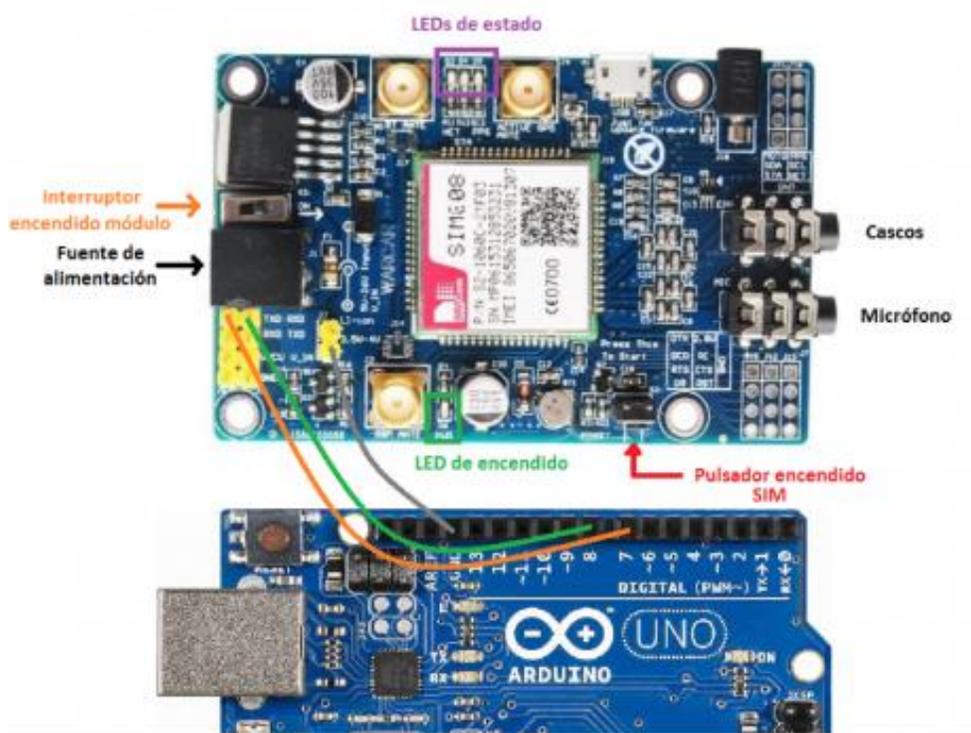
En esta etapa se llevó a cabo la interconexión entre la plataforma de prototipos electrónica Arduino y el módulo GPS SIM 808, para la obtención de la ubicación, fecha, hora y velocidad, tanto de manera física entre placas que es el hardware como por medio del lenguaje de programación que es el software.

#### 2.2.1.1 Enlace Arduino – SIM 808 GSM/GPRS + GPS



**Figura 2.10** Enlace Arduino UNO - SIM 808

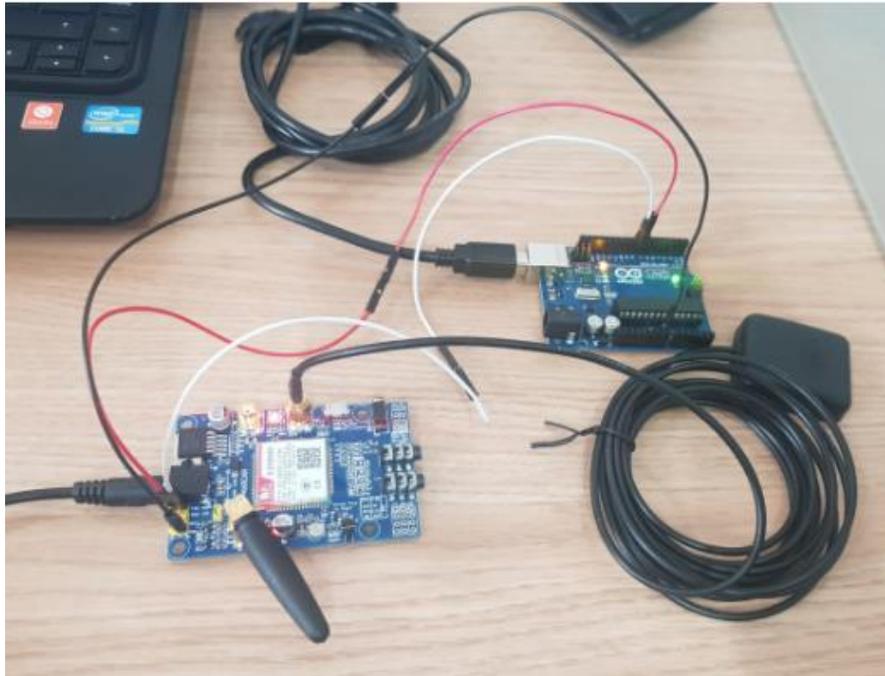
Como se ha mencionado antes, el primer paso fue conectar y entrelazar los elementos para obtener la información necesaria, para lo que se realizó la conexión. Empezando desde el computador conectado al Arduino UNO mediante comunicación serial USB, luego mediante cables macho-hembra comunicado con el SIM 808 GSM/GPRS+GPS y por último conectado la antena GPS y una fuente externa de 5V y 2A, como se muestra en el diagrama de la Figura 2.10.



**Figura 2.11** Conexión entre Arduino Uno y la SIM 808 (PROMETEC, 2019, pág. 1)

En la Figura 2.11 se muestra la forma de conexión física donde se conecta los pines 7 y 8 del Arduino a los pines de Tx y Rx del módulo *Shield* respectivamente, adicionalmente se conecta el pin GND entre los dos. Hay que tomar en cuenta que tanto el Arduino como el módulo *Shield* deben tener una fuente de alimentación por separado para su correcto funcionamiento.

Adicionalmente tener presente que el módulo SIM 808 cuenta con un interruptor de encendido que hay que mantener presionado por 3s.

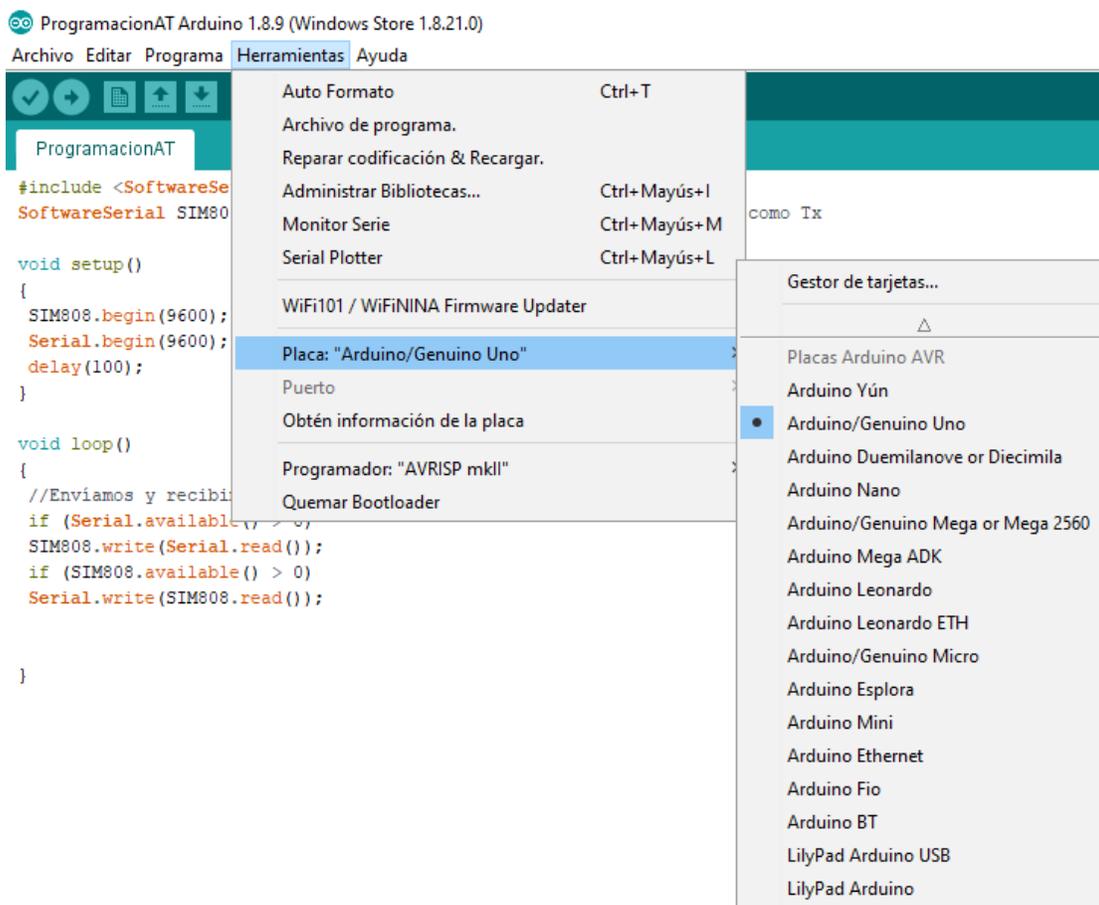


**Figura 2.12** Conexión física PC – Arduino - SIM 808

La conexión física completa entre todos los componentes para obtener la información se muestra en la Figura 2.12, donde se encuentra conectado al computador para su programación, entre módulos y a una fuente externa, los leds deben estar encendidos en ambos dispositivos.

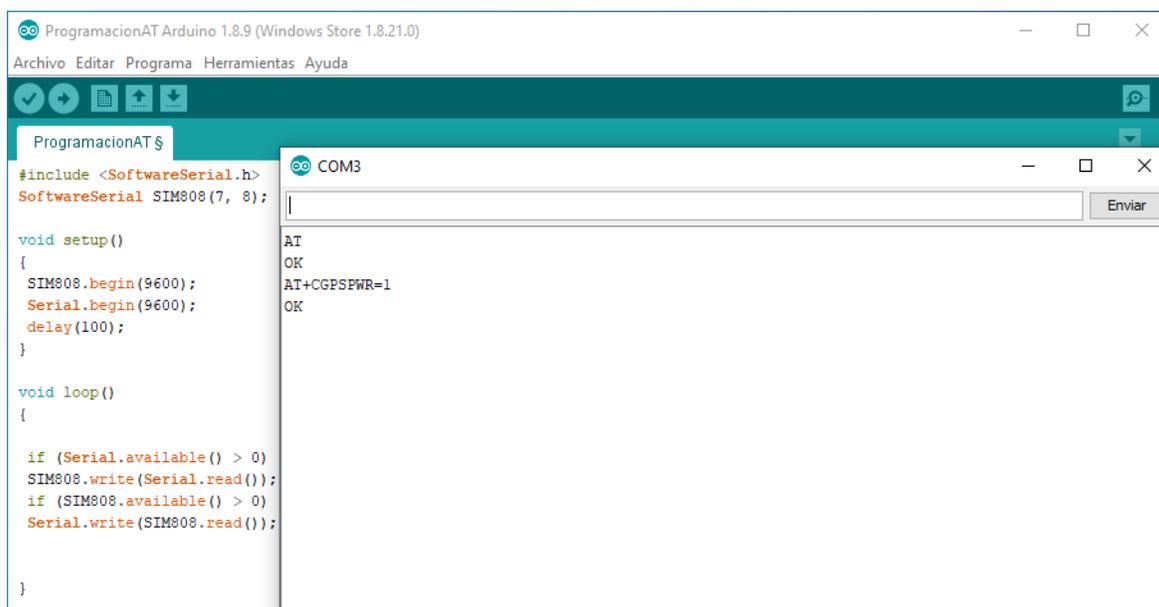
### **2.2.1.2 Comunicación GPS mediante comandos AT**

Luego de haber realizado la conexión física como se mostró anteriormente, se procede a verificar el estado del módulo, así como la información receptada por el GPS mediante código de programación en el IDE.



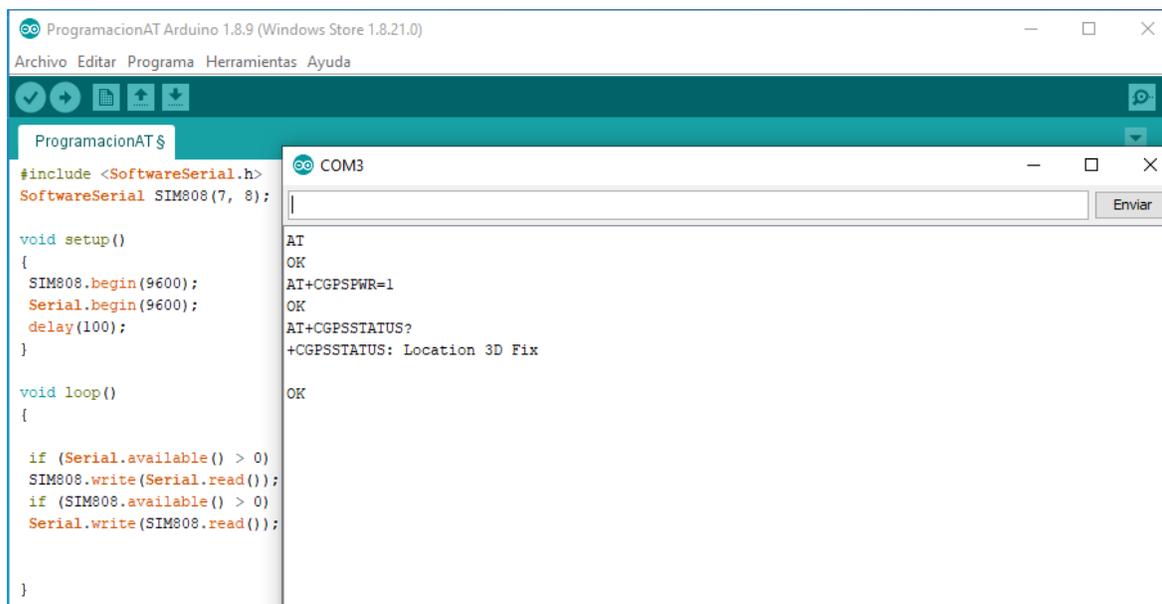
**Figura 2.13** Placa Arduino y Puerto COM

- Primeramente, dentro del IDE de Arduino se debe escoger en herramientas/placa el tipo de Arduino que se va a usar, en este caso se usó “Arduino/Genuino Uno”, así también el puerto COM que se va a usar dentro de la misma pestaña de herramientas, como se indica en la Figura 2.13.
- Luego se realiza la programación, claro está, ejecutando primero la compilación para verificar que este correcto y posteriormente cargar a la placa Arduino.



**Figura 2.14** Encender GPS comandos AT

- c. A continuación, como se muestra en la Figura 2.14 se abre el panel de comandos y mediante la palabra “AT” se comprueba el estado de la comunicación con el módulo, si esta resulta afirmativa se obtiene una respuesta de “OK”. Seguidamente, se enciende el GPS del módulo utilizando el comando “AT+CGPSPWR=1”.



**Figura 2.15** Estado GPS comandos AT

- d. Posteriormente, se pueden realizar consultas al módulo sobre su estado, o si encontró la información necesaria para definir en qué ubicación se encuentra, por lo que es necesario el uso del comando “AT+CGPSSTATUS” y como se indica en la Figura 2.15 se debe tener una respuesta de “+CGPSSTATUS: Location 3D fix” para poder continuar.

The screenshot shows the Arduino IDE interface with a code editor on the left and a serial monitor on the right. The code editor contains the following code:

```

#include <SoftwareSerial.h>
SoftwareSerial SIM808(7, 8);

void setup()
{
  SIM808.begin(9600);
  Serial.begin(9600);
  delay(100);
}

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

```

The serial monitor shows the following output:

```

AT
OK
AT+CGPSFWR=1
OK
AT+CGPSSTATUS?
+CGPSSTATUS: Location 3D Fix
OK
AT+CGPSINF=0
+CGPSINF: 0,22.037600,7806.767500,2046.700000,20191107153327.000,0,8,1.018600,290.970001

```

**Figura 2.16** Datos GPS comandos AT

- e. Finalmente, para obtener los datos del GPS se usa el comando “AT+CGPSINF=0” el cual arroja como resultados varios segmentos de números, para identificarlos es necesario tomar en cuenta las comas que los separan, en el caso de la Figura 2.16 la información proporcionada es:

- 0, Modo
- 22.037600, Latitud
- 7806.767500, Longitud
- 2046.700000, Altitud
- 20191107153327.000, Horario UTC
- 0, Tiempo de Respuesta
- 8, Número de satélites
- 1.018600, Velocidad en nudos
- 290.970001 Curso

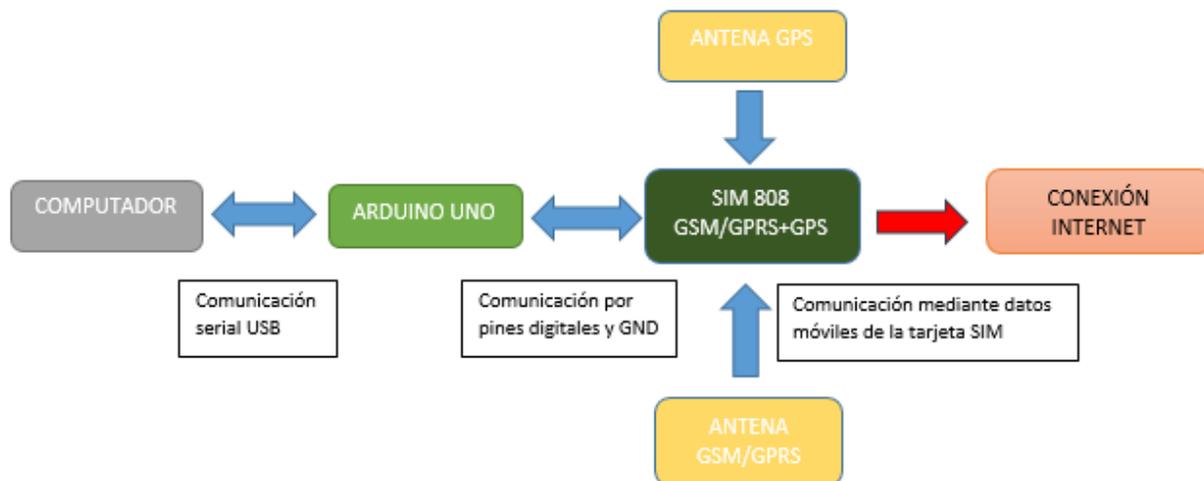
### 2.2.1.3 Pruebas de campo

Para la prueba de campo se realizó la conexión del Computador – Arduino – SIM 808 y se obtuvo los datos del GPS, verificando con la longitud y latitud el punto en el que se encuentra los dispositivos, así como su velocidad. Fue indispensable realizar los ensayos en tres diferentes lugares para verificar que la información recibida es correcta y poder continuar con el proyecto (la ubicación se puede confirmar ingresando la longitud y latitud en *Google Maps*).

## 2.2.2 IMPLEMENTACIÓN MODULO DE INTERNET

Luego de haber obtenido toda la información del módulo GPS, se necesita enviar mediante internet a una plataforma externa, por lo que en esta etapa se adicionó el módulo que usa la tecnología GPRS a Arduino, seguidamente se presenta su forma de conexión y funcionamiento para este proyecto.

### 2.2.2.1 Comunicación GPS-Arduino-internet



**Figura 2.17** Enlace Arduino - GPS - Internet

La Figura 2.17 muestra un diagrama donde se detalla lo realizado en este paso, es necesario un computador conectado a Arduino UNO que a su vez se conecta al SIM 808 GSM/GPRS/GPS y mediante programación en el IDE se envía la información receptada por el GPS hacia una

conexión externa, es decir, a una página web con base de datos, gracias al sistema GPRS que utiliza datos móviles de una tarjeta SIM.

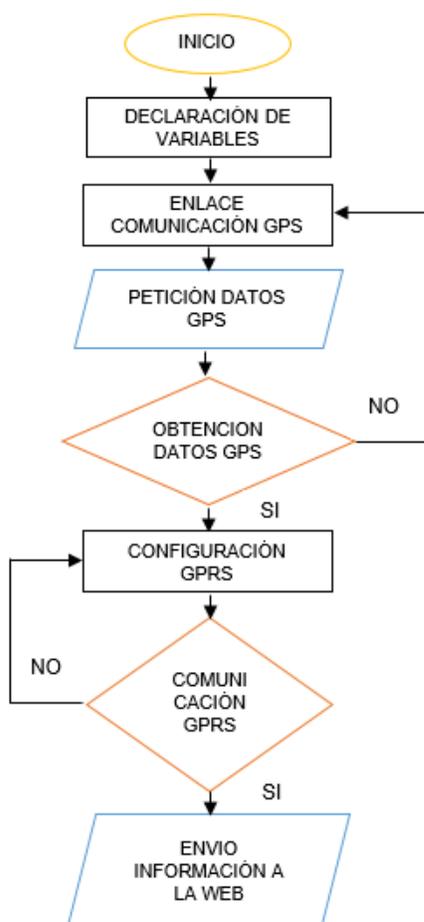


**Figura 2.18** Ranura tarjeta SIM

- a. Como primer paso para el uso de la tecnología GPRS en la SIM 808 se colocó una tarjeta SIM de la operadora CLARO en la parte posterior del módulo *Shield*, esta debe mantener activo un paquete de datos móviles, en este caso se inició con 1000 MB para poder enviar la información receptada por el GPS a la página web. En la Figura 2.18 se nota la ranura donde se colocó la tarjeta SIM.
- b. Para la conexión a la tecnología GPRS no son necesarias más conexiones físicas de las que se mencionó anteriormente para la conexión al GPS (Tx, Rx y GND), solamente debe contener la antena de GSM/GPRS. Finalizando así la estructura física (hardware) del sistema por lo que los siguientes pasos son de programación.
- c. Principalmente los datos obtenidos del GPS deben estar almacenados en tres variables dentro de Arduino: latitud, longitud y velocidad.
- d. Para empezar a hacer uso de la red GPRS y poder enviar los datos mediante Internet se debe utilizar comandos, por ejemplo, AT+CGATT el cual nos indica si la SIM tiene activo un plan de datos. Si es afirmativo se empieza indicando que se va a hacer uso de la red GPRS con el comando AT+SAPBR=3,1"CTYPE","GPRS", luego de esto se coloca la dirección APN (Nombre del Punto de Acceso) del chip. En este

caso al ser de la operadora Claro se colocó de la siguiente manera  
 AT+SAPBR=3,1\ "APN\ "-"internet.claro.com.ec\.

- e. Posteriormente, se usa el comando AT+HTTPINIT que permite abrir el camino para empezar la conexión por HTTP.
- f. Al haber iniciado la conexión HTTP antes mencionada, se empieza el envío de datos obtenidos del GPS hacia la página web, para lo cual es necesario realizar la estructuración del comando en base a una URL junto con la información receptada (longitud, latitud, velocidad) de la siguiente manera:  
 "AT+HTTTPARA=\ "URL\","\ "http://controlvelocidad.somee.com/InsertarDatos.aspx?&longitud=%s&latitud=%s&velocidad=%s&id=1\\"",lon,lat,vel.
- g. Por último, se utiliza el comando AT+HTTPACTION=1 para lograr que los datos enviados lleguen a la página web satisfactoriamente.



**Figura 2.19** Configuración del sistema dentro del IDE de Arduino

En la Figura 2.19 se presenta de forma didáctica la configuración que se desarrolló dentro del IDE de Arduino mediante códigos de programación. Inicialmente se declaró las variables a utilizar, luego se realizó el enlace con el GPS para poder obtener la información requerida y finalmente fue necesario establecer la conexión GPRS que permitió el envío de la información hacia la página web donde se tiene la base de datos.

### **2.2.3 NUBE DE ALMACENAMIENTO**

Luego de haber realizado la conexión física, la programación para que el módulo GPS capte la ubicación y velocidad, la configuración para el uso de la tecnología GPRS mediante datos móviles, se diseñó el modelo de la página web y se vinculó la base de datos en la nube.

