

**TÍTULO:**

SISTEMA DE LOCALIZACIÓN MONITOREO Y CONTROL  
VEHICULAR BASADO EN LOS PROTOCOLOS  
GPS/GSM/GPRS

**CATEGORÍA:**

Trabajo Finales de Carreras de Grado

**AUTORES:**

Juan Pablo Astudillo León.  
jastudillo@est.ups.edu.ec  
Edgar Gustavo Delgado Tello.  
gdelgadot@est.ups.edu.ec  
Carrera de Ingeniería Electrónica  
Facultad de Ingenierías  
Universidad Politécnica Salesiana

**DIRECTOR:**

Ing. Vladimir Robles B.  
Ingeniero de Sistemas  
Carrera de Ingeniería de Sistemas  
Docente de la Universidad Politécnica Salesiana  
Facultad de Ingenierías  
vrobles@ups.edu.ec

Cuenca, Abril del 2012

# Sistema de localización, monitoreo y control vehicular basado en los protocolos GPS/GSM/GPRS

**Resumen** Este ensayo describe la implementación de un sistema de localización, monitoreo y control vehicular, cuyo objetivo es el almacenamiento, procesamiento y gestión de los datos que son enviados desde los dispositivos vehiculares hacia los servidores alojados en la nube, facilitando las consultas de reportes, historial y monitoreo desde cualquier dispositivo que tenga conexión de Internet. La técnica empleada para el desarrollo de la aplicación, es la utilización de los protocolos GPS/GSM/GPRS en conjunto con los servidores de aplicaciones web y sockets.

## 1. Introducción

Debido al gran incremento de vehículos y objetos de valor que se venden y utilizan a nivel mundial, se vuelve una necesidad conocer su ubicación exacta en el planeta, bien sea esto por seguridad en caso de robo, o por control y monitoreo de flotas en el caso de empresas que quieran saber cómo están utilizando sus bienes.

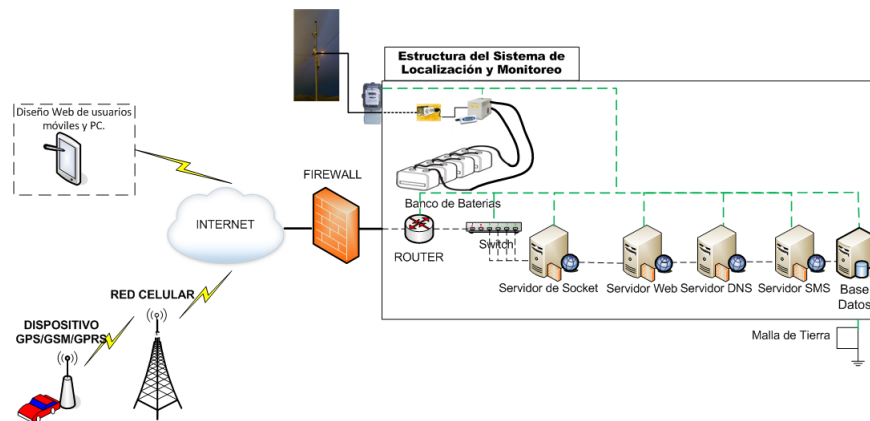
En este proyecto se plantea la construcción de una plataforma web que nos posibilita el monitoreo de dispositivos de rastreo GPS desde cualquier computador personal o dispositivo móvil que tenga acceso a Internet, lo que permitirá a los usuarios conocer la ubicación del dispositivo en tiempo real, que incluye información detallada de todo los eventos producidos por el vehículo, así como realizar acciones remotas en los equipos. Con la implementación de este sistema se puede optimizar las rutas recorridas por los vehículos, lo que se ve reflejado en un ahorro de gasolina, así como la reducción de costo por mantenimiento y disminución de la contaminación ambiental [7].

## 2. Modelo Propuesto

Lo que proponemos en este proyecto es la utilización de un equipo de rastreo que capturará señales de diferentes sensores del vehículo, que se enviarán a través de la red GSM/GPRS hacia nuestros servidores y serán procesadas y almacenadas en una base de datos, lo que permitirá que los usuarios ingresen a la aplicación web a través de ordenador o equipos móviles y que tengan acceso a la siguiente información en tiempo real:

1. Ubicación georeferenciada del vehículo (latitud y longitud) en un sistema de información geográfica (GIS).
2. Consultas sobre la trayectoria y generación de reportes de recorrido del vehículo, definido en dos instantes de tiempo gracias al manejo de una base de datos.
3. Soporte de una interfaz web amigable, robusta y segura que facilite la realización de las siguientes acciones:
  - Monitoreo del nivel de gasolina.
  - Control de estado de las puertas.
  - Velocidad y dirección del vehículo.
  - Reportes detallados del monitoreo.
  - Alarmas por botón de pánico SOS, desconexión de batería, excesos de velocidad y geocercas
  - Realizar el bloqueo del vehículo en caso de robo y levantamiento de seguros.

Las tecnologías y herramientas involucradas se ilustran en la figura 1, y se detallan a continuación.



**Figura 1.** Diagrama del Sistema.

### 2.1. Sistema de Posicionamiento Global GPS

En la actualidad, la tecnología disponible permite conocer la localización de puntos de interés, como ciudades, países, centros comerciales, incluyendo nuestros hogares y puntos muy frecuentados. Esto lo ha hecho posible la georeferenciación, que utiliza mapas digitales, que nos permiten marcar puntos con gran exactitud, conociendo sus coordenadas geográficas. Este sistema fue desarrollado por la Fuerza aérea de Estados Unidos en 1973 que se le conoce como

NAVSTAR-GPS<sup>1</sup> que utiliza un sistema de tres dimensiones: latitud, longitud y altitud [9].

## 2.2. Sistema Celular GSM/GPRS

Este sistema de red celular permite la comunicación de los equipos móviles con el servidor web mediante la nube de Internet. La red celular GSM (Global System for Mobile Communications) conocida como segunda generación (2G) utiliza TDMA<sup>2</sup>. Este sistema se caracteriza por que sus transmisiones son digitales y a diferencia de las primeras redes 1G que existieron, no se limita únicamente a voz, sino ya permite enviar datos, entre ellos mensajes de texto y se puede realizar transmisiones a una velocidad de hasta 14.4 Kbps en una estructura de conmutación de circuitos[5].

**General Packet Radio Service (GPRS)** La red GPRS es una tecnología que se utiliza para la transmisión de datos por conmutación de paquetes<sup>3</sup>, también conocida como GSM-IP. Surgió debido a los inconvenientes que presentaba la red GSM para la transmisión de datos, principalmente por que trabaja con conmutación de circuitos<sup>4</sup>, tecnología que funciona correctamente para la transmisión de voz[13].

