



Revista EIA  
ISSN: 1794-1237  
revista@eia.edu.co  
Escuela de Ingeniería de Antioquia  
Colombia

# SISTEMA DE GEOLOCALIZACIÓN DE VEHÍCULOS A TRAVÉS DE LA RED GSM/GPRS Y TECNOLOGÍA ARDUINO

Castro Correa, Jhon Alejandro; Sepúlveda Mora, Sergio Basilio; Medina Delgado, Byron; Guevara Ibarra, Dinael; López Bustamante, Oriana Alexandra  
SISTEMA DE GEOLOCALIZACIÓN DE VEHÍCULOS A TRAVÉS DE LA RED GSM/GPRS Y TECNOLOGÍA ARDUINO

Revista EIA, vol. 16, núm. 31, 2019

Escuela de Ingeniería de Antioquia, Colombia

**Disponible en:** <https://www.redalyc.org/articulo.oa?id=149258931011>

**DOI:** <https://doi.org/10.24050/reia.v16i31.1269>



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivar 4.0 Internacional.

# SISTEMA DE GEOLOCALIZACIÓN DE VEHÍCULOS A TRAVÉS DE LA RED GSM/GPRS Y TECNOLOGÍA ARDUINO

VEHICLE GEOLOCATION SYSTEM USING THE GSM/GPRS NETWORK AND ARDUINO TECHNOLOGY

SISTEMA DE GEOLOCALIZAÇÃO DE VEÍCULOS ATRAVÉS DA REDE DE TECNOLOGIA GSM / GPRS E ARDUINO

Jhon Alejandro Castro Correa [jhonalejandrocc@ufps.edu.co](mailto:jhonalejandrocc@ufps.edu.co)  
*Universidad Francisco de Paula Santander, Cúcuta-Colombia, Colombia*

Sergio Basilio Sepúlveda Mora [sergio.sepulveda@ufps.edu.co](mailto:sergio.sepulveda@ufps.edu.co)  
*Universidad Francisco de Paula Santander, Cúcuta-Colombia, Colombia*

Byron Medina Delgado [byronmedina@ufps.edu.co](mailto:byronmedina@ufps.edu.co)  
*Universidad Francisco de Paula Santander, Cúcuta-Colombia, Colombia*

Dinael Guevara Ibarra [dinaelgi@ufps.edu.co](mailto:dinaelgi@ufps.edu.co)  
*Universidad Francisco de Paula Santander, Cúcuta-Colombia, Colombia*

Oriana Alexandra López Bustamante  
[orianaalexandrallb@ufps.edu.co](mailto:orianaalexandrallb@ufps.edu.co)

*Universidad Francisco de Paula Santander, Cúcuta-Colombia, Colombia*

Revista EIA, vol. 16, núm. 31, 2019

Escuela de Ingeniería de Antioquia,  
Colombia

Recepción: 03 Octubre 2018  
Aprobación: 11 Noviembre 2018

DOI: <https://doi.org/10.24050/reia.v16i31.1269>

Redalyc: <https://www.redalyc.org/articulo.oa?id=149258931011>

**Resumen:** Para localizar y guardar datos del posicionamiento geográfico de vehículos en tiempo real, se diseñó un sistema de geolocalización compuesto de dos elementos: un prototipo para el envío de coordenadas usando tecnología GPRS y un servidor web para la recepción y almacenamiento de la información recibida desde un GPS. Se utilizó el sistema embebido Raspberry Pi 3, configurado como servidor web HTTP bajo Apache 2.4, integrando PHP y MySQL. Se definió una IP estática local por medio de DHCP para establecer la comunicación del servidor con Internet. El hardware utilizado para enviar los datos corresponde a una tarjeta Arduino UNO; un módulo GPRS, programado mediante comandos AT y un GPS, cuya trama de datos \$GPRMC corresponde al estándar NMEA. Los resultados obtenidos muestran una diferencia promedio de 6.50 m entre los datos de ubicación geográfica del dispositivo frente a los valores obtenidos mediante la geolocalización por HTML5; asimismo, se confirmó que el consumo diario del sistema es de 367.97 kB. Finalmente, a través de la puesta en marcha del prototipo se comprobó la precisión del dispositivo en comparación a otras alternativas de geolocalización, así como la viabilidad del uso de datos por medio de SIM card a través del protocolo GPRS.

**Palabras clave:** Arduino, GPS, Servidor Web, Raspberry Pi, GPRS.

**Abstract:** We designed a system to locate and store data of the geographical position of vehicles in real time. The system consists of two elements: a prototype for sending coordinates using the GSM/GPRS network and a web server for the reception and storage of the information coming from a GPS. We used the Raspberry Pi 3 embedded system as an HTTP web server running Apache 2.4 software as well as PHP and MySQL. A local static IP was established through DHCP and consequently the communication with any computer connected to the Internet was possible. The transmitter hardware

includes an Arduino UNO card, a GPRS module, and a GPS module. The data frame \$GPRMC used for the GPS complies with NMEA standard. Our results show an average difference of 6,50 m between the geographical location of the device compared to the real location point obtained from HTML5 geolocation; likewise, the daily consumption of the geolocation system is 367,97 kB. Finally, the accuracy of the proposed prototype was comparable to other geolocation alternatives, as well as the feasibility of using a SIM card with the GPRS protocol to transmit the data.