#### **2.2.3.1 Diseño Pagina Web**

Para cargar el diseño de la página web se usó el *hosting somee.com* que sirve para el alojamiento de páginas web a bajo costo y en muchos casos hasta gratuito. Entre las principales características que ofrece este *hosting* es el alojamiento para ASP.net, que es donde se desarrolló la programación de la página web. A continuación, se presenta los pasos que se siguió para su creación.

	<b>Hosting package "Premium" 3 months</b> For web projects with advanced scripting requirements, email support and MSSQL database. Hosted in dedicated application pool. Billing interval is 3 months.	\$28.05	 Order
	<b>Hosting package "Premium" 6 months</b> For web projects with advanced scripting requirements, email support and MSSQL database. Hosted in dedicated application pool. Billing interval is 6 months.	\$51.90	 Order
	<b>Hosting package "Premium" 12 months</b> For web projects with advanced scripting requirements, email support and MSSQL database. Hosted in dedicated application pool. Billing interval is 12 months.	\$95.40	 Order
	<b>Hosting package "Developer" 3 months</b> For space, traffic and support demanding web projects with advanced scripting requirements, email support and 4 MSSQL databases. Hosted in dedicated application pool. Billing interval is 3 months.	\$59.85	 Order
	<b>Hosting package "Developer" 6 months</b> For space, traffic and support demanding web projects with advanced scripting requirements, email support and 4 MSSQL databases. Hosted in dedicated application pool. Billing interval is 6 months.	\$113.70	 Order
	<b>Hosting package "Developer" 12 months</b> For space, traffic and support demanding web projects with advanced scripting requirements, email support and 4 MSSQL databases. Hosted in dedicated application pool. Billing interval is 12 months.	\$215.00	 Order
	<b>Free hosting package</b> Package supports: ASP, ASP.Net, MS Access, MS SQL database. Single third level domain. Single addon web domain. Limitations: - No adult or illegal content; - Can't be used as a file storage or for "hot linking"; - Website must be visited at least 5 times per month; - Hiding of our advertisement is not allowed.	\$0.00	 Order

**Figura 2.20** Tipo de alojamiento en el Hosting  
(Somee, 2020, p. 2)

- a. Primero, el *hosting* solicitó la creación de una cuenta, pero antes del registro fue necesario escoger el tipo de alojamiento que se requiere, como lo indica la Figura 2.20 se escogió el alojamiento gratuito.

[Create new account](#)

[Click here if already registered](#)

First name:

Last Name:

User ID:

Password:

Confirm password:

Email address:

Prove you're not a robot

r r f z 5 6

Type the text:

I agree to the [Terms of service](#) and [Privacy policy](#)

**Figura 2.21** Registro en el Hosting Web  
(Somee, 2020, p. 1)

- b. El siguiente paso es la creación de la cuenta, donde se registró una dirección electrónica y una contraseña, el *hosting* verificó la dirección enviando un email mediante el cual se confirmó la identidad, como se puede notar en la Figura 2.21.

**Create website**

Hosting plan: Hosting plan "Freebie"

⚠ Although your hosting plan supports global domains you still need to provide initial default domain name which is hosted within our zone. You will be able to add additional domains later in control panel.

Site name (Subdomain):  Site name is required! Site name already registered!

Zone name:

Operating system:

ASP.Net version:

Site title:

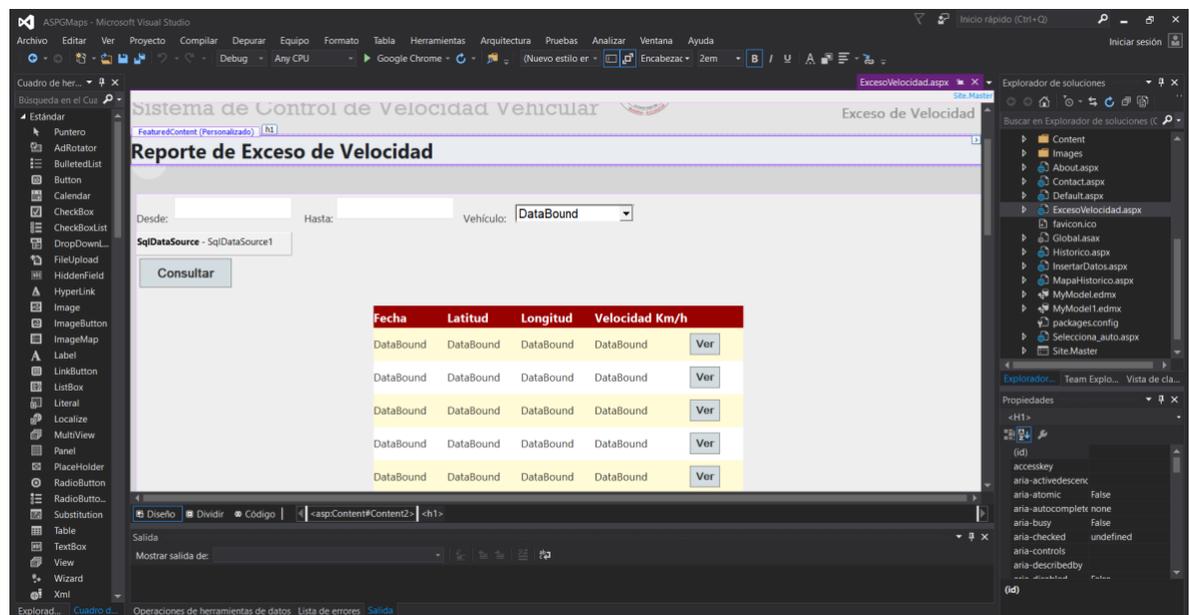
Site description:

**i** Your site will have default domain names as 'Sub domain','Zone name' and www.'Sub domain','Zone name'. If supported by hosting plan, you can use your own domain names, registered with domain name registrar. DNS record will be created instantly, but because of DNS replication delay it may take up to 24 hours for your site to be visible to all users on the Internet. You can try to access your site right after it's created. If it's not working retry it every 30 minutes.

The process may take up to several minutes. Please be patient.

**Figura 2.22** Nombre del sitio web  
(Somee, 2020, p. 2)

- c. A continuación, se creó el dominio, donde se colocó el nombre con el que se podrá ingresar a la página web para obtener los detalles de la información del vehículo que envía el Módulo SIM 808, en este caso se usa “controlvelocidad.somee.com” como se puede observar en la Figura 2.22.



**Figura 2.23** Diseño en Visual Studio

- d. Por otra parte, se realizó el diseño de la página web utilizando el programa Visual Studio, donde se construyó las pestañas y cada una de las opciones para ingresar a la información de una manera rápida y sencilla para el usuario. En la Figura 2.23 se

muestra el diseño de la pestaña “Reporte de Exceso de Velocidad” en donde se puede observar que cada botón cuenta con la programación que le permite realizar la función que le corresponde.

```

MS SQL hosting plan: MS SQL Plan "Novice"
MS SQL Server version: MS SQL 2014 Express

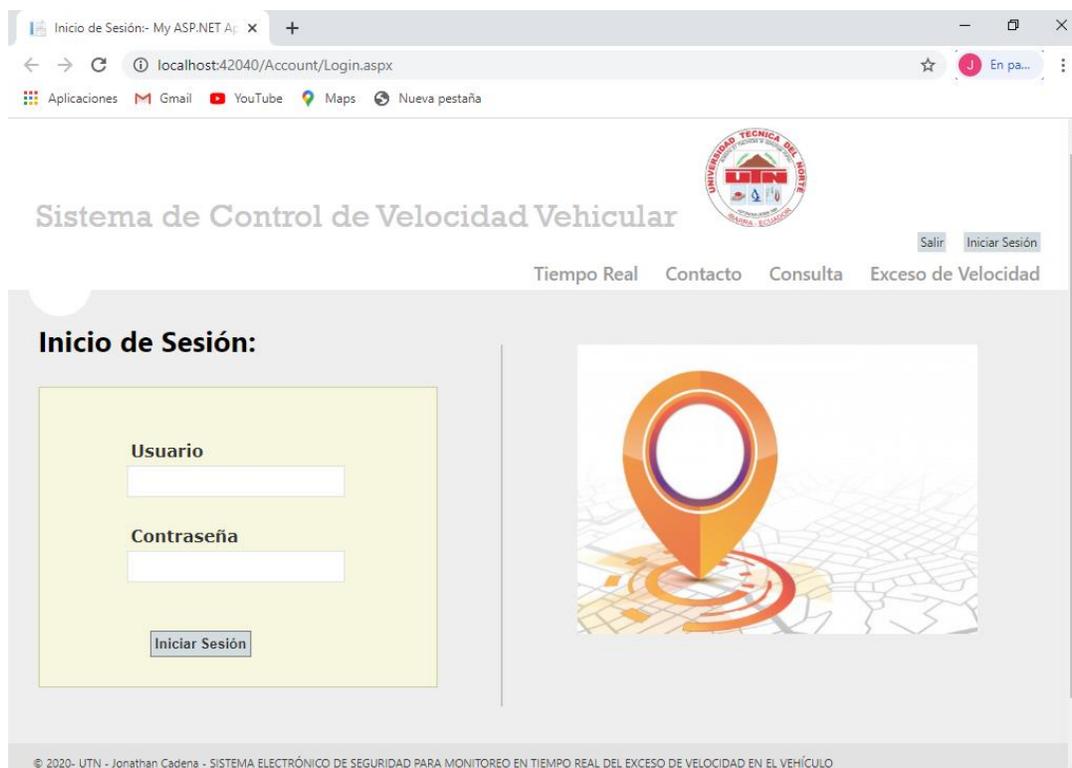
MS SQL Server address: control.mssql.somee.com
Login name: HenriC89_SQLLogin_1
Login password: 7v9oyfkbck

Connection string: workstation id=control.mssql.somee.com;packet size=4096;user id=HenriC89_SQLLogin_1;pwd=7v9oyfkbck;data
source=control.mssql.somee.com;persist security info=False;initial catalog=control

```

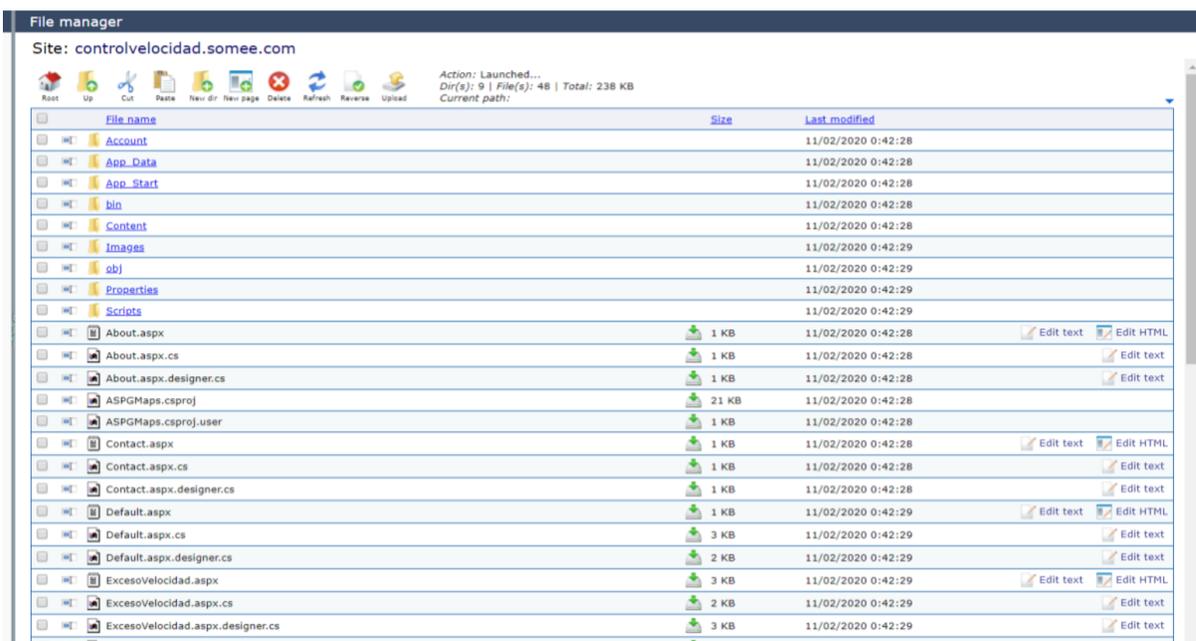
**Figura 2.24** Enlace de conexión al dominio

- e. El *hosting* donde se creó el dominio de la página web nos da una cadena de conexión (Figura 2.24) la cual se debe añadir y configurar dentro de la programación de Visual Studio, para que pueda enlazarse con el dominio creado.



**Figura 2.25** Pre visualización en Localhost

- f. Lo que hace atractivo a Visual Studio es que permite ir diseñando la página web y a la vez se puede observar cómo va quedando nuestro trabajo antes de terminarlo, esto se permite al ejecutarse como *Localhost* dentro del navegador como se puede notar en la Figura 2.25.



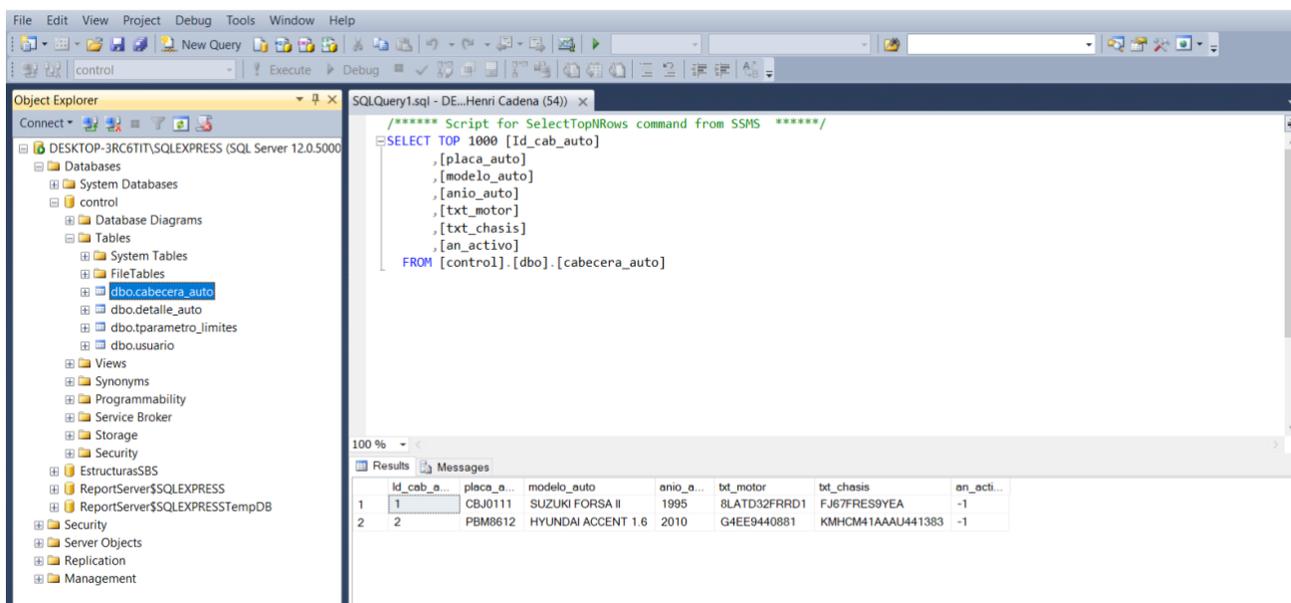
**Figura 2.26** Cargar archivos al Dominio

- g. Una vez terminado el diseño y programación, es necesario cargar los ficheros creados por Visual Studio al dominio correspondiente. El *hosting* brinda la opción de subir los archivos uno por uno o guardarlos en formato ZIP y cargar la carpeta completa, en la Figura 2.26 se puede observar a los archivos cargados en el dominio de Somee.

### 2.2.3.2 Desarrollo base de datos en la nube

Se necesita crear una base de datos mediante tablas en donde se recopile los registros que el sistema Arduino-GPS-GPRS envía a la página web para que estos queden guardados y se puedan observar en cualquier momento.

Para el desarrollo de la base de datos se usó el programa Microsoft SQL Server en el cual se fue generando tablas para el ingreso de la información, un punto adicional es que, se puede digitar manualmente la información y consultar en la base de datos los registros ingresados, por lo que una vez confirmado su funcionamiento se cargó la programación dentro del *hosting*.



**Figura 2.27** Base de Datos

En la Figura 2.27 se indica los vehículos que se encuentran en la base de datos, en este caso un vehículo Suzuki Forsa y un automóvil Hyundai Accent con sus respectivos datos: placa, año, número de motor y chasis.

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left displays the 'control' database structure, including tables like 'detalle\_auto'. The central query window shows the following SQL script:

```

/***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [id_det_auto]
, [longitud_auto]
, [latitud_auto]
, [fec_registro]
, [velocidad_reg]
, [id_cab_auto]
, [sn_exceso]
FROM [control].[dbo].[detalle_auto]

```

The Results pane at the bottom displays the following data:

	id_det_a...	longitud_a...	latitud_a...	fec_registro	velocidad_...	id_cab_a...	sn_exce...
1	1	1.65342	0.564746	2020-01-21 01:04:55.110	100	1	0
2	2	-0.056674	-0.056674	2020-01-21 01:32:58.377	90	1	0
3	3	1.65342	0.564746	2020-01-21 01:42:48.233	100	1	0
4	4	1.65342	0.564746	2020-01-21 01:43:03.570	100	1	-1
5	5	-78.244713	0.204995	2020-02-11 01:25:10.390	100	1	-1
6	6	-78.244713	0.204995	2020-03-10 11:04:38.597	100	1	0
7	7	-78.244713	0.204995	2020-03-10 11:04:38.623	100	1	0
8	8	-78.244713	0.204995	2020-03-10 11:07:11.107	100	1	0
9	9	-78.244713	0.204995	2020-03-10 11:08:13.100	100	1	0
10	10	-78.244713	0.204995	2020-03-10 11:10:45.643	100	1	0

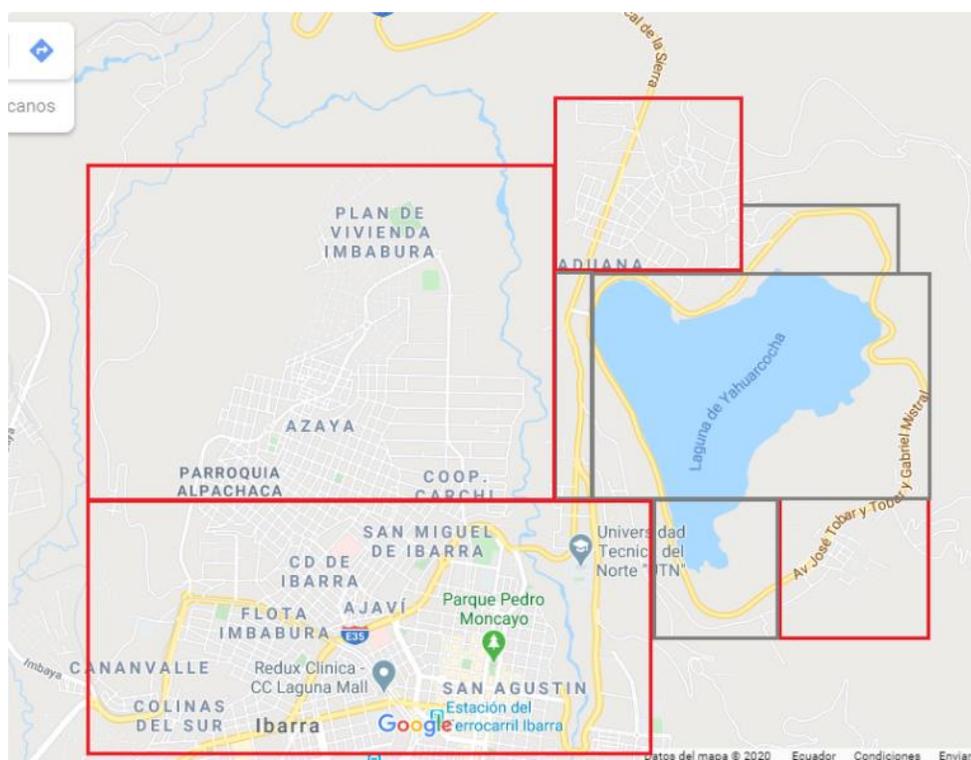
**Figura 2.28** Tablas de la base de datos

En la Figura 2.28 se muestra la tabla que se creó para el ingreso de la información, donde se tiene:

- longitud\_auto y latitud\_auto. – dentro de este apartado es donde se va almacenando las coordenadas que receipta el GPS y que se envían mediante el uso de GPRS a la página web.
- fec\_registro. – en esta columna se guarda tanto la fecha como la hora del momento que ingresa la información.
- velocidad. – es donde se ingresa la información de la velocidad en la que se moviliza el vehículo.
- id\_cab\_auto. – es el ID o identificación que se utiliza para saber qué vehículo está enviando los datos.
- sn\_exceso. – es una parte fundamental ya que en este recuadro el programa evalúa si el vehículo se encuentra dentro o fuera de los límites de velocidad.

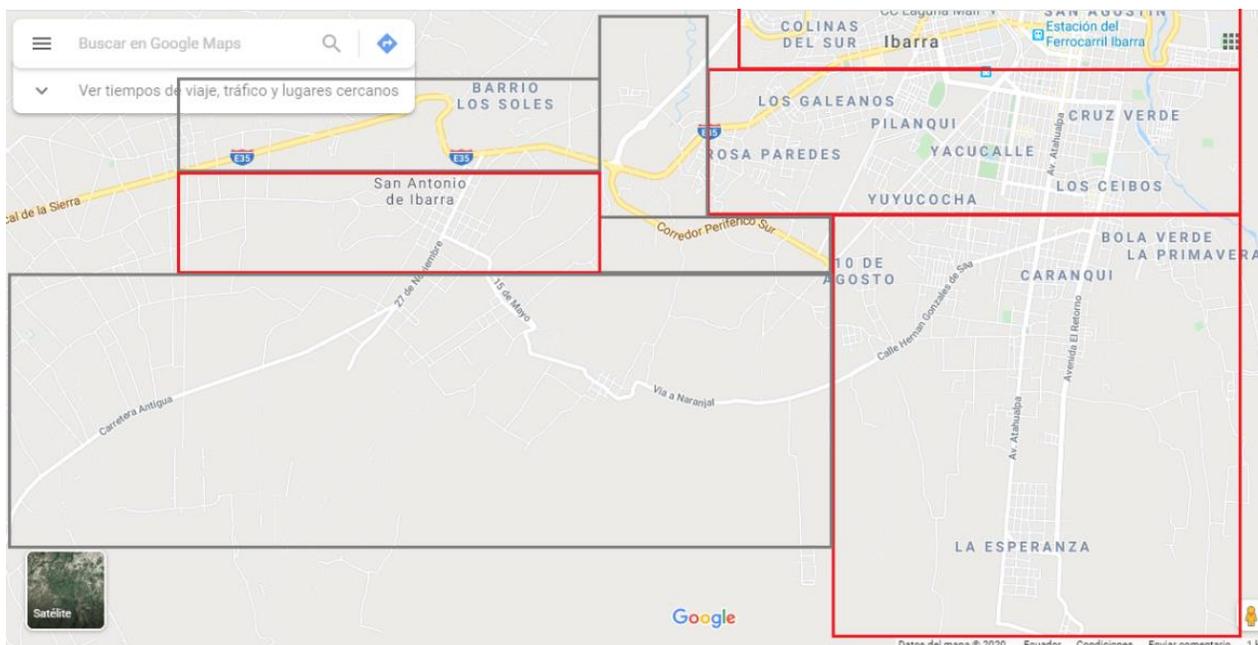
### 2.2.3.3 Parametrización de los límites de velocidad

Para que la página web almacene los excesos de velocidad en una pestaña, su base de datos necesita conocer cuál es el límite de velocidad dentro del lugar de donde está recibiendo las coordenadas de ubicación, para lo cual se creó un documento en Microsoft Excel con las coordenadas de cuadrantes realizados entre el sector de Tababela cerca del Aeropuerto Mariscal Sucre en la provincia de Pichincha hasta el norte de Imbabura. A continuación, se muestra gráficamente los resultados de la parametrización (cuadrantes).



