

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIERO E INGENIERA ELECTRÓNICOS**

**TEMA:
DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO
PARA LA COMPAÑÍA DE TRANSPORTES PLANETA
TRANSPLANETA S.A. UTILIZANDO TECNOLOGÍAS
INALÁMBRICAS GPS Y GPRS**

**AUTORES:
EDISON VINICIO OJEDA CAPA
DAISY MARICELA SEMANATE TRÁVEZ**

**DIRECTOR:
MANUEL RAFAEL JAYA DUCHE**

Quito, noviembre del 2015

Cesión de derechos de autor

Nosotros Edison Ojeda, con documento de identificación N° 1718373713 y Daisy Semanate con documento de identificación N° 1721973301, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: "Diseño e implementación de un Sistema de Monitoreo para la Compañía de Transportes Planeta Transplaneta S.A. utilizando tecnologías inalámbricas GPS y GPRS", mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Daisy Maricela Semanate Trávez

1721973301



Edison Vinicio Ojeda Capa

1718373713

Quito, noviembre del 2015

Declaratoria de coautoría del docente tutor

Yo declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO PARA LA COMPAÑÍA DE TRANSPORTES PLANETA TRANSPLANETA S.A. UTILIZANDO TECNOLOGÍAS INALÁMBRICAS GPS Y GPRS realizado por Edison Vinicio Ojeda Capa y Daisy Maricela Semanate Trávez , obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerados como trabajo final de titulación.

Quito, noviembre del 2015



Manuel Rafael Jaya Duche

1710631035

Índice

Introducción	1
1 Marco conceptual	4
1.1 Condición actual de la ruta Cutuglagua – San Roque	4
1.2 Tecnologías involucradas	5
1.2.1 Sistema de posicionamiento global – GPS	5
1.2.1.1 Protocolo de comunicación GPS	6
1.2.1.2 Módulo receptor GPS u-blox neo-7m	8
1.2.2 Servicio general de paquetes por radio – GPRS	8
1.2.2.1 Protocolo de comunicación GPRS	9
1.2.2.2 Módulo GPRS/GSM sim900	9
1.2.3 Microcontrolador AVR	10
1.2.3.1 Microcontrolador atmega164P	11
1.2.3.2 Compilador bascom-AVR	12
1.2.3.3 Lenguaje de programación basic	12
1.3 Bases conceptuales para el desarrollo de la página web	13
1.3.1 Servidor web	13
1.3.1.1 Servidor web apache	14
1.3.1.2 WampServer 2.5	14
1.3.2 Lenguaje de marcas de hipertexto HTML	15
1.3.3 Hojas de estilo CCS y bootstrap	15
1.3.4 Lenguaje de programación PHP	16
1.3.5 Lenguaje de programación Javascript	17
1.3.6 API de Google Maps	17
1.3.7 SQL	17
1.3.7.1 MySQL	18
1.3.8 Servicio de alojamiento web	19
2 Diseño e implementación	20
2.1 Introducción a la propuesta del diseño	20
2.2 Modelamiento físico del sistema de geolocalización	21
2.2.1 Etapa dispositivo GPS	22
2.2.2 Etapa dispositivo GPRS	23
2.2.3 Etapa microcontrolador AVR	24
2.2.4 Desarrollo del hardware del dispositivo electrónico de geolocalización	25
2.3 Modelamiento lógico del sistema de geolocalización	27

2.3.1 Rutina de inicialización	27
2.3.2 Rutina de adquisición y procesamiento de datos	28
2.3.2.1 Subrutina adquisición y procesamiento de datos GPS.	29
2.3.2.2 Subrutina de envío de datos por medio del módulo GPRS.	31
2.4 Canal de transmisión	33
2.5 Etapa de diseño de la interfaz web	33
2.5.1 Bloque creación de la base de datos	35
2.5.2 Bloque maquetación interfaz web	36
2.5.2.1 Elementos de HTML, CSS, bootstrap	36
2.5.2.2 Elementos de Javascript y API de Google Maps	37
2.5.3 Bloque codificación de funcionalidades PHP	38
2.5.3.1 Conexión a la base de datos MySQL con PHP	38
2.5.3.2 Programación página inicio.....	38
2.5.3.3 Programación página invitado.....	41
2.5.3.4 Programación página administrador y mensaje	42
2.5.3.5 Programación página visualización de reportes del recorrido.....	43
2.6 Etapa de recopilación de información del recorrido y base de datos	45
2.7 Implementación del sistema	46
2.7.1 Implementación página web en el hosting	46
2.7.2 Implementación del circuito en el autobús.....	49
3 Pruebas y resultados	52
3.1 Prueba de margen de error en el reconocimiento de paradas	52
3.2 Pruebas de funcionamiento	55
3.2.1 Pruebas de recepción de datos GPS	55
3.2.2 Pruebas de transmisión de datos GPRS.....	56
3.2.3 Pruebas de recepción de información en la base de datos.....	57
3.3 Análisis de resultados.....	58
Conclusiones	62
Recomendaciones.....	63
Referencias.....	64
Anexos.....	65

Índice de tablas

Tabla 1.....	54
Tabla 2.....	58
Tabla 3.....	61
Tabla 4.....	61

Índice de figuras

Figura 1. Comandos NMEA	6
Figura 2. Comando GPRMC.....	7
Figura 3. Comando GPVTG	7
Figura 4. Servidor web.....	13
Figura 5. Página web.....	15
Figura 6. Etapas del sistema.....	20
Figura 7. Sistema de monitoreo	21
Figura 8. Hardware del sistema.....	21
Figura 9. GPS u-blox-7m	22
Figura 10. GPRS sim900.....	23
Figura 11. Atmega164P	24
Figura 12. Esquemático del prototipo.....	26
Figura 13. Inicialización del sistema.....	28
Figura 14. Adquisición y procesamiento de datos	30
Figura 15. Envío de datos.....	32
Figura 16. Interfaz web	35
Figura 17. HTTP Post	39
Figura 18. Ingreso a interfaces web	40
Figura 19. Almacenamiento de información.....	45
Figura 20. Página inicio	47
Figura 21. Página invitado y administrador	48
Figura 22. Página visualización reportes	49
Figura 23. Prototipo instalado	50
Figura 24. Sensores en las puertas	50
Figura 25. Cámaras instaladas.....	51
Figura 26. Área de cada parada.....	52
Figura 27. Perímetro de paradas.....	53
Figura 28. Exactitud del perímetro de paradas.....	53
Figura 29. Margen de error	54
Figura 30. Visualización de sentencias NMEA	55
Figura 31. Visualización de comandos AT	56
Figura 32. Transmisión de datos a la web.....	57
Figura 33. Base de datos	57
Figura 34. Base de dato actualizada.....	58
Figura 35. API de Google insertado en la página web.....	59
Figura 36. Reporte diario de contravenciones.....	60

Índice de anexos

Anexo 1. Distribución de pines u-blox-7m.....	65
Anexo 2. Distribución de pines sim900.....	65
Anexo 3. Distribución de pines atmega164P.....	65
Anexo 4. Funcionamiento del protocolo HTTP.....	66
Anexo 5. Módulo GPS u-blox-7m.....	66
Anexo 6. Diseño PCB del módulo u-blox-7m.....	66
Anexo 7. Módulo GPRS sim900.....	67
Anexo 8. Diseño PCB general del hardware del sistema.....	67
Anexo 9. Dispositivo de geolocalización ensamblado.....	68
Anexo 10. Vista frontal del hardware del sistema.....	68
Anexo 11. Cámaras de control.....	68
Anexo 12. Instalación de SIM CARD en módulo GPRS.....	69
Anexo 13. Ventana de información del WampServer.....	69
Anexo 14. Implementación base de datos “proyecto” con las tablas informes, rutas, y usuarios.....	69
Anexo 15. Etiquetas HTML.....	70
Anexo 16. Hosting DonWeb.....	72
Anexo 17. Archivos alojados en el hosting DonWeb.....	73
Anexo 18. Unidad de transporte 151 donde se instaló el sistema.....	73
Anexo 19. Costos del prototipo y proyección para implementación en las 145 unidades de la Compañía Planeta Transplaneta S.A.....	74

Resumen

En el presente documento se describe el proceso del diseño e implementación del hardware y software de un sistema de monitoreo de paradas de buses en la Compañía de Transportes Planeta Transplaneta para la ruta Cutuglagua San Roque en el Distrito Metropolitano de Quito usando tecnologías inalámbricas: Sistema de Posicionamiento Global (Global Positioning System, GPS) y Servicio General de Paquetes Vía Radio (General Packet Radio Service, GPRS).

Con el fin de controlar el uso indebido de las paradas de transporte público y optimizar tiempos en el recorrido realizado, este sistema utilizó un módulo GPS que permite obtener datos de la localización de la unidad de transporte que posteriormente son enviados hacia un sitio web diseñado exclusivamente para este sistema por medio de tecnología GPRS. La comunicación e interacción de los sistemas mencionados se los realizó mediante un microcontrolador Atmega164P.

Los datos alojados en el servidor web reflejaron el recorrido realizado por una unidad de la compañía, los cuales fueron analizados y evaluados para establecer un patrón de tiempos empleados, proceder de recorrido y faltas cometidas por el conductor. Estos parámetros sirvieron como indicadores para mejorar la calidad de servicio brindado por la compañía.

El correcto funcionamiento del sistema determinó que el índice de infracciones cometidas por las unidades de la compañía es elevado, lo que ocasiona un retardo en los tiempos de realización del recorrido, por lo que este sistema se presenta como una herramienta potencial para controlar y sancionar este tipo de contravenciones.

Abstract

In the present document it describes the process of the design and implementation of hardware and software of a monitoring system of bus stops, in the Transportation Company Planeta Transplaneta for the route Cutuglagua San Roque, in the Metropolitan District of Quito using wireless technologies: Global Positioning System (GPS) and General Packet Radio Service (GPRS).

In order to control improper use of public transportation stops and optimize time on the distance traveled, this system used a GPS module it provides localization data of the transport unit, which are subsequently sent to a website designed exclusively for this system through GPRS technology. The communication and interaction of the mentioned systems, was performed through microcontroller Atmega164P.

The data stored on the web server reflected the route made by unit of the Company, which were analyzed and evaluated, to establish a pattern of time used, behavior of travel and infractions committed by the driver. This parameters were used as indicators to improve the quality of service provided by the company.

The proper functioning of the system determined that the rate of infringements committed is high, causing a delay in the time of realization of route, so this system is presented as a potential tool to controlling and punishing such contravention.

Introducción

Uno de los aspectos importantes que influyen en el desarrollo de las diferentes actividades de la población es la movilidad urbana, por lo que esta debe tener una planificación adecuada que sea capaz de mitigar problemas inevitables tales como son la irregularidad geográfica en la ciudad de Quito impidiendo un buen desarrollo de una infraestructura vial, así como también la sobredemanda que existe en el uso del transporte público.

Teniendo en cuenta que Quito se ha convertido en una metrópolis, su población busca siempre optimizar el tiempo, para lo cual se ha implementado algunos planes de mejoramiento en beneficio del transporte y movilidad de toda la ciudadanía en conjunto tales como el Sistema de Regulación Pico y Placa, sin embargo, la mayoría de ciudades urbanas en el Ecuador carecen de mecanismos adecuados para regular y sancionar infracciones de tránsito producidas por faltas cometidas principalmente por la manera incorrecta de utilizar las paradas designadas para el transporte público urbano por parte de los choferes.

Según datos de la Agencia Nacional de Tránsito las infracciones mencionadas anteriormente son la mayor causa del caos vehicular, debido a que los buses en ocasiones se detienen o circulan a velocidades menores de las normales, cortando así la velocidad de desplazamiento de los demás vehículos, provocando una baja en la capacidad de la vía que se determina por la velocidad y el tiempo de ocupación de un carril cuando un vehículo se detiene, o por el contrario exceden el límite de velocidad permitida. Estas contravenciones son penadas por la Ley de Transporte Terrestre, Tránsito y Seguridad Vial en el artículo 141.

Además de problemas de caos vehicular, existen inconvenientes para controlar la frecuencia de cada unidad y tiempo empleado en el recorrido realizado puesto que actualmente se utilizan máquinas de registro que imprimen en una tarjeta la hora de paso del automotor, siendo este un sistema obsoleto y poco eficiente, es por ello que surge la necesidad de proveer una solución que ayude a contrarrestar dicho problema como es el caso de los sistemas de comunicación globales GPS y GPRS.

En virtud de aquello es importante controlar que los buses urbanos circulen sin exceder el límite de velocidad, con las puertas cerradas y únicamente puedan dejar y recoger pasajeros en las paradas establecidas, por lo que el Sistema de Posicionamiento Global (GPS) se encargará de proporcionar la localización del automotor y de esta manera se obtendrá la ubicación exacta donde realice las paradas, además de la velocidad a la que se moviliza. Junto con esto se adecuará una cámara y un sensor en las puertas que detecten si estas se abren en lugares que no correspondan a las paradas, entonces se enviará información sobre estos eventos utilizando el Servicio General de Paquetes Vía Radio (GPRS) instalado en el autobús hacia la web, al terminar toda la ruta se obtendrá información del recorrido realizado. Todos los datos transmitidos se recopilarán en un software que se diseñará para adquirir los datos del sistema GPRS de modo que sean mostrados y administrados amigablemente.

El diseño e implementación del presente proyecto se describe en tres capítulos, que tratan sobre el control del uso indebido de las paradas de transporte público y optimización de los tiempos empleados en cada recorrido.

Los capítulos se dividen según su contenido, refiriéndose a una descripción, diseño, desarrollo y evaluación, guardan una secuencia y relación entre sí en cuanto a su forma y fondo para describir de una manera real y correcta el funcionamiento del prototipo.

El análisis y la conceptualización de parámetros involucrados se realizarán en el capítulo uno, donde se hará una recopilación de datos de la condición actual de la Compañía Planeta Transplaneta en ruta Cutuglagua – San Roque, además de profundizar en definiciones tales como dispositivos GPS, GPRS, microcontroladores, páginas web y bases de datos.

La parte de diseño e implementación se encuentra en el capítulo dos, en el cual se describe la integración de los componentes de hardware y software del sistema, y además se desarrollan los modelamientos físico y lógico del sistema de geolocalización y la interfaz de comunicación web.

Por último está el capítulo tres, donde se realizan pruebas de funcionamiento del prototipo y se hace una recopilación de resultados a través de tablas y gráficos descriptivos, que determinarán la viabilidad del proyecto.

1 Marco conceptual

En este capítulo, se describirá brevemente la condición actual del recorrido realizado por una unidad de transporte de la Compañía Planeta Transplaneta S.A. en la ruta Cutuglagua – San Roque y las bases teóricas que servirán de sustento para el desarrollo del hardware y software del sistema de monitoreo.

1.1 Condición actual de la ruta Cutuglagua – San Roque

La Compañía de Transportes Planeta Transplaneta S.A. ubicada en la calle Emilio Uzcategui Oe-6121 y Alonso Pablos Ciudadela Martha Bucaram, brinda sus servicios a los habitantes del sur de la ciudad de Quito desde hace 13 años. Actualmente cuenta con 145 unidades de transporte que cubren 7 rutas en el centro-sur de la ciudad.

Una de las rutas de servicio de transporte es Cutuglagua-San Roque, la cual es recorrida cada mes y medio durante una semana, normalmente 23 unidades cubren esta ruta puesto que es la de mayor trayecto, ya que tiene una distancia aproximada de 21 Km. Un autobús tarda 1 hora 5 minutos en condiciones normales y 1 hora 15 minutos con tráfico vehicular en trasladarse desde Cutuglagua hasta San Roque, ocupando aproximadamente 2 horas y 30 minutos en el recorrido total (Cutuglagua-San Roque-Cutuglagua).

Para identificar las paradas establecidas de la ruta y la utilidad que se le da a las mismas se realizó un reconocimiento del trayecto, con lo que se identificó un total de 81 paradas que un bus realiza durante el recorrido, divididas en 41 de ida y 40 de vuelta, tomando en cuenta que 3 paradas de vuelta (Centro Comercial El Recreo) hacen referencia a una misma parada. Los datos obtenidos después del recorrido realizado indican que la unidad en análisis realizó un total de 123 paradas en todo el recorrido para recoger o dejar pasajeros, de las cuales 42 resultaron mal ejecutadas equivalentes al 34% del recorrido total, con lo cual incurren en la contravención penada por la Ley de Transporte Terrestre, Tránsito y Seguridad Vial en el artículo 141.

Adicionalmente se observó que en algunas ocasiones el automotor excedió los límites de velocidad permitidos llegando hasta los 80 km/h, principalmente a lo largo de la

Av. Pedro Vicente Maldonado en los trayectos comprendidos entre la entrada a Cutuglagua hasta la altura de la Hacienda Hato Verde y entre la calle Joaquín Gutiérrez hasta la calle Moraspungo.

1.2 Tecnologías involucradas

Para un óptimo funcionamiento de un sistema de geolocalización y monitoreo, es necesaria la integración de diferentes tecnologías tanto de hardware como de software, por tal razón, se realiza un análisis de sistemas de comunicación y de control involucrados en la realización del proyecto, incluyendo comandos de gestión, lenguajes de programación y programas de compilación.

En ítems siguientes se describe brevemente el funcionamiento de sistemas inalámbricos GPS, GPRS, dispositivos de control (microcontroladores), lenguaje de programación Basic y comandos AT.

1.2.1 Sistema de posicionamiento global – GPS

Desde sus inicios a principios de los 70 los sistemas GPS han tenido diversas aplicaciones, siendo la localización, navegación y cartografía las más comunes tanto para el Servicio Preciso de Posición – PPS el cual es de gran exactitud y de uso exclusivo militar, como para el Servicio Estándar de Posición – SPS disponible para todos los civiles a nivel mundial.

Al utilizar el servicio SPS la recepción de las señales GPS no tienen costo alguno, siendo esta la gran ventaja de este sistema, por tal razón, existen varias compañías que fabrican receptores GPS de todo tipo, dependiendo de las aplicaciones.

Algunas características que diferencian a los distintos receptores GPS son la rapidez con la que pueden conectarse a la cantidad de satélites necesarios para poder calcular su posición (Start Time) o la precisión del módulo ante la presencia de obstáculos.

La alimentación para el funcionamiento de los módulos GPS en algunos casos es 3.3V, mientras que en otros se necesitan 5V. El consumo de corriente también varía de 30mA a 80mA dependiendo del módulo.

1.2.1.1 Protocolo de comunicación GPS

Con la proliferación de mecanismos de navegación y localización, muchas compañías dedicadas a las telecomunicaciones han empezado a fabricar diversos módulos GPS, con el fin de estandarizar estos sistemas se han creado protocolos que normalicen la transmisión de datos, siendo el NMEA el más utilizado.

Las siglas NMEA identifican a la Asociación de Marina y Electrónica Nacional de los Estados Unidos, quienes son los precursores en establecer un código normalizado de comunicación entre equipos y dispositivos de navegación, para permitir a los barcos transmitir datos a un piloto automático respecto a una trayectoria predeterminada, pero ahora también son utilizados en la navegación terrestre.

En la actualidad el protocolo más utilizado para la comunicación del módulo GPS con un controlador es el NMEA-183, más conocido como GPS Venus, protocolo que utiliza un código cifrado en el encabezado de la trama de datos GPS. En la figura 1, se muestra las sentencias NMEA más utilizadas.

Comandos NMEA	
Option	Description
GGA	Time, position and fix type data.
GSA	GPS receiver operating mode, active satellites used in the position solution and DOP values.
GSV	The number of GPS satellites in view satellite ID numbers, elevation, azimuth, and SNR values.
RMC	Time, date, position, course and speed data. Recommended Minimum Navigation Information.
VTG	Course and speed information relative to the ground.

Figura 1. Lista de sentencias de salida NMEA
Fuente: (GlobalTop Technology Inc., 2011)

La sentencia RMC permite obtener datos de posición (latitud-longitud), fecha y la hora mundial, la cual se transmite con una cadena de caracteres de la siguiente manera:

\$GPRMC,064951.000,A,2307.1256,N,12016.4438,E,0.03,165.48,260406,3.05,W,A*2C

Los datos recibidos mediante la sentencia GPRMC se los interpreta de acuerdo a la tabla mostrada en la figura 2.

Comando GPRMC

Name	Example	Units	Description
Message ID	\$GPRMC		RMC protocol header
UTC Time	064951.000		hhmmss.sss
Status	A		A=data valid or V=data not valid
Latitude	2307.1256		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12016.4438		dddmm.mmmm
E/W Indicator	E		E=east or W=west
Speed over Ground	0.03	knots	
Course over Ground	165.48	degrees	True
Date	260406		ddmmyy
Magnetic Variation	3.05, W	degrees	E=east or W=west (Need GlobalTop Customization Service)
Mode	A		A= Autonomous mode D= Differential mode E= Estimated mode
Checksum	*2C		
<CR> <LF>			End of message termination

Figura 2. Datos de configuración de la sentencia GPRMC
Fuente: (GlobalTop Technology Inc., 2011)

La velocidad a la que se desplaza un móvil en tiempo real se la obtiene utilizando la sentencia VTG de la siguiente forma: \$GPVTG,165.48,T,,M,0.03,N,0.06,K,A*37. Los distintos parámetros que ofrece GPVTG se muestran en la figura 3.

Comando GPVTG

Name	Example	Units	Description
Message ID	\$GPVTG		VTG protocol header
Course	165.48	degrees	Measured heading
Reference	T		True
Course		degrees	Measured heading
Reference	M		Magnetic (Need GlobalTop Customization Service)
Speed	0.03	knots	Measured horizontal speed
Units	N		Knots
Speed	0.06	km/hr	Measured horizontal speed
Units	K		Kilometers per hour
Mode	A		A= Autonomous mode D= Differential mode E= Estimated mode
Checksum	*06		
<CR> <LF>			End of message termination

Figura 3. Datos de configuración de la sentencia GPVTG
Fuente: (GlobalTop Technology Inc., 2011)

1.2.1.2 Módulo receptor GPS u-blox neo-7m

Para recibir datos de geolocalización existe una amplia gama de módulos, que varían en costo y tamaño de acuerdo a la precisión de la triangulación satelital necesaria en cada aplicación. Entre los receptores GPS más ofertados en el mercado está el U-blox neo-7m, que es capaz de recibir hasta 56 canales para lograr una precisión de posicionamiento de 2.5 metros, ideal para aplicaciones en automotores.

Entre las características relevantes del U-blox neo-7m se tienen encontradas en (u-blox, 2014, pp. 6,13), se tiene:

- Tiempo de adquisición para arranque en caliente 1 segundo y arranque en frío 30 segundos.
- Velocidad de comunicación serial 9600 bps
- Voltaje de funcionamiento de 2.7 V a 5.0 V
- Corriente de trabajo 35mA
- Protocolo de comunicación por defecto NMEA
- Velocidad de actualización de 1Hz a 10Hz

La distribución de los pines del módulo U-blox se muestra en el Anexo 1.

1.2.2 Servicio general de paquetes por radio – GPRS

Los sistemas para comunicaciones móviles (GSM) de segunda generación surgieron para migrar los servicios de telefonía móvil analógicos de primera generación a comunicaciones digitales. Una vez implantado el sistema GSM se plantea la necesidad de ofrecer un mejor servicio de datos en movilidad. Con este objetivo nace el GPRS, que además de servicio de datos permite la transmisión de mensajes cortos (SMS) y el protocolo de acceso inalámbrico (WAP). La tarificación de tráfico de datos se realiza por volumen (Rodrigo & Santiago, 2002, pp. 34-35).

GPRS es de generación 2.5 capaz de transmitir datos con velocidades típicas de 54 kbit/s en el enlace descendente y 9,6 kbit/s en el ascendente, a pesar de que su velocidad de transmisión teórica es de 171,2 kbit/s. Existen 14 bandas de frecuencias definidas para todos los servicios GSM, siendo las bandas de 850 MHz y 1900 MHz las utilizadas

por GPRS en América del Sur, pero en Ecuador se utiliza únicamente la frecuencia de 850 MHz.