**Keywords:** Arduino, GPS, Web Service, Raspberry Pi, GPRS.

**Resumo:** Para localizar e salvar os dados em tempo real do posicionamento geográfico de veículos, um sistema de geolocalização composto de dois elementos foi projetado: um protótipo para o transporte de coordenadas utilizando a tecnologia GPRS e um servidor web para a recepção e armazenamento da informação recebida a partir de um GPS. Foi utilizado o sistema embarcado Raspberry Pi 3, configurado como um servidor web HTTP através de Apache 2.4 e também integrando PHP e MySQL. Foi definido um IP estático local, por meio de DHCP para configurar a comunicação do servidor com a Internet. O hardware utilizado para enviar os dados corresponde a uma placa Arduino UNO; um módulo GPRS, programado através de comandos AT e um GPS, cujo quadro de dados \$GPRMC corresponde ao padrão NMEA. Os resultados obtidos mostram uma diferença média de 6,33 m entre os dados de localização geográfica do dispositivo em comparação com os valores obtidos através da geolocalização por HTML5; além disso, foi confirmado que o consumo diário do sistema é de 367,97 kB. Finalmente, através da implementação do protótipo foi verificada a precisão do aparelho em comparação a outras alternativas de geolocalização, assim como a viabilidade do uso de dados por meio de SIM card através do protocolo GPRS

**Palavras-chave:** Arduino, GPS, Servidor Web, Raspberry Pi, GPRS.

## 1. INTRODUCCIÓN

A través de los años, el desarrollo de nuevas tecnologías hizo posible la aparición de un nuevo concepto denominado como el Internet de las Cosas o IoT (Internet of Things) que hace referencia a la interconexión de los objetos del mundo físico a través de Internet y que están equipados con sensores, actuadores y tecnología de comunicación (Bonilla Fabela, et al., 2016); según Cisco Internet Business Solutions Group (IBSG) el Internet de las Cosas surgió entre los años 2008 y 2009, tiempo en el cual habían más cosas conectadas a Internet que personas (Cisco Internet Business Solutions Group, 2018). Con el pasar de los años esta idea se ha hecho más popular, siendo utilizada en la industria de producción en masa, control de infraestructura urbana, control ambiental, seguridad y el sector salud (Sanmartín Mendoza, et al., 2016).

La clave del IoT yace en la operación remota (Zhang, et al., 2012), pues cada uno de los dispositivos conectados a Internet posee una IP única que permite, mediante diversos protocolos de comunicación, ser accedido para recibir instrucciones e información, con el fin de ser almacenada o realizar alguna tarea específica (Alcaraz, 2014). Con el pasar de los años se han desarrollado diversos trabajos cuyo propósito es solucionar problemáticas, utilizando tecnologías relacionadas al Internet de las Cosas, orientados al rastreo, posicionamiento, supervisión de vías de transporte o servicios de emergencia como es presentado en la propuesta de Zapata Murillo & Baldoquin (2018).

En su trabajo, Escobar Amado, et al. (2016), hicieron uso del Internet de las Cosas para la supervisión de un proceso mediante una aplicación

móvil utilizando como servidor un sistema embebido Raspberry Pi, en el cual se implementó un servidor DHCP con el fin de monitorizar una intersección semafórica en la ciudad de Cúcuta.

Asimismo, en India, Singla & Bhatia (2015) propusieron un sistema de rastreo para buses basado en GPS utilizando una aplicación Android para visualizar la ruta del vehículo así como las posiciones del usuario y el bus en donde se instaló el sistema.

Por otra parte, en Sri Lanka, Haleem SL & Samsudeen (2016) desarrollaron un sistema en tiempo real para el rastreo de buses e itinerario utilizando sensores inalámbricos y tecnología móvil, obteniendo alta precisión de identificación a un bajo costo.

Además, Morales Valenzuela, et al. (2015) implementaron un sistema de monitorización para pacientes de alto riesgo utilizando módulos GPS, GSM/GPRS y Zigbee, comprobando que la tecnología utilizada es idónea debido a su robustez, rapidez y de fácil interpretación para el receptor.

Usando tecnologías GPS/GSM/GPRS, Lee, et al. (2014) desarrollaron un sistema de rastreo de vehículos mediante una aplicación móvil, demostrando la efectividad del dispositivo con respecto a la localización del vehículo, así como un diseño accesible y de bajo costo.

Incluso el uso del IoT es extendido a sectores agrarios. En este campo Lomillos-Pérez, et al (2016) realizaron un estudio de reses mediante el uso de tecnología GPS-GPRS, evidenciando la eficacia del método para el estudio de su comportamiento en el pastoreo y la obtención de datos del vínculo diferenciado y permanente entre animales emparentados.

Del mismo modo, el internet de las cosas puede ser aplicado a la seguridad y las ciencias forenses, Conti, et al. (2018) destacan un método potencialmente prometedor para solucionar problemas de seguridad y análisis forense en entornos de IoT.

En su investigación, Liu (2017) llevó a cabo la optimización de un sistema de transmisión inalámbrico basado en GPRS, obteniendo un sistema con gran expansibilidad y que puede ser usado en control industrial y en el campo de seguridad.

Otro campo del IoT está encaminado a la localización de personas extraviadas; Padilla, et al. (2015) desarrollaron un sistema basado en Arduino y GSM/GPRS para la monitorización de personas seguro y de bajo costo.

Finalmente, Satyanarayana, et al. (2012) desarrollaron un sistema para la tele-monitorización del transporte de emergencia de pacientes, utilizando tecnología GPS y GPRS. En su investigación se realizó un sistema médico integrado de adquisición de datos con un bajo costo, instalado en ambulancias encargadas para el transporte de pacientes.

Los sistemas de información geográfica tienen aplicaciones tales como: descripción del terreno en función de coordenadas espaciales, caracterización de asentamientos humanos, definición de uso de suelo, diseño de mapas de riesgo específico, y muchas otras (Sosa Pedroza & Martínez Zúñiga, 2009). Por consiguiente, es común la aplicación de sistemas cuyo propósito principal es la localización y rastreo. La implementación de dichos dispositivos y procesos hace imprescindible

conocer, en tiempo real, datos como: longitud, latitud, velocidad y curso, que son elementos necesarios para establecer la ubicación y trayectoria de un objeto; asimismo, de poderse aplicar e integrar con software que permita manipular, tratar y mostrar los datos obtenidos. Este trabajo propone la realización de un sistema de posicionamiento utilizando la tarjeta Arduino UNO la cual obtiene y envía la información de la ubicación a través del protocolo TCP/IP empleando un módulo GPRS SIM900 a un servidor web operando en el sistema embebido Raspberry Pi 3.