**Figura 2.29** Parametrización Norte de Ibarra

La Figura 2.29 indica los cuadrantes realizados en el norte de Ibarra dependiendo los límites de velocidad, para los rojos el límite de velocidad es de 50 km/h, mientras que para los cuadrantes de color gris de 90 km/h.



**Figura 2.30** Parametrización Sur de Ibarra

Como se puede observar en la Figura 2.30 los cuadrantes se realizaron dependiendo los límites de la velocidad en el sur de Ibarra, con características diferentes para el sector urbano y rural.

Para que la base de datos lea lo mostrado gráficamente, lo evalué y sepa si está o no dentro de los límites de velocidad se tuvo que tomar cuatro puntos de coordenadas y realizar un documento Excel, el cual nos permite subir la información al *hosting*.

Lo que hace el sistema es evaluar las coordenadas receptadas e identificar dentro de que cuadrante se encuentra, por lo que de esa manera se conoce cuál es el límite de velocidad y si el vehículo se excedió o no.

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2		Lugares	latitud_1	longitud_1	latitud_2	longitud_2	latitud_3	longitud_3	latitud_4	longitud_4	limite_vel	
3	1	Norte	0,48164	-78,135471	0,479237	-78,103371	0,394954	-78,103542	0,395469	-78,136673	100	
4	2	Aduana	0,391993	-78,113842	0,391907	-78,100366	0,379547	-78,100967	0,380406	-78,114271	50	
5	3	N. Yaguarco	0,385813	-78,100624	0,384826	-78,084102	0,379419	-78,084445	0,379547	-78,100967	90	
6	4	Ent. Yaguarco	0,380406	-78,114271	0,381307	-78,110623	0,363969	-78,110537	0,364141	-78,113799	90	
7	5	Cent.Yaguarco	0,381307	-78,110623	0,379419	-78,084445	0,363884	-78,086076	0,363969	-78,110537	90	
8	6	Ciud.Yaguarco	0,364484	-78,09629	0,363884	-78,086076	0,354099	-78,085303	0,35367	-78,096461	50	
9	7	Sur.Yaguarco	0,364227	-78,106611	0,364484	-78,09629	0,35367	-78,096461	0,352269	-78,10644	90	
10	8	Azaya	0,392465	-78,147466	0,391993	-78,113842	0,364141	-78,113799	0,365343	-78,146694	50	
11	9	UTN	0,365343	-78,146694	0,364227	-78,106611	0,352296	-78,10644	0,354571	-78,146501	50	
12	10	Pilantqui	0,354571	-78,146501	0,352296	-78,10644	0,331182	-78,103586	0,332984	-78,146072	50	
13	11	Esperanza	0,331611	-78,139206	0,331182	-78,103586	0,296078	-78,103672	0,296335	-78,138605	50	
14	12	Corredor Sur	0,333843	-78,159719	0,331611	-78,139206	0,326461	-78,139206	0,328006	-78,159805	90	
15	13	Inicio Corredor	0,35603	-78,159204	0,354571	-78,146501	0,332984	-78,146072	0,333843	-78,159719	90	
16	14	Los Soles	0,352339	-78,192506	0,35603	-78,159204	0,336976	-78,159633	0,336461	-78,192592	100	
17	15	San Antonio	0,336461	-78,192592	0,336976	-78,159633	0,328006	-78,159805	0,326548	-78,192678	50	
18	16	N. Atuntaqui	0,340066	-78,210869	0,336461	-78,192592	0,331826	-78,192672	0,331998	-78,210354	90	
19	17	Atuntaqui	0,340409	-78,239879	0,340066	-78,210869	0,323415	-78,193187	0,323243	-78,240909	50	
20	18	S. Atuntaqui	0,323243	-78,240909	0,320668	-78,22031	0,303918	-78,220855	0,306592	-78,241253	100	
21	19	Est.Andrade	0,320668	-78,22031	0,320396	-78,193618	0,303783	-78,194114	0,303918	-78,220855	50	
22	20	Iluman	0,307006	-78,259452	0,303918	-78,220855	0,243823	-78,218274	0,245636	-78,259948	100	

Figura 2.31 Datos de Parametrización Excel

Si el vehículo sobrepasa los límites automáticamente se guarda la información como reporte de exceso de velocidad. La Figura 2.31 muestra la manera en la que se realizó el documento con las variables de coordenadas y velocidad.

```

/***** Script for SelectTopRows command from SSMS *****/
SELECT TOP 1000 [id_param_limit]
      ,[bt_lugar]
      ,[latitud_noc]
      ,[longitud_noc]
      ,[latitud_nor]
      ,[longitud_nor]
      ,[latitud_soc]
      ,[longitud_soc]
      ,[latitud_sor]
      ,[longitud_sor]
      ,[limite_vel]
FROM [control].[dbo].[tparametro_limites]

```

id_param_limit	bt_lugar	latitud_noc	longitud_noc	latitud_nor	longitud_nor	latitud_soc	longitud_soc	latitud_sor	longitud_sor	limite_vel
1	Norte	0.48164	-78.135471	0.479237	-78.103371	0.394954	-78.103542	0.395469	-78.136673	100
2	Aduana	0.391993	-78.113842	0.391907	-78.100366	0.379547	-78.100967	0.380406	-78.114271	50
3	N. Yaguarco	0.385813	-78.100624	0.384826	-78.084102	0.379419	-78.084445	0.379547	-78.100967	90
4	Ent. Yaguarco	0.380406	-78.114271	0.381307	-78.110623	0.363969	-78.110537	0.364141	-78.113799	90
5	Cent.Yaguarco	0.381307	-78.110623	0.379419	-78.084445	0.363884	-78.086076	0.363969	-78.110537	90
6	Ciud.Yaguarco	0.364484	-78.09629	0.363884	-78.086076	0.354099	-78.085303	0.35367	-78.096461	50
7	Sur.Yaguarco	0.364227	-78.106611	0.364484	-78.09629	0.35367	-78.096461	0.352269	-78.10644	90
8	Azaya	0.392465	-78.147466	0.391993	-78.113842	0.364141	-78.113799	0.365343	-78.146694	50
9	UTN	0.365343	-78.146694	0.364227	-78.106611	0.352296	-78.10644	0.354571	-78.146501	50
10	Pilantqui	0.354571	-78.146501	0.352296	-78.10644	0.331182	-78.103586	0.332984	-78.146072	50
11	Esperanza	0.331611	-78.139206	0.331182	-78.103586	0.296078	-78.103672	0.296335	-78.138605	50
12	Corredor Sur	0.333843	-78.159719	0.331611	-78.139206	0.326461	-78.139206	0.328006	-78.159805	90
13	Inicio Corredor	0.35603	-78.159204	0.354571	-78.146501	0.332984	-78.146072	0.333843	-78.159719	90
14	Los Soles	0.352339	-78.192506	0.35603	-78.159204	0.336976	-78.159633	0.336461	-78.192592	100

Figura 2.32 Datos Parametrización en la Base de Datos

Como se nota en la Figura 2.32 una vez realizada la parametrización de los límites de velocidad, se subió la información a la base de datos con un formato similar al de Microsoft Excel, declarando las variables de coordenadas y del límite de velocidad. Estos registros se cargan a la base para que cuando el vehículo se encuentre en movimiento y envíe los datos internamente coteje con la información almacenada y pueda conocer si se excedió o no los límites de velocidad.

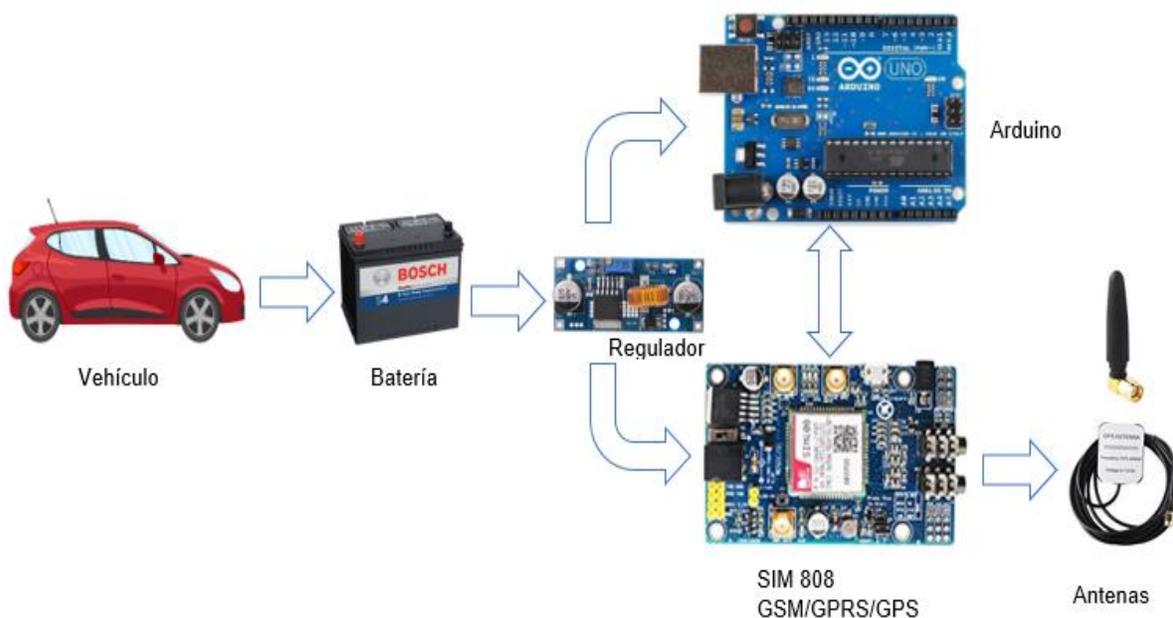
#### **2.2.4 IMPLEMENTACIÓN DEL SISTEMA EN EL VEHICULO**

Una vez que los componentes Arduino y Modulo SIM 808 están entrelazados de manera física, mediante una interfaz electrónica, se encuentre diseñada la página web y se haya corroborado mediante pruebas que el prototipo está totalmente funcional, se procede a instalarlo dentro del vehículo.

Por lo que hay que tomar en cuenta ciertos puntos:

- Localización. - Donde el prototipo no interrumpa otra función dentro del vehículo.
- Señal. - Que el GPS pueda recibir la información de los satélites sin ninguna interferencia.
- Fuentes de Alimentación. - Se debe colocar el prototipo de tal manera que cuente con el voltaje y corriente necesarios, realizando una instalación adicional en el vehículo.
- Señal GPRS. - Para el envío de datos a una página web se debe contar con una señal fuerte.
- Seguridad. - El dispositivo no se debe encontrar al alcance de personas sin autorización mismas que puedan interrumpir su funcionamiento.

En el vehículo se instaló el dispositivo tomando directamente la corriente de su batería (12V) y gracias a un regulador de voltaje se logró disminuir la carga a 5V y 2A que es lo necesario para la SIM 808 GSM/GPRS/GPS y Arduino.

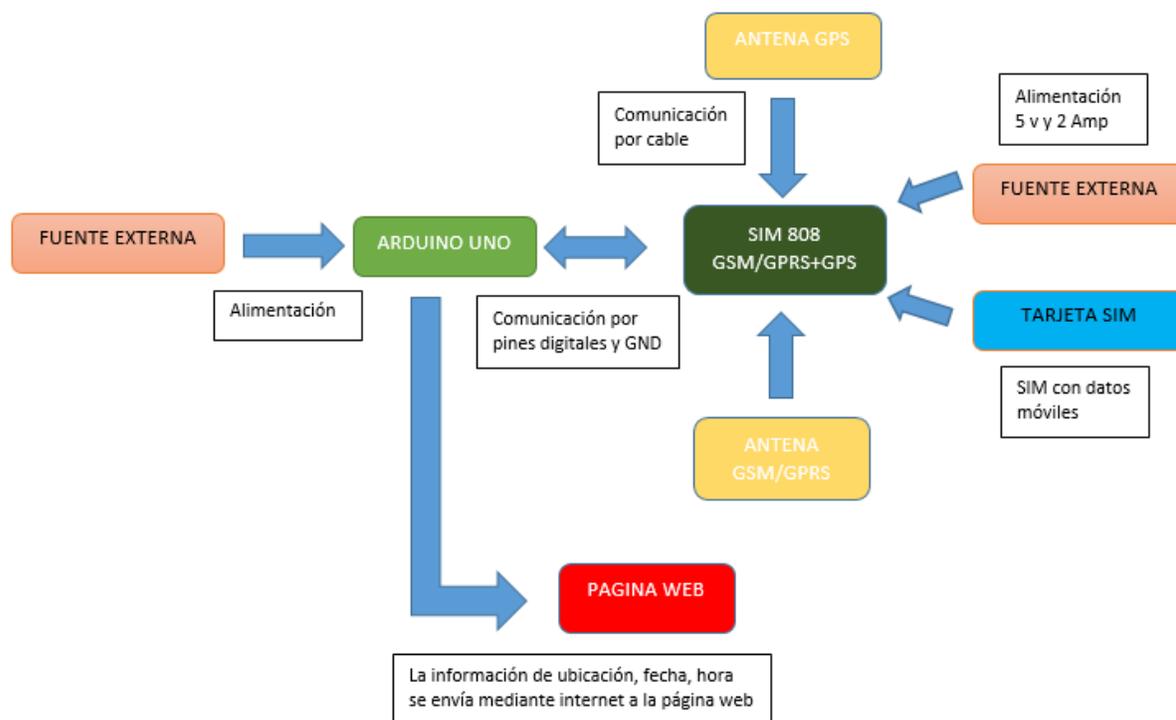


**Figura 2.33** Esquema del dispositivo conectado al vehículo

En la Figura 2.33 se muestra el esquema de la manera en que se conectó el dispositivo en el vehículo; la instalación se realizó en la parte delantera del vehículo detrás del tablero principal, donde la antena del GPS sobresale por el parante derecho para no perder la conexión de los satélites.

#### 2.2.4.1 Pruebas en tiempo real

Finalmente, luego de haber generado la conexión entre Arduino, GPS, el módulo *Shield* de internet y la base de datos en la nube, se procedió a realizar las pruebas en tiempo real, para lo cual el dispositivo se encuentra instalado en el vehículo y funcionando.



**Figura 2.34** Esquema completo del sistema

La Figura 2.34 muestra un esquema completo del sistema de seguridad donde se indica las conexiones y características de cada parte del sistema para que sea funcional, existiendo entradas como fuentes de alimentación, conexión de antenas, enlace entre dispositivos y la salida de envío de información hacia la página web usando datos móviles de una SIM.

Para las pruebas en tiempo real, se realizó en dos situaciones diferentes, dentro de la ciudad que es la zona urbana y en zonas rurales, donde los valores de velocidad son distintos. Se excedió los límites oportunamente para que los datos queden grabados en la página web.

## CAPÍTULO III

### 3 RESULTADOS Y DISCUSIÓN

Con la ayuda del sistema de seguridad electrónico propuesto se puede tener un monitoreo en tiempo real del vehículo conociendo su ubicación y velocidad, así como también un registro sobre las ocasiones que existió excesos de velocidad. Adicionalmente, puede utilizarse como un sistema de seguridad o control del vehículo, ya sea que exista algún percance como robo, accidente de tránsito o que el vehículo no lo conduzca el propietario, e incluso mediante multas se puede ayudar a concientizar a la población que el exceso de velocidad es un peligro en las vías.

El sistema es cómodo para el usuario, debido a que se puede acceder a la información desde cualquier lugar del mundo, simplemente es necesario contar con acceso a internet y con el permiso para ingresar a la página web. A continuación, se detalla los resultados obtenidos luego de haber finalizado las conexiones físicas, los enlaces de software y el diseño de la página web; presentando las fortalezas y debilidades del sistema.

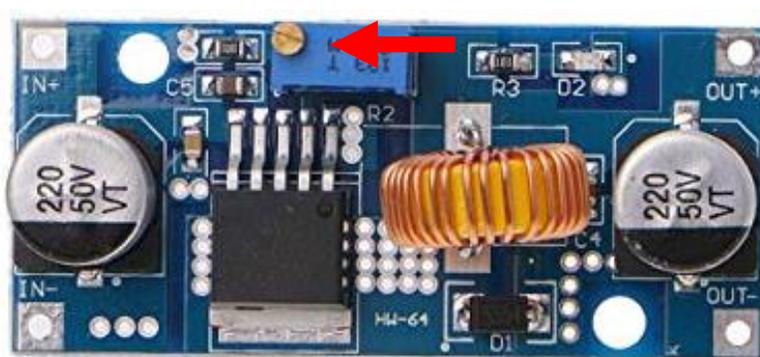
#### 3.1 DESCRIPCIÓN E IMPLEMENTACIÓN PROTOTIPO

El sistema electrónico de seguridad para el monitoreo en tiempo real del exceso de velocidad se dividió en cuatro fases principales como se mencionó en el capítulo anterior. La primera fase fue la de elaborar la interfaz electrónica para obtener las coordenadas y velocidad del vehículo, luego fue la implementación del módulo *Shield* para enviar la información a la base de datos en la nube, la tercera fase fue la visualización de la información receptada en la página web y por último la instalación del sistema dentro del vehículo.

Por otra parte, es importante describir la adecuación realizada dentro del vehículo, donde fue necesaria una fuente de alimentación para la SIM 808 y para Arduino. Inicialmente existió la idea de tomar la corriente desde un puerto encendedor de cigarrillos añadiendo un adaptador,

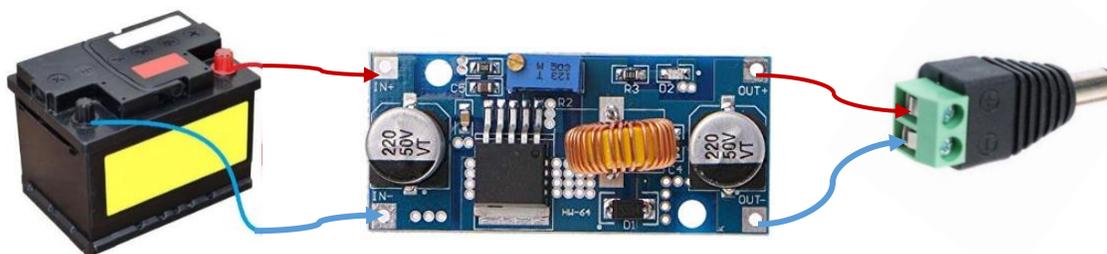
pero el amperaje que brinda es menor a 2 Amperios por lo que no fue suficiente para mantener al módulo SIM encendido y utilizando la red GSM/GPRS.

El módulo SIM 808 necesita 5V y 2A para su correcto funcionamiento, por lo que la corriente se tuvo que tomar directamente de la batería del vehículo (12 V de corriente continua). El cual al poner en marcha el motor su carga se eleva aproximadamente a 13,7 V debido a la activación del alternador, motivo por el que fue necesario instalar un regulador de voltaje que mantenga la corriente requerida estable.



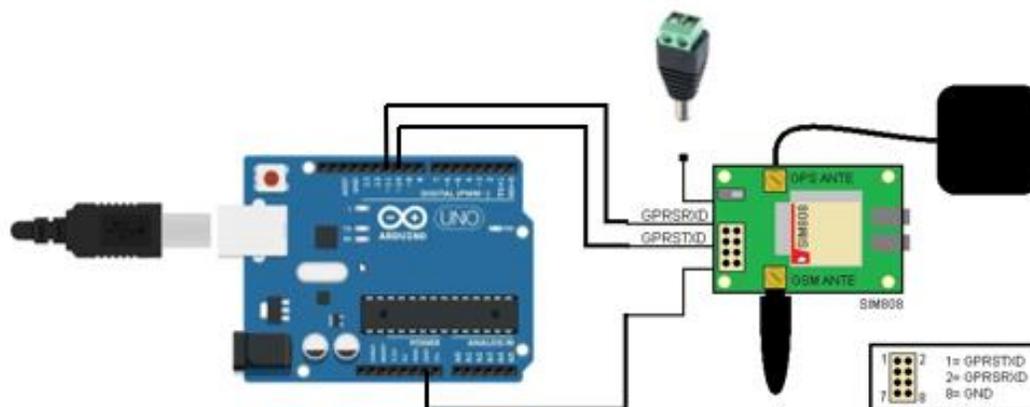
**Figura 3.1** Regulador DC-DC XL4015 con su potenciómetro

Se instaló un regulador de voltaje DC-DC XL4015 con un amperaje en la salida de 5A, que permite la entrada de un máximo de 36V y una salida desde 1.25V a 32V los cuales son regulables a la necesidad del usuario. En la Figura 3.1 se observa el regulador con el potenciómetro que permitió modificar la corriente de salida.



**Figura 3.2** Conexión Regulador de Voltaje

La conexión se realizó mediante soldadura de estaño tanto a la entrada como salida del regulador, la Figura 3.2 indica que al ingreso se conectó los cables positivo y negativo que provienen directamente de la batería del vehículo y por el otro lado se conectó un *plug* de carga. Con la ayuda de un multímetro se reguló la corriente de salida para obtener los 5 voltios y 2 amperios constantes requeridos para el correcto funcionamiento del sistema.



**Figura 3.3** Conexión Hardware

En conclusión, se logró conseguir la fuente de alimentación correcta para el prototipo, por lo que se realizó la conexión completa entre el módulo SIM 808 GSM/GPRS/GPS y Arduino como se presenta en la Figura 3.3 donde se muestra:

- Fuente de alimentación mediante *plug* de carga para el módulo SIM 808 desde el regulador de voltaje DC-DC XL4015 con una corriente de 5V y 2A necesarios para una conexión GSM/GPRS/GPS estable.
- Fuente de alimentación para Arduino por parte del regulador de voltaje DC-DC XL4015; para las primeras pruebas se realizó la conexión al computador mediante cable USB.
- Conexión física para el envío y recepción de datos, desde el pin Tx del módulo SIM hacia el pin 7 de Arduino y el pin Rx del módulo SIM al pin 8 de Arduino.
- Enlace entre los pines GND de ambos dispositivos para tener una correcta transferencia de información
- Conexión de la antena GPS y la antena GSM al módulo SIM 808.
- Colocar una tarjeta SIM con un paquete activo de internet, en este caso se colocó de la empresa Claro.

### 3.2 ANÁLISIS FINANCIERO

Para diseñar el sistema electrónico de seguridad se necesitó varios equipos y materiales los cuales se presentan en la Tabla 3.1 con su respectivo valor monetario en el mercado ecuatoriano, concluyendo que para contar con el sistema fue necesario alrededor de noventa y nueve dólares americanos (\$ 99.00). Adicional a este valor se tendrá un costo mensual de \$10.00, el cual corresponde al paquete de datos móviles de la operadora Claro fundamental para mantener activo el sistema.

**Tabla 3.1** Costo Sistema de Seguridad

<b>MATERIALES/EQUIPOS</b>	<b>COSTO</b>
Arduino UNO	\$ 10.00
Sim 808 GSM/GPRS + GPS	\$ 50.00
SIM compañía telefónica Claro	\$ 5.00
Regulador DC-DC XL4015	\$ 4,50
Cable conductor/Estaño	\$ 4,50
Paquete de datos móviles por un mes	\$ 10.00
Carcasa protectora de Equipos	\$ 15.00
<b>TOTAL</b>	<b>\$ 99.00</b>

Un punto importante a considerar es que el presente proyecto usó un hosting gratuito para el alojamiento de la página web, reduciendo costos debido a que en promedio el valor mensual de este alojamiento oscila entre \$7,85 y \$17, 49.