1.2.2.1 Protocolo de comunicación GPRS

Con el objetivo de estandarizar la configuración y parametrización de módems se crean los comandos Hayes, llamados así en honor a su creador Dennis Hayes. A esta codificación también se la conoce como comandos AT, a razón de que estas siglas preceden a todas las instrucciones de Hayes. Los caracteres AT se los utiliza como prefijos para indicar atención.

A pesar que la finalidad de los comandos AT es la comunicación con módems, la telefonía móvil GSM ha adoptado este lenguaje para comunicarse con sus distintos terminales, teniendo comandos específicos que sirve de interfaz para configurar y proporcionar instrucciones a sus equipos (Bluehack, 2005).

Este tipo de lenguaje es independiente del canal de comunicación a través del cual estos comandos son enviados, ya sea cable serial, radiofrecuencia, etc. El conjunto de instrucciones AT son utilizadas entre otras para realizar llamadas de voz, leer y escribir mensajes de texto SMS, pero el documento se enfocará en la transmisión y recepción de datos. Algunas de las sentencias AT son únicas para cualquier aplicación como son la inicialización del equipo y otras que son específicas para la transmisión-recepción de datos.

1.2.2.2 Módulo GPRS/GSM sim900

Los Equipos Locales del Cliente (CPE) son unidades terminales asociadas a equipamientos de telecomunicaciones localizadas en el lado del suscriptor, que en el caso de la telefonía móvil son los celulares, tablets o módems de datos, los cuales traen incorporados de fábrica funciones de interacción con las diferentes tecnologías móviles como GSM, UMTS, LTE, etc., no obstante para realizar aplicaciones de control y seguimiento en la industria se han creado módulos de trabajo tales como el Sim900.

El módulo Sim900 fue diseñado para trabajar con tecnologías GSM y GPRS en las bandas de frecuencia de 850, 900, 1800 y 1900 MHz, controlado y configurado por medio de un puerto serial tal como se muestra en la distribución de pines del Anexo 2.

Como características importantes se tiene que el voltaje de operación es de 3.1 a 4.8 voltios con un consumo de 15 mA en modo sleep y el control se realiza a través de comandos AT. Las sentencias utilizadas en este proyecto se describen a continuación:

- AT.- comprueba la disponibilidad del dispositivo.
- AT+SAPBR.- ajusta parámetros para aplicaciones IP, tales como información de APN, usuario y contraseña de la radio base móvil.
- AT+HTTPINIT.- inicializa servicios HTTP del puerto 80.
- AT+HTTTPARA.- envía parámetros HTTP hacia una url de destino.
- AT+HTTPACTION.- selecciona el método de envío de datos a la web ya sea por medio de POST, GET o HEAD.
- AT+CCLK.- utilizado para obtener la hora y fecha de la red de telefonía móvil.

1.2.3 Microcontrolador AVR

El microcontrolador al ser un computador de baja escala integra mecanismos de procesamiento, memoria RAM, memoria EPROM y líneas de entrada-salida, características ideales para manejo y control de procesos que integran dispositivos electrónicos.

Varias empresas se dedican a la fabricación de microcontroladores tales como Intel o Freescale, los cuales son de utilidad específica en su gran mayoría, mientras que empresas como Atmel o Microchip son los encargados de producir microcontroladores que se adapten a las necesidades de los usuarios.

Los microcontroladores AVR de Atmel y los PICs de Microchip en sus distintas presentaciones poseen características similares de acuerdo a la capacidad de procesamiento y almacenamiento de datos. En cuanto a la velocidad de procesamiento por ciclo de máquina los AVR tienen ventaja sobre los PICs, debido a que muchas instrucciones se ejecutan en dos ciclos de máquina y los PICs necesitan cuatro ciclos de reloj para cada ciclo de máquina, mientras que los AVR solo necesitan uno, cualidad

que permite seguir leyendo la siguiente instrucción y poniéndola en cola mientras se ejecuta la instrucción inicial, por esta razón es ideal para manejar dos puertos seriales al mismo tiempo, que en este caso son los módulos GPS y GPRS.

Como característica adicional por parte de los micros de Atmel se tiene que es posible programar el micro mientras está en funcionamiento utilizando el puerto serial con comunicación SPI.

1.2.3.1 Microcontrolador atmega164P

El Atmega 164P es un microcontrolador RISC (instrucciones de complejidad reducida) de 8 bits fabricado por Atmel perteneciente a la familia de AVR's Atmega 16. Tiene incorporado en hardware dos puertos seriales con comunicación de receptor/transmisor asincrónico universal (Universal Asynchronous Receiver-Transmitter, UART), capaces de controlar y monitorear simultáneamente los receptores ublox-7m y Sim900, motivo por el cual se adapta perfectamente a este proyecto.

Otros parámetros importantes para el diseño extraídos de la hoja de datos técnicos de Atmel (2015) son:

- 16 Kbyte de FLASH auto programable en sistema.
- 512 bytes de EEPROM.
- 1K bytes de SRAM Interna.
- Programación de FLASH, EEPROM, fusibles y bits de bloqueo a través de la interfaz JTAG.
- 131 instrucciones. La mayoría de un solo ciclo de reloj de ejecución.
- 32 registros de trabajo de 8 bits para propósito general.
- Dos puertos Seriales USART Programables.
- Interfaz Serial SPI maestro-esclavo.
- Oscilador interno de 20 MHz.
- Voltaje de funcionamiento de 2.7 a 5.5V.
- Corriente máxima de 200mA en los terminales VCC y GND.

La distribución de pines del ATMEGA 164P se muestra en el Anexo 3.

1.2.3.2 Compilador bascom-AVR

Los microcontroladores al igual que los microprocesadores, computadores, circuitos integrados programables, etc., solamente interpretan sentencias binarias únicas para cada máquina llamadas “código de máquina”.

Con el propósito de mejorar la eficiencia de programación se han desarrollado lenguajes de programación de bajo y de alto nivel que reemplazan el código binario por instrucciones. El lenguaje de bajo nivel está representado por Assembler, mientras que en lenguajes de alto nivel se puede mencionar C, C++, Java, Basic entre otros. El código realizado con sentencias de alto o bajo nivel no lo interpreta el microcontrolador y por ende debe ser traducido a lenguaje de máquina utilizando programas llamados compiladores.

Bascom-AVR es un software compilador de alto nivel desarrollado por MCS electronics que permite escribir el código del programa en lenguaje BASIC donde incluye al Atmega 164P en sus librerías y que además permite:

- Escribir programas en Basic.
- Traducir estos programas a código de máquina en la PC.
- Simular el código compilado.
- Cargar el código compilado en el microcontrolador AVR Atmel.

Bascom-AVR genera un archivo .bas antes de que el programa sea compilado y al compilar el programa se obtiene un archivo .hex, archivo que se carga en el microcontrolador.

1.2.3.3 Lenguaje de programación basic

Basic es un lenguaje de programación de alto nivel debido a que expresa el algoritmo de una manera adecuada a la capacidad cognitiva humana. Su parecido con el álgebra elemental es muy útil para resolver problemas de ingeniería, además sus instrucciones coinciden con el nombre en inglés de la operación que se debe llevar a cabo, como es el caso de la instrucción PRINT que sirve para escribir o visualizar un valor (Martín Cuenca, Angulo Uzategui, & Angulo Martínez, 2003, pp. 485-487).

1.3 Bases conceptuales para el desarrollo de la página web

Para llevar a cabo el diseño de la página web del proyecto de manera eficiente y sencilla es necesario utilizar herramientas informáticas, generalmente de edición, compilación y otras utilidades, las cuales serán descritas a continuación.

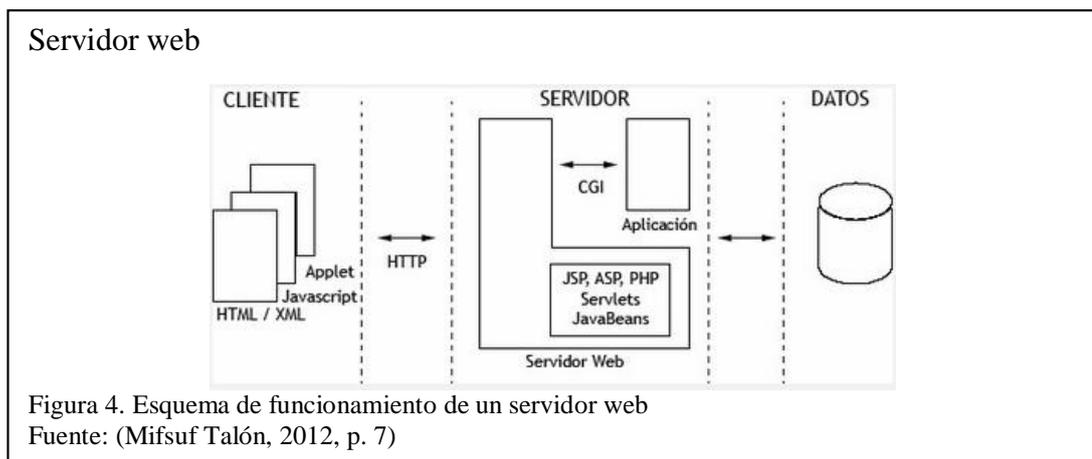
1.3.1 Servidor web

A cualquier dispositivo que responde a una solicitud de un cliente se le denomina servidor. “Un servidor web es un programa que tiene como misión atender y responder las peticiones que recibe desde los navegadores, en base al protocolo HTTP” (Giménez Guzmán, López Merayo, & Moreno Martín, 2012, pág. 41).

En la figura 4, se muestra el esquema de funcionamiento de un servidor web, en la cual se observa los tres elementos básicos que componen las peticiones web dinámicas:

- El cliente que hace la petición
- El servidor que atiende la petición
- Y los datos solicitados mediante la petición

(Mifsuf Talón, 2012, p. 7)



Un servidor web opera mediante el protocolo HTTP (Protocolo de transferencia de hipertexto), el cual está diseñado para transferir hipertextos, páginas web o páginas HTML: textos complejos con enlaces, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. HTTP es un protocolo cliente/servidor

cuyo funcionamiento está basado en el envío de mensajes. Cuando se establece una conexión HTTP se tiene dos tipos de mensajes: una petición por parte del cliente y una respuesta por parte del servidor. En el Anexo 4 se muestra el funcionamiento del protocolo.

Los métodos de petición más comunes en HTTP son GET y POST. Con el método GET los datos de un formulario se concatenan a la URL, mientras que con POST los datos se cargan directamente al servidor debido a que por su magnitud no es posible colocarlos a continuación de la URL.

1.3.1.1 Servidor web apache

Existen multitud de servidores web, pero los dos más conocidos e importantes por sus niveles de implantación son Apache e IIS (Internet Information Service). Apache es el servidor web de software libre más utilizado, de código abierto y con características particulares descritas por Mifsuf (2012):

- Es multiplataforma ya que está disponible para diferentes plataformas como GNU/Linux, Windows, MacOS.
- Flexible por trabajar con el estándar HTTP/1.1 y con la mayor parte de las extensiones web que existen en la actualidad, como son los módulos PHP, SSL, CGI, SSI, proxy.
- Estable puesto que su robustez impide cambios inesperados en el servidor (p. 12).

1.3.1.2 WampServer 2.5

En este proyecto se utilizará un paquete llamado WampServer versión 2.5, como entorno de desarrollo web para Windows. Este software dispone de un panel de control muy completo que integra herramientas como Apache 2.4.9, PHP 5.5.12 y la base de datos MySQL 5.6.17, entre otras, con las cuales se crearán aplicaciones web. Además contiene PHPMyAdmin 4.1.14 para administrar fácilmente las bases de datos.

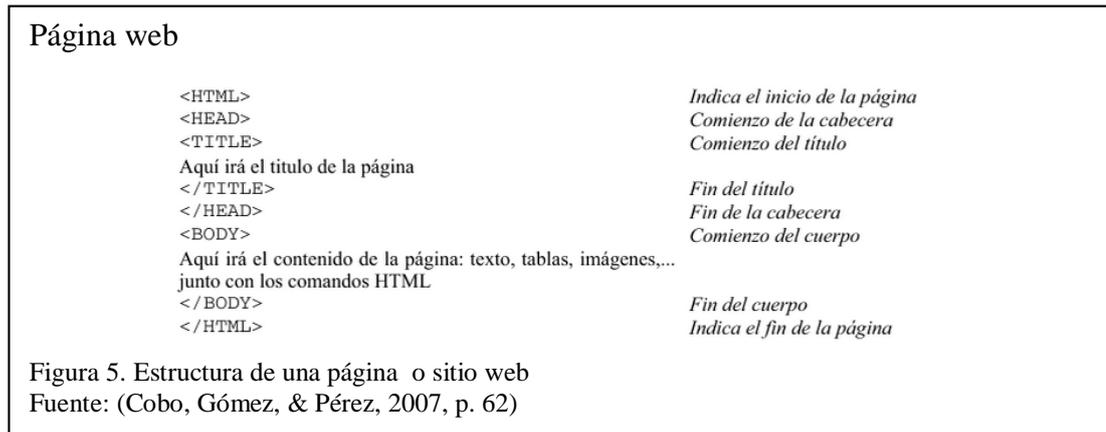
Sánchez Morales (1999) detalla algunas de las funcionalidades del WampServer 2.5:

- Gestiona los servicios de Apache y MySQL.

- Permite instalar y cambiar de versiones de Apache, MySQL y PHP.
- Administra la configuración de servidores y sus registros de acceso.
- Permite un acceso local o global.
- Permite acceder a sus archivos de configuración.
- Tiene incluido la aplicación PhpMyAdmin que mediante una interfaz de usuario programado en PHP permite gestionar las bases de datos (p. 86).

1.3.2 Lenguaje de marcas de hipertexto HTML

Una página web independientemente de su utilidad, basa su desarrollo en una secuencia estructurada de texto. El lenguaje de marcas de hipertexto HTML está compuesto por una serie de etiquetas que el navegador interpretará de una u otra forma para mostrar los distintos contenidos por pantalla. Utiliza etiquetas que aparecen especificadas a través de corchetes o paréntesis angulares: < y >, donde para finalizar una etiqueta se utiliza la siguiente nomenclatura </y>. En la figura 5, se muestra como está estructurada una página web:



HTML permite incrustar ciertos códigos que se conocen como scripts. Entre los más conocidos y utilizados son JavaScript y PHP, los cuales brindan instrucciones específicas a los navegadores que se encargan de procesar el lenguaje.

1.3.3 Hojas de estilo CCS y bootstrap

La estructura HTML está limitada a la hora de aplicarle forma a un documento. Por lo que la solución a este inconveniente reside en la separación por un lado de la

información y por otro lado el formateo de dicha información, lo que se traduce como CSS hojas de estilo en cascada, tecnología que hace posible definir el estilo visual de un sitio entero sin necesidad de hacerlo etiqueta por etiqueta, para cada una de las páginas (Ferrer Martínez, 2014, p. 283).

Uno de los frameworks CSS más famosos y utilizados es Bootstrap, ya que muchos sitios web están creados con el mismo. Permite dar forma a un sitio web mediante el uso de librerías CSS que incluyen menús desplegables, tipografías, botones, cuadros entre otros elementos de diseño que pueden ser usados en cualquier sitio web. Tiene como particularidad adaptar en su totalidad la interfaz de usuario a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Otra característica que se puede destacar es que tiene añadido un sistema GRID que permite diseñar utilizando un GRID de 12 columnas donde se plasma el contenido, haciendo más fácil e intuitivo el desarrollo web responsive.

1.3.4 Lenguaje de programación PHP

Las páginas web creadas sólo mediante HTML generan una información permanente, para construir una página web dinámica, es decir, aquella cuyo contenido no es el mismo siempre sino que pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc., se puede hacer uso de PHP (Preprocesador de hipertexto), este es un lenguaje de programación de código abierto usado principalmente en interpretación del lado del servidor y diseñado para la creación de este tipo de páginas. El código PHP puede ser incrustado en uno o más documentos HTML estándar utilizando “etiquetas” o delimitadores especiales, también puede ser combinado con JavaScript.

Los scripts de PHP se ejecutan más rápido que los scripts creados en otros lenguajes y pueden trasladarse entre plataformas como UNIX, Microsoft Windows, Mac OS u otras. A pesar de que el código fuente de PHP es gratuito a través de Internet, cada vez tiene más seguidores, debido a que soporta una amplia gama de diferentes bases de datos, entre las cuales se incluyen MySQL, PostgreSQL, Oracle y Microsoft SQL Server (Vaswani, 2010, pp. 6,7).

1.3.5 Lenguaje de programación Javascript

Este lenguaje de programación funciona del lado del cliente, los navegadores web son los encargados de interpretar estos códigos. No requiere ser compilado, además fue diseñado para añadir interactividad a las páginas HTML y validar formularios antes de ser enviados, evitando así la sobrecarga de información en el servidor.

Con el surgimiento de PHP como lenguaje del lado del servidor y Javascript del lado del cliente, surgió Ajax (Asynchronous Javascript And XML). El mismo no es un nuevo lenguaje de programación sino es una técnica para crear aplicaciones web interactivas, utilizando los estándares ya existentes.

1.3.6 API de Google Maps

El dinamismo de Google Maps está basado en HTML, CSS, JavaScript los cuales pueden trabajar juntos. Los mapas que se cargan en el fondo a través de peticiones ejecutadas por la tecnología de AJAX son solo imágenes que se insertan en un <div> en la página HTML. Entretanto que se navega en el mapa, el API envía información acerca de las nuevas coordenadas y los niveles de “zoom” del mapa a través de AJAX y esto devuelve las imágenes.

Los archivos de JavaScript que contienen las clases, métodos y propiedades que se utilizan para el comportamiento de los mapas constituyen el API de Google Maps.

1.3.7 SQL

En el desarrollo de aplicaciones web las bases de datos desempeñan un papel esencial, ya que se ocupan del almacenamiento de datos relacionados entre sí, mediante lenguajes de consulta.

- SQL es el lenguaje de consulta estructurado utilizado para el acceso a bases de datos relacionales. Este lenguaje permite definir la estructura de las tablas que componen la base de datos, y a su vez efectuar consultas y realizar operaciones

como inserción, borrado, actualización de los datos que contiene. Además se caracteriza por:

- Estructura simple.
- Flexibilidad de los sistemas relacionales, lo cual facilita las operaciones necesarias sobre estos.
- Sus sentencias permiten manejar conjuntos de registros.
- Con una sola sentencia se puede efectuar consultas complejas.
- Su código no necesita compilarse puesto que no es un lenguaje de programación.

(Cobo, Gómez, & Pérez, 2007, pp. 309-316)

1.3.7.1 MySQL

Al conjunto de programas que actúan como intermediarios entre la información almacenada y el usuario se le denomina SGBD: Sistema Gestor de Bases de Datos que permiten crear y gestionar bases de datos.

MySQL es un SGBD relacional, de código abierto que puede ser instalado en múltiples plataformas. Este gestor de acuerdo a (Mifsuf Talón, 2012, p. 282) es rápido sólido y flexible que proporciona al usuario las herramientas necesarias para:

- Definir las estructuras de los datos.
- Manipular los datos: consultar, insertar, modificar, y borrar los datos existentes.
- Mantener la integridad de la información.
- Proporcionar control de privacidad y seguridad de los datos en la BBDD, permitiendo el acceso sólo a los usuarios autorizados.

Con el propósito de acceder a la administración de MYSQL desde una interfaz web, se puede utilizar una herramienta denominada PHPMyAdmin que es una aplicación escrita en PHP. La cual permitirá crear bases de datos, tablas, borrar o modificar datos, añadir registros, hacer copias de seguridad, etc.

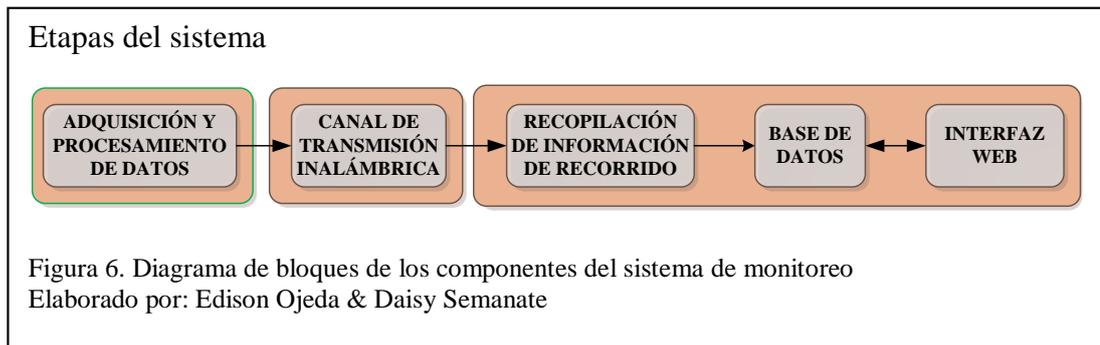
1.3.8 Servicio de alojamiento web

Una vez que se concluya la creación de la página web, el siguiente proceso será publicarlo en Internet, para lo cual se debe contratar un servicio de alojamiento web o Hosting. Un hosting web es un servicio donde un proveedor alquila un servidor conectado a Internet para alojar todo tipo de información, imágenes, o cualquier contenido para que se pueda acceder a ellos vía Internet, esto se asemeja a copiar ficheros en una carpeta local de un computador, con la diferencia de que esa carpeta se encuentra en el servidor del proveedor y su contenido es, por tanto, visible en Internet. El proveedor de hosting también puede proporcionar un interfaz web (panel de control) para manejar el servidor web (López, 2013).

2 Diseño e implementación

2.1 Introducción a la propuesta del diseño

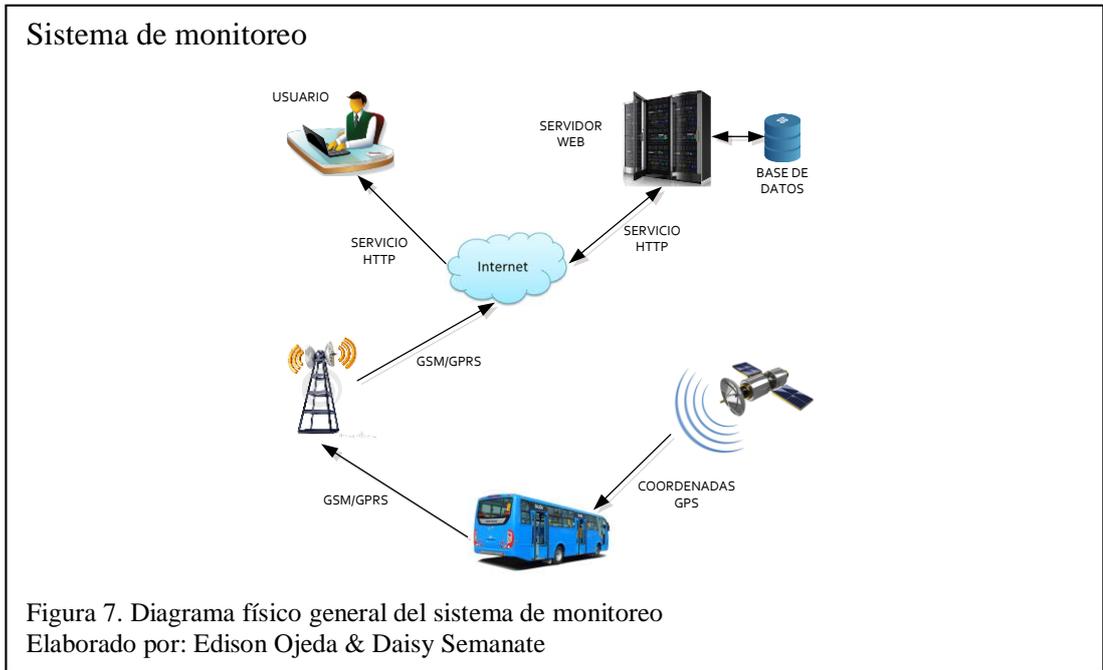
Después de realizar un estudio de los conceptos básicos, características y especificaciones de los componentes electrónicos que forman parte del proyecto, se detalla la secuencia lógica del proceso de monitoreo, donde el sistema se ha dividido en tres partes fundamentales: adquisición y procesamiento de datos, canal de transmisión inalámbrica, y recopilación de datos en una interfaz de usuario, tal como se muestra en la figura 6.