**Ventajas de GPRS** Entre las principales ventajas tenemos[10]:

- Permite enviar y recibir información hacia una dirección IP.
- Utiliza conmutación de paquetes, por lo que se tarifa por tráfico consumido.
- Se utiliza el canal de comunicación por demanda, es decir, únicamente cuando se lo necesite, permaneciendo disponible el resto del tiempo.
- Permite tasas de transmisión de 56 a 144 Kbps.
- Permite enviar información (GPRS) y realizar llamadas (GSM) al mismo instante.

## 2.3. Equipo rastreador AVL

Este dispositivo electrónico AVL<sup>5</sup> nos permite monitorear un vehículo utilizando mensajes de texto SMS<sup>6</sup> o mediante GPRS hacia un servidor. Internamente dispone de un chip *With Sirf Star III GPS* que nos ayuda a determinar su

<sup>1</sup> NAVSTAR-GPS (NAVigation System Time And Ranging) es el sistema de navegación satelital controlado por el departamento de defensa de los EUA.

<sup>2</sup> Time Division Multiple Access, asigna a cada usuario un intervalo o slot de tiempo para que utilice el canal de comunicación[8].

<sup>3</sup> Conmutación de Paquetes, divide a la información en paquetes y los transmite individualmente, en el destino se re ensamblan y forman uno solo.

<sup>4</sup> Conmutación de circuitos, establece un canal dedicado durante toda la sesión.

<sup>5</sup> AVL, acrónimo de Automatic Vehicle Location.

<sup>6</sup> SMS, acrónimo en español de Servicios de Mensajes Cortos

posición, velocidad, fecha y dirección. Además el equipo AVL nos permite saber el estado de puertas, ventanas y cortar el suministro de combustible para lograr bloquear el vehículo.

## 2.4. Servidores

Para permitir el acceso de los usuarios hacia los servicios de la plataforma web, se necesita la configuración e instalación de diferentes servidores. Se denomina servidor a cualquier dispositivo que responde a una solicitud de un cliente[11]. Los servidores que se implementarán en la plataforma web son los siguientes:

**Servidor DNS** El Sistema de Nombres de Dominios (DNS) se encarga de resolver los nombres asociados a las direcciones numéricas en la red o IP.

**Servidor web** Cuando se escribe una dirección web en un explorador de Internet, el explorador establece una conexión con el servicio web del servidor que utiliza el protocolo HTTP<sup>7</sup>. Los exploradores web son las aplicaciones de cliente para conectarse con la Internet y acceden a los recursos almacenados en un servidor web.

El navegador de Internet primero verifica con un servidor de nombres para convertir el dominio a una dirección numérica para conectarse con el servidor.

**Servidor Gateway SMPP** El servidor Gateway <sup>8</sup> SMPP<sup>9</sup> es un sistema que permite conectar o enlazar dos sistemas que trabajan con formatos distintos de información, permitiendo la comunicación bidireccional entre ellos. En si, un SMS Gateway conecta la red de telefonía celular con un conjunto de servicios o aplicaciones SMS, para lo cual generalmente provee de una serie de interfaces para que desarrolladores de software puedan recibir, procesar y enviar un alto número de mensajes.

**Serversocket** Un Serversocket es un sistema que permite la comunicación entre diferentes procesos de dos dispositivos finales[12]. El Serversocket cumple un rol muy importante en el sistema, está implementado directamente en el servidor, es el encargado de escuchar procesos en el puerto y cuando llegan datos provenientes del dispositivo verifica el formato de estos e interpreta esta información para posteriormente almacenarlos en las tablas de la base de datos si la información es válida.

---

<sup>7</sup> HTTP o Hypertext Transfer Protocol, especifica un protocolo de solicitud/respuesta.

<sup>8</sup> Gateway: en telecomunicaciones es una puerta dedicada a intercomunicar sistemas de protocolos diferentes o incompatibles.

<sup>9</sup> SMPP acrónimo de Short Message Peer-to-Peer es un protocolo para el intercambio de mensajes de texto.

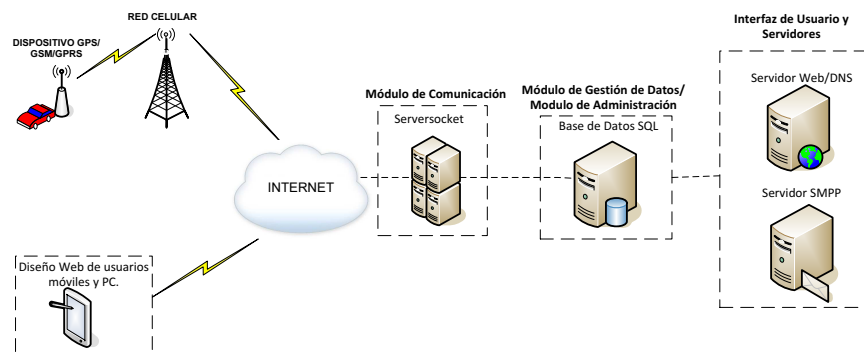
El equipo recibe internamente la información de sus periféricos (posición GPS, sensores analógicos, sensores digitales, velocidad y alarmas), y las encapsula en una trama, para posteriormente enviarla al servidor. Utilizando este formato el equipo envía el IMEI<sup>10</sup>, el tipo de alarma en caso de que ocurra una en ese instante, la posición del equipo GPS, el estado de los sensores digitales, las entradas analógicas y se agrega al final información relacionada a la detección de errores con la ayuda del checksum y CRC<sup>11</sup>.

### 3. Experimentación realizada

La implementación de nuestro sistema se realiza en base al diseño, configuración, programación e instalación de los siguientes módulos:

- Módulo de Comunicación: Define el protocolo de comunicación entre el cliente y el servidor.
- Módulo de Gestión de Datos: Almacena los datos de los usuarios y variables de los vehículos, además define la relación de dependencia entre usuarios y dispositivos.
- Módulo de Administración: Encargado de la gestión y manejo de usuarios y dispositivos.
- Interfaz de Usuario: Es el medio por el cual el usuario puede interactuar con el vehículo al realizar cualquier consulta o una acción en él.

El módulo de comunicación es el núcleo del sistema (como se puede ver en la figura 2), debido a que trabaja en conjunto con el módulo de gestión de datos permite que el usuario pueda consultar y tomar decisiones en los dispositivos.



**Figura 2.** Diseño del Sistema

<sup>10</sup> IMEI acrónimo de International Mobile Equipment Identity es un código único que nos ayuda a identificar los equipos en la red GSM

<sup>11</sup> Comprobación de Redundancia Cíclica es un algoritmo que permite detectar si existen errores en los datos.

La implementación de cada uno de los módulos se detallan a continuación:

### 3.1. Módulo de Comunicación

El módulo de comunicación consta básicamente de la programación de un servidor de sockets. Existen dos tipos de Serversocket para implementar: Serversocket TCP y Serversocket UDP. Para el proyecto se implementó un Serversocket UDP, debido a que se requiere velocidad en la transmisión de datos, menos uso de recursos por parte del servidor, la confiabilidad en los datos se realizará con CRC, además UDP tiene menos bytes en su cabecera de información, por lo cual la facturación mensual del servicio GPRS será menor.