Los gastos presentados anteriormente representan únicamente los utilizados para el sistema de seguridad, sin embargo, para lograr finalizar el proyecto se necesitó incurrir en gastos adicionales como se puede observar detalladamente en la Tabla 3.2

**Tabla 3.2** Costos adicionales del Proyecto

<b>GASTOS ADICIONALES</b>	<b>COSTO</b>
Curso de Programación de Arduino	\$ 180.00
Cautin	\$ 7.00
Pruebas de Funcionamiento	\$ 80.00
Materiales de Oficina, impresiones, internet	\$ 80.00
<b>TOTAL</b>	<b>\$ 347.00</b>
Equipos/Materiales + Gastos Adicionales	\$ 446.00

### 3.3 PRUEBAS DE CAMPO DATOS GPS

Luego de haber desarrollado la interfaz electrónica Arduino-GPS y adaptado el sistema en el vehículo, se realizaron las primeras pruebas en tres lugares diferentes, donde se muestran los datos obtenidos del GPS y con la ayuda de *Google Maps* se evaluó si las coordenadas tanto de latitud como de longitud son correctas para poder continuar con el proyecto.



**Figura 3.4** Primera conexión del prototipo

Primeramente, la corriente utilizada por el módulo SIM 808 se tomó de la batería del vehículo pasando por un regulador de voltaje que permitió alimentar al sistema con una corriente de 5V y 2A.

Para lograr la alimentación y conexión de Arduino se enlazo mediante un cable USB al computador y se cargó el *sketch* de programación dentro del IDE, en la Figura 3.4 se observa el prototipo conectado y funcional para iniciar las pruebas.

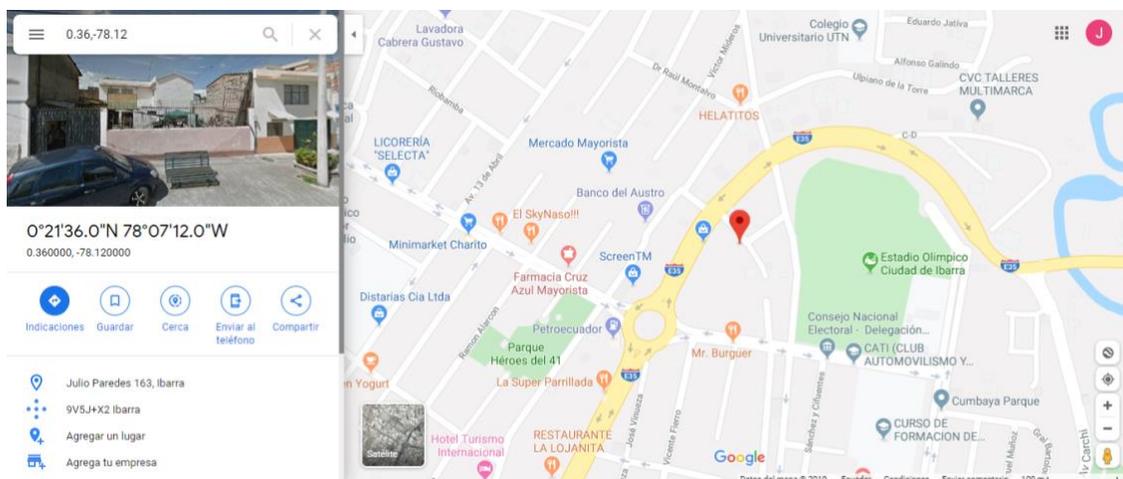
```

1 /*
2 *** Get GP
3 1. This ex Encendido de GPS exitoso
4 2. Open th 2019/12/3 22:29:57:30
5 3. Downloa Latitud :0.36
6 4. open se Longitud :-78.12
7 4. Place i Velocidad_kph :50.86
8 create on
9 by jason
10 */
11 #include <
12 #include <
13
14 //char lat
15 //char lon
16 //char wsp
17 char datet
18

```

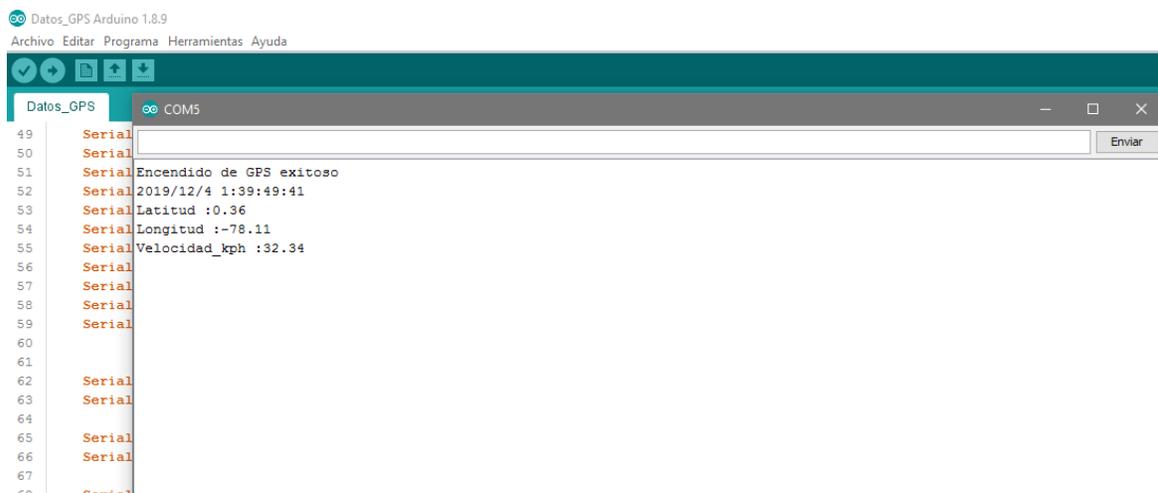
**Figura 3.5** Primer prueba de coordenadas, panamericana norte 03/12/2019, 22:59

Es importante indicar que al encender la SIM 808 el GPS tarda aproximadamente 30 segundos en reconocer la red y empezar a recibir la información, luego de este tiempo se inició las pruebas con el vehículo en movimiento donde se tomó una captura de la información que ingresó al monitor serial del IDE de Arduino. En la Figura 3.5 se puede notar la respuesta afirmativa del encendido del GPS y los datos de fecha, hora, coordenadas y velocidad en kilómetros por hora.



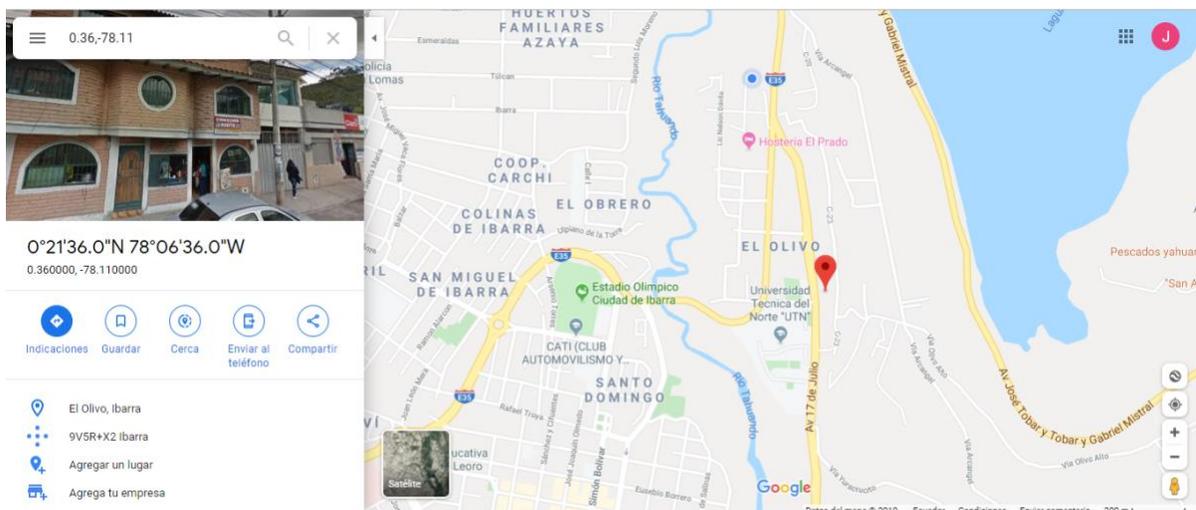
**Figura 3.6** Primeras coordenadas Google Maps

Para verificar que la información proporcionada es correcta, se confirma la ubicación en *Google Maps*, para hacerlo se inserta las coordenadas de latitud y de longitud separadas por una coma, y como se muestra en la Figura 3.6 esta información coincide con la obtenida por el sistema siendo el lugar en Ibarra cerca de la Panamericana E35 a una velocidad de 50 km/h.



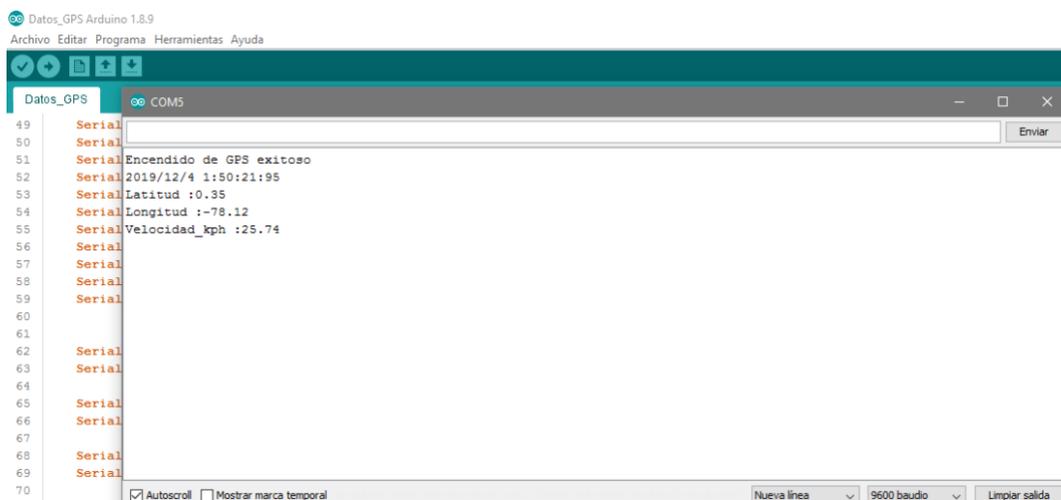
**Figura 3.7** Segunda prueba de coordenadas, Avenida 17 de Julio 04/12/2019, 01:39

Se realizó una segunda prueba con el prototipo conectado al PC y el vehículo en movimiento en otra parte de la ciudad de Ibarra, esta prueba arrojó información diferente como se muestra en la Figura 3.7.



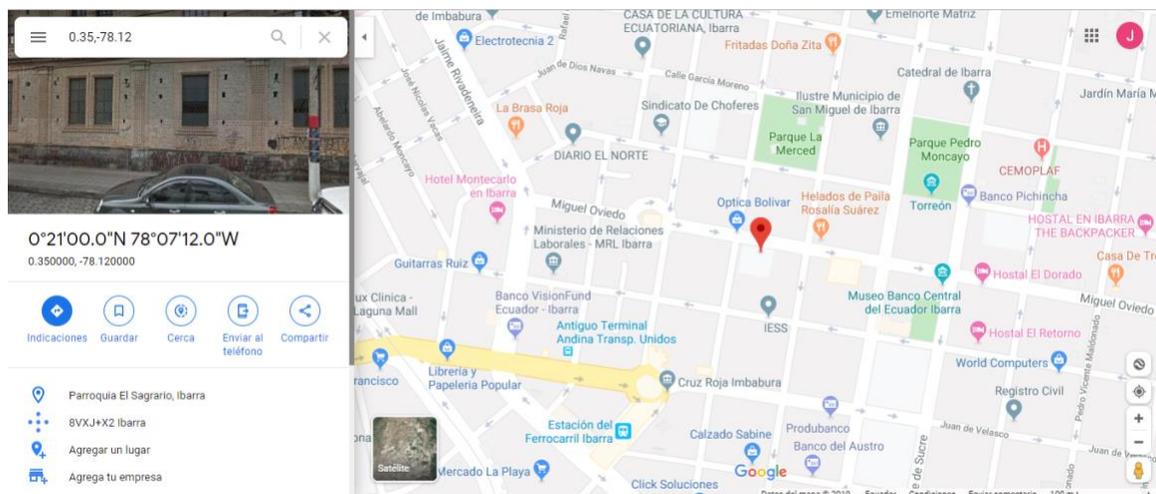
**Figura 3.8** Segundas coordenadas Google Maps

En la Figura 3.8 se puede ver la ubicación mediante las coordenadas colocadas de 0.36, -78.11, las cuales muestran que la información fue receptada al norte de Ibarra en la Avenida 17 de Julio, cerca de la Universidad Técnica del Norte a una velocidad de 32 km/h.



**Figura 3.9** Tercera prueba coordenadas, calle Miguel Oviedo 04/12/2019, 01:50

Finalmente, se realizó una tercera prueba antes de continuar con el trabajo, para validar que la información enviada por el módulo SIM 808 es correcta; en la Figura 3.9 se muestra los datos recibidos gracias a esta última prueba.



**Figura 3.10** Terceras coordenadas Google Maps

En esta ocasión también se verificó las coordenadas indicadas por el GPS del módulo, donde indica que la ubicación es en el centro de Ibarra en la calle Miguel Oviedo a una velocidad de 25 km/h, como se observa en la Figura 3.10.

Es importante indicar que en las primeras pruebas se obtuvo los valores de latitud y longitud con solamente dos decimales razón por la que la ubicación no es igual de exacta como las

siguientes pruebas donde se modificó la programación y se obtuvo los valores con cinco decimales.

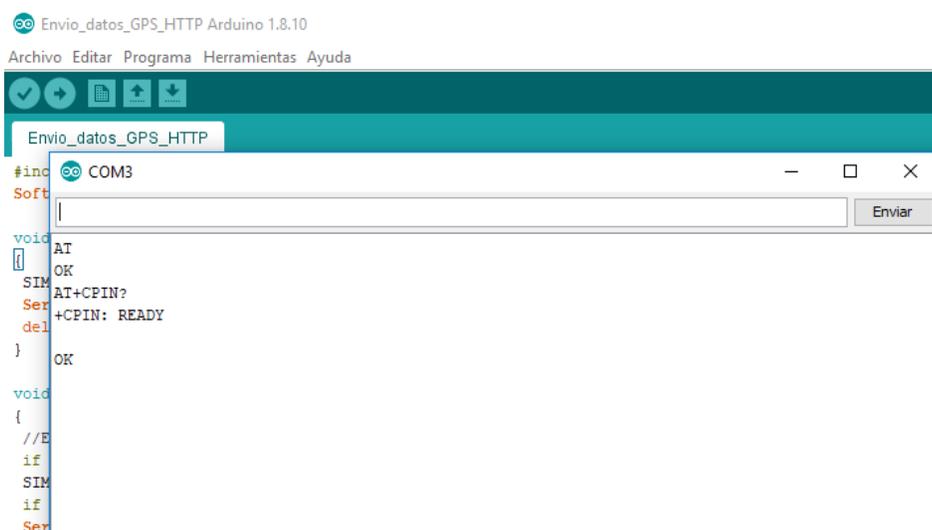
### 3.4 ENVÍO DE INFORMACIÓN MEDIANTE DATOS MÓVILES

Luego de tener la conexión física del prototipo dentro del vehículo y de haber obtenido la información necesaria como latitud, longitud y velocidad, previamente validada por *Google Maps*, se pudo continuar con el trabajo propuesto. El siguiente paso fue enviar los datos a la página web.

#### 3.4.1 ENVIO DATOS HTTP

La información que fue receptada por el GPS se envió mediante datos móviles a la base de datos de la página web, para su posterior procesamiento y visualización en tiempo real, así como también para tener los registros almacenados.

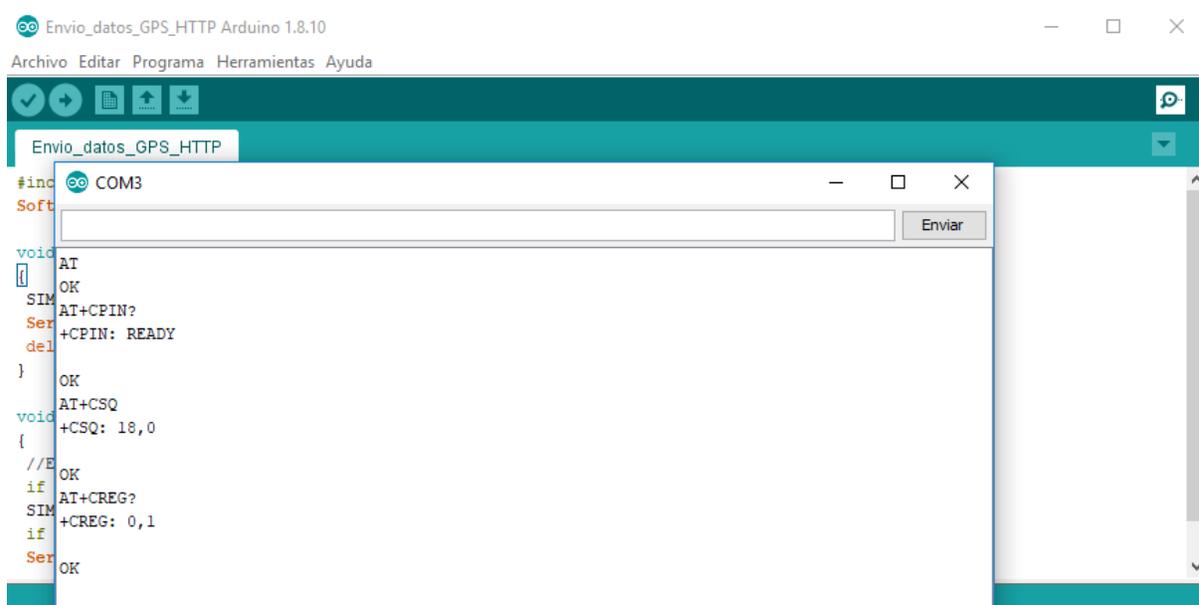
Teniendo lista la conexión física explicada anteriormente, el siguiente paso fue añadir a la programación comandos AT los que permiten recibir la información de las coordenadas y velocidad del vehículo y enviarla por conexión HTTP.



```
Envio_datos_GPS_HTTP Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda
Envio_datos_GPS_HTTP
#inc COM3
Soft
void
  AT
  OK
  SIM
  AT+CPIN?
  Set +CPIN: READY
  del
}
  OK
void
{
  //E
  if
  SIM
  if
  Set
```

**Figura 3.11** Conexión Ready

En la Figura 3.11 se puede notar como empieza el enlace, primero se obtiene la confirmación de que existe comunicación, y seguidamente mediante el comando “AT+CPIN?” indica si es necesario el ingreso de una contraseña, si no es el caso se puede utilizar directamente la SIM (CLARO, inicialmente con un plan activo de 1GB) obteniendo la respuesta “Ready”.



The screenshot shows the Arduino IDE interface with the serial monitor open. The title bar of the serial monitor window reads "COM3". The code in the background includes the following AT commands and their responses:

```

void setup()
{
  AT
  OK
  SIM
  Ser AT+CPIN?
  +CPIN: READY
  del
}

void loop()
{
  OK
  AT+CSQ
  +CSQ: 18,0
  //E
  OK
  if AT+CREG?
  SIM
  if +CREG: 0,1
  Ser OK

```

**Figura 3.12** Cobertura de red Claro

Después, se continuó utilizando comandos AT los cuales podemos ver en el monitor serial, se colocó (AT+CSQ) y (AT+CREG?), donde el primer comando mide la calidad de la señal y el segundo nos indica el estado del registro de la red.

El rango más alto de señal que puede obtener el dispositivo es de [(31,99),(7,99)], en este caso como se observa en la Figura 3.12 se tiene (+CSQ:18,0) lo que indica una buena calidad de señal y además al obtener la respuesta (+CREG:0,1) se confirmó que está registrado en la red.

The screenshot shows the Arduino IDE interface with the following code in the editor:

```

#include <SoftwareSerial.h>
SoftwareSerial COM3(2, 3);

void setup() {
  COM3.begin(9600);
  AT+CSQ
  +CSQ: 18,0
  OK
  AT+CREG?
  +CREG: 0,1
  OK
  //E
  AT+CGATT?
  +CGATT: 1
  OK
  AT+SAPBR=3,1,"CONTYPE","GPRS"
  OK
}

void loop() {
  //E
  AT+SAPBR=3,1,"CONTYPE","GPRS"
  OK
}

```

**Figura 3.13** Conexión módulo SIM 808 a la red GPRS

Si siguiendo con los comandos se utilizó (AT+CGATT?) el cual pregunta por el estado del GPRS obteniendo una respuesta positiva lo que significa que el prototipo está listo para usarse. Luego mediante el comando (AT+SAPBR=3,1,"CONTYPE","GPRS") se indica al módulo SIM 808 que se requiere realizar una conexión mediante el uso de GPRS, obteniendo una respuesta positiva con un "OK", lo que se puede notar en la Figura 3.13.

The screenshot shows the Arduino IDE interface with the following code in the editor:

```

#include <SoftwareSerial.h>
SoftwareSerial COM3(2, 3);

void setup() {
  COM3.begin(9600);
  AT+CREG?
  +CREG: 0,1
  OK
  AT+CGATT?
  +CGATT: 1
  OK
  AT+SAPBR=3,1,"CONTYPE","GPRS"
  OK
  AT+SAPBR=3,1,"APN","internet.claro.com.ec"
  OK
  AT+SAPBR=1,1
  OK
}

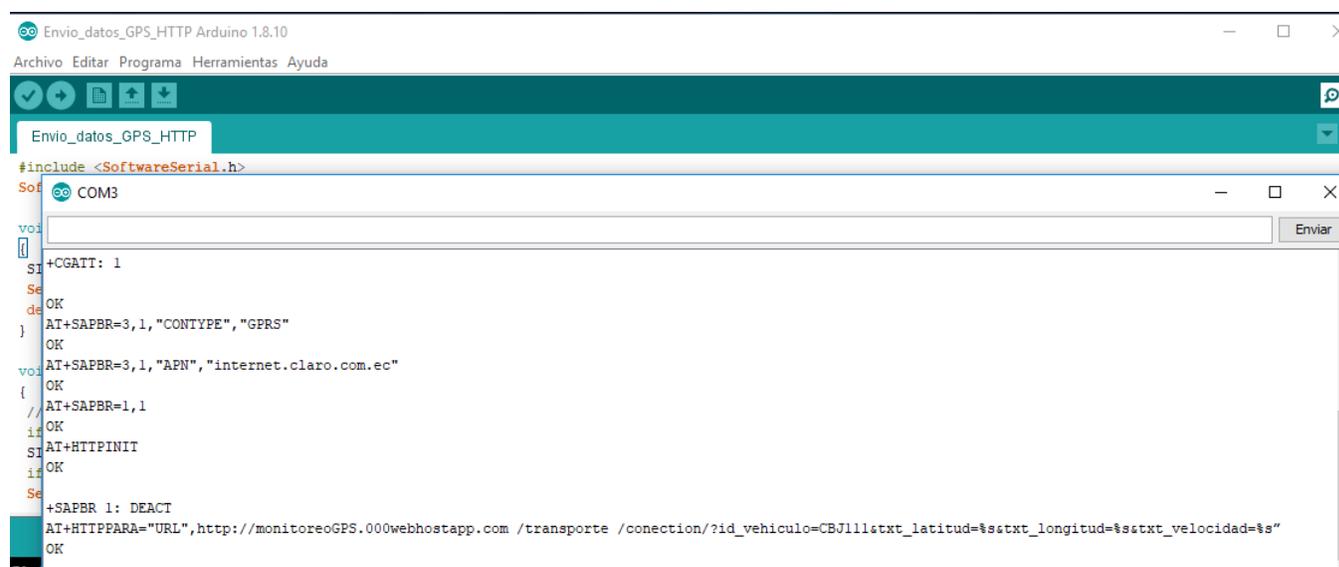
void loop() {
  //E
  AT+SAPBR=3,1,"CONTYPE","GPRS"
  OK
  AT+SAPBR=3,1,"APN","internet.claro.com.ec"
  OK
  AT+SAPBR=1,1
  OK
}

```

**Figura 3.14** Conexión a la APN Claro

Luego de esto se debe indicar el APN o Nombre del Punto de Acceso que se va a usar dependiendo de la operadora, en este caso al ser un SIM de Claro el APN es “internet.claro.com.ec”. Entonces el comando completo que se utilizó fue: (AT+SAPBR=3,1,“APN”, “internet.claro.com.ec”), dando una respuesta positiva de la conexión como se muestra en la Figura 3.14.

