



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO  
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA  
ESCUELA DE INGENIERÍA EN SISTEMAS**

**“ANÁLISIS ESTADÍSTICO DE MECANISMOS DE EVENTOS ASÍNCRONOS  
BASADO EN LA WEB PARA LA INTEGRACIÓN DE LA COMUNICACIÓN EN  
TIEMPO REAL”**

**TESIS DE GRADO**

**Previo a la obtención del Título de:**

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**Presentado por:**

**MANUEL RAMIRO MORETA RODRIGUEZ**

**Riobamba – Ecuador**

**2014**

La presente tesis ha sido elaborada para completar mis estudios de Ingeniería en Sistemas Informáticos. Me gustaría agradecer a todas y cada una de las personas que de una u otra manera han aportado a realizar esta investigación, así como a quienes han aportado en mi formación tanto humana como profesional. Especialmente quisiera agradecer a mis maestros y mentores que han impartido sus conocimientos y experiencias dentro de la Escuela de Ingeniería en Sistemas de la Escuela Superior Politécnica de Chimborazo.

Cabe una mención especial de agradecimiento a Jorge Menéndez y a toda su familia por el apoyo, paciencia y enseñanzas brindadas en el tiempo que se desarrolló la presente investigación. Me gustaría también agradecer a Paúl Paguay, Narcisa Salazar y Francisco Quiroga por los aportes a esta investigación.

Con mucho afecto también quisiera agradecer a mis amigos que me ayudaron a terminar mi tesis. Finalmente quisiera extender un profundo agradecimiento a mis padres María y Wilson, por el apoyo, paciencia y formación brindada a lo largo de mi vida, sin ustedes ninguno de mis logros se hubiesen podido concretar.

## FIRMAS DE RESPONSABILIDAD Y NOTA

NOMBRES	FIRMA	FECHA
ING. GONZALO SAMANIEGO ERAZO <b>DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA</b>	_____	_____
DR. JULIO SANTILLÁN CASTILLO <b>DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS</b>	_____	_____
ING. JORGE MENÉNDEZ VERDECÍA <b>DIRECTOR DE TESIS</b>	_____	_____
ING. PAÚL PAGUAY SOXO <b>MIEMBRO DE TESIS</b>	_____	_____
<b>DIRECTOR DEL CENTRO DE DOCUMENTACIÓN</b>	_____	_____

**NOTA** \_\_\_\_\_

## **RESPONSABILIDAD DEL AUTOR**

“Yo Manuel Ramiro Moreta Rodriguez, soy responsable de las ideas, doctrinas y resultados expuestos en esta: Tesis, Tesina, Memoria y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo”.

---

Manuel Ramiro Moreta Rodriguez

## ÍNDICE DE ABREVIATURAS

<b>ANOVA</b>	Analysis of Variance
<b>API</b>	Application Programming Interface
<b>AVL</b>	Automatic Vehicle Location
<b>CERN</b>	European Organization for Nuclear Research
<b>CPU</b>	Central Processing Unit
<b>GPS</b>	Global Positioning System
<b>GPRS</b>	General Packet Radio Service
<b>GSM</b>	Global System for Mobile
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>ISO</b>	International Organization for Standardization
<b>RAM</b>	Random Access Memory
<b>TCP</b>	Transmission Control Protocol
<b>URI</b>	Uniform Resource Identifier

# ÍNDICE GENERAL

## Contenido

FIRMAS RESPONSABLES Y NOTAS

RESPONSABILIDAD DEL AUTOR

INDICE DE ABREVIATURAS

INDICE GENERAL

INDICE DE TABLAS

INDICE DE GRÁFICOS

INTRODUCCIÓN

## CAPÍTULO I

### 1. MARCO REFERENCIAL

1.1. ANTECEDENTES.....	14
1.2. JUSTIFICACIÓN.....	15
1.2.1. JUSTIFICACIÓN TEÓRICA.....	15
1.2.2. JUSTIFICACIÓN PRÁCTICA.....	16
1.3. OBJETIVOS .....	17
1.3.1. OBJETIVO GENERAL.....	17
1.3.2. OBJETIVOS ESPECÍFICOS.....	18
1.4. HIPÓTESIS .....	18

## CAPÍTULO II

## **2. MARCO TEÓRICO**

<b>2.1. LA WEB</b> .....	19
<b>2.1.1. HTML</b> .....	20
<b>2.1.2. HTTP</b> .....	21
<b>2.2. COMUNICACIÓN EN LA WEB</b> .....	22
<b>2.2.1. MECANISMOS DE COMUNICACIÓN ASÍNCRONA</b> .....	22
<b>2.2.2. FRAMEWORKS PARA LA COMUNICACIÓN EN TIEMPO REAL</b> .....	27
<b>2.2.3. SOCKET.IO</b> .....	29

## **CAPÍTULO III**

### **3. ANÁLISIS ESTADÍSTICO DE MECANISMOS DE EVENTOS ASÍNCRONOS BASADO EN LA WEB PARA LA INTEGRACIÓN DE LA COMUNICACIÓN EN TIEMPO REAL**

<b>3.1. MÉTODOS, TÉCNICAS Y PROCEDIMIENTOS</b> .....	38
<b>3.1.2. TÉCNICAS Y PROCEDIMIENTOS</b> .....	39
<b>3.2. DEFINICIÓN DE LOS PARÁMETROS A ESTUDIAR</b> .....	39
<b>3.3. INSTRUMENTOS DE MEDICIÓN</b> .....	41
<b>3.4. DESCRIPCIÓN DEL ESCENARIO DE PRUEBAS</b> .....	42
<b>3.4.1. Equipo Utilizado</b> .....	42
<b>3.4.2. Software Utilizado</b> .....	43
<b>3.5. DESARROLLO DE LAS PRUEBAS CON LOS PARÁMETROS DE MEDICIÓN/COMPARACIÓN</b> .....	43
<b>3.5.1. Muestra</b> .....	43

3.5.2. Resultados .....	44
3.6. DEMOSTRACIÓN DE LA HIPÓTESIS.....	57
<b>CAPÍTULO IV</b>	
<b>4. DESARROLLO DE LA PLATAFORMA WEB “RASTREO DIRECTO SATELITAL 2.0”</b>	
4.1. INTRODUCCIÓN .....	60
4.2. FASTNOTEQ S.A. ....	60
4.2.1. MISIÓN .....	62
4.2.2. VISIÓN .....	62
4.2.3. ESTRUCTURA.....	63
4.3. METODOLOGÍA DE DESARROLLO DE LA PLATAFORMA .....	63
4.3.1. EQUIPO DE TRABAJO .....	64
4.3.2. HISTORIAS DE USUARIO .....	64
4.3.3. BACKLOG DEL PRODUCTO .....	67
4.3.4. CICLOS DE DESARROLLO – SCRUM.....	69
<b>CONCLUSIONES</b>	
<b>RECOMENDACIONES</b>	
<b>RESUMEN</b>	
ABSTRACT	
BIBLIOGRAFÍA	
<b>ANEXOS</b>	

## ÍNDICE DE TABLAS

Tabla I : Ponderación de Parámetros a Utilizar.....	41
Tabla II : Tabla de Recursos Hardware.....	42
Tabla III: Tabla de recursos Software.....	43
Tabla IV : Estadística Descriptiva Latencia 100 Usuarios.....	45
Tabla V : Estadística Descriptiva Uso de Red 500 Usuarios.....	45
Tabla VI : Estadística Descriptiva Latencia 1000 Usuarios.....	46
Tabla VII : Análisis de Varianza Latencia 1000 Usuarios.....	46
Tabla VIII : Estadística Descriptiva Uso de Red 100 Usuarios.....	48
Tabla IX : Estadística Descriptiva Uso de Red 500 Usuarios.....	49
Tabla X : Estadística Descriptiva Uso de Red 1000 Usuarios.....	49
Tabla XI : Análisis de Varianza Uso de Red 1000 Usuarios.....	50
Tabla XII : Estadística Descriptiva Uso de RAM 100 Usuarios.....	52
Tabla XIII: Estadística Descriptiva de Uso de RAM 500 Usuarios.....	52
Tabla XIV : Estadística Descriptiva Uso de RAM 1000 Usuarios.....	52
Tabla XV : Análisis de Varianza de Uso de RAM 1000 Usuarios.....	53
Tabla XVI : Estadística Descriptiva Uso de CPU 100 Usuarios.....	55
Tabla XVII : Estadística Descriptiva Uso de CPU 500 Usuarios.....	55
Tabla XVIII : Estadística Descriptiva Uso de CPU 1000 Usuarios.....	56
Tabla XIX : Análisis de Varianza de Uso de CPU 1000 Usuarios.....	56
Tabla XX: Asignación Proporcional de Valores.....	58
Tabla XXI : Equipo SCRUM.....	64

## ÍNDICE DE ILUSTRACIONES

Figura 1 : HTTP Pollig.....	23
Figura 2 : HTTP Long Polling.....	24
Figura 3 : HTTP Streaming.....	25
Figura 4 : WebSocket.....	26
Figura 5 : Respuesta Handshake .....	32
Figura 6 : Solicitud GET XHR Polling .....	34
Figura 7 : Respuesta XHR Polling a petición GET.....	34
Figura 8 : Petición POST XHR Polling.....	34
Figura 9 : Respuesta a petición POST.....	34
Figura 10 : Petición GET JSONP Polling .....	35
Figura 11 : Respuesta a petición GET JSON Polling.....	35
Figura 12 : Petición POST JSONP Polling .....	35
Figura 13 : Solicitud GET HTMLFile .....	36
Figura 14 : Respuesta a Solicitud GET HTMLFile .....	36
Figura 15 : Solicitud POST HTMLFile .....	36
Figura 16 Latencia 1000 Usuarios .....	46
Figura 17 : Resumen de Promedios Latencia .....	47
Figura 18 Uso de Red 1000 Usuarios .....	49
Figura 19 : Resumen Uso Promedio de Red.....	50
Figura 20 Uso de RAM 1000 Usuarios.....	53
Figura 21 : Resumen Consumo Promedio de RAM .....	54
Figura 22 Uso de CPU 1000 Usuarios.....	56
Figura 23 Resumen Uso Promedio de CPU .....	57

Figura 24 : Organigrama FASTNOTEQ .....	63
Figura 25: Árbol de Objetos .....	72

## INTRODUCCIÓN

La necesidad de lograr comunicaciones en tiempo real a través de la web ha permitido desarrollar nuevas tecnologías y protocolos que conlleven a optimizar el uso de los recursos y mejores tiempos de respuesta entre el cliente y el servidor. Es una necesidad imperante en las empresas tener información en tiempo real que facilite la toma de decisiones para así tener un mayor control sobre sus procesos.

FASTNOTEQ SA es una empresa ubicada en la ciudad de Ambato dedicada al rastreo satelital de vehículos, por medio de dispositivos AVL que envían su información, a través de paquetes UDP sobre el protocolo GPRS, a los servidores dedicados a su decodificación y registro, la misma que es enviada, para su conocimiento, a los clientes, por medio de una plataforma web. El principal problema que se presenta en esta infraestructura es la no inmediatez en el despliegue de la información provocando que sus usuarios no conozcan la ubicación, estado y/o eventos de sus vehículos.

Por lo anteriormente expuesto, en la presente investigación se determina una solución de comunicación en la web que permita conocer la información requerida en tiempo real. Existen muchos mecanismos de comunicación web que logran acercarse a la comunicación en tiempo real, entre las que se encuentran los de eventos asíncronos Polling, Long Polling y WebSocket.

Este trabajo investigativo está orientado a establecer el mecanismo o mecanismos de eventos asíncronos basados en la web para la comunicación en tiempo real, que optimice el uso de recursos en cuanto a consumo de CPU, utilización de RAM, utilización de Red y latencia.

Para ello se realiza un análisis estadístico de los tres principales mecanismos de eventos asíncronos basados en la web para la integración de la comunicación en tiempo real, y así recomendar a la empresa FASNOTEQ S.A. y futuras empresas interesadas en el campo, el mecanismo óptimo que permita llevar a cabo la comunicación en tiempo real entre un servidor y un cliente web.

En el primer capítulo I se hace una revisión a los antecedentes, justificación y objetivos de la presente investigación, además de plantear la hipótesis. En el capítulo II se detalla teóricamente los conceptos involucrados o referidos en la investigación.

En el capítulo III se realiza el análisis estadístico de los mecanismos y finalmente en el capítulo IV se desarrolla la nueva plataforma web de “Rastreo Directo Satelital 2.0”

# CAPÍTULO I

## 1. MARCO REFERENCIAL

En este capítulo se presentará el marco referencial, en el cuál se encuentra la descripción de los antecedentes, la justificación del proyecto de tesis, el objetivo general y los específicos y el planteamiento de la hipótesis.

### 1.1. ANTECEDENTES

La comunicación en tiempo real permite a las empresas tener un mayor control sobre sus procesos, y así poder tomar decisiones trascendentales con mayor facilidad. A nivel mundial estas comunicaciones se han venido optimizando de manera que se busca un menor consumo de recursos y una mayor fiabilidad sobre los datos transmitidos. Así mismo se ha buscado escalar estos procesos de comunicación a la Web, de manera que se pueda acceder a la información desde cualquier navegador.

“Para permitir establecer canales de comunicación bidireccionales entre el navegador y el servidor web se utilizan varios mecanismos de eventos asíncronos como Polling, Long Polling y Websocket” (1).

La empresa FASNOTEQ S.A. ubicada en la ciudad de Ambato, se dedica a prestar servicios de Monitoreo y Rastreo Satelital de Vehículos, dichos servicios los provee a través de una página web, que permite a los clientes de dicha empresa a supervisar sus flotas de vehículos, y llevar un control de las mismas. Dicha página web está desarrollada sobre una plataforma basada en mecanismos síncronos. Donde si bien es cierto la recolección de datos es inmediata, la entrega al usuario es lenta y muy inflexible, lo que ha generado malestar en los mismos.

El propósito de este trabajo investigativo es realizar el estudio estadístico de los mecanismos de eventos asíncronos basados en la web para la comunicación en tiempo real Polling, Long Polling y Websocket, el cual permitirá seleccionar el de mejor rendimiento para desarrollar la plataforma “Rastreo Directo Satelital 2.0”.

## 1.2. JUSTIFICACIÓN

El presente trabajo surge por la necesidad que tiene la plataforma “Rastreo Directo Satelital 1.0”, en tiempo real, de conocer la ocurrencia de algún evento en los dispositivos AVL y comunicarles a sus clientes el mismo.

### 1.2.1. JUSTIFICACIÓN TEÓRICA

Las aplicaciones web son síncronas por naturaleza. El usuario interactúa con la interfaz web que se muestra en el navegador, el navegador realiza peticiones de vuelta al servidor de acuerdo a la interacción del usuario, y el servidor responde con HTML. Esto

representa un problema ya que el usuario no podrá conocer los cambios que surjan en el servidor a menos que realice una nueva petición.

“El uso de mecanismos de eventos asíncronos basados en la web Polling, Long Polling y WebSocket permiten establecer canales de comunicación bidireccionales” (2), permitiendo así que el usuario reciba cualquier cambio espontáneo que surja en el servidor, reduciendo de esta manera el esfuerzo de los clientes de FASNOTEQ S.A. para conocer los cambios de estado y posición de sus AVLS.

### 1.2.2. JUSTIFICACIÓN PRÁCTICA

La comunicación es la base del desarrollo del ser humano, a nivel empresarial, esta son la base para el buen desenvolvimiento de las actividades productivas. Pero así mismo representan un gasto, el cual debe ser optimizado en cuanto a recursos, para un mejor desarrollo. Además se debe buscar la minimización de impacto ambiental asegurando generar una tecnología verde, con un mínimo uso de ancho de banda y bajo uso de servidores, y de esta manera también aportar con el Plan del Buen Vivir.

La empresa FASNOTEQ S.A. ante el requerimiento de sus clientes de la inmediatez de la información de lo que ocurre en sus AVLS, desea proveer una plataforma web de comunicación, con respuestas que muestren en tiempo real cambios de posición, eventos y control de los AVLS.

La plataforma web “Rastreo Directo Satelital 1.0” en explotación tiene tiempos de respuesta y consumo de recursos de red muy altos, dado que cada solicitud de monitoreo desde el navegador del cliente se realiza utilizando Polling la misma que provoca una consulta en miles de registros de posiciones que se tienen para poder

informarle al usuario la posición actual de su vehículo así como la ocurrencia de algún evento. Estas peticiones son realizadas por cientos de usuarios en lapsos pequeños de tiempo lo que no permite que el usuario de forma inmediata sea notificado de la ocurrencia de algún evento en el que se necesite de su intervención.

Cada petición Polling y Long Polling abren una conexión HTTP con el servidor lo que genera varias conexiones TCP, lo cual hace que en múltiples peticiones genere una sobrecarga de la red.

“En el nuevo estándar de HTML5 propone el uso del protocolo Websocket que plantea usar una sola conexión TCP para establecer el canal de comunicación” (3), y permitir la transmisión de varias peticiones por el mismo.

Así que nace la necesidad de realizar un estudio comparativo entre los mecanismos de eventos asíncronos para la comunicación basada en la web para así determinar la que más se adapte para desarrollar la solución.

### **1.3. OBJETIVOS**

En la presente sección se desarrollarán los objetivos que se han planteado en el trabajo investigativo propuesto.

#### **1.3.1. OBJETIVO GENERAL**

Realizar el estudio estadístico de los mecanismos de eventos asíncronos Polling, Long Polling y Websocket para la comunicación basada en la web, el cual permitirá seleccionar el de mejor rendimiento para la plataforma “Rastreo Directo Satelital 2.0”.

### 1.3.2. OBJETIVOS ESPECÍFICOS

- Estudiar los mecanismos de eventos asíncronos para la comunicación basada en la web (Polling, Long Polling y WebSocket).
- Definir los parámetros de comparación y realizar el estudio estadístico de los mecanismos de eventos asíncronos para la comunicación basada en la web Polling, Long Polling y WebSocket.
- Implementar la plataforma web “Rastreo Directo Satelital 2.0” para el monitoreo satelital de automotores.

### 1.4. HIPÓTESIS

El estudio estadístico de los mecanismos de eventos asíncronos, Polling, Long Polling y WebSocket, para la comunicación basada en la web permitirá seleccionar el de mejor rendimiento para la comunicación en la plataforma web de “Rastreo Directo Satelital 2.0”.

## CAPÍTULO II

### 2. MARCO TEÓRICO

En el capítulo II se hace una recopilación conceptual de los temas relacionados con la web y los mecanismos de comunicación que en ella se utiliza.

#### 2.1. LA WEB

“La Web se basa en documentos con referencias cruzadas, que fue inventada en 1989 por un informático del CERN (Organización Europea de Investigación Nuclear) llamado Tim Berners-Lee. Era un sistema de hipertexto para compartir información basado en internet, concebido originalmente para servir como herramienta de comunicación entre los científicos nucleares del CERN. Los documentos necesitaban un formato que fuera adecuado para su misión. En aquella época casi todo el mundo utilizaba TeX y PostScript, pero éstos eran demasiado complicados teniendo en cuenta que debían ser leídos por todo tipo de computadoras, desde las terminales tontas hasta las estaciones de trabajo

gráficas X-Windows. Así tanto el lenguaje de intercambio (HTML), como el protocolo de red (HTTP) se diseñaron para ser realmente muy simples” (4).