El módulo de comunicación para nuestra aplicación se presenta en la figura 3.

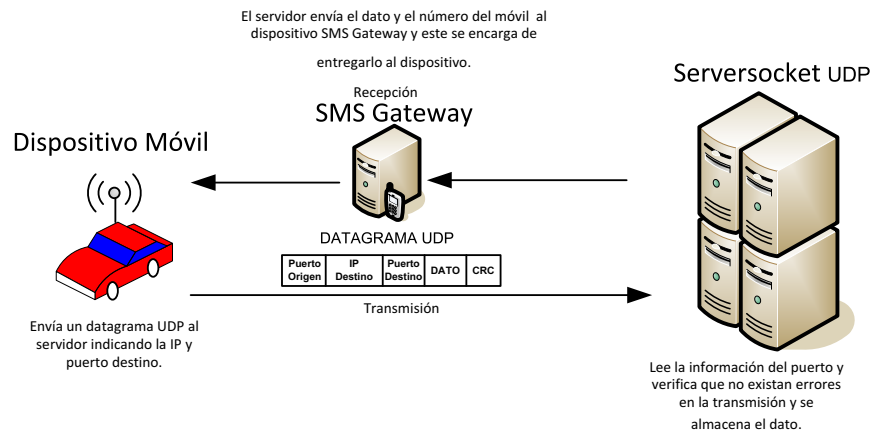


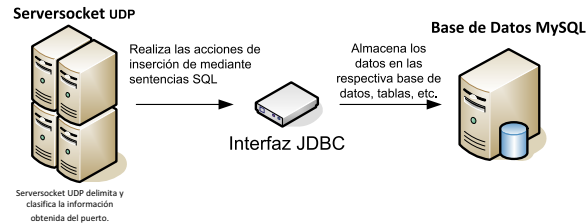
Figura 3. Módulo de Comunicación

### 3.2. Módulo de Gestión de Datos

Para la gestión, administración y manejo de los datos hemos optado por la base de datos relacional MySQL.

**Base de Datos MySQL** La administración de la base de datos se realizó con la aplicación phpMyAdmin que es un software gratuito escrito en PHP. Esta herramienta permite gestionar las bases MySQL mediante una interfaz web y permite realizar muchas tareas como: administrar bases, tablas, campos, relaciones, gestionar usuarios, permisos, importar y exportar para realizar backups entre otras opciones más.

La interfaz que nos permite almacenar la información en la base de datos desde Java es el conector JDBC(Java Database Connectivity). En la figura 4 se indica el diagrama del módulo de gestión de datos.



**Figura 4.** Módulo de Gestión de Datos

### 3.3. Módulo de Administración

El módulo de administración es el encargado de gestionar los datos de los usuarios tales como agregar, editar y eliminar usuarios y vehículos, y controlar el acceso a los datos para garantizar que múltiples usuarios consulten la información según su nivel de privilegios, todas estas características son gestionadas por la base de datos.

### 3.4. Interfaz de Usuario

La plataforma web implementada permite a los usuarios realizar las siguientes acciones:

- Autenticación de usuarios y manejo de sesiones.
- Rastreo en tiempo real de uno o más vehículos.
- Generar reportes con información detallada del nivel de gasolina, temperatura, niveles de voltaje, geocercas, contador de distancias y alarmas.
- Configuración de geocercas para saber cuando un vehículo ingresó o salió de una área geográfica o de interés.
- Configuración de la cuenta.
- Envío de mensajes del servidor a los dispositivos o usuarios desde la pagina web o desde el Serversocket.
- Bloqueo del vehículo y manejo remoto de los seguros.

Las tablas, gráficas estadísticas y los indicadores gráficos tales como la velocidad están basados en HTML5<sup>12</sup>, lo que permite manipular y dar eventos a las gráficas, mejorando la experiencia del usuario al consultar datos específicos.

Las herramientas implementadas para la programación de la página web se detallan en la siguiente sección.

<sup>12</sup> HTML5 nueva versión de HTML que engloba todas las funciones de la Web en una sola, por ejemplo la implementación de gráficos, imágenes o animaciones sin el uso de herramientas como Flash



**Lenguaje de Programación del lado del Servidor (PHP).** Este lenguaje de programación del lado del servidor nos permite realizar las consultas y acciones desde la página web hacia la base de datos.

**Lenguaje de Programación del lado del cliente (Javascript)** Además de añadir interactividad a las páginas HTML y validar información en la página, permite manejar los eventos, servicios y superposiciones en los mapas y tablas de reportes.

**AJAX** Ajax permite que los eventos o acciones desde la página web con Javascript puedan comunicarse directamente con el servidor, por lo cual pueden intercambiar datos sin recargar la página.

**XML y JSON** Para nuestro proyecto hemos estructurado la información en estos formatos. Con estas estructuras se facilita la implementación de esta información de datos geográficos y variables del vehículo para su posterior visualización en la página, y elementos web que utilizan la técnica de Ajax.

### **3.5. Sistemas de Información Geográfica(SIG)**

Un sistema de información geográfica es un sistema que proporciona los cuatro siguientes conjuntos de capacidades para el manejo de datos georeferenciados: entrada de los datos, gestión de los datos(almacenamiento y recuperación), manipulación y análisis, y salida de datos[14].

La información de los SIG se encuentra por capas. Para nuestro proyecto se implementó el sistema de información geográfica de Google, debido a que se encuentra en permanente actualización y brinda muchas posibilidades a los desarrolladores para incluir en sus páginas y brindar servicios en ella. Entre los servicios más importantes se encuentra la geocodificación inversa, que es el proceso de entregar una dirección para ciertos valores de latitud y longitud.

## **4. Resultados obtenidos**

A continuación se explicarán los resultados obtenidos en nuestro proyecto, para cada una de las opciones de la página web expuestas anteriormente.

### **4.1. Rastreo en tiempo real**

Conocer cómo se manejan nuestros dispositivos móviles en tiempo real es un aspecto vital en nuestro proyecto, por lo que se implementó una interfaz web que permite conocer la última localización del vehículo, la velocidad y dirección que ha tomado, el tipo de alarma y su localización, los estados de las geocercas y puertas. Para realizar la acción de rastreo es necesario indicar el usuario y vehículo, como se puede observar en la figura 5.