Después de haber configurado el uso de la SIM 808 y la red a utilizar con la SIM Claro, se inicia una petición para enviar la información mediante conexión HTTP con la ayuda del comando (AT+HTTPIPINIT) de donde se obtiene una respuesta afirmativa.



```

Envio_datos_GPS_HTTP Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda
Envio_datos_GPS_HTTP
#include <SoftwareSerial.h>
Soft COM3
vo:
SI
+CGATT: 1
Se
OK
de
AT+SAPBR=3,1,"CONTYPE","GPRS"
}
OK
AT+SAPBR=3,1,"APN","internet.claro.com.ec"
vo:
{
OK
AT+SAPBR=1,1
//
OK
if
AT+HTTPIPINIT
SI
if
OK
Se
+SAPBR 1: DEACT
AT+HTTTPARA="URL","http://monitoreoGPS.000webhostapp.com /transporte /conexion/?id_vehiculo=CBJ111stxt_latitud=%s&stxt_longitud=%s&stxt_velocidad=%s"
OK

```

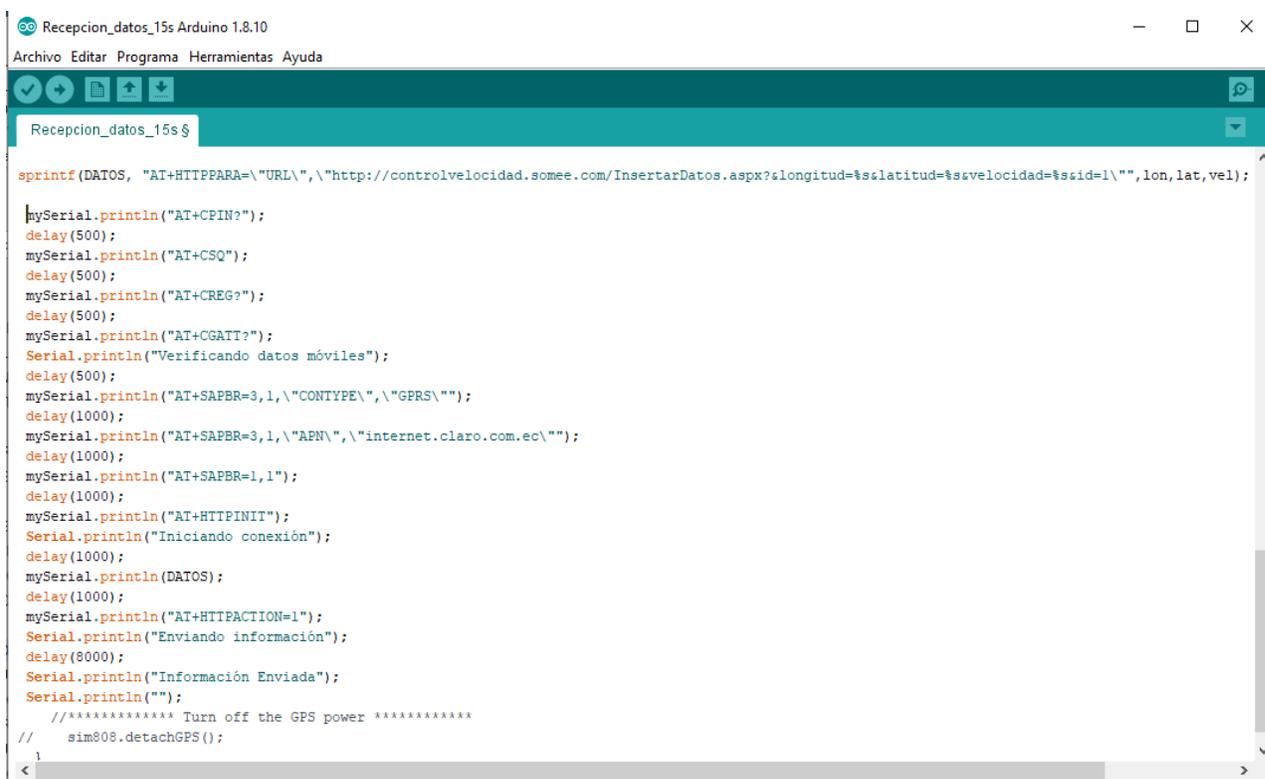
**Figura 3.15** Envío de datos mediante HTTP

Seguidamente de iniciar la comunicación HTTP se imprime el comando (AT+HTTTPARA="URL","\,"http://controlvelocidad.somee.com/InsertarDatos.aspx?&longitud=%s&latitud=%s&velocidad=%s&id=1\","lon,lat,vel) como se indica en la Figura 3.15, el cual se encarga de enviar la información de las variables receptada por el GPS a la base de datos que tiene la página web donde:

- **controlvelocidad.somee.com/InsertarDatos.aspx**: es la dirección URL a donde se conecta el módulo SIM 808 mediante datos móviles.

- **longitud=%s&latitud=%s&velocidad=%s:** Son las columnas de la base de datos a donde va a ingresar la información, además de qué %s es donde se va a colocar los valores de las variables.
- **id=1:** es la identificación sobre a qué vehículo pertenece la información enviada para guardarse en la base de datos.
- **lon, lat y vel:** Son las variables dentro de Arduino a las que va ingresando la información receptada del GPS para luego tomarlas y enviarlas a la página web.

Para finalizar el envío de datos a la página web se coloca el comando (AT+HTTPACTION=1) el cual realiza la acción de enviar la información a la base de datos de la página web. Además, se puede usar un comando adicional para conocer la cantidad de información enviada a la base de datos, y si esta fue receptada o leída por la página web por ejemplo (AT+HTTPACTION=0).



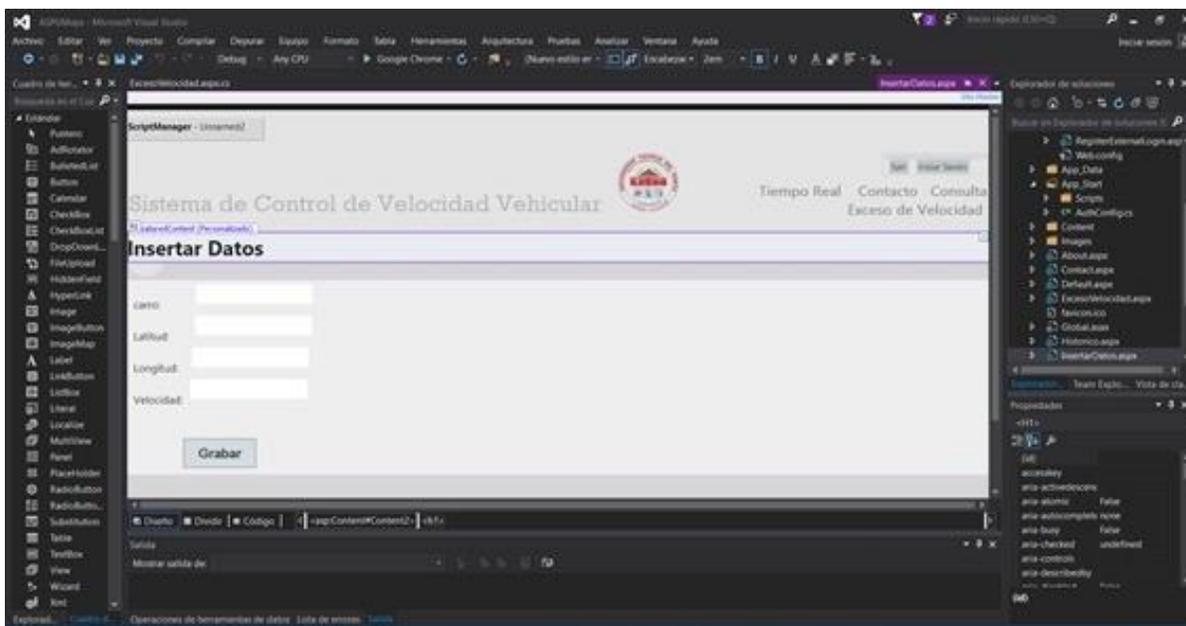
```

Recepcion_datos_15s Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda
Recepcion_datos_15s $
printf(DATOS, "AT+HTTPPARA=\URL\", \"http://controlvelocidad.somee.com/InsertarDatos.aspx?&longitud=%s&latitud=%s&velocidad=%s&id=1\", lon, lat, vel);
mySerial.println("AT+CPIN?");
delay(500);
mySerial.println("AT+CSQ");
delay(500);
mySerial.println("AT+CREG?");
delay(500);
mySerial.println("AT+CGATT?");
Serial.println("Verificando datos móviles");
delay(500);
mySerial.println("AT+SAPBR=3,1,\"CONTYPE\", \"GPRS\");
delay(1000);
mySerial.println("AT+SAPBR=3,1,\"APN\", \"internet.claro.com.ec\");
delay(1000);
mySerial.println("AT+SAPBR=1,1");
delay(1000);
mySerial.println("AT+HTTPIPINIT");
Serial.println("Iniciando conexión");
delay(1000);
mySerial.println(DATOS);
delay(1000);
mySerial.println("AT+HTTPACTION=1");
Serial.println("Enviando información");
delay(8000);
Serial.println("Información Enviada");
Serial.println("");
//***** Turn off the GPS power *****
// sim908.detachGPS();
1

```

**Figura 3.16** Comandos para el envío de Datos

En la Figura 3.16 se muestra los comandos completos utilizados para el envío, por lo que luego de esto ya se puede apreciar la información dentro de la web.



**Figura 3.17** Base de datos de la página web

En la Figura 3.17 se puede observar los recuadros de identificación del vehículo, latitud, longitud y velocidad en los cuales va ingresando la información enviada por el módulo SIM 808 y de donde se toma los datos para procesarlos dentro de la página web y mostrar con una interfaz más amigable para el usuario.

### 3.4.2 DISCUSIÓN ENVIO DE DATOS

#### FORTALEZAS

- El envío de datos puede realizarse de forma continua desde cualquier lugar del mundo, en el caso de este sistema se envía la información cada 15 segundos a la base de datos en la nube.
- La programación establecida es un bucle que se repite constante y automáticamente, sin tener necesidad de realizarla de forma manual.
- Mediante el monitor serial de Arduino IDE se puede realizar un seguimiento de la información que se está enviando y conocer si la página web está recibiendo los datos.

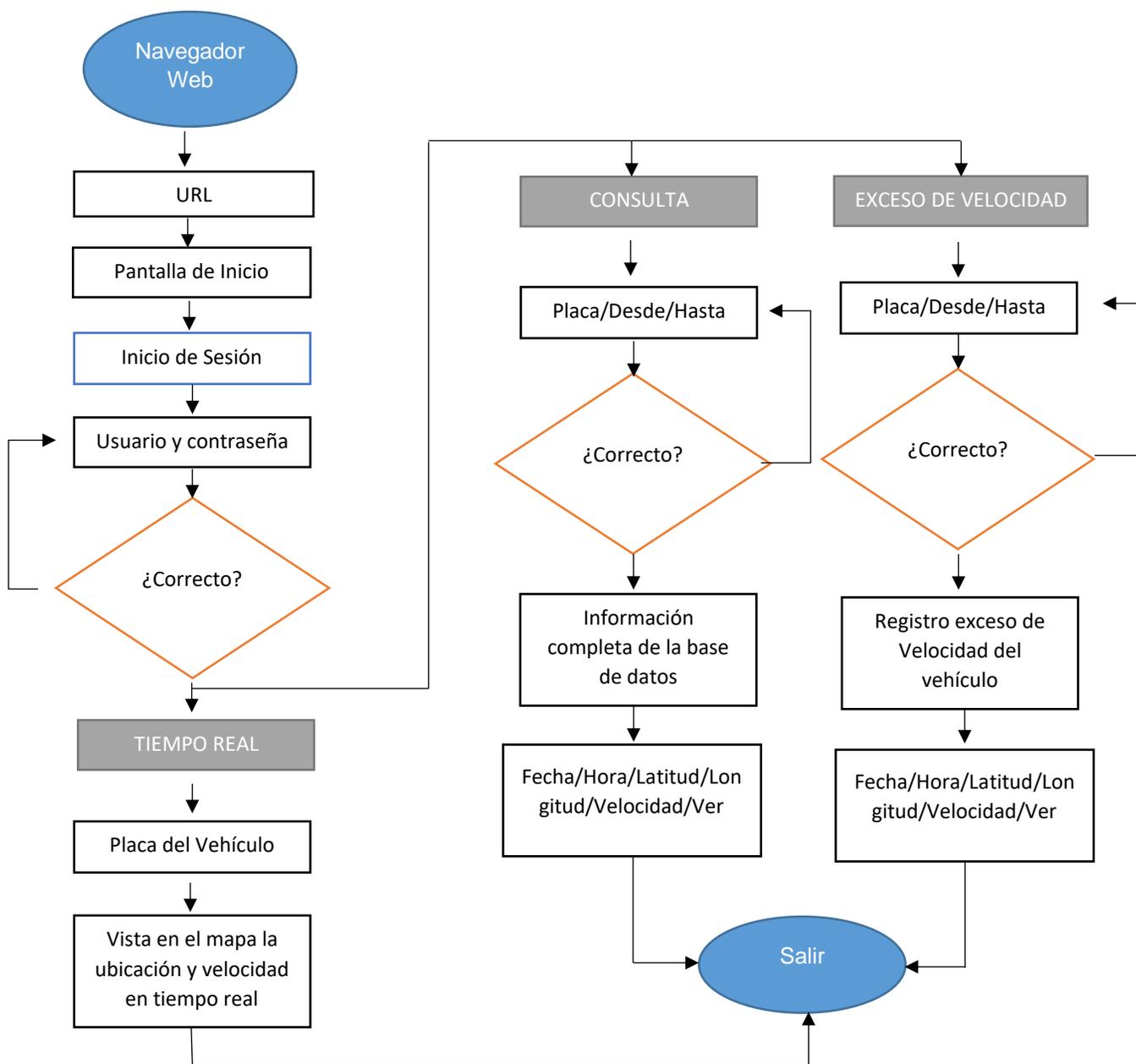
## **DEBILIDADES**

- Aunque el envío es continuo, hay que tener en cuenta que de la misma manera que un celular necesita tener cobertura de red, la SIM 808 también necesita tener suficiente señal para usar el GPRS y que este pueda enviar los datos al almacenamiento en la nube.
- La SIM en este caso de la operadora Claro, debe contar con un paquete activo de datos móviles para que logre enviar la información a la web.

### **3.5 VISUALIZACIÓN DE DATOS EN LA PÁGINA WEB**

Al haber enviado la información del GPS desde el módulo SIM 808 hasta la base de datos que se encuentra en la nube, ya se puede visualizar los registros desde la página web: coordenadas de latitud, longitud, velocidad, fecha y hora en la que se recibió la información.

Para entender más el desarrollo y manejo de la página web, a continuación, se presenta un flujograma sobre su funcionamiento:



**Figura 3.18** Funcionamiento página web

En la Figura 3.18 se muestra el funcionamiento completo de la página web, cuando se ingresa al URL aparece la página principal que pide un usuario y contraseña para permitir el acceso, luego se tiene tres pestañas principales: monitorear en tiempo real el vehículo, consultar toda la información recibida y ver los registros del exceso de velocidad. Se debe escoger la placa del vehículo, fecha y hora para poder realizar la consulta.

### 3.5.1 DESCRIPCIÓN PÁGINA WEB

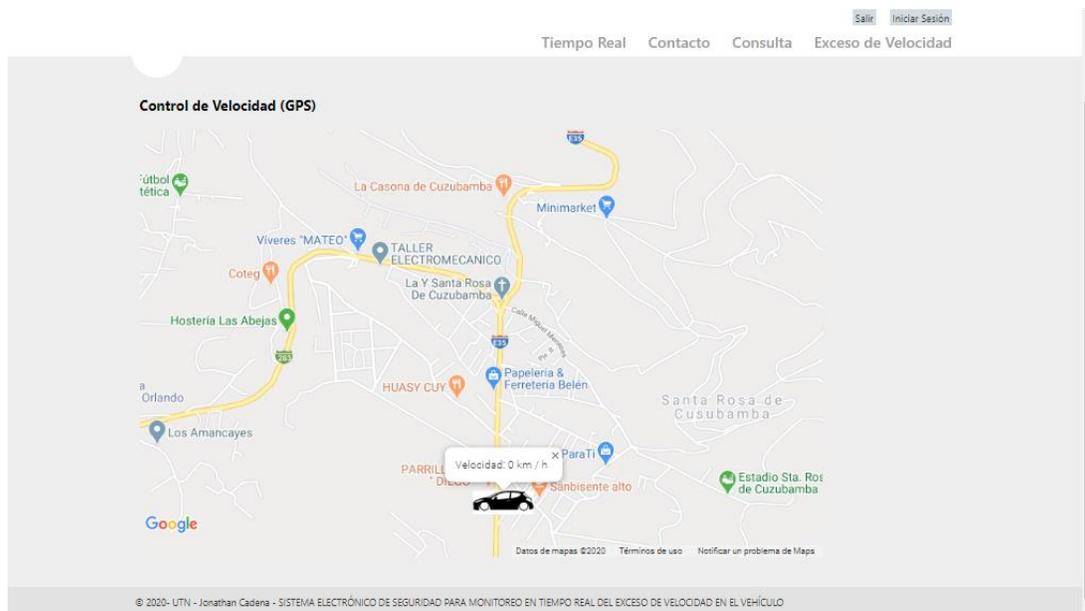


**Figura 3.19** Inicio de Sesión Pagina web

La página web es de fácil interacción para el usuario, la cual presenta sus opciones de forma clara y precisa. En la Figura 3.19 se puede apreciar las opciones de la página principal, las cuales aparecen en la parte superior derecha:

- **Inicio de Sesión:** Para tener acceso a la información de la página web se debe colocar un usuario y contraseña predeterminado, esto con el propósito de que los datos sean confidenciales y no todas las personas puedan obtener.
- **Tiempo Real:** Después de escoger la placa del vehículo que se quiere monitorear, se presenta su ubicación y velocidad en tiempo real.
- **Consulta:** Para consultar toda la información receptada dentro de la base de datos se debe escoger la placa del vehículo, las fechas y hora de los datos que se quiere examinar. El informe se presenta desde los datos más actuales hasta los más antiguos, visualizando 30 registros por página.
- **Exceso de velocidad:** En esta pantalla se muestra las ocasiones que el vehículo excedió los límites de velocidad permitidos y junto se encuentra toda la información de lo ocurrido: latitud, longitud, fecha, hora y velocidad; además mediante el botón “VER” se re direcciona a una nueva pestaña que indica la ubicación exacta dentro del mapa.

Seguido de ingresar con un usuario y contraseña, se presenta la opción de escoger la placa del vehículo que se quiere monitorear, posteriormente el sistema se dirige a la opción de tiempo real, en donde dentro del mapa de Google se muestra un logo que representa al vehículo y un pequeño cuadro que indica la velocidad a la que se encuentra.



**Figura 3.20** Pantalla TIEMPO REAL

En el caso de la Figura 3.20 se tiene una velocidad de 0 km/h debido a que se realizó una prueba estática cerca al sector de La Y de Cuzubamba/Cayambe.



**Figura 3.21** Reporte por fechas

La Figura 3.21 muestra la opción de consulta donde se presenta el reporte del momento que el usuario desee, por lo que se debe escoger la placa del vehículo, fechas y horas.

Reporte de Velocidad

Desde: 10/02/2020 00:00 Hasta: 11/02/2020 07:00 Vehículo: CBJ0111 Consultar

Fecha	Latitud	Longitud	Velocidad	Ver
11/02/2020 06:59:54	0.28564	-78.23609	87.00	Ver
11/02/2020 06:59:39	0.26197	-78.23750	94.00	Ver
11/02/2020 06:59:22	0.27828	-78.23813	92.00	Ver
11/02/2020 06:59:07	0.27482	-78.23908	93.00	Ver
11/02/2020 06:58:50	0.27153	-78.24056	88.00	Ver
11/02/2020 06:58:35	0.26842	-78.24191	83.00	Ver
11/02/2020 06:58:20	0.26521	-78.24319	78.00	Ver
11/02/2020 06:53:07	0.23989	-78.25454	65.00	Ver
11/02/2020 06:52:51	0.23773	-78.25594	57.00	Ver
11/02/2020 06:40:32	0.18388	-78.21373	86.00	Ver
11/02/2020 06:40:12	0.18152	-78.21122	90.00	Ver
11/02/2020 06:39:46	0.17714	-78.20708	89.00	Ver
11/02/2020 06:39:30	0.17394	-78.20590	83.00	Ver

Hosted Windows Virtual Server. 2.5GHz CPU, 1.5GB RAM, 60GB SSD. Try it now for \$1!

**Figura 3.22** Reporte de velocidad

En la Figura 3.22 se puede observar la consulta realizada, presentando un reporte que consta de la fecha (dd/mm/aa), hora, latitud, longitud, velocidad y un botón con la opción “VER”, el cual al presionar re direcciona a una nueva pestaña donde se indica la ubicación del registro almacenado.

El reporte se muestra desde los datos más actuales que se obtuvo hasta los más antiguos con una capacidad de 30 casos por página.

Reporte de Exceso de Velocidad

Desde: 10/02/2020 00:00 Hasta: 11/02/2020 07:15 Vehículo: CBJ0111 Consultar

Fecha	Latitud	Longitud	Velocidad Km/h	
11/02/2020 07:14:53	0.34669	-78.13389	64.00	Ver
11/02/2020 07:13:15	0.34562	-78.13835	59.00	Ver
11/02/2020 07:12:59	0.34420	-78.14040	51.00	Ver
11/02/2020 07:12:43	0.34299	-78.14211	55.00	Ver
11/02/2020 07:11:07	0.33974	-78.14936	84.00	Ver
11/02/2020 07:04:11	0.33197	-78.20950	89.00	Ver
11/02/2020 07:03:55	0.33008	-78.21250	94.00	Ver
11/02/2020 07:03:38	0.32726	-78.21526	105.00	Ver
11/02/2020 07:03:23	0.32437	-78.21784	93.00	Ver
11/02/2020 07:03:06	0.32116	-78.21999	99.00	Ver
11/02/2020 07:02:52	0.31776	-78.22224	104.00	Ver
11/02/2020 06:52:51	0.23773	-78.25594	57.00	Ver
11/02/2020 06:23:48	0.07351	-78.13890	86.00	Ver

Hosted Windows Virtual Server. 2.5GHz CPU, 1.5GB RAM, 60GB SSD. Try it now for \$1!

**Figura 3.23** Reporte del exceso de velocidad

En la pantalla del reporte Exceso de Velocidad, se realiza la consulta de la misma manera que en la pantalla de Reporte de velocidad, escogiendo la placa del vehículo, fecha y hora; de esta manera se presenta las veces que se ha excedido los límites de velocidad, sea en ciudad o carretera. En la Figura 3.23 se puede notar la pantalla del exceso de velocidad.

### 3.5.2 DISCUSIÓN PAGINA WEB

#### FORTALEZAS

- Se puede ingresar a la página web desde cualquier lugar del mundo mediante un computador, tableta, celular o cualquier dispositivo que tenga acceso al servicio de internet.
- La página web en general contiene una interfaz cómoda para el usuario, donde se puede navegar por todas las opciones de manera interesante, observando siempre un título de lo que se puede apreciar.