### 2.1.1. HTML

HTML son las siglas de “Hyper Text Mar-up Language”. “Mark-up” es un término de imprenta que significa el conjunto de instrucciones estilísticas detalladas escritas en un manuscrito que debe ser tipografiado. Así, HTML podría ser traducido como “Lenguaje de Formato de Documentos para Hipertexto”. HTML es una aplicación de SGML, un lenguaje muy general para definir lenguajes de formato de documentos.

#### *Historia del HTML*

Según Luque Rubio(2007), a principios de 1990, Tim Berners-Lee define por fin el HTML como un subconjunto del conocido SGML. En 1991 Tim Berners-Lee crea el primer navegador de HTML que funcionaría en modo texto y para UNIX.

Los trabajos para crear un sucesor de HTML, posteriormente llamado ‘HTML+’, comenzaron a finales de 1993. El HTML+ se diseñó originalmente para ser un superconjunto del HTML que permitiera evolucionar gradualmente desde el formato HTML anterior. A la primera especificación formal de HTML+ se le dio, por lo tanto el número de versión 2.0 para distinguirla de esos “estándares no oficiales” previos. Los trabajos sobre HTML+ continuaron, pero nunca se convirtió en un estándar.

El borrador del estándar HTML 3.0 fue propuesto por el recién formado W3C en marzo de 1995. Con él se introdujeron muchas nuevas capacidades, tales como facilidades de crear tablas, hacer que el texto flúyase alrededor de las figuras y mostrar elementos matemáticos complejos. Aunque se diseñó para ser compatible con HTML 2.0, era

demasiado complejo ser implementado con la tecnología de la época y, cuando el borrador del estándar expiró en septiembre de 1995, se abandonó debido a la carencia de apoyos de los fabricantes de navegadores web. El HTML 3.1 nunca llegó a ser propuesto oficialmente, y el estándar siguiente fue el HTML 3.2, que abandonaba la mayoría de las nuevas características del HTML 3.0 y, a cambio, adoptaba muchos elementos desarrollados inicialmente por los navegadores web Netscape y Mosaic. La posibilidad de trabajar con fórmulas matemáticas se había propuesto en el HTML 3.0 pasó a quedar integrada en un estándar distinto llamado MathML.

El HTML 4.0 también adoptó muchos elementos específicos desarrollados inicialmente para un navegador web concreto, pero al mismo tiempo comenzó a limpiar el HTML señalando algunos de ellos como ‘desaprobados’.

En el año 2004, un grupo llamado WHATWG y formado por Apple, Mozilla, Opera, se forma y comienza el desarrollo de un HTML más cercano a la gente. El W3C rectifica, y se une a WHATWG para desarrollar HTML5.

### 2.1.2. HTTP

“HTTP (Hypertext Transfer Protocol) Protocolo de Transferencia de Hipertexto, es un protocolo a nivel de aplicación con la ligereza y velocidad necesaria para sistemas de información distribuidos colaborativos, hipermedia. Se trata de un protocolo genérico, sin estado, orientado a objetos que puede ser usado para muchas tareas, tales como servidores de nombres y sistemas de gestión de objetos distribuidos, a través de la extensión de sus métodos de petición (comandos). Una característica de HTTP es la

tipificación de representación de datos, permitiendo que los sistemas que se construyan de forma independiente de los datos que se transfieren” (5).

## 2.2. COMUNICACIÓN EN LA WEB

La comunicación en la web “utiliza protocolos basados en un modelo cliente-servidor síncrono: uno o varios clientes realizan una conexión hacia el servidor y le transmiten una acción a realizar, éste la procesa y les retorna la respuesta, cerrándose la conexión de manera inmediata” (6).

“Internet se ha creado en gran parte a partir del llamado paradigma solicitud/respuesta de HTTP. Un cliente carga una página web y no ocurre nada hasta que el usuario hace clic en la página siguiente” (7).

De los conceptos anteriormente mencionados tomamos la definición que la comunicación en la web se basa en un cliente que hace solicitudes a un servidor, el cuál cerrará la conexión una vez entregada la respuesta.

Dado que la web funciona básicamente con la interacción entre el cliente y servidor, siendo el cliente quién siempre solicita la información al servidor estableciendo una conexión por cada solicitud, no hay manera que el servidor sea quién entregue la información a los clientes sin que ellos la hayan solicitado. Para lograr que el servidor notifique a los clientes se desarrollaron mecanismos de comunicación asíncrona como son “Polling, Long Polling, HTTP Streaming, Websocket, entre otros” (8).

### 2.2.1. MECANISMOS DE COMUNICACIÓN ASÍNCRONA

En el intento de desarrollar una comunicación en tiempo real en la web, se han desarrollado mecanismos de comunicación asíncrona, que inicialmente establecían

conexiones continuas con el servidor para conocer cambios en el mismo, hasta el desarrollo de WebSocket que establece canales de comunicación. A continuación se explorará los principales mecanismos:

### *Polling (HTTP Polling)*

Por naturaleza del protocolo HTTP no permite el envío de respuestas asíncronas al cliente, para lograrlo se usa un mecanismo de consulta, también llamado HTTP Polling.

“El HTTP Polling envía peticiones a intervalos regulares de tiempo, con lo que el sistema tiene continuas oportunidades de actualizar la presentación. Esta técnica que se ve en la **Figura 1**, no es ideal, ya que no existe un intervalo de consulta ideal. Siempre hay un balance que mantener entre la frecuencia de actualización y la verbosidad del sistema. Como se puede ver, es posible que ocurran múltiples eventos entre cada consulta, pero también es posible que no ocurran eventos. En un análisis final, el mecanismo de consultas no es un mecanismo asíncrono real” (8).

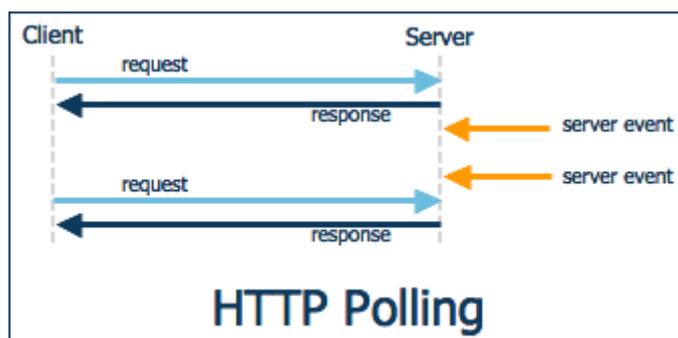


Figura 1 : HTTP Polling

**Fuente:** <http://www.theserverside.com/news/1363576/What-is-the-Asynchronous-Web-and-How-is-it-Revolutionary>

“El navegador emite periódicamente una llamada XMLHttpRequest para obtener nueva información, por ejemplo, cada cinco segundos. La solución hace uso de la capacidad de programación de eventos del navegador para proporcionar un medio que mantenga informado de los últimos cambios al usuario” (9).

El HTTP Polling realiza peticiones HTTP en intervalos de tiempo regulares, para mantener actualizado de los cambios a los clientes.

### *HTTP Long Polling*

“El HTTP Long Polling o consulta larga realiza una petición por adelantado para una respuesta futura, y esta respuesta se bloquea hasta que ocurra un evento que la dispare. Este mecanismo, como se puede visualizar en la **Figura 2**, es muy eficiente y es totalmente compatible con las configuraciones de proxy/firewall ya que resulta indistinguible de un servidor que responde lento” (8).

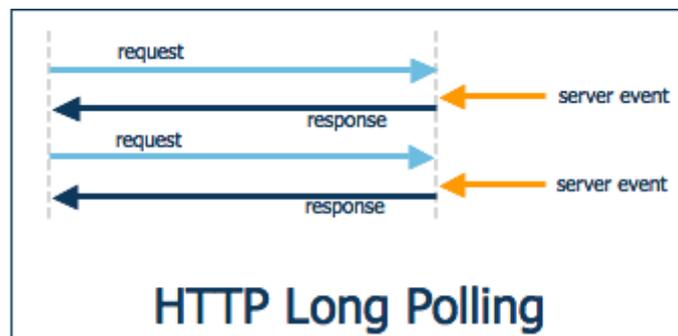


Figura 2 : HTTP Long Polling

**Fuente:** <http://www.theserverside.com/news/1363576/What-is-the-Asynchronous-Web-and-How-is-it-Revolutionary>

“El servidor intenta ‘mantener abierta’ (no responder inmediatamente a) cada solicitud HTTP, respondiendo sólo cuando hay eventos para enviar. De esta manera, siempre hay

una solicitud pendiente para que el servidor pueda responder a efectos que pueda entregar eventos a medida que estos ocurran, minimizando de este modo la latencia en la entrega de mensajes” (10).

HTTP Long Polling es un mecanismo que también hace solicitudes regulares, pero a diferencia de la anterior, este mecanismo trata de mantener abierta la conexión hasta que haya una nueva actualización.

### *HTTP Streaming*

“El servidor envía múltiples respuestas a una única petición, como se muestra en la **Figura 3**. Este es un mecanismo eficiente, pero desafortunadamente no funciona en todas las configuraciones de proxy/firewall que existen, lo cual lo hace inconveniente para despliegues de propósito general” (8).

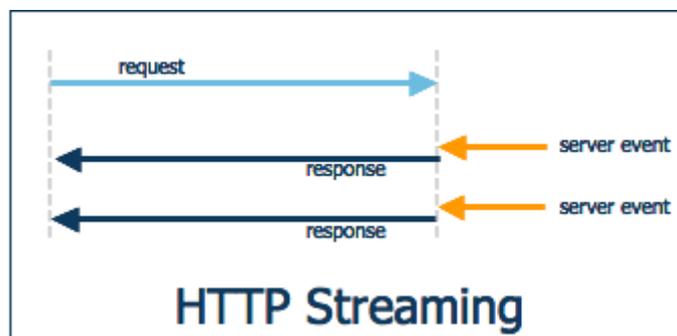


Figura 3 : HTTP Streaming

**Fuente:** <http://www.theserverside.com/news/1363576/What-is-the-Asynchronous-Web-and-How-is-it-Revolutionary>

“El mecanismo HTTP Streaming mantiene una solicitud HTTP abierta de forma indefinida. Nunca termina la solicitud o cierra la conexión, incluso después de que el servidor envía los datos al cliente. Este mecanismo reduce significativamente la latencia

en la red debido a que el cliente y el servidor no necesitan abrir y cerrar la conexión” (10).

### *WebSocket*

“WebSocket ofrece una canal de comunicación full-duplex bidireccional que opera a través de un solo conector a través de internet y puede ayudar a construir aplicaciones escalables y en tiempo real”. (11)

“WebSocket es un protocolo de comunicación bidireccional basado en un única conexión TCP, que permite transmisión simultánea de datos en ambas direcciones entre el cliente y el servidor. Mediante la utilización de tecnología WebSocket, se crea una conexión persistente entre el cliente y el servidor que reducen significativamente los mensajes intercambiados mediante la eliminación de solicitudes HTTP innecesarias y excesivos encabezados” (12).

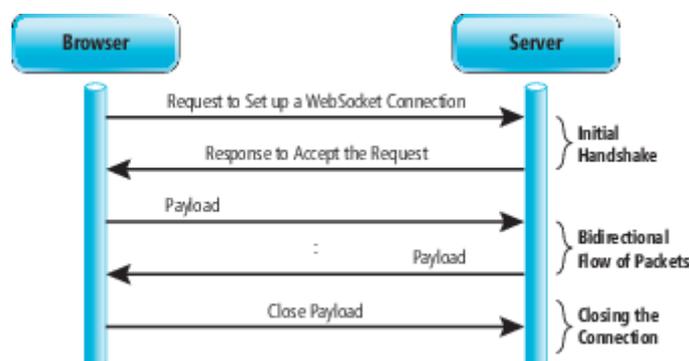


Figura 4 : WebSocket

**Fuente:** <http://www.theserverside.com/news/1363576/What-is-the-Asynchronous-Web-and-How-is-it-Revolutionary>

De los mecanismos revisados en esta investigación se van a analizar Polling, Long Polling y WebSocket, Polling porque es el que se está utilizando actualmente, Long Polling como

una mejora al mecanismo usado y Websocket como una nueva alternativa a la implementación.

### 2.2.2. FRAMEWORKS PARA LA COMUNICACIÓN EN TIEMPO REAL

Un Framework “es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación” (13)

Según Mayol Ramis (2004) los Frameworks son puentes que ayudan a desarrollar aplicaciones más fácil y más rápidamente, agregando capas abstractas que se encargan de funciones internas y complementarias. El objetivo de estos es aliviar la sobrecarga asociada con las actividades comunes realizadas en el desarrollo, la promoción de reutilización de código. Al igual que una tecnología, un Framework tiene ventajas y desventajas a considerar antes de tomar la decisión de iniciar a utilizar uno:

- Eficiencia: usando funciones pre-construidas o clases que ahorran miles de líneas de código y que ya están probadas y optimizadas.
- Costo: casi todos los Frameworks son open-source pero algunos de ellos están sujetos a licenciamiento.
- Soporte: esto es muy importante para usar un Framework con fuerte y amplia aceptación por la comunidad de desarrolladores, así como fóruns de comunidades son la mejor fuente de documentación de los mismos. (Especialmente en open-source)
- Limitaciones: el comportamiento del código de los Frameworks no puede ser modificado, lo que significa que se ven obligados a respetar sus limitaciones y trabajar de la manera que se requiera.

- **Código Público:** como están disponibles para todo mundo, también está disponible para personas que quieran encontrar y explotar vulnerabilidades del mismo.

Para esta investigación un Framework es un conjunto de librerías y componentes que facilitan el desarrollo de aplicaciones.

En el mercado existen varios Frameworks para la comunicación en tiempo real, de ellos listaremos los de mayor uso y características que sirvan en la investigación:

### *Xsockets*

Xsockets es una librería de comunicación en tiempo real construida en Tecnología Microsoft .NET y provee APIs para el cliente y el servidor.

### *SignalR*

SignalR es una librería escrita en Tecnología Microsoft .NET que simplifica las comunicaciones en tiempo real para servicios web proveyendo dos APIs para las capas de comunicación: Conexiones persistentes y Hub. La primera provee acceso a capa de bajo nivel, con esto logra una comunicación bidireccional abstracta y persistente entre el cliente y el servidor (permitiendo enviar datos crudos). La segunda provee acceso a una alta y más abstracta capa, que provee serialización y llamada de procedimientos remotos en ambas direcciones.

### *Socket.IO*

Socket.IO es una librería JavaScript para aplicaciones web en tiempo real usando WebSocket como protocolo preferido de transporte y con otras múltiples tecnologías antiguas de transporte.

En el lado del servidor usa NodeJS que es una plataforma software escrita en JavaScript que habilita correr un servidor web sin el uso de software externo como Apache, que le da más control de cómo funciona el servidor web.

Ésta librería es una de las más usadas para aplicaciones web en tiempo real con un extensa documentación y soporte de la comunidad de desarrolladores.

De los Frameworks mencionados en el apartado anterior podemos destacar que Socket.IO está disponible para Windows, Linux y Mac y que tiene la implementación de varios mecanismos de comunicación asíncrona, además tiene optimizado para establecer canales con el mecanismo que mejor se adapte al escenario sobre el cuál se ejecute (navegador utilizado).

Fastnoteq S.A. tiene como nicho de mercado el sector público, y este último pondera los productos desarrollados con Software Libre por sobre los de Software propietario, y al estar Socket.io también disponible para Linux y ser de software libre, se eligió analizar este Framework para su uso en esta investigación.

### 2.2.3. SOCKET.IO

“Socket.IO pretende llevar una API de Websocket para muchos navegadores y dispositivos, con algunas características específicas para ayudar a la creación de juegos y aplicaciones en tiempo real.

- Soporte de múltiples transportes (viejos agentes de usuario, navegadores móviles, etc.).
- Múltiples sockets bajo una misma conexión (namespaces).
- Detección de desconexión a través de hearbeats.

- Soporte de reconexión con buffering (ideal para dispositivos móviles y redes lentas.
- Protocolo ligero que se asienta encima de HTTP.” (14)

### *Anatomía de un socket en Socket.IO*

Para realizar el proceso de establecer una conexión con Socket.IO se utiliza un socket, que se define luego de seguir el siguiente proceso:

Según Rauch (2014) un cliente Socket.IO primero decide el transporte a utilizar para conectarse.

El estado de un socket de Socket.IO puede ser: *disconnected*, *disconnecting*, *connected* y *connecting*.

Una conexión puede tener los siguientes estados: *closed*, *closing*, *open* y *opening*.

Se da un simple HTTP handshake al inicio de una conexión Socket.IO. El handshake, si tiene éxito, el cliente recibirá:

- Un identificador de sesión que se le acreditará al transporte al abrir las conexiones.
- Un número de segundos que se tiene que esperar para que se dé un handshake (*heartbeat timeout*).
- Un número de segundos que define el cierre de la conexión, cuando el socket es considerado desconectado si la conexión no puede ser reabierta (*close timeout*).

En este punto el socket se considera conectado y el transporte es marcado para abrir la conexión.

Si la conexión está cerrada, ambos extremos guardan los mensajes y luego los arma adecuadamente para ser enviados como un lote cuando la conexión se reanuda.

Si la conexión no se reanuda en el tiempo de espera negociado el socket se considera desconectado. En este punto, el cliente puede decidir volverse a conectar, lo que implica un nuevo handshake.

### *Handshake*

Según Rauch (2014) el cliente de Socket.IO deberá realizar una solicitud HTTP POST inicial como la siguiente:

<http://example.com/socket.io/1/>

La ausencia de los segmentos *transport id* y *session id* señalarán al servidor que se trata de un nueva conexión.

El servidor puede responder en tres diferentes maneras:

- 401 Unauthorized

Si el servidor rechaza autorizar al cliente conectarse, basado en la información provista (ej. Cookies, consultas personalizadas)

- 503 Service Unavailable

Si el servidor rechaza la conexión por cualquier razón (ej. Sobrecarga)

No es requerido cuerpo en la respuesta.

- 200 OK

Si el handshake fue exitoso

El cuerpo de la respuesta debería contener el identificador de sesión (*sid*) dado al cliente, seguido por el *heartbeat timeout*, el *close timeout* y la lista de transportes soportados separados por dos puntos (:).

La ausencia de un *heartbeat timeout* es interpretado como que el cliente y servidor no esperarán *heartbeats*.

Una respuesta de handshake se vería como se muestra en la **Figura 5**

```
HTTP/1.1 200 OK
Content-Type: text/plain
Connection: keep-alive
Content-Length: 73

1234567890:15:25:websocket,flashsocket,htmlfile,xhr-polling,jsonp-polling
```

**Figura 5 : Respuesta Handshake**

**Fuente:** <https://github.com/learnboost/socket.io-spec>

### *Transporte*

Según Rauch (2014) una vez que el ciclo de petición-respuesta del handshake es completado (y este termina exitosamente), una nueva conexión es abierta por el transporte que fue negociado, con la petición HTTP GET.

El transporte puede modificar la URI si el transporte lo requiere, siempre y cuando no se pierda ninguna información.