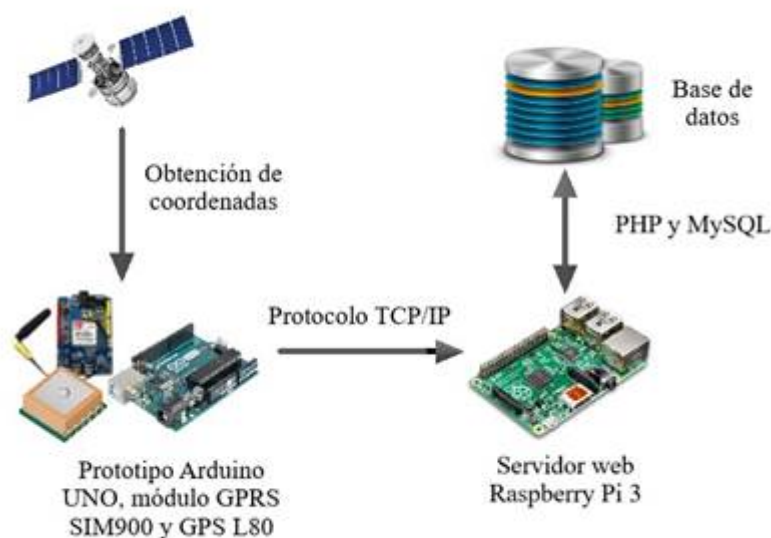
El artículo se encuentra estructurado de la siguiente manera, comenzando con los materiales y métodos para el desarrollo del prototipo, cuya función es captar y enviar las coordenadas por medio de Internet hacia el servidor; luego se presentan los resultados obtenidos una vez implementado el prototipo; finalmente las conclusiones y recomendaciones para futuras implementaciones de sistemas del mismo tipo.

## 2. MATERIALES Y MÉTODOS

El sistema ha sido diseñado para la recepción y almacenamiento de las variables de posicionamiento en tiempo real, utilizando el estándar TCP/IP (Transmission Control Protocol/Internet Protocol) para el envío de las variables desde el dispositivo a la base de datos alojada en el servidor.

Los dispositivos que hacen posible el funcionamiento del sistema se enuncian y explican a continuación: primeramente, el proceso de obtención de datos se realiza por medio de un módulo GPS L80, cuyo propósito es obtener la información sobre la ubicación del vehículo, por medio de la triangulación con los satélites a los cuales se conecta el dispositivo; una tarjeta Arduino UNO, encargada de tratar la información proporcionada por el GPS para transmitirla a través de Internet y un módulo GPRS SIM900, que a través de datos móviles e instrucciones programadas mediante comandos AT es usado para enviar las coordenadas geográficas a través del protocolo TCP/IP por medio de una petición GET utilizando HTTP.

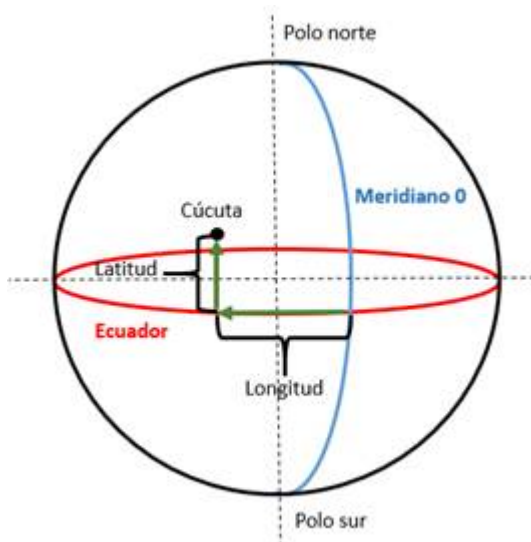
Por su parte, la información transmitida se recibe en un servidor web, el cual funciona bajo Apache 2.4, que es un servidor web libre desarrollado por Apache Server Project (Proyecto Servidor Apache) cuyo objetivo es la creación de un servidor web fiable, eficiente y fácilmente extensible con código de fuente abierto (Márquez Díaz, et al., 2002). Asimismo, se utilizó el sistema Raspberry Pi 3 como host, éste aloja una base de datos y un aplicativo en PHP encargado de la inserción de los valores en la base de datos. En la Figura 1 se muestra la arquitectura del sistema de posicionamiento previamente mencionado, donde se detalla cada uno de los componentes que hacen parte de la arquitectura del sistema y el papel que desempeña cada uno para llevar a cabo las tareas de obtención, tratamiento, envío y almacenamiento de datos de posición y movimiento.



**Figura 1.**

Esquema del sistema de posicionamiento implementado.

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, tal como se detalla en la Figura 2. A su vez, para garantizar la monitorización en tiempo real del movimiento, posición y trayectoria son necesarios como mínimo el curso, la velocidad, la fecha y la hora del dispositivo en cada lectura realizada.



**Figura 2**

Latitud y longitud de un punto ubicado en el globo terrestre.

La obtención de los datos de posicionamiento se realizó por medio del GPS L80 haciendo uso del estándar NMEA, que determina la señal eléctrica, el protocolo y la hora de transmisión de la información para un



bus de datos en serie de 4.800 baudios; a su vez, la secuencia que se usó para capturar los datos fue la \$GPRMC, que es la mínima recomendada para la correcta localización (QUECTEL, 2014); un ejemplo de esta se presenta a continuación:

\$GPRMC,195726.000,A,3324.9257,S,07036.3802,W,1.08,183.58,151210,\*,0E

El valor “195726.000” representa la hora del dispositivo GPS, el tiempo en UTC; el valor “A” representa la intensidad de la señal de los satélites (A activo, V de otro modo); la latitud es el cuarto valor y representa la distancia actual al norte o al sur del Ecuador; el siguiente dato hace referencia a la posición respecto a la línea del Ecuador donde “S” corresponde al sur y “N” al norte; el sexto valor representa la distancia actual al Este o al Oeste del Meridiano de Greenwich; el séptimo carácter, la longitud, contiene hacia qué lado del meridiano de Greenwich se midió esa distancia, siendo “E” el este y “W” el oeste; asimismo la cadena “1.08” indica la velocidad actual (en Nudos); el valor “183.58” indica la dirección actual en la que se mueve el dispositivo, medido como un “azimut”; el penúltimo valor se refiere a la fecha y hora del dispositivo, calculada a partir de las señales de los satélites GPS; por último, el valor “0E” es la suma de control que se utiliza para identificar los errores en los datos producidos durante la transmisión.