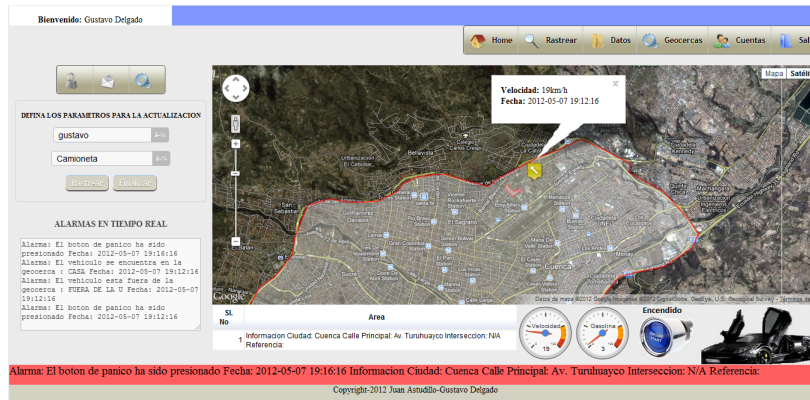


Figura 5. Rastreo del vehículo en tiempo en real..

#### 4.2. Historial de ruta del vehículo y reportes de geolocalización

Esta interfaz web facilita al usuario conocer la ruta vehicular que tomó el dispositivo en dos instantes de tiempo, indicando la dirección y velocidad de cada punto con el uso de marcadores personalizados en un mapa. Además de esto podemos generar reportes detallados que indican la hora, fecha, velocidad y lugar donde se encontró el vehículo en cierto punto y con la ayuda de un mapa adicional se puede analizar de manera más fácil los reportes generados, ya que cualquier evento seleccionado de la tabla se visualizará de forma inmediata en el mapa gracias a la utilización de HTML 5. Igualmente, para realizar una consulta de la ruta vehicular es necesario seleccionar además del usuario y vehículo, la fecha de inicio y fin de recorrido, como también el algoritmo de geocodificación inversa. El resultado de esta consulta se visualiza en las figuras 6 y 7. Cabe recalcar que la aplicación es capaz de calcular la distancia recorrida aproximada del vehículo.

Nuestro sistema es capaz de generar reportes PDF (como se puede ver en la figura 8), los cuales explican de una manera más detallada los eventos relacionados con la ruta del vehículo. Por ejemplo, el periodo de tiempo en el que el vehículo estuvo en una zona y velocidades de recorrido.

#### 4.3. Reportes de los niveles de voltaje analógicos, gasolina y velocidad.

Conocer las diferentes variaciones de gasolina, velocidad, voltaje ya sea de la alimentación del dispositivo o de la batería, apoya a las personas que están a cargo del control de estos vehículos en la toma de decisiones y optimización de las rutas y el uso de las mismas. En virtud de esto, implementado una interfaz web para que estos parámetros sean visualizados por medio de una gráfica, que además consta de una tabla para generar reportes que detallan los valores físicos que mide el equipo. Con el uso de un mapa se puede consultar la localización



Figura 6. Consulta de la ruta del vehículo, dirección y velocidad en el mapa de Google.

Fecha	Hora	Velocidad	Distancia Recorrida	Localización
2012-04-11	22:30:55	0km/h	0m	Ciudad: Cuenca Calle Principal: EL Universo Intersección: El Mercurio
2012-04-11	22:38:00	13km/h	0.0364328821770292km	Ciudad: Cuenca Calle Principal: EL Universo Intersección: El Mercurio
2012-04-11	22:39:31	48km/h	0.547201850612206km	Ciudad: Cuenca Calle Principal: Av. Ventiuno de Mayo Intersección: Las Princesas
2012-04-11	22:39:43	34km/h	0.655465168102773km	Ciudad: Cuenca Calle Principal: Av. Ventiuno de Mayo Intersección: Las Princesas
2012-04-11	22:39:53	35km/h	0.771124398493059km	Ciudad: Cuenca Calle Principal: Av. Ventiuno de Mayo Intersección: N/A
2012-04-11	22:40:00	10km/h	0.82020988168176029km	Ciudad: Cuenca Calle Principal: Montevideo Intersección: N/A
2012-04-11	22:40:17	22km/h	0.8934498168177762km	Ciudad: Cuenca Calle Principal: Popayan Intersección: Santiago
2012-04-11	22:40:20	20km/h	0.8910964739524154km	Ciudad: Cuenca Calle Principal: Popayan Intersección: Santiago
2012-04-11	22:40:40	0km/h	0.995395042019179km	Ciudad: Cuenca Calle Principal: Popayan Intersección: Paito
2012-04-11	22:41:01	21km/h	0.998284971343849km	Ciudad: Cuenca Calle Principal: Av. Ventiuno de Mayo Intersección: Paito
2012-04-11	22:41:12	39km/h	1.113812127132404km	Ciudad: Cuenca Calle Principal: Av. Ventiuno de Mayo Intersección: Basilia
2012-04-11	22:41:24	19km/h	1.220168134030950km	Ciudad: Cuenca Calle Principal: Av. Ventiuno de Mayo Intersección: Rayoloma
2012-04-11	22:41:37	20km/h	1.283057396785054km	Ciudad: Cuenca Calle Principal: Av. Ventiuno de Mayo Intersección: Rayoloma
2012-04-11	22:41:51	39km/h	1.422096120110320km	Ciudad: Cuenca Calle Principal: Rayoloma Intersección: Emiliano Zapata

Figura 7. Reporte generado de la geolocalización.

geográfica exacta donde se generó cierto valor. Igualmente el usuario tiene la posibilidad de generar documentos PDF de cada una de las consultas realizadas.

**Reporte de velocidad** Los reportes de velocidad detallan la hora a la que un vehículo rebasó la velocidad o dejó de moverse, como se puede apreciar en la figura 9 el sistema es capaz de visualizar por medio de una tabla, gráfica o mediante un mensaje que indica el momento y lugar ocurrió un exceso de velocidad.

**Reporte del nivel de voltaje de la batería y voltaje de alimentación del dispositivo** Estos reportes son muy importantes, debido a que informan en el momento en el cual se ha desconectado la batería principal de vehículo, indicando un posible robo, desmantelamiento o un uso inapropiado del vehículo.

**UNIVERSIDAD POLITECNICA SALESIANA**

## Reporte de recorrido de vehiculo

**Usuario:** Gustavo\_Delgado  
**Vehiculo:** Camioneta PSJ-839

Consulta realizada desde: 2012-04-12 13:20:00 hasta: 2012-04-13 23:20:00

HORA	FECHA	ESTADO	LOCALIZACION
2012-04-12	13:20:27	El vehiculo se movio a una velocidad de 28km/h	Ciudad: Cuenca Calle Principal: Av. Hurtado de Mendoza Interseccion: N/A
2012-04-12	13:22:12	El vehiculo se movio a una velocidad de 10km/h	Ciudad: Cuenca Calle Principal: Av. Hurtado de Mendoza Interseccion: Av. Paseo de los Canaris
2012-04-12	13:23:57	El vehiculo se movio a una velocidad de 16km/h	Ciudad: Cuenca Calle Principal: Av. Paseo de los Canaris Interseccion: Pajan
2012-04-12	13:25:44	El vehiculo se movio a una velocidad de 43km/h	Ciudad: Cuenca Calle Principal: Av. Max Uhle Interseccion: Ivan Salgado
2012-04-12	13:27:30	El vehiculo estuvo en este lugar 1 horas, 26 minutos con 22 segundos	Ciudad: Cuenca Calle Principal: EL Universo Interseccion: El Mercurio

**Figura 8.** Documento PDF generado de la geolocalización.

Los voltajes nominales para la alimentación del dispositivo varían de 12V a 24V, mientras que el voltaje interno de la batería oscila entre los 3.5V y 4.2V. Como se aprecia en la figura 10, los niveles de voltaje suministrado al dispositivo presentan oscilaciones, por ejemplo los picos de voltaje indican un posible arranque del vehículo mientras que un voltaje igual a cero indica que el dispositivo GPS fue removido o que las conexiones de alimentación de la batería principal del vehículo han sido desconectadas del circuito eléctrico.