La URI contiene toda la información requerida por Socket.IO para continuar el intercambio de mensajes (protocolo, seguridad, espacio de nombres, versión del protocolo, transporte, etc.).

Los mensajes pueden ser enviados y recibidos siguiendo esta convención. Como los mensajes de codifican y enmarcan depende de cada transporte, pero por lo general se reduce a si el transporte se ha incorporado en la construcción del mismo. Puede haber transportes unidireccionales y/o bidireccionales. En el caso de transportes unidireccionales (done el servidor puede escribir en el cliente, pero no a la inversa), deben realizar peticiones POST para enviar datos de vuelta al servidor al mismo punto final URI.

Socket.IO tiene implementado los siguientes transportes:

- WebSocket
- Flashsocket
- XHR Polling
- Htmlfile
- JSONP Polling

WebSocket y Flashsocket usan el protocolo WebSocket para realizar los handshakes y transportar los paquetes, esta comunicación es bidireccional.

XHR Polling, Htmlfile y JSONP Polling son transportes unidireccionales, lo que significa que se usará un canal para enviar mensajes y otro canal para recibirlos. Además el canal del servidor es una solicitud GET con respuesta tardía (XHR Polling y JSON Polling) o una respuesta htmlfile y el canal del cliente es una secuencia de peticiones POST con respuestas simples de confirmación.

### **XHR Polling**

Cuando una conexión XHR Polling es establecida una petición GET se hace a la espera de una respuesta retardada del servidor como se ve en la **Figura 6**.

```
GET /socket.io/1/xhr-polling/15725592722050266739?t=1312200210467 HTTP/1.1
Host: 127.0.0.1:5000
Connection: Keep-Alive
```

**Figura 6 : Solicitud GET XHR Polling**

**Fuente:** <https://github.com/learnboost/socket.io-spec>

Cuando el servidor quiere enviar un mensaje, este responde de la siguiente manera:

```
HTTP/1.1 200 OK
Content-Type: text/plain
Content-Length: 47

5:::{"args":[{"vti":"vti"}],"name":"nicknames"}
```

**Figura 7 : Respuesta XHR Polling a petición GET**

**Fuente:** <https://github.com/learnboost/socket.io-spec>

Cuando el cliente quiere enviar un mensaje, este crea una petición POST como se ve en la **Figura 8**.

```
POST /socket.io/1/xhr-polling/15725592722050266739?t=1312200250539 HTTP/1.1
Host: 127.0.0.1:5000
Content-Length: 45
Connection: Keep-Alive

5:::{"name":"user message","args":["hello!"]}
```

**Figura 8 : Petición POST XHR Polling**

**Fuente:** <https://github.com/learnboost/socket.io-spec>

Y debe recibir una respuesta 200 del servidor como se muestra en la **Figura 9**

```
HTTP/1.0 200 OK
Content-Length: 1

1
```

**Figura 9 : Respuesta a petición POST**

**Fuente:** <https://github.com/learnboost/socket.io-spec>

## JSONP Polling

JSONP Polling es similar a XHR Polling. Cuando el canal del servidor es creado es como una petición GET como se visualiza en la **Figura 10** que esperará un respuesta retardada.

```
GET /socket.io/1/jsonp-polling/4168417008084729/?t=1312874409449&i=0 HTTP/1.1
Host: localhost:5000
Connection: keep-alive
```

**Figura 10 : Petición GET JSONP Polling**

**Fuente:** <https://github.com/learnboost/socket.io-spec>

Cuando el servidor quiere enviar un mensaje este responde como se muestra en la **Figura 11** y cierra la conexión.

```
HTTP/1.1 200 OK^M
Content-Type: text/javascript; charset=UTF-8
Content-Length: 69

io.j[0]("5:::{\"args\":[{\\"vti\\":\\"vti\\"}],\"name\\":\\"nicknames\\"});
```

**Figura 11 : Respuesta a petición GET JSON Polling**

**Fuente:** <https://github.com/learnboost/socket.io-spec>

Cuando el cliente quiere enviar un mensaje este crea una petición POST como se ve en la **Figura 12**, y el cliente debería recibir una simple respuesta 200 del servidor.

```
POST /socket.io/1/jsonp-polling/4168417008084729?t=1312874410509&i=0 HTTP/1.1
Host: localhost:5000
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 85

d=5%3A%3A%3A%7B%22name%22%3A%22user+message%22%2C%22args%22%3A%5B%22hello%21%22%5D%7D
```

**Figura 12 : Petición POST JSONP Polling**

**Fuente:** <https://github.com/learnboost/socket.io-spec>

## Htmlfile

Htmlfile es un transporte para Internet Explorer. Cuando el canal del servidor es creado es como una petición GET como puede visualizar en la **Figura 13** que esperará un respuesta.

```
GET /socket.io/1/htmlfile/4168417008084729/?t=1312874409449 HTTP/1.1
Host: localhost:5000
Connection: keep-alive
```

**Figura 13 : Solicitud GET HTMLFile**

**Fuente:** <https://github.com/learnboost/socket.io-spec>

El servidor responde una respuesta fragmentada como se muestra en la **Figura 14**

```
HTTP/1.1 200 OK
Content-Type: text/html
Connection: keep-alive
Transfer-Encoding: chunked

100
<html><body><script>var _ = function (msg) { parent.s._(msg, document); };</script>
```

**Figura 14 : Respuesta a Solicitud GET HTMLFile**

**Fuente:** <https://github.com/learnboost/socket.io-spec>

Cuando el cliente quiere enviar un mensaje este hace una petición POST como se indica en la **Figura 15** y el cliente debería recibir una simple respuesta 200 del servidor.

```
POST /socket.io/1/xhr-polling/15725592722050266739?t=1312200250539 HTTP/1.1
Host: 127.0.0.1:5000
Content-Length: 45
Connection: Keep-Alive

5:::{"name":"user message","args":["hello!"]}
```

**Figura 15 : Solicitud POST HTMLFile**

**Fuente:** <https://github.com/learnboost/socket.io-spec>

Una vez que se ha profundizado en la teoría de los mecanismos a analizar como son Polling, Long Polling y Websocket, se procedió a realizar las pruebas necesarias para poder demostrar la hipótesis.

## CAPÍTULO III

### 3. ANÁLISIS ESTADÍSTICO DE MECANISMOS DE EVENTOS ASÍNCRONOS BASADO EN LA WEB PARA LA INTEGRACIÓN DE LA COMUNICACIÓN EN TIEMPO REAL

En el presente capítulo se desarrolla el análisis estadístico de los mecanismos de eventos asíncronos basado en la web para la integración de la comunicación en tiempo real, definiendo los métodos, técnicas, procedimientos, mecanismos de eventos asíncronos y los parámetros de análisis.

#### 3.1. MÉTODOS, TÉCNICAS Y PROCEDIMIENTOS

En el desarrollo del presente proyecto de investigación se utilizó el método científico, técnicas y procedimientos que permitieron la ejecución de la misma, los cuales se detallan a continuación.

##### **3.1.1. Métodos**

La presente investigación se estableció como marco principal al método científico experimental, que de forma general es:

- Plantear el problema
- Describir la hipótesis
- Recopilar información para la comprobación de la hipótesis
- Analizar y difundir los resultados.

### 3.1.2. TÉCNICAS Y PROCEDIMIENTOS

Para la comprobación de la hipótesis planteada en esta investigación: “El estudio estadístico de los mecanismos de eventos asíncronos, Polling, Long Polling y WebSocket, para la comunicación basada en la web permitirá seleccionar el de mejor rendimiento para la comunicación en la plataforma web de ‘Rastreo Directo Satelital 2.0’”, se especificó un tamaño de población al cuál va a ser expuesto el experimento, que con estadística descriptiva e inferencial permitió alcanzar los objetivos planteados.

#### **Población**

Para desarrollar las pruebas de la investigación se determinó analizar los mecanismos en el escenario al que va a llegar la empresa FastNoteq S.A. que son 1000 usuarios conectados al mismo tiempo. Para medir el comportamiento actual y una proyección de comportamiento se analizará con 100 y 500 usuarios. En este escenario se enviaron 10 mensajes continuos y secuenciales cada 10 segundos

### 3.2. DEFINICIÓN DE LOS PARÁMETROS A ESTUDIAR

Según la ISO2015, según Rodríguez (2010) para medir el rendimiento (eficiencia) de un producto software se debe evaluar las siguientes condiciones:

- Comportamiento Temporal, que es la capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados bajo condiciones determinadas.
- Utilización de Recursos, capacidad del producto software de usar las cantidades y tipos de recursos adecuados cuando el software lleve a cabo su función bajo condiciones determinadas.
- Cumplimiento de la eficiencia, capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia.”

De lo descrito en los párrafos anteriores los parámetros que se evaluarán en este estudio son:

- **Latencia**, el objetivo de este parámetro es determinar el tiempo que tarda en ser entregado un paquete. Para nuestro experimento el mecanismo de menor latencia será el ganador, debido a que la experiencia de usuario se determina que mientras más rápido se entrega la información, mejor experiencia tendrá el usuario, además que el caso práctico de este tema de investigación es un sistema de monitoreo en tiempo real, el cual requiere tiempos mínimos de entrega de información.
- **Uso de la Red**, el objetivo de este parámetro es determinar el tamaño de los paquetes.
- **Uso del CPU**, el objetivo de ese parámetro es conocer la incidencia en el CPU el procesamiento de los mensajes en el servidor.

- **Uso de RAM**, el objetivo de este parámetro es conocer la incidencia de memoria en el procesamiento de mensajes en el servidor.

Concluimos que estos son los parámetros a analizar estadísticamente dentro de la investigación, y en base a la importancia de los mismos se ha decidido establecer los siguientes pesos a los parámetros a analizar:

**Tabla I : Ponderación de Parámetros a Utilizar**

<b>Parámetro</b>	<b>Ponderación</b>
Latencia	50
Uso de la Red	30
Uso de RAM	10
Uso del CPU	10
<b>Total</b>	<b>100</b>

Los valores definidos en la **Tabla I** están tomados de acuerdo a la importancia que posee cada parámetro dentro de la implementación de la comunicación en tiempo real. Para el caso en el que se está estudiando tiene mayor importancia el uso de la red y la latencia, ya que se tiene un alto número de usuarios, con un número de vehículos que supera el millar en total, lo que genera un alto tráfico dentro de la red.

Una vez que se ha definido la ponderación de los parámetros, se procederá al análisis estadístico de cada uno de estos para lo cual se hará uso de la Estadística Inferencial.

### 3.3. INSTRUMENTOS DE MEDICIÓN

Los instrumentos de medición que se utilizaron son los siguientes:

**WireShark**, antes conocido como Ethereal, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos y como una herramienta didáctica.

**Sar (Comando Linux)**, El nombre sar proviene de las siglas “system activity report” (Informe de actividad del sistema). En Linux se encuentra normalmente en el paquete sysstat. El paquete sysstat incluye programas y scripts para recopilar y mostrar información sobre el rendimiento del sistema, generado informes detallados. Este conjunto de programas puede resultar de mucha utilidad a la hora de detectar cuellos de botellas y para hacernos una idea de cómo se utiliza el sistema a lo largo del día.

### 3.4. DESCRIPCIÓN DEL ESCENARIO DE PRUEBAS

Para efectuar las pruebas de los mecanismos de eventos asíncronos se estableció el siguiente escenario, que derivan de las características con la que cuenta la empresa FASNOTEQ S.A.

#### 3.4.1. Equipo Utilizado

Para la realización de pruebas se usó una PC de escritorio que dispone de un procesador Intel(R) Core(TM)2 Quad de 2.4 GHz. La memoria RAM es de 2 GB. Con un disco duro Samsung Serial ATA de 250 GB. La red sobre la cual se desplegó las pruebas es sobre un Poi Gigabit Ethernet Broadcom Netlink (NIC 10/100/100) modem v9.2 de 56 k.

Tabla II : Tabla de Recursos Hardware

Hardware	Características
Procesador	Intel(R) Core(TM)2 Quad de 2.4 GHz
Memoria	Memoria DDR2 SDRAM 2 GB
Disco Duro	SATA de 250 GB de 5400 rpm
Comunicaciones	Controlador PCI Gigabit Ethernet Broadcom NetLink (NIC 10/100/1000)

### 3.4.2. Software Utilizado

El software utilizado en las pruebas fue:

En el servidor se usó sistema operativo CentOS 6.2, en el cuál se instaló el Frameworks Node.JS en su versión 0.10.2, además se instaló el conjunto de librerías Socket.IO en su versión 0.9.

Tabla III: Tabla de recursos Software

Características	Descripción
Sistema Operativo	CentOS 6
NodeJS	V0.10.2
Socket.IO	V0.9

### 3.5. DESARROLLO DE LAS PRUEBAS CON LOS PARÁMETROS DE MEDICIÓN/COMPARACIÓN

Una vez que se ha definido el escenario sobre el cuál se realizaron las pruebas se procedió a tomar las muestras en diferentes escenarios simulados para su posterior análisis.

#### 3.5.1. Muestra

El tamaño total de la muestra para los casos de análisis de uso de recursos es 111, ya que se evalúa 111 segundos el envío de los 10 mensajes para los respectivos escenarios. Para el caso de la latencia se tomará como tamaño de la muestra el resultado de

multiplicar el número de usuarios por el número de mensajes a ser enviados en los diferentes escenarios.

### 3.5.2. Resultados

Una vez definido el escenario se procedió a realizar las pruebas y evaluados los indicadores establecidos anteriormente. Para la evaluación de los indicadores se aplicó ANOVA.

#### *Latencia*

Para evaluar la hipótesis en la latencia que tienen los paquetes al ser enviados con los diferentes mecanismos de comunicación, se ha definido las siguientes variables independientes:

$\mu_1 = \text{Latencia con el mecanismo Polling}$

$\mu_2 = \text{Latencia con el mecanismo Long Polling}$

$\mu_3 = \text{Latencia con el mecanismo WebSocket}$

Una vez que se tiene definidas las variables se procede a definir la hipótesis:

**Hipótesis Nula:** la latencia del mecanismo Polling es menor que el mecanismo Long Polling y este también es menor que WebSocket.

$H_0: \mu_1 = \mu_2 = \mu_3$

**Hipótesis de Investigación:** la latencia del mecanismo Polling es mayor que el mecanismo Long Polling y este a su vez mayor que WebSocket.

$H_1: \mu_1 \neq \mu_2 \neq \mu_3$

Nivel de Confianza: 5%

Con una concurrencia de 100 usuarios conectados al prototipo se obtuvo un conjunto de datos de la latencia de los paquetes, a través del uso de una característica de Socket.io que se implementó para las pruebas, los cuales luego con la ayuda de Microsoft Excel se hizo los cálculos para obtener la estadística descriptiva, los que se muestran en la **Tabla V**, obteniendo una latencia promedio de 126,046 milisegundos con el mecanismo Polling, 60,866 milisegundos con Long Polling y 39,866 con Websocket:

**Tabla IV : Estadística Descriptiva Latencia 100 Usuarios**

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
Polling	111	126046	126,046	1,407291291
Long Polling	111	60337	60,337	7,350781782
Websocket	111	39866	39,866	1,215259259

Con una concurrencia de 500 usuarios se obtuvo la siguiente información, que con la estadística descriptiva descrita en la **Tabla VII**, donde se visualiza que se obtuvo una latencia promedio de 134,029 milisegundos con Polling, 66,581 milisegundos con Long Polling y 42,25 milisegundos con Websocket:

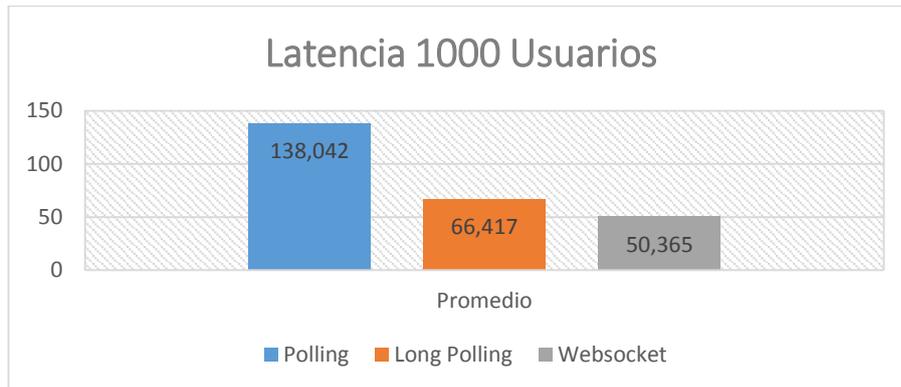
**Tabla V : Estadística Descriptiva Uso de Red 500 Usuarios**

<i>Grupos</i>	<i>Promedio</i>	<i>Varianza</i>
Long Polling	134,029	1,429588589
Polling	66,581	1,95139039
Websocket	42,25	2,12962963

Con una concurrencia de 1000 usuarios conectados al prototipo se recolectaron datos que al ser procesados en Excel se obtuvo el siguiente resumen mostrado en la **Tabla IX**.

**Tabla VI : Estadística Descriptiva Latencia 1000 Usuarios**

<i>Grupos</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
Polling	138042	138,042	1,489725726
Long Polling	66417	66,417	2,531642643
Websocket	50365	50,365	3,669444444



**Figura 16 Latencia 1000 Usuarios**

**Fuente:** Manuel Moreta

Con los datos de la **Tabla IX**, se procedió a aplicar Análisis de Varianza obteniendo los siguientes resultados:

**Tabla VII : Análisis de Varianza Latencia 1000 Usuarios**

<i>Origen de las variaciones</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	2179177,276	850044,3306	0	2,99872874
Dentro de los grupos	2,563604271			
Total				

De los resultados anteriormente mencionados se observa que el mecanismo de menor latencia es Websocket, además se puede observar que independiente del número de usuarios conectados se mantiene la tendencia de Websocket ser mejor que Long Polling y éste a su vez mejor que Polling.

Estadísticamente se puede analizar con un nivel de significancia del 5% que el valor de probabilidad que nos proporciona ANOVA es 0 en el escenario en el cual se aplicó el experimento, lo cual nos permite deducir que no existe probabilidad alguna que se apruebe la hipótesis nula, por lo que se acepta la hipótesis de investigación.

Dentro de la investigación además se realizó pruebas con menor cantidad de usuarios, donde como vemos en la Figura 17, conforme aumenta el número de usuarios el mecanismo WebSocket mantiene un crecimiento uniforme.