El componente de adquisición y procesamiento de datos es el encargado de recibir la información de los eventos del recorrido y procesarla, mientras tanto el canal de transmisión inalámbrica permite el acceso a la red del operador local para realizar el envío de la información previamente procesada hacia la web. Este canal de transmisión es proporcionado por la compañía de telefonía celular mediante un plan de datos contratado y es la encargada de recibir la trama de datos enviados por el sistema de monitoreo para posteriormente reenviarlos con direccionamiento IP hacia la web.

El tercer componente del sistema lo conforma la recopilación de la información del recorrido, una base de datos para almacenar los registros de los eventos, y la interfaz de usuario que permite interactuar con los datos almacenados.

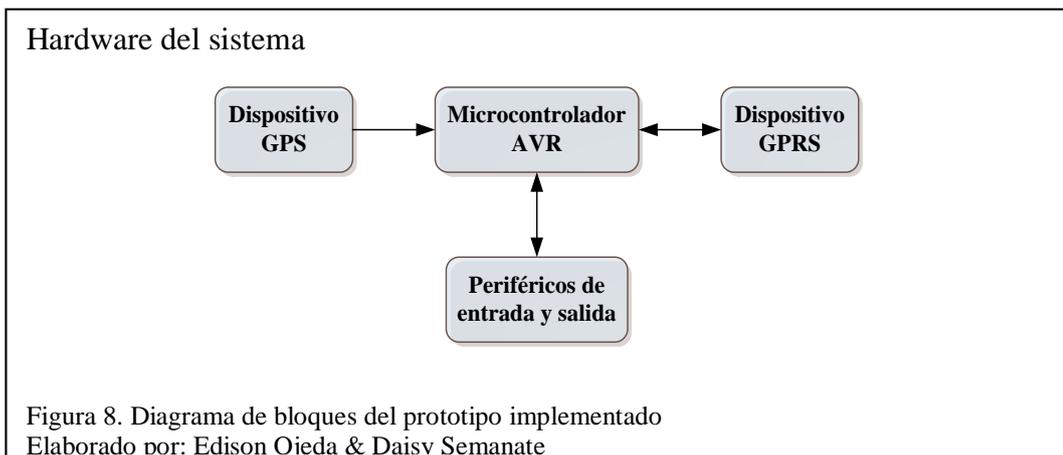
Para visualizar mejor el funcionamiento del sistema de monitoreo se utiliza el esquemático de la figura 7.



Tanto la adquisición y procesamiento de datos en su parte física y lógica, como el desarrollo de la página web con todos sus componentes de recopilación de información, codificación de funcionalidades y estilos de página se detallan en los siguientes ítems.

2.2 Modelamiento físico del sistema de geolocalización

El monitoreo de las actividades cumplidas por el autobús durante el recorrido se realiza mediante un dispositivo electrónico instalado en la unidad de transporte. El hardware consta de tres módulos independientes, pero que interactúan entre sí, tal como se muestra en la figura 8.



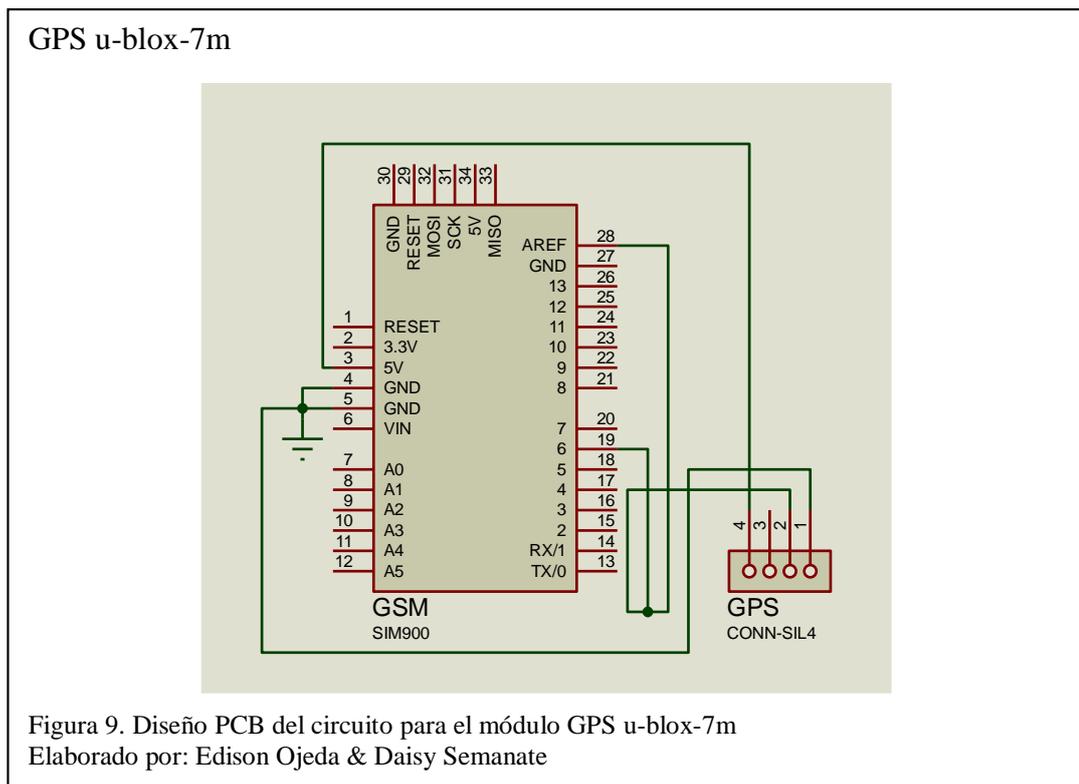
El dispositivo ensamblado en el hardware consta de tres etapas que son las siguientes:

- Dispositivo GPS con el U-blox-7M.
- Dispositivo GPRS con el Sim900 shield Geeetech compatible con arduino uno.
- Microcontrolador AVR con el Atmega164P

Se interpretan como periféricos de entrada a los sensores colocados en las puertas del autobús, los cuales indican cuando se abren o se cierran dichas puertas, mientras que como periféricos de salida están las cámaras colocadas una en la parte delantera y la otra en la parte posterior de la unidad, con la finalidad de verificar los sucesos realizados en el transcurso del recorrido.

2.2.1 Etapa dispositivo GPS

Para esta etapa el dispositivo escogido es el U-blox-7M que se muestra en el Anexo 5. Este módulo cuenta con cuatro pines: dos para polarización GND, VCC y dos para transmisión-recepción de datos con el microcontrolador. En la figura 9, se muestra el diagrama de conexiones del módulo GPS, que posteriormente se anclará a su símil GPRS.



En la realización del circuito se utilizó el simulador Proteus en su versión Isis, donde se realizaron las conexiones entre módulos y en su versión Ares para desarrollar el recorrido de las pistas del circuito impreso que se observa en el Anexo 6.

2.2.2 Etapa dispositivo GPRS

El módulo sim900 utilizado se muestra en el Anexo 7, donde está impregnado en un shield fabricado por la empresa Geeetech y es compatible con Arduino UNO. Esta placa tiene incorporado en la parte inferior un slot para la tarjeta SIM de telefonía celular y un conector externo para la antena GSM.

Para realizar el circuito impreso (PCB) del dispositivo GPRS se utilizó la base de Arduino UNO por su compatibilidad con el shield Geeetech, tal como se observa en la figura 10.

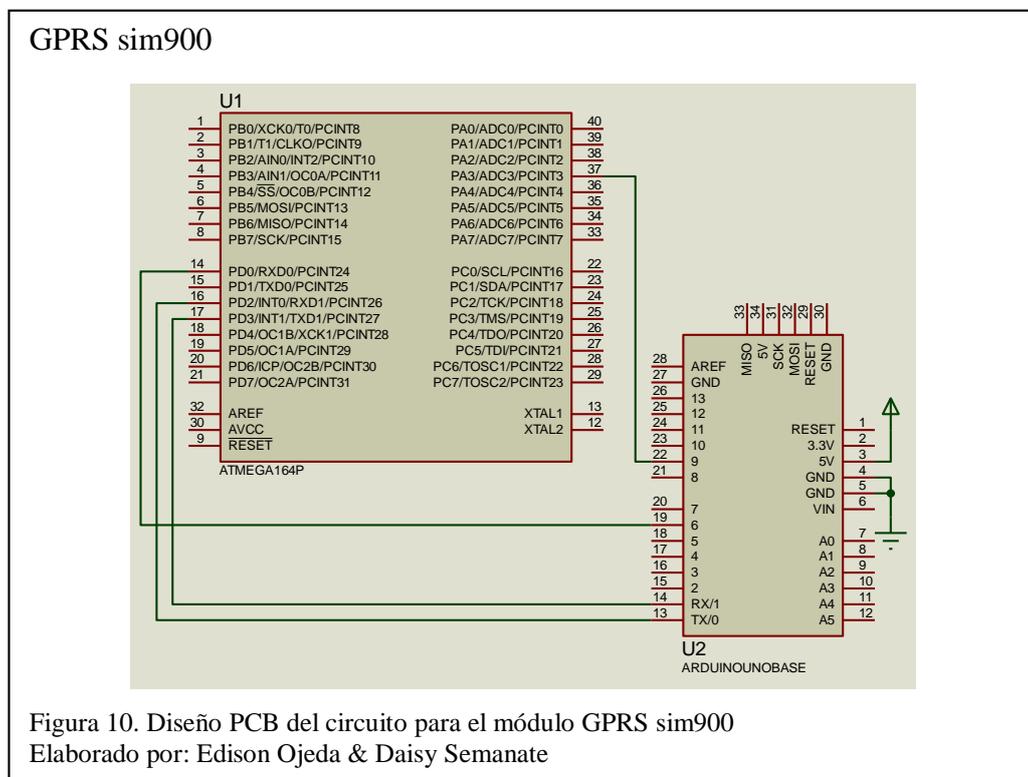


Figura 10. Diseño PCB del circuito para el módulo GPRS sim900
Elaborado por: Edison Ojeda & Daisy Semanate

Los pines 13 y 14 de la base Arduino correspondientes a los pines de transmisión Tx y recepción Rx del Sim900 son conectados con sus homólogos del puerto serial 2 del microcontrolador Atmega164P, en tanto que el pin 6 es el pin correspondiente al Tx

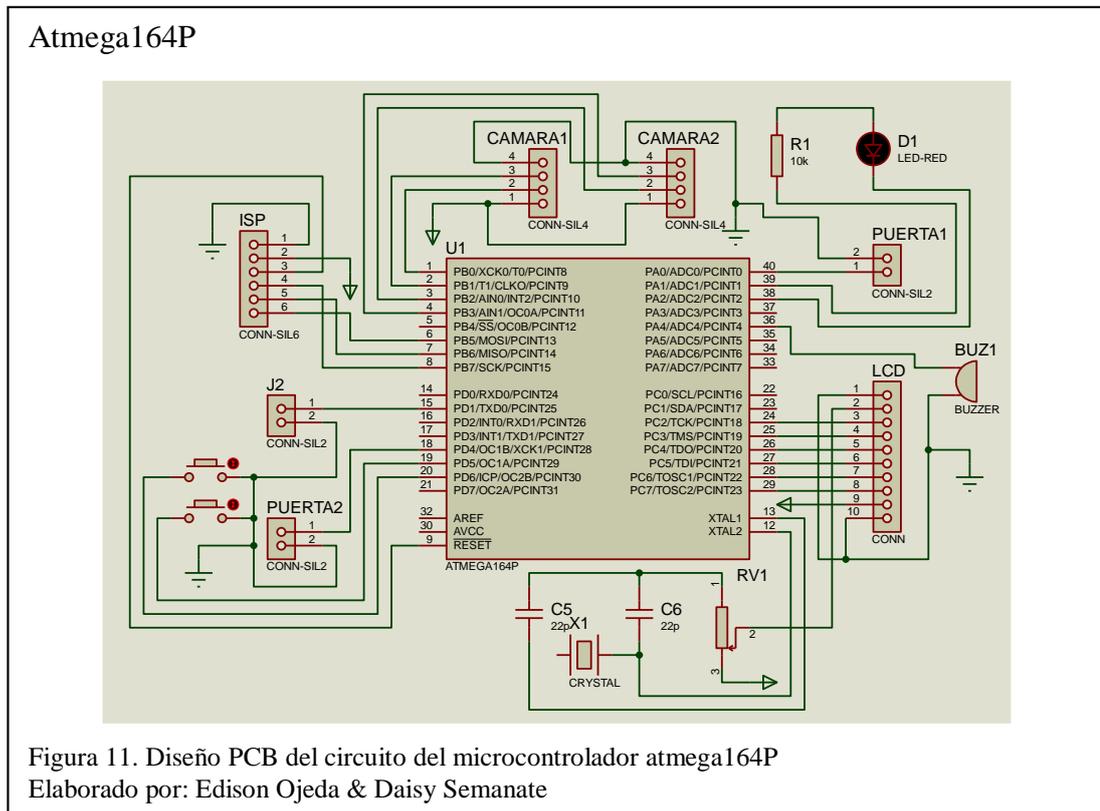
del módulo GPS montado sobre el shield Sim900, que a su vez corresponde al Rx del puerto serial 1 en el microcontrolador.

El pin 22 de la base arduino corresponde al PWRKEY del Sim900, el cual resetea el módulo y de esta manera controla el encendido y apagado del mismo. La polarización del dispositivo se realiza en GND y 5V tal como indica la figura 10.

2.2.3 Etapa microcontrolador AVR

El microcontrolador de Atmel Atmega164P, es utilizado para la adquisición y procesamiento de datos recibidos por el módulo GPS y de los periféricos de entrada y salida, para posteriormente enviar la información hacia el servidor web, utilizando tecnología inalámbrica GPRS.

En la figura 11, se indican las conexiones del microcontrolador con periféricos de entradas que son: la puerta 1, puerta 2, pulsadores de control de rutas y con los periféricos de salida que son: cámara 1, cámara 2, LCD 16x2, un speaker y los pines para la comunicación serial ISP para la conexión con el compilador Bascom-AVR.



El Atmega164P posee dos puertos seriales físicos, el primero localizado en los puertos D.0 y D.1 por donde se obtendrán los datos de ubicación enviados por el GPS, mientras que el segundo periférico serial ubicado en los puertos D.2 y D.3 que son utilizados para la comunicación bidireccional con el GPRS.

Las cámaras de control se comunicaran por el pÓrtico B, siendo B.0 y B.1 para la cámara 1, mientras B.2 y B.3 para la cámara 2. En el puerto C está conectado el LCD 16x2, donde los datos se envían del puerto C.2 al C.5, el pin RS del LCD en C.7 y el pin de Enable en el C.6.

Para tomar los datos de paradas autorizadas se utiliza un pulsador conectado en el puerto D.6, en tanto que en el puerto D.5 se conecta el pulsador encargado de seleccionar la ruta del recorrido actual. Los puertos A.0 y D.4 son los encargados de tomar datos de las acciones realizadas por las puertas 1 y 2 respectivamente.

Como dato relevante se tiene que los puertos B.5, B.6 y B.7 son utilizados para los pines MISO, MOSI Y SCK respectivamente de comunicación serial ISP con el computador, que permite programar el AVR desde el compilador Bascom-AVR.

2.2.4 Desarrollo del hardware del dispositivo electrónico de geolocalización

El ensamblaje integral de los componentes electrónicos que forman parte del dispositivo de geolocalización se realizó de acuerdo al diseño que se muestra en la figura 12.

Además de las etapas de diseño descritas anteriormente, en la figura 12, se encuentra integrado un LM2596, el cual es una fuente de alimentación DC-DC con voltajes de entrada de hasta 40V y voltajes de salida de 1.5V a 4.5V.

Esquemático del prototipo

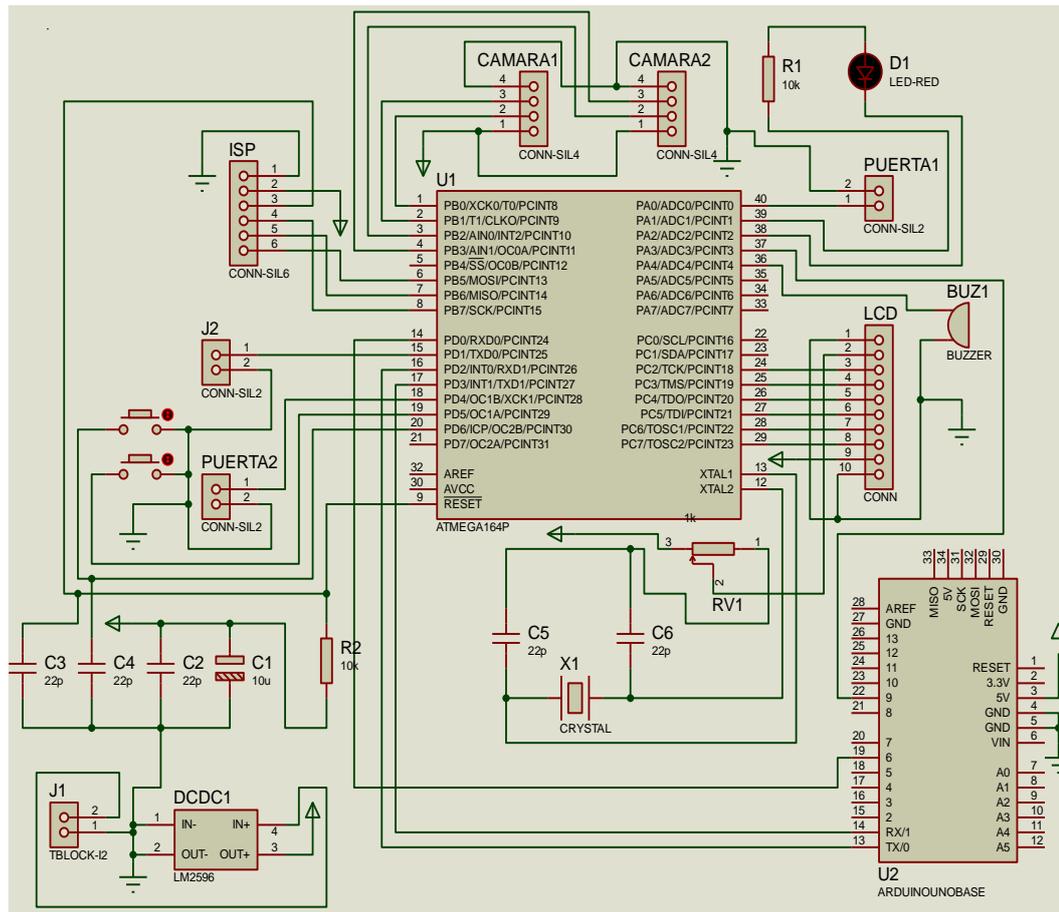


Figura 12. Diseño integral del dispositivo de Geolocalización para el reconocimiento de parada de buses

Elaborado por: Edison Ojeda & Daisy Semanate

El diseño PCB empleado para realizar el circuito impreso integral del hardware del sistema de monitoreo se observa en el Anexo 8.

El circuito y sus componentes fueron colocados en una caja que facilita su traslado y proporciona seguridad contra agentes externos o la mala manipulación del mismo, es por eso que el dispositivo electrónico está sujeto y aislado a una estructura de acrílico tal como se indica en el Anexo 9.

En el Anexo 10, se muestra la parte frontal del prototipo de monitoreo y control de parada de buses, donde se encuentra un pulsador para seleccionar la ruta del autobús y además un LCD, donde se puede observar la hora proporcionada por el dispositivo GPS y la ruta seleccionada por el conductor para el recorrido.

Las cámaras utilizadas en el sistema que se observan en el Anexo 11, son distribuidas por la tienda de elementos electrónicos ALLUC y constan de dos cables para Tx y Rx, un cable para GND y un cable para VCC de 12V. Al recibir la sentencia “grabar” por parte del microcontrolador toma una secuencia de fotos para posteriormente realizar un video de 10 segundos, que es almacenado en la memoria SD integrada en cada cámara.

2.3 Modelamiento lógico del sistema de geolocalización

El funcionamiento del sistema de geolocalización está basado en un grupo de instrucciones lógicas realizadas en el microcontrolador que controlan los diferentes periféricos y dispositivos externos, por esta razón, se analizará mediante diagramas de flujo la programación realizada en lenguaje Basic y compilado en Bascom-AVR. La lógica de programación implementada en el control de paradas de buses utiliza dos rutinas: una para inicializar los componentes del circuito electrónico y otra para realizar la adquisición y procesamiento de datos.

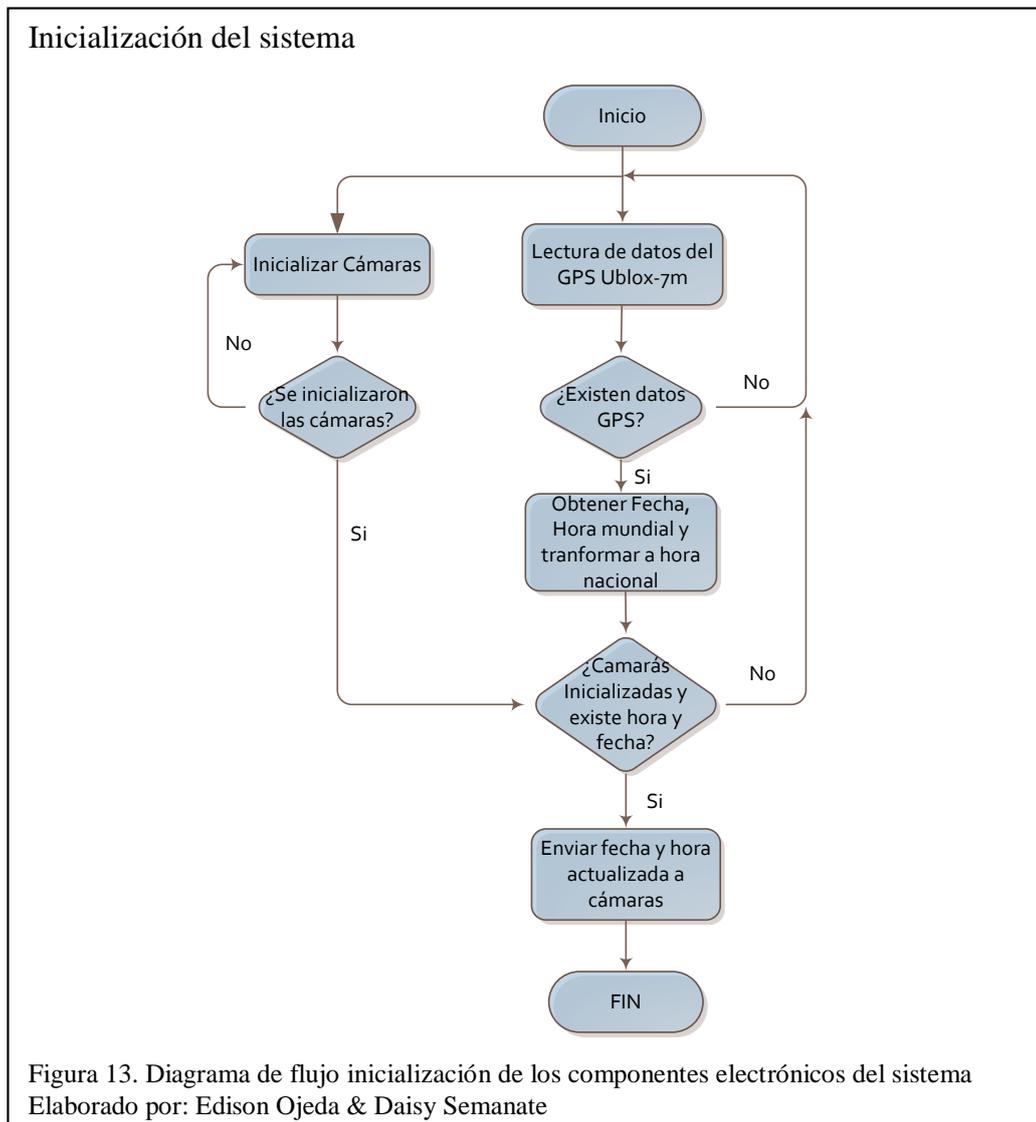
2.3.1 Rutina de inicialización

La rutina de inicialización es la encargada de mantener en espera el funcionamiento de la rutina macro del microcontrolador, hasta que el módulo GPS, el módulo GPRS y las 2 cámaras del sistema estén listas para trabajar.