La tarjeta Arduino UNO fue usada para procesar los datos del GPS, debido a que la información correspondiente a las coordenadas no es recibida de forma directa sino en formato \$GPRMC, lo que hace necesario decodificar cada uno de estos datos y mostrarlos de forma clara y eficiente para que puedan ser leídos con normalidad y manipulados con facilidad; para este caso, se utilizó la librería TinyGPS++ desarrollada para el sistema Arduino la cual permite la obtención de los datos desde el GPS de forma sencilla y óptima, brindando la posibilidad de adquirir los valores por medio de objetos y métodos declarados dentro de la librería.

La segunda función de la tarjeta Arduino es la de configurar el módulo GPRS SIM900 para enviar los datos a través del protocolo TCP/IP mediante una petición GET al servidor. Este módulo funciona por medio de comandos AT, que permiten controlar el encendido, apertura del canal, asignación de IP y APN, configuración de datos, protocolo, método de envío, transmisión de información y finalmente el cierre del canal.

Tanto el módulo GPS como el GPRS funcionan por medio de transmisión serial a una tasa de 19.200 Baudios; sin embargo, la tarjeta Arduino cuenta con solo un par de pines seriales (TX y RX, digitales 0 y 1), esto se remedió mediante la librería SoftwareSerial desarrollada por Arduino, la cual permite la configuración de cualquier par de pines digitales de la tarjeta como puerto serial; de este modo se habilitó la utilización de ambos dispositivos al mismo tiempo.

El algoritmo que describe de forma general el funcionamiento del programa en Arduino se presenta en la Figura 3, donde se realiza el encendido del GPRS por medio de pulsos digitales, luego de verificar si éste se encuentra apagado, para proseguir con la asignación de IP y APN para luego obtener los datos del GPS y enviarlos por medio de una petición GET a través de Internet.

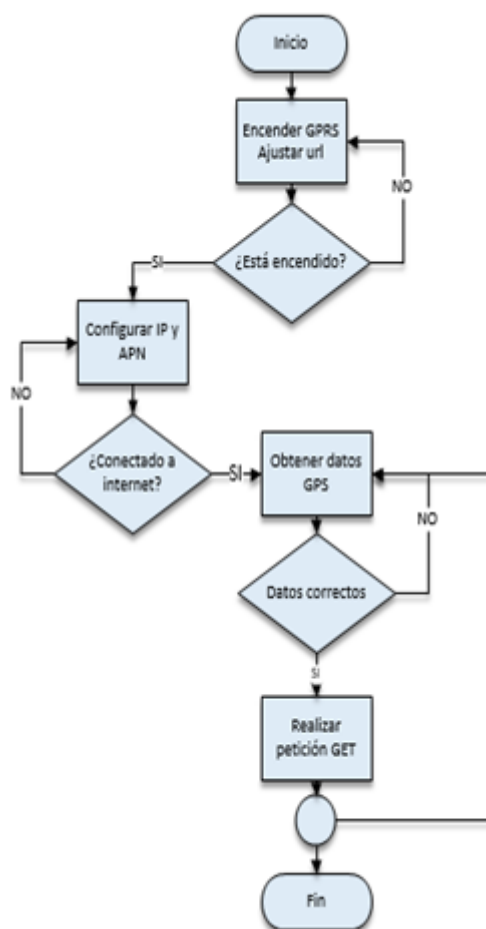


Figura 3

Diagrama de flujo del algoritmo desarrollado para el envío de datos mediante GPRS.

Como se mencionó con anterioridad, el servidor utilizado para la gestión de la información fue Apache 2.4 junto con PHP. Se diseñó una base de datos con funcionalidad en tiempo real generada a través de un servicio desarrollado en PHP, el cual permite la consulta de los registros en formato JSON (JavaScript Object Notation). JSON es un formato compacto basado en ASCII y compuesto por objetos los cuales son conformados por los datos y el valor (Silva Pinto & Silva Centeno, 2012).

Para la obtención de los valores almacenados en el servidor, a través de un servicio web, se desarrollaron scripts en PHP para consulta de información teniendo en cuenta filtros como: los últimos datos consultados o identificadores del dispositivo, todo esto por medio del puerto 80 y peticiones realizadas por los métodos GET y POST.

En cuanto a la base de datos, se utilizó el gestor MySQL debido al respaldo dentro de la comunidad de desarrolladores, su compatibilidad con Apache, su robustez, y su capacidad para permitir un número indefinido de registros y su facilidad e integración con PHP para el manejo de variables en cuanto a la consulta e inserción de datos en las tablas.

La base de datos realizada para la recepción de los valores de posición se denominó GPSService que contiene la tabla GPSTracker (Tabla 1),



la cual se encarga de almacenar los valores obtenidos desde el GPS. A continuación se describen los campos definidos dentro de la tabla.

- Id: Identificador de la trama de datos recibida (llave primaria auto-incremental).
- Tag: Valor predeterminado para identificar el dispositivo desde el cual se están enviando los datos.
- Lng: Dato correspondiente a la longitud geográfica del dispositivo (en grados).
- Lat: Valor referente a la latitud geográfica del dispositivo (en grados).
- Speed: Rapidez a la cual se está moviendo el vehículo (km/h).
- Date: Fecha en el cual se envió el dato.
- Time: Hora a la cual se generó el envío.
- Card: Rumbo del vehículo (Sur – Norte – Este – Oeste).