#### 4.4. Historial de alarmas y eventos

Esta interfaz web permite al usuario conocer el historial de todas las alarmas y eventos que ocurrieron en el vehículo, como las alarmas SOS, excesos de velocidad, estados de los niveles de voltaje del dispositivo y puertas del vehículo, entre otras. El sistema es capaz de realizar la consulta de una alarma en particular o de todas las alarmas en el mismo momento. Al igual que las interfaces web anteriores, un mapa nos permite conocer la localización específica donde se generó el evento. Un reporte generado para la alarma SOS se indica en la figura 11.

#### 4.5. Configuración, gestión y reportes de geocercas

Con la implementación de geocercas el usuario puede conocer cuando un vehículo ingresa o abandona un área geográfica determinada. Esto es muy útil cuando se manejan flotas de vehículos, cuando el gerente o administrador necesita saber a qué hora el vehículo llegó a un lugar determinado o si tomó una ruta diferente a la habitual.

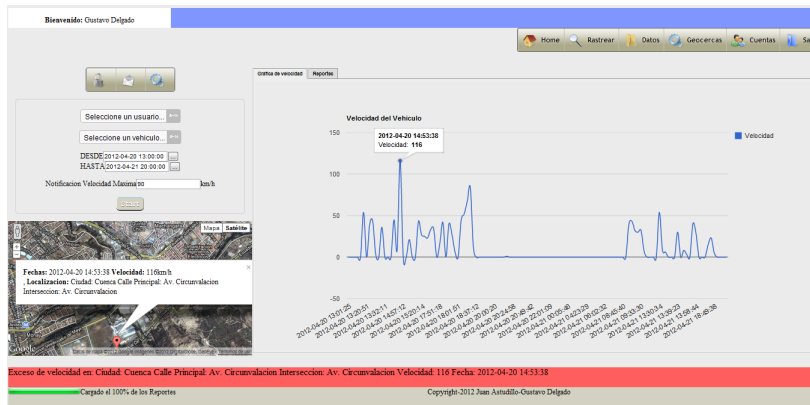


Figura 9. Gráfica de la velocidad del vehículo en dos instantes de tiempo.

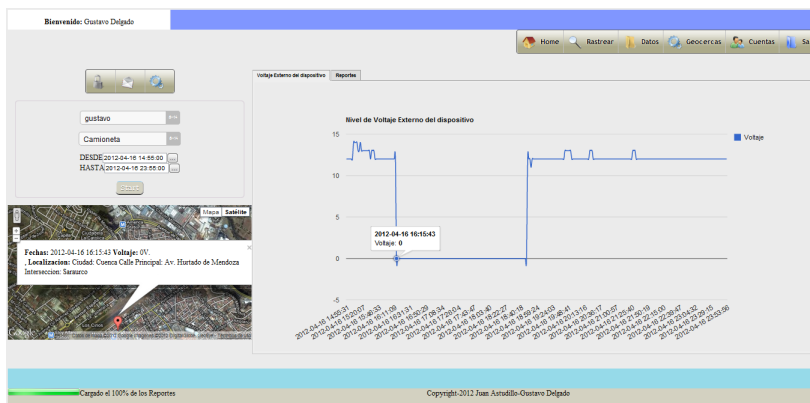


Figura 10. Gráfica del nivel de voltaje suministrado al dispositivo.

Esta interfaz web posibilita al usuario crear cualquier tipo de geocerca sin importar la forma o tamaño. Los vértices del polígono que conforman la geocerca se configuran a través de eventos del ratón en el mapa. También el usuario tiene la posibilidad de modificar, eliminar la geocerca o definir una nueva con el mismo nombre. Para generar una consulta en esta página web basta con indicar la geocerca, las fechas de inicio-fin, y definir el tipo de notificación que informa al usuario el estado del vehículo en la geocerca, esto se lo puede apreciar en la figura 12, en lo cual se indica los puntos en los cuales el vehículo se encontró fuera de la geocerca. Cabe recalcar que se ha generado una tabla que detalla de manera específica los datos de la geocerca.

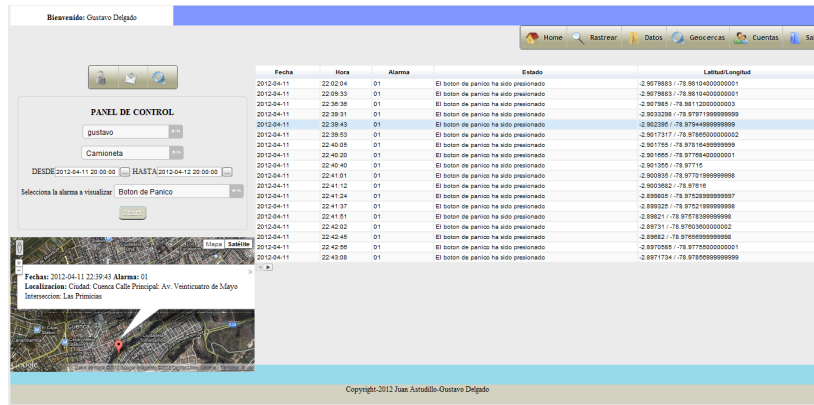


Figura 11. Reporte generado para la alarma SOS.

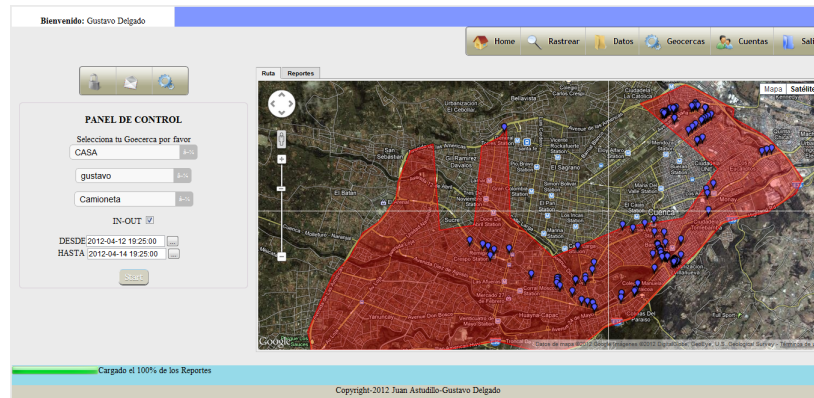


Figura 12. Consulta de un reporte de geocercas en el mapa.