Los procesos que rigen la rutina de inicialización se los describe mediante el diagrama de flujo que se muestra en la figura 13, mismo que es explicado a continuación.

1. Inicia el programa.
2. Se leen los parámetros GPS por el puerto serial 1, además se espera la inicialización de las cámaras.

3. Si la validación de los datos GPS es correcta, se obtienen los parámetros de fecha y hora mundial.
4. Comprobación de cámaras listas para trabajar y existencia de parámetros de hora y fecha, caso contrario se vuelve a leer parámetros del puerto serial 1.
5. Envío de hora y fecha hacia las cámaras.
6. Finalización del lazo finito.



2.3.2 Rutina de adquisición y procesamiento de datos

Los datos de ubicación, hora, velocidad, etc., que proporciona el módulo GPS son receptados y procesados constantemente con el lazo infinito Do-Loop, mediante una rutina cíclica que consta de dos partes:

- Subrutina adquisición y procesamiento de datos GPS.
- Subrutina de envío de datos por medio del módulo GPRS.

2.3.2.1 Subrutina adquisición y procesamiento de datos GPS.

Los parámetros proporcionados por el módulo U-Blox-7M están estructurados como una cadena de caracteres e identificados de acuerdo al protocolo NMEA, es por esta razón, que es imprescindible segmentar esta información y parametrizar estos datos de acuerdo a los requerimientos del presente proyecto.

El diagrama de flujo de figura 14 hace referencia a la subrutina encargada de tratar los datos de geolocalización, el cual se describe a continuación:

1. Inicio de la rutina.
2. Se leen los parámetros GPS.
3. Si existen valores GPS, se obtienen datos de geolocalización, caso contrario vuelve a leer los parámetros GPS.
4. La trama es recorrida y se adquieren datos de la sentencia GPRMC.
 - 4.1 Se captura la hora del módulo GPS correspondiente a la hora mundial de Greenwich y se transforma al uso horario nacional.
 - 4.2 Captura del estatus del GPS, si es dato válido "A", se guardan las coordenadas de latitud y longitud, caso contrario vuelve a revisar el estatus del GPS.
 - 4.3 Transformación de coordenadas de grados y minutos a grados decimales.
5. La trama es recorrida y se adquieren datos de la sentencia GPVTG.
 - 5.1 Se guarda el valor de la velocidad (km/h) en variable Vnueva.
 - 5.2 Buscar si en la variable Vmax existen parámetros, comparar si $V_{max} > V_{nueva}$, caso contrario $V_{max} = V_{nueva}$.
 - 5.3 Si V_{max} es mayor que V_{nueva} , V_{max} mantiene su valor caso contrario el valor V_{max} es igual al valor de V_{nueva} .

Adquisición y procesamiento de datos GPS

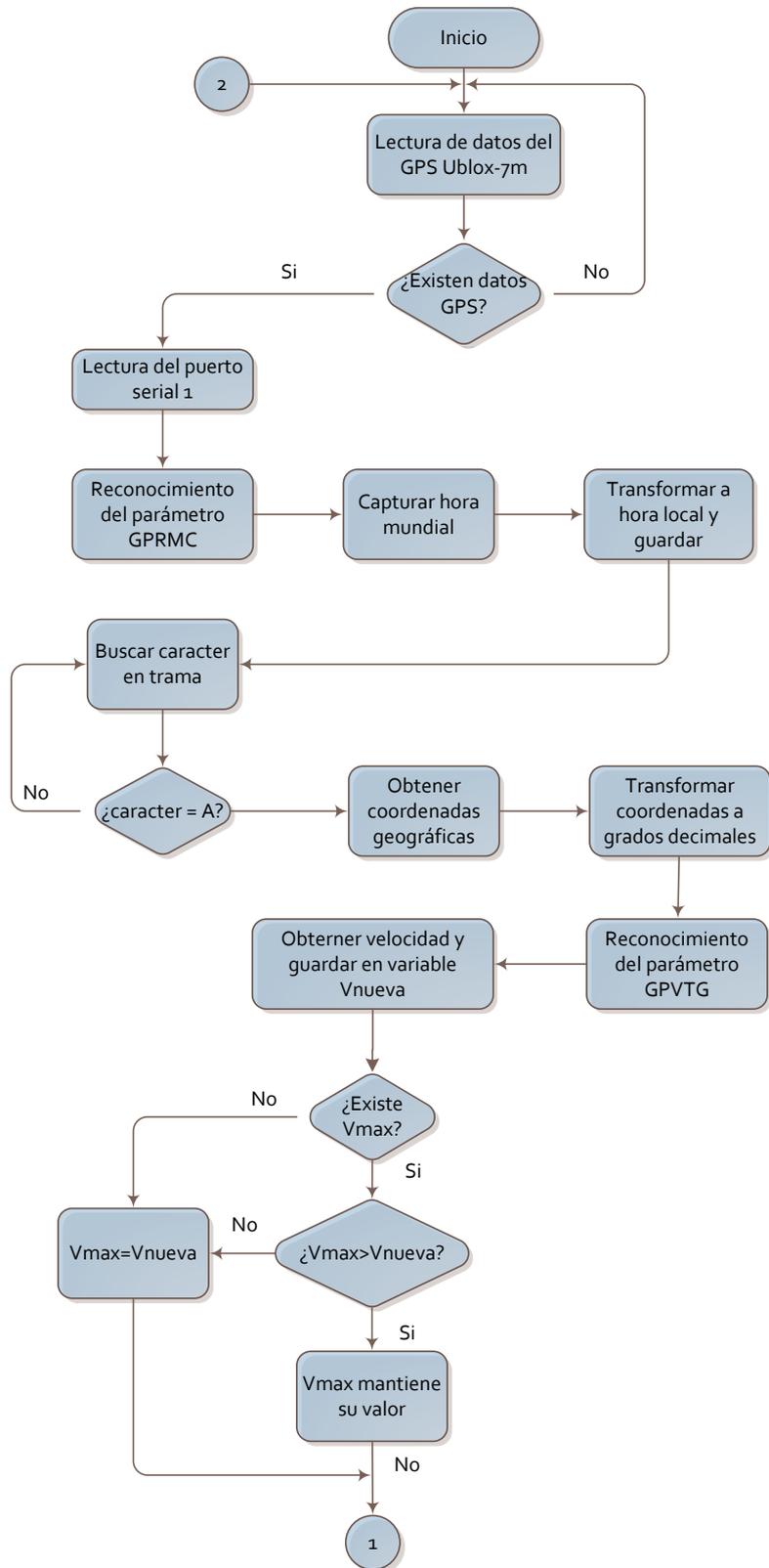


Figura 14. Diagrama de flujo de la subrutina de adquisición y procesamiento de datos GPS
Elaborado por: Edison Ojeda & Daisy Semanate

Una vez obtenidos los valores de geolocalización, se realizará el envío de estos datos hacia el sitio web tal como se describe en la subrutina del ítem siguiente.

2.3.2.2 Subrutina de envío de datos por medio del módulo GPRS.

La continuación de la rutina de adquisición y procesamiento de datos, corresponde al envío de datos a través del módulo GPRS hacia la web, tal como se muestra en la figura 15 y es descrita a continuación:

1. Se reciben los datos del GPS.
2. Comprobar si módulo Sim900 está listo para trabajar.
3. Si el Sim900 no está listo para trabajar, se inicializa el módulo.
4. Leer el puerto D.5
5. Si puerto D.5 es igual a cero se incrementa la variable ruta_e en la memoria EPROM, caso contrario si es igual a uno leer el puerto D.6.
6. Si puerto D.6 es igual a cero se envía coordenadas de parada establecida a la web por puerto serial 2, caso contrario si es igual a uno leer los puertos A.0 y D.4.
7. Si puerto A.0 o D.4 es igual a cero se comprobará la hora actual del sistema, caso contrario si ambos puertos son igual a uno el proceso regresa al inicio de la rutina para empezar a realizar un nuevo ciclo.
8. Si la hora está entre la 6 de la mañana y las 9 de la noche, se envían coordenadas, Vmax y hora del recorrido por puerto serial 2 y además se envía la orden de tomar fotos a las cámaras mediante los comB.0 y comB.2.

La totalidad del código del microcontrolador Atmega164P utilizado en este proyecto, se lo puede encontrar en los archivos adjuntos con el nombre de “gsmgprs” con la extensión “.bas”.

Envío de datos

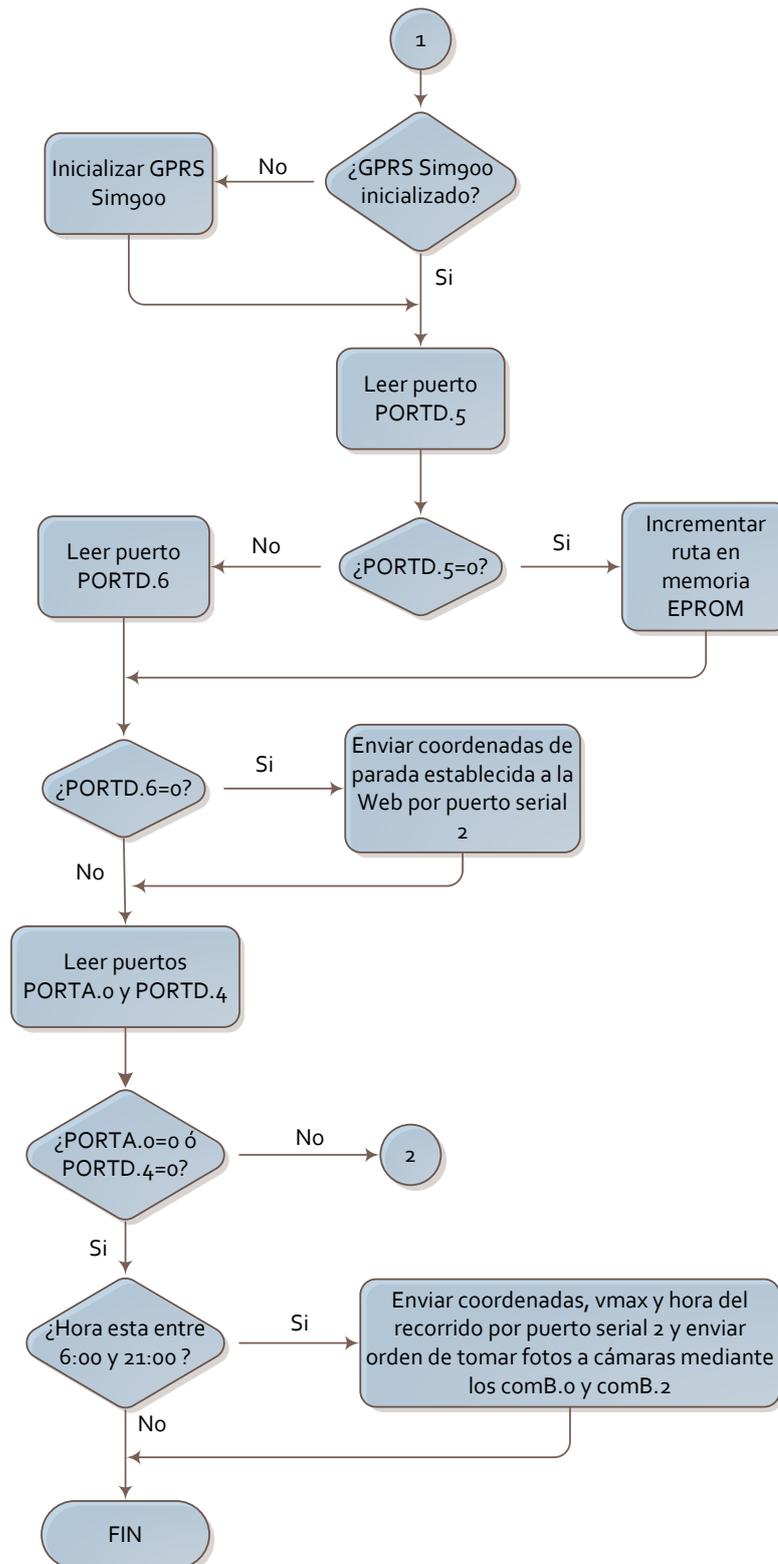


Figura 15. Diagrama de flujo de la subrutina de envío de datos por medio del módulo GPRS.
Elaborado por: Edison Ojeda & Daisy Semanate

2.4 Canal de transmisión

Este ítem detalla la forma en que los datos procesados por el microcontrolador se envían hacia la base de datos. El canal de transmisión GPRS encargado de enviar los datos es controlado y diseñado por la operadora telefónica contratada CONECEL S. A. (Claro).

El proveedor de servicios GPRS contratado cuenta con un plan que se ajusta a las necesidades del proyecto, proporcionando un paquete de datos de 250MB mensualmente que es el plan mínimo que ofrece la operadora, suficiente para la transmisión de los datos que conforman la trama GPRS, debido a que el tamaño en bytes de toda la trama es 1KB, lo que implica que en el mes se utilizaría un promedio de entre 16MB y 20MB, puesto que en una semana se tiene alrededor de 3772 datos receptados en la base de datos.

Para transmitir los datos generados en el hardware del sistema mediante el canal GPRS, se configuró los parámetros de aplicaciones IP tales como: información de Nombre del Punto de Acceso (APN), usuario y contraseña de la radio base móvil de la operadora Claro en el módulo Sim900, que permiten direccionar la información hacia el dominio sistemadebuses.com correspondiente a la dirección IP 200.58.111.99, asignada por el servicio de alojamiento web contratado. En el Anexo 12, se observa la Sim-Card instalada en el módulo Sim900.

En el siguiente ítem se describirá el desarrollo y funcionamiento del tercer componente del sistema de monitoreo, desde el diseño, maquetación y programación de la página web, así como la creación de tablas y almacenamiento de la información en la base de datos MySQL.

2.5 Etapa de diseño de la interfaz web

Para posibilitar a los usuarios la consulta y visualización de los reportes del recorrido del autobús, además de que puedan interactuar con la información guardada en la base de datos, se realizó la construcción de un sitio web, el mismo que consta de una página de inicio y varias páginas secundarias.

El aplicativo web de este sistema de monitoreo cumplirá las siguientes funciones:

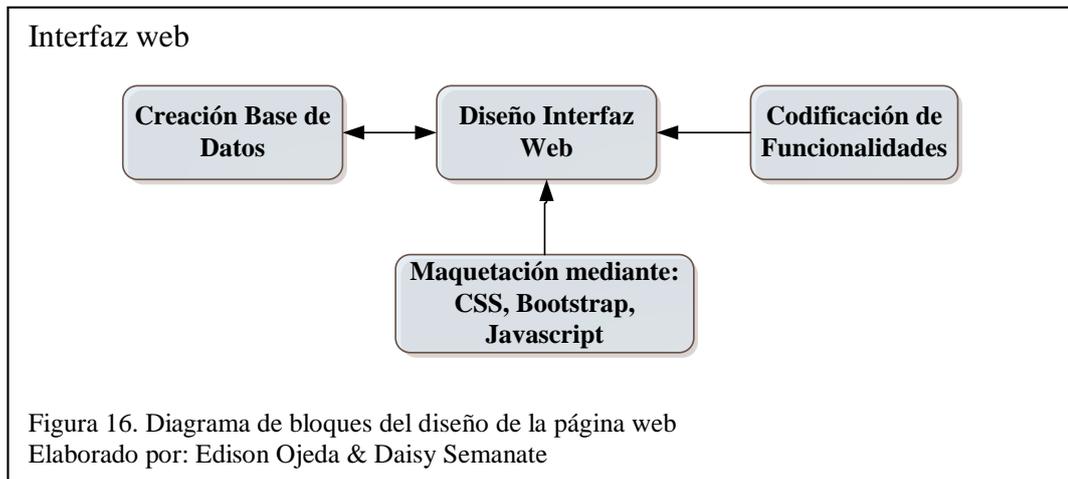
- Autenticación de usuarios y manejo de sesiones.
- Creación, modificación o eliminación de usuarios.
- Consulta de recorrido realizado.
- Generación de reportes con información sobre los recorridos efectuados por la unidad de transporte, así como los horarios en los que realizó las paradas y contravenciones tales como excesos de velocidad entre paradas.
- Visualización del recorrido realizado en un mapa con marcadores de diferentes colores.

Es así, que en el diseño y programación de las páginas web se utilizarán archivos tales como: archivo HTML, hoja de estilos CSS, aplicación de JavaScript para la programación del mapa, archivo PHP que realizan la consulta MySQL.

Algunas herramientas indispensables para la creación de sitios web como lo es PHP y MySQL solo funcionan en servidores web. Por lo tanto para desarrollar la aplicación web de este proyecto, y probarla de manera local antes de subirla al servidor del internet o hosting, se instaló Wamp Server, esta herramienta crea un servidor local en el computador llamado localhost. (Incluye: APACHE, PHP, MySQL, PHPMyAdmin).

Al instalar el entorno de trabajo se creó un directorio “www” que es la raíz en donde se debe guardar los archivos con extensión .php de la aplicación del proyecto, por defecto viene un archivo index.php en este directorio, el cual sirve para verificar el correcto funcionamiento del Wamp. Se puede acceder a dicha página digitando en el navegador web la dirección que corresponde al localhost: <http://localhost/>, en donde se podrá visualizar una página de información tal como se muestra en el Anexo 13.

La elaboración de las interfaces web se las realizó de tal manera que al usuario le resulte amigable y fácil de utilizar. Para realizar el diseño de estas páginas se siguió el proceso que se muestra en el diagrama de bloques de la figura 16.



A continuación se describe cada uno de los bloques que conforman el diseño de la interfaz web.

2.5.1 Bloque creación de la base de datos

Una vez que se ha instalado el Wamp Server, se creó la base de datos utilizando el gestor de base de datos PHPMYAdmin. Esta base de datos la conforma un conjunto de tablas con sus respectivos campos (columnas) y registros (filas), donde se almacenarán los datos para manipularlos. En este caso la base se llama “proyecto” y consta de tres tablas:

Tabla usuarios: Guarda los datos relacionados con los usuarios y el nivel de privilegios para la administración de los datos. Existirán dos tipos de usuario: invitado y administrador.

Tabla rutas: La compañía de transporte puede tener varias rutas y en su defecto diferentes paradas, por lo que en esta tabla se guardará la ruta y las coordenadas de las paradas correspondientes a dicha ruta.

Tabla informes: Almacena los datos del recorrido realizado, parámetros como latitud, longitud, velocidades, número de bus, y de ruta.

Luego de haber creado las tablas, se crearon los campos de cada tabla. No todos los campos son iguales, existe varios valores los más utilizados en las tablas fueron los siguientes: INT (almacena solo números enteros), VARCHAR (almacena cadenas de texto), DOUBLE (almacena número en coma flotante de tamaño normal), DATE

(almacena fechas en formato AAAA-MM-DD), TIME (almacena hora en formato hh:mm:ss). En el Anexo 14, se muestra la creación de la base de datos con sus respectivas tablas.

2.5.2 Bloque maquetación interfaz web

El entorno gráfico, visual, de comunicación y de opciones de un aplicativo web es muy importante, puesto que es de lo que dispondrá el usuario para interactuar con el sistema, básicamente constituye la presentación y navegación en el sitio web. En el diseño gráfico de la interfaz se utilizó los elementos que se detallan en los siguientes ítems.

2.5.2.1 Elementos de HTML, CSS, bootstrap

La aplicación web del sistema consta de una página de inicio o index y 4 páginas secundarias (administrar, mensaje, invitado, visualizar), el proceso de elaboración y diseño de las mismas se describe a continuación.

Para la página de inicio se utilizó el modelo de una plantilla web gratuita a la cual se le realizó varias modificaciones acorde a las necesidades del proyecto. Esta plantilla a más del index incluye una carpeta donde se albergan las imágenes que se utilizan en el diseño, así como también una carpeta donde se encuentra el archivo de hoja de estilos o CSS. Con el objetivo de definir las propiedades de la página se modificó el archivo CSS para determinar parámetros básicos como el tamaño de la fuente, el tipo de fuente, el color, imagen, márgenes y otras generalidades.

En base a la plantilla de la página de inicio se realizó el diseño de las plantillas para las páginas secundarias, se hizo algunas modificaciones de tal manera que solo quedó el color de fondo y los márgenes. En estas páginas también se utilizó Bootstrap con el fin de maquetar el sitio web con mayor facilidad, herramienta que resultó ser muy práctica a la hora del formateo de las páginas. Las etiquetas HTML más utilizadas en el diseño tanto de la página de inicio como en las páginas secundarias se muestran en el Anexo 15.

2.5.2.2 Elementos de Javascript y API de Google Maps

Para dar interactividad y dinamismo a la aplicación web, también se utilizó el lenguaje de programación de JavaScript, así como también algunas de sus librerías (jQuery), para lo cual se utilizó la etiqueta script, que permite escribir código de JavaScript dentro de HTML, o hacer referencia a un script ejecutable externo a través del atributo src. Esta etiqueta debe estar dentro de la cabecera del archivo HTML para que el código JavaScript pueda ejecutarse. Es así que JavaScript permitió la manipulación de imágenes, validación de formularios, y los cambios dinámicos de contenido.

Para insertar un mapa de Google en una de las páginas del aplicativo web, se utilizó el API de Google Maps JavaScript, para lo cual fue necesario solicitar una clave o API Key de uso en el sitio de Google Maps, y especificar en qué URL se la iba a utilizar, es decir la raíz del dominio web, para que sea útil y funcional para los subdominios y directorios del sitio. Una vez que se obtuvo la clave se la incluyó dentro del encabezado del archivo HTML de la página. Adicionalmente se creó una función en JavaScript, la cual inicializará el mapa y lo desplegará, una vez que se haga el llamado.

La función contiene ciertos parámetros y características que definen la visualización del mapa tales como la compatibilidad del navegador con la API, la ubicación del mapa, nivel de zoom, tipo de mapa, entre otras. La llamada a la función se lo hizo desde el cuerpo del documento HTML a través del evento onload del <body> y utilizando un <div> con una id de map se definió el ancho y alto del mapa de modo que se visualice de manera óptima.

Finalmente para insertar marcas en el mapa se utilizó 3 métodos:

- GLatLng que establece la ubicación de la marca.
- Gmarker contiene la marca que se requiere insertar.
- addOverlay() que permite añadirla sobre el mapa.

El comportamiento del mapa lo definieron las clases, métodos y propiedades contenidos en los archivos de JavaScript.

2.5.3 Bloque codificación de funcionalidades PHP

En este apartado se describirá las funciones PHP más utilizadas en la programación de las páginas web y las consultas MySQL que se utilizaron para la creación de los comportamientos dinámicos del sitio web.

2.5.3.1 Conexión a la base de datos MySQL con PHP

Para tener acceso y manipulación de los datos almacenados en la base, es necesario establecer las sentencias de conexión y consulta o modificación con la BBDD, conforme se vaya desarrollando y programando los scripts de PHP de manera que cuando se ejecute el script la conexión a la base sea satisfactoria.

Para establecer la conexión con una base de datos MySQL desde PHP, se utilizó la función con estilo orientado a objetos que tiene la siguiente estructura:

```
$mysqli= new mysqli (“NombreHost”, “Usuario”, “Contraseña”, “Nombrebd”);
```

NombreHost: Indica a donde se conecta con la base de datos. Indica el nombre del servidor (o IP del mismo).

Usuario: Es el usuario de acceso a la base de datos.

Contraseña: Es la clave que se corresponde con ese usuario.

Nombrebd: Se corresponde con el nombre de la base de datos.

Y para realizar una consulta a la base de datos se añade las sentencias:

```
$consulta=“select * from NombreTablaBd”;
```

```
$petición=$mysqli->query ($consulta);
```

2.5.3.2 Programación página inicio

La página de inicio conforma el sistema de autenticación de usuarios que permite el ingreso al sitio web solo a usuarios autorizados, por lo que en el archivo de HTML se agregó un formulario que una vez que se llene los campos de usuario, contraseña y se escoja el tipo de usuario dicho formulario sea enviado al servidor y procesado mediante consultas MySQL, con el fin de constatar que los datos ingresados corresponden a un usuario registrado en la tabla usuarios. Se realizó un script PHP que cuenta con el método HTTP POST que permite acceder a los datos del formulario

HTML, en la figura 17, se muestra cómo funciona dicho método. Se crea un formulario indicando el método a utilizar, y se crea un script de PHP para recibir los datos que introduce el usuario.