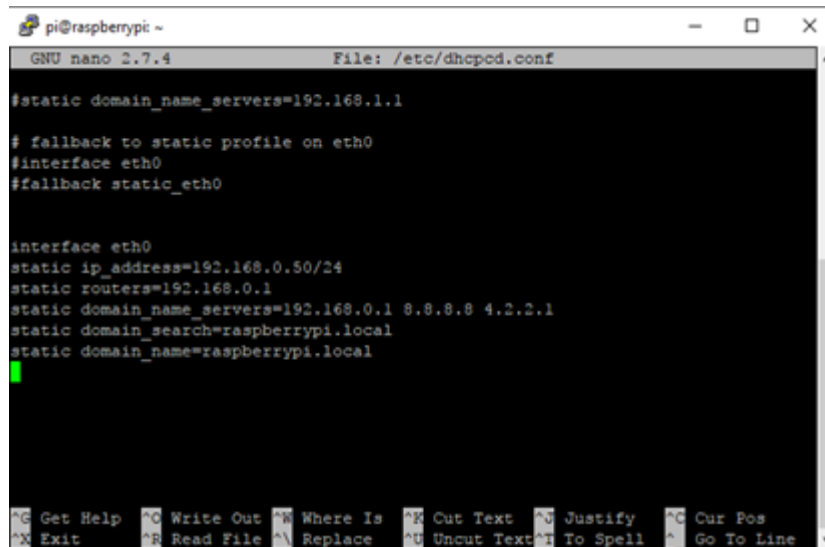
**Tabla 1**

Representación de los datos en la tabla GPSTracker.

| Id | Tag | Lng [°]  | Lat [°] | Speed [km/h] | Date       | Time     | Card |
|----|-----|----------|---------|--------------|------------|----------|------|
| 1  | 1   | -72,8443 | 7,9243  | 36,4         | 2017-10-21 | 17:08:04 | ENE  |
| 2  | 3   | -72,8532 | 7,9145  | 30,3         | 2017-11-12 | 12:05:15 | N    |
| 3  | 3   | -72,8656 | 7,9327  | 28,1         | 2017-11-12 | 12:07:32 | N    |
| 4  | 4   | -72,8289 | 7,9524  | 10,5         | 2017-08-22 | 16:46:59 | NE   |
| 5  | 2   | -72,8249 | 7,9352  | 41,3         | 2017-11-02 | 09:14:37 | S    |
| 6  | 2   | -72,8560 | 7,9356  | 21,4         | 2017-11-02 | 09:20:32 | SE   |
| 7  | 4   | -72,8023 | 7,9234  | 40,1         | 2017-12-03 | 18:09:26 | NE   |
| 8  | 3   | -72,8639 | 7,9225  | 35,3         | 2017-11-12 | 13:41:45 | NE   |
| 9  | 2   | -72,8458 | 7,9435  | 30,2         | 2017-11-02 | 10:20:48 | S    |
| 10 | 4   | -72,8726 | 7,9583  | 27,7         | 2017-12-03 | 18:20:14 | SW   |

Tabla 1 Representación de los datos en la tabla GPSTracker.

El sistema Raspberry Pi es accesible remotamente mediante la IP local, sin embargo durante cada reinicio de éste, la dirección IP cambiaba de forma dinámica, haciendo necesario configurarla manualmente para que siempre se asignara la misma por el enrutador; esta opción se modificó en el directorio “/etc/dhcpd.conf” encontrado en los archivos de configuración de Apache. La IP estática se estableció al puerto Ethernet de la Raspberry como se muestra en la Figura 4, en la cual se eligió la dirección IP “192.168.0.50” que perduraría durante cada reinicio de la tarjeta.



```
pi@raspberrypi: ~  
GNU nano 2.7.4 File: /etc/dhcpd.conf  
#static domain_name_servers=192.168.1.1  
  
# fallback to static profile on eth0  
#interface eth0  
#fallback static_eth0  
  
interface eth0  
static ip_address=192.168.0.50/24  
static routers=192.168.0.1  
static domain_name_servers=192.168.0.1 8.8.8.8 4.2.2.1  
static domain_search=raspberrypi.local  
static domain_name=raspberrypi.local  
  
Get Help Write Out Where Is Cut Text Justify Cur Pos  
Exit Read File Replace Uncut Text To Spell Go To Line
```

**Figura 4**  
Configuración de IP estática por medio de DHCP

En la Figura 4 se observan los diferentes ajustes que se realizaron para la configuración de la IP:

- Interface eth0: Hace referencia al puerto Ethernet de la Raspberry que se configuró.
- Static ip\_address: Dirección IPv4 que se estableció. Además cabe resaltar que al final de esta se encuentra el sufijo “/24” la cual es una abreviatura de la máscara subred 255.255.255.0.
- Static Routers: Es la dirección IP del Router.
- Static domain\_name\_servers: Es la dirección IP del Router y DNS.
- Static domain\_search y Static domain\_name: Nombre con el que se encontrará a la Raspberry en la red.

### 3. RESULTADOS Y DISCUSIÓN

Se realizaron pruebas constantes para verificar el envío de datos desde el Arduino hacia el servidor en la Raspberry Pi con un intervalo entre cada petición de 15 segundos. El tiempo entre cada uno de los envíos de datos fue configurado dentro del algoritmo desarrollado en Arduino.

Respecto al servidor, se obtuvo la información de la posición del prototipo en formato JSON a partir de la tabla GPSTracker, que hizo más sencilla la consulta de los datos a través de PHP para después mostrarlos en un sitio web. El vector JSON está compuesto por los valores descritos en la Tabla 1; y es compuesto por el nombre, la clave y el número, el valor del array.

El prototipo desarrollado fue expuesto a diversas pruebas de funcionamiento así como a un estudio en el cual se comparó la información obtenida desde el GPS con datos adquiridos a través de la función de geolocalización de HTML5. Un ejemplo de esto puede verse reflejado en la Figura 5, donde se observan los datos geográficos suministrados por el navegador.



Figura 5.