- Dentro de la opción TIEMPO REAL se cuenta con un mapa donde se observa la ubicación y velocidad del vehículo en ese instante. La página se vuelve a cargar automáticamente cada 16 segundos.
- La página web es de uso exclusivo para el beneficiario, esta seguridad se logra mediante la implementación de un usuario y contraseña para el acceso, por lo que el sistema es confiable.

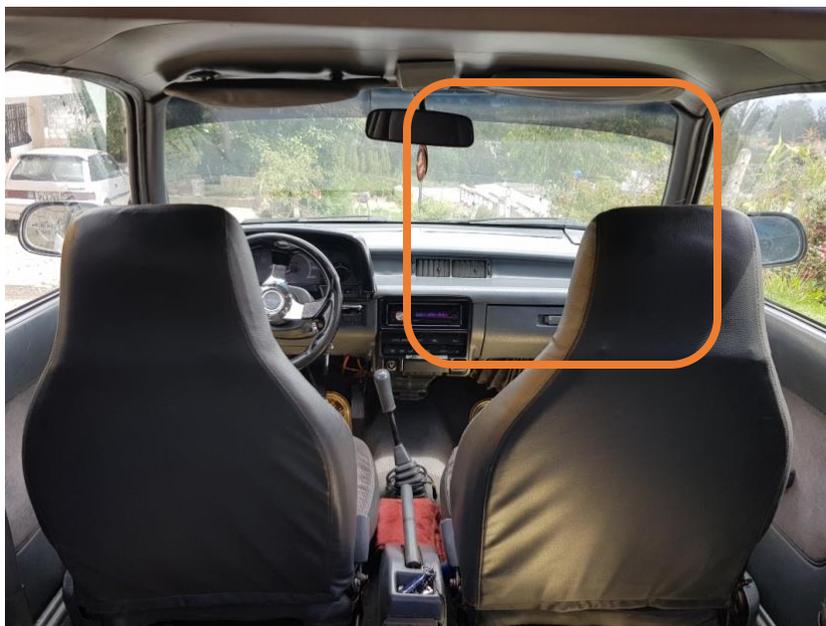
## **DEBILIDADES**

- Para acceder a los datos de la página web es necesario contar con acceso a internet sin importar el dispositivo que se utilice.
- En el *hosting* gratuito usado para el funcionamiento de la página web, tienden a realizar mantenimiento, por lo que en un periodo aproximadamente menor a 24 horas no se puede acceder a la página web.

## **3.6 INSTALACIÓN Y PRUEBAS EN TIEMPO REAL**

### **3.6.1 MONTAJE**

El sistema electrónico de seguridad para el monitoreo de la velocidad después de haber superado de manera positiva las pruebas iniciales de campo en cada fase, se instaló en un sitio fijo dentro del vehículo para que su operación sea continua y se pueda verificar el funcionamiento en largos periodos de tiempo.



**Figura 3.24** Vehículo de Instalación

El sistema se colocó en un vehículo de la marca Suzuki Forsa II año 95, en la Figura 3.24 se indica que la instalación se realizó en la parte frontal derecha dentro del habitáculo del automóvil donde existe mayor espacio para los dispositivos junto con sus antenas.



**Figura 3.25** Montaje antena GPS

Primero se conectó desde la batería del vehículo el regulador de voltaje DC-DC XL4015 para obtener una corriente constante, seguidamente se desmontó la tapicería del lado derecho del

parante que como se muestra en la Figura 3.25 fue para colocar la antena del GPS en la parte superior donde se tiene una mayor conexión con los satélites.



**Figura 3.26** Instalación dispositivo

Finalmente, se conectó el sistema electrónico a la fuente de alimentación proveniente del regulador de voltaje, todos los dispositivos se montaron dentro de una caja de (9 x 7 x 6) cm y se colocó en la parte posterior derecha del tablero del vehículo como se indica en la Figura 3.26, siendo este un sitio seguro donde personas sin autorización no pueden manipular el sistema.

### **3.6.2 PRUEBA DE FUNCIONAMIENTO EN CIUDAD**

En el Ecuador el límite máximo de velocidad dentro de la ciudad es de 50 km/h, por lo tanto, si el dispositivo lee una velocidad mayor a esta, se almacena la información en la base de datos para luego observar dentro de la pestaña de exceso de velocidad en la página web, con datos de ubicación, velocidad, fecha y hora de lo ocurrido.



**Figura 3.27** Imagen tomada en la ciudad de Ibarra el día 11 de febrero, 07:15

La Figura 3.27 muestra la velocidad del vehículo en el momento que excedió el límite de velocidad, la cual después se comparó con los datos que se encuentran en la base de datos de la página web, en este caso la velocidad marcada es de 58 km/h.

Se realizó la consulta dentro de la página web sobre los excesos de velocidad donde se confirmó que el vehículo excedió los límites de velocidad dentro de la ciudad de Ibarra, en la Tabla 3.1 se presenta la información tomada de la base de datos donde se tiene que el día 11 de febrero del 2020 a las 07:15 se alcanzó una velocidad de 58Km/h.

**Tabla 3.3** Excesos de velocidad almacenados el día 11 de febrero del 2020

Fecha	Latitud	Longitud	Velocidad Km/h
11/02/2020 07:15:23	0.34800	-78.13187	58.00
11/02/2020 07:14:53	0.34669	-78.13389	64.00
11/02/2020 07:13:15	0.34562	-78.13835	59.00
11/02/2020 07:12:59	0.34420	-78.14040	51.00
11/02/2020 07:12:43	0.34299	-78.14936	55.00
11/02/2020 07:11:07	0.33974	-78.14936	84.00

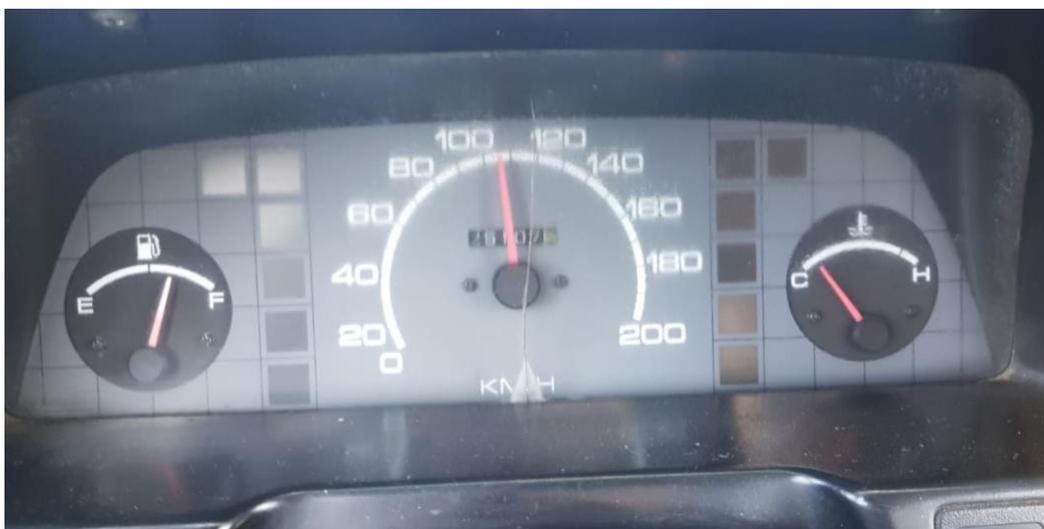


**Figura 3.28** Mapa exceso de velocidad en ciudad

Para visualizar la ubicación exacta de donde se ocasionó el exceso de velocidad se presiona el botón VER dentro de la página web el cual re direcciona a una nueva página con el mapa de Google graficando los datos, la Figura 3.28 muestra que el registro se produjo en la ciudad de Ibarra, cerca del Redondel de la madre en la E35.

### 3.6.3 PRUEBA DE FUNCIONAMIENTO EN CARRETERA

El límite de velocidad dentro del sector perimetral es de 90 km/h y en carretera de 100 km/h, por lo tanto, dependiendo la zona en el que el vehículo se encuentre si la lectura supera la velocidad permitida automáticamente se almacenará en la nube para su posterior consulta desde la página web.



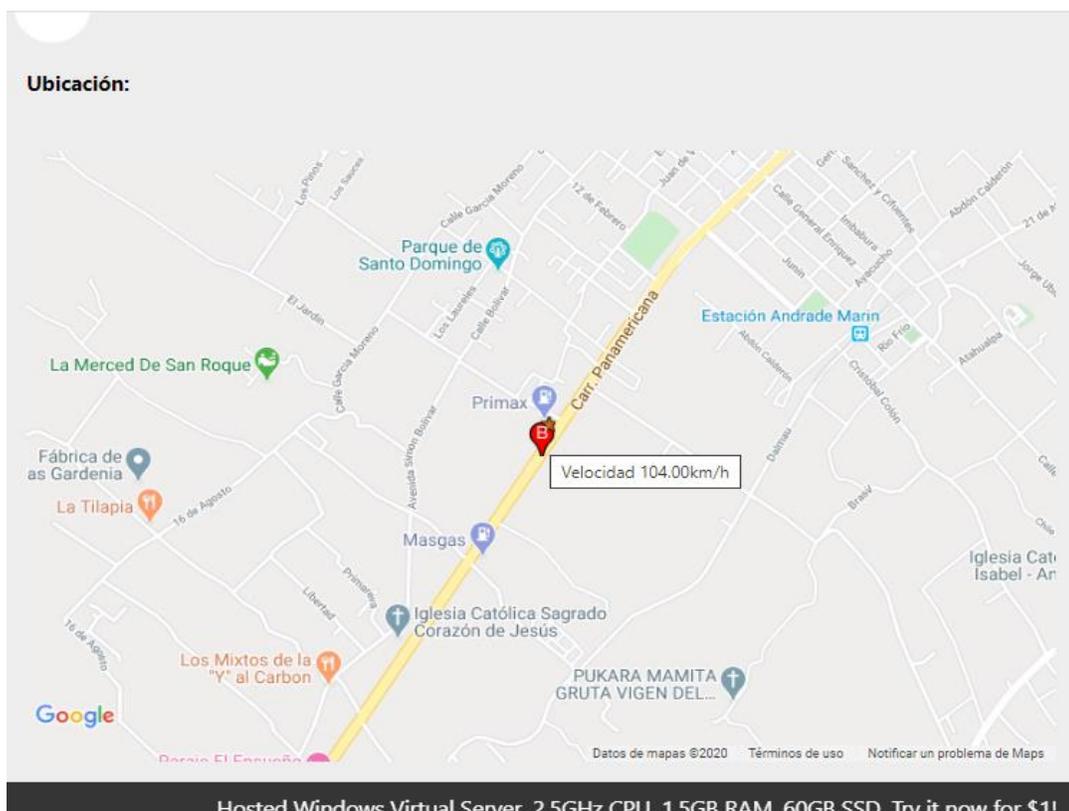
**Figura 3.29** Imagen tomada en carretera el día 11 de febrero a las 07:02:52

La Figura 3.29 indica la velocidad en el panel del vehículo dentro de una zona rural, donde se excedió el límite de velocidad que aproximadamente fue 104 km/h, por lo que todos los datos del registro se almacenaron en el apartado de “Exceso de velocidad” dentro de la página web.

Revisando la información afirmativamente existe el registro de cuando el vehículo excedió los límites de velocidad, en la Tabla 3.2 se observa la información almacenada, comparando con la fotografía se encuentra que fue el día 11 de febrero del 2020 a las 07:02: 52 con una velocidad de 104km/h.

**Tabla 3.4** Excesos de velocidad en carretera

Fecha	Latitud	Longitud	Velocidad Km/h
11/02/2020 07:04:11	0.33197	-78.20950	89.00
11/02/2020 07:03:55	0.33008	-78.21250	94.00
11/02/2020 07:03:38	0.32726	-78.21526	105.00
11/02/2020 07:03:23	0.32437	-78.21784	93.00
11/02/2020 07:03:06	0.32116	-78.21999	99.00
11/02/2020 07:02:52	0.31776	-78.22224	104.00



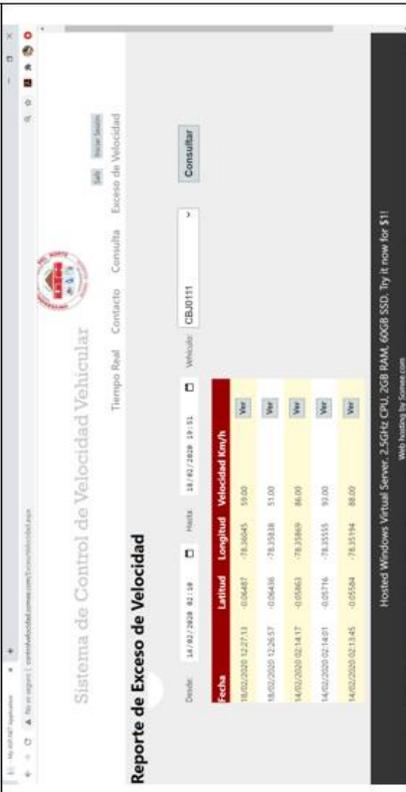
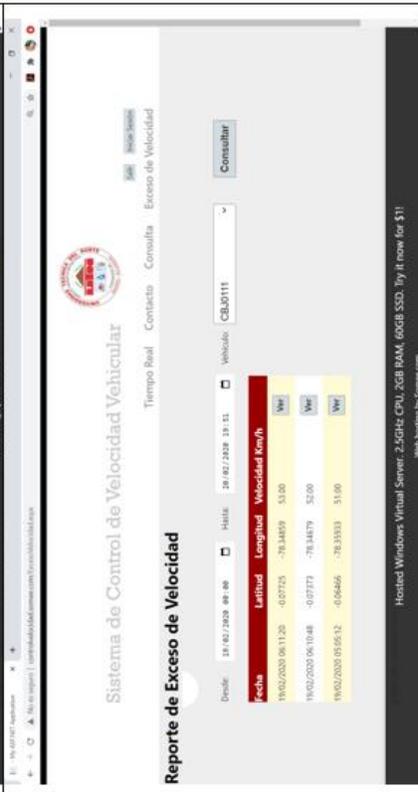
**Figura 3.30** Mapa exceso de velocidad en carretera

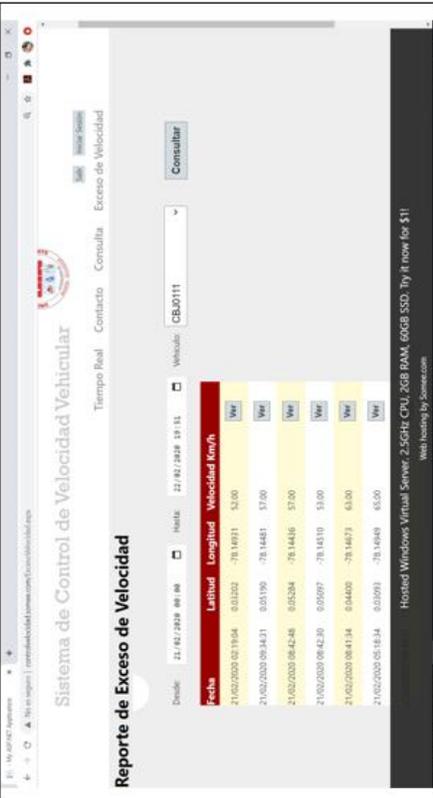
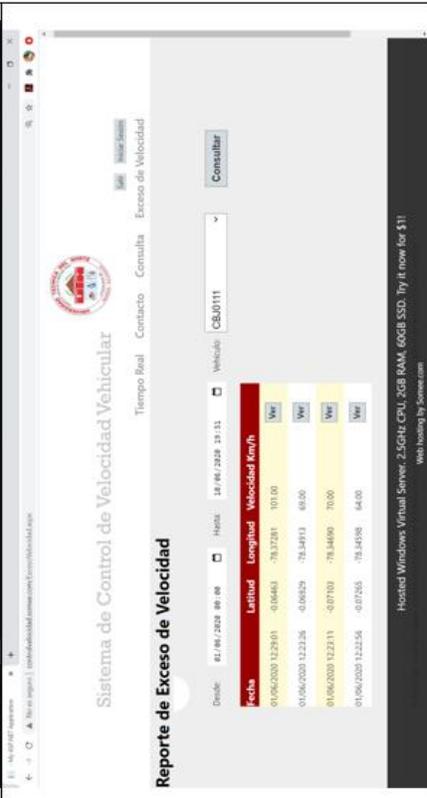
De la misma manera que en el anterior ejemplo, para conocer en donde se registró el exceso de velocidad nos dirigimos al botón VER el cual abre una nueva pestaña y grafica las coordenadas almacenadas. Dentro del mapa se señala con un punto de color rojo el lugar de donde se obtuvo la información, como se muestra en la Figura 3.30 fue en la Panamericana Norte cerca del sector Atuntaqui, provincia de Imbabura.

### 3.6.4 ANÁLISIS DE PRUEBAS

En la Tabla 3.3 se muestra cuatro pruebas adicionales a las que se presentó anteriormente, donde se tiene las fotografías que fueron tomadas del velocímetro en el vehículo con la respectiva fecha y hora, además se puede observar la información que indica la página web sobre los excesos de velocidad. Con la información proporcionada se puede comparar si el sistema está en perfecto funcionamiento o no.

Tabla 3.5 Comparación Pruebas

DESCRIPCIÓN	FOTOGRAFÍA	DETALLE	IMAGEN																								
<p>Fotografía del día 14/02/2020 02:14</p>		<p>02:14:01 93km</p>	 <table border="1"> <thead> <tr> <th>Fecha</th> <th>Latitud</th> <th>Longitud</th> <th>Velocidad Km/h</th> </tr> </thead> <tbody> <tr> <td>14/02/2020 02:14:01</td> <td>-036487</td> <td>-78.36645</td> <td>93.00</td> </tr> <tr> <td>14/02/2020 02:26:57</td> <td>-036438</td> <td>-78.35338</td> <td>51.00</td> </tr> <tr> <td>14/02/2020 02:34:17</td> <td>-036463</td> <td>-78.37286</td> <td>86.00</td> </tr> <tr> <td>14/02/2020 02:34:01</td> <td>-036716</td> <td>-78.35555</td> <td>93.00</td> </tr> <tr> <td>14/02/2020 02:33:45</td> <td>-036584</td> <td>-78.35194</td> <td>86.00</td> </tr> </tbody> </table>	Fecha	Latitud	Longitud	Velocidad Km/h	14/02/2020 02:14:01	-036487	-78.36645	93.00	14/02/2020 02:26:57	-036438	-78.35338	51.00	14/02/2020 02:34:17	-036463	-78.37286	86.00	14/02/2020 02:34:01	-036716	-78.35555	93.00	14/02/2020 02:33:45	-036584	-78.35194	86.00
Fecha	Latitud	Longitud	Velocidad Km/h																								
14/02/2020 02:14:01	-036487	-78.36645	93.00																								
14/02/2020 02:26:57	-036438	-78.35338	51.00																								
14/02/2020 02:34:17	-036463	-78.37286	86.00																								
14/02/2020 02:34:01	-036716	-78.35555	93.00																								
14/02/2020 02:33:45	-036584	-78.35194	86.00																								
<p>Fotografía tomada el 19/02/2020 06:10</p>		<p>06:10:48 52km</p>	 <table border="1"> <thead> <tr> <th>Fecha</th> <th>Latitud</th> <th>Longitud</th> <th>Velocidad Km/h</th> </tr> </thead> <tbody> <tr> <td>19/02/2020 06:11:20</td> <td>-037725</td> <td>-78.34829</td> <td>53.00</td> </tr> <tr> <td>19/02/2020 06:10:48</td> <td>-037773</td> <td>-78.34879</td> <td>52.00</td> </tr> <tr> <td>19/02/2020 05:56:12</td> <td>-036466</td> <td>-78.33553</td> <td>51.00</td> </tr> </tbody> </table>	Fecha	Latitud	Longitud	Velocidad Km/h	19/02/2020 06:11:20	-037725	-78.34829	53.00	19/02/2020 06:10:48	-037773	-78.34879	52.00	19/02/2020 05:56:12	-036466	-78.33553	51.00								
Fecha	Latitud	Longitud	Velocidad Km/h																								
19/02/2020 06:11:20	-037725	-78.34829	53.00																								
19/02/2020 06:10:48	-037773	-78.34879	52.00																								
19/02/2020 05:56:12	-036466	-78.33553	51.00																								

<p>Fotografía del día 21/02/2020 08:41</p>		<p>08:41:34 63km</p>	 <p><b>Reporte de Exceso de Velocidad</b></p> <table border="1"> <thead> <tr> <th>Fecha</th> <th>Latitud</th> <th>Longitud</th> <th>Velocidad Km/h</th> <th>Ver</th> </tr> </thead> <tbody> <tr> <td>21/02/2020 08:41:34</td> <td>-0.0325</td> <td>-78.1491</td> <td>63.00</td> <td>Ver</td> </tr> <tr> <td>21/02/2020 09:18:31</td> <td>0.01590</td> <td>-78.14481</td> <td>57.00</td> <td>Ver</td> </tr> <tr> <td>21/02/2020 08:42:48</td> <td>0.02384</td> <td>-78.14416</td> <td>57.00</td> <td>Ver</td> </tr> <tr> <td>21/02/2020 08:42:30</td> <td>0.05997</td> <td>-78.14310</td> <td>53.00</td> <td>Ver</td> </tr> <tr> <td>21/02/2020 08:41:34</td> <td>0.04450</td> <td>-78.14673</td> <td>63.00</td> <td>Ver</td> </tr> <tr> <td>21/02/2020 05:18:34</td> <td>0.03093</td> <td>-78.14548</td> <td>65.00</td> <td>Ver</td> </tr> </tbody> </table>	Fecha	Latitud	Longitud	Velocidad Km/h	Ver	21/02/2020 08:41:34	-0.0325	-78.1491	63.00	Ver	21/02/2020 09:18:31	0.01590	-78.14481	57.00	Ver	21/02/2020 08:42:48	0.02384	-78.14416	57.00	Ver	21/02/2020 08:42:30	0.05997	-78.14310	53.00	Ver	21/02/2020 08:41:34	0.04450	-78.14673	63.00	Ver	21/02/2020 05:18:34	0.03093	-78.14548	65.00	Ver
Fecha	Latitud	Longitud	Velocidad Km/h	Ver																																		
21/02/2020 08:41:34	-0.0325	-78.1491	63.00	Ver																																		
21/02/2020 09:18:31	0.01590	-78.14481	57.00	Ver																																		
21/02/2020 08:42:48	0.02384	-78.14416	57.00	Ver																																		
21/02/2020 08:42:30	0.05997	-78.14310	53.00	Ver																																		
21/02/2020 08:41:34	0.04450	-78.14673	63.00	Ver																																		
21/02/2020 05:18:34	0.03093	-78.14548	65.00	Ver																																		
<p>Fotografía tomada el 01/06/2020</p>		<p>101km 12:29:01</p>	 <p><b>Reporte de Exceso de Velocidad</b></p> <table border="1"> <thead> <tr> <th>Fecha</th> <th>Latitud</th> <th>Longitud</th> <th>Velocidad Km/h</th> <th>Ver</th> </tr> </thead> <tbody> <tr> <td>01/06/2020 12:29:01</td> <td>-0.04643</td> <td>-78.37281</td> <td>101.00</td> <td>Ver</td> </tr> <tr> <td>01/06/2020 12:23:26</td> <td>-0.08229</td> <td>-78.34913</td> <td>68.00</td> <td>Ver</td> </tr> <tr> <td>01/06/2020 12:23:11</td> <td>-0.07103</td> <td>-78.34800</td> <td>70.00</td> <td>Ver</td> </tr> <tr> <td>01/06/2020 12:22:56</td> <td>-0.07265</td> <td>-78.34708</td> <td>64.00</td> <td>Ver</td> </tr> </tbody> </table>	Fecha	Latitud	Longitud	Velocidad Km/h	Ver	01/06/2020 12:29:01	-0.04643	-78.37281	101.00	Ver	01/06/2020 12:23:26	-0.08229	-78.34913	68.00	Ver	01/06/2020 12:23:11	-0.07103	-78.34800	70.00	Ver	01/06/2020 12:22:56	-0.07265	-78.34708	64.00	Ver										
Fecha	Latitud	Longitud	Velocidad Km/h	Ver																																		
01/06/2020 12:29:01	-0.04643	-78.37281	101.00	Ver																																		
01/06/2020 12:23:26	-0.08229	-78.34913	68.00	Ver																																		
01/06/2020 12:23:11	-0.07103	-78.34800	70.00	Ver																																		
01/06/2020 12:22:56	-0.07265	-78.34708	64.00	Ver																																		

### **3.6.5 RESULTADO DE LAS PRUEBAS**

Después de haber realizado las pruebas pertinentes enviando la información en varias ubicaciones y con diferentes circunstancias de manera continua cada 15 segundos (aproximadamente 5 760 veces al día), se puede decir que el sistema electrónico de seguridad para el monitoreo en tiempo real del exceso de velocidad tuvo un éxito en su diseño e implementación en el vehículo, con ciertas fortalezas y debilidades del sistema en general que a continuación se detalla.