#### 4.6. Servicio SMS

Es importante dar a conocer al usuario de todos los eventos que están sucediendo en el sistema y vehículo, por ejemplo, mediante mensajes de texto notificamos al usuario el momento que ingresó en el sistema o cuando las alarmas SOS fueron activadas, estas acciones corresponden al servidor de sockets y el servidor web, debido a que trabajan en conjunto con el servidor SMS, por lo que la administración de los mensajes se pueden realizar mediante clientes Java y PHP. Cabe recalcar que el usuario es capaz de realizar varias acciones en el vehículo, tales como el bloqueo de gasolina y apertura remota de seguros. En la figura 13 se aprecia algunos de tipos de mensajes que son generados por los diferentes servidores.



**Figura 13.** Ejemplos de mensajes SMS enviados por el servidor.

## 5. Conclusiones

Las comunicaciones mediante el protocolo GPRS son inmediatas y nos ayudan a optimizar los tiempos de envío en contraste con el servicio SMS, que no nos garantiza que los datos lleguen en el instante que fueron enviados, debido a retardos producidos por la cantidad de usuarios que utilizan el servicio. Entre otras de las ventajas que tiene GPRS ante SMS para la comunicación, es que la transmisión con el uso de paquetes IP permite al sistema detectar y corregir errores en la transmisión con el uso de algoritmos como CRC, por lo que a la presencia de alteraciones en la trama el servidor solicita la retransmisión del paquete si es que es necesario.

Para optimizar la transmisión de paquetes se optó por la implementación del protocolo de la capa de transporte UDP, debido a que los requerimientos del sistema necesitan que la comunicación entre el dispositivo y el servidor sea en tiempo real con una baja sobrecarga de información, lo que mejora el procesamiento de información sin recargar los servidores.

El uso de herramientas de software libre para la programación de la plataforma web y los servidores, permite que nuestro sistema no se restrinja a licencias, y además todas estas herramientas pueden funcionar en todos los sistemas operativos presentes en el mercado.

El servidor SMS Gateway permite la administración, gestión y optimización del envío de los mensajes masivos de texto a los destinatarios, cabe recalcar que con el uso de este tipo de servicios no es necesario el uso de un módem para el envío de mensajes, ya que el mismo resultado se puede conseguir con el uso de llamadas HTTP encriptadas, en sí esto depende de los requerimientos tanto técnicos y económicos del sistema a implementarse.

## 6. Trabajo Futuro

Se tiene previsto implementar nuestra propuesta en aplicaciones móviles que utilicen herramientas como Android, lo que permitiría al usuario conocer el estado de su dispositivo sin el uso de un navegador web para el monitoreo, mejorando así la experiencia entre el usuario y el sistema.

Siguiendo la misma estructura planteada para la comunicación y monitoreo de equipos GPRS, pensamos incursionar en áreas de Telemática para controlar, manejar y gestionar remotamente equipos y procesos industriales como sistemas SCADA y también en áreas como la domótica.

## References

1. Robert Vance Barbara Mims David Moss, Vince Celindro. An Introduction to Firewalls. *Northwestern University Information Technology*, page 21.
2. Universidad de Valladolid. Sistemas Distribuidos::Sockets en Java. *Departamento de informatica*, page 13.
3. Yerko Muñoz Felipe Fernández. JDBC. *Departamento de Ciencias de la Computacion de la Universidad de Chile*.
4. Darel Rex Finley. Determining Whether A Point Is Inside A Complex Polygon.
5. T. Halonen, J. Romero, and J. Melero. *GSM, Gprs and Edge Performance: Evolution Towards 3g/Umts*. John Wiley & Sons, 2004.
6. Automatic Vehicle Location. TZ-AVL05 User Guide and Manual. page 48.
7. Jerzy Mikulski. Transport Systems Telematics 10th Conference, Using Telematics in Transport. Volume 104:Pages 175–182, 2011.
8. T. S. Rappaport and S. B. O. Firme. *Wireless communications: Principles and practice*, volume 2. Prentice Hall PTR New Jersey, 1996.
9. J. R. Rey. El Sistema de Posicionamiento Global-GPS. 2010.
10. A. K. Salkintzis, C. Fors, and R. Pazhyannur. WLAN-GPRS integration for next-generation mobile data networks. *Wireless Communications, IEEE*, 9(5):112 – 124, 2002.
11. I. Sommerville and M. I. A. Galipienso. *Ingenieria Del Software*. Fuera de colección Out of series. Pearson Educación, 2005.
12. A. S. Tanenbaum. *Redes de Computadoras*. Pearson Education, 2003.
13. J. Tisal. The GSM network the GPRS evolution: One step towards UMTS. *Recherche*, 67:02, 2001.
14. N. L. Zarzosa, Andrés, and A. N. *Sistemas de información geográfica. Prácticas con Arc View*. Politext (Universidad Politécnica de Cataluña). UPC, 2002.

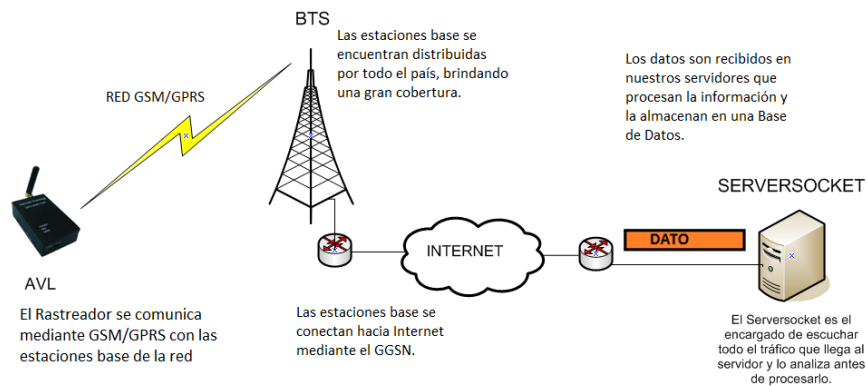


## ANEXOS

### 1. Programación del Serversocket

El Serversocket cumple un rol muy importante en el sistema, está implementado directamente en el servidor, es el encargado de escuchar procesos en el puerto y cuando llegan datos provenientes del dispositivo verifica el formato de estos e interpreta esta información para posteriormente almacenarlos en las tablas de la base de datos si la información es válida.

En la figura 1 se puede apreciar una representación de su trabajo.