HTTP Post.

Formulario HTML

```
<form action="pagina.php" method="POST" >
  Usuario: <input type="text" name="usuario" />
  Clave: <input type="password" name="clave" />
  <input type="submit" value="Enviar" />

</form >
```

Script PHP

```
<?php
$usuario = $_POST['usuario'];
$clave = $_POST['clave'];
echo "Usuario " . $usuario . " y clave ". $clave;
?>
```

Figura 17. Ejemplo del envío de datos mediante el método HTTP Post
Fuente: (Velasquez, 2013)

Además para impedir que los usuarios no ingresen a otros directorios que no corresponden al nivel de privilegios se utilizó las variables de sesión de PHP `session_start()` y `session_id()`, para registrar cuando un usuario ha ingresado a la página, siendo esta información distinta para cada usuario, y se guarda en el sistema de forma temporal. Si los datos corresponden al usuario registrado podrá acceder a la siguiente página. Cada interfaz será distinta dependiendo del usuario que esté ingresando a la página.

En la figura 18, se muestra el diagrama de flujo diseñado para el procedimiento de ingreso a todas las interfaces del sitio web.

Ingreso a interfaces web

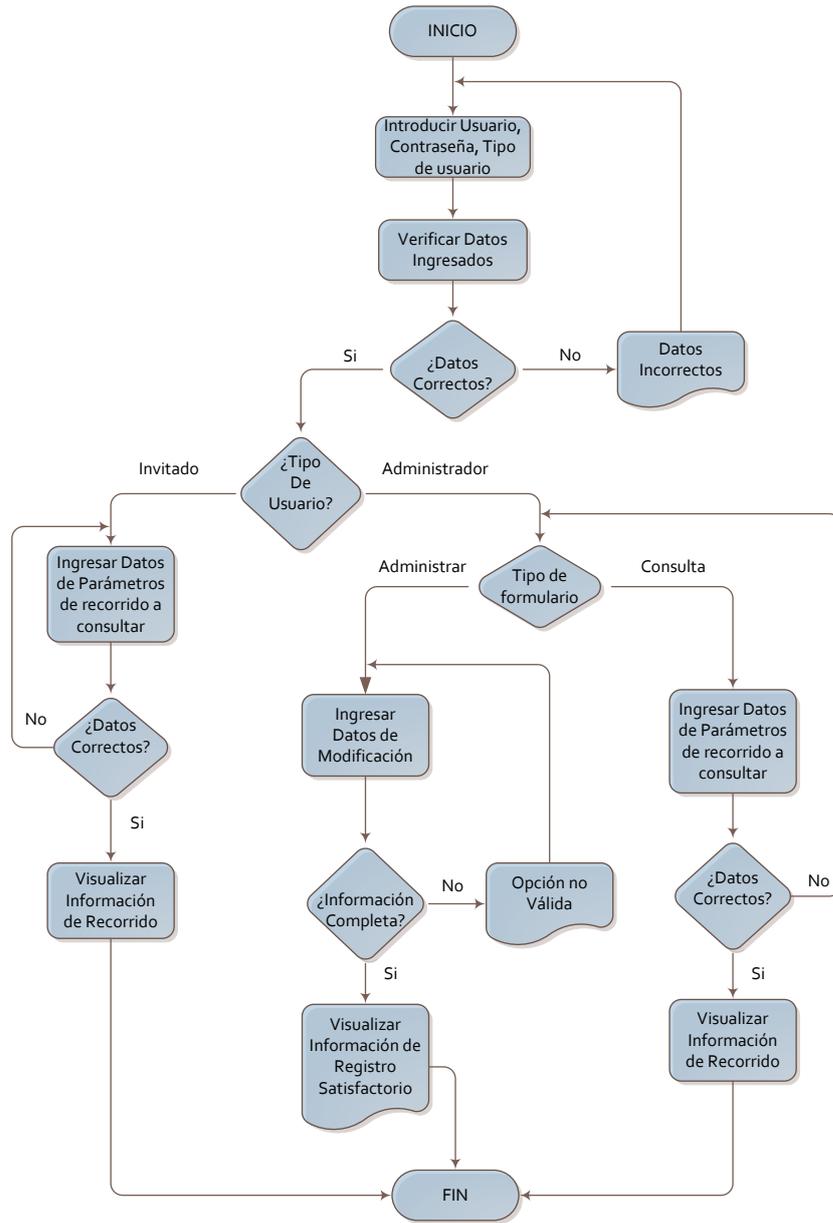


Figura 18. Diagrama de flujo de la página index para navegar entre las diferentes interfaces
Elaborado por: Edison Ojeda & Daisy Semanate

A continuación se describe el diagrama de flujo de la figura 18:

1. Inicio del programa.
2. Inserción de campos de datos de usuario, contraseña y tipo; y verificación de datos.
3. Validación de que los datos sean correctos, si lo son se verifica qué tipo de usuario es, en caso de no ser correctos se emitirá un mensaje: “Usuario no

encontrado, ingrese todos los datos correctos” y regresa a solicitar nuevo ingreso de datos.

4. Si el usuario es invitado ingresa a una nueva interfaz que consta de un formulario que solicita parámetros del recorrido a consultar, si los datos ingresados son correctos se podrá visualizar la interfaz donde se muestra la información del recorrido, caso contrario regresa a solicitar nuevos parámetros de recorrido.
5. Si el usuario es administrador ingresa a una nueva interfaz que presenta dos formularios: administrar y consulta.
 - 5.1 En el formulario administrar, se debe ingresar los datos que se desea modificar, si todos los campos de la información a completar están correctamente llenos se podrá visualizar la interfaz que muestra que la información a modificar ha tenido un resultado satisfactorio, caso contrario se emitirá un mensaje: “Opción no válida” y regresa a solicitar nuevo ingreso de datos.
 - 5.2 En el formulario consulta, se debe llenar los campos que solicitan parámetros del recorrido a consultar, si los datos ingresados son correctos se podrá visualizar la interfaz donde se muestra la información del recorrido, caso contrario regresa a solicitar nuevos parámetros de recorrido.
6. Fin del programa.

En base a esta lógica de programación se explicará cómo se realizó la programación en las otras interfaces.

2.5.3.3 Programación página invitado

Esta página web está conformada por un formulario que le permite al usuario ingresar cierta información para consultar el reporte del recorrido, además tiene un botón “INGRESAR” con el cual una vez que se haya llenado el formulario se envía los datos al archivo visualizar que se encuentra en el servidor para procesarlos utilizando el método GET, el mismo que permite concatenar los datos ingresados en el formulario a la URL, la cual se utilizará en la función JavaScript para graficar los puntos en el mapa.

Para realizar una consulta de recorrido, el usuario debe especificar datos como: el número de unidad de transporte, el número de ruta y la fecha del recorrido.

2.5.3.4 Programación página administrador y mensaje

La interfaz de administrador es similar a la de invitado, con la diferencia que esta página posee dos formularios: el formulario administración de usuarios y el formulario consulta de recorrido, a continuación se describe las funcionalidades de cada uno:

- El formulario de administración de usuarios permite al usuario administrador realizar acciones como agregar, actualizar datos, y eliminar usuarios. Este formulario se distingue por utilizar aparte de cajas de texto, dos menús desplegables con opciones dentro del mismo, un menú con opciones de tipo de operación y otro con opciones de tipo de usuario.

Para realizar alguna modificación el usuario administrador debe elegir el tipo de operación que desea ejecutar entre 3 opciones: Nuevo Ingreso, Actualización de Datos y Eliminar Usuario. Luego debe seleccionar el tipo de usuario aquí se tiene dos opciones: Invitado y Administrador; después deberá llenar los campos Usuario y Password con los datos del nuevo usuario, finalmente sólo en caso de que la acción primera haya sido eliminar o en su defecto actualizar se deberá llenar el campo Usuario Antiguo.

Una vez que se haya completado la información del formulario haciendo uso del botón “PROCESAR” se envía los datos mediante el método POST hacia un archivo para que procese la información ingresada. Este archivo es una nueva interfaz llamada mensaje, en este script PHP se realizó la programación con las siguientes especificaciones.

1. Este script recibe los datos mediante el método POST.
2. Establece la conexión con la base de datos y realiza la consulta como se expuso con anterioridad en el ítem 3.4.3.1., y a su vez utiliza nuevas instrucciones que permiten la inserción, modificación y eliminación de usuarios.

- Inserción de nuevos registros a una tabla de la base: Insert into, Values
 - Actualización de registros existentes en una tabla de la base de datos: Update, Set, Where
 - Eliminar un registro de cualquier tabla de la base: Delete, From, Where
3. Cada una de las instrucciones anteriormente mencionadas se las colocará dentro de una sentencia condicional para que el código se ejecute sólo si la condición especificada es verdadera.
 4. Esta página una vez que recibe los datos y los procesa también se encargará de mostrar al usuario administrador un mensaje informando si el proceso que solicitó se realizó de forma satisfactoria.
- La estructura y funcionalidad del formulario consulta de recorrido es idéntico al descrito en la página invitado.

2.5.3.5 Programación página visualización de reportes del recorrido

Esta interfaz web consta de un mapa, un cuadro de registro de los horarios cumplidos por el recorrido y un cuadro de reporte de contravenciones. Tanto el usuario invitado como administrador una vez que se hayan autenticado y mediante el formulario de consulta de recorrido podrán acceder a esta interfaz.

En el mapa se visualizarán dos tipos de marcas (bandera e icono) que representarán las paradas realizadas por el autobús. Las banderas rojas simbolizan las paradas correctas, mientras que los iconos representan las paradas incorrectas, diferenciados por colores de acuerdo a la cantidad de recorridos realizados. El script puntosmapa.php permite visualizar el recorrido tal como se indica a continuación:

1. Captura los datos enviados a través del método GET, al llenar los formularios de consulta de recorrido tanto del usuario administrador como del usuario invitado.
2. Realiza la conexión a la base de datos y la consulta a las tablas: informes y rutas.
3. Mediante el uso de la función PHP foreach() se pudo recorrer los vectores y objetos de una manera controlada. Esta función permite recorrer cada elemento del vector y procesar sus valores.

4. Los valores de latitud y longitud obtenidos del bucle foreach se almacenarán en otro vector, el cual será utilizado para graficar las coordenadas en el mapa. Para lo cual se utilizó la función `json_encode()`, que permite pasar arrays PHP a cadenas JSON y matrices de JavaScript. Esto es necesario hacerlo debido a que en la interfaz de visualización se está utilizando código de JavaScript para insertar el mapa.

En la página visualización de recorrido en la cabecera del documento que es donde estará incrustado el código de JavaScript, se debe utilizar la función `getJSON(url, function [data])`, la cual cargará los datos de la función `json_encode()` desde el servidor utilizando una petición GET. Los datos que se envían al servidor se adjuntan a la dirección URL como una cadena de consulta. Si la solicitud tiene éxito la función de devolución de llamada se ejecutará satisfactoriamente.

Adicionalmente se utilizará la función `each()` la cual permitirá iterar sobre la matriz definida por la estructura `getJSON`. La devolución de llamada se pasa un índice de matriz y un valor de matriz correspondiente cada vez. Con la utilización de estas funciones y con el código para graficar marcas, expuesto con anterioridad en el ítem 3.4.2.2 se graficó las coordenadas de las paradas que realizó el automotor.

En el cuadro registro de los horarios cumplidos por el recorrido el usuario podrá visualizar de forma detallada la hora en la que el autobús realizó las paradas correctas, mientras que en el cuadro reporte de contravenciones observará las coordenadas y hora en la que el bus abrió las puertas indebidamente además del exceso de velocidad durante el recorrido.

En la programación de este apartado se realizó lo siguiente:

1. Captura de datos enviados a través del método GET, al llenar los formularios de consulta de recorrido tanto del usuario administrador como del usuario invitado.
2. Conexión a la base de datos y la consulta a las tablas: informes y rutas.
3. Combinación de la función PHP `foreach()` con la sentencia condicional `if` para realizar los algoritmos que permitan definir los datos que deben visualizarse en los cuadros expuestos anteriormente.

2.6 Etapa de recopilación de información del recorrido y base de datos

Una vez que la información del recorrido es transmitida por el canal GPRS hacia el servidor web, es necesario almacenarla en una base de datos para que posteriormente se pueda tener acceso a esta y se la pueda manipular y procesar de acuerdo a las necesidades del usuario.

En el diagrama de flujo de la figura 19 se muestra el proceso de captura de datos transmitidos a través del canal GPRS. Una vez que se tiene la URL es necesario extraer la información de la trama GPRS, para lo cual se realiza la programación de un script en PHP que permita mediante el método GET seleccionar los parámetros de la cadena de texto y guardarlos en la base de datos MySQL.

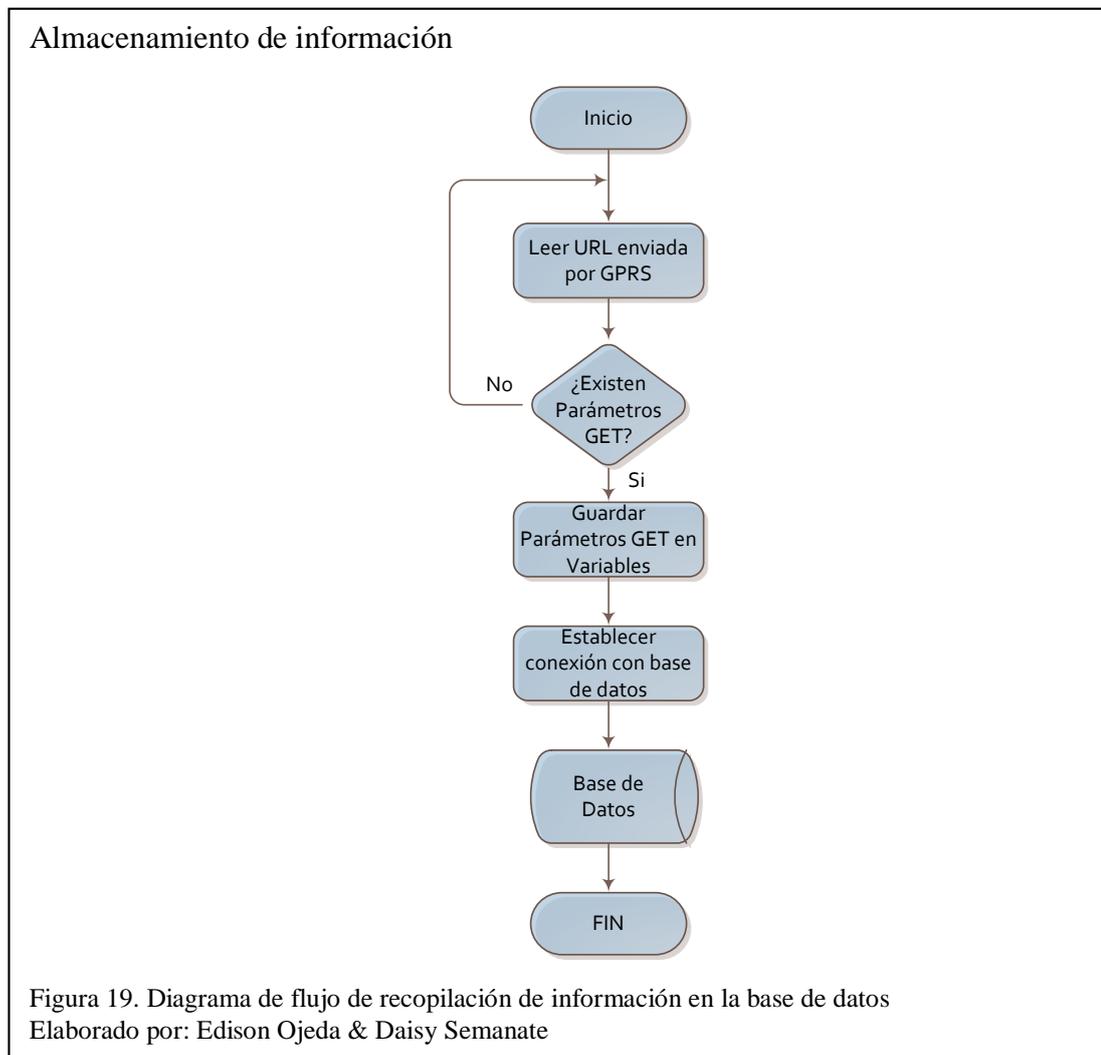


Figura 19. Diagrama de flujo de recopilación de información en la base de datos
Elaborado por: Edison Ojeda & Daisy Semanate

A continuación se describe la lógica de programación implementada en esta etapa.

1. Inicio del programa.
2. Lectura de URL enviada por módulo GPRS.
3. Verificación de existencia de datos o parámetros.
4. Si existen datos se los guarda en variables, caso contrario retorna al inicio.
5. Se establece la conexión al servidor de la base de datos (MySQL) mediante PHP.
6. Se almacena los datos mediante la instrucción de inserción de registros a una tabla de la base de datos.
7. Fin del Programa.

La programación realizada en el script PHP se detalla a continuación:

1. Mediante el método GET se realiza la extracción y captura de los datos que son enviados en la trama del dispositivo GPRS.
2. Establecimiento de conexión a la base de datos y consulta de registros de la tabla rutas. Se guarda en un vector los datos de la tabla rutas.
3. Se utilizó la función PHP `fetch_array` dentro de un bucle `while`, para acceder a los datos almacenados y recuperar una fila de resultados como un vector asociativo. Los elementos del vector serán utilizados para comparaciones y operaciones con los datos obtenidos por el método GET.
4. Luego los resultados se los guarda en la base de datos en la tabla informes a través de la instrucción `INSERT`.

2.7 Implementación del sistema

En base a los diseños expuestos anteriormente, se presenta el proyecto implementado, el cual consta de dos partes: implementación de la página web en el hosting, e implementación del circuito en el autobús.

2.7.1 Implementación página web en el hosting

Una vez que se terminó con el diseño y programación de las interfaces del sitio web, es necesario almacenar dichos archivos en un servidor de alojamiento web, para lo cual se contrató el hosting DonWeb (Ver Anexo 16) porque a pesar de que las

características de capacidad de almacenamiento y prestación de servicios de los diferentes hosting en el mercado son similares, este a diferencia de otros proporciona constante soporte técnico en idioma español ideal para solucionar los problemas suscitados y que además posee planes iniciales y corporativos con costos accesibles de acuerdo a las necesidades del usuario.

Para subir los archivos del proyecto al servidor web del Internet se utilizó el protocolo de transferencia de datos FTP mediante la herramienta Filezilla, tal como se muestra en el Anexo 17.

En la figura 20, se muestra la interfaz de la página inicio o index.

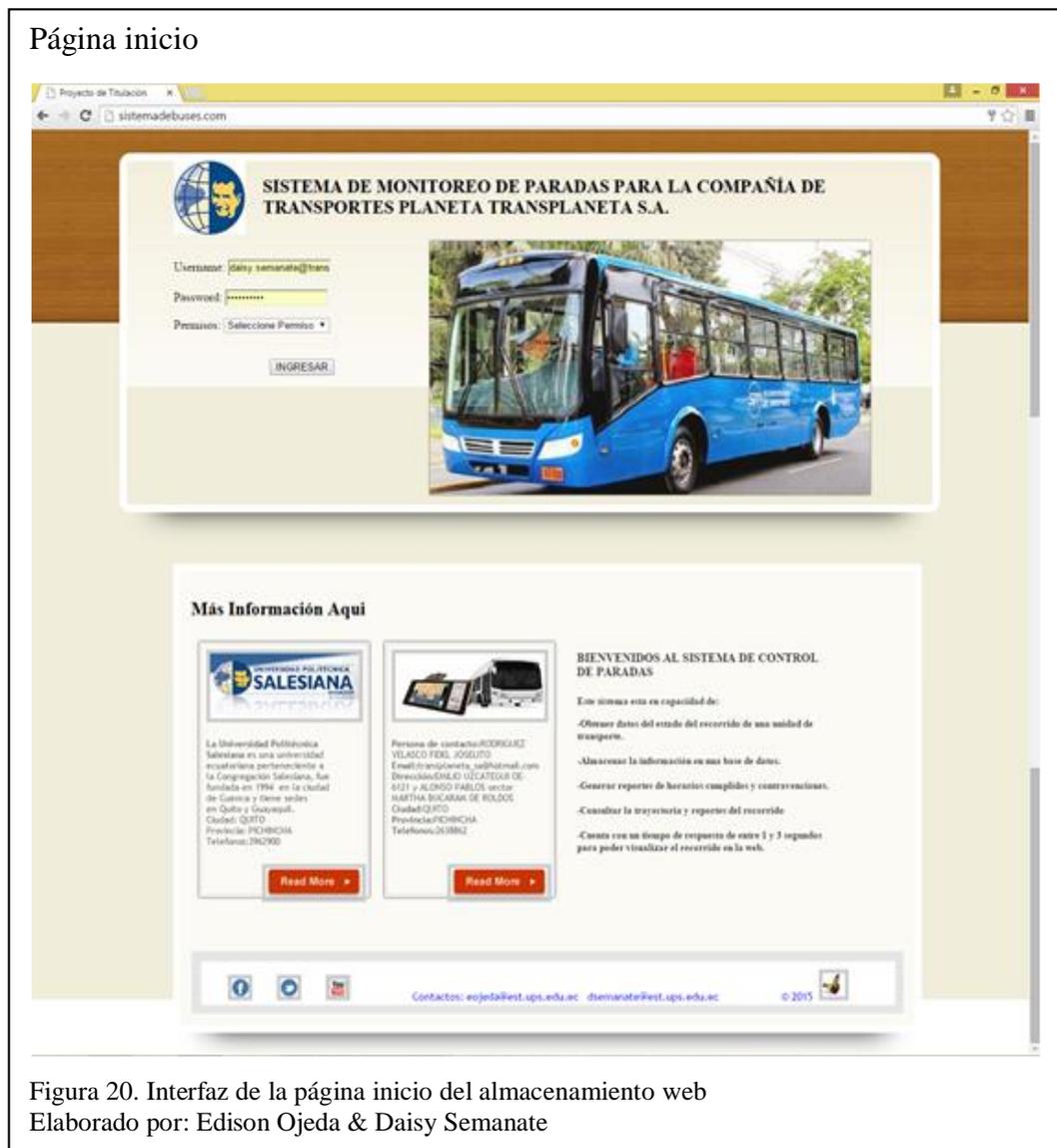
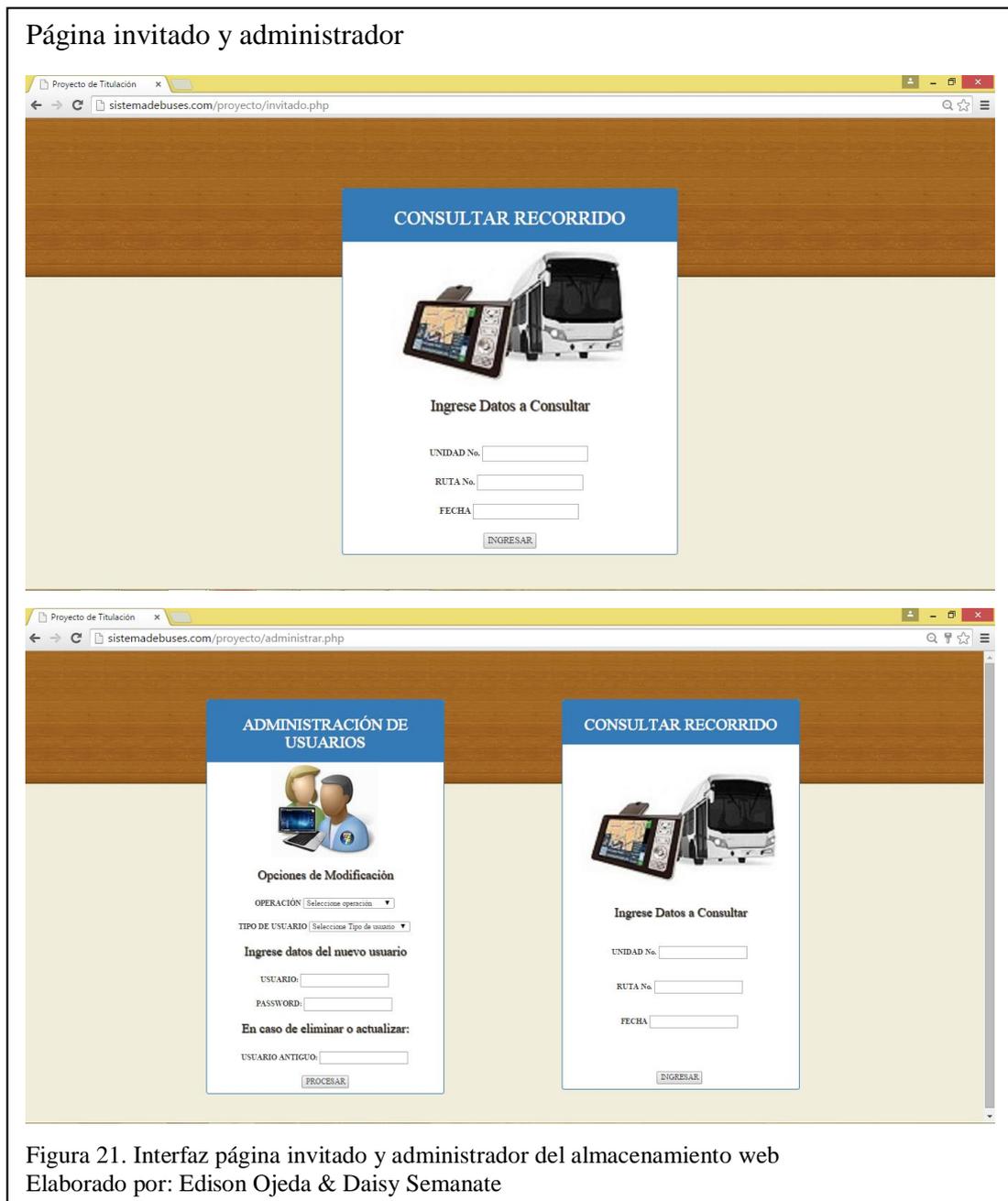


Figura 20. Interfaz de la página inicio del almacenamiento web
Elaborado por: Edison Ojeda & Daisy Semanate

Los diseños de la interfaz de invitado y la de administrador que permiten acceder a consultas de recorrido se muestran en la figura 21.