Datos generados mediante la geolocalización en HTML5. Fuente: Google Maps.

En la Tabla 2 se muestran algunas de las pruebas de precisión realizadas con el prototipo. Las primeras columnas corresponden a los datos obtenidos desde el GPS, mientras que las siguientes hacen referencia a las coordenadas adquiridas mediante el navegador web. Los valores de longitud (-72,518055) y latitud (7,911825) presentados en la Figura 5, junto a los valores obtenidos mediante pruebas de funcionamiento se observan en la primera fila de la Tabla 2.

Tabla 2.

Comparación de los datos de longitud y latitud provistos por el GPS y HTML5.

| Longitud<br>GPS [°] | Latitud<br>GPS [°] | Longitud<br>HTML5 [°] | Latitud<br>HTML5 [°] | Diferencia |
|---------------------|--------------------|-----------------------|----------------------|------------|
| -72,518070          | 7,911870           | -72,518055            | 7,911825             | 5,267 m    |
| -72,528513          | 7,915411           | -72,528547            | 7,915368             | 6,070 m    |
| -72,509673          | 7,913849           | -72,509713            | 7,913787             | 8,177 m    |
| -72,513142          | 7,903726           | -72,513074            | 7,903695             | 8,240 m    |
| -72,508684          | 7,897233           | -72,508661            | 7,897195             | 4,924 m    |
| -72,524873          | 7,908801           | -72,524826            | 7,908770             | 6,216 m    |
| -72,528238          | 7,911624           | -72,528254            | 7,911579             | 5,302 m    |
| -72,528183          | 7,912281           | -72,528196            | 7,912216             | 7,364 m    |
| -72,500333          | 7,860016           | -72,500368            | 7,859970             | 6,402 m    |
| -72,505051          | 7,848194           | -72,505067            | 7,848144             | 5,829 m    |
| -72,510714          | 7,830805           | -72,510676            | 7,830755             | 6,956 m    |
| -72,517239          | 7,808485           | -72,517194            | 7,808437             | 7,280 m    |
| Promedio            |                    |                       |                      | 6,50 m     |

Tabla 2 Comparación de los datos de longitud y latitud provistos por el GPS y HTML5

Para el cálculo de la distancia entre dos puntos sobre la superficie terrestre, obtenidos mediante los valores de longitud y latitud (ver Tabla

2), se utilizó como primera opción la Ecuación (1), que corresponde a la fórmula de la ley de cosenos para trigonometría esférica (Antúnez Romanuel & Hernández Montero, 2014), en la cual se emplea el radio terrestre en kilómetros y los valores de latitud y longitud de los dos puntos geográficos en radianes.

$$\begin{aligned}a &= \text{sen}(\varphi_1)\text{sen}(\varphi_2) \\ b &= \text{cos}(\varphi_1)\text{cos}(\varphi_2) \\ d &= r * \text{arccos}[a + b\text{cos}(\lambda_1 - \lambda_2)]\end{aligned}$$

[Ecuación 1]

d

Donde:

$\varphi_1, \varphi_2$  → Latitudes de los puntos 1 y 2 en radianes.

$\lambda_1, \lambda_2$  → Longitudes de los puntos 1 y 2 en radianes.

$d$  → Distancia entre dos puntos geográficos relacionados en el plano esférico (ver Figura 2).

$r$  → Radio de la tierra en km (6.367,4 km)

Sin embargo, aunque la Ecuación (1) es matemáticamente exacta, es poco fiable para el cálculo de distancias pequeñas, debido a que el arco coseno puede introducir errores considerables. Para solventar lo anterior, se realizó la transformación de la Ecuación (1) a otra expresión que permitiera utilizarse de forma general; a la expresión resultante se le conoce como la fórmula de Haversine (Alam, et al., 2016), la cual proporciona resultados exactos para cualquier distancia; y es calculable mediante propiedades trigonométricas.

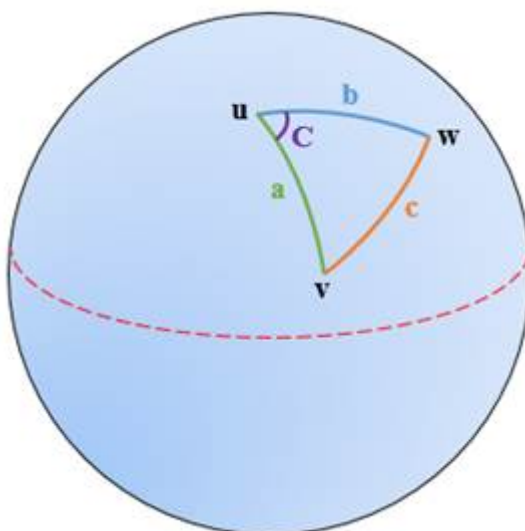
En la Figura 6 se presenta la ley de estados Haversinos, en donde se considera un triángulo en la superficie esférica con vértices u, v y w; mientras que a, b y c corresponden a las distancias a ser calculadas con la aplicación de las leyes de Haversine. Para el caso de la Tierra, las longitudes de a, b y c son iguales a su ángulo central multiplicado por el radio terrestre. La Ecuación (2) muestra la relación entre las distancias y ángulo delimitados por el triángulo u, v, w.

$$\text{harv}(c) = \text{harv}(a-b) + \sin(a)\sin(b)\text{harv}(C)$$

[Ecuación 2]

Donde:  $\text{harv}$  → Función Haversine.

$$\text{harv}(\theta) = \text{sen}^2\left(\frac{\theta}{2}\right)$$



**Figura 6**

Triángulo esférico solucionado con las leyes de Haversine.