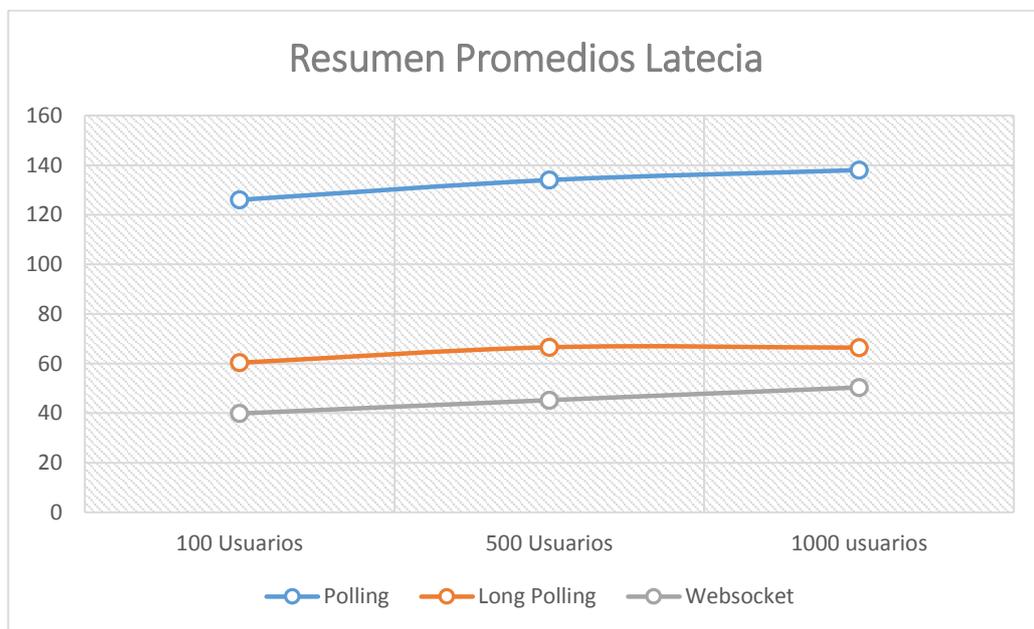


Figura 17 : Resumen de Promedios Latencia

Fuente: Manuel Moreta

### *Uso de la Red*

Para evaluar la hipótesis en el uso de la Red, se definen las variables independientes:

$\mu_1 =$  El uso de la Red con el mecanismo Polling

$\mu_2 =$  El uso de la Red con el mecanismo Long Polling

$\mu_3 =$  El uso de la Red con el mecanismo Websocket

Una vez que se tiene definidas las variables se procede a definir la hipótesis:

**Hipótesis Nula:** El uso de red del mecanismo Polling es menor que el mecanismo Long Polling y este también es menor que Websocket.

$$H_0: \mu_1 \leq \mu_2 \leq \mu_3$$

**Hipótesis de Investigación:** El uso de la red del mecanismo Polling es mayor que el mecanismo Long Polling y este a su vez mayor que Websocket.

$$H_0: \mu_1 > \mu_2 > \mu_3$$

Nivel de Confianza: 5%

Con una concurrencia de 100 usuarios conectados al prototipo se obtuvo un conjunto de datos de la cantidad de bytes transmitidos por segundo, a través de la herramienta **sar**, los cuales luego con la ayuda de Microsoft Excel se hizo los cálculos para obtener la estadística descriptiva, los que se muestran en la **Tabla IV**, donde se obtuvo una transmisión promedio de 3,96 bytes/segundo con el mecanismo Polling, 3,96 bytes/segundo con el mecanismo Long Polling y 2,34 bytes/segundo evaluado en 111 segundos:

Tabla VIII : Estadística Descriptiva Uso de Red 100 Usuarios

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
Polling	111	424,35	3,96588785	109,660198
Long Polling	111	424,35	3,96588785	109,660198
Websocket	111	251,28	2,348411215	56,9519625

Con una concurrencia de 500 usuarios se obtuvo la siguiente información, que con la estadística descriptiva descrita en la **Tabla XII**, donde se visualiza que se obtuvo una transmisión promedio de 11,55 bytes/segundo con el mecanismo Polling, 11,53 bytes/segundo con Long Polling y 6,56 bytes/segundo con WebSocket:

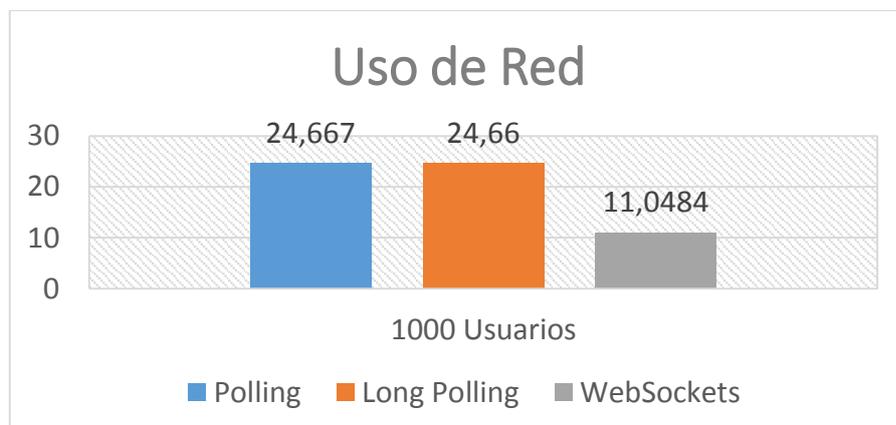
**Tabla IX : Estadística Descriptiva Uso de Red 500 Usuarios**

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
Polling	111	1235,9	11,55046729	866,479093
Long Polling	111	1233,78	11,53065421	865,604195
WebSocket	111	702,16	6,562242991	430,647635

Con una concurrencia de 1000 usuarios conectados al prototipo se recolectaron datos que al ser procesados en Excel se obtuvo el siguiente resumen mostrado en la **Tabla XIV**.

**Tabla X : Estadística Descriptiva Uso de Red 1000 Usuarios**

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
Polling	111	2468,14	22,8531481	1837,14287
Long Polling	111	2467,04	22,842963	1837,60696
WebSocket	111	1186,69	10,9878704	689,152443



**Figura 18 Uso de Red 1000 Usuarios**

**Fuente:** Manuel Moreta

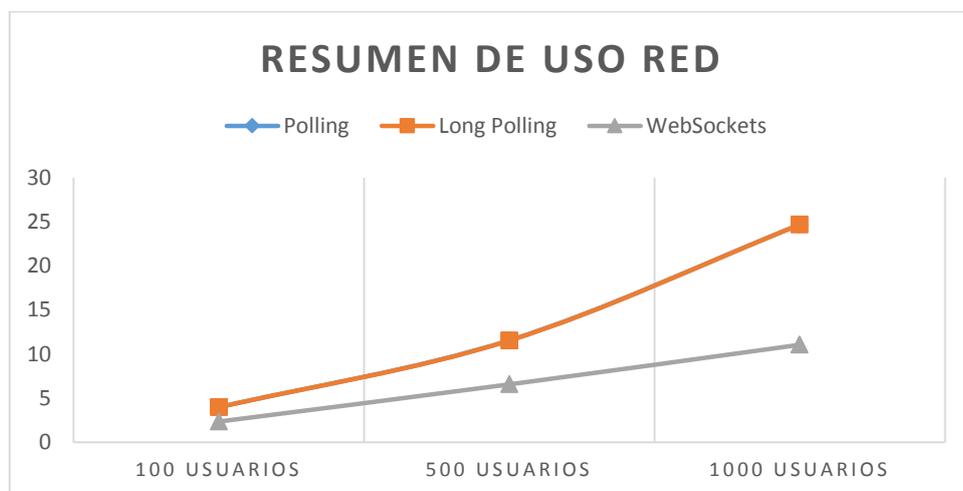
Con los datos de la **Tabla XIV**, se procedió a aplicar Análisis de Varianza obteniendo los siguientes resultados:

**Tabla XI : Análisis de Varianza Uso de Red 1000 Usuarios**

<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	10127,81307	2	5063,9065	3,48122361	0,031936454	3,023864701
Dentro de los grupos	466937,5426	321	1454,6340			
Total	477065,3557	323				

De éstos resultados se puede observar que a mayor cantidad de usuarios las diferencias entre las varianzas son significativas. Esto quiere decir que los mecanismos son diferentes y que el de mejor rendimiento en uso de la Red es WebSocket. Y entre los mecanismos Polling y Long Polling para cualquier análisis no es mucha la diferencia.

Estadísticamente se puede analizar con un nivel de significancia del 5% que el valor de probabilidad que nos proporciona la tabla ANOVA es menor que 5% ( $0,031 < 0,05$ ) en consecuencia el resultado cae en la región de rechazo de la hipótesis nula, lo que involucra que se acepta la hipótesis de investigación en el caso de 1000 usuarios.



**Figura 19 : Resumen Uso Promedio de Red**  
**Fuente: Manuel Moreta**

### *Uso de la RAM*

Para evaluar la hipótesis en el uso de la RAM del servidor, se definen las variables independientes:

$\mu_1 =$  *El uso de la RAM del servidor con el mecanismo Polling*

$\mu_2 =$  *El uso de la RAM del servidor con el mecanismo Long Polling*

$\mu_3 =$  *El uso de la RAM del servidor con el mecanismo WebSocket*

Una vez definidas las variables se procede a definir la hipótesis:

**Hipótesis Nula:** El uso de la RAM del servidor del mecanismo Polling es menor que el mecanismo Long Polling y este también es menor que WebSocket.

$$H_0: \mu_1 \leq \mu_2 \leq \mu_3$$

**Hipótesis de Investigación:** El uso de la RAM del servidor del mecanismo Polling es mayor que el mecanismo Long Polling y este a su vez mayor que WebSocket.

$$H_0: \mu_1 > \mu_2 > \mu_3$$

Nivel de Confianza: 5%

Con una concurrencia de 100 usuarios conectados al prototipo se obtuvo un conjunto de datos del porcentaje de uso de la RAM del servidor usando **sar**, los cuales luego con la ayuda de Microsoft Excel se realizaron los cálculos para obtener la estadística descriptiva, los que se muestran en la **Tabla XVII**, donde se obtuvo un uso promedio de 20,67 con Polling, 20,67 con Long Polling y 20,68 con WebSocket:

**Tabla XII : Estadística Descriptiva Uso de RAM 100 Usuarios**

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
Polling	111	2294,55	20,67162162	0,08717371
Long Polling	111	2294,55	20,67162162	0,08717371
Websocket	111	2296,17	20,68621622	0,09203828

Con una concurrencia de 500 usuarios conectados al prototipo se obtuvo un uso promedio de RAM de 22,02 con Polling, 22,02 con Long Polling y 21,54 con Websocket como se muestra en la **Tabla XIX**:

**Tabla XIII: Estadística Descriptiva de Uso de RAM 500 Usuarios**

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
Polling	111	2444,7	22,02432432	0,90703204
Long Polling	111	2444,7	22,02432432	0,90703204
Websocket	111	2391,26	21,54288288	0,36446434

Con una concurrencia de 1000 usuarios conectados al prototipo se obtuvo un uso promedio de RAM de 22,81 con Polling, 22,81 con Long Polling y 22,67 con Websocket como se muestra en la **Tabla XXI**:

**Tabla XIV : Estadística Descriptiva Uso de RAM 1000 Usuarios**

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
Polling	111	2531,93	22,81018018	1,63894906
Long Polling	111	2531,93	22,81018018	1,63894906
Websocket	111	2516,91	22,67486486	1,17917975

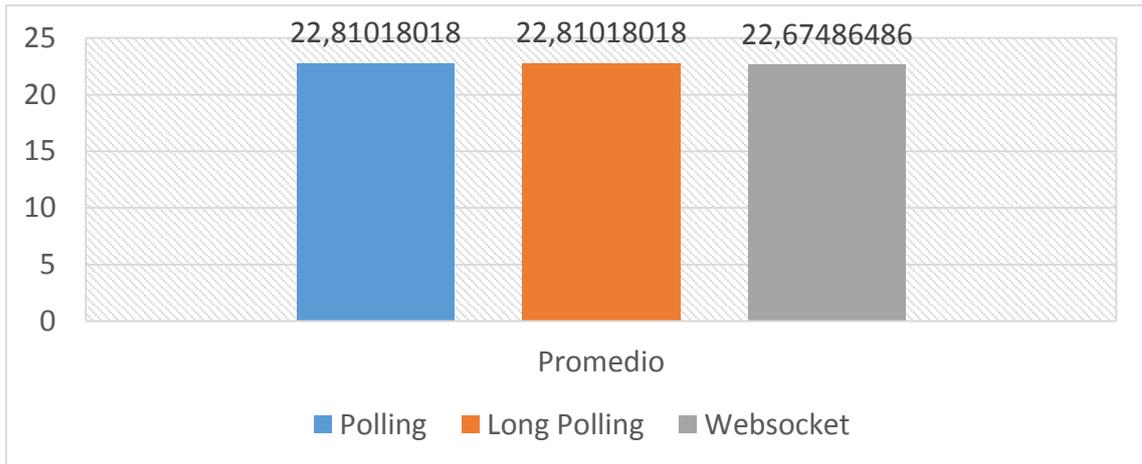


Figura 20 Uso de RAM 1000 Usuarios

Fuente: Manuel Moreta

Con estos valores se procedió a aplicar Análisis de Varianza, y se obtuvieron los datos que se puede observar en la **Tabla XV**:

Tabla XV : Análisis de Varianza de Uso de RAM 1000 Usuarios

Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Entre grupos	1,35495736	2	0,677478679	0,45600191	0,63421139	3,023092796
Dentro de los grupos	490,278566	330	1,485692624			
Total	491,633523	332				

Con estos resultados se puede observar que la diferencia entre las varianzas no son significativas, por lo que se concluye que Polling usa menos o igual memoria RAM del servidor que Long Polling y este a su vez usa menos que Websocket.

Estadísticamente se puede analizar con un nivel de significancia del 5% que el valor de probabilidad proporcionado por la tabla ANOVA es mayor que 5% en consecuencia se acepta la hipótesis nula, lo que involucra que se rechaza la hipótesis de investigación.

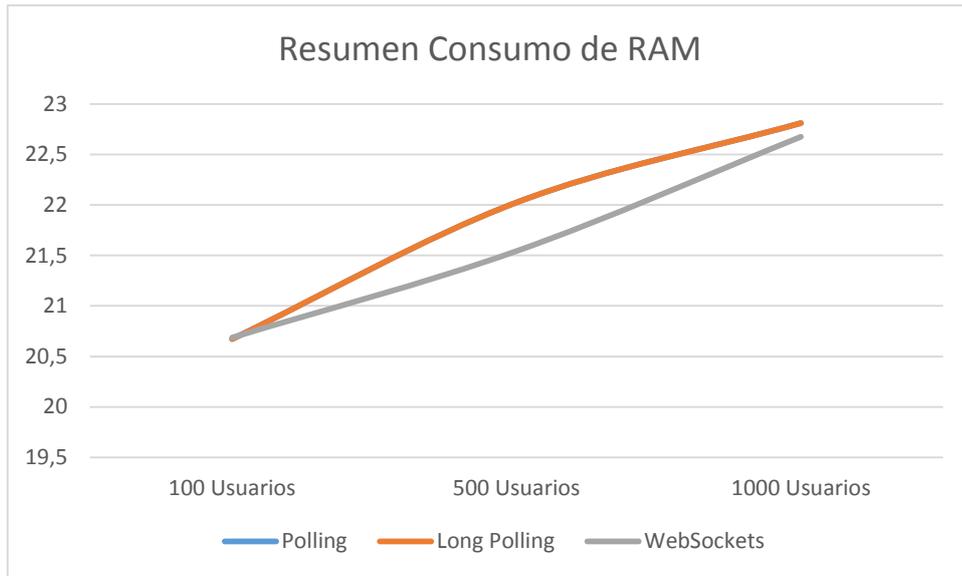


Figura 21 : Resumen Consumo Promedio de RAM

Fuente: Manuel Moreta

### *Uso de la CPU*

Para evaluar la hipótesis en el uso del CPU del servidor, se definen las variables independientes:

$\mu_1 =$  El uso del CPU del servidor con el mecanismo Polling

$\mu_2 =$  El uso del CPU del servidor con el mecanismo Long Polling

$\mu_3 =$  El uso del CPU del servidor con el mecanismo WebSocket

Una vez definidas las variables se procede a la definición de la hipótesis:

**Hipótesis Nula:** El uso del CPU del servidor del mecanismo Polling es menor que el mecanismo Long Polling y este también es menor que WebSocket.

$$H_0: \mu_1 < \mu_2 \leq \mu_3$$

**Hipótesis de Investigación:** El uso del CPU del servidor del mecanismo Polling es mayor que el mecanismo Long Polling y este a su vez mayor que Websocket.

$$H_0: \mu_1 > \mu_2 > \mu_3$$

Nivel de Confianza: 5%

Con una concurrencia de 100 usuarios conectados al prototipo se obtuvo un conjunto de datos del porcentaje de uso del CPU del servidor usando **sar**, los cuales luego con la ayuda de Microsoft Excel se realizó los cálculos para obtener la estadística descriptiva, los que se muestran en la **Tabla XXIII**, donde se obtuvo un uso promedio de 0,18 con Polling, 0,17 con Long Polling y 0,13 con Websocket:

**Tabla XVI : Estadística Descriptiva Uso de CPU 100 Usuarios**

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
Polling	111	20,01	0,18027027	0,28252629
Long Polling	111	19,51	0,175765766	0,28300827
Websocket	111	14,76	0,132972973	0,24809017

Con una concurrencia de 500 usuarios, se obtuvo los datos que se visualizan en la **Tabla XXV**:

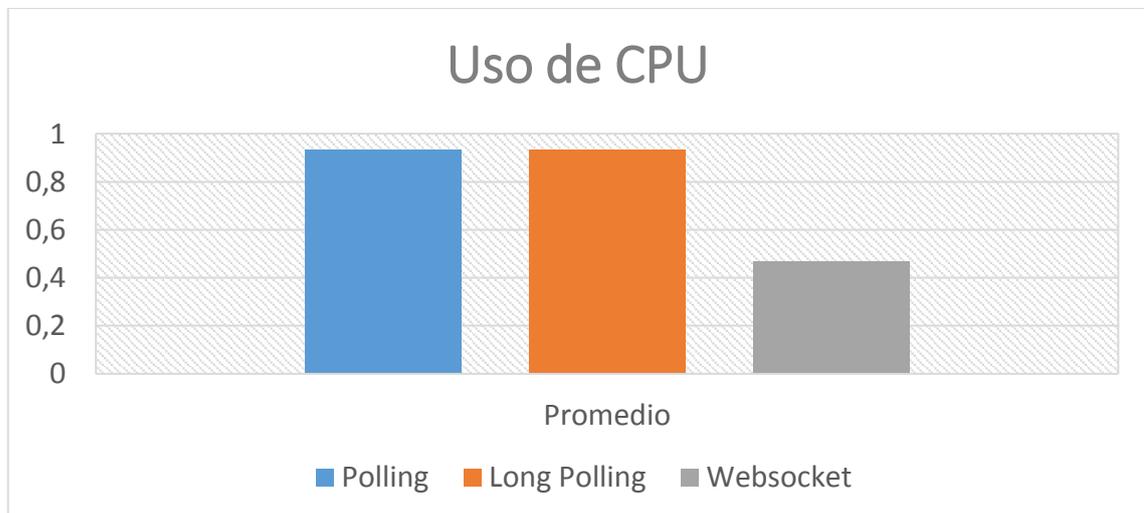
**Tabla XVII : Estadística Descriptiva Uso de CPU 500 Usuarios**

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
Polling	111	59,14	0,53279279	1,69486577
Long Polling	111	58,27	0,52495495	1,65166704
CPU WS	111	31,87	0,28711712	0,78747707

Con una concurrencia de 1000 usuarios conectados simultáneamente al prototipo se obtuvieron los siguientes datos:

**Tabla XVIII : Estadística Descriptiva Uso de CPU 1000 Usuarios**

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
Polling	111	103,32	0,930810811	2,85620025
Long Polling	111	103,32	0,930810811	2,85620025
Websocket	111	51,73	0,466036036	1,46664778



**Figura 22 Uso de CPU 1000 Usuarios**

**Fuente:** Manuel Moreta

En base a los datos obtenidos se procedió a hacer el análisis de Varianzas, para su respectiva demostración de hipótesis:

**Tabla XIX : Análisis de Varianza de Uso de CPU 1000 Usuarios**

<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	15,985154	2	7,992576877	3,3399595	0,0366404	4,670037698
Dentro de los grupos	789,69531	330	2,393016091			
Total	805,68046	332				

De resultados de los cálculos de las muestras tomadas se puede observar que a mayor cantidad de usuarios las diferencias entre las varianzas son significativas. Esto quiere decir que los mecanismos son diferentes y que el de mejor rendimiento en uso del CPU del Servidor es Websocket. Y entre los mecanismos Polling y Long Polling para cualquier análisis no es mucha la diferencia.