El informe del recorrido se podrán observar con detalle en la interfaz visualización de reportes, como se muestra en la figura 22.

Página visualización reportes

Proyecto de Tlalcala

sistemadebuses.com/proyecto/visualizar.php?bus=151&ruta=1&fecha=2015-08-27

INICIO ADMINISTRAR UPS

INFORME GENERAL DEL RECORRIDO REALIZADO

En el mapa se puede visualizar los siguientes indicadores:
La bandera roja simboliza las paradas realizadas correctamente.
Los iconos que se presentan a continuación representan las paradas incorrectas de los recorridos.

El icono de color azul	— Rec1.	El icono de color verde	— Rec2.
El icono de color amarillo	— Rec3.	El icono de color rojo	— Rec4.
El icono de color naranja	— Rec5.	El icono de color violeta	— Rec6.

RESULTADOS DEL RECORRIDO

A continuación se presenta el registro de los horarios cumplidos por el recorrido:
EL DETALLE DEL RECORRIDO 1 ES EL SIGUIENTE :

REPORTE DE CONTRAVENCIONES

INFORMACIÓN DE PARADAS INCORRECTAS

EXCESO DE VELOCIDAD EN EL RECORRIDO

Figura 22. Interfaz de la página visualización reportes del almacenamiento web
Elaborado por: Edison Ojeda & Daisy Semanate

2.7.2 Implementación del circuito en el autobús

En la Figura 23, se muestra el módulo que se instaló en la unidad de transporte asignada para realizar las pruebas del sistema.

Prototipo instalado

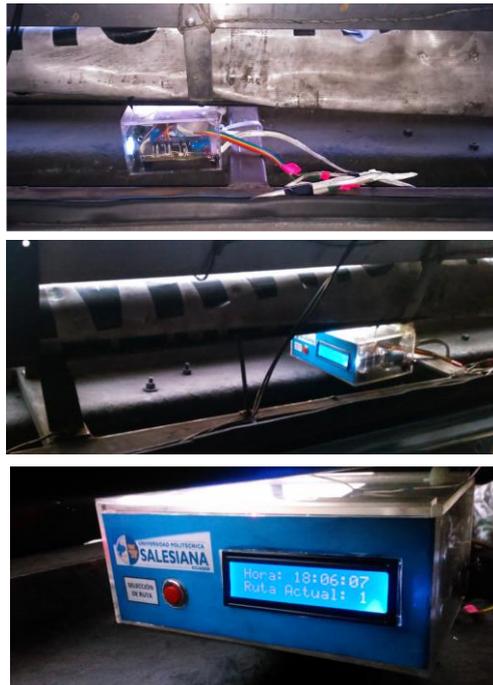


Figura 23. Módulos GPS, GPRS y microcontrolador instalados en la unidad de transporte
Elaborado por: Edison Ojeda & Daisy Semanate

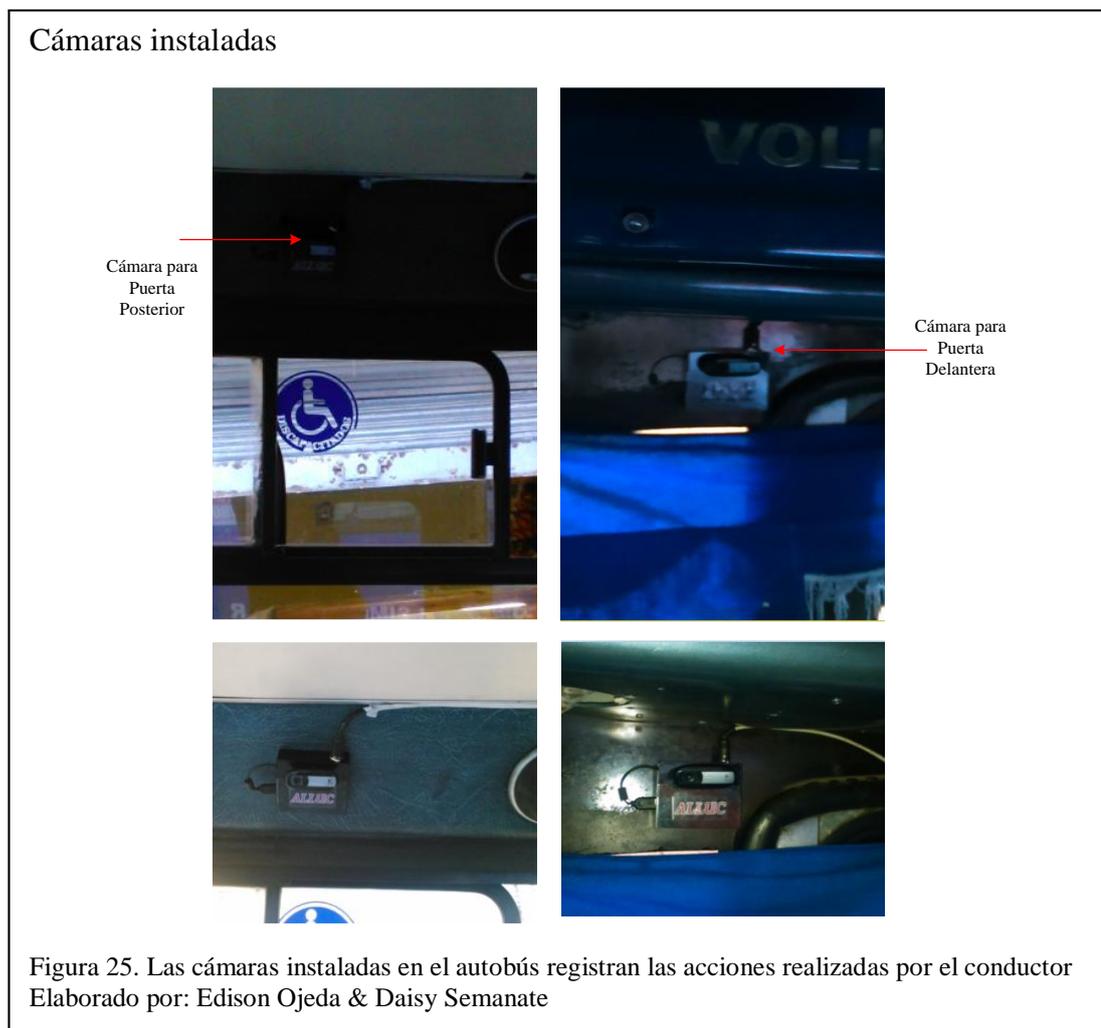
Los sensores que se instaló en las puertas delantera y posterior, se muestran en la imagen de la figura 24.

Sensores en las puertas



Figura 24. Cada puerta tiene un sensor magnético, que indica su estado (abierto o cerrado)
Elaborado por: Edison Ojeda & Daisy Semanate

Las cámaras de video encargadas de constatar los procedimientos realizados por el conductor se instalaron tanto en la parte delantera como en la parte posterior del autobús, como se muestran en la figura 25. Los videos grabados por las cámaras cuando las puertas del autobús se abren, se almacenan en una memoria flash extraíble incorporada en cada cámara. Teniendo en cuenta que la unidad de transporte realiza alrededor de 4000 paradas semanales y que cada video ocupa 400KB de memoria, implica que al mes ocupará aproximadamente 6.4GB, por tal razón la capacidad de almacenamiento de la memoria flash instalada es de 8GB.



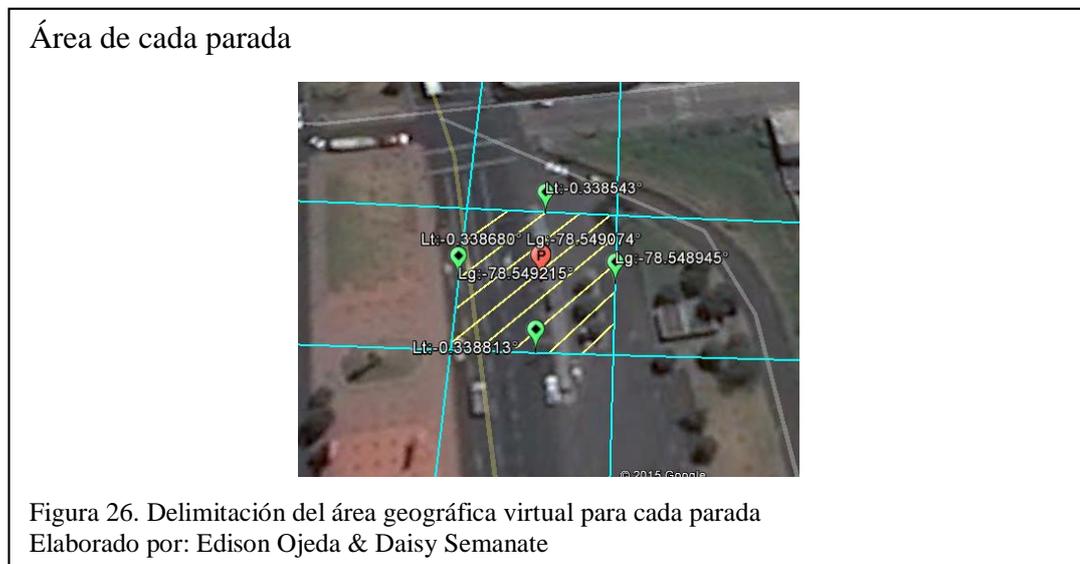
Una vez que se instalaron todos los componentes del prototipo en la unidad de transporte 151 de la Compañía Transplaneta con registro No. 2058 que se observa en el Anexo 18, se procede a realizar las pruebas de funcionamiento y análisis de resultados del sistema.

3 Pruebas y resultados

Para validar el desempeño del sistema implementado, en el presente capítulo se realizan pruebas de margen de error en el reconocimiento de paradas y pruebas de funcionamiento del prototipo, además se analizará los resultados obtenidos para determinar las conclusiones del proyecto.

3.1 Prueba de margen de error en el reconocimiento de paradas

A cada una de las coordenadas tomadas en las paradas del recorrido Cutuglagua-San Roque, se les asignó un área geográfica determinada capaz de identificar el momento en el que el autobús ingresa o abandona la parada, tal como se observa en la figura 26.



El área geográfica sombreada mostrada en la figura anterior, se realizó a partir de un algoritmo, que consiste en la adición y sustracción de coordenadas que representan 15 metros de distancia a cada valor de latitud y longitud. La distancia de 15 metros es el margen de error que tiene el autobús para arribar a la parada establecida.

En la figura 27 se muestran las operaciones realizadas para encontrar los límites del perímetro de aceptación en cada parada, para lo cual es necesario tener en cuenta los siguientes aspectos:

- **coor_límite:** representa tanto los valores de longitud mínima y máxima como los valores de latitud mínima y máxima de cada área geográfica.
- **coor_original:** representa los valores longitud y latitud de cada parada.
- **0.015 km:** es el valor de error que tiene cada bus para arribar o abandonar cada parada.
- **40075 km:** es la distancia de la línea ecuatorial.

Perímetro de paradas

$$\text{coor_límite} = \text{coor_original} \pm (0,015 \text{ km} / (40075 \text{ km} / 360^\circ))$$

$$\text{coor_límite} = \text{coor_original} \pm (0,015 / 111,3194)^\circ$$

$$\text{coor_límite} = \text{coor_original} \pm 0,000135^\circ$$

Figura 27. Operaciones para encontrar los límites del perímetro de aceptación de paradas
Elaborado por: Edison Ojeda & Daisy Semanate

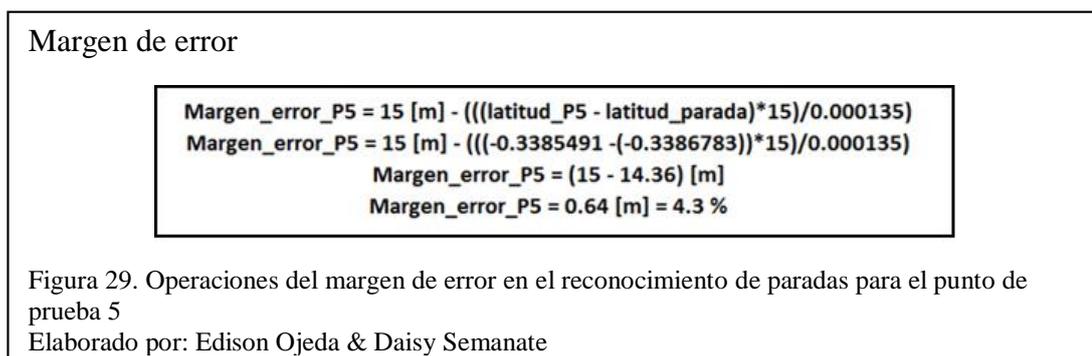
En base a lo expuesto anteriormente, se tomó como referencia la parada 8 (Antiguo Peaje S-N), para realizar las pruebas de exactitud. Se escogió 7 puntos de prueba como se muestra en la figura 28, donde se observa que los puntos 1, 4, 5, 6 y 7 posiblemente están dentro del diámetro permitido y los puntos 2 y 3 se puede afirmar con certeza que están fuera del límite establecido por lo que serán excluidos del posterior análisis.

Exactitud del perímetro de paradas

Figura 28. Puntos de prueba para determinar la precisión en el reconocimiento de paradas
Elaborado por: Edison Ojeda & Daisy Semanate

Para determinar la distancia de los puntos de prueba hasta las coordenadas de la parada en cuestión, se utilizan las operaciones que se muestra en la figura 29, con el fin de

establecer el margen de error que puede tener un autobús en el momento de ingresar o abandonar una parada establecida.



El detalle de los valores de prueba enviados por el módulo hacia la base de datos se muestra en la tabla 1. En el campo opción se indica con el valor 1 que las coordenadas han sido tomadas correctamente, es decir que están dentro del rango establecido, caso contrario con el valor 2 se indica que la parada es incorrecta o está fuera del área. En la tabla también se indica la distancia entre cada punto de prueba y el punto de referencia, donde se observa que los puntos de prueba 2, 3, 4 y 7 superan los 15 m por lo que el sistema los valida como paradas incorrectas, mientras que el punto 6 si está dentro del rango, sin embargo el sistema lo toma como parada incorrecta debido al error de exactitud que tiene el receptor GPS al momento de realizar la triangulación de los satélites para obtener su ubicación.

Tabla 1.
Datos obtenidos en la web

Ptos prueba	id informe	opcion	bus	ruta	latitud almacenada en BBDD	longitud almacenada en BBDD	hora	Latitud original de los ptos prueba	Distancia entre pto prueba y pto ref	Margen de error en [m]
Pto-referencia	1	1	151	1	-0,3386783	-78,5490798	15:21:18	-	-	-
Pto-1	2	1	151	1	-0,3386783	-78,5490798	15:22:56	-0,338807	14,30	0,70
Pto-2	3	2	151	1	-0,3388254	-78,5491039	15:24:07	-0,3388254	16,34	-
Pto-3	4	2	151	1	-0,3388187	-78,5490033	15:25:32	-0,3388187	15,60	-
Pto-4	5	2	151	1	-0,3385424	-78,54913	15:26:48	-0,3385424	15,10	0,10
Pto-5	6	1	151	1	-0,3386783	-78,5490798	15:28:02	-0,3385491	14,36	0,64
Pto-6	7	2	151	1	-0,3385451	-78,5490113	15:29:48	-0,3385451	14,80	0,20
Pto-7	8	2	151	1	-0,3385424	-78,5490469	15:32:17	-0,3385424	15,10	0,10

Nota: Elaborado por: Edison Ojeda & Daisy Semanate

El punto de prueba 5 es el último dato que el sistema valida como parada correcta, donde la distancia del punto de prueba 5 hasta el límite del perímetro establecido (15

m) es de 0.64 metros, siendo este el valor de margen de error mínimo equivalente al 4.33% que tiene el sistema en el momento de reconocer una parada como correcta, error que impidió que los puntos de prueba 4, 6 y 7 de la figura 28 sean válidos en la base de datos, puesto que su margen de error es menor a 0.64 m, como se observa en el caso del punto de prueba 6 que el margen de error es de 0.20 m.

3.2 Pruebas de funcionamiento

En este apartado, se explicará las pruebas que se realizaron para verificar el correcto funcionamiento de los módulos GPS, GPRS y también las pruebas de recepción de las tramas GPRS en la base de datos.

3.2.1 Pruebas de recepción de datos GPS

Para probar la eficiencia de la recepción de las sentencias NMEA, se realizó una prueba en el Terminal Br@y, donde se pudo observar como el módulo GPS obtiene las tramas de geolocalización proporcionadas por los satélites. En la figura 30, se observan las sentencias GPRMC Y GPVTG que se utilizaron en el desarrollo del proyecto, las cuales proporcionan datos de posición, fecha, hora y velocidad, tal como se expone en el apartado 2.2.1.1.

Visualización de sentencias NMEA

```

19.S,07829.71749,W,11.292,10.34,200715,...D*61
$GPVTG,10.34,T,M,11.292,N,20.913,K,D*DE
$GPGGA,232211.00,0010.83019,S,07829.71749,W,2.12,0.77,2855.9,M,13.8,M,0.000*61
$GPGSA,A,3,05,17,13,12,46,51,20,02,48,28,15,06,1,51,0.77,1.30*05
$GPGSV,4,1,16,02,24,349,44,05,66,302,44,06,15,029,27,09,00,050*75
$GPGSV,4,2,16,10,28,022,33,12,32,309,47,13,37,173,42,15,16,205,32*78
$GPGSV,4,3,16,17,38,090,39,20,28,239,33,24,09,244,24,28,14,153,23*75
$GPGSV,4,4,16,30,01,127,46,67,268,39,48,28,270,44,51,56,270,38*79
$GPRMC,232212.00,A,0010.82721,S,07829.71701,W,10.640,9.57,200715,...D*54
$GPVTG,9.57,T,M,10.640,N,19.706,K,D*3A
$GPGGA,232212.00,0010.82721,S,07829.71701,W,2.12,0.77,2855.8,M,13.8,M,0.000*62
$GPGSA,A,3,05,17,13,12,46,51,20,02,48,28,15,06,1,51,0.77,1.30*05
$GPGSV,4,1,16,02,24,349,44,05,66,303,45,06,15,029,33,09,00,050*76
$GPGSV,4,2,16,10,28,022,32,12,32,309,37,13,37,173,41,15,16,205,34*78
$GPGSV,4,3,16,17,38,090,40,20,28,239,38,24,09,244,21,28,14,153,23*75
$GPGSV,4,4,16,30,01,127,46,67,268,40,48,28,270,42,51,56,270,38*7F
$GPGLL,0010.82721,S,07829.71701,W,232212.00,A,D*6A

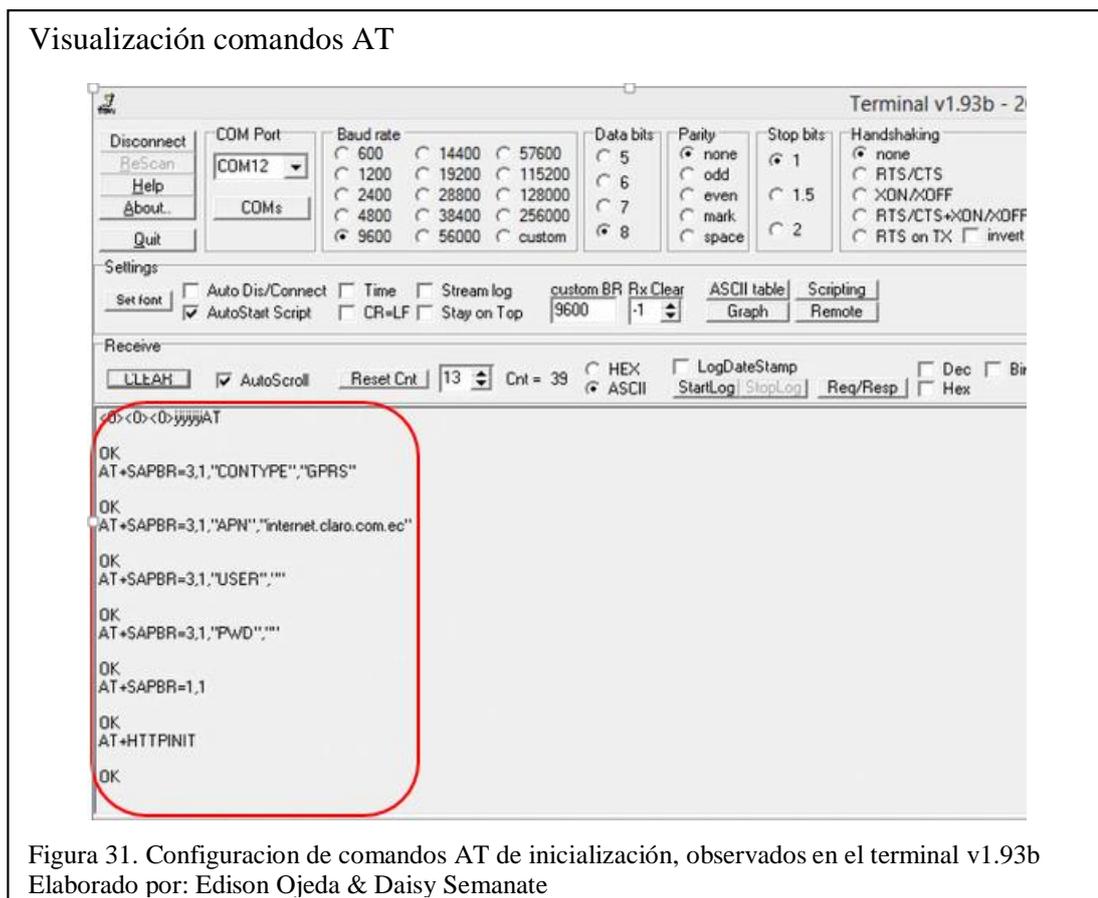
```

Figura 30. Recepción de sentencias NMEA en el terminal de prueba v1.93b
Elaborado por: Edison Ojeda & Daisy Semanate

3.2.2 Pruebas de transmisión de datos GPRS

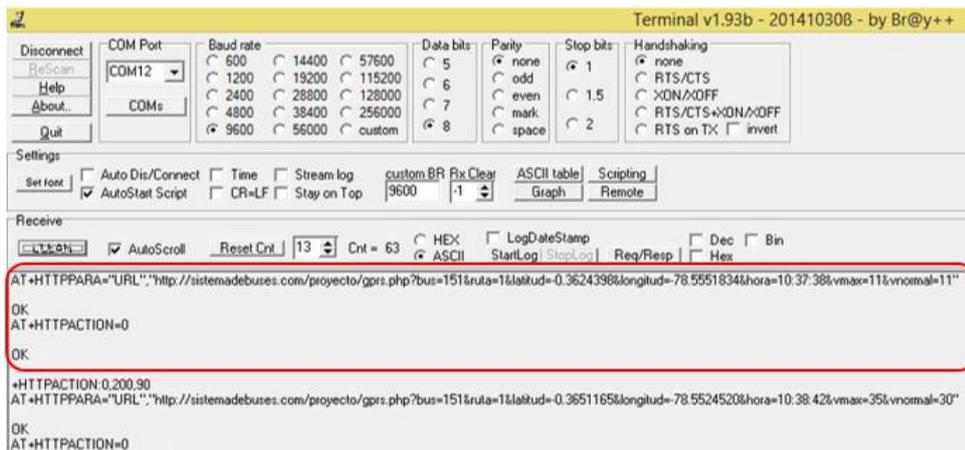
Para comprobar que el módem GPRS esté funcionando correctamente, se verificó mediante el Terminal Br@y que los comandos AT de inicialización del módulo, enviados por el microcontrolador obtengan respuesta satisfactoria, de modo que esté listo para enviar los datos procesados por el ATMEGA164P hacia la página web.