#### **FORTALEZAS**

- El Sistema tuvo una excelente respuesta cumpliendo así con el objetivo principal de monitorear en tiempo real la velocidad del vehículo y almacenar los datos del momento que excedió los límites de velocidad para en otro momento consultar estos registros.
- Se trata de un sistema confiable en cuestión de que la información receptada y enviada tanto en fecha, hora, velocidad y coordenadas son correctas, además de que el envío de datos continua sin importar que el vehículo se encuentre apagado.
- Es un sistema de bajo costo en cuanto a instalación y mantenimiento, pero de grandes prestaciones para un control continuo y monitoreo en tiempo real del vehículo, el cual puede tener una gran variedad de usos en el ámbito de transporte.
- La forma de monitoreo y consulta es de fácil acceso desde cualquier dispositivo conectado a una red de internet, donde la interfaz es amigable para que el usuario.

#### **DEBILIDADES**

- El usuario debe estar pendiente de contratar mensualmente un paquete de datos móviles para que el sistema siga funcionando sin inconvenientes.
- Si por algún motivo se desconecta la alimentación del sistema, sería necesario volver a encender el dispositivo manualmente

## CAPÍTULO IV

### 4 CONCLUSIONES Y RECOMENDACIONES

#### 4.1 CONCLUSIONES

- Se desarrolló la interfaz entre la plataforma de prototipos electrónica Arduino y el módulo SIM 808 conectados físicamente entre sí mediante los pines Tx, Rx y GND. Para obtener una correcta comunicación entre ambos dispositivos se realizó la programación dentro del IDE de Arduino mediante lenguaje processing lo que permitió que el sistema GPS (el mismo que se encuentra incorporado por defecto en el módulo SIM 808), tome lectura de la velocidad del vehículo en tiempo real. Además, proporcionar coordenadas con 5 decimales las cuales permiten obtener una referencia de localización con un rango de precisión menor a 2,5 metros dando como resultado el 99.9% de coincidencia entre la información receptada por el GPS y la señalada por el velocímetro del vehículo.
- Se implementó el módulo Shield con tecnología GPRS al módulo Arduino mediante la programación por comandos AT dentro del IDE de Arduino, configurando una conexión HTTP la cual permite enviar la información receptada por el GPS a una base de datos en la página web (servidor externo) cada 15 segundos. Para el envío de dicha información es indispensable el uso continuo de datos móviles motivo por el cual se adquirió un SIM de la operadora Claro que cuenta con mayor cobertura a nivel nacional.
- Se empleó un servidor gratuito en la nube llamado Somee el cual se creó un dominio (controlvelocidad.somee.com) donde se cargaron tanto los archivos de la página web diseñada en la herramienta de desarrollo Visual Studio mediante lenguaje ASP.net, como una base de datos creada mediante tablas en Microsoft SQL Server en la que ingresa la información enviada por el SIM 808. Este dominio permite que el usuario realice consultas en tiempo real de forma rápida y sencilla. La página web actualiza sus registros cada 16 segundos en caso de que la visualización en tiempo real y en caso de revisar un

histórico de velocidad la página muestra 30 resultados por pestaña. Para monitorear el vehículo y revisar el exceso de velocidad desde cualquier parte del mundo es necesario ingresar a la página web con la ayuda de internet, un usuario y una contraseña.

## 4.2 RECOMENDACIONES

- Al realizar la conexión física se recomienda usar una placa de circuito impreso (baquelita) junto con pines macho y hembra para que los módulos se ensamblen uno sobre otro con el propósito de que no se presente un problema de conexión mientras el vehículo se encuentra en movimiento. Además, al momento de realizar la programación se debe configurar la cantidad de decimales en la variable de coordenadas, tomando en cuenta que mientras más decimales mejor precisión de ubicación y velocidad se tendrá.
- Tener presente que antes de realizar la configuración HTTP se deben crear variables distintas para cada dato que se obtiene del GPS, de esta manera se enviarán de forma ordenada a la base de datos e irán ingresando en las tablas correspondientes. Adicionalmente, es fundamental instalar las antenas de GPS y GPRS del Módulo SIM en un área despejada para que puedan operar correctamente.
- Emplear un almacenamiento para la página web que no requiera excesivos mantenimientos y por lo tanto permita al usuario utilizar el sitio las 24 horas del día sin interrupciones. También es necesario mantener el SIM claro con un paquete continuo de datos móviles, para evitar que el módulo deje de enviar la información y perder conexión al apagarse el módulo.
- Implementar una batería de emergencia que alimente al sistema en caso de que la batería propia del vehículo se desconecte por problemas o percances externos, por ejemplo, un robo.
- Para futuros estudios se recomienda añadir sensores en el vehículo tales como un medidor de temperatura en el sistema de frenado, un indicador del ángulo de giro en la

dirección, entre otros, con el fin de utilizar el sistema con la función una caja negra ante un accidente de tránsito, para conocer las condiciones previas al percance. Además, se recomienda realizar un estudio de investigación para regular los límites de velocidad o verificar el porcentaje de reducción de accidentes de tránsito con el uso del sistema.

## BIBLIOGRAFÍA

1. (2019). Obtenido de Aprendiendo Arduino: <https://aprendiendoarduino.wordpress.com/tag/entradas-analogicas/>
2. Agencia Nacional de Tránsito. (14 de Noviembre de 2016). *Reglamento a ley de transporte terrestre, tránsito y seguridad vial*. Obtenido de Agencia Nacional de Tránsito: <https://www.ant.gob.ec/index.php/ant/base-legal/reglamento-general-para-la-aplicacion-de-la-lotstsv>
3. Agencia Nacional de Tránsito. (2018). *Siniestros 2017*. Obtenido de <https://www.ant.gob.ec/index.php/ley-de-transparencia/ley-de-transparencia-2020/file/-4959-siniestros-diciembre-2017>
4. Ahuja, P., & Bhavsar, K. (2018). Microcontroller Based Smart Helmet Using GSM & GPRS. *2nd International Conference on Trends in Electronics and Informatics (ICOEI)*. doi:10.1109/ICOEI.2018.8553802
5. Al Agha, K., Pujolle, G., & Ali-Yahiya, T. (2016). *Mobile and Wireless Networks* (Primera ed.). John Wiley & Sons, Incorporated. Obtenido de <https://ebookcentral.proquest.com/lib/utn/detail.action?docID=4648734>
6. Alda Montes, J. F., Baldominos Gómez, A., García Nieto, J. M., González Cabañas, J. C., Mochón Morcillo, F., & Navas Delgado, I. (2016). *Introducción al Big Data*. Madrid: García-Maroto Editores.
7. Arenas Mas, M. (Junio de 2008). Diseño e implementación de un sistema de adquisición de aceleraciones con procesamiento mediante microcontrolador. Sevilla, España: Universidad de Sevilla. Obtenido de <http://bibing.us.es/proyectos/abreproy/11638/fichero/Capitulo+2.pdf>
8. Banzi, M., & Shiloh, M. (2016). *Introducción a Arduino*. Madrid, España: Anaya Multimedia.
9. Barra Zapata, O. E., & Barra Zapata, F. (2011). *Microcontroladores PIC con programación PBP. RA-MA*. Obtenido de [https://books.google.com.ec/books?id=cI6fDwAAQBAJ&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.ec/books?id=cI6fDwAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
10. Berné Valero, J., Garrido Villén, N., & Capilla Romá, R. (2019). *GNSS: GPS, GALILEO, GLONASS, BEIDOU Fundamentos y métodos de posicionamiento*. Valencia, España: Universitat Politècnica de València. Obtenido de <https://ebookcentral.proquest.com/lib/utnortesp/reader.action?docID=5809635&query=division%20BGPS#>
11. Bibri, S. E. (2015). *The Shaping of Ambient Intelligence and the Internet of Things*. Atlantis Press.

12. Caicedo Pedrera, A. (2017). *ARDUINO PARA PRINCIPIANTES*. Obtenido de [https://books.google.es/books?hl=es&lr=&id=Fw\\_RDgAAQBAJ&oi=fnd&pg=PA5&dq=pines+arduino&ots=QUjAWjM883&sig=8vFvMpoW49EMBBsarxbtre9MOTA#v=onepage&q=pines+arduino&f=false](https://books.google.es/books?hl=es&lr=&id=Fw_RDgAAQBAJ&oi=fnd&pg=PA5&dq=pines+arduino&ots=QUjAWjM883&sig=8vFvMpoW49EMBBsarxbtre9MOTA#v=onepage&q=pines+arduino&f=false)
13. Castro Correa, J., Sepúlveda Mora, S., Medina Delgado, B., Guevara Ibarra, D., & López Bustamante, O. (2019). Sistema de geolocalización de vehículos a través de la red GSM/GPRS y tecnología Arduino. *EIA*, 16, 145, 157. doi:<https://doi.org/10.24050/reia.v16i31.1269>
14. Das. (2016). *Mobile Terminal Receiver Design* (Primera ed.). John Wiley & Sons, Incorporated. Obtenido de <https://ebookcentral.proquest.com/lib/utn/detail.action?docID=4698000>
15. Dincer, A., & Uraz, B. (2013). *Google Maps API Cookbook*. Packt Publishing Ltd. Obtenido de <https://ebookcentral.proquest.com/lib/utn/detail.action?docID=1572949>
16. eltelégrafo. (20 de Julio de 2017). *Costo de accidentes viales puede llegar al 3% del PIB de los países*. Obtenido de eltelégrafo: <https://www.eltelegrafo.com.ec/noticias/ecuador/3/costo-de-accidentes-viales-puede-llegar-al-3-del-pib-de-los-paises>
17. Fylaktopoulos, G., Goumas, G., Skolarikis, M., Sotiropoulos, A., & Maglogiannis, I. (2016). An overview of platforms for cloud based development. *Springer Plus*.
18. Goilav, N., & Loi, G. (2016). *Arduino: Aprender a desarrollar para crear objetos inteligentes*. Barcelona: Angel M Sanchez Conejo. Obtenido de [https://books.google.es/books?hl=es&lr=&id=R6RCxQl\\_H6YC&oi=fnd&pg=PA9&dq=diferentes+shield+arduino&ots=0bAeu6nm02&sig=mGuSTfo\\_tNVPAnkKJFseEf9ZzWw#v=onepage&q=diferentes+shield+arduino&f=false](https://books.google.es/books?hl=es&lr=&id=R6RCxQl_H6YC&oi=fnd&pg=PA9&dq=diferentes+shield+arduino&ots=0bAeu6nm02&sig=mGuSTfo_tNVPAnkKJFseEf9ZzWw#v=onepage&q=diferentes+shield+arduino&f=false)
19. González Villamil, C. (6 de Abril de 2016). *Smartphones*. Obtenido de [https://cincodias.elpais.com/cincodias/2016/04/06/smartphones/1459939294\\_104285.html](https://cincodias.elpais.com/cincodias/2016/04/06/smartphones/1459939294_104285.html)
20. Harrington, M., & Cross, M. (2015). GPS, GIS, and Google Earth. *Forensics*, 35-47. doi:<https://doi.org/10.1016/B978-0-12-800216-2.00003-9>
21. Instituto nacional de estadística y censos. (9 de Diciembre de 2016). *El parque automotor de Ecuador creció 57% en cinco años*. Obtenido de Instituto nacional de estadística y censos: <http://www.ecuadorencifras.gob.ec/el-parque-automotor-de-ecuador-crecio-57-en-cinco-anos/>
22. Joyanes Aguilar, L. (2015). *Sistemas de Información en la empresa. El impacto de la nube, la movilidad y los medios sociales* (Primera ed.). México: Alfaomega.

23. Leick, A., Rapoport, L., & Tatarnikov, D. (2015). *GPS Satellite Surveying*. John Wiley & Sons, Incorporated. Obtenido de <https://ebookcentral.proquest.com/lib/utn/detail.action?docID=1895254>
24. Lozano Equisoain, D. (2017). *Manual Imprescindible. Arduino Práctico*. Madrid: Anaya Multimedia.
25. Lynn, T., Morris, J., & Kenny, D. (2018). *Heterogeneity, High Performance Computing, Self-Organization and the Cloud*. Ireland: macmillan.
26. Miller, J. (08 de Agosto de 2018). Obtenido de MAKER PRO: <https://maker.pro/arduino/projects/build-a-car-tracking-system-with-the-sim808module>
27. Minoli, D. (2015). *Innovations in Satellite Communication and Satellite Technology* (Primera ed.). John Wiley & Sons, Incorporated. Obtenido de <https://ebookcentral.proquest.com/lib/utn/detail.action?docID=1895862>
28. *Módulo Celular GSM GPRS/GPS – SIM808 con Antenas GPS y GSM*. (2019). Obtenido de Electronilab: <https://electronilab.co/tienda/modulo-celular-gsm-gprs-gps-sim-808/>
29. *Módulo SIM808 con antena GSM Y GPS*. (2019). Obtenido de MACTRONICA ELECTRÓNICA Y TECNOLOGÍA: <https://www.mactronica.com.co/modulo-sim808-con-antena-gsm-gps-88598752xJM>
30. Murugesan, S., & Bojanova, I. (2016). *Encyclopedia on Cloud Computing* (Primera ed.). John Wiley & Sons, Incorporated. Obtenido de <https://ebookcentral.proquest.com/lib/utn/detail.action?docID=4526670>
31. Noticias ONU. (05 de Mayo de 2017). *El exceso de velocidad causa el 33% de las muertes de tránsito*. Obtenido de Noticias ONU: <https://news.un.org/es/story/2017/05/1378351>
32. Organización Mundial de la Salud. (2015). *Informe sobre la situación mundial de la seguridad vial 2015*. Obtenido de Organización Mundial de la Salud: [https://www.who.int/violence\\_injury\\_prevention/road\\_safety\\_status/2015/Summary\\_GSRRS2015\\_SPA.pdf?ua=1](https://www.who.int/violence_injury_prevention/road_safety_status/2015/Summary_GSRRS2015_SPA.pdf?ua=1)
33. Organización Mundial de la Salud. (2017). *Control de la Velocidad*. Obtenido de Organización Mundial de la Salud: <http://apps.who.int/iris/bitstream/handle/10665/255305/WHO-NMH-NVI-17.7-spa.pdf;jsessionid=0B1C4A6F4268DB77C1687800DF4357FD?sequence=1>
34. Ortiz, S. (8 de Junio de 2015). *Un promedio de 95 accidentes de tránsito ocurren cada día en el país*. Obtenido de El Comercio: <https://www.elcomercio.com/actualidad/accidenten-transito-choques-ecuador-velocidad.html>
35. Penttinen, J. (2015). *The Telecommunications Handbook* (Primera ed.). John Wiley & Sons, Incorporated. Obtenido de <https://ebookcentral.proquest.com/lib/utn/detail.action?docID=1921186>

36. Penttinen, J. (2016). *Wireless Communications Security* (Primera ed.). John Wiley & Sons, Incorporated. Obtenido de <https://ebookcentral.proquest.com/lib/utn/detail.action?docID=4690021#>
37. Pérez Rodríguez, D. (2015). *Arcview 3.2 y GPS* (Segunda ed.). (ICB, Ed.) España: ICB. Obtenido de <https://ebookcentral.proquest.com/lib/utnortesp/reader.action?docID=5757753&query=division%2BGPS#>
38. *Probots*. (2020). Obtenido de <https://probots.co.in/xl4015-step-down-buck-converter-adjustable-max-output-36v-5a.html>
39. PROMETEC. (2019). Obtenido de PROMETEC: <https://www.prometec.net/sim808/>
40. Puerchambud Ladino, C. E., & Henao Ariza, J. D. (2018). SISTEMA AUTÓNOMO PARA EL MANEJO DE UNA TORRE VERTICAL DE PARQUEADEROS. *Ingeniería, Matemáticas y Ciencias de la Información*, V(9), 41-54. doi:<http://dx.doi.org/10.21017/rimci.2018.v5.n9.a39>
41. Ravindran , A., & Prakash, E. (2016). *The ICT Age* (Primera ed.). Cambridge Scholars Publishing. Obtenido de <https://ebookcentral.proquest.com/lib/utn/detail.action?docID=4535300>
42. Reyes Cortés, F., & Cid Monjaraz, J. (2015). *Arduino. Aplicaciones en robótica, mecatrónica e ingenierías* (Primera ed.). México: Alfaomega.
43. Reyes Nuñez, J. J. (2018). Ideas y experiencias sobre el uso de servicios de mapas web en la educacion. *Boletim Paulista de Geografia*, 99, 86-101. Obtenido de <https://agb.org.br/publicacoes/index.php/boletim-paulista/article/view/1469/1341>
44. Secretaría Nacional de Planificación y Desarrollo (Senplades). (2012). *Transformación de la Matriz Productiva*. Obtenido de Secretaría Nacional de Planificación y Desarrollo: [http://www.planificacion.gob.ec/wp-content/uploads/downloads/2013/01/matriz\\_productiva\\_WEBtodo.pdf](http://www.planificacion.gob.ec/wp-content/uploads/downloads/2013/01/matriz_productiva_WEBtodo.pdf)
45. Secretaría Nacional de Planificación y Desarrollo (Senplades). (22 de Septiembre de 2017). *Plan Nacional de Desarrollo 2017-2021*. Obtenido de Secretaría Nacional de Planificación y Desarrollo: [http://www.planificacion.gob.ec/wp-content/uploads/downloads/2017/10/PNBV-26-OCT-FINAL\\_0K.compressed1.pdf](http://www.planificacion.gob.ec/wp-content/uploads/downloads/2017/10/PNBV-26-OCT-FINAL_0K.compressed1.pdf)
46. Somee. (2020). Obtenido de <https://somee.com/FreeAspNetHosting.aspx>
47. Somee. (2020). *Somee*. Obtenido de Somee: <https://somee.com>
48. Stackowiak, R., Licht, A., Mantha, V., & Nagode, L. (2015). *Big Data and the Internet of Things*. doi:<https://doi.org/10.1007/978-1-4842-0986-8>
49. Torrente Artero, Ó. (2016). *El mundo genuino-arduino Curso práctico de formación* (Primera ed.). México: Alfaomega.

50. Toulson, R., & Wilmshurst, T. (2017). *Fast and Effective Embedded Systems Design*. Obtenido de <https://doi.org/10.1016/B978-0-08-100880-5.00001-3>
51. Universidad de Cádiz. (2012). Unidad de Innovación Docente. 3. Obtenido de [http://www.uca.es/recursos/doc/Unidades/Unidad\\_Innovacion/Innovacion\\_Docente/A\\_NEXOS\\_2011\\_2012/22232441\\_310201212102.pdf](http://www.uca.es/recursos/doc/Unidades/Unidad_Innovacion/Innovacion_Docente/A_NEXOS_2011_2012/22232441_310201212102.pdf)
52. Vázquez Seisedos, C., Palacios Meléndez, E., Córdova Rivadeneira, L., & Romero Paz, M. (2016). Dispositivo de adquisición y transmisión de la posición de un vehículo mediante GPS y Wi-Fi. *Revista de Ingeniería Electrónica, Automática y Comunicaciones*, XXXVII, 15-27.
53. Watson, J. (2016). *Aerospace Navigation Systems* (Primera ed.).(A. Nebylov, Ed.) John Wiley & Sons, Incorporated. Obtenido de <https://ebookcentral.proquest.com/lib/utn/detail.action?docID=4532757>
54. Yeluri, R., & Castro Leon, E. (2014). *Building the Infrastructure for Cloud Security*. doi:<https://doi.org/10.1007/978-1-4302-6146-9>
55. Zúñiga Rocha, L., & González Camacho, Á. C. (2014). *Código Orgánico Integral Penal*. Obtenido de Ministerio de Justicia, derechos humanos y cultos: [https://www.justicia.gob.ec/wpcontent/uploads/2014/05/c%C3%B3digo\\_org%C3%A1nico\\_integral\\_penal\\_-\\_coip\\_ed.\\_sdn-mjdhc.pdf](https://www.justicia.gob.ec/wpcontent/uploads/2014/05/c%C3%B3digo_org%C3%A1nico_integral_penal_-_coip_ed._sdn-mjdhc.pdf)

## **ANEXOS**

## ANEXO 1

### PROGRAMACIÓN IDE DE ARDUINO

```
#include <DFRobot_sim808.h>

#include <SoftwareSerial.h>

char lat[12];

char lon[12];

char vel[12];

char DATOS[300];

//***** Designación de pines de transmision de datos *****

#define PIN_TX 7

#define PIN_RX 8

//***** Utilización Libreria *****

SoftwareSerial mySerial(PIN_TX,PIN_RX);

DFRobot_SIM808 sim808(&mySerial);//Connect RX,TX,PWR,

void setup() {

  mySerial.begin(9600);

  Serial.begin(9600);

//***** Enciende el módulo SIM 808 *****
```

```
while(!sim808.init()) {  
    delay(1000);  
    Serial.print("Sim808 error de Encendido");  
}  
  
//***** Enciende el GPS del módulo *****  
  
if( sim808.attachGPS())  
    Serial.println("Encendido de GPS exitoso");  
else  
    Serial.println("Falla en el encendido del GPS");  
}  
  
void loop() {  
  
//***** Obtiene los datos del GPS *****  
  
if (sim808.getGPS()) {  
  
    Serial.print(sim808.GPSdata.year);  
  
    Serial.print("/");  
  
    Serial.print(sim808.GPSdata.month);  
  
    Serial.print("/");  
  
    Serial.print(sim808.GPSdata.day);  
  
    Serial.print(" ");  
  
    Serial.print(sim808.GPSdata.hour);
```

```
Serial.print(":");  
  
Serial.print(sim808.GPSdata.minute);  
  
Serial.print(":");  
  
Serial.println(sim808.GPSdata.second);
```

```
float latitud = sim808.GPSdata.lat;  
  
float longitud = sim808.GPSdata.lon;  
  
float velocidad = sim808.GPSdata.speed_kph;
```

dtostrf(latitud, 6, 5, lat); //el valor float de latitud se convierte en char array, permitiendo 6 decimales antes de la coma y 5 decimales luego de la coma.

dtostrf(longitud, 6, 5, lon); //el valor float de longitud se convierte en char array, permitiendo 6 decimales antes de la coma y 5 decimales luego de la coma.