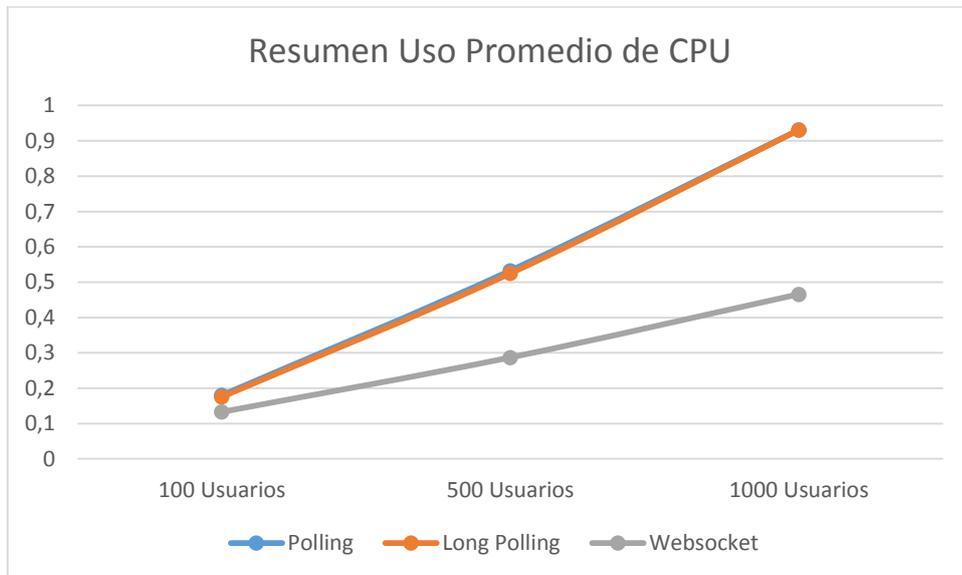


Figura 23 Resumen Uso Promedio de CPU

Fuente: Manuel Moreta

Estadísticamente se puede analizar con un nivel de significancia del 5% que el valor de probabilidad que nos proporciona la tabla ANOVA es menor que 5% ( $0,0366 < 0,05$ ) en consecuencia el resultado cae en la región de rechazo de la hipótesis nula, lo que involucra que se acepta la hipótesis de investigación para el caso de 1000 usuarios.

### 3.6. DEMOSTRACIÓN DE LA HIPÓTESIS

Luego de haber realizado el análisis estadístico de los mecanismos de eventos asíncronos, Polling, Long Polling y Websocket, para la comunicación basada en la web con concurrencia de 1000 usuarios, enviando 10 mensajes, en un intervalo de 10

segundos entre cada uno se observa que con el parámetro uso de memoria RAM del servidor no existe diferencia significativa de las varianzas por lo que no se verifica la hipótesis de investigación, pero como los parámetros Uso de CPU del servidor, uso de la Red y la latencia se llega a observar que la diferencia es significativa en consecuencia se verifica la hipótesis de investigación, en la cual concluyo que el mecanismo Websocket es el más aceptable por tener menor uso de CPU del servidor, menos uso de Red y menor latencia.

Haciendo la asignación de valores proporcionales a cada uno de los parámetros haciendo referencia a la **Tabla I** tenemos:

**Tabla XX: Asignación Proporcional de Valores**

<b>Parámetro</b>	<b>Polling</b>	<b>Long Polling</b>	<b>Websocket</b>	<b>Valor</b>
Latencia	16	34	50	50
Uso de la Red	14	14	30	30
Uso de RAM	10	10	10	10
Uso del CPU	5	5	10	10
<b>Total</b>	45	63	100	100

Websocket es el mecanismo con mejor puntaje lo que nos indica que es el de mejor rendimiento obteniendo 100 de 100 puntos posibles, en segundo lugar tenemos a Long Polling con 63 puntos y Polling último con 45 puntos, por consiguiente se acepta la hipótesis “El estudio estadístico de los mecanismos de eventos asíncronos, Polling, Long Polling y Websocket, para la comunicación basada en la web permitirá seleccionar el de mejor

rendimiento para la comunicación en la plataforma web de “Rastreo Directo Satelital 2.0”.

## CAPÍTULO IV

### 4. DESARROLLO DE LA PLATAFORMA WEB “RASTREO DIRECTO SATELITAL 2.0”

#### 4.1. INTRODUCCIÓN

El desarrollo del de la plataforma web “Rastreo Directo Satelital 2.0” se realizó para la empresa “FASNOTEQ S.A.” una empresa ecuatoriana que brinda servicios de rastreo y seguimiento de vehículos. Dicha empresa brindaba sus servicios mediante un sistema el cual puede ser administrado y utilizado a través de internet en su página web que se comunica con dispositivos GPS instalados en los vehículos de los que se pueden obtener datos de importancia para los usuarios y empresas que contratan el sistema, pero dado al crecimiento de la empresa y a los avances tecnológicos en cuanto a dispositivos de comunicación como a tecnologías de desarrollo de aplicaciones web, fue necesario desplegar una nueva versión del sistema.

#### 4.2. FASTNOTEQ S.A.

FASNOTEQ S.A. ofrece productos basados en GSM/GPRS, GPS y satelitales vía Inmarsat para la localización segura y rápida de vehículos en el país con mayor cobertura y un

mínimo de costo, ideal para rastreo de todo tipo de transporte, control de rutas y recuperación del mismo.

Toda nuestra tecnología se puede ver por internet desde cualquier parte del mundo en nuestra página web [www.rastreodirecto.com](http://www.rastreodirecto.com), donde se tiene mapas digitalizados con los desplazamientos y ubicaciones de los vehículos.

El principal servicio es el sistema “Rastreo Directo Satelital”, este sistema de rastreo hace uso de varias tecnologías de posicionamiento y telecomunicaciones, funciona a través de equipos AVL, señales de celulares, mapas digitalizados y comunicación vía internet. Todos coordinados por un software profesional con el respaldo de un staff de ingenieros altamente calificados y entrenados en cuanto a la administración y uso del sistema.

La persona interesada en conocer la posición y recorrido del vehículo, carga o personal puede ingresar a la página web, teclear su usuario y contraseña y acceder a un mapa digital don de un ícono representa la posición actual de su vehículo. Puede usar zoom para alejarse y ver, si fuera el caso, toda una flota de vehículos en todo el país o acercarse más y ver por regiones, por ciudades, o incluso hacer un acercamiento extremo y saber en qué calles está ubicado cada uno.

La página web también puede generar reportes tabulados de tiempos y recorridos realizados en los días, semanas o meses anteriores.

Productos que ofrece la empresa:

- Divisiones Especializadas.

- Rastreo de mercadería en camiones de carga y rastreo de flotas de vehículos.
- Rastreo de tanqueros para evitar contrabando de combustible.
- División especializada en buques, lanchas y todo tipo de embarcaciones navieras.
- División aviónica, especializada en aeronaves a nivel mundial.

Datos del responsable de la empresa:

- **Nombre:** Ing. Rodrigo Ortega(GERENTE GENERAL)
- **Ciudad:** Ambato (Ecuador)
- **E-mail:** [gerencia@rastreodirecto.com](mailto:gerencia@rastreodirecto.com)
- **Teléfono:** 0999293029

#### 4.2.1. MISIÓN

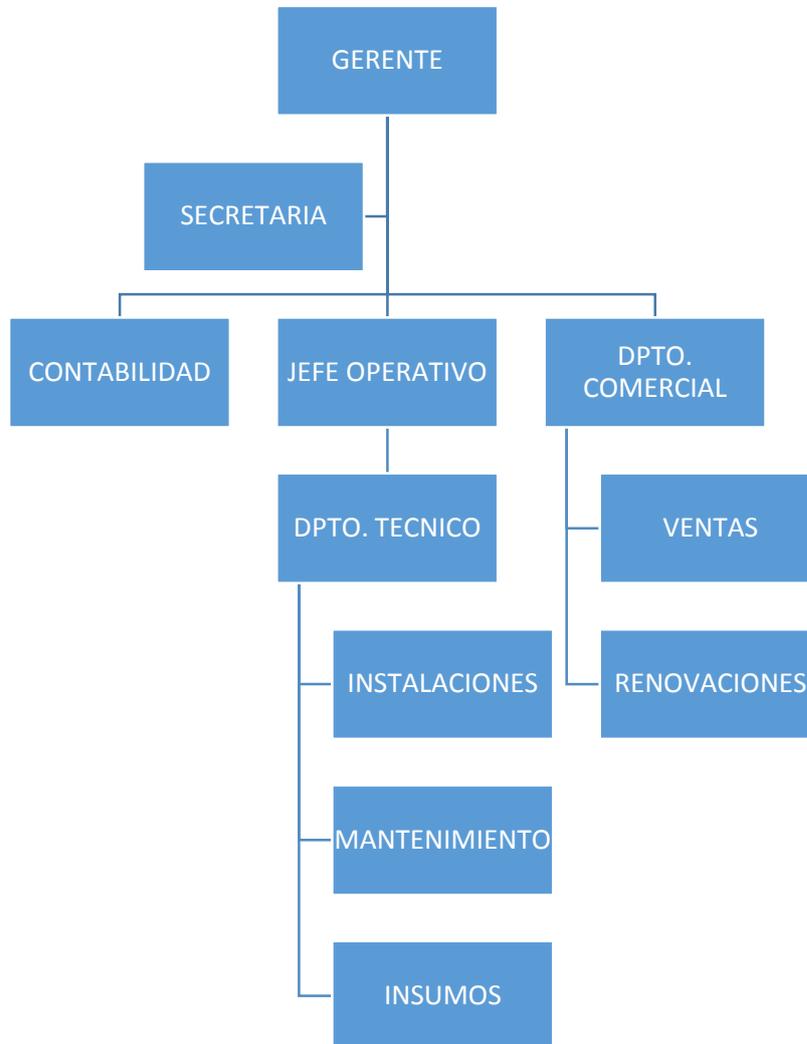
Brindar a nuestros clientes un sistema de seguridad mediante el cual se pueda monitorear en tiempo real la ubicación de sus vehículos con reportes exactos de velocidad, paradas, kilometraje recorrido, eventos varios, ingreso y egreso de zonas predefinidas, control de rutas y control de flotas. Para lo cual contamos con personal altamente calificado y entrenado para brindar todo el asesoramiento técnico tanto en hardware como en software al momento que lo requiera mediante el uso de nuestro aplicativo que es cien por ciento producción nacional.

#### 4.2.2. VISIÓN

Llegar a ser líderes en la provisión de sistemas electrónicos de seguridad y control electrónico para el buen uso y aprovechamiento de bienes patrimoniales y personales mediante la creación de soluciones tecnológicas de acuerdo a los requerimientos de cada uno de nuestros clientes.

#### 4.2.3. ESTRUCTURA

FASNOTEQ S.A. como empresa posee el siguiente organigrama estructural:



**Figura 24 : Organigrama FASTNOTEQ**

**FUENTE: FASTNOTEQ**

#### 4.3. METODOLOGÍA DE DESARROLLO DE LA PLATAFORMA

Para la implementación se hizo uso de la Metodología SCRUM, la cual nos permite hacer un desarrollo adaptable, orientado a las personas y que se basa en la construcción de incrementos funcionales del sistema.

#### 4.3.1. EQUIPO DE TRABAJO

El equipo de trabajo que intervino en el desarrollo de la plataforma web “Rastreo Directo Satelital 2.0” fueron:

Tabla XXI : Equipo SCRUM

Persona	Contacto	Rol
Ing. Edwin Rocha	<a href="mailto:gerencia@rastreodirecto.com">gerencia@rastreodirecto.com</a>	Product Owner
Ing. Jorge Menendez	<a href="mailto:jmenendez@esPOCH.edu.ec">jmenendez@esPOCH.edu.ec</a>	Scrum Master
Manuel Moreta	<a href="mailto:manukorp@outlook.com">manukorp@outlook.com</a>	Desarrollo

#### 4.3.2. HISTORIAS DE USUARIO

Para el desarrollo de la plataforma web “Rastreo Directo Satelital 2.0” se tomó las historias de usuario que debe cumplir la nueva plataforma y estas son:

Historia de Usuario	
<b>Número:</b> 1	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Monitorear Vehículo	
<b>Prioridad en Negocio:</b> 1	<b>Riesgo en Desarrollo:</b> Baja
<b>Descripción:</b> Quiero monitorear la posición de mi vehículo, y verlo en un mapa digital	
<b>Validación:</b> El cliente puede ver la posición de su vehículo en tiempo real	
Historia de Usuario	
<b>Número:</b> 2	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Ver logs del sistema	
<b>Prioridad en Negocio:</b> 4	<b>Riesgo en Desarrollo:</b> Baja
<b>Descripción:</b> Quiero ver los logs de las notificaciones que hagan los vehículos	
<b>Validación:</b> El cliente puede ver los logs de las notificaciones	

Historia de Usuario	
<b>Número:</b> 3	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Visualizar Históricos del vehículo	
<b>Prioridad en Negocio:</b> 3	<b>Riesgo en Desarrollo:</b> Baja
<b>Descripción:</b> Quiero visualizar el recorrido de un vehículo en un rango de fechas determinado	
<b>Validación:</b> El cliente puede listar el recorrido de un vehículo en un rango de fechas determinado	

Historia de Usuario	
<b>Número:</b> 4	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Administrar datos del vehículo	
<b>Prioridad en Negocio:</b> 2	<b>Riesgo en Desarrollo:</b> Baja
<b>Descripción:</b> Quiero visualizar y editar los datos del vehículo	
<b>Validación:</b> El cliente puede editar y visualizar los datos de los vehículos	

Historia de Usuario	
<b>Número:</b> 5	<b>Usuario:</b> Product Owner
<b>Nombre de la Historia:</b> Administrar usuarios	
<b>Prioridad en Negocio:</b> 3	<b>Riesgo en Desarrollo:</b> Baja
<b>Descripción:</b> Quiero administrar los usuarios que acceden al sistema	
<b>Validación:</b> El Product Owner puede administrar (agregar, editar, eliminar) los usuarios del sistema	

Historia de Usuario	
<b>Número:</b> 6	<b>Usuario:</b> Product Owner
<b>Nombre de la Historia:</b> Administrar Empresas	
<b>Prioridad en Negocio:</b> 4	<b>Riesgo en Desarrollo:</b> Baja
<b>Descripción:</b> Quiero administrar (agregar, editar, eliminar) las empresas que son mis clientes.	
<b>Validación:</b> El cliente puede administrar (agregar, editar, eliminar) las empresas que son clientes.	

Historia de Usuario	
<b>Número:</b> 7	<b>Usuario:</b> Product Owner
<b>Nombre de la Historia:</b> Administrar Eventos	
<b>Prioridad en Negocio:</b> 4	<b>Riesgo en Desarrollo:</b> Baja
<b>Descripción:</b> Quiero administrar (agregar, editar, eliminar) los eventos que reportan los vehículos.	
<b>Validación:</b> El cliente puede administrar (agregar, editar, eliminar) los eventos que reportan los vehículos.	

Historia de Usuario	
<b>Número:</b> 8	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Administrar alertas por Eventos	
<b>Prioridad en Negocio:</b> 5	<b>Riesgo en Desarrollo:</b> Baja
<b>Descripción:</b> Quiero administrar (agregar, editar, eliminar) alertas por los eventos que reportan los vehículos.	
<b>Validación:</b> El cliente puede administrar (agregar, editar, eliminar) alertas por los eventos que reportan los vehículos.	

Historia de Usuario	
<b>Número:</b> 9	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Administrar Geocercas	
<b>Prioridad en Negocio:</b> 3	<b>Riesgo en Desarrollo:</b> Baja
<b>Descripción:</b> Quiero administrar (agregar, editar, eliminar) geocercas	
<b>Validación:</b> El cliente puede administrar (agregar, editar, eliminar) geocercas	

Historia de Usuario	
<b>Número:</b> 10	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Administrar Puntos de Interés	
<b>Prioridad en Negocio:</b> 3	<b>Riesgo en Desarrollo:</b> Baja
<b>Descripción:</b> Quiero administrar (agregar, editar, eliminar) puntos de interés	
<b>Validación:</b> El cliente puede administrar (agregar, editar, eliminar) puntos de interés	

#### 4.3.3. BACKLOG DEL PRODUCTO

El Backlog del Producto es el espacio en el que se detallará los requerimientos de la Plataforma Web “Rastreo Directo Satelital 2.0” que se obtuvieron en base a las historias de usuario que se definieron anteriormente.

<b>N°</b>	<b>Descripción</b>	<b>Prioridad</b>
1.	Crear el módulo de comunicación	1
2.	Crear módulo de procesamiento de la base de datos	1
3.	Crear decodificador de AVL para el modelo SkyPatrol TT8750	2
4.	Crear módulo de Logs de comunicación	4
5.	Crear módulo de alertas de eventos	4
6.	Crear la base de datos	1
7.	Crear interfaz de administración de vehículos	2
8.	Crear interfaz de administración de Usuarios	3
9.	Crear interfaz de administración de Empresas	4
10.	Crear interfaz de administración Eventos	4
11.	Crear interfaz de administración de Puntos de interés	3
12.	Crear interfaz de administración de Geocercas	3
13.	Crear interfaz de administración de Mantenimientos de AVLs	5
14.	Crear interfaz de administración de Alertas	5
15.	Diseñar y Crear reportes	3
16.	Crear interfaz de monitoreo de vehículos	2
17.	Preparar ambiente de desarrollo	1
18	Definir estándares de codificación	1
19	Definir arquitectura del sistema	1

#### 4.3.4. CICLOS DE DESARROLLO – SCRUM

Para el desarrollo de la plataforma se usó SCRUM para la gestión del proyecto, el proyecto se lo dividió en 7 Iteraciones de 15 días laborables cada uno.

##### *Iteración 0 – Investigación*

En la primera iteración fue dedicada a la investigación y la preparación del ambiente de desarrollo.

N°	Tarea	Tipo
1	Preparar ambiente de desarrollo	Análisis
2	Crear la base de datos	Análisis / Diseño / Desarrollo
3	Definir estándar de codificación	Análisis
4	Definir arquitectura del sistema	Análisis y Diseño

Dentro de esta iteración se escogió las herramientas de desarrollo y el ambiente en el cuál se va a desplegar el sistema, en esta iteración se generaron los siguientes entregables:

**Estándar de Codificación:** Camel-Case

**Herramientas de Desarrollo (Ambiente):** Como herramienta de desarrollo se utilizará Netbeans 7, con PHP y JavaScript como lenguajes y con CentOS6 en el servidor.