A su vez, la Ecuación (2) no es completamente aplicable para obtener la distancia entre dos puntos geográficos. Para obtener la fórmula de Haversine, utilizada para tal fin, se considera el caso especial donde u es el polo norte, mientras que v y w son los puntos cuya separación d es determinada. Esto es: a y b son  $\pi/2 - \varphi$ , C es igual a  $\Delta\lambda$  y c es la distancia deseada ( $d/r$ ). Adicionalmente se utilizó la Ecuación (3), que corresponde a una diferente representación de la ley de cosenos para trigonometría esférica (Obuhuma, et al., 2018).

$$\cos(c) = \cos(a) \cos(b) + \sin(a) \sin(b) \cos(C) \quad [\text{Ecuación 3}]$$

Donde:  $\cos(\theta) = 1 - 2\text{havr}(\theta)$

Desarrollando la Ecuación (2) para las condiciones descritas y haciendo uso de (3), se deriva finalmente la ley de Haversine presentada en la Ecuación (4).

$$d = 2r \cdot \arcsin(\sqrt{\text{havr}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{havr}(\lambda_2 - \lambda_1)}) \quad \text{Ecuación (4)}$$

Utilizando la Ecuación (4) se estimó la distancia existente entre los puntos proporcionados por el GPS y los obtenidos mediante la localización por HTML5. Como se observa en la Tabla 2, se presentó una diferencia promedio de 6,50 m con respecto al servicio de geolocalización que ofrecen los navegadores web, probando así la eficacia del sistema de posicionamiento ante la alternativa de geolocalización ofrecida a través de HTTPS por la API de HTML5.

Por último, el consumo de datos por parte del sistema de geolocalización, se calculó de la siguiente forma: se tuvo en cuenta un funcionamiento al día de 10 horas diarias con un intervalo de 15 segundos

entre el envío de cada dato, es decir, 4 por minuto. Lo anterior se desarrolló como se muestra a continuación:

$$10h * \frac{60min}{1h} * \frac{4 envios}{1min} = 2.400 envios/día$$

La trama de datos enviada hacia el servidor desde el prototipo posee una longitud promedio de 157 caracteres, es decir 157 Bytes. Para calcular la cantidad de información enviada al día se multiplicó el valor de peticiones diarias por los Bytes correspondientes a la información, esto es:

$$157Bytes * 2.400 = 367,96875 kB$$

De este modo, el sistema presenta un consumo de datos de 367,96875 kB por día o de 10,78 MB por mes. Para efectos de comparación, de acuerdo con Cisco, el tráfico promedio mensual de un usuario móvil en el 2016 fue de 977 MB a nivel mundial (Cisco Systems Inc., 2018).

## 4. CONCLUSIONES

La comparación entre los datos obtenidos por el GPS L80 y las coordenadas adquiridas mediante el servicio de georreferenciación provisto por navegadores con HTML5, arrojó un desfase de 6,50 m. Este desfase evidencia la precisión del prototipo propuesto para ubicar un vehículo en tiempo real en el globo terrestre.

El uso de la Raspberry Pi 3 como servidor web dedicado brinda al proyecto portabilidad y disminución de costos de mantenimiento debido a las dimensiones y consumo energético de la misma. Adicionalmente, esto brinda escalabilidad al proyecto haciendo más sencilla la integración del servidor en cualquier punto.

El uso de Arduino facilita la realización de proyectos relacionados con el Internet de las cosas, pues esta plataforma permite la integración de diversos dispositivos, como por ejemplo módulos GPS y GPRS, así como librerías establecidas para manejar las tareas de una forma más sencilla.

El consumo de datos del prototipo desarrollado fue de 10,78 MB mensuales, lo que representa el 1,1 % del consumo promedio mensual de un usuario a nivel mundial.

Se comprobó la capacidad de procesamiento del sistema Raspberry Pi en el tratamiento de los datos recibidos desde Internet, como servidor web y dispositivo de almacenamiento de información en tiempo real a través del protocolo TCP/IP y el puerto 80.

## 5. REFERENCIAS

- Alam, C. N., Manaf, K., Atmadja, A. R. & Aurum, D. K., 2016. 2016 4th International Conference on Cyber and IT Service Management. Bandung, IEEE, pp. 1-6. <https://doi.org/10.1109/CITSM.2016.7577575>. [En línea]. Disponible en: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7577575&isnumber=7577456>
- Alcaraz, M., 2014. Internet de las cosas. Universidad Católica Nuestra Señora de la Asunción, pp. 2-3.