dtostrf(velocidad, 3, 0, vel); //el valor float de velocidad se convierte en char array, permitiendo 3 decimales antes de la coma y 0 decimales luego de la coma.

```
Serial.print("Latitud :");  
  
Serial.println(lat);  
  
Serial.print("Longitud :");  
  
Serial.println(lon);  
  
Serial.print("Velocidad_km/h :");  
  
Serial.println(vel);  
  
Serial.print("");  
  
Serial.println();
```

```

//***** Se crea la cadena de texto con el http y las variables a enviar
*****

sprintf(DATOS,
"AT+HTTTPARA=\"URL\", \"http://controlvelocidad.somee.com/InsertarDatos.aspx?&longitud=%s&l
atitud=%s&velocidad=%s&id=2\"", lon, lat, vel);

//***** Se utiliza comandos AT para configurar el GPRS y realizar una conexión
HTTP*****

mySerial.println("AT+CPIN?");

delay(500);

mySerial.println("AT+CSQ");

delay(500);

mySerial.println("AT+CREG?");

delay(500);

mySerial.println("AT+CGATT?");

Serial.println("Verificando datos móviles");

delay(500);

mySerial.println("AT+SAPBR=3,1,\"CONTYPE\", \"GPRS\");

delay(1000);

mySerial.println("AT+SAPBR=3,1,\"APN\", \"internet.claro.com.ec\");

delay(1000);

mySerial.println("AT+SAPBR=1,1");

delay(1000);

mySerial.println("AT+HTTTPINIT");

Serial.println("Iniciando conexión");

```

```
delay(1000);  
mySerial.println(DATOS);  
delay(1000);  
mySerial.println("AT+HTTPACTION=1");  
Serial.println("Enviando información");  
delay(8000);  
Serial.println("Información Enviada");  
Serial.println("");  
}  
}
```

## ANEXO 2

### DATOS PARAMETRIZACIÓN

	Lugares	latitud 1	longitud 1	latitud 2	longitud 2	latitud 3	longitud 3	latitud 4	longitud 4	limite vel
1	Norte	0,48164	-78,135471	0,479237	-78,103371	0,394954	-78,103542	0,395469	-78,136673	100
2	Aduana	0,391993	-78,113842	0,391907	-78,100366	0,379547	-78,100967	0,380406	-78,114271	50
3	<u>N. Yaguarco</u>	0,385813	-78,100624	0,384826	-78,084102	0,379419	-78,084445	0,379547	-78,100967	90
4	<u>Ent. Yaguarco</u>	0,380406	-78,114271	0,381307	-78,110623	0,363969	-78,110537	0,364141	-78,113799	90
5	<u>Cent. Yaguarco</u>	0,381307	-78,110623	0,379419	-78,084445	0,363884	-78,086076	0,363969	-78,110537	90
6	<u>Ciud. Yaguarco</u>	0,364484	-78,09629	0,363884	-78,086076	0,354099	-78,085303	0,35367	-78,096461	50
7	<u>Sur. Yaguarco</u>	0,364227	-78,106611	0,364484	-78,09629	0,35367	-78,096461	0,352269	-78,10644	90
8	Azaya	0,392465	-78,147466	0,391993	-78,113842	0,364141	-78,113799	0,365343	-78,146694	50
9	UTN	0,365343	-78,146694	0,364227	-78,106611	0,352296	-78,10644	0,354571	-78,146501	50
10	<u>Pilanci</u>	0,354571	-78,146501	0,352296	-78,10644	0,331182	-78,103586	0,332984	-78,146072	50
11	Esperanza	0,331611	-78,139206	0,331182	-78,103586	0,296078	-78,103672	0,296335	-78,138605	50
12	Corredor Sur	0,333843	-78,159719	0,331611	-78,139206	0,326461	-78,139206	0,328006	-78,159805	90
13	Inicio Corredor	0,35603	-78,159204	0,354571	-78,146501	0,332984	-78,146072	0,333843	-78,159719	90
14	Los Soles	0,352339	-78,192506	0,35603	-78,159204	0,336976	-78,159633	0,336461	-78,192592	100
15	San Antonio	0,336461	-78,192592	0,336976	-78,159633	0,328006	-78,159805	0,326548	-78,192678	50
16	<u>N. Atuntaqui</u>	0,340066	-78,210869	0,336461	-78,192592	0,331826	-78,192672	0,331998	-78,210354	90
17	<u>Atuntaqui</u>	0,340409	-78,239879	0,340066	-78,210869	0,323415	-78,193187	0,323243	-78,240909	50
18	<u>S. Atuntaqui</u>	0,323243	-78,240909	0,320668	-78,22031	0,303918	-78,220855	0,306592	-78,241253	100
19	<u>Est. AndradeMa.</u>	0,320668	-78,22031	0,320396	-78,193618	0,303783	-78,194114	0,303918	-78,220855	50
20	<u>Ihuman</u>	0,307006	-78,259452	0,303918	-78,220855	0,243823	-78,218274	0,245636	-78,259948	100
21	<u>Cotacachi</u>	0,307594	-78,275842	0,307006	-78,259452	0,290425	-78,260837	0,291291	-78,275842	50
22	Otavaló	0,244175	-78,279633	0,243823	-78,218274	0,218981	-78,23347	0,220525	-78,28073	50
23	<u>Ent. S. Otavaló</u>	0,220525	-78,28073	0,218981	-78,23347	0,196997	-78,23538	0,196997	-78,28073	90
24	Gonzales S.	0,196997	-78,23538	0,196265	-78,177594	0,133969	-78,180155	0,133969	-78,236233	90
25	<u>Vía Tabacun</u>	0,133969	-78,236233	0,133969	-78,180155	0,049852	-78,184604	0,048876	-78,237025	100
26	<u>Vía Cayambe</u>	0,133969	-78,180155	0,13275	-78,13584	0,049364	-78,138522	0,049852	-78,184604	100
27	Ayora	0,079598	-78,136084	0,079842	-78,12243	0,047901	-78,122674	0,049364	-78,138522	50
28	Cayambe	0,048145	-78,16266	0,047901	-78,122674	0,031565	-78,12755	0,034003	-78,160466	50
29	<u>Caya-Tabac</u>	0,049852	-78,184604	0,048145	-78,16266	0,034003	-78,160466	0,034491	-78,181435	90
30	<u>Y Tabacundo</u>	0,051802	-78,210388	0,049852	-78,184604	0,034491	-78,181435	0,034003	-78,206304	90
31	<u>Tabacundo</u>	0,051924	-78,231966	0,051802	-78,210388	0,034003	-78,206304	0,041806	-78,231479	50
32	S. Cayambe	0,034003	-78,160466	0,031565	-78,12755	0,026935	-78,131376	0,028608	-78,158018	50
33	<u>Cuzubamba</u>	0,03494	-78,347437	0,034003	-78,160466	-0,056574	-78,156046	-0,05538	-78,351022	100
34	<u>Ascazubi</u>	-0,054125	-78,299752	-0,054749	-78,280788	-0,093299	-78,281038	-0,09305	-78,298504	100
35	<u>Guayllafuera</u>	-0,053876	-78,364751	-0,05538	-78,351022	-0,060238	-78,348907	-0,060613	-78,364127	100
36	<u>Villacis</u>	-0,05538	-78,351022	-0,055248	-78,32882	-0,076707	-78,329444	-0,077455	-78,350653	50
37	<u>Cent. Guaylla</u>	-0,060987	-78,358264	-0,060238	-78,348907	-0,077455	-78,350653	-0,07758	-78,358139	50
38	<u>Puen. Guayll</u>	-0,059115	-78,406296	-0,060613	-78,364127	-0,106773	-78,363254	-0,103904	-78,408542	100
39	Quinche	-0,088858	-78,368293	-0,093299	-78,281038	-0,188978	-78,283644	-0,190746	-78,370061	90



```

namespace ASPGMaps
{
    public partial class _Default : Page
    {
        string connstring =
        ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
        protected void Page_Load(object sender, EventArgs e)
        {
            int id_cab_auto = Convert.ToInt16(Request.QueryString["id"]);

            if (Session["username"] != null)
            {
                if (!IsPostBack)
                {
                    double latitud = 0;
                    double longitud = 0;
                    int velocidad = 0;

                    string consulta = "select longitud_auto, latitud_auto, velocidad_reg from detalle_auto
                    where id_det_auto = (select max(id_det_auto) from detalle_auto where id_cab_auto = " + id_cab_auto
                    + ") and id_cab_auto = " + id_cab_auto;
                    SqlDataAdapter da = new SqlDataAdapter(consulta, connstring);

                    DataSet ds = new DataSet();
                    da.FillSchema(ds, SchemaType.Source, "detalle_auto");
                    da.Fill(ds, "detalle_auto");

                    DataTable detalle_auto;
                    detalle_auto = ds.Tables["detalle_auto"];

                    foreach (DataRow drCurrent in detalle_auto.Rows)
                    {
                        longitud = Convert.ToDouble(drCurrent["longitud_auto"].ToString());
                        latitud = Convert.ToDouble(drCurrent["latitud_auto"].ToString());
                        velocidad = Convert.ToInt16(drCurrent["velocidad_reg"].ToString());
                    }

                    string skey =
                    ConfigurationManager.AppSettings["googlemaps.subgurim.net"];
                    GLatLng mainLocation = new GLatLng(latitud, longitud);
                    GMap1.setCenter(mainLocation, 15);

                    XPinLetter xpinLetter = new XPinLetter(PinShapes.pin, "A", Color.Blue, Color.White,
                    Color.Chocolate);

                    //Pone la marca de gota de agua con el nombre de la ubicacion
                    GMarker marker = new GMarker(mainLocation, new GMarkerOptions(new
                    GIcon("../Images/Logo carro PNG FINAL.png", "../Images/Logo carro PNG FINAL.png")));
                    string strMarker = "Velocidad: " + Convert.ToString(velocidad) + " km / h";
                    GInfoWindow window = new GInfoWindow(marker, strMarker, true);
                }
            }
        }
    }
}

```

```

        GMap1.addInfoWindow(window);

    }
}
else
{
    Response.Redirect("~/Account/Login.aspx");
}
}

protected void BtnLocalizar_Click(object sender, EventArgs e)
{
    GLatLng mainLocation = new GLatLng(Convert.ToDouble(TxtLatitud.Text),
Convert.ToDouble(txtLongitud.Text));
    GMap1.setCenter(mainLocation, 15);

    XPinLetter xpinLetter = new XPinLetter(PinShapes.pin_star, "JC", Color.Blue, Color.White,
Color.Chocolate);
    GMap1.Add(new GMarker(mainLocation, new GMarkerOptions(new
GIcon(xpinLetter.ToString(), xpinLetter.Shadow()))));
}
}
}
}

```

- **Pantalla consulta histórica:**

**Diseño:**

```

<% @ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="Historico.aspx.cs" Inherits="ASPGMaps.Historico" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="FeaturedContent" runat="server">
    <h1>Reporte de Velocidad</h1>
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="MainContent" runat="server">
    <asp:Label ID="Label1" runat="server" Text="Desde: "></asp:Label>
<asp:TextBox ID="TxtDesde" runat="server" Font-Size="Medium"
TextMode="DateTimeLocal"></asp:TextBox>
    &nbsp;
<asp:Label ID="Label2" runat="server" Text="Hasta: "></asp:Label>
<asp:TextBox ID="TxtHasta" runat="server" Font-Size="Medium"
TextMode="DateTimeLocal"></asp:TextBox>
    &nbsp;&nbsp;&nbsp;&nbsp;<asp:Label ID="Label3" runat="server" Text="Vehiculo: "></asp:Label>
    &nbsp;
    <asp:DropDownList ID="DropDownList1" runat="server" DataSourceID="SqlDataSource1"
DataTextField="placa_auto" DataValueField="Id_cab_auto" Height="35px" Width="192px" Font-
Size="Medium" OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged">

```



**Proceso:**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace ASPGMaps
{
    public partial class Historico : System.Web.UI.Page
    {
        string connstring =
        ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["username"] == null)
            {
                Response.Redirect("~/Account/Login.aspx");
            }
            else
            {
                if (!IsPostBack)
                {
                    TxtDesde.Text = DateTime.Now.ToString("yyyy-MM-dd");
                    TxtHasta.Text = DateTime.Now.ToString("yyyy-MM-dd");
                }
            }
        }

        protected void BtnConsultar_Click(object sender, EventArgs e)
        {
            LLenarDatos();
        }

        private void LLenarDatos()
        {
            string consulta = "set dateformat ymd select top 1000 fec_registro, longitud_auto, latitud_auto,
            velocidad_reg from detalle_auto where id_cab_auto = " + DropDownList1.SelectedValue + " and
            fec_registro between replace('" + TxtDesde.Text + "', 'T', ' ') and replace('" + TxtHasta.Text + "', 'T', ' ')
            order by id_det_auto desc";
            SqlDataAdapter da = new SqlDataAdapter(consulta, connstring);

```

```

        DataTable dt = new DataTable();
        da.Fill(dt);
        GridView1.DataSource = (dt);
        GridView1.DataBind();
    }

    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {
        int fila = GridView1.SelectedIndex;
        string latitud = GridView1.Rows[fila].Cells[1].Text;
        string longitud = GridView1.Rows[fila].Cells[2].Text;
        string velocidad = GridView1.Rows[fila].Cells[3].Text;

        Response.Write("<script> window.open('" +
"http://controlvelocidad.somee.com/MapaHistorico.aspx?longitud=" + longitud + "&latitud=" + latitud
+ "&velocidad=" + velocidad + "','_blank'); </script>");
    }

    protected void GridView1_PageIndexChanging(object sender, GridViewPageEventArgs e)
    {
        //GridView gv = (GridView)sender;
        GridView1.PageIndex = e.NewPageIndex;
        GridView1.DataBind();
        LLenarDatos();
    }
}
}
}

```

- **Visualización del mapa en histórico:**

**Diseño:**

```

<% @ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="MapaHistorico.aspx.cs" Inherits="ASPGMaps.MapaHistorico" %>
<% @ Register assembly="GMaps" namespace="Subgurim.Controles" tagprefix="cc1" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="FeaturedContent" runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="MainContent" runat="server">
    <h3>Ubicación:</h3>
    <p>
        &nbsp;</p>
    <cc1:GMap ID="GMap1" runat="server" Width="800px" Height="500px"/>
    <br />
    <br />
</asp:Content>

```

**Proceso:**

```

using Subgurim.Controles;
using Subgurim.Controles.GoogleChartIconMaker;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace ASPGMaps
{
    public partial class MapaHistorico : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string skey = ConfigurationManager.AppSettings["googlemaps.subgurim.net"];
            double longitud = Convert.ToDouble(Request.QueryString["longitud"]);
            double latitud = Convert.ToDouble(Request.QueryString["latitud"]);
            string velocidad = Convert.ToString(Request.QueryString["velocidad"]);

            GLatLng mainLocation = new GLatLng(latitud, longitud);
            GMap1.setCenter(mainLocation, 15);

            XPinLetter xpinLetter = new XPinLetter(PinShapes.pin_star, "B", Color.Red, Color.White,
            Color.Chocolate);
            GMap1.Add(new GMarker(mainLocation, new GMarkerOptions(new
            GIcon(xpinLetter.ToString(), xpinLetter.Shadow()), "Velocidad " + velocidad + "km/h")));
        }
    }
}

```

- **Site.Master:**

**Diseño:**

```

<% @ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs"
Inherits="ASPGMaps.SiteMaster" %>

```

```

<!DOCTYPE html>
<html lang="en">

```

```

<head runat="server">
  <meta charset="utf-8" />
  <title><%: Page.Title %>- My ASP.NET Application</title>
  <link href="~/Content/Site.css" rel="stylesheet" />
  <link href="favicon.ico" rel="shortcut icon" type="image/x-icon" />
  <asp:PlaceHolder runat="server">
    <script src="<%: ResolveUrl("~/Scripts/modernizr-2.5.3.js") %>"></script>
  </asp:PlaceHolder>
  <meta name="viewport" content="width=device-width" />
  <asp:ContentPlaceHolder runat="server" ID="HeadContent" />
</head>
<body>
  <form runat="server">
    <asp:ScriptManager runat="server">
      <Scripts>
        <asp:ScriptReference Name="jquery" />
        <asp:ScriptReference Name="jquery.ui.combined" />
      </Scripts>
    </asp:ScriptManager>
    <header>
      <div class="content-wrapper">
        <div class="float-left">
          <p class="site-title"><a runat="server" href="~/Account/Login.aspx">Sistema de Control de
Velocidad Vehicular</a><asp:Image ID="Logo" runat="server" Height="94px"
ImageUrl="~/Images/UTN_logo.png" Width="144px" />
          </p>
        </div>
        <div class="float-right">
          <section id="login">
            <asp:LoginView runat="server" ViewStateMode="Disabled">
              <AnonymousTemplate>
                <ul>
                  <li><a id="registerLink" runat="server"
href="~/Account/Login.aspx">Salir</a></li>
                  <li><a id="loginLink" runat="server" href="~/Account/Login.aspx">Iniciar
Sesión</a></li>
                </ul>
              </AnonymousTemplate>
              <LoggedInTemplate>
                <p>
                  Hello, <a runat="server" class="username" href="~/Account/Manage.aspx"
title="Manage your account">
                    <asp:LoginName runat="server" CssClass="username" />
                  </a>!
                  <asp:LoginStatus runat="server" LogoutAction="Redirect" LogoutText="Log off"
LogoutPageUrl="~/Account/Logout.aspx" />
                </p>
              </LoggedInTemplate>
            </asp:LoginView>
          </section>
        </div>
      </div>
    </header>
  </form>
</body>

```

```

    <nav>
      <ul id="menu">
        <li><a runat="server" href="~/Selecciona_auto.aspx">Tiempo Real</a></li>
        <li><a runat="server" href="~/Contact.aspx">Contacto</a></li>
        <li><a runat="server" href="~/Historico.aspx">Consulta</a></li>
        <li><a runat="server" href="~/ExcesoVelocidad.aspx">Exceso de Velocidad</a></li>
      </ul>
    </nav>
  </div>
</div>
</header>
<div id="body">
  <asp:ContentPlaceHolder runat="server" ID="FeaturedContent" />
  <section class="content-wrapper main-content clear-fix">
    <asp:ContentPlaceHolder runat="server" ID="MainContent" />
  </section>
</div>
<footer>
  <div class="content-wrapper">
    <div class="float-left">
      <p>&copy; <%= DateTime.Now.Year %>- UTN - Jonathan Cadena - SISTEMA
ELECTRÓNICO DE SEGURIDAD PARA MONITOREO EN TIEMPO REAL DEL EXCESO DE
VELOCIDAD EN EL VEHÍCULO </p>
    </div>
  </div>
</footer>
</form>
</body>
</html>

```

**Proceso:**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;

```

```

namespace ASPGMaps
{
  public partial class SiteMaster : MasterPage
  {
    private const string AntiXsrfTokenKey = "__AntiXsrfToken";
    private const string AntiXsrfUserNameKey = "__AntiXsrfUserName";
    private string _antiXsrfTokenValue;
  }
}

```

```

protected void Page_Init(object sender, EventArgs e)
{
    // The code below helps to protect against XSRF attacks
    var requestCookie = Request.Cookies[AntiXsrfTokenKey];
    Guid requestCookieGuidValue;
    if (requestCookie != null && Guid.TryParse(requestCookie.Value, out
requestCookieGuidValue))
    {
        // Use the Anti-XSRF token from the cookie
        _antiXsrfTokenValue = requestCookie.Value;
        Page.ViewStateUserKey = _antiXsrfTokenValue;
    }
    else
    {
        // Generate a new Anti-XSRF token and save to the cookie
        _antiXsrfTokenValue = Guid.NewGuid().ToString("N");
        Page.ViewStateUserKey = _antiXsrfTokenValue;

        var responseCookie = new HttpCookie(AntiXsrfTokenKey)
        {
            HttpOnly = true,
            Value = _antiXsrfTokenValue
        };
        if (FormsAuthentication.RequireSSL && Request.IsSecureConnection)
        {
            responseCookie.Secure = true;
        }
        Response.Cookies.Set(responseCookie);
    }

    Page.PreLoad += master_Page_PreLoad;
}

protected void master_Page_PreLoad(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        // Set Anti-XSRF token
        ViewState[AntiXsrfTokenKey] = Page.ViewStateUserKey;
        ViewState[AntiXsrfUserNameKey] = Context.User.Identity.Name ?? String.Empty;
    }
    else
    {
        // Validate the Anti-XSRF token
        if ((string)ViewState[AntiXsrfTokenKey] != _antiXsrfTokenValue
            || (string)ViewState[AntiXsrfUserNameKey] != (Context.User.Identity.Name ??
String.Empty))
        {
            throw new InvalidOperationException("Validation of Anti-XSRF token failed.");
        }
    }
}

```

```

    }
  }
}

```

- **Web config:**

```

<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <configSections>
    <section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework,
Version=4.4.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
  </configSections>
  <appSettings>
    <add key="googlemaps.Subgurim.net" value="AIzaSyBwT8nqBBXziOI-jCIWBn3I_7aqG8T-
IzE"/>
  </appSettings>
  <connectionStrings>
    <add name="DefaultConnection" connectionString="Data Source=(LocalDb)\v11.0;Initial
Catalog=aspnet-ASPGMaps-20131105214218;Integrated
Security=SSPI;AttachDBFilename=|DataDirectory|\aspnet-ASPGMaps-20131105214218.mdf"
    providerName="System.Data.SqlClient" />
    <add name="MyDatabaseEntities"
connectionString="metadata=res://*/MyModel.csd|res://*/MyModel.ssd|res://*/MyModel.msl;provide
r=System.Data.SqlClient;provider connection string=&quot;Data
Source=(LocalDB)\v11.0;attachdbfilename=|DataDirectory|\MyDatabase.mdf;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;"
    providerName="System.Data.EntityClient" />
    <add name="MyDatabaseEntities1"
connectionString="metadata=res://*/MyModel1.csd|res://*/MyModel1.ssd|res://*/MyModel1.msl;pro
vider=System.Data.SqlClient;provider connection string=&quot;data
source=(LocalDB)\MSSQLLocalDB;attachdbfilename=|DataDirectory|\MyDatabase.mdf;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;"
    providerName="System.Data.EntityClient" />
    <add name="ConnectionString" connectionString="Data Source=DESKTOP-
3RC6TIT\SQLEXPRESS;Initial Catalog
=control;Trusted_Connection=true;MultipleActiveResultSets=True;Application
Name=EntityFramework"
    providerName="System.Data.SqlClient" />
  </connectionStrings>
  <system.web>
    <customErrors mode="Off" />
    <compilation debug="true" targetFramework="4.0" />
    <authentication mode="Forms">

```

```

    <forms loginUrl="~/Account/Login.aspx" timeout="2880" />
</authentication>
<profile defaultProvider="DefaultProfileProvider">
  <providers>
    <add name="DefaultProfileProvider" type="System.Web.Providers.DefaultProfileProvider,
System.Web.Providers, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
connectionStringName="DefaultConnection" applicationName="/" />
  </providers>
</profile>
<membership defaultProvider="DefaultMembershipProvider">
  <providers>
    <add name="DefaultMembershipProvider"
type="System.Web.Providers.DefaultMembershipProvider, System.Web.Providers, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" connectionStringName="DefaultConnection"
enablePasswordRetrieval="false" enablePasswordReset="true" requiresQuestionAndAnswer="false"
requiresUniqueEmail="false" maxInvalidPasswordAttempts="5" minRequiredPasswordLength="6"
minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="10" applicationName="/"
/>
  </providers>
</membership>
<roleManager defaultProvider="DefaultRoleProvider">
  <providers>
    <add name="DefaultRoleProvider" type="System.Web.Providers.DefaultRoleProvider,
System.Web.Providers, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
connectionStringName="DefaultConnection" applicationName="/" />
  </providers>
</roleManager>
<sessionState mode="InProc" customProvider="DefaultSessionProvider">
  <providers>
    <add name="DefaultSessionProvider" type="System.Web.Providers.DefaultSessionStateProvider,
System.Web.Providers, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
connectionStringName="DefaultConnection" />
  </providers>
</sessionState>
</system.web>
<system.webServer>
  <modules runAllManagedModulesForAllRequests="true" />
</system.webServer>
<entityFramework>
  <defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory,
EntityFramework" />
</entityFramework>
</configuration>

```