**Arquitectura del Sistema:** para el despliegue del sistema se usará una arquitectura a tres capas, con Modelo Vista Controlador como patrón.

**Esquema de la Base de Datos:** el esquema de la base de datos que se definió se encuentra en el Anexo 1 del presente documento.

### *Iteración 1 – Desarrollo*

En la segunda iteración se inició con el desarrollo de los módulos de mayor prioridad.

<b>N°</b>	<b>Tarea</b>	<b>Tipo</b>
1	Crear el módulo de comunicación	Desarrollo
2	Crear decodificador de AVL para el modelo SkyPatrol TT8750	Desarrollo
3	Crear módulo de procesamiento de la base de datos	Desarrollo
4	Crear módulo de Logs de comunicación	Desarrollo

Los entregables de esta iteración fueron los siguientes:

Módulo de Comunicación, el presente módulo permite comunicar a los clientes conectados la información provista de cada uno de los vehículos en el momento que estos emiten un reporte de posición

Decodificador de AVL, los decodificadores se encontraban desarrollados en Java, y dado que se empezó a usar NodeJS se hizo necesario migrar estos al mismo lenguaje que el Módulo de Comunicación, para poder facilitar la comunicación entre los mismos.

Todas las transacciones que el Módulo de Comunicación realice se registraran en Logs del sistema, para ello se implementó un Módulo gestione los Logs. Como punto adicional dentro de

este módulo se agregó la funcionalidad de poder emitir dichos registros a través del Módulo de Comunicación hacia el cliente.

Para poder contar con un histórico de las posiciones de los vehículos se debió almacenar dichos registros en una base de datos, para ello implementamos un Módulo que Gestione la inserción y actualización de dichos registros.

### *Iteración 2 – Desarrollo*

En la segunda iteración se inició con el desarrollo de los módulos de mayor prioridad.

<b>N°</b>	<b>Tarea</b>	<b>Tipo</b>
1	Crear interfaz de monitoreo de vehículos	Desarrollo
2	Crear interfaz de administración de Usuarios	Desarrollo
2	Crear interfaz de administración de vehículos	Desarrollo

Dentro de esta iteración se desarrolló la interfaz para el usuario, dentro las primeras funcionalidades que se desarrollaron fueron:

Interfaz de Monitoreo, esta permite a los clientes ver la última posición reportada de sus vehículos y poder hacer un seguimiento en tiempo real. Para poder seleccionar uno o varios vehículos se creó una estructura tipo árbol como en la **Figura 18** donde puede navegar entre los diferentes niveles de objetos. Además para esta se adicionó una opción de búsqueda rápida que permite a los usuarios encontrar sus vehículos en base a tres parámetros definidos dentro de la base de datos que son: MDMID, Alias o Placa.



Figura 25: Árbol de Objetos

Interfaz de Administración de Usuarios, para el Product Owner es de vital importancia poder gestionar los usuarios que les serán asignados a sus clientes, para ello se implementó una interfaz por medio de la cual permita cumplir con este requerimiento. Para poder cumplir con este requerimiento y mantener el Monitoreo se hizo el despliegue de la interfaz de administración sobre el esquema tipo árbol que se había estado usando para listar los elementos del sistema como son Usuarios y vehículos. Para lograr esa administración se cambió las vistas del componente que se creó para la administración del mismo.

Interfaz de Administración de Vehículos, además de los usuarios, el Product Owner requiere poder llevar control sobre los dispositivos conectados en los vehículos de los clientes.

### *Iteración 3 – Desarrollo*

En la presente iteración se desarrolló los módulos de mayor prioridad, que le continúan a los anteriores en el BackLog.

<b>N°</b>	<b>Tarea</b>	<b>Tipo</b>
1	Crear interfaz de administración de Empresas	Desarrollo
2	Crear interfaz de administración de Puntos de interés	Desarrollo
3	Crear interfaz de administración de Geocercas	Desarrollo

Para esta iteración y ya teniendo como base las pruebas que se realizaron a los entregables anteriores se siguió con el desarrollo de:

Interfaz de administración de Empresas, para el Product Owner las empresas son las bases de su esquema de distribución de clientes, para ello le es necesario tener una interfaz de administración que le permita Agregar, Editar y Eliminar empresas bajo las cuales serán desplegados los demás objetos como usuarios, vehículos y responsables.

Interfaz de Administración de Puntos de Interés, los clientes necesitan definir puntos importantes en el mapa, para ello se desarrolló esta interfaz que permite la creación, modificación, visualización en mapa y eliminación de puntos de interés. Para la realización de esta interfaz se optó por desarrollar otra estructura tipo árbol que permita encontrar y manipular de manera fácil.

Interfaz de Administración de Geocercas, los usuarios necesitan definir áreas o zonas que les permita marcar áreas de trabajo, vivienda, etc. para ello se realizó una interfaz

que a través de la marcación de puntos en el mapa de vaya definiendo las geocercas. Así mismo esta interfaz se la colocó en la estructura antes mencionada.

#### *Iteración 4 – Desarrollo*

En la Iteración 4 En la segunda iteración se inició con el desarrollo de los módulos de mayor prioridad.

<b>N°</b>	<b>Tarea</b>	<b>Tipo</b>
1	Crear interfaz de administración Eventos	Desarrollo
2	Crear interfaz de administración de Alertas	Desarrollo
3	Crear módulo de alertas de eventos	Desarrollo

#### *Iteración 5 – Desarrollo*

En la quinta iteración se inició con el desarrollo de los módulos de mayor prioridad.

<b>N°</b>	<b>Tarea</b>	<b>Tipo</b>
1	Diseñar y Crear reportes	Desarrollo
2	Crear interfaz de administración de Mantenimientos de vehículos	Desarrollo

## CONCLUSIONES

- Se pudo evidenciar que con el parámetro Uso de RAM del servidor los tres mecanismos tiene un comportamiento similar, obteniendo con ANOVA que no hay diferencia significativa entre los mecanismos analizados.
- La plataforma web de “Rastreo Directo Satelital 2.0” fue desarrollada con el mecanismo WebSocket que mostró mejor rendimiento en las pruebas realizadas en la presente investigación, obteniendo 100 de 100 puntos posibles.
- Los parámetros analizados para este estudio estadístico entre los mecanismos de comunicación Polling, Long Polling y WebSocket fueron: latencia, uso de red, uso de RAM del servidor y uso de CPU del servidor; con sus respectivos indicadores permitieron el mecanismo de mejor rendimiento para la plataforma web “Rastreo Directo Satelital”.
- Los Framework son herramientas software que facilitan la creación de nuevos sistemas.
- Se recomienda adoptar WebSocket en el desarrollo de aplicaciones web de comunicación en tiempo real, para el óptimo desenvolvimiento de las mismas.

## RECOMENDACIONES

- Se recomienda realizar un análisis estadístico de los Frameworks para la comunicación en tiempo real la web, para estudiar el comportamiento y determinar su influencia sobre las implementaciones de sistemas.
- Para el desarrollo de proyectos con SCRUM se debe conocer bien el alcance y roles de cada uno de los actores del equipo SCRUM, para poder lograr un desarrollo integrado.
- Se recomienda optimizar la base de datos de la empresa para que, con el crecimiento de la misma, los datos no corran ningún riesgo en el sistema "Rastreo Directo Satelital 2.0".

## RESUMEN

Investigación para el análisis estadístico de mecanismos de eventos asíncronos (Polling, Long Polling y WebSocket), basados en la web para la integración de la comunicación en tiempo real en la plataforma web “Rastreo Directo Satelital 2.0”, de la empresa FASTNOTEQ S.A de la ciudad de Ambato.

Para desarrollar la investigación se utilizó método científico, para la recopilación de información se usó herramientas de benchmarking que permiten conocer, en tiempo real, el consumo del CPU, de la RAM, de la red y la latencia; para el análisis se aplicó estadística descriptiva e inferencial, el estudio comparativo de mecanismos permitió determinar el de mejor rendimiento, se desarrolló un ambiente simulado con características del escenario que se va a implementar en la Plataforma Web “Rastreo Directo Satelital 2.0”.

En el análisis estadístico se observó que: el mecanismo de menor latencia, menor uso de Red y menor consumo de memoria RAM WebSocket; los mecanismos estudiados, presentan un rendimiento similar en el uso del CPU del servidor.

Aplicando análisis de varianza a los resultados de las pruebas realizadas y luego de la respectiva ponderación, se pudo determinar que el mecanismo WebSocket es con 100 de 100 puntos, el de mejor rendimiento para implementar la plataforma web “Rastreo Directo Satelital 2.0”.

Se recomienda optimizar la base de datos de la empresa para que, con el crecimiento de la misma, los datos no corran ningún riesgo.

**Palabras Claves:** /MECANISMOS DE COMUNICACIÓN WEB/ /COMUNICACIÓN EN TIEMPO REAL/ /ESTUDIO COMPARATIVO/ /ANÁLISIS ESTADÍSTICO.

## **ABSTRACT**

Research for statistical analysis of mechanisms for asynchronous events (Polling, Long Polling and WebSocket), based on the web for the integration of real time communication on the website "Rastreo Directo Satelital 2.0", from FASTNOTEQ S.A. Company from Ambato city.

In order to perform this research, the Scientific method was used, for data collection it was used benchmarking tools which allow to know in real time the consumption of the CPU, the RAM, the web and the latency; for the analysis it was applied descriptive and inferential statistics, the comparative study of mechanisms allowed to determine the best performance, it was develop a simulated environment with features of the scenario to implemented the Web Platform " Rastreo Directo Satelital 2.0".

In the statistical analysis it was observed that: the mechanism of lower latency for concurrence of 100, 500 and 1000 users is WebSocket; the studied mechanism show a similar performance of 1000 users, WebSocket is the one with the best performance.

By applying the variance analysis to the results of applied tests, it was determined that WebSocket mechanism is 80%, according to the measure of the set parameters, the one of the best performance to implemnet web platform "Direct Satellite Tracking 2.0".

It is recommended to optimize the company 's database, with the growth of the same, data do not get into any risk.

## Bibliografía

1. **Beverstein, Roxanne , Popescu, Alexandru y Marinescu, Floyd** . InfoQ. [En línea] 2006. [Citado el: 20 de Agosto de 2012.] [www.infoq.com/articles/Web-Sockets-Proxy-Servers](http://www.infoq.com/articles/Web-Sockets-Proxy-Servers).
2. **Lubbers, Peter, Greco, Frank y Kaazing Corporation**. *Fat Cow*. [En línea] 23 de Diciembre de 2011. [Citado el: 15 de Agosto de 2012.] <http://www.fatcow.com/edu/websocket-quantum-bg/>.
3. **Qveflander, Nikolai**. [En línea] 17 de Agosto de 2010. [Citado el: 26 de Septiembre de 2012.] <http://umu.diva-portal.org/smash/get/diva2:354621/FULLTEXT01.pdf>.
4. **Luján Mora, Sergio**. *GPLSI*. [En línea] 31 de Octubre de 2002. <http://gplsi.dlsi.ua.es/almacenes/ver.php?pdf=42>.
5. **Berners-Lee, Tim, Fielding, Roy T. y Frystyk Nielsen, Henrik**. *IETF*. [En línea] Mayo de 1996. <http://tools.ietf.org/html/rfc1945>.
6. **Aguilar, José M**. *Variable Not Found*. [En línea] 17 de Enero de 2012. <http://www.variablenotfound.com/2012/01/introduccion-signalr-i-conceptos.html>.
7. **Ubl, Malte y Kitamura, Eiji** . *HTML5 ROCKS*. [En línea] 20 de Octubre de 2010. [Citado el: 10 de 01 de 2013.] <http://www.html5rocks.com/es/tutorials/websockets/basics/>.

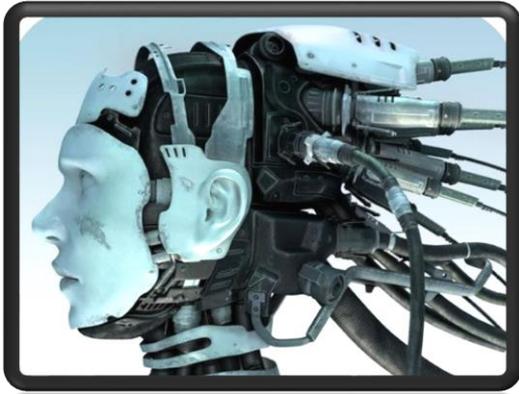
8. **Maryka, Sthephen.** *TheServerSide.com*. [En línea] 1 de Abril de 2009.  
<http://www.theserverside.com/news/1363576/What-is-the-Asynchronous-Web-and-How-is-it-Revolutionary>.
9. **Mahemoff, Michael.** *Ajax Design Patterns: Creating Web 2.0 Sites With Programming And Usability Patterns*. *Ajax Design Patterns: Creating Web 2.0 Sites With Programming And Usability Patterns*. San Fancisco : O'Reilly Media Inc., 2006, págs. 210-221.
10. **Loreto, S., y otros.** *Internet Engineering Task Force*. [En línea] Abril de 2011.  
<https://tools.ietf.org/html/rfc6202>.
11. **Pimentel, Victoria y Nickerson, Bradford G.** IEEE Xplore. [En línea] Agosto de 2012.  
10.1109/MIC.2012.64.
12. **Lubbers, Peter, Albers, Brian y Salim, Frank.** *Pro HTML5 Programming*. s.l. : Apress, 2011.
13. **Gutiérrez, Javier J.** [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf).  
[http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf). [En línea] 2010. [Citado el: 27 de Julio de 2014.] [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf).
14. **Rauch, Guillermo.** <https://github.com/learnboost/socket.io-spec>.  
<https://github.com/learnboost/socket.io-spec>. [En línea] 14 de Enero de 2014.  
<https://github.com/learnboost/socket.io-spec>.
15. **Rodríguez, Moisés.** Aarcos. [En línea] 16 de Julio de 2010.  
<http://alarcos.esi.uclm.es/per/fruiz/curs/santander/mrodriguez-iso25000-update.pdf>.
16. **Luque Rubio, Aarón.** [En línea] 12 de Junio de 2007.  
[http://ddd.uab.cat/pub/trerecpro/2007/hdl\\_2072\\_8966/PFCLuqueRubio.pdf](http://ddd.uab.cat/pub/trerecpro/2007/hdl_2072_8966/PFCLuqueRubio.pdf).

17. **Mayol Ramis, Guillem Carles.** <http://upcommons.upc.edu/>. [En línea] 24 de Febrero de 2014.

<http://upcommons.upc.edu/pfc/bitstream/2099.1/21049/4/memoria.pdf>.

**ANEXO**





# MANUAL DE USUARIO

## RDS

*“Sistema de Rastreo Satelital”*

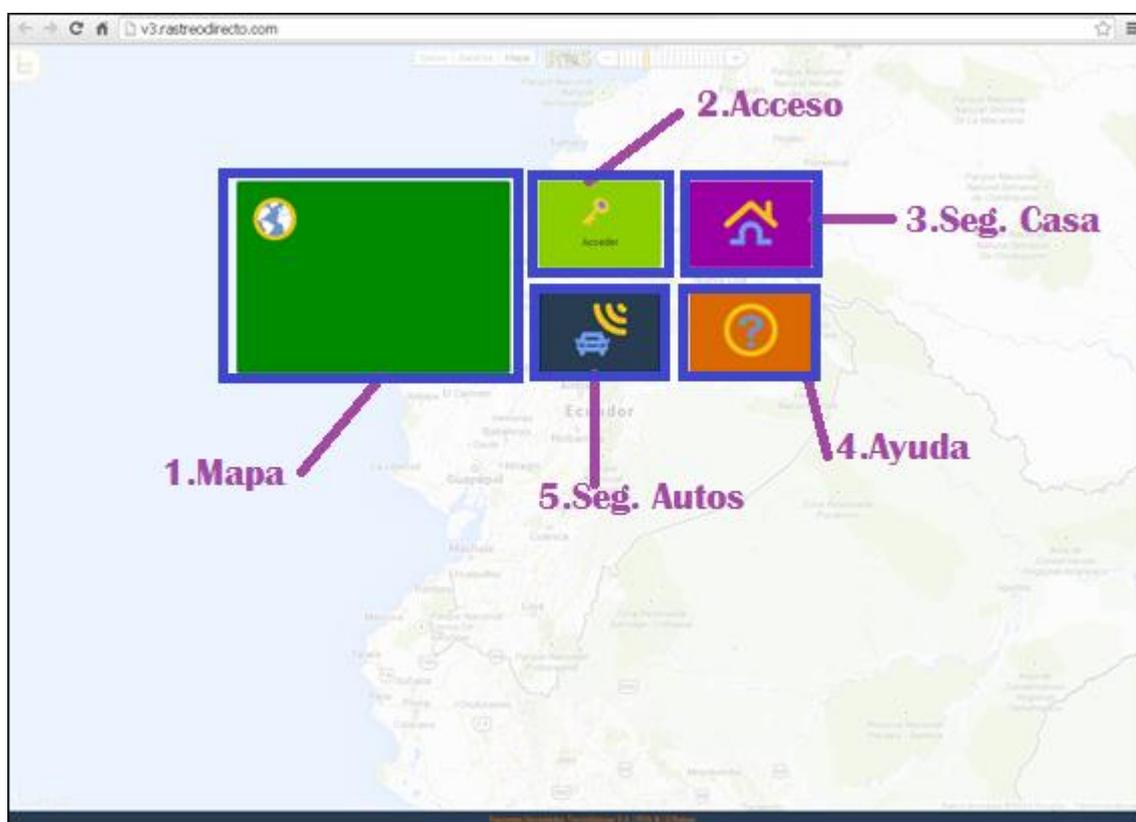
## ÍNDICE

1. CONEXIÓN AL SISTEMA .....	87
2. ACCESO AL SISTEMA .....	89
3. GESTIÓN DE VEHÍCULOS.....	90
a) Agregar nueva empresa .....	90
b) Editar una empresa .....	91
c) Agregar nuevo usuario .....	92
d) Editar usuario .....	94
e) Eliminar usuario.....	94
f) Reportes de usuarios.....	95
Generación de Reportes.....	96
g) Agregar vehículos.....	96
h) Comando vehículo.....	98
i) Reportes de vehículos .....	99
4. GESTIÓN ADMINISTRATIVA .....	100
a) Crear nueva unidad administrativa .....	101
b) Crear nueva salvoconducto .....	102
c) Crear nueva geocerca .....	102
d) Agregar localización .....	103
e) Dar mantenimiento .....	103
f) Crear un nuevo grupo evento .....	104
g) Editar un grupo evento .....	106
h) Crear un nuevo evento .....	106
4.1. MENÚ DE OPCIONES .....	107

a) Opción Contáctenos.....	108
b) Opción Estado de los equipos.....	108
c) Opción Eventos.....	108
d) Opción registro.....	110
e) Opciones.....	111
f) Estado de conexión.....	111

## 1. CONEXIÓN AL SISTEMA

- Ingreso a la dirección URL: <http://mtop.rastredirecto.com/> perteneciente a la empresa.
- Despliegue de la página principal RDS con cinco opciones diferentes tal como se presenta en la siguiente figura expuesta a continuación.