**Figura 1.** Serversocket.

La programación del Serversocket se realizó en Java, y debido a que el equipo utiliza un formato propio para encapsular el paquete, nuestra aplicación debe conocer como está estructurado este para poder analizarlo, procesarlo y verificar que no existan errores en la transmisión.

#### 1.1. Sockets en Java.

Java dispone de herramientas para poder trabajar directamente en la red y realizar aplicaciones que comuniquen equipos. Para implementar los sockets utilizamos las clases del paquete java.net, lo que nos evita trabajar directamente con los protocolos de la capa de transporte. Este paquete dispone de varias clases, las cuales son de gran utilidad en función de la aplicación a utilizar[2].

Para la programación de nuestro Serversocket hemos implementado la clase DatagramSocket<sup>1</sup> que escucha el tráfico de entrada y lo almacena en una variable para su posterior manipulación.

El funcionamiento es el siguiente:

1. El Socketsserver establece el puerto por el que recibirá los datos, por lo tanto se encuentra pendiente, a la espera de que llegue información, la clase DatagramSocket es la encargada de realizar esta tarea.
2. Una vez que tiene datos en el buffer de memoria, una clase de verificación implementada analiza la cadena almacenada y valida si cumple con las características del formato.
3. Inmediatamente se procede a separar los diversos parámetros embebidos en la trama y gracias a las clases y métodos del paquete java.sql se procede a almacenar los datos en las tablas de la base.

Para evitar problemas de seguridad en los equipos, tanto en el servidor como en el cliente, se deben configurar firewalls<sup>2</sup> para evitar que los puertos abiertos en el lado del servidor y del cliente no nos causen problemas de seguridad o accesos no autorizados. En la figura 2 se aprecia la implementación de firewall en la comunicación.



**Figura 2.** Firewall en la comunicación.

Los Firewalls contienen las restricciones y reglas necesarias para permitir el acceso únicamente de ciertas aplicaciones a determinados puertos, controlando así su uso.

### 1.2. Formato de Datos que envía el dispositivo.

El equipo recibe internamente la información de sus periféricos (posición GPS, sensores analógicos, sensores digitales, velocidad y alarmas), y las encapsula en una trama, para posteriormente enviarla al servidor.

El formato de los datos enviados se lo puede apreciar en la figura 3.

<sup>1</sup> DatagramSocket, permite implementar un cliente y un servidor que utilice UDP para comunicarse, este socket no está orientado a la conexión.

<sup>2</sup> Firewall es un hardware o software o una combinación de los dos, que se usa para prevenir que programas o usuarios de Internet no autorizados accedan a redes privadas o computadores [1].

Format: \$\$ (2 Bytes) + Len (2 Bytes) + IMEI (15 Bytes) + | + AlarmType (2 Bytes) + GPRMC + | + PDOP + | + HDOP + | + VDOP + | + Status (12 Bytes) + | + RTC (14 Bytes) + | + Voltage (8 Bytes) + | + ADC (8 Bytes) + | + LACCI (8 Bytes) + | + Temperature (4 Bytes) | + Mile-meter (14 Bytes) + | Serial (4 Bytes) + | + Checksum (4 Bytes) + \r\n (2 Bytes)

**Figura 3.** Formato de la trama.

En el cuadro 1 se describe cada uno de los parámetros que conforma la trama GPRS.

Código	Descripción
\$\$	Indica la cabecera del paquete al servidor.
Len	Indica la longitud del paquete.
IMEI	Identificador del módulo GSM.
Alarma Type	El tipo de alarma activada.
GPRMC	Cadena de texto que transmite los datos del vehículo.
PDOP	Variación posicional(3D) de la medición del GPS.
HDOP	Variación horizontal de la medición del GPS
VDOP	Variación vertical de la medición del GPS
ADC	Conversor analógico/digital.
LACCI	Información relacionada a la red celular.
Milimeter	Temperatura del dispositivo.
Serial	Contador de paquetes GPRS.
Checksum	CRC de la trama GPRS.

**Cuadro 1.** Parámetros de la trama GPRS.

Utilizando este formato el equipo envía el IMEI, el tipo de alarma en caso de que ocurra una en ese instante, la posición del equipo GPS, el estado de los sensores digitales, las entradas analógicas y se agrega al final información relacionada a la detección de errores con la ayuda del checksum y CRC<sup>3</sup>.

En virtud de que se conoce el tamaño en bytes de cada dato que está contenido en la trama, y también se cuenta con un caracter “|” que identifica el inicio del siguiente dato, se puede separar y obtener la información de cada variable.

### 1.3. Procesamiento de la trama.

Una vez que tenemos una trama con datos en el servidor, el paso siguiente es extraer la información que tiene encapsulada. En Java programamos una clase para manipular la cadena de texto que contiene los datos, esta clase tiene métodos que se encargan de seleccionar cada uno de los parámetros. Para delimitar la trama de forma correcta nos basamos en la información de la figura 3 que nos

<sup>3</sup> CRC (Código de redundancia cíclica) es un código usado para detectar errores.

indica detalladamente la posición de cada valor y los caracteres que identifican el tipo de información.

En la figura 4 se muestra un ejemplo de una trama que llega en el servidor UDP.

```

$$B2359772035580357|61$GPRMC,032930.000,A,0254.4715,S,07858.8745,W,0.00,030212,,
A*7E|02.9|01.7|02.3|000000000000|20120203032930|04040000|00000000|27F5346E|000
0|0.0000|0071|278F

```

**Figura 4.** Ejemplo de dato enviado[6].

Cuando la información de la trama esta separada, se procede a almacenarla en la base de datos MySQL desde Java. Java tiene un paquete que nos permite guardar, leer y manipular información de bases de datos en MySQL, para luego utilizarla en la plataforma web. Para poder almacenar desde Java los datos en MySQL se debe hacer uso de un API<sup>4</sup>. El API usado es JDBC (Java DataBase Connectivity), que es un paquete de Java que nos permite trabajar directamente con bases relacionales con el uso de comandos SQL<sup>5</sup> y procesar los resultados[3].

## 2. Algoritmo para la implementación de la Geocodificación Inversa

Un método recursivo fue implementado para el proceso de geocodificación inversa, debido a que el periodo de tiempo en el que sistema se demora en calcular las direcciones para ciertos valores de latitud y longitud es variable, los factores que influyen en este cálculo es la capacidad de los servidores de mapas para distinguir intersecciones con calles principales, secundarias y avenidas o si la zona consultada no se encuentra correctamente georeferenciada. Los resultados de la geocodificación inversa se manejan en formato JSON (como se puede ver en la figura 5), y el algoritmo de la figura 6 es el encargado de procesar estos datos.

Como se puede observar la función **geocodificacionInversa(i)** utiliza para el calculo de este proceso el método **ReverseGeocode**, el cual requiere como parámetros de ingreso los valores de la latitud, longitud y dos funciones adicionales que manejan los resultados y errores respectivamente. El rol que juega el método recursivo en el calculo de direcciones, es que el sistema debe asegurar que los resultados de la geocodificación inversa se realicen de manera cronológica, ya que para realizar un nuevo calculo el sistema debe esperar a que se haya finalizado el anterior.