- Antúnez Romanuel, R. & Hernández Montero, L., 2014. Propuesta de paradigma para análisis geométricos en SIG independiente del sistema de referencia. *Revista Geográfica Venezolana*, 55(1), pp. 11-26. [En línea]. Disponible en: <http://www.redalyc.org/articulo.oa?id=347732465002>
- Bonilla Fabela, I., Tavizon Salazar, A., Morales Escobar, M., Guajardo Muñoz, L. T. & Laines Alamina, C. I., 2016. IoT, el internet de las cosas y la innovación de sus aplicaciones. *Vinculatégica efan*, Issue 1, pp. 2313-2340. [Online]. Disponible en: [https://www.researchgate.net/publication/326129401\\_IOT\\_el\\_internet\\_de\\_las\\_cosas\\_y\\_la\\_innovacion\\_de\\_sus\\_ap](https://www.researchgate.net/publication/326129401_IOT_el_internet_de_las_cosas_y_la_innovacion_de_sus_ap)
- Cisco Internet Business Solutions Group, 2018. The Internet of Things. [Online]. Disponible en: [https://www.cisco.com/c/dam/en\\_us/about/ac79/docs/innov/IoT\\_IBSG\\_0411FINAL.pdf](https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf) [Last access: 11 Junio 2018].
- Cisco Systems Inc., 2018. VNI Mobile Forecast Highlights, 2016-2021. [Online]. Disponible en: [https://www.cisco.com/c/dam/assets/sol/sp/vni/forecast\\_highlights\\_mobile/index.html](https://www.cisco.com/c/dam/assets/sol/sp/vni/forecast_highlights_mobile/index.html) [Last access: 10 Junio 2018].
- Conti, M., Dehghantanha, A., Franke, K. & Watson, S., 2018. Internet of Things security and forensics: Challenges and opportunities. *Future Generation Computer Systems*, 78(2), pp. 544-546. <https://doi.org/10.1016/j.future.2017.07.060>. [Online]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S0167739X17316667>
- Escobar Amado, C. D., Celis Peñaranda, J. M., Sepúlveda Mora, S. B. & Castro Casadiego, S. A., 2016. Servidor web y punto de acceso basado en un sistema embebido para la supervisión de un proceso desde una aplicación móvil con sistema operativo Android. *Ingenium*, Volumen 10, pp. 11-18. <https://doi.org/10.21774/ing.v10i27.616>. [Online]. Disponible en: <http://revistas.usc.edu.co/index.php/Ingenium/article/view/616/504#.W6pxHWhKjDc>
- Haleem SL, A. & Samsudeen, S. N., 2016. Real Time Bus Tracking and Scheduling System Using Wireless Sensor and Mobile Technology. *Journal of Information Systems and Information Technology*, Volumen 1, pp. 18-23. [Online]. Disponible en: <https://www.irjet.net/archives/V3/i4/IRJET-V3I4128.pdf>
- Lee, S., Tewolde, G. & Kwon, J., 2014. Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application. Seoul, s.n., pp. 353-358. [Online]. Disponible en: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6803187&isnumber=6803102>
- Liu, Z., 2017. Optimization of Wireless Data Transmission System Based on GPRS. *Boletín Técnico*, 55(17), pp. 631-635. [Online]. Disponible en: <http://revistadelafacultaddeingenieria.com/index.php/ingenieria/article/view/2608>
- Lomillos-Pérez, J., Varga Marta, A.-d. I., García-García, J. & Gaudioso-Lacasa, V., 2016. Study of Lidia cattle social behavior using technology GPS-GPRS. *Abanico Veterinario*, 6(3), pp. 35-46. [Online]. Disponible en: <http://www.medigraphic.com/cgi-bin/new/resumenI.cgi?IDARTICULO=69501>
- Márquez Díaz, J., Samp Pedro, L. & Vargas, F., 2002. Instalación y configuración de Apache, un servidor Web gratis. *Ingeniería y Desarrollo*, Issue 12,

- pp. 10-23. [Online]. Disponible en: <http://www.redalyc.org/articulo.oa?id=85201202>
- Morales Valenzuela, O. S., Camargo López, J. R. & Gaona García, E. E., 2015. Sistema de monitoreo para pacientes de alto riesgo integrando módulos GPS, GSM/GPRS y Zigbee. *Tecnura*, 19(spe), pp. 97-111. <https://doi.org/10.14483/22487638.10376>. [Online]. Disponible en: <https://revistas.udistrital.edu.co/ojs/index.php/Tecnura/article/view/10376>
- Obuhuma, J., Okoyo, H. & Mcoyowo, S., 2018. Real-time Driver Advisory Model: Intelligent Transportation Systems. Gaborone, 2018 IST-Africa Week Conference (IST-Africa), pp. 1-11. [Online]. Disponible en: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8417372&isnumber=8417286>
- Padilla, R., Quintero-Rosas, V. & Díaz-Ramírez, A., 2015. Monitoreo y localización de personas extraviadas utilizando Arduino y GSM/GPS. *Industrial Data*, 18(1), pp. 128-134. [Online]. Disponible en: <http://www.redalyc.org/articulo.oa?id=81642256015>
- QUECTEL, 2014. QUECTEL L80 Protocol. [Online] Disponible en: [http://www.quectel.com/UploadImage/Downlad/Quectel\\_L80\\_GPS\\_Protocol\\_Specification\\_V1.4.pdf](http://www.quectel.com/UploadImage/Downlad/Quectel_L80_GPS_Protocol_Specification_V1.4.pdf) [Last access: 03 Octubre 2017].
- Sanmartín Mendoza, P., Ávila Hernández, K., Vilora Núñez, C. & Jabba Molinares, D., 2016. Internet de las cosas y la salud centrada en el hogar. *Salud Uninorte*, Volumen 32, pp. 337-351. [Online]. Disponible en: <http://www.redalyc.org/articulo.oa?id=81748361017>
- Satyanarayana, K., Sarma, A., Sravan, J., Malini, M., & Venkateswarlu, G., 2012. GPS and GPRS Based Telemonitoring System for Emergency Patient Transportation. *Journal of Medical Engineering*, Volumen 2013, p. 9. [Online]. Disponible en: <https://www.hindawi.com/journals/jme/2013/363508/>
- Silva Pinto, F. & Silva Centeno, J. A., 2012. A realidade aumentada em smartphones na exploração de informações estatísticas e cartográficas. *Boletim de Ciências Geodésicas*, 18(2), pp. 282-301. [Online]. Disponible en: <http://www.redalyc.org/articulo.oa?id=393937724007>
- Singla, L. & Bhatia, P., 2015. GPS based bus tracking system. 2015 International Conference on Computer, Communication and Control (IC4), Septiembre, pp. 1-6. <https://doi.org/10.1109/IC4.2015.7375712>. [Online]. Disponible en: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7375712&isnumber=7374772>
- Sosa Pedroza, J. & Martínez Zúñiga, F., 2009. Los sistemas de información geográfica y su aplicación en enlaces de comunicaciones. *Científica*, 13(1), pp. 27-34. [Online]. Disponible en: <http://www.redalyc.org/articulo.oa?id=61412184005>
- Zapata Murillo, P. & Baldoquin, M., 2018. Vehicle location models for Emergency Medical Services. An application for a Colombian company. *Revista EIA*, 15(29), pp. 31-46. <https://doi.org/10.24050/reia.v15i29.1211>. [Online]. Disponible en: <https://revistas.eia.edu.co/index.php/reveia/article/view/1211>
- Zhang, D., Ning, H., Xu, K. S., Lin, F. & Tianruo Y. L., 2012. Internet of things. *Journal of Universal Computer Science*, Volumen 18(9), pp.

1069-1071. [Online]. Disponible en: [http://www.jucs.org/jucs\\_18\\_9/internet\\_of\\_things](http://www.jucs.org/jucs_18_9/internet_of_things)