En la figura 31, se muestra el estado OK de los comandos AT que inicializan parámetros de autenticación con la APN, servicios HTTP y método de envío hacia la web del módulo GPRS, tal como se explica en el apartado 2.2.2.2.



El mismo Terminal Br@y es utilizado para visualizar las sentencias AT-HTTPPARA y AT-HTTPACTION señaladas en la figura 32, las cuales son las encargadas de enviar los datos de localización, ruta, hora y velocidades en formato URL, generados por el microcontrolador hacia la página web al momento que las puertas del autobús son abiertas.

Transmisión de datos a la web



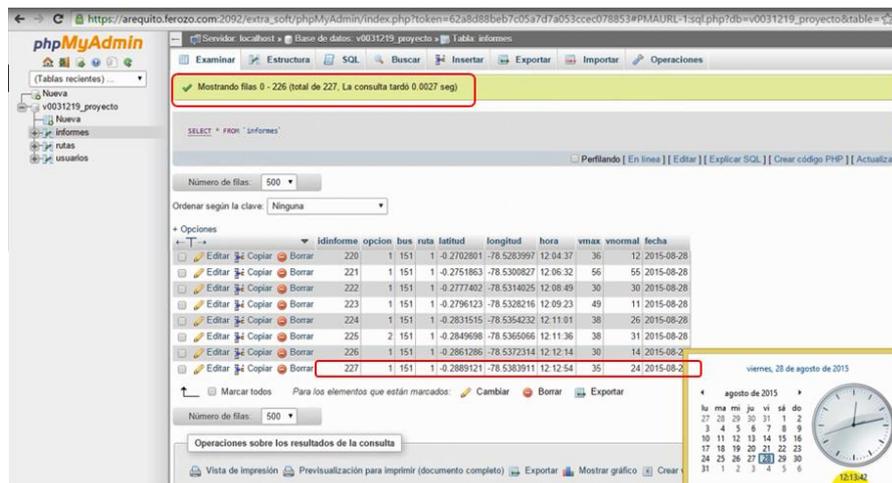
```
Terminal v1.93b - 20141030b - by Br@y++  
Disconnect: COM Port: COM12, Baud rate: 9600, Data bits: 8, Parity: none, Stop bits: 1, Handshaking: none  
Settings: Auto Dis/Connect, AutoStart Script, Time, Stream log, custom BR: 9600, Fix Clear, ASCII table, Scipiting  
Receive: AutoScroll, Reset Ctrl, 13, Cnt = 63, HEX, LogDateStamp, StartLog, StopLog, Req/Resp, Dec, Bin  
AT+HTTTPARA="URL":"http://sistemadebuses.com/proyecto/gprs.php?bus=151&ruta=1&latitud=0.3624398&longitud=-78.5551834&hora=10:37:38&vmax=11&vnormal=11"  
OK  
AT+HTTTPACTION=0  
OK  
+HTTTPACTION:0,200,90  
AT+HTTTPARA="URL":"http://sistemadebuses.com/proyecto/gprs.php?bus=151&ruta=1&latitud=0.3651165&longitud=-78.5524520&hora=10:38:42&vmax=35&vnormal=30"  
OK  
AT+HTTTPACTION=0
```

Figura 32. Trama GPRS que se envía a la web, observados en el terminal v1.93b
Elaborado por: Edison Ojeda & Daisy Semanate

3.2.3 Pruebas de recepción de información en la base de datos

En la figura 33, se muestra como la base de datos va almacenando las tramas enviadas por el GPRS. Como se puede observar se tiene 227 datos guardados, luego de unos 47 a 60 segundos se actualizó la base de datos y se verificó que se tenía un nuevo dato, como se puede observar en la figura 34. Esto refleja la confiabilidad que existe en la transmisión de datos, así como también que el tiempo de respuesta del sistema para la adquisición y almacenamiento de datos es menor a 1 minuto.

Base de datos



Mostrando filas 0 - 226 (total de 227. La consulta tardó 0.0027 seg)

Idinforme	opcion	bus	ruta	latitud	longitud	hora	vmax	vnormal	fecha
220	1	151	1	-0.2702801	-78.5283997	12:04:37	36	12	2015-08-28
221	1	151	1	-0.2751883	-78.5300827	12:06:32	56	55	2015-08-28
222	1	151	1	-0.2777402	-78.5314025	12:08:49	30	30	2015-08-28
223	1	151	1	-0.2796123	-78.5328216	12:09:23	49	11	2015-08-28
224	1	151	1	-0.2831515	-78.5354232	12:11:01	38	26	2015-08-28
225	2	151	1	-0.2849098	-78.5365066	12:11:36	38	31	2015-08-28
226	1	151	1	-0.2861286	-78.5372314	12:12:14	30	14	2015-08-28
227	1	151	1	-0.2889121	-78.5383911	12:12:54	35	24	2015-08-28

Figura 33. Datos de geolocalización almacenados constantemente en la base
Elaborado por: Edison Ojeda & Daisy Semanate

Base de datos actualizada

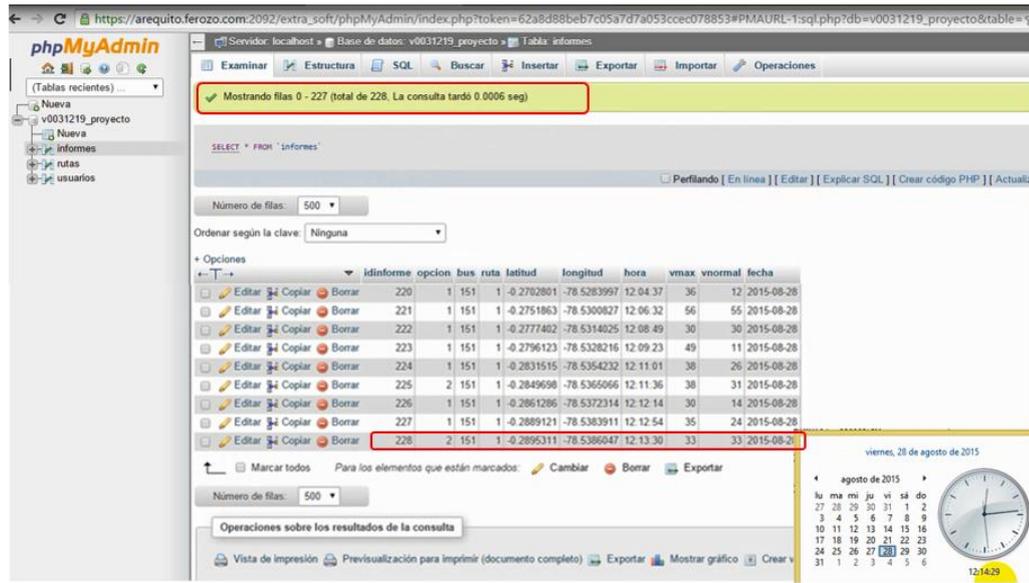


Figura 34. Actualización de datos de geolocalización almacenados en la base
Elaborado por: Edison Ojeda & Daisy Semanate

3.3 Análisis de resultados

Los resultados obtenidos de los recorridos realizados por la unidad 151 tomando como referencia una semana de recorrido, se muestran en la tabla 2.

Tabla 2.
Datos estadísticos del recorrido

Días	Horarios	Recorridos	Paradas Correctas	Paradas Incorrectas	Total Paradas	% Paradas Correctas	% Paradas Incorrectas	Promedio de Paradas Utilizadas
Lunes	05:30-19:00	5	349	169	518	67,37	32,63	70
Martes	04:45-18:15	5	393	117	510	77,06	22,94	79
Miércoles	04:30-19:30	6	462	195	657	70,32	29,68	77
Jueves	04:45-18:15	5	361	182	543	66,48	33,52	72
Viernes	05:00-18:30	5	355	179	534	66,48	33,52	71
Sábado	04:40-20:00	6	403	212	615	65,53	34,47	67
Domingo	05:00-16:30	4	320	75	395	81,01	18,99	80
Informe Semanal:		36	2643	1129	3772	70,07	29,93	73

Nota: Elaborado por Edison Ojeda & Daisy Semanate

Luego del análisis de la tabla 2, se tiene como resultado que en la semana que la unidad 151 cumple con el recorrido en la ruta Cutuglagua - San Roque, infringe en la contravención penada por la Ley de Transporte Terrestre, Tránsito y Seguridad Vial en el artículo 141 al no dejar y recoger pasajeros exclusivamente en las paradas señalizadas en un 29. 93% del recorrido total, y de las 81 paradas establecidas en la ruta sólo utiliza un promedio de 73 paradas por recorrido.

Los datos de los recorridos realizados durante la semana, los usuarios registrados lo pueden apreciar en la interfaz gráfica del aplicativo web, donde se puede observar en un mapa con indicadores o marcas de diferentes colores que hacen referencia a las paradas realizadas diariamente por el autobús en los distintos recorridos, como se muestra en la figura 35.



El detalle completo de las actividades realizadas por el automotor se las visualiza en la figura 36, donde se observa un registro de tiempos empleados en desplazarse entre las distintas paradas, coordenadas y horas en las que la unidad desembarcó y embarcó pasajeros indebidamente, y los trayectos en los que el bus excedió los límites de velocidad permitidos.

Reporte diario de actividades

RESULTADOS DEL RECORRIDO

A continuación se presenta el registro de los horarios cumplidos por el recorrido:

EL DETALLE DEL RECORRIDO 1 ES EL SIGUIENTE :

La unidad llegó a la parada: **P1 Cutuglagua Inicio** a las **05:05:44** horas
La unidad llegó a la parada: **P2 Cutuglagua 2 S-N** a las **05:06:46** horas
La unidad llegó a la parada: **P3 Entrada Cutuglagua S-N** a las **05:07:37** horas
La unidad llegó a la parada: **P4** a las **05:08:03** horas
La unidad llegó a la parada: **P5** a las **05:09:17** horas
La unidad llegó a la parada: **P6** a las **05:10:50** horas
La unidad llegó a la parada: **P7 Trefilec** a las **05:12:43** horas
La unidad llegó a la parada: **P8 Antiguo Peaje S-N** a las **05:13:57** horas
La unidad llegó a la parada: **P9 Guamaní S-N** a las **05:16:04** horas

REPORTE DE CONTRAVENCIONES

INFORMACIÓN DE PARADAS INCORRECTAS

La unidad abrió las puertas indevidamente en : LATITUD: **-0.3634374** ; LONGITUD: **-78.5500259** a las **05:07:57** HORAS
La unidad abrió las puertas indevidamente en : LATITUD: **-0.3502668** ; LONGITUD: **-78.5492248** a las **05:11:41** HORAS
La unidad abrió las puertas indevidamente en : LATITUD: **-0.33658** ; LONGITUD: **-78.5498275** a las **05:14:24** HORAS
La unidad abrió las puertas indevidamente en : LATITUD: **-0.3329478** ; LONGITUD: **-78.5503005** a las **05:15:32** HORAS
La unidad abrió las puertas indevidamente en : LATITUD: **-0.3251512** ; LONGITUD: **-78.550209** a las **05:18:19** HORAS
La unidad abrió las puertas indevidamente en : LATITUD: **-0.3248026** ; LONGITUD: **-78.5501561** a las **05:18:31** HORAS
La unidad abrió las puertas indevidamente en : LATITUD: **-0.3178356** ; LONGITUD: **-78.5491027** a las **05:19:54** HORAS
La unidad abrió las puertas indevidamente en : LATITUD: **-0.304289** ; LONGITUD: **-78.5427322** a las **05:24:13** HORAS
La unidad abrió las puertas indevidamente en : LATITUD: **-0.300873** ; LONGITUD: **-78.5421676** a las **05:25:32** HORAS
La unidad abrió las puertas indevidamente en : LATITUD: **-0.2999994** ; LONGITUD: **-78.5420227** a las **05:26:00** HORAS
La unidad abrió las puertas indevidamente en : LATITUD: **-0.2875928** ; LONGITUD: **-78.5376739** a las **05:30:07** HORAS

EXCESO DE VELOCIDAD EN EL RECORRIDO

La unidad excedió la VELOCIDAD hasta: **62 Km/h** ; entre las paradas **P3 Entrada Cutuglagua S-N** y **P4** del recorrido **1**
La unidad excedió la VELOCIDAD hasta: **58 Km/h** ; entre las paradas **P7 Trefilec** y **P8 Antiguo Peaje S-N** del recorrido **1**
La unidad excedió la VELOCIDAD hasta: **67 Km/h** ; entre las paradas **P11 Iglesia S-N** y **P12 Entrada Beaterio S-N** del recorrido **1**
La unidad excedió la VELOCIDAD hasta: **69 Km/h** ; entre las paradas **P78** y **P79 Entrada Cutuglagua N-S** del recorrido **1**
La unidad excedió la VELOCIDAD hasta: **56 Km/h** ; entre las paradas **P3 Entrada Cutuglagua S-N** y **P4** del recorrido **2**
La unidad excedió la VELOCIDAD hasta: **63 Km/h** ; entre las paradas **P4** y **P5** del recorrido **2**
La unidad excedió la VELOCIDAD hasta: **63 Km/h** ; entre las paradas **P6** y **P7 Trefilec** del recorrido **2**

Figura 36. Detalle de horarios y contravenciones del recorrido realizado
Elaborado por: Edison Ojeda & Daisy Semanate

Para ampliar la información citada en el presente capítulo, dirigirse a la página web <http://www.sistemadebuses.com>

El hosting contratado almacena hasta 1GB de información en la base de datos, siendo este suficiente para guardar los datos de este prototipo. En la tabla 3, se muestra una aproximación del espacio necesario en el hosting para almacenar la información que

generarían las 145 unidades que conforman la compañía de transporte de buses. Es así que el plan de hosting contratado resultaría ineficiente, debido a que la capacidad mínima necesaria para almacenar los datos de todas las unidades sería de 3.7GB.

Tabla 3.

Capacidad de almacenamiento en la BD del hosting DonWeb

Cant. de Buses	Cant. de semanas en el año	Total Paradas anuales	Tipo Hosting	Cant. De KB por Dato	Capacidad BD en MB	Datos almacenados en MB
1	53	199916	Inicial	0,125	1000	24,9895
145	53	28987820	Inicial	0,125	1000	3623,4775
145	53	28987820	Premium	0,125	10000	3623,4775

Nota: Elaborado por Edison Ojeda & Daisy Semanate

En la tabla 4, se visualiza el análisis comparativo entre los datos preliminares y los datos actuales obtenidos del recorrido realizado por la unidad 151, en donde se puede apreciar que el porcentaje mínimo de mejora del correcto uso de las paradas en cada recorrido es un 13% y en el mejor de los casos alcanza un porcentaje del 55%.

Tabla 4.

Comparación de datos preliminares con datos actuales del recorrido

Días	Recorridos	Paradas Incorrectas	# paradas incorrectas actuales / recorrido	# paradas incorrectas iniciales / recorrido	Reducción de faltas	% reducción de faltas
Lunes	5	169	33,80	42	8,20	20%
Martes	5	117	23,40		18,60	44%
Miércoles	6	195	32,50		9,50	23%
Jueves	5	182	36,40		5,60	13%
Viernes	5	179	35,80		6,20	15%
Sábado	6	212	35,33		6,67	16%
Domingo	4	75	18,75		23,25	55%

Nota: Elaborado por Edison Ojeda & Daisy Semanate

Una vez que se terminó con el desarrollo del sistema de monitoreo y se verificó su correcto funcionamiento, en el Anexo 19 se detallan los costos de implementación del prototipo instalado en la unidad 151 y de la proyección de implementación en las 145 unidades de la Compañía Planeta Transplaneta S.A.

Conclusiones

- Se implementó la página web con todas sus interfaces gráficas desarrolladas en el hosting DonWeb, donde se puede almacenar un máximo de 1GB información en la base de datos. Cada trama almacenada ocupa 0.125 KB de espacio, lo que significa que una unidad de bus utilizaría al año 24.99 MB de espacio en el disco, mientras que las 145 unidades de la compañía ocuparían 3.623 GB, por lo tanto, sería necesario un hosting con mayor espacio de almacenamiento.
- Mediante pruebas de funcionamiento, se verificó que el sistema presenta un error aproximado de 65cm que equivale al 4.33% al momento de reconocer una parada señalizada, y que todos los datos enviados por el hardware se actualizan entre 47 y 60 segundos en la base de datos, siendo retardos de tiempo despreciables con relación a la frecuencia con que un usuario realiza una consulta al sitio web para visualizar un reporte de recorrido.
- Con el análisis de resultados, se constató que el sistema redujo de un 34.14% a un 29.93% las paradas incorrectas respecto a los datos preliminares, debido a que el conductor realizó un promedio de 36 paradas no autorizadas de un total de 109, mientras que en los datos preliminares realizó un promedio de 42 paradas incorrectas de un total de 123, estableciendo una reducción en los tiempos de retardo de cada recorrido de 15 a 5 minutos.
- Las pruebas realizadas a partir del prototipo instalado en la unidad 151 de la compañía, mejoraron el uso correcto de las paradas hasta un 55% en días con baja afluencia vehicular y un 13% en días con alta congestión vehicular con relación a los datos iniciales obtenidos antes de instalar el sistema de monitoreo, permitiendo retirar las máquinas de registro actuales y facilitar la sanción de las infracciones cometidas puesto que el reporte del recorrido realizado se encuentra alojado en la base de datos.

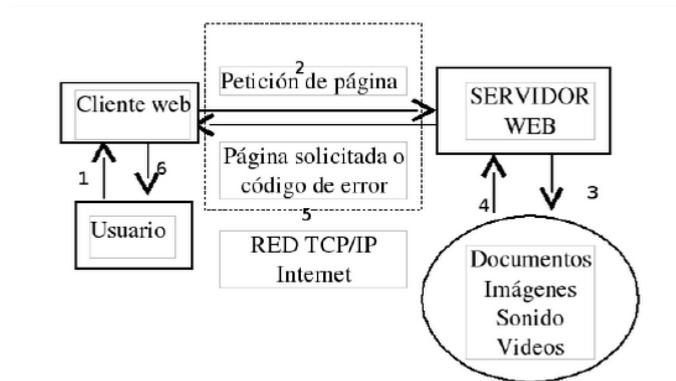
Recomendaciones

- Se recomienda hacer un análisis previo del sitio donde se va a ubicar el dispositivo electrónico del sistema antes de instalarlo en el autobús, puesto que para obtener una mejor recepción de las señales satelitales la antena receptora del módulo GPS debe estar situada con línea de vista al cielo.
- En función de ahorrar tiempo en el desarrollo del proyecto y tener la mayor veracidad en la ubicación de las paradas establecidas por las autoridades del municipio de Quito en cada recorrido, se sugiere realizar una investigación previa del tema en conjunto con esta entidad, debido a que actualmente en campo las paradas de transporte público colectivo carecen de organización y coherencia en su ubicación en la mayoría de recorridos.
- Para la producción macro del prototipo desarrollado en el presente proyecto, es necesario incrementar la capacidad de almacenamiento en la base de datos, debido a que su tamaño en el plan inicio del hosting contratado en la compañía DonWeb, está limitado a guardar información hasta 1GB, por lo que se recomienda contratar un plan con mayor capacidad de almacenamiento como lo es el Cloud Hosting, el cual proporciona de 10 a 60 GB.
- Para trabajos futuros se plantea el mejoramiento de la interfaz gráfica del sitio web añadiendo interactividad y dinamismo a cada página, de manera que además de las funcionalidades que posee actualmente, incorpore tablas flexibles de organización de datos, enlaces dinámicos entre los detalles del recorrido y el mapa de Google, y generación de reportes del recorrido en formato PDF.

Referencias

- Atmel. (Septiembre de 2015). *Atmel*. Obtenido de Atmel:
http://www.atmel.com/Images/Atmel-8011-8-bit-AVR-Microcontroller-ATmega164P-324P-644P_datasheet.pdf
- Bluehack. (2005). *Bluehack*. Recuperado el 01 de Agosto de 2015, de Bluehack:
<http://bluehack.elhacker.net/proyectos/comandosat/comandosat.html>
- Cobo, Á., Gómez, P., & Pérez, D. (2007). *PHP y MySQL: tecnologías para el desarrollo de aplicaciones web*. España: Ediciones Díaz de Santos.
- Ferrer Martínez, J. (2014). *PHP*. España: RA-MA Editorial.
- Giménez Guzmán, J. M., López Merayo, M. T., & Moreno Martín, M. (2012). *Aplicaciones de la Web*. España: Servicio de Publicaciones. Universidad de Alcalá.
- GlobalTop Technology Inc. (2011). *FGPMMOPA6H Data Sheet*. Obtenido de FGPMMOPA6H Data Sheet:
<https://www.adafruit.com/datasheets/GlobalTop-FGPMMOPA6H-Datasheet-VOA.pdf>
- López, B. (13 de mayo de 2013). *Ciudadano 2.0*. Obtenido de Ciudadano 2.0:
<http://www.ciudadano2cero.com/que-es-un-hosting-web-tipos-alojamiento-cual-elegir/>
- Martín Cuenca , E., Angulo Uzategui, J. M., & Angulo Martínez, I. (2003). *PBASIC, el lenguaje más fácil del mundo*. Madrid: Internacional Thomson.
- Mifsuf Talón, E. (2012). *Introducción a Apache*. España: Ministerio de Educación de España.
- Rodrigo, J. G., & Santiago, G. M. (2002). *Transmisión de datos sobre GSM (GPRS y EDGE)*. Madrid: Ediciones Paraninfo.
- Sánchez Morales, M. (1999). Instalación de WampServer. En M. Sánchez Morales, *Manual de Desarrollo Web* (pág. 86). España: Nick Popio Lulu Enterprise.
- u-blox. (11 de Noviembre de 2014). *ublox*. Obtenido de ublox: [https://www2.u-blox.com/images/downloads/Product_Docs/NEO-7_DataSheet_\(GPS.G7-HW-11004\).pdf](https://www2.u-blox.com/images/downloads/Product_Docs/NEO-7_DataSheet_(GPS.G7-HW-11004).pdf)
- Vaswani, V. (2010). En V. Vaswani, *Fundamentos de PHP* (págs. 6,7). México: McGraw-Hill Interamericana.
- Velasquez, R. (24 de Septiembre de 2013). *CODEHERO*. Obtenido de CODEHERO:
<http://codehero.co/php-desde-cero-estructuras-de-control>

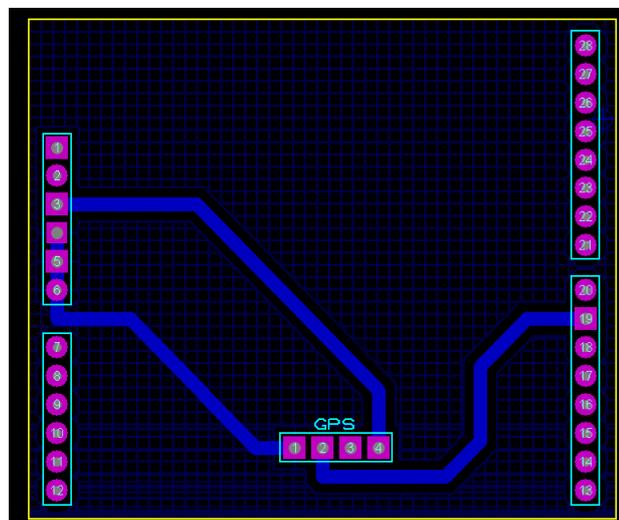
Anexo 4. Funcionamiento del protocolo HTTP.