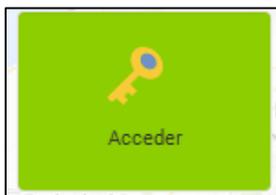


1. **Mapa:** Despliegue de un mapa.
2. **Acceso:** Permite acceder al sistema mediante el ingreso de un usuario y contraseña.
3. **Seguridad en casa**
4. **Seguridad en automóviles**
5. **Ayuda del sistema**

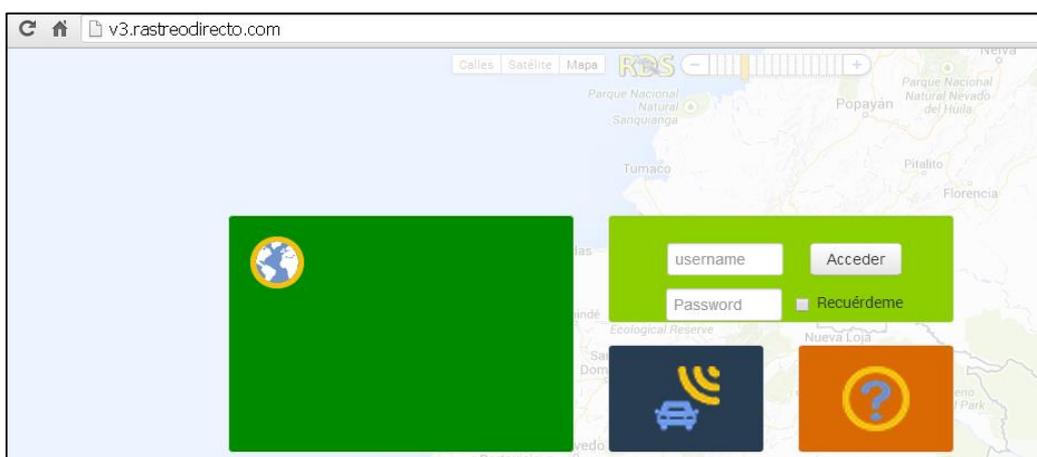


## 2. ACCESO AL SISTEMA

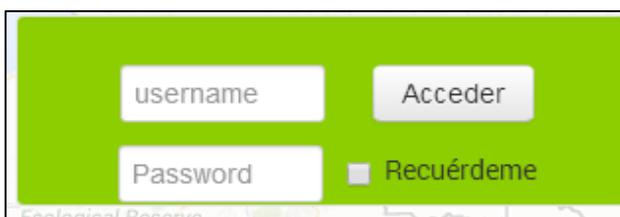
1. Acercar el puntero con el mouse en la opción Acceder.



2. Se despliega la siguiente ventana.



3. Ingresar su usuario y contraseña.



4. Clic en acceder.

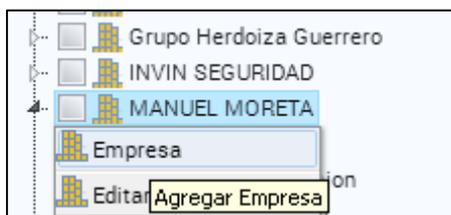
### 3. GESTIÓN DE VEHÍCULOS

#### a) Agregar nueva empresa

1. Seleccionar la opción empresa.
2. Clic derecho del mouse sobre el módulo Empresa.



3. Clic en agregar empresa.



4. Llenar los datos del formulario siguiente:

Ingrese los Datos de la Empresa

Nombre : Mi Empresa

Descripción : Descripción de la Empresa

Dirección : Dirección de la Empresa

Teléfono : 0985171193

Fecha de inicio : 2014-05-20 Fecha de Fin : 2014-05-20

Cancelar Guardar

**Nombre:** Nombre de la nueva empresa.

**Descripción:** Breve descripción de la empresa.

**Dirección:** Dirección de donde se encuentra la empresa.

**Teléfono:** Número de teléfono de la empresa.

**Fecha de inicio:** Por defecto se muestra fecha actual.

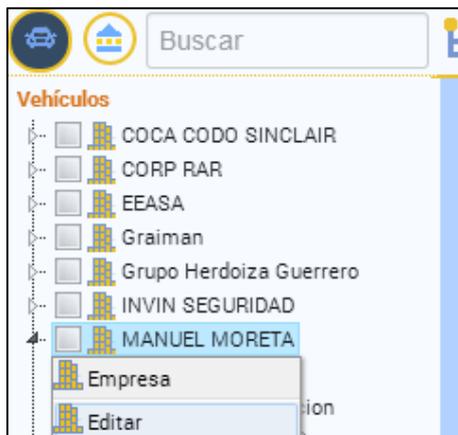
**Fecha fin:** Escoger la fecha de finalización del contrato de la empresa con RDS.

## b) Editar una empresa

1. De las empresas listadas, seleccionamos la empresa a editar.



2. Clic derecho del mouse sobre dicha empresa



3. Seleccionar la opción editar

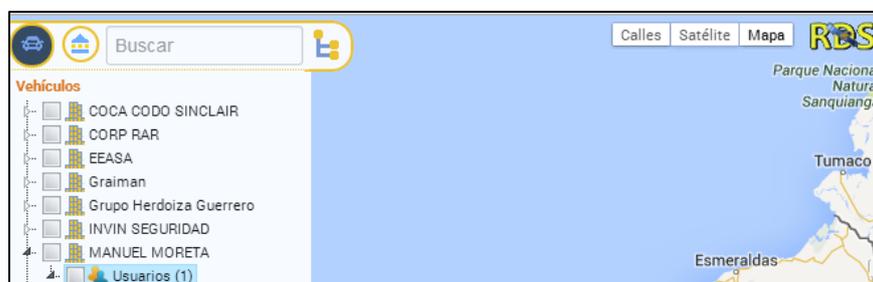


4. Editar los campos necesarios.

5. Guardar cambios, hacer clic en el botón guardar.

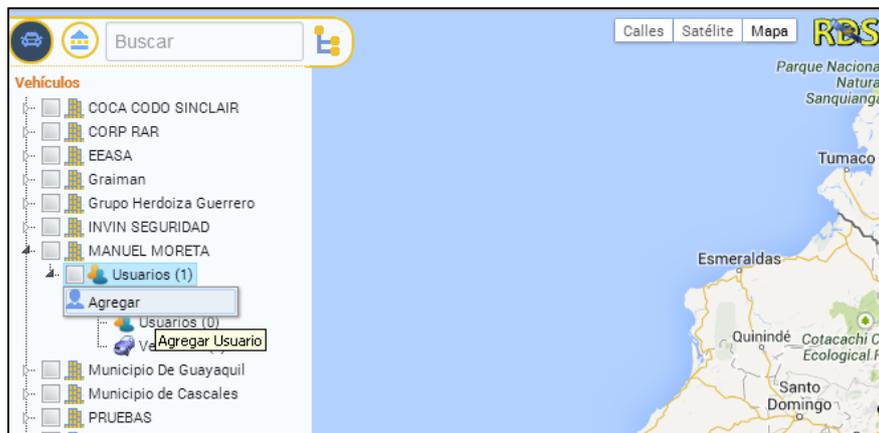
c) Agregar nuevo usuario

1. Desplegar la información de una empresa.



2. Clic derecho en la opción de usuarios.

3. Seleccionar la opción agregar.



4. Llenar los datos del formulario:

**Ingrese los Datos del Usuario**

Nombre Completo:

Usuario:

Contraseña:

Confirmar Contraseña:

Grupo:

Correo Electrónico:

Teléfono:

Dirección:

**Nombre:** Nombre del usuario a ingresar.

**Usuario:** Nombre con el que se va a autenticar.

**Contraseña:** Clave que permite junto con el usuario acceder al sistema RDS.

**Confirmar contraseña:** Volver a ingresar la contraseña.

**Grupo:** Seleccionar a que grupo que va a pertenecer el nuevo usuario, los grupos disponibles son: Admin\_RDS, Monitor RDS, SuperAdmin\_CMP, Admin\_CMP, Monitores\_CMP, Usuarios\_CMP, Responsables.

**Correo electrónico:** Correo electrónico del usuario.

**Teléfono:** Número de teléfono del nuevo usuario.

**Dirección:** Ubicación del nuevo usuario.

5. Hacer clic en el botón guardar

#### d) Editar usuario

1. Seleccionamos el usuario que se va a editar desplegando la información de una empresa tal como se presenta en la figura.



2. Clic derecho del mouse sobre el usuario.
3. Seleccionar la opción editar.



4. Editar los campos necesarios
5. Guardar cambios, hacer clic en el botón guardar

#### e) Eliminar usuario

1. Seleccionamos el usuario que se va a eliminar desplegando la información de una empresa tal como se presenta en la figura.



2. Clic derecho del mouse sobre el usuario.
3. Seleccionar la opción eliminar.

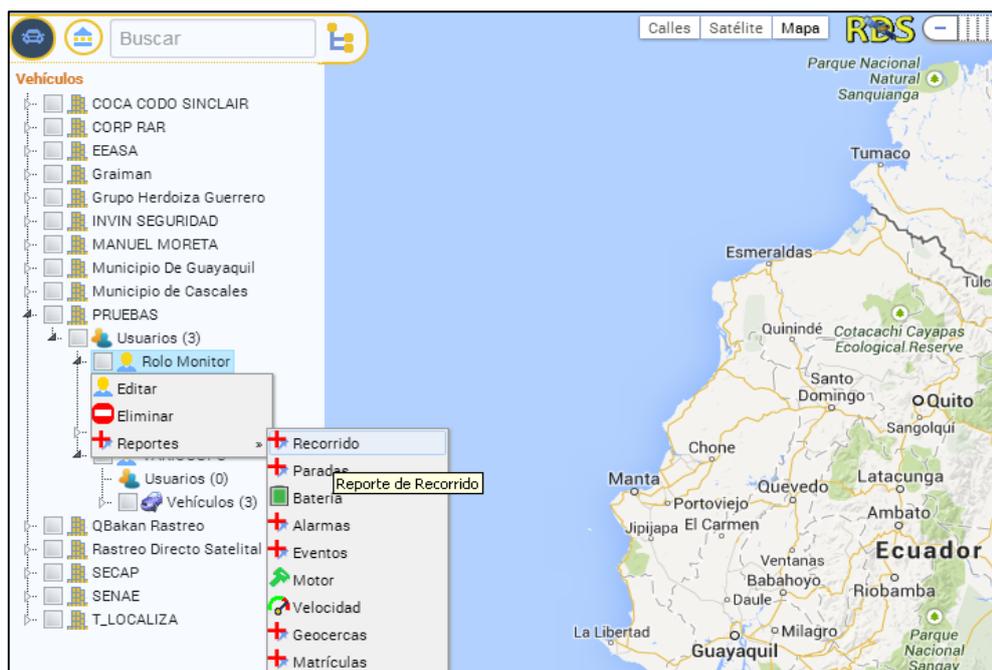
#### f) Reportes de usuarios

La siguiente pantalla muestra las empresas creadas en el sistema y de una de ellas se procede a desplegar sus respectivos usuarios. Se ha seleccionado un usuario de la cual se desea ver reportes, los reportes disponibles para cualquier usuario son:

- Recorrido
- Paradas
- Batería
- Alarmas
- Eventos
- Motor
- Velocidad
- Geo cercas
- Matrículas

### Generación de Reportes

1. Clic derecho del mouse sobre el usuario que desee generar su reporte.
2. Ir a la opción reportes
3. Seleccionar el reporte a generar

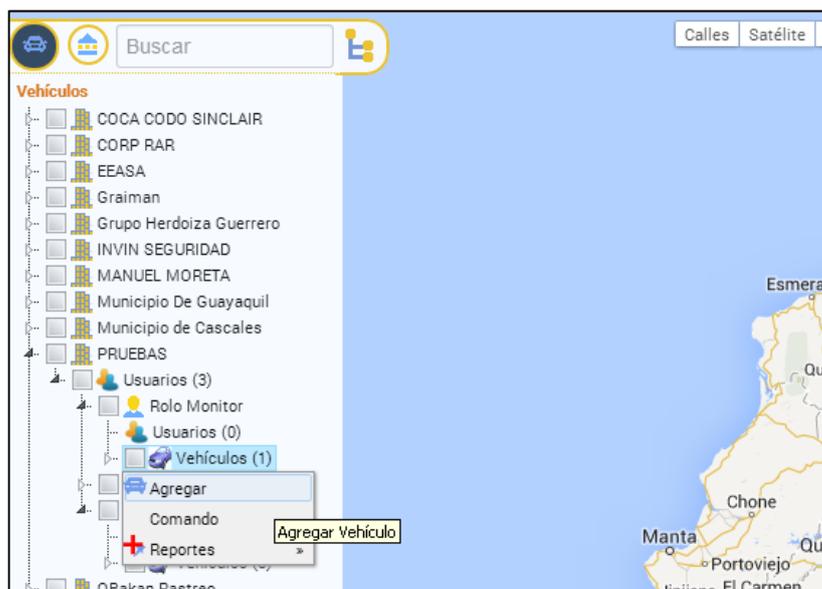


### g) Agregar vehículos

1. Desplegar la información de una empresa.
2. Clic derecho en la opción de usuarios para todos los usuarios disponibles.



3. Seleccionar el usuario a quien se le desea agregar un vehículo.
4. Desplegar las opciones de usuario y vehículo.
5. Clic derecho sobre la opción vehículo.
6. Clic derecho del mouse en la opción de vehículos y agregar.



7. Llenar los datos del formulario.

**Nombre:** Nombre con la que se identifica al vehículo.

**Alias:** Sobrenombre con la que se identificará al vehículo.

**IP de la base de datos:** Identificador único del vehículo.

**Más información:** Descripción del vehículo.

**Placa:** Placa del vehículo.

**Marca:** Marca del vehículo.

**Modelo:** Modelo del vehículo.

**Tipo:** Tipo de vehículo a registrar.

**Color:** Color característico del vehículo.

**Fecha de contrato:** Fecha desde cuándo se realiza el contrato de dicho vehículo.

**Fecha fin:** Fecha en la que concluye el contrato del vehículo.

Ingrese los Datos del Vehículo

Nombre :

Alias :

IP de la Base de Datos :

Más Información :

Placa :

Marca :

Modelo:

Tipo :

Color :

Fecha de Contrato :  Fecha de Fin :

8. Hacer clic en el botón guardar.

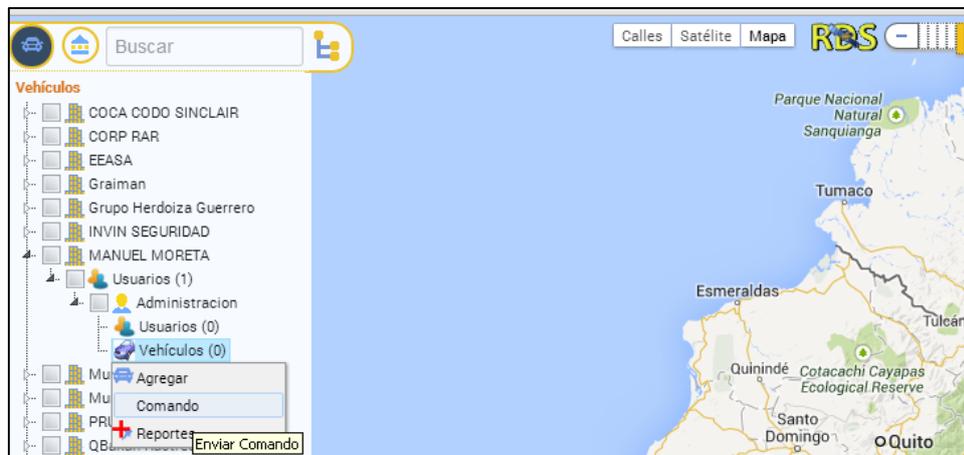
#### h) Comando vehículo

1. Desplegar la información de una empresa.
2. Clic derecho en la opción de usuarios para todos los usuarios disponibles.



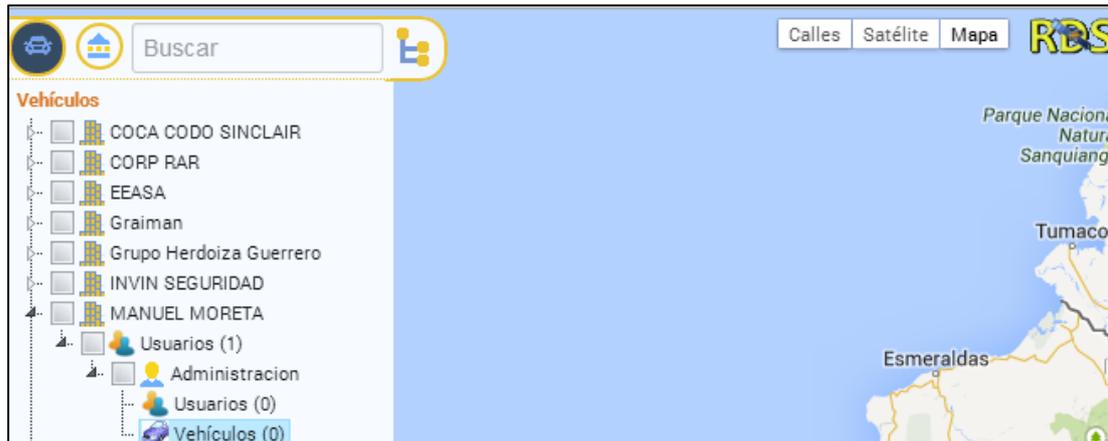
3. Desplegar las opciones de usuario y vehículo.
4. Clic derecho sobre la opción vehículo.

5. Seleccionar la opción “Comando” para generar un comando global el cual van a ejecutarlo un conjunto de vehículos.



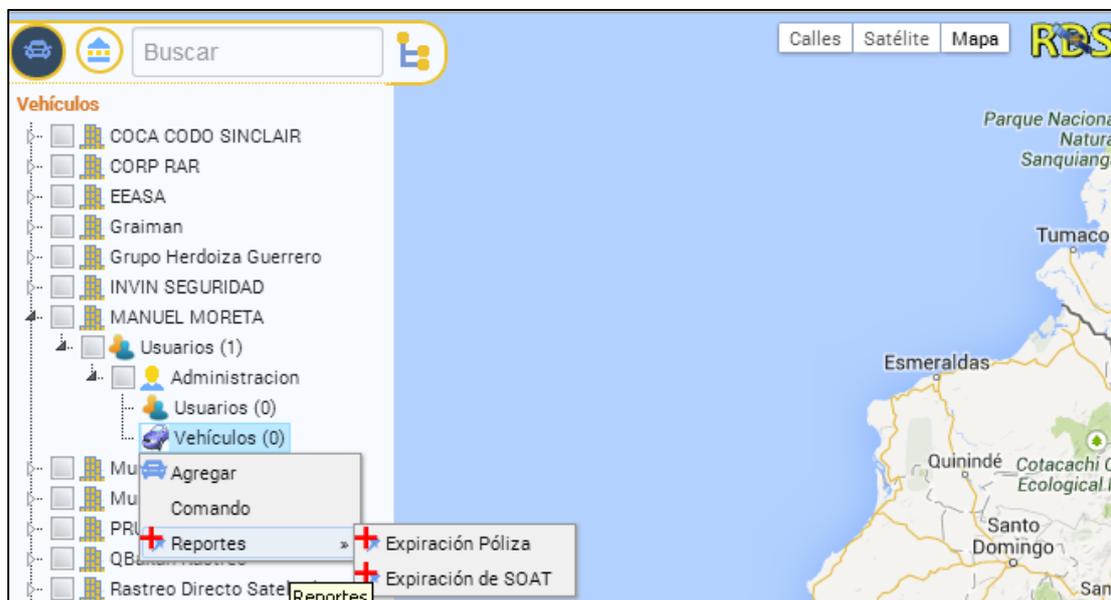
#### i) Reportes de vehículos

1. Desplegar la información de una empresa.
2. Clic derecho en la opción de usuarios para desplegar todos los usuarios disponibles en la empresa.
3. Seleccionar un usuario de la cual se desea ver un vehículo.