<sup>4</sup> API, Interfaz de Programación de Aplicaciones.

<sup>5</sup> SQL es un lenguaje que nos permite realizar consultas en las bases de datos.

<b>longitude</b> [Number]	Coordenadas Geográficas.
<b>latitute</b> [Number]	Coordenadas Geográficas.
<b>successFunction(results)</b>	La función que obtendrá los resultados de la consulta a ser procesados. Los resultados devueltos se encuentran en formato JSON y son:  <pre>{ State: value,   City: value,   Subcity: value,   Reference: value,   MainStreet: {     Alias: value,     MainName: value,     Nomenclature: value   },   Intersection: {     Alias: value,     MainName: value,     Nomenclature: value   },   Coordinates: {     Latitude: value,     Longitude: value   } }</pre>
<b>errorFunction(results)</b>	[DESCRIPCION]

**Figura 5.** Parámetros de la función ReverseGeocode para la geocodificación inversa.

### 3. Algoritmo que detecta el estado de las geocercas

El algoritmo que verifica si un punto geográfico se encuentra dentro o fuera de geocerca hace lo siguiente:

1. Trazar una horizontal a lo largo del polígono desde el punto de interés, y cada lado que corta la horizontal se le conoce como nodo.
2. Si el número de nodos a cada lado del punto de interés es impar, entonces el punto de interés se encuentra dentro del polígono, caso contrario se encuentra fuera de este, como se aprecia en la figura 7.
3. Además el algoritmo a implementar debe diferenciar cuando la línea horizontal interseca un vértice de dos lados del polígono, ya que podría considerarlo como dos nodos, además debe registrar solamente un nodo cuando varios nodos pertenecen a una misma cara del polígono, ver figura 7.

El algoritmo implementado en nuestro proyecto(ver figura 8 ) está basado en uno realizado previamente en lenguaje de programación C[4], sin embargo el algoritmo está modificado para que funcione con coordenadas geográficas en conjunto con las funciones de Googlemaps.

El algoritmo **verificarGeocerca** necesita como parámetros de ingreso el punto geográfico a analizar y los vértices del polígono de la geocerca. Cabe recalcar que este algoritmo este implementado en la página web y en una aplicación

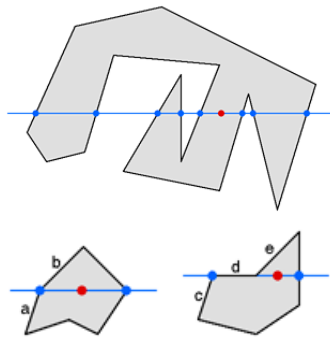
```

function geocodificacionInversa(i){
    objGeoprocessing.ReverseGeocode(longitudes[i],latitudes[i] ,
    MostrarResultados, ManejarError);
}

function MostrarResultados(results) {
    if (results != null) {
        cadena = " Ciudad: "
        cadena += results.City;
        if (results.City == null)
        {
            cadena="Lugar no Encontrado";localizacion[cont]=cadena;
            cont=cont+1;
            if(cont<latitudes.length){
                geocodificacionInversa(cont);
            }
            else
            {
                cargarElementos();
            }
        }
    }
    else {
        if (results.MainStreet.MainName != null)
        {
            cadena += "\nCalle Principal: "
            cadena += results.MainStreet.MainName;
        }
        if (results.Intersection.MainName != null)
        {
            cadena += "\n Interseccion: "
            cadena += results.Intersection.MainName;
        }
        if (results.Reference != null)
        {
            cadena += "\nReferencia: "
            cadena += results.Reference;
        }
        localizacion[cont]=cadena;
        cont=cont+1;
        if(cont<latitudes.length){
            geocodificacionInversa(cont);
        }
        else
        {
            cargarElementos();
        }
    }
}
}
}

```

**Figura 6.** Algoritmo recursivo para el proceso de geocodificación inversa.



**Figura 7.** Proceso para verificar si un punto se encuentra dentro o fuera de un polígono[4].

de Java, por lo cual el usuario siempre estará informado del estado de las geo-

```

function verificarGeocerca(location,geofence){
  var vertices = geofence.getPath();
  var x = location.lng();//Longitud
  var y = location.lat();//Latitud
  var j=0;
  var oddNodes = false;
  for (var i=0; i < vertices.length; i++)
  {
    j++;
    if (j == vertices.length) {j = 0;}
    if (((vertices.getAt(i).lat() < y) && (vertices.getAt(j).lat() >= y))
    || ((vertices.getAt(j).lat() < y) && (vertices.getAt(i).lat() >= y))) {
      if ( vertices.getAt(i).lng() + (y - vertices.getAt(i).lat()
      / (vertices.getAt(j).lat()-vertices.getAt(i).lat()
      * (vertices.getAt(j).lng() - vertices.getAt(i).lng())<x ) {
        oddNodes = !oddNodes
      }
    }
  }
  return oddNodes;
}

```

**Figura 8.** Algoritmo que detecta el estado de las geocercas.

cercas y otras tipos de alarmas mediante SMS sin la necesidad de utilizar la plataforma web.

#### 4. Utilización de objetos de Javascript para reportes de alarmas

Los objetos de Javascript trabajan de la misma manera que las tablas hash, lo que facilita el proceso de búsqueda de una variable en particular o crear variables con parámetros específicos. En nuestra aplicación hemos utilizado estos objetos para analizar las alarmas que suceden en el vehículo tales como SOS, exceso de velocidad, geocercas, entre otras, ya que el sistema debe ser capaz de informar el momento en el cual se genero una alarma sin redundar con información al cliente; por ejemplo, solo se notifica el estado de una alarma si es que esta se genero en ese momento con la comparación de fechas, o en el caso de geocercas se notifica solamente si el vehículo paso de estar dentro de la geocerca a fuera de la geocerca. Un extracto del código que declara las alarmas mediante objetos y reportes de geocercas se visualiza en la figura 9.

```

var alarmFlags = new Object();
alarmFlags['sos'] = "";
alarmFlags['buttonA'] = "";
alarmFlags['autoShutdown'] = "";
alarmFlags['lowBattery'] = "";
alarmFlags['overSpeed'] = "";
alarmFlags['recoverOverSpeed'] = "";

if (alarmFlags.sos!=fecha){
  alarmFlags.sos=fecha;
  alarma="El boton de panico ha sido presionado";
  escribirAlarma(alarma, fecha, longitud, latitud);
}

```

**Figura 9.** Uso de objetos de Javascript para notificar alarmas.

Como se puede observar en el algoritmo, solo en el momento en el cual las fechas de la alarma son diferentes se notifica al cliente, ya sea mediante SMS o mensajes de advertencia en la pagina web.