

SISTEMA DE CAJA NEGRA PARA MONITOREO Y REGISTRO DE
AUTOMOTORES TIPO BUSETA

MAURICIO FERNANDO GUTIÉRREZ LANCHEROS
JOSÉ FELIPE ZAMBRANO PÉREZ
ALEXANDER GORDILLO GAITAN

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
NEIVA-HUILA

2013

SISTEMA DE CAJA NEGRA PARA MONITOREO Y REGISTRO DE
AUTOMOTORES TIPO BUSETA

MAURICIO FERNANDO GUTIÉRREZ LANCHEROS

JOSÉ FELIPE ZAMBRANO PÉREZ

ALEXANDER GORDILLO GAITAN

Trabajo de grado para obtener el título de ingeniero electrónico

DIRECTOR

GERMAN EDUARDO MARTINEZ BARRETO

Ingeniero Electrónico Especialista en Sistema Dinámicos y Automatización
Industrial

UNIVERSIDAD SURCOLOMBIANA

FACULTAD DE INGENIERÍA

INGENIERÍA ELECTRÓNICA

NEIVA-HUILA

2013

Nota de aceptación

Firma del director de proyecto

Firma del primer jurado

Firma del segundo jurado

Neiva, 2 de septiembre del 2013

Agradezco especialmente a mi padre por guiarme en esta importante etapa de mi vida, además de enseñarme, exigirme y apoyarme en todo momento. Agradezco especialmente a mi madre por inculcarme los valores necesarios para ser una gran persona. A todas esas personas que me aconsejaron, me apoyaron y celebraron mis éxitos. ¡Muchas gracias!

MAURICIO FERNANDO GUTIERREZ

Agradezco a mis padres José Jairo Zambrano F. y Carmen Evelia Pérez por su continuo acompañamiento, colaboración y paciencia durante toda esta etapa de la vida. A mis hermanas quienes me aconsejaron y estuvieron pendientes de que manera podían ayudarme. Y a todas aquellas personas que agregaron de su parte para que todo esto fuera posible.

JOSE FELIPE ZAMBRANO PEREZ

Agradezco a mi familia, mis padres Mary Luz Gaitán G. y Raúl Gordillo L. por la formación y autonomía en mi proceso educativo, a la construcción como persona una integral y profesional. A las personas que incondicionalmente me han apoyado y siempre he podido contar con ellas.

ALEXANDER GORDILLO GAITAN

AGRADECIMIENTOS

A la Universidad Surcolombiana por permitirnos una formación integral, a los docentes por acompañarnos y guiarnos durante el proceso de aprendizaje como profesionales.

A los Ingenieros Germán Martínez Barreto, Edilberto Polanía Puentes y Vladimir Mosquera por su apoyo, aporte en ideas y acompañamiento en el proceso del proyecto de grado.

El apoyo en nuestro proyecto de grado a Henry Ruiz, técnico en manómetros por compartir sus conocimientos en mecánica.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	15
OBJETIVO.....	16
OBJETIVOS ESPECIFICOS	16
ESTADO DEL ARTE.....	17
JUSTIFICACIÓN.....	20
1 DIAGRAMA GENERAL DEL SISTEMA	24
1.1 Sensores	24
1.2 Caja negra	25
1.3 Dispositivo móvil.....	25
1.4 Entorno al usuario.....	25
1.5 Cámara IP	25
2 ARDUINO.....	26
2.1 ¿Qué es Arduino?.....	26
2.2 Características Arduino Mega ADK	30
2.3 Software Arduino	32
2.3.1 Lenguaje de programación Arduino	35
2.3.2 Operadores	35
3 ANDROID.....	37
3.1 Arquitectura	37
3.2 Versiones.....	39
3.3 Herramientas para el desarrollo.....	40
3.4 Componentes de una aplicación.....	42
3.4.1 Actividades.....	43
3.4.2 Servicios.....	44
4 SENSORES	48
4.1 Velocímetro	48
4.1.1 Efecto hall	48
4.1.2 Sensor de efecto Hall.....	48

4.1.3	Circuito acondicionador del sensor de velocidad	49
4.2	Medidor de combustible.....	51
4.3	Acelerómetro	53
5	RECEPCIÓN Y TRANSMISIÓN DE ARDUINO, ANDROID Y PHP	57
5.1	Recepción Arduino	57
5.2	Transmisión Arduino a celular	58
5.3	Recepción al celular	60
5.4	Recepción del GPS	61
5.4.1	Software GPS móvil	62
5.5	Recepción de la hora.....	63
5.6	Proceso de comunicación entre celular y PHP	63
5.6.1	JSON	63
6	CONFIGURACIÓN DEL SERVIDOR	65
6.1	XAMPP	65
6.2	PHP	65
6.2.1	Bloque de recepción	66
6.2.2	Bloque base de datos y otros.....	67
6.2.3	Bloque interfaz de usuario.....	69
6.3	Cámara IP	73
6.4	Portal web.....	78
6.4.1	Interfaz de usuario portal Web	78
7	PRESUPUESTO	80
8	CONCLUSIONES.....	81
9	RECOMENDACIONES	83
10	FUENTES DE CONSULTA.....	85
10.1	Bibliográficas	85
10.2	Medios virtuales.....	85
11	ANEXOS.....	89

LISTA DE FIGURAS

Ilustración 1. Diagrama general del sistema.	24
Ilustración 2. Arduino Mega.	27
Ilustración 3. Arduino con un "Shield".	28
Ilustración 4. Arduino Severino.	28
Ilustración 5. Arduino Mega ADK.	30
Ilustración 6. Ventana de administrador de dispositivos.	32
Ilustración 7. Entorno de trabajo Arduino.	33
Ilustración 8. Ejemplos IDE Arduino.	33
Ilustración 9. Código ejemplo Arduino.	34
Ilustración 10. Selección de placa.	34
Ilustración 11. Capa de aplicaciones.	37
Ilustración 12. Capa de Frameworks.	38
Ilustración 13. Capa de librerías y capa Runtime.	38
Ilustración 14. Capa Kernel de Linux.	39
Ilustración 15. Configuración de variables de entorno.	40
Ilustración 16. Configuración del SDK en path del sistema.	41
Ilustración 17. Configuración del plugin de Android en Eclipse.	41
Ilustración 18. Pila de actividades de Android.	42
Ilustración 19. Ciclo de vida de una actividad.	44
Ilustración 20. Ciclo de vida de un servicio.	45
Ilustración 21. Código generado al crear el proyecto en Android Manifest.	46
Ilustración 22. Código generado librería usb.	47
Ilustración 23. Etiquetas Intent Filter.	47
Ilustración 24. Indicador de Velocidad.	49
Ilustración 25. Simulación LM2907.	51
Ilustración 26. Sistema medidor de combustible.	52
Ilustración 27. Comprobación de deflexión de la aguja.	52
Ilustración 28. Medidor de combustible.	53
Ilustración 29. Ejemplos de aceleraciones.	54
Ilustración 30. Acelerómetro de 3 ejes MMA7361L con regulador de voltaje, vista con dimensiones.	55
Ilustración 31. Simulación circuito indicador de vibraciones.	55
Ilustración 32. Transmisión de variables a Arduino.	57
Ilustración 33. Conexión USB Arduino a un dispositivo móvil.	59
Ilustración 34. Código switch case para la transmisión de Arduino.	59
Ilustración 35. Bytes de conexión.	60

Ilustración 36. Etiquetas de texto.....	60
Ilustración 37. Comparador buffer recepción.....	61
Ilustración 38. Clase LocationManager.....	62
Ilustración 39. Permisos para acceder a los servicios de localización.....	62
Ilustración 40. Código de obtención de mejor proveedor.....	63
Ilustración 41. Comunicación entre Android y PHP.	64
Ilustración 42. Diagrama de bloques de la programación PHP.....	66
Ilustración 43. Formula Harvesine.	67
Ilustración 44. Base de datos.....	68
Ilustración 45. Archivo JavaScript de Google.	70
Ilustración 46. Configuraciones básicas del mapa.....	70
Ilustración 47. Código variable map.....	71
Ilustración 48. Icono personalizado.	71
Ilustración 49. Código para ubicar el marcador en la posición deseada.....	71
Ilustración 50. Código para cargar el icono personalizado.	72
Ilustración 51. Código para dibujar una PolyLine.....	72
Ilustración 52. Captura de imagen Últimos datos recibidos.	72
Ilustración 53. Captura de imagen Resumen de la jornada	73
Ilustración 54. Captura de imagen Banner video.....	73
Ilustración 55. Cámara IP.	74
Ilustración 56. Router 3G Wi-Fi.....	76
Ilustración 57. Internet móvil Tigo.	77
Ilustración 58. Abrir puertos del router.....	77
Ilustración 59. Inversor de corriente 12v a 110v.	78
Ilustración 60. Captura interfaz de usuario portal web.....	78
Ilustración 61. Captura reporte de la jornada.....	79

LISTA DE TABLAS

Tabla 1. Características Arduino ADK.	30
Tabla 2. Voltaje en los diferentes niveles de combustible.....	53
Tabla 3. Especificaciones cámara IP.....	74
Tabla 4. Presupuesto.....	80

LISTA DE ANEXOS

Anexo A. Placas oficiales de Arduino:	89
Anexo B. Shields producidos por usuarios.	90
Anexo C. Operadores y funciones	90
Anexo D. Versiones de Android.....	92
Anexo E. Diseño de la placa.....	94
Anexo F. Montaje de los indicadores.....	94

GLOSARIO

API: Interfaz de Programación de Aplicaciones (Application Programming Interface). Conjunto de especificaciones e implementación de comunicación entre componentes software. Trata de un conjunto de funciones o clases y sus métodos cuyo propósito es ofrecer acceso a los servicios de un sistema.