Anexo 5. Módulo GPS u-blox-7m.



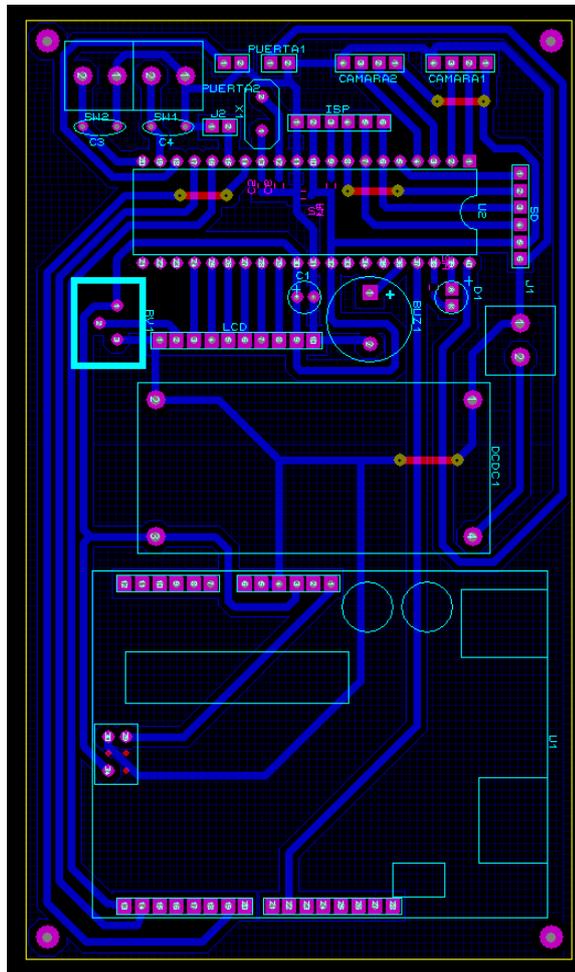
Anexo 6. Diseño PCB del módulo u-blox-7m.



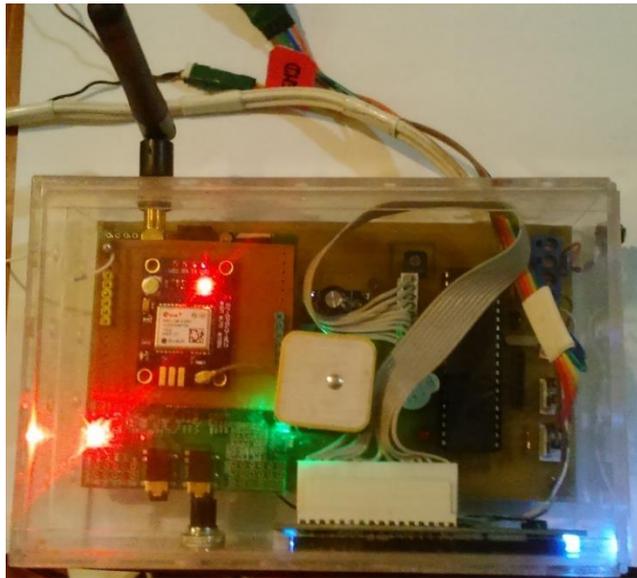
Anexo 7. Módulo GPRS sim900.



Anexo 8. Diseño PCB general del hardware del sistema.



Anexo 9. Dispositivo de geolocalización ensamblado.



Anexo 10. Vista frontal del hardware del sistema.



Anexo 11. Cámaras de control.



Anexo 12. Instalación de SIM CARD en módulo GPRS.



Anexo 13. Ventana de información del WampServer.

WampServer Version 2.5 Version Française

Server Configuration
 Apache Version : 2.4.9 - Documentation
 PHP Version : 5.5.12 - Documentation
 Server Software: Apache/2.4.9 (Win64) PHP/5.5.12

Loaded Extensions:

- apache2handler
- Core
- ereg
- gd
- imap
- mhash
- openssl
- Phar
- soap
- tokenizer
- xmlrpc
- bcmath
- ctype
- exif
- gettext
- json
- mysql
- pcrc
- Reflection
- sockets
- wddx
- xmlwriter
- bz2
- curl
- fileinfo
- gmp
- libxml
- mysqli
- PDO
- session
- SPL
- xdebug
- xsl
- calendar
- date
- filter
- hash
- mbstring
- mysqlnd
- pdo_mysql
- shmop
- sqlite3
- xml
- zip
- com_dotnet
- dom
- ftp
- iconv
- mcrypt
- odbc
- pdo_sqlite
- SimpleXML
- standard
- xmlreader
- zlib

MySQL Version : 5.6.17 - Documentation

Tools
 phpinfo()
 phpmyadmin

Your Projects
 proyecto

Your Aliases
 phpmyadmin
 phpsysinfo
 sqlbuddy
 webgrind

WampServer Donate Alter Way

Anexo 14. Implementación base de datos “proyecto” con las tablas informes, rutas, y usuarios.

localhost / mysql wampserver: X

localhost/phpmyadmin/#PMAURL-17:db_structure.php?db=proyecto&table=&server=1&target=&token=e992eb3c73530f8500bdce75ac00a118

Servidor: mysql wampserver » Base de datos: proyecto

Tabla	Acción	Archivos	Tipo	Cotejamiento	Tamaño
informes	Examinar Estructura Buscar Insertar Vaciar Eliminar	-11	InnoDB	latin1_swedish_ci	16 KB
rutas	Examinar Estructura Buscar Insertar Vaciar Eliminar	-34	InnoDB	latin1_swedish_ci	16 KB
usuarios	Examinar Estructura Buscar Insertar Vaciar Eliminar	-6	InnoDB	latin1_swedish_ci	16 KB
3 tablas	Número de filas	51	InnoDB	latin1_swedish_ci	48 KB

↑ Marcar todos Para los elementos que están marcados: ↓

Vista de impresión Diccionario de datos

Campos de tabla informes

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	idinforme	int(11)			No	Ninguna	AUTO_INCREMENT
2	opcion	int(11)			No	Ninguna	
3	bus	int(11)			No	Ninguna	
4	ruta	int(11)			No	Ninguna	
5	latitud	double			No	Ninguna	
6	longitud	double			No	Ninguna	
7	hora	time			No	Ninguna	
8	vmax	int(11)			No	Ninguna	
9	vnormal	int(11)			No	Ninguna	
10	fecha	date			No	Ninguna	

Campos de tabla rutas

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	idruta	int(11)			No	Ninguna	AUTO_INCREMENT
2	ruta	int(11)			No	Ninguna	
3	latitudfija	double			No	Ninguna	
4	longitudfija	double			No	Ninguna	
5	lonmin	double			No	Ninguna	
6	lonmax	double			No	Ninguna	
7	lamin	double			No	Ninguna	
8	lamax	double			No	Ninguna	

Campos de tabla usuarios

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	idusuario	int(11)			No	Ninguna	AUTO_INCREMENT
2	nombreusuario	varchar(50)	latin1_swedish_ci		No	Ninguna	
3	claveusuario	varchar(50)	latin1_swedish_ci		No	Ninguna	
4	tipousuario	int(11)			No	Ninguna	
5	fecharegistro	date			No	Ninguna	

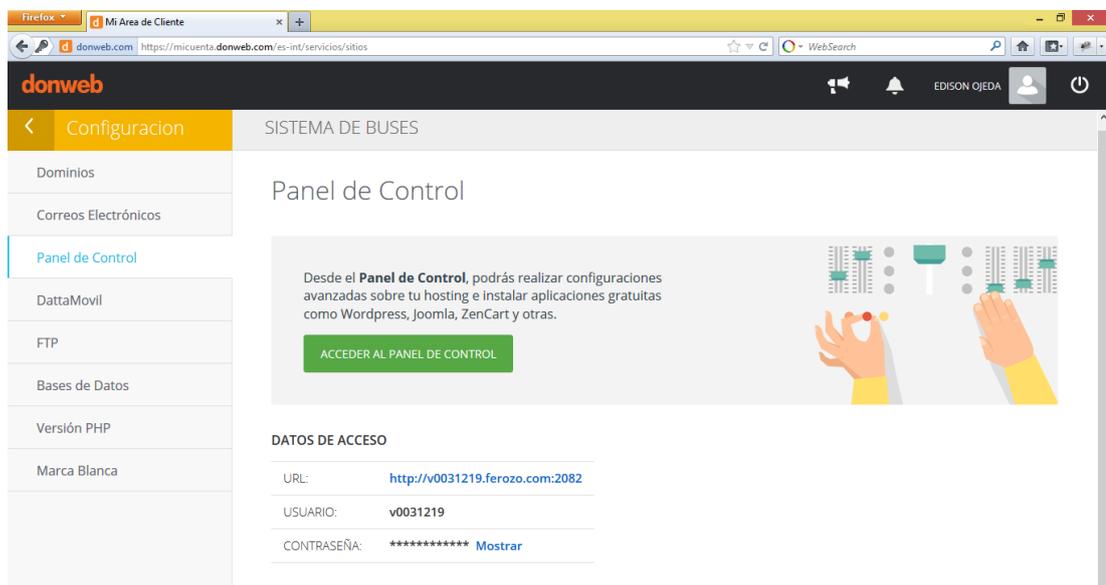
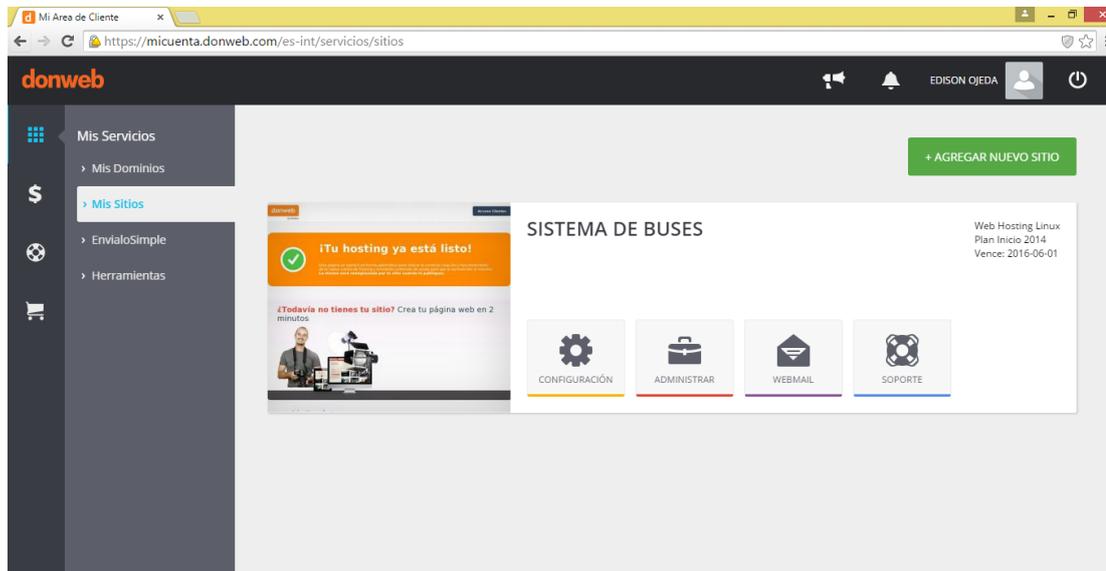
Anexo 15. Etiquetas HTML.

ETIQUETA	DESCRIPCIÓN
<html> </html>	Al principio y al final de todo documento.
<head> </head>	Cabecera del documento.
<title> </title>	Indica el título de la página para el navegador.
<meta charset />	El atributo charset especifica la codificación de caracteres para el documento HTML.
<body> </body>	Dentro de esta etiqueta se insertan los contenidos del documento. El cierre de la etiqueta </body> se coloca justo antes del cierre </html>.

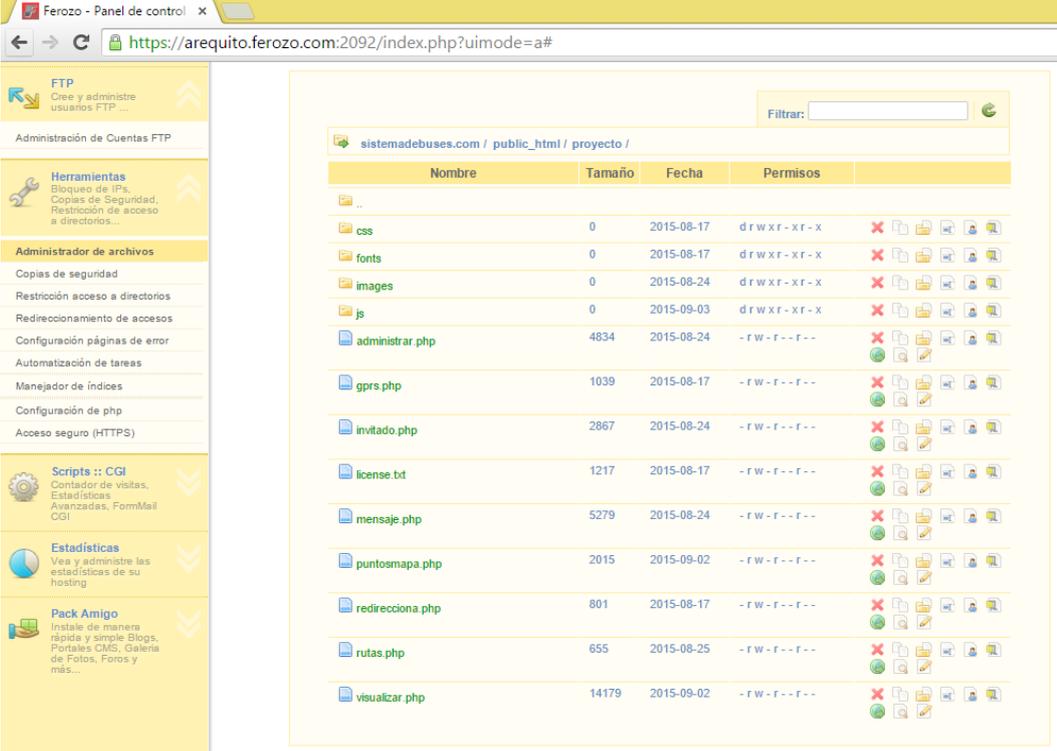
<link/>	Define un enlace entre el documento y algún recurso externo.
 	negrita (también sirve la etiqueta ...).
 	Puede incluir tres parámetros (tamaño, fuente y color):
<div> </div>	Se utiliza para definir una sección dentro del documento.
<div class> </div>	Asigna un nombre de clase.
<div id> </div>	Asigna un nombre a un elemento.
<h1> </h1>	Texto muy grande.
<h2> </h2>	Texto grande.
<h3> </h3>	Texto algo más grande de lo normal.
 	Espacio en blanco.
<p></p>	Salto de párrafo.
 	Salto de línea.
<p align=center>	Párrafo centrado.
<p align=left>	Párrafo alineado a la izquierda.
<p align=right>	Párrafo alineado a la derecha.
	Indica la ruta de la imagen.
	Establece un borde de X píxeles en torno a la imagen.
	Establece un tamaño de la imagen (altura y anchura) en píxeles.
<table>.....</table>	Define dónde comienza y termina la tabla.
<tr>.....</tr>	Determina cada una de las filas de la tabla.
<td>.....</td>	Determina cada una de las columnas dentro de las filas.
 	Enunciado del enlace.
FORMULARIOS	DESCRIPCIÓN
<form> </form>	Todo lo que vaya dentro de estas etiquetas, serán partes del formulario.
<form method="get" action="datos.php"> </form>	El atributo " action " indica el tipo de acción que va a realizar el formulario. El atributo " method " indica al formulario la forma en la que el formulario será enviado. Existen dos valores posibles: get y post .
<input type="text" id="nombredelacaja" name="nombredelacaja"/>	Caja de texto básica.
<input type="password" name="contraseña"/>	Crea una caja de texto con caracteres ocultos.
<input type="submit" value="Enviar información"/>	Botón que indica que el formulario ha concluido y que puede enviar la información.
<input type="button" value="Ingresar"/>	Se crea un botón pulsador.

<label> </label>	Se utiliza para adjuntar información a los controles. El valor del atributo for debe ser igual al valor id del elemento input para que los reconozca como asociados.
<select> </select>	Crea un menú desplegable. Cada opción ofrecida por dicho menú se representa por un elemento option .

Anexo 16. Hosting DonWeb



Anexo 17. Archivos alojados en el hosting DonWeb.



The screenshot shows a web hosting control panel interface. The browser address bar displays <https://arequito.ferozo.com:2092/index.php?uimode=a#>. The main content area shows a file directory listing for the path `sistemadebuses.com / public_html / proyecto /`. The listing includes columns for file name, size, date, and permissions. The files listed are:

Nombre	Tamaño	Fecha	Permisos
..			
css	0	2015-08-17	drwxr-xr-x
fonts	0	2015-08-17	drwxr-xr-x
images	0	2015-08-24	drwxr-xr-x
js	0	2015-09-03	drwxr-xr-x
administrar.php	4834	2015-08-24	-rw-r--r--
gprs.php	1039	2015-08-17	-rw-r--r--
invitado.php	2867	2015-08-24	-rw-r--r--
license.txt	1217	2015-08-17	-rw-r--r--
mensaje.php	5279	2015-08-24	-rw-r--r--
puntosmapa.php	2015	2015-09-02	-rw-r--r--
redirecciona.php	801	2015-08-17	-rw-r--r--
rutas.php	655	2015-08-25	-rw-r--r--
visualizar.php	14179	2015-09-02	-rw-r--r--

Anexo 18. Unidad de transporte 151 donde se instaló el sistema.



Anexo 19. Costos del prototipo y proyección para implementación en las 145 unidades de la Compañía Planeta Transplaneta S.A.

SISTEMA DE MONITOREO PARA LA COMPAÑIA DE TRANSPORTES PLANETA TRANSPLANETA S.A UTILIZANDO TECNOLOGÍAS INALÁMBRICAS GPS Y GPRS

COSTOS	ITEM	CANT	PRECIO UNITARIO	PRECIO TOTAL
MATERIALES	GPS UBLOX NEO 7M	1	42,00	\$ 42,00
	GPRS SIM900	1	90,00	\$ 90,00
	CAMARAS	2	90,00	\$ 180,00
	ATMEGA164P	1	10,00	\$ 10,00
	SENSORES	2	5,00	\$ 10,00
	FUENTE DC-DC	1	4,00	\$ 4,00
	LCD 16x2	1	6,00	\$ 6,00
	CAJA EN ACRÍLICO	1	35,00	\$ 35,00
	CABLE MULTIPAR (m)	10	0,50	\$ 5,00
	CABLE GEMELO #18	4	0,35	\$ 1,40
	PCB	1	5,00	\$ 5,00
MATERIALES VARIOS	1	11,00	\$ 11,00	
Total MATERIALES				\$ 399,40
FUNCIONAMIENTO	ALOJAMIENTO WEB ANUAL	1	42,00	\$ 42,00
	CONSUMO PLAN DE DATOS OPERADORA CLARO ICLUIDO SIM MENSUAL	12	11,93	\$ 143,16
Total FUNCIONAMIENTO				\$ 185,16
DISEÑO E INSTALACIÓN	PROGRAMACIÓN AVR (HORAS)	20	20,00	\$ 400,00
	PROGRAMACIÓN WEB (HORAS)	40	20,00	\$ 800,00
	IMPLEMENTACIÓN DEL HARDWARE (HORAS)	10	20,00	\$ 200,00
Total DISEÑO E INSTALACIÓN				\$ 1.400,00
COSTOS INDIRECTOS	MOVILIZACIÓN	2	250,00	\$ 500,00
	ALIMENTACIÓN			
	GASTOS MENORES			
Total COSTOS INDIRECTOS				\$ 500,00
Total general				\$ 2.484,56

Precio por Instalación por dispositivo para la compañía que cuenta con 145 Buses

COSTOS	ITEM	CANT	PRECIO UNITARIO	DESCUENTO	PRECIO TOTAL AÑO 1	PRECIO TOTAL AÑO 2	PRECIO TOTAL AÑO 3
MATERIALES	GPS UBLOX NEO 7M	1	42,00	12,60	\$ 29,40	\$ -	\$ -
	GPRS SIM900	1	90,00	27,00	\$ 63,00	\$ -	\$ -
	CAMARAS	2	90,00	27,00	\$ 126,00	\$ -	\$ -
	ATMEGA164P	1	10,00	3,00	\$ 7,00	\$ -	\$ -
	SENSORES	2	5,00	1,50	\$ 7,00	\$ -	\$ -
	FUENTE DC-DC	1	4,00	1,20	\$ 2,80	\$ -	\$ -
	LCD 16x2	1	6,00	1,80	\$ 4,20	\$ -	\$ -
	CAJA EN ACRÍLICO	1	35,00	10,50	\$ 24,50	\$ -	\$ -
	CABLE MULTIPAR (m)	10	0,50	0,15	\$ 3,50	\$ -	\$ -
	CABLE GEMELO #18	4	0,35	0,11	\$ 0,98	\$ -	\$ -
	PCB	1	10,00	-	\$ 10,00	\$ -	\$ -
	MATERIALES VARIOS	1	11,00	-	\$ 11,00	\$ -	\$ -
Total MATERIALES					\$ 289,38	\$ -	\$ -
FUNCIONAMIENTO	ALOJAMIENTO WEB ANUAL	1	240,00	-	\$ 1,66	\$ 1,66	\$ 1,66
	CONSUMO PLAN DE DATOS OPERADORA CLARO ICLUIDO SIM MENSUAL	12	8,00	-	\$ 96,00	\$ 96,00	\$ 96,00
Total FUNCIONAMIENTO					\$ 97,66	\$ 97,66	\$ 97,66
DISEÑO E INSTALACIÓN	PROGRAMACIÓN AVR (HORAS)	2	20,00	-	\$ 40,00	\$ -	\$ -
	PROGRAMACIÓN WEB (HORAS)	1	800,00	-	\$ 5,52	\$ -	\$ -
	IMPLEMENTACIÓN DEL HARDWARE (HORAS)	2	20,00	-	\$ 40,00	\$ -	\$ -
Total DISEÑO E INSTALACIÓN					\$ 85,52	\$ -	\$ -
COSTOS INDIRECTOS	MOVILIZACIÓN	2	7,00	-	\$ 14,00	\$ -	\$ -
	ALIMENTACIÓN					\$ -	\$ -
	GASTOS MENORES					\$ -	\$ -
Total COSTOS INDIRECTOS					\$ 14,00	\$ -	\$ -
MANEJO	MANO DE OBRA	2	10,00	-	\$ -	\$ 20,00	\$ 20,00
Total MANTENIMIENTO					\$ -	\$ 20,00	\$ 20,00
Total general					\$ 486,55	\$ 117,66	\$ 117,66
Valor a Pagar Mensual					\$ 40,55	\$ 9,80	\$ 9,80