4. De dicha opción vehículo se puede generar Reportes.
5. Seleccionar la opción reportes y se tiene la posibilidad de generar distintos reportes tal como se presenta en la figura siguiente, los cuales son:

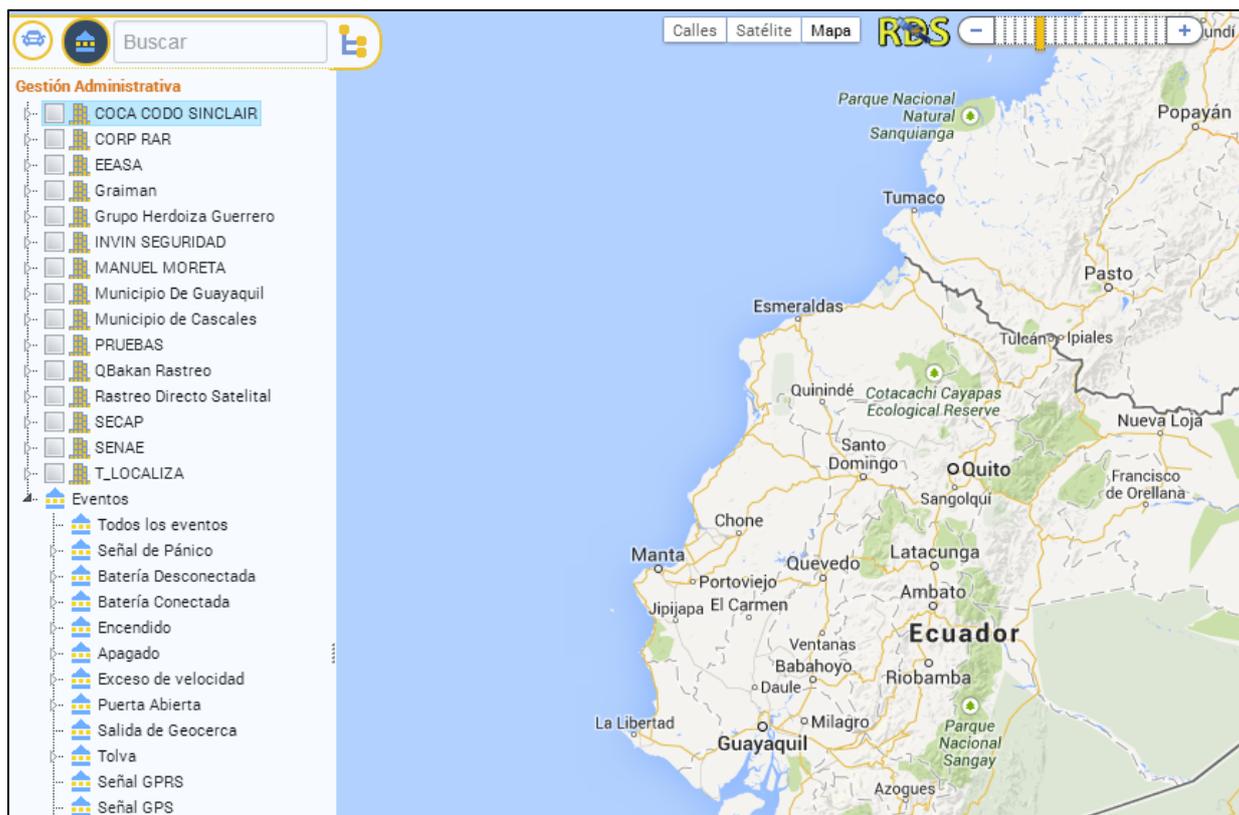
- Expiración póliza
- Expiración de SOAT



#### 4. GESTIÓN ADMINISTRATIVA

La pantalla de la gestión administrativa presentada a continuación está conformada por la gestión de Unidades Administrativas, Salvoconductos, Geocercas, Puntos de Interés y Mantenimientos de una determinada empresa, al igual para eventos que son:

- Señal de pánico
- Baterías desconectadas
- Baterías conectadas
- Encendido del vehículo
- Apagado del vehículo
- Exceso de velocidad
- Puerta abierta
- Salida de geocerca
- Tolva
- Señal GPRS
- Señal GPS
- Puerta cerrada
- Inhibidor activado



a) Crear nueva unidad administrativa

1. Seleccionar la empresa a la cual se le desea crear una nueva unidad administrativa.
2. Seleccionar la opción Unidades Administrativa.
3. Clic derecho del mouse sobre la opción de Unidades administrativas.
4. Clic izquierdo en la opción de agregar.



5. Llenar los campos del formulario que se requieren para crear una nueva unidad administrativa que son:

**Nombre:** Nombre de la unidad administrativa.

**Descripción:** Breve descripción de la nueva unidad administrativa.



El formulario 'Agregar Unidad Administrativa' tiene un título en naranja y un botón de cerrar (X) en la esquina superior derecha. Contiene dos campos de texto: 'Nombre:' con el texto de ejemplo 'Nombre de la Unidad Administrativa' y 'Descripción:' con el texto de ejemplo 'Descripción de la Unidad Administrativa'. En la parte inferior derecha hay dos botones: 'Cancelar' y 'Guardar'.

6. Clic en el botón guardar.

#### b) Crear nueva salvoconducto

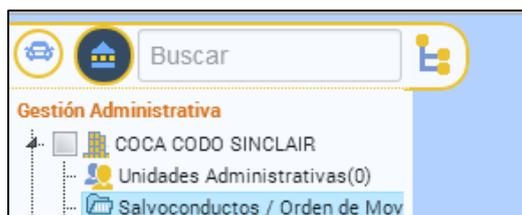
1. Seleccionar la empresa a la cual se le desea crear un nuevo salvoconducto.
2. Seleccionar la opción Salvoconducto.
3. Clic derecho del mouse sobre la opción de salvoconducto.
4. Clic izquierdo en la opción de agregar.



5. Llenar los campos del formulario.
6. Clic en el botón guardar.

#### c) Crear nueva geocerca

1. Clic derecho sobre la opción de salvoconductos.



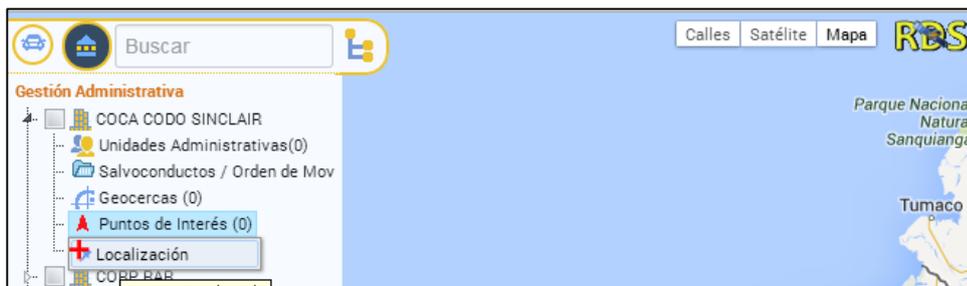
2. Seleccionar la opción de agregar.
3. Llenar el campo descripción del salvoconducto en el formulario.



4. Hacer clic en el botón Guardar

#### d) Agregar localización

1. Ubicarse en la opción punto de interés que se encuentra dentro de una empresa tal como se presenta a continuación.



2. Clic derecho del mouse
3. Seleccionar la opción localización
4. Escribir la descripción de la localización del formulario.



5. Clic en el botón guardar.

#### e) Dar mantenimiento

1. Ubicarse en la opción mantenimiento dentro de una empresa.
2. Clic derecho del mouse.
3. Seleccionar la opción mantenimiento



4. Llenar los datos del formulario.

**Nombre:** Nombre del mantenimiento que se desea realizar al vehículo.

**Realizar cada:** El mantenimiento se realizar cada cierto kilómetro, el cual se debe ingresar en éste campo.

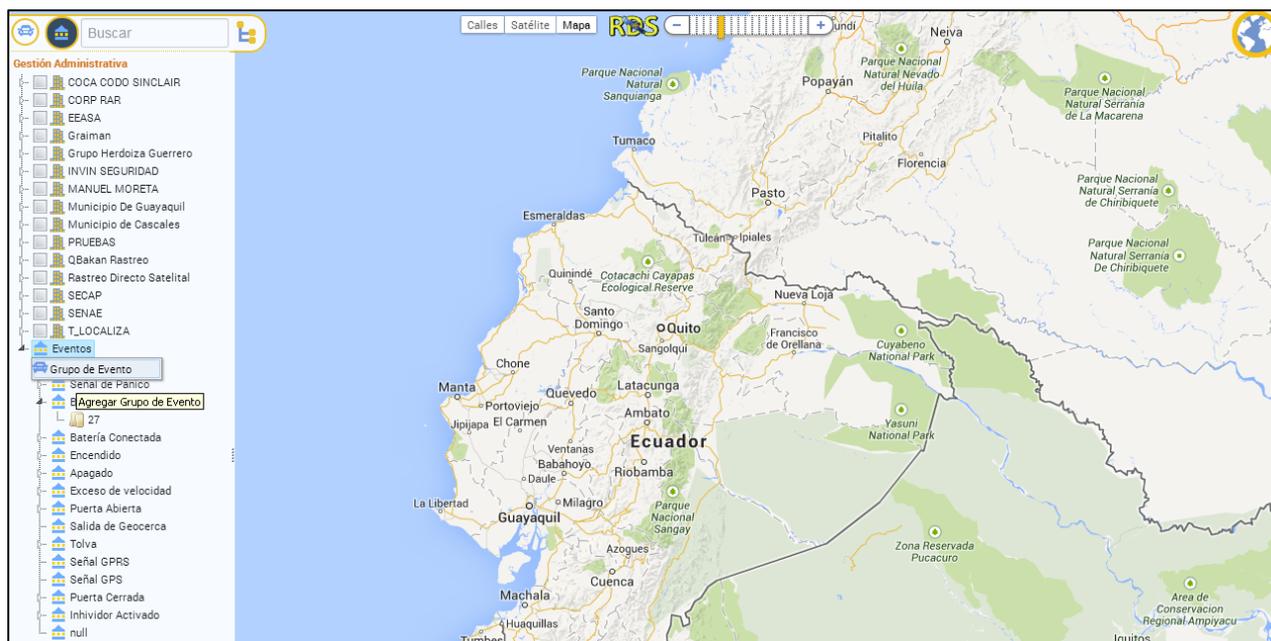
**Avisar:** Avisar antes de llegar al kilómetro que se debe realizar el mantenimiento, l cual el campo a llenar es en kilómetros.

**Recordar cada:** Para recordar que se necesita hacer un mantenimiento al vehículo escribir el kilómetro donde se desea hacer el recordatorio.

5. Clic en el botón guardar.

f) [Crear un nuevo grupo evento](#)

Pantalla que permite visualizar las empresas y eventos creados (íconos de color azul).

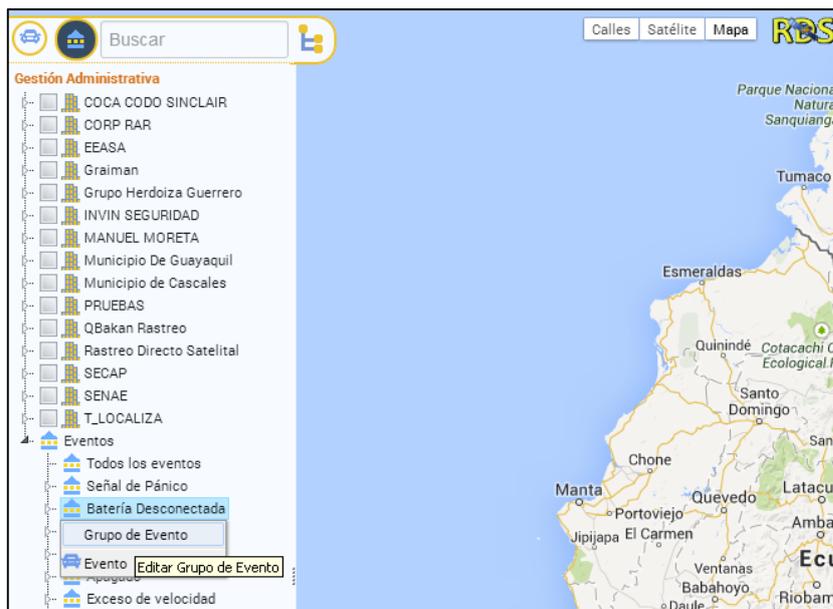


1. Ubicarse en los eventos que se presentan en la pantalla de gestión de administración ya mostrada.
2. Clic derecho del mouse sobre la opción de eventos.
3. Seleccionar la opción Grupo de evento
4. Se presenta un formulario a llenar para crear un grupo de evento, consta del campo Nombre que será el nombre del grupo del evento.

5. Clic en el botón cargar ícono para subir un ícono que representará el grupo evento a crear.
6. Por último, clic en el botón guardar.

g) Editar un grupo evento

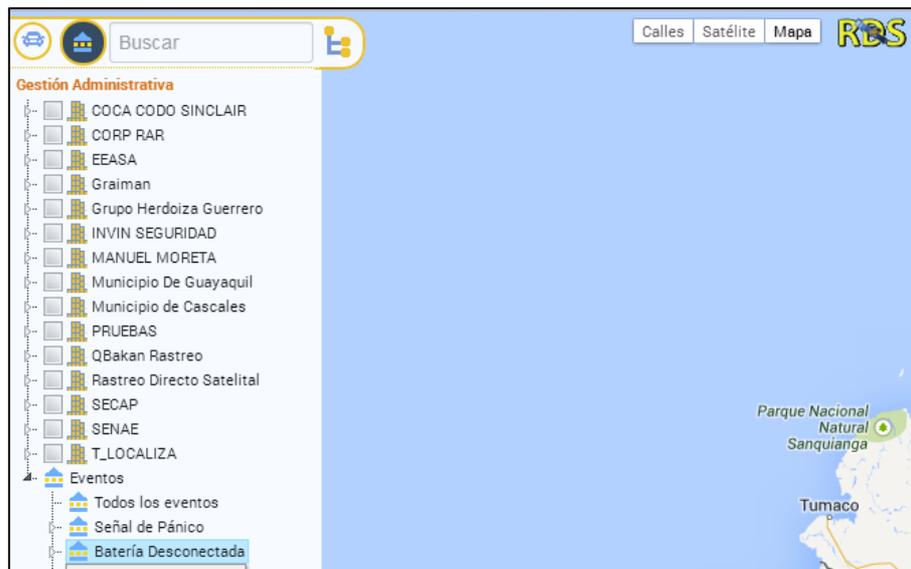
1. Desplegar la opción eventos.
2. Seleccionar un evento.
3. Ubicarse en el grupo evento a editar.
4. Clic derecho en el grupo evento a editar.



5. Seleccionar la opción grupo evento.
6. Editar los campos que se deseen.
7. Clic en el botón guardar.

h) Crear un nuevo evento

1. Desplegar la información de grupos de eventos para ver los grupos eventos creados.
2. Seleccionar un grupo evento a la cual se va a crear un nuevo evento.
3. Clic en la opción evento.



4. Escribir el código de evento del formulario.

5. Hacer clic en el botón guardar

#### 4.1. MENÚ DE OPCIONES

En la gráfica siguiente se muestra las opciones disponibles en el sistema RDS.



Éstas se encuentran en la parte inferior derecha de la página web de RDS, consta de seis opciones que son:

- Contáctenos
- Estado de los equipos
- Eventos
- Registros

- Opciones
- Estado de conexión (conectado y desconectado).

#### a) Opción Contáctenos



**Funcionalidad:** Permite contactarse desde diferentes redes sociales en las que les pueden seguir.

**Como hacerlo:** Hacer clic sobre la opción contáctenos y elegir seguir ya sea por Facebook o por tweets, además se puede dejar un comentario en caso lo desea.

#### b) Opción Estado de los equipos



**Funcionalidad:** Muestra todos los vehículos y permite ver en el mapa el vehículo que se desea y el estado en el que se encuentra mediante eventos.

**Como hacerlo:** Hacer clic sobre la opción Estado de equipos y seleccionar el vehículo de seguimiento con un solo clic sobre el vehículo.

#### c) Opción Eventos

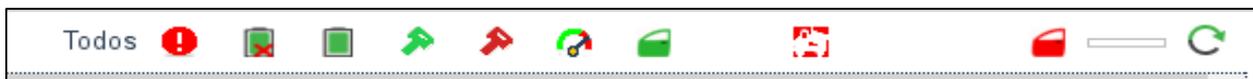


**Funcionalidad:** Muestra un registro de eventos.

**Como hacerlo:** Al hacer clic sobre la opción Eventos se muestra información de todos los vehículos con sus respectivos eventos



La siguiente pantalla muestra en forma de ícono los eventos, las cuales mediante ellas se puede filtrar el registro de eventos a mostrar en la ventana de la gráfica anterior.



La tabla siguiente permite mostrar los eventos representados por íconos y describir su funcionalidad.

Ícono	Funcionalidad
	Muestra todos los eventos
	Muestra señal de pánico
	Muestra batería desconectada
	Muestra batería conectada
	Muestra encendido
	Muestra apagado
	Muestra exceso de velocidad
	Muestra puerta abierta
	Muestra salida de geo cerca
	Muestra tolva
	Muestra señal GPRS
	Muestra señal GPS
	Muestra puerta cerrada
	Muestra nulo

#### d) Opción registro

Ícono: 

**Funcionalidad:** Muestra los registros del sistema

**Como hacerlo:** Hacer clic sobre la opción Registro.

## e) Opciones

**Ícono:** 

**Funcionalidad:** Muestra las opciones generales tal como se observa en la figura posterior.

**Como hacerlo:** Hacer clic sobre el ícono opciones del menú, se puede realizar cualquiera de las acciones disponibles en dicha pantalla.



Opciones generales

Antes de ayer       Seleccionar por fecha y hora (2)       Mostrar estelas.

Ayer

Hoy      2014-05-21      2014-05-21       Mostrar geocercas.

Ésta semana      00:00      23:59       Mostrar las localizaciones.

Semana pasada       Reproducir sonidos.

## f) Estado de conexión

**Ícono:** 

**Funcionalidad:** Permite conectar al servidor y forzar desconectado.

**Como hacerlo:** Hacer clic sobre la opción conectado al servidor.