DNS: Sistema de nombres de dominio (Domain Name System) es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignados a cada uno de los participantes. Su función es traducir nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red, con el propósito de poder localizar y direccionar estos equipos mundialmente.

IP: Es un protocolo de comunicación de datos digitales clasificado funcionalmente en la Capa de Red según el modelo internacional OSI.

ROUTER 3G: Un enrutador es un dispositivo que proporciona conectividad a nivel de red. Esta unidad funciona con un modem de tercera generación conectado a través de un puerto U.S.B. (Bus Universal en Serie) y proporciona internet ó acceso a red a cualquier dispositivo habilitado para Wifi.

TELEFONIA MOVIL 3G: Se refiere a la tercera generación de transmisión de voz y datos a través de telefonía móvil. Aunque esta tecnología estaba orientada a la telefonía móvil, desde hace algunos años las operadoras de telefonía móvil ofrecen servicios exclusivos de conexión a Internet mediante módem USB, sin necesidad de adquirir un teléfono móvil, por lo que cualquier computadora puede disponer de acceso a Internet.

WEB: Red informática mundial (World Wide Web), utilizada para denominar uno de los servicios de la red Internet. Son páginas que usan lenguaje HTML, que permite presentar en pantalla textos y gráficos en el formato deseado. Estas páginas contienen referencias ó enlaces (links) que permiten acceder a otras páginas.

RESUMEN

La finalidad del trabajo de proyecto de grado, denominado “**SISTEMA DE CAJA NEGRA PARA MONITOREO Y REGISTRO DE AUTOMOTORES TIPO BUSETA**” consiste en extraer información de sensores ubicados en el vehículo de transporte público, mediante una placa Arduino que se comunica con un dispositivo móvil, para su posterior transmisión a un servidor web.

En la ciudad de Neiva, el registro de tiempo entre las rutas se hace de forma manual, poco precisa y poco fiable, lo cual resulta complicado al momento de registrarlo en un computador, debido a la cantidad de datos obtenidos en los distintos puntos de control y adicionalmente, es costosa a largo plazo por la cantidad de operarios que se requieren. Además no existe control alguno sobre las rutas que se realizan después de determinada hora, razón por la cual se necesita una solución que sea eficaz para el registro y posterior envío de estos datos.

Debido a la deficiente calidad del servicio de transporte público, es necesario el monitoreo de diferentes clases de variables tales como velocidad, posición, aceleración, seguimiento de sus rutas, medidor de combustible, kilómetros recorridos y tiempo entre puntos de control, con el fin de que esta sea una herramienta que permita mejorar el servicio.

Se implementa una cámara para monitorear y registrar lo que pasa en la cabina del conductor, con el fin de determinar posibles causas de culpabilidad en caso de accidentes de tránsito ó irregularidades en la prestación del servicio, además de prestar alguna seguridad tanto al conductor como a los usuarios.

El primer paso del proyecto es digitalizar todas las salidas análogas de los distintos sensores, el segundo paso consiste en procesar todas estas salidas digitalizadas, transformarlas a valores contables, el tercer paso es enviar estos datos ya procesados a un celular por cable USB, el cuarto paso es transmitir los datos a un host y por último el quinto paso es visualizar estos datos a través de un portal web en tiempo real.

ABSTRACT

The purpose of the degree project work, called "BLACK BOX SYSTEM FOR MONITORING AND REGISTRATION OF MOTOR TYPE BUSETA" is to extract information from sensors placed in the public transport vehicle, using an Arduino board that communicates with a mobile device, for onward transmission to a web server.

In the Neiva city, the registration of time between the routes is done manually which is not precise and unreliable, which is difficult when recording the data into a computer, because of the large amounts of data obtained at various checkpoints and it is costly in the long run because of the number of workers required. Furthermore there is no manual control over the routes after a certain time, which is why you need a solution that is effective for recording and sending back data at all times.

Due to the poor quality of public transport service, it is necessary to monitor different types of variables such as speed, position, acceleration, track their routes, fuel gauge, kilometers traveled and time between control points in order to that this is a tool to improve service.

It implements a camera to monitor and record what happens in the driver's cab, in order to determine possible causes of fault in case of traffic accidents or irregularities in service delivery as well as providing some security for both the driver and the passengers.

The first step of the project is to digitize all the analog outputs of the various sensors, the second step is to process all outputs digitized, transform them into accounting values, the third step is to send these processed data to a cell phone via USB cable, the fourth step is to send the data to a host, and finally the fifth step is to display this data dare of a real-time web portal.

INTRODUCCIÓN

Este libro describe los procesos de diseño, programación e implementación de un prototipo, además de las principales conclusiones y beneficios que puede traer el proyecto que por sus siglas se ha denominado como **S.M.Y.R.T.** (Sistema de monitoreo y registro de transporte).

Se busca, un dispositivo que pudiera solucionar los problemas que padece el transporte urbano, en especial los automotores tipo buseta, que además de los beneficios que potencialmente podría traer a las empresas de transporte público, también, pudiéramos constituir un servicio de prevención de accidentes viales que, junto a una plataforma tecnológica y una interfaz amigable con el usuario podría prevenir en tiempo real, posibles accidentes monitoreando la velocidad del vehículo, además de videograbaciones durante los turnos de las diferentes rutas, que en caso de potenciales incidentes podrían corregir un camino para evitar futuros imprevistos.

Se piensa, no solo en desarrollar un dispositivo novedoso, sino además útil y necesario, que gracias a la utilización de software libre resulta de fácil acceso. En el siguiente trabajo, entonces se plantean las principales definiciones de los distintos equipos, tanto software como hardware, que ayudaron a implementar el dispositivo, así como los pasos que se debieron seguir para tal fin.

Fueron usadas tecnologías novedosas, como la obtención de datos a través de los sensores ya incorporados en el autobús, valiéndose del módulo Arduino Mega ADK, el cual se conecta fácilmente a cualquier Dispositivo con un Sistema Operativo Android, que permitirá, en tiempo real, enviar los datos que arroje el automotor a un portal web, además de una imagen inmediata de la cabina del conductor, que gracias a la tecnología de Cámara IP, pueden generar una comunicación Empresa-Conductor.

Se procede entonces, en el siguiente escrito, a plasmar las recomendaciones, los motivos que impulsan a crear este proyecto orientado a la seguridad vial en el perímetro urbano, además de los inconvenientes y pasos a seguir, para la implementación, programación y utilización del sistema "SMYRT".

OBJETIVO

Diseñar un sistema de registro y monitoreo de: tiempos de control, posición, velocidad, aceleración, medición de combustible y kilómetros recorridos integrados en una caja negra para su posterior transmisión vía internet a un servidor.

OBJETIVOS ESPECIFICOS

- Sensar las variables tiempos de control, posición, velocidad, aceleración, combustible en el tanque y kilómetros recorridos; mediante acelerómetro, velocímetro, módulo GPS y medidores de nivel utilizando los circuitos disponibles en el mercado.
- Implementar un sistema de acondicionamiento de variables físicas para su digitalización.
- Implementar un sistema de monitoreo de cámara ubicada en la cabina del conductor que registre lo ocurrido en el horario de trabajo.
- Crear una aplicación Android para tecnología móvil que reciba y transmita los datos a un host en tiempo real.
- Crear un portal web para poder recibir los datos: tiempos de control, posición, velocidad, aceleración y video en tiempo real.

ESTADO DEL ARTE

El problema del servicio público ha despertado especial interés en cuanto al desarrollo de nuevas tecnologías y herramientas para prestar un servicio y hacer que este pase de ser meramente útil a ser un servicio eficaz, incitando el interés de varios países que han llevado la delantera en cuanto a la vanguardia tecnológica como es el caso de Estados Unidos y España, que hoy por hoy son líderes en el control de transporte público, entendiéndose por control, el manejo del tiempo en el que se realiza una ruta de transporte, la especificación de dicha ruta y la forma en que efectivamente son transportado los pasajeros, es decir las aptitudes de los empleados de estas empresas al manejar dichos vehículos¹.

Para iniciar tomaremos como referentes los medios utilizados por Estados Unidos, cuyo sistema de transporte en términos de control presenta las siguientes características:

- Sistema guiado por G.P.S. (Sistema de posicionamiento global) para evitar desvíos de ruta, el vehículo da aviso de la distancia que puede recorrer con el combustible disponible, los limpia parabrisas se activan automáticamente en caso de lluvia, incluso el conductor es avisado de situaciones de tráfico denso ó posibles accidentes en la vía.

Todos estos avances son parte de lo que se conoce como sistemas inteligentes de transporte (ITS, siglas en inglés de Inteligente Transportación Sistemas).

Los ITS son sistemas que combinan la informática, la robótica y la comunicación aplicadas al transporte. La base de todos ellos es la recopilada de datos a través de diferentes dispositivos que posteriormente son procesados informáticamente para elaborar una información útil y comprensible para el ser humano. Esto permite que los elementos de un sistema de transporte equipados con estas tecnologías interactúen entre sí de forma inalámbrica, con el objetivo de aumentar la seguridad del pasajero, mejorar la eficiencia del transporte y reducir la contaminación medio ambiental.

¹ Instituto español de Comercio Exterior (ICEX), Sistemas Inteligentes de Transporte en EEUU, 18 de octubre 2011. Extraído de: www.navactiva.com

Así mismo, en España, se encuentra ubicada INDRA², la empresa líder en reformas tecnológicas en el campo del transporte que ha propuesto en Madrid y el cual está soportado en las siguientes características:

- Los autobuses cuentan con sistemas inteligentes que disponen de billetes sin contacto, acceso al momento de las rutas en tiempo real, diversos sensores que dan razón del estado de bus, posibilidad de llevar bicicletas, diseños interiores más amplios y mejores accesos para personas discapacitadas, la cual presta una muy buena calidad del servicio.

Así mismo Indra Sistemas en medio de su proceso de expansión, el pasado mes de agosto firmó un contrato con Medellín³, para instalar una red más eficiente y moderna basada en las siguientes características:

- **GESTIÓN EFICIENTE Y EN TIEMPO REAL**

El Sistema de Ayuda a la Explotación (SAE) de Indra está basada en tecnologías de localización (GPS), sistema de información geográfica (GIS) y comunicaciones móviles, facilita al centro de control información en tiempo real sobre los vehículos, así mejora la gestión y agiliza la toma de decisiones ante cualquier incidente.

La solución incluye en los autobuses un sistema de conteo de pasajeros basado en un circuito cerrado de televisión (CCTV), que registra el número de pasajeros tanto en subida como en bajada y hace posible conocer la ocupación para evitar sobrepasar la capacidad máxima permitida, solicitar vehículos de refuerzo en picos de demanda o realizar estudios de movilidad en este medio de transporte.

Todas estas tecnologías redundan en una mejora de la calidad del servicio público y de la información ofrecida a los usuarios, así como en un mayor conocimiento del servicio y de la demanda por parte de Metro de Medellín.

A nivel institucional, se han propuesto en años anteriores proyectos con características y propósitos similares, los que aquí contemplamos, los cuales incluyen implementaciones en el servicio de transporte consistente en las siguientes características:

- Una interface de usuario conectada a un equipo transreceptor, el cual se instala en la cabina de cada vehículo.
- Un módulo transmisor (control) que le proporcione la ubicación temporal al vehículo; esta ubicación es transmitida desde el vehículo hacia la central cuando esta lo indique.

² Negocio tecnológico. Nuevos autobuses europeos para un transporte público eficiente. 15 de noviembre 2012. Extraído de <http://www.negociotecnologico.com>

³ INDRA. INDRA implanta su tecnología inteligente para la gestión del tráfico y el transporte público de Medellín por 9 m€. (14 de agosto 2012). Extraído de <http://www.indracompany.com>

- “Un equipo transreceptor que se encarga de hacer el monitoreo de los vehículos y a su vez enviar los datos adquiridos al computador.”⁴
- “Un sistema de seguimiento y rastreo de un vehículo del servicio público conformado por un hardware y un software que permiten llevar la información desde el receptor GPS hasta un computador, a través de una página W.A.P. (protocolo de aplicaciones inalámbricas) usando la tecnología G.P.R.S. (Servicio general de paquetes vía radio), mediante el protocolo N.M.E.A. (Asociación nacional de electrónica marina) y utilizando comunicación serial”⁵

⁴ Pastrana Jhon, Polanco Camilo y otros. diseño e implementación de un sistema de monitoreo por radiofrecuencia para el registro de horarios de buses de transporte. Neiva 2005. Tesis de grado. Universidad Surcolombiana. Facultad de ingeniería.

⁵ Morales Leidy y Trujillo Omar. MONITOREO DE VEHICULOS DE TRANSPORTE DE SERVICIO PÚBLICO. (2011).Neiva 2005. Tesis de grado. Universidad Surcolombiana, Facultad de Ingeniería.

JUSTIFICACIÓN

Se tiene un sistema de transporte público, en el que los buses se quedan estáticos a pesar de que la luz del semáforo este en verde; no respetan las señales de tránsito, golpean vehículos a diario y ponen en riesgo los usuarios del servicio y peatones. Este problema, está asociado a una negligencia de los conductores y no existe un registro físico que pueda determinar la culpabilidad en un accidente de tránsito.

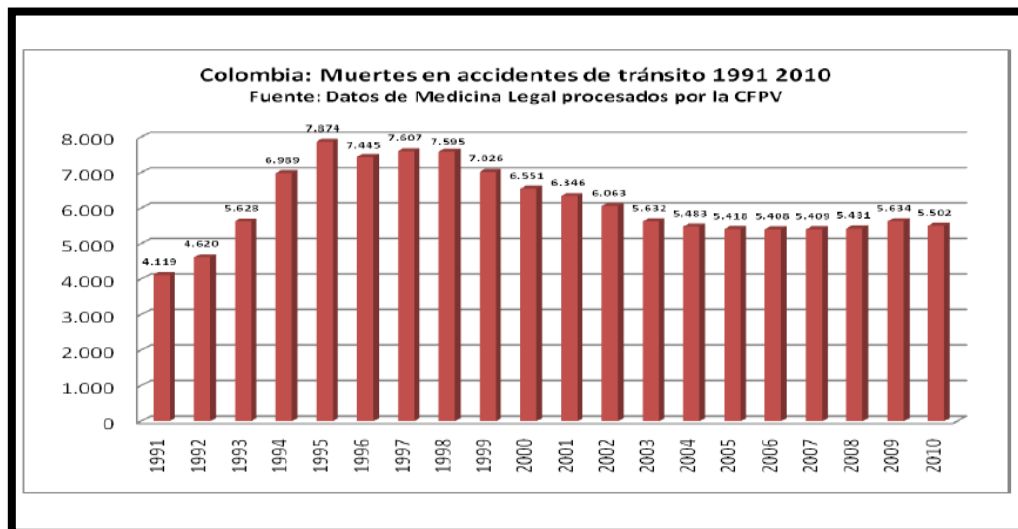
Al no haber un control del seguimiento de las rutas, los autobuses toman diferentes vías que no son permitidas por la ley, también el modo de manejar de algunos conductores no es el más apropiado porque exceden los límites de velocidad permitidos dentro de la ciudad. A pesar de los controles impuestos por las diferentes empresas de transporte público, estos no son suficientes.

El vigente método de registro de puntos viales de los diferentes automotores de servicio público se hace manualmente por medio de operadores, lo que produce una baja fiabilidad de la información. Esta técnica pudo haber funcionado para el momento en el que existían pocos automotores de servicio público, pero con el aumento de la población se hizo necesario incrementar la cantidad de vehículos y mejorar la calidad del servicio, esto acarrea un aumento en la cantidad de la información que se debe registrar y sistematizar.

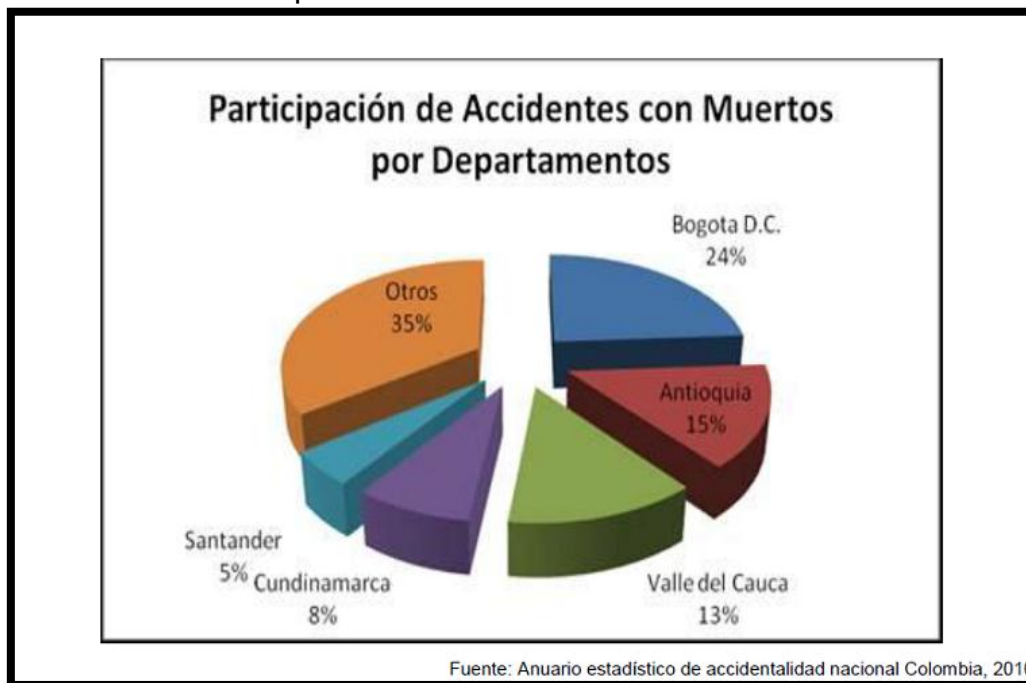
El documento, no solo aporta un sistema totalmente sofisticado y en alza tecnológica como los sistemas de otros países a nivel mundial, tal es el caso de España y los otros casos expuestos en el apartado de estado de arte, si no que enfocamos este mismo a lograr una mayor seguridad vial. Todas las grandes tecnologías a lo largo de la historia no solo buscan comodidad y simplicidad a la hora de afrontar alguna tarea sino que también busca proteger la vida humana en todas sus formas, la actividad de conducción, es considerada como de alto riesgo, debido a que en el desarrollo de esta, se generan innumerables accidentes que involucran a millares de vidas humanas todos los días, para demostrar tal afirmación citaremos a continuación las estadísticas arrojadas por el estudio del Plan Nacional de Seguridad Vial en Colombia 2011 a 2016⁶, en lo referente a las pérdidas humanas generadas por la conducción de Vehículos de transporte público:

- En Colombia, en general, por cada 100 mil habitantes el 12.3% mueren a causas de accidentes de tránsito, En el 2010, esta estadística llegó a su máximo, representando el 22% de las muertes en el país.

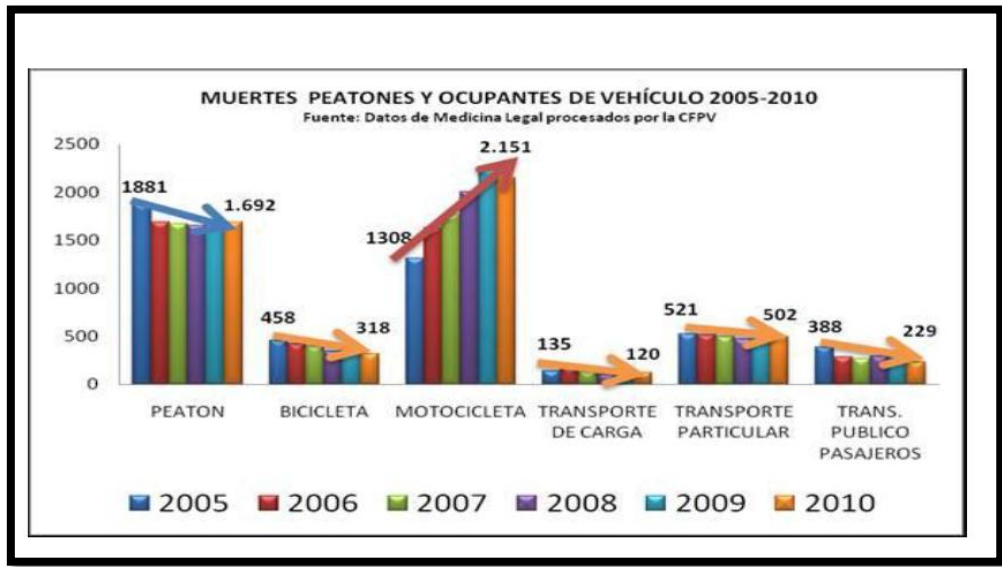
⁶ Ministerio de Transporte. Plan de Seguridad Vial 2011-2016. Bogotá D.C. Extraído de: <https://www.mintransporte.gov.co/descargar.php?idFile=8>



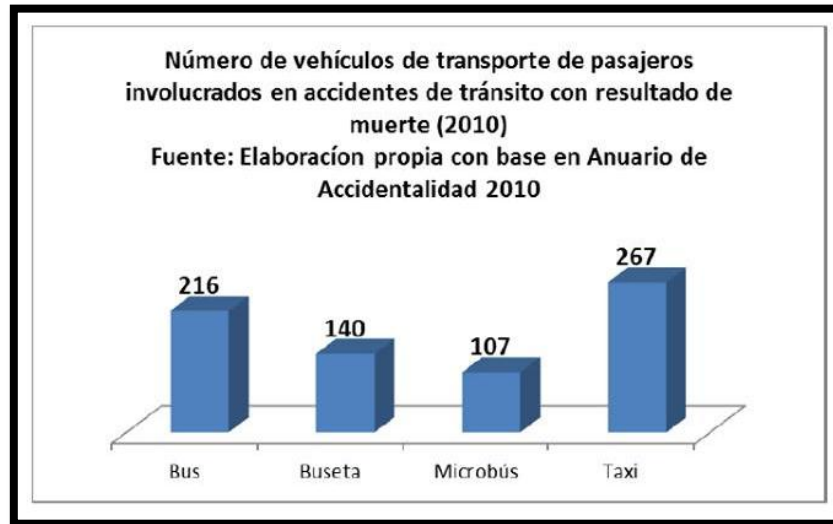
- En nuestro departamento, considerado dentro del ámbito de “otros”, los accidentes terrestres provocaron el 35% de muertes violentas.



- A su vez, en términos de vidas cuantificables, y aunque en menor escala, vehículos como motocicletas, se contó con 338 muertes para el 2010 a manos del servicio público.



- De las anteriores 356 muertes fueron a causa de transporte público a cargo de vehículos tipo buseta. generando el 9% de las muertes en accidentes de tránsito en el país, de las cuales el 79% de éstas suceden en un entorno urbano, como es el caso de nuestra ciudad.



Para mitigar tal mortalidad, en el plan citado a cargo del ministerio de transporte se formularon algunas posibles soluciones, que involucran en su mayoría la educación vial de los conductores, tales medidas abarcan temas como “estudios de reglamentación para la instalación y utilización del cinturón de seguridad en los asientos de los buses intermunicipales”⁷, hasta lo que es de nuestro especial interés “La expedición de la normativa que regula la instalación de GPS o

⁷ Pag. 57. Plan Nacional de seguridad vial. 2010-2014. Ministerio de transporte.

localizadores de flota en los camiones y buses intermunicipales”⁸, dichas estrategias se cumplieron, pero como bien lo dice el estudio están enfocadas para el transporte intermunicipal, dejando a un lado, las zonas urbanas, donde como ya lo indicamos, se presenta la mayor tasa de mortalidad dentro del ámbito de servicio público, allí, en buses urbanos, es a donde va dirigido nuestro proyecto.

Con el proyecto de caja negra, se busca la implementación de un servicio público más seguro, gracias a las herramientas que éstas brindarán, tales como la regulación de rutas mediante un sistema GPS, además de un acelerómetro que previene y advierte a los operarios de una colisión lo que activará una respuesta más temprana ante eventuales choques, además de la regulación de la velocidad del automotor, ya que al dejarlo de todo en manos de la educación vial de los conductores, existe un margen de error humano, con los constantes reportes en el portal Web que dispone para el efecto, esta información podría utilizarse para formular reprensiones directas con evidencias concretas a los conductores, y así lograr una “educación” vial más regulada, dándole una seguridad más mediata a los pasajeros, y un motivo de tranquilidad a las empresas de transporte que eventualmente podrán monitorear a sus empleados mediante un sistema de videograbación que podría utilizarse para posteriores estudios para la prevención de futuros accidentes.

El proyecto es orientado a evitar que más accidentes se presenten y con la disminución de ellos, procurar que los medios de transportes urbanos terrestres tengan una mayor confiabilidad para los usuarios. Con la ola tecnológica que busca modernizar el servicio de transporte, este sistema puede hacerse estándar a nivel nacional, logrando que todos los automotores de servicio público adopten una caja negra que les dará solución a los diferentes problemas ya existentes.

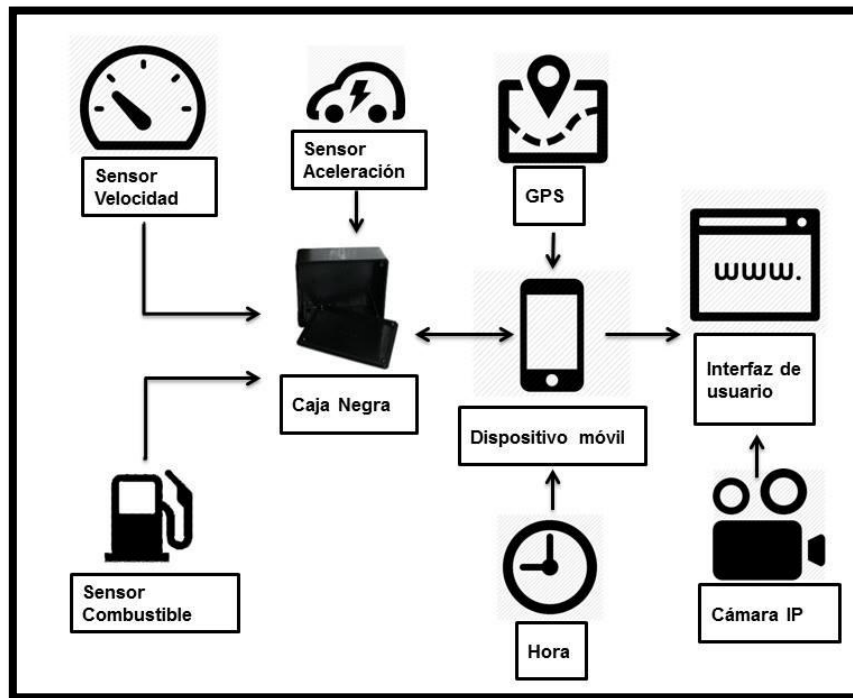
Las diferentes empresas de servicio público que optarán por este sistema tendrán entonces, un mayor control de sus vehículos en las rutas, así como en el tiempo entre puntos viales, el combustible consumido y su habilidad de conducción, por ende prestarían un servicio de mayor eficacia, seguridad y rentabilidad tanto para el usuario como para la empresas de transporte.

⁸ Pag. 63. Plan Nacional de seguridad vial. 2010-2014. Ministerio de transporte.

1 DIAGRAMA GENERAL DEL SISTEMA

A continuación se muestra el diagrama del sistema completo y la función que cumple cada componente.

Ilustración 1. Diagrama general del sistema.



1.1 Sensores

- Acelerómetro

Es un dispositivo que permitirá medir el movimiento y las vibraciones a las que está sometido el vehículo.

- GPS

El Sistema de Posicionamiento Global GPS recibe la información de la posición a través de al menos, tres satélites calculando su propia ubicación en la Tierra mediante la triangulación.

- Indicador de velocidad

El velocímetro electrónico funciona en base a un sensor de campo magnético, que al rotar produce una tensión que varía proporcionalmente en amplitud y en frecuencia con la velocidad del vehículo.

- Indicador de combustible

Es un indicador que será usado para medir la cantidad de combustible en el tanque del automotor.

1.2 Caja negra

Conteniente el circuito convertidor de señal, el circuito de sensor de vibración y la tarjeta Arduino.

1.3 Dispositivo móvil

Es un dispositivo inalámbrico para acceder a los servicios de la red de telefonía celular. Utilizado para la transmisión de tiempos de control, posición, velocidad, nivel de combustible y hora desde el vehículo hacia una base de datos.

1.4 Entorno al usuario

- Base de datos

Es un conjunto de datos (tiempos de control, posición, velocidad, fecha, identificación del bus, ruta, kilómetros recorridos, combustible y alerta de choque) almacenados constantemente en una tabla.

- Portal web

Contiene los datos más importantes, organizados en dos tablas (Últimos datos recibidos y resumen de la jornada), un mapa donde se visualiza la ubicación actual del vehículo y tres botones, uno de ellos para exportar los datos en formato PDF, otro para vaciar la base de datos cuando se acabe la jornada laboral y el ultimo para acceder a la cámara IP del vehículo.

1.5 Cámara IP

Registra lo ocurrido dentro del vehículo en el horario de trabajo; además de tener una comunicación bilateral entre conductor y operario.

2 ARDUINO

El monitoreo de los autobuses no solo se limita a saber su ubicación respecto a una ruta predeterminada en tiempo real, va más allá dando a conocer información importante para el operador de rutas sobre el estado del automotor en un tiempo determinado.

Para esto dispondremos de los sensores, marcadores e indicadores del automotor usando así la información proporcionada por ellos, la cual enviaremos a una placa Arduino que procesara todas estas variables, Para ello y a efectos de la investigación entraremos a definir “Arduino”.

2.1 ¿Qué es Arduino?

Arduino es una plataforma de hardware libre (Open Source Hardware) se basa en una placa con un microcontrolador y un entorno de desarrollo.

El microcontrolador usado en la placa es un Atmel AVR y varios puertos entrada/salida, tanto análogos como digitales, así como salidas PWM (Modulación por ancho de pulso) y de comunicación, los cuales controlan objetos físicos.

Algunos de los microcontroladores más usados son el ATmega 168 y el ATmega 328 para las placas básicas, el ATmega1280 para la de mayor capacidad y el ATmega8 para las placas más antiguas.

El microcontrolador de la placa Arduino se puede programar mediante el lenguaje de programación básico de Arduino y el entorno de desarrollo Arduino.

El hardware de fuente libre (open-source) comparte muchos de los principios del código libre tales como:

- Ofrecer una licencia que permita producir derivados y modificaciones, además de su re-distribución bajo la licencia original.
- Publicar la documentación necesaria que permita la modificación y distribución, incluyendo los archivos de los diseños mismos.
- La licencia no debe discriminar a ningún grupo o persona.
- La licencia no debe restringir a ningún campo o actividad el uso de la obra.
- La licencia no debe restringir otro hardware o software, es decir que no puede insistir en que otros componentes de hardware o software externos a los dispositivos sean también open-source.
- La licencia tiene que ser neutral.

Como se ve, los diseños y esquemas de Arduino se distribuyen bajo la licencia “Creative Commons Attribution-Share A like 2.5”.⁹

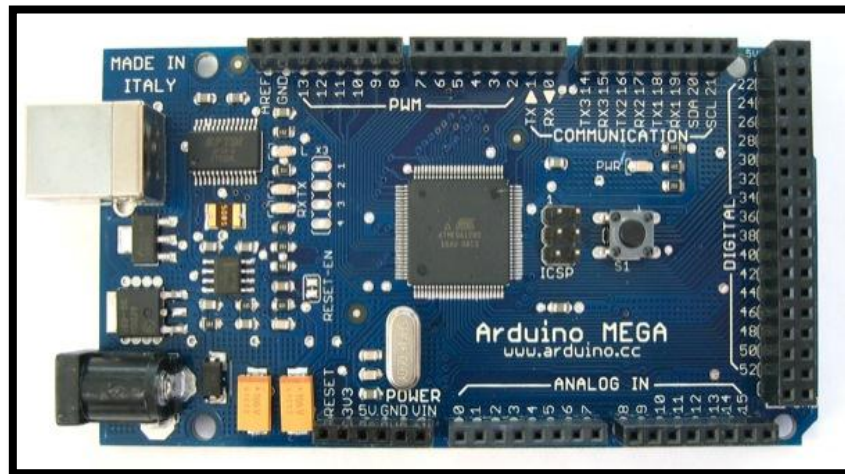
Al seguir estas licencias donde la información es pública, todas las placas pueden ser construidas por uno mismo o bien comprarlas ya montadas. El software puede ser descargado de forma gratuita, al igual que los ficheros de referencia pueden ser adaptados a las necesidades particulares.

Las razones por las cuales en la elaboración de este proyecto fue usada una placa Arduino fue para simplificar procesos además de brindar ventajas tales como:

- Entorno de programación: El entorno de programación de Arduino es fácil de entender además de ser flexible para usuarios tanto principiantes como avanzados.
- Varias plataformas - El software de Arduino puede ser instalados en varios sistemas operativos Windows, Linux y Macintosh OSX. En cambio la mayoría de los microcontroladores ya están acotados al uso de Windows solamente.
- Código abierto - El software de Arduino está publicado bajo una licencia libre y está preparado para ser ampliado por programadores experimentados. Está basado en librerías de C++ lo cual permite que se pueda añadir directamente código en C en los programas.

Existe una gran variedad de placas Arduino y a todas ellas se les han hecho varias revisiones.

Ilustración 2. Arduino Mega.



Fuente: <http://arduino.cc/en/Main/ArduinoBoardMega2560>

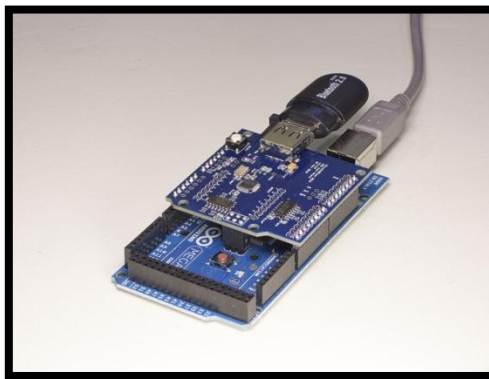
⁹ Creative Commons CC, Attribution-ShareAlike. 2012. Página Web. (Fecha de consulta 03 de Julio de 2013)Fuente: <http://creativecommons.org>

Las placas oficiales: **(Ver Anexo A)**.

El objetivo principal de Arduino es ofrecer al consumidor la posibilidad de seguir la filosofía “*hágalo usted mismo*”.

Otra parte importante de las placas Arduino, son los “*Shields*”, estos son placas que se colocan sobre un Arduino y que amplían una nueva función para que sea controlada desde Arduino, ya sea para controlar diferentes aparatos o adquirir datos entre otras funciones.

Ilustración 3. Arduino con un “Shield”.



Fuente: http://www.circuitsathome.com/wp/wp-content/uploads/2011/02/UHS2_mega2560_bt.jpg

Existen además de éstos, una infinidad de Shields producidos por usuarios. **(Ver Anexo B)**.

Arduino al ser OSHW puede ser montada por uno mismo. La placa Arduino mono capa o Single-Sided Serial ha sido diseñada para que sea fácil de montar ajustándose a las necesidades del usuario.

Ilustración 4. Arduino Severino.



Fuente: http://farm6.staticflickr.com/5300/5472604986_cd65fc0dd8_z.jpg

Se puede encontrar documentación, incluyendo manuales y componentes en la web oficial.¹⁰

Para adquirir una placa Arduino o Shield se puede contactar a los distribuidores oficiales a través de distintas tiendas online de electrónica.

En la página oficial se encontrará una variada lista de distribuidores en distintos países.

Después de evaluar la mayoría de tiendas online, por precio, gastos de envío y disponibilidad, se recomienda:

- Tienda De Robótica <http://tienda.tdrobotica.co/>¹¹

La elección que se hizo fue la de la placa Arduino Mega ADK.

El principal objetivo es el de lograr una comunicación USB entre la tarjeta de adquisición de datos y un móvil, que a su vez estará comunicado con una base de datos alojada en un servidor local. Para poder desarrollar cabalmente este objetivo, es necesario un módulo Arduino que posea una conexión vía USB, además de un reconocimiento con el Sistema Operativo Android 2.3.

Existen dos posibilidades:

- Adquirir la placa Arduino UNO y adaptar un Shield USB para la comunicación con el dispositivo móvil, pero lamentablemente esto acarrearía algunos inconvenientes con la compatibilidad entre el modulo y el sistema operativo Android.
- Adquirir la Placa Arduino Mega ADK para el manejo de aplicaciones Android, la cual no representa ningún problema con la conexión requerida, sin embargo esta posibilidad impide usar al máximo el potencial de la placa.

Después de un detallado análisis se tomó la decisión de comprar la placa Arduino Mega ADK, la cual se ajusta perfectamente a las necesidades de diseño.

¹⁰ Arduino Team, Arduino Homepage. 2013. Página Web. (Fecha de consulta 01 de julio de 2013).
DISPONIBLE EN: <http://www.arduino.cc/>

¹¹ Tienda De Robótica, Productos.2013. Página Web. (Fecha de consulta 03 de julio de 2013). Disponible en:
<http://tienda.tdrobotica.co/>

Ilustración 5. Arduino Mega ADK.



Fuente: <http://arduino.cc/en/Main/ArduinoBoardADK>

2.2 Características Arduino Mega ADK

Se puede encontrar una descripción completa de la placa Arduino Mega ADK en la página web oficial¹². A continuación se da un resumen de las características principales:

Tabla 1. Características Arduino ADK.

Microcontrolador	ATmega2560
Voltaje de operación	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limite)	6-20V
Pines I/O digitales	54 (de los cuales 15 proveen salida PWM)
Pines I análogos	16
Intensidad por Pin I/O	40mA
Intensidad 3.3v en pin	50mA
Memoria Flash	256 KB de los cuales 8 son usados para el gestor de arranque
SRAM	8 KB
EEPROM	4 KB
Velocidad del reloj	16MHz

La placa Arduino Mega ADK puede ser alimentada a través de una conexión USB o con una fuente de alimentación externa. Esta fuente puede ser un adaptador AC

¹² <http://www.arduino.cc/>

DC o una batería. El adaptador se puede conectar a un enchufe de 2,1mm de centro positivo en el conector de alimentación de la placa.

La tarjeta puede funcionar con un suministro externo de 5,5v a 16v voltios. Si se proporciona menos de 7v, el pin de 5v puede que suministre menos de cinco voltios y puede ser inestable. Si se utiliza más de 12v, el regulador de voltaje puede sobrecalentarse y dañar la placa.

Los pines de alimentación son:

VIN. El voltaje de entrada a la placa Arduino cuando se trata de utilizar una fuente de alimentación externa (en lugar de una conexión USB).

5V. Este pin es usado como una salida regulada de 5v.

3V. Este pin suministra 3.3 voltios generados por el regulador de la placa. La intensidad máxima es de 50 mA.

GND. Tierra.

IOREF. Este pin proporciona la referencia de tensión con la que opera el microcontrolador.

En cuanto a la memoria, el ADK tiene 256 Kb de memoria flash para el almacenamiento de código, 8 Kb de SRAM y 4 KB de EEPROM.

Cada uno de los 50 pines digitales de la placa Arduino Mega ADK puede ser utilizado como una entrada o salida usando diferentes funciones en la programación. Todos estos pines funcionan a 5 voltios. Cada pin puede recibir un máximo de 40 mA y tiene una resistencia interna de 20 K Ω a 50 K Ω . Además posee algunos pines que tienen funciones especializadas.

- Serie: 0 (RX) y 1 (TX); Serie 1: 19 (RX) y 18 (TX); serie 2: 17 (RX) y 16 (TX); serie 3: 15 (RX) y 14 (TX).
- Interrupciones externas: 2(interrupción 0), 3 (alarma 1), 18 (alarma 5), 19 (alarma de 4), 20 (interrupción 3) y 21 (alarma 2). Estos pines pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio de valor.
- PWM: 2 a 13 y 44 a 46 son pines que proporcionan PWM de 8 bits.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Estos pines admiten la comunicación SPI utilizando la biblioteca de SPI.
- USB HOST: MAX3421, comunica al MAX 3421 con la placa Arduino mediante el bus SPI.
- LED: 13 hay un LED conectado al pin digital 13. Cuando el pin tiene un valor en alto, el LED estará encendido y cuando el pin está en un valor bajo, el LED estará apagado.

- Restablecer. Cuando se lleva a Low (bajo) el microcontrolador se reiniciará.

La placa Arduino ADK tiene un número de instalaciones para la comunicación con un computador, además de otras placas Arduino u otros microcontroladores. El Atmega2560 proporciona cuatro UART hardware para la comunicación en serie TTL. A través de USB proporciona un puerto COM virtual con el software del equipo.

La interfaz de host USB dada por el MAX3421 permite que la placa Arduino ADK interactúe con cualquier tipo de dispositivo que tenga un puerto USB.¹³

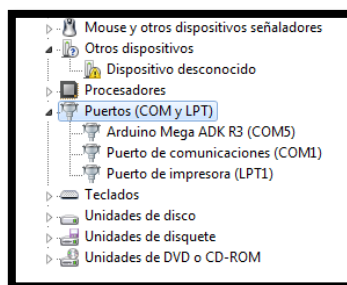
2.3 Software Arduino

Para empezar a trabajar en el entorno de Arduino, se debe descargar el software Arduino de la página oficial.¹⁴ Este paquete está contenido en un fichero comprimido. No es necesario instalarlo, lo que se debe hacer es extraerlo en la carpeta deseada.

Se instalan los controladores (Drivers) de la placa Arduino, esto se logra conectando el Arduino al computador. Los drivers se encuentran en la subcarpeta “drivers” del paquete que se extrajo con anterioridad. Luego debemos indicarle a Windows que busque el software en esa carpeta.

Dependiendo de la versión de Windows usada y la clase de placa Arduino que se tenga, este proceso puede variar, en este caso en específico, se accede al “administrador de dispositivos” y aparecerá la siguiente pantalla:

Ilustración 6. Ventana de administrador de dispositivos.



Se selecciona el dispositivo desconocido, el Arduino ADK. Se da click derecho y se actualiza el controlador indicando la ruta donde se encuentra la subcarpeta “drivers”.

¹³. Arduino Team, Arduino ADK.2013. Página Web. (Fecha de consulta 04 de mayo de 2013). Disponible en: <http://www.arduino.cc/>

¹⁴ Arduino Team, Download the Arduino software.2013. Página Web. (Fecha de consulta, 22 de julio 2013). Disponible en: www.arduino.cc/.

Una vez finalizado este proceso, el dispositivo debe ser visto en el puerto de comunicaciones, en este caso el COM5 que de ahora en adelante será el Arduino comportándose como un puerto serie en el computador.

Ahora nos dirigimos a la carpeta raíz donde se había extraído el software de Arduino y ejecutamos el archivo “arduino.exe” y se abrirá el entorno de trabajo, el IDE (“Integrated Development Environment”).

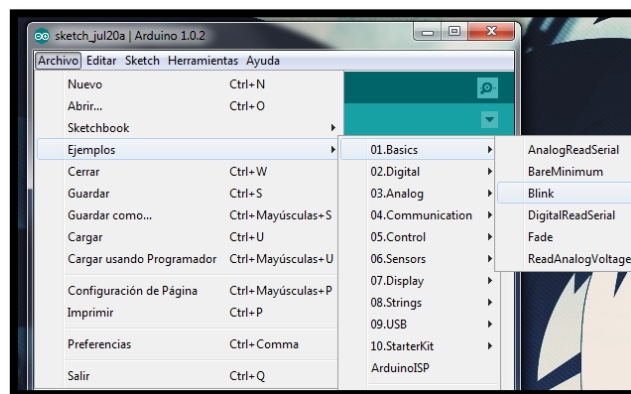
Ilustración 7. Entorno de trabajo Arduino.



Después de conectar la placa Arduino en el computador, instalar los drivers y ejecutar el IDE de Arduino, procedemos a la verificación de que la placa funciona perfectamente con el mensaje “Arduino Mega 2560 or Mega ADK on COM33” en la parte inferior derecha.

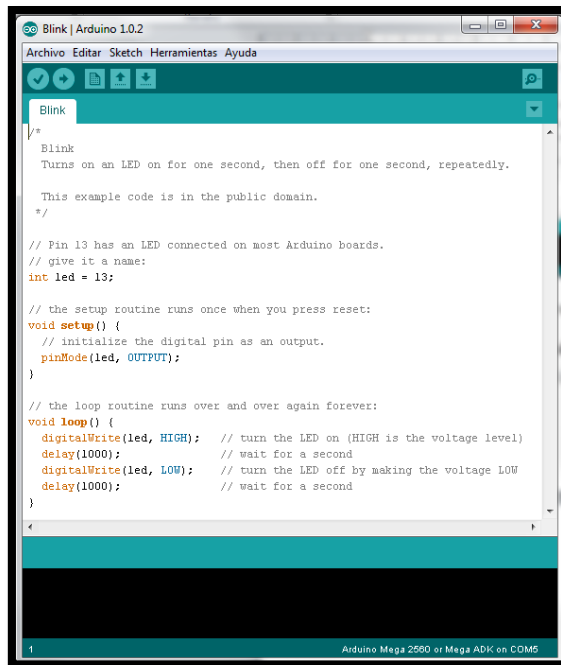
En los menús de aplicación, se selecciona “file”, submenú “examples” y dentro de “1.Basics” se elige el boceto (“sketch”) “Blink”.

Ilustración 8. Ejemplos IDE Arduino.



En una ventana se podrá visualizar el código de este ejemplo, el objetivo es encender y apagar el LED test (Arduino), cada segundo en una secuencia infinita.

Ilustración 9. Código ejemplo Arduino.



```
Blink | Arduino 1.0.2
Archivo Editar Sketch Herramientas Ayuda
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

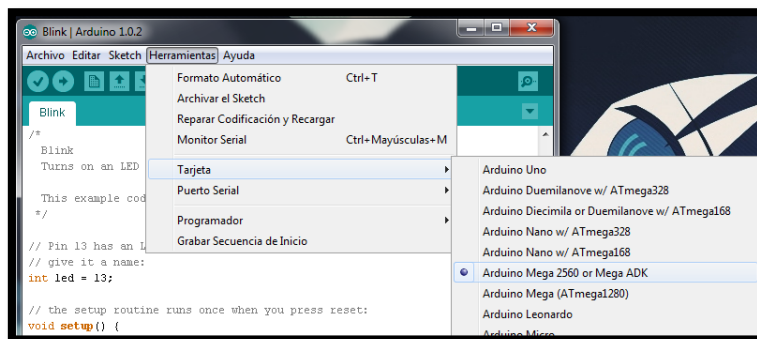
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
1 Arduino Mega 2560 or Mega ADK on COM5
```

Luego un click en el icono “verify”, el cual iniciara la verificación y compilación del código. En la parte inferior de la ventana se visualiza el resultado de la verificación donde se podrá apreciar el estado del proceso, además del tamaño de memoria usada y disponible en la placa Arduino.

Luego de este importante paso, se carga el código en la placa Arduino. Para este paso es necesario escoger la placa que se va a utilizar.

Ilustración 10. Selección de placa.



Seguido, se pulsa el icono “upload”. Si se observa en la placa, el LED empieza a prenderse y apagarse cada segundo como lo estipula el código.

2.3.1 Lenguaje de programación Arduino

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel Processing¹⁵, este a su vez está basado en Java.

El compilador usado por esta plataforma es el WinAVR¹⁶. Este compilador está configurado para el lenguaje C y C++.

Además Arduino tiene varias funciones que ya están implementadas, de este modo es más fácil realizar tareas comunes como leer un puerto digital o escribir en él, entre otras más.

2.3.2 Operadores

Todo código de Arduino debe contener dos funciones básicas Setup () y Loop ().

La función Setup () se ejecuta cuando inicia el programa. Se emplea para inicializar variables, establecer entradas y salidas, además de llamar las librerías, etc. Esta función solo se ejecuta una sola vez.

La función Loop () se ejecuta de manera consecutiva y sin interrupciones. Cuando se llega al final de esta función, se vuelve a ejecutar desde el principio hasta que se reinicie la placa Arduino. El código para controlar la placa Arduino es insertado dentro de esta función.

Al estar basado en Processing, Java y C, las estructuras de control son muy similares, para ver más de ejemplos de operadores (**Anexo C**).

Una parte importante de la programación en el IDE de Arduino son las librerías. Estas librerías suelen entregarse en un fichero comprimido que contiene una carpeta con dos archivos.

Muchas de las librerías no vienen junto con Arduino, así que se instalan por separado. Para lograr esto, se debe copiar el contenido del fichero comprimido en la carpeta “libraries” que se encuentra en la carpeta sketchbook de Arduino.

La mayoría de las librerías se pueden encontrar en una lista de la wiki de Arduino.¹⁷

Sensores:

¹⁵ Processing programming language, Homepage. Página Web. (Fecha de consulta: 06 de mayo de 2013). Disponible en: <http://www.processing.org/>

¹⁶ Source Forge, WinAVR. 2010. En línea. (Fecha de consulta, 19 de mayo de 2013). Extraído de: <http://www.winavr.sourceforge.net/>

¹⁷ Arduino c.c. Arduino playground, Libraries for Arduino. 2011. (Fecha de consulta, 03 de junio de 2013). Extraído de: <http://www.arduino.cc/>

- Capacitive Sensing: Implementa un sensor táctil capacitivo uniendo un pin de escritura con una resistencia del rango de los 10 MΩ, una lámina de metal y un condensador de 100pF a un pin de lectura, detectando cambios en este al tocar la lámina con el dedo.
- Debounce: Para una lectura de entradas digitales conectadas a botones.

Displays y LED:

- GLCD: Rutinas gráficas para LCD basados en chipset KS0108.
- LedControl: Controla las matrices de LED's o displays de siete segmentos.
- LedDisplay: Control para marquesina de LED HCMS-29xx.
- Sprite: Manipulación de sprites para usar en animaciones con LED's.

Medición de tiempo:

- Metro: Cronometra acciones.
- MsTimer2: Utiliza la interrupción de un temporizador para disparar acciones cada x milisegundos.

Muchos de los desarrolladores de Shields crean sus propias librerías para utilizar sus respectivas Shields, de modo que la cantidad de librerías es enorme.

3 ANDROID

Android es un sistema operativo basado en Linux con la interfaz de programación Java. Android SDK (Software Development Kit) provee todas las herramientas necesarias para desarrollar aplicaciones Android. El kit incluye un compilador, un emulador de dispositivo (en caso de no tener alguno) que sirve para correr todos los programas Android.

Este sistema operativo fue elegido por su versatilidad con los dispositivos móviles, además de ser gratuito y novedoso presenta una ventaja considerable frente a otras propuestas de software.

Android se caracteriza por que todas sus aplicaciones se programan en lenguaje Java y son ejecutadas en una máquina virtual diseñada para esta plataforma, además cubre áreas como el desarrollo de aplicaciones basadas en medios de comunicación, sensores y conectividad.

A cualquier desarrollador que desee programar en esta plataforma, se le proporcionará gratuitamente el entorno de desarrollo SDK. Además se agrega un emulador donde se podrán hacer las pruebas necesarias para luego ser exportadas a un dispositivo móvil.

La finalidad que tiene Android es la de aprovechar cualquier funcionalidad que se dé desde el sistema operativo y sacar de sí el mayor provecho de la plataforma. Cabe destacar que Android ofrece muchas posibilidades para integrar todo tipo de aplicaciones con la Web de Google.

3.1 Arquitectura

Android cuenta con una arquitectura distribuida en cinco diferentes capas:

Ilustración 11. Capa de aplicaciones.



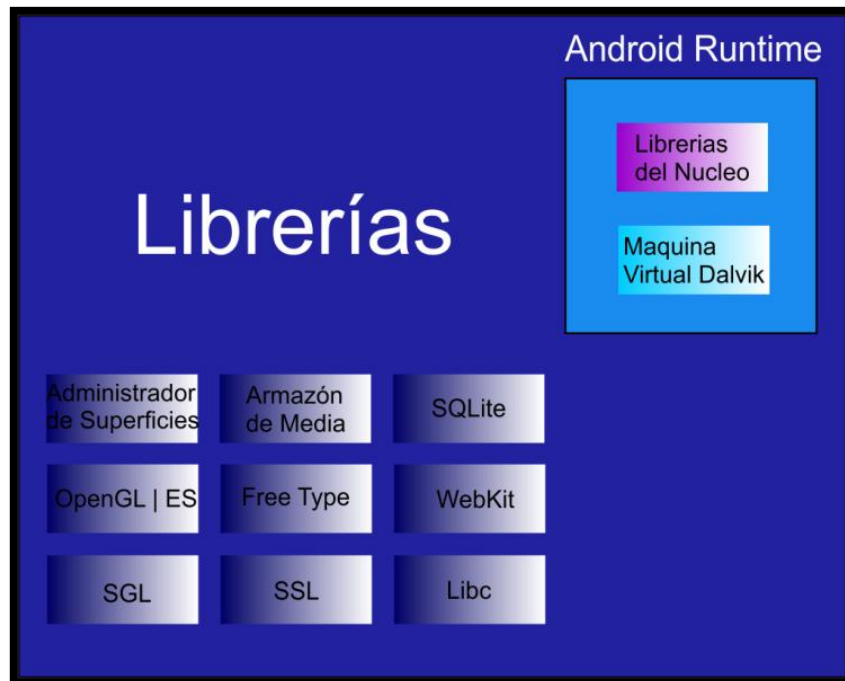
- 1) La capa de aplicaciones es la capa más alta en la arquitectura de Android, sencillamente se refiere a todas las aplicaciones con las que el usuario interactúa sin conocer las acciones que hay por debajo.

Ilustración 12. Capa de Frameworks.



- 2) La capa de Framework de Aplicaciones es donde se proporcionan todas las clases necesarias para poder crear las aplicaciones de Android. Permite a los desarrolladores tener acceso al conjunto de API's utilizadas en las aplicaciones principales cuyo objetivo es el de poder reutilizar sus componentes.

Ilustración 13. Capa de librerías y capa Runtime.



- 3) En la capa de librerías se incluye un conjunto de bibliotecas escritas en C++/C usado por diversos elementos del sistema. Dentro de las librerías más importantes están:
 - Libc: Incluye las cabeceras y funciones según el estándar de C.
 - SSL: Es la encargada de integrar un navegador para internet y proporcionar la seguridad cuando se accede a esta.
 - FreeType: Es la encargada de generar una imagen a partir de fuentes, para mapas de bits y vectores.
 - OpenGL/ES: Es la encargada de manejar los gráficos en 3D en el caso de que el hardware del dispositivo móvil lo disponga
 - SLG: Es la encargada de proporcionar el motor gráfico de 2D
 - Administrador de Superficies: Es la encargada de gestionar todas aquellas pantallas de las diferentes aplicaciones activas en ese momento.
 - Media: Es la encargada de reproducir archivos de audio y video, además de proporcionar los códec necesarios para poder reproducir los formatos más usados.

- 4) En la capa Android Runtime se incluye un conjunto de bibliotecas que proporcionan la mayor parte de la funcionalidad disponible en las bibliotecas del núcleo del lenguaje de programación Java.

Ilustración 14. Capa Kernel de Linux.



- 5) Android se basa en la versión 2.6 de Linux para el sistema de servicios, como la seguridad, la gestión de procesos, la pila de red, la gestión de memoria y el modelo del controlador. También incluye diversos drivers lo cual la hace igual a cualquier otro kernel que corre en un computador.

3.2 Versiones

Android inicio en el año 2007, y desde entonces han existido muchas versiones, pero a partir de la versión 1.5 cada una recibió un nombre específico además de que cada versión cuenta con una notoria mejora.

Para ver las características más sobresaliente de cada versión (**Ver Anexo D**).

3.3 Herramientas para el desarrollo

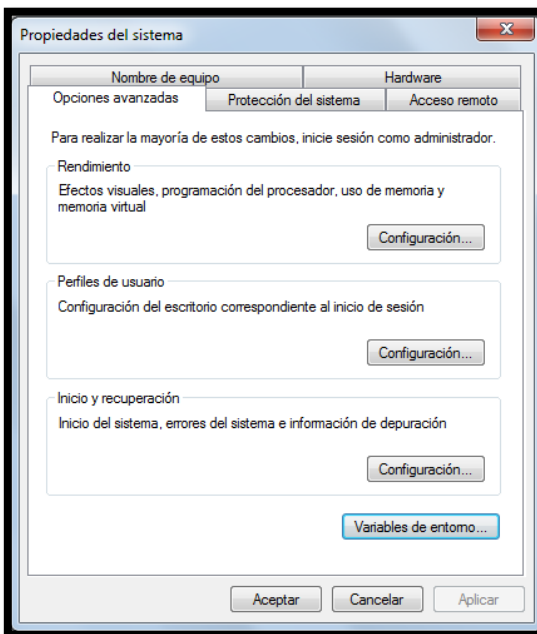
Para programar en Android, lo primero que se hace es preparar el entorno de trabajo. El SDK de Android para desarrolladores provee un conjunto de herramientas que facilitan el desarrollo de aplicaciones para múltiples clientes. La SDK de Android es compatible con varios sistemas operativos, Windows, Mac y Linux.

El procedimiento para instalar las herramientas es el siguiente:

Descargar el SDK de acuerdo al sistema operativo que estemos manejando. En el siguiente sitio web se descarga <http://developer.android.com/sdk/index.html>.¹⁸ El posterior paso a dar es la configuración del “path” de la SDK como una variable de entorno.

Se abre *propiedades del sistema / variables de entorno / variables del sistema*

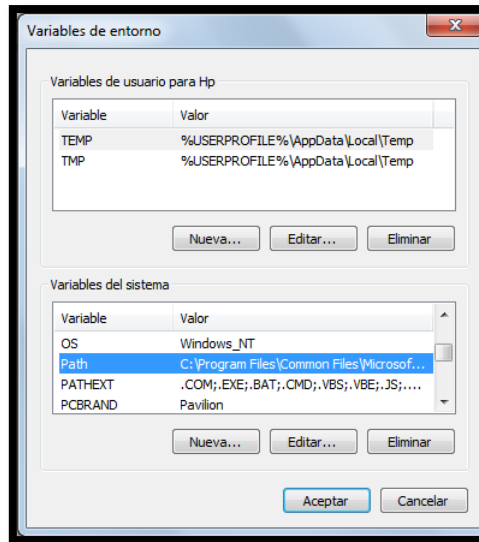
Ilustración 15. Configuración de variables de entorno.



¹⁸ Android Developers, Get the Android SDK. 2013. Página Web. (Fecha de consulta 16 de Julio de 2013). Disponible en: <http://developer.android.com/>

Dentro de la opción de variables del sistema, se busca la variable “path” y se modifica introduciendo la ruta en la cual se encuentra la SDK de Android anteriormente descargada.

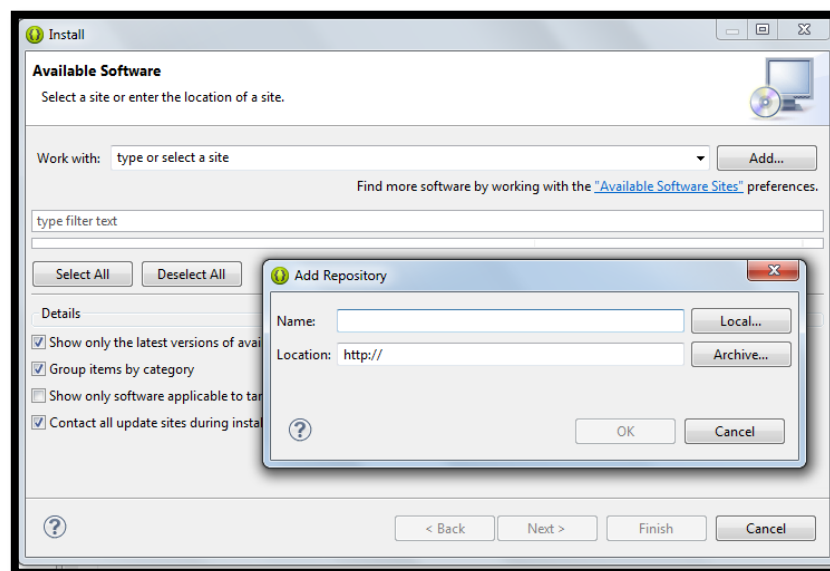
Ilustración 16. Configuración del SDK en path del sistema.



Luego de configurar el SDK, se instala el plugin del entorno de desarrollo Eclipse.

Abierto el IDE Eclipse, se selecciona la pestaña *help / install new software* y en la ventana que emerge se pulsa el botón “add” y se introduce la siguiente URL <https://dl-ssl.google.com/android/eclipse/>

Ilustración 17. Configuración del plugin de Android en Eclipse.



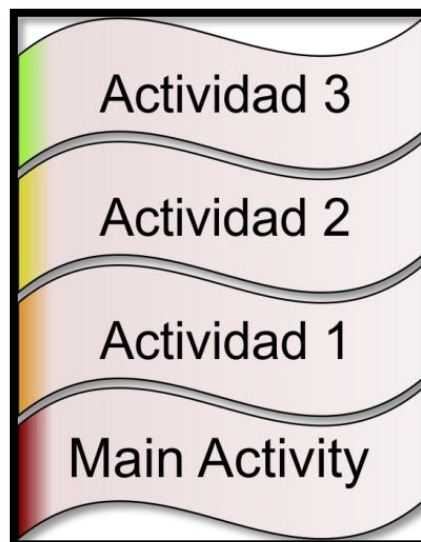
Comenzará a descargar el software necesario para comenzar a desarrollar aplicaciones en este entorno de trabajo. Por último se reinicia el entorno Eclipse para que se puedan configurar los cambios realizados.

Es recomendable tener conocimientos en el desarrollo con el lenguaje Java, ya que las técnicas, la gramática y la sintaxis se traducirán directamente en Android.

3.4 Componentes de una aplicación

Los componentes proporcionan recursos necesarios para todas las aplicaciones, tal vez la más importante es el componente de la actividad ya que es la base de todas las pantallas de interfaz de usuario. Los dispositivos tienen un único objetivo, la ejecución principal, que es la aplicación que está visible en la pantalla, cada aplicación tiene una pila de tareas que es en sí la secuencia de ejecución de procesos de Android. Estas se componen de actividades que se van apilando según son llamadas y solo pueden finalizarse cuando las tareas que tienen por encima están terminadas, o cuando el sistema decide destruirlas porque necesita memoria. El sistema también automáticamente eliminará la aplicación que lleve más tiempo detenida.

Ilustración 18. Pila de actividades de Android.



Los componentes son elementos básicos con los que se construye un proyecto. Hay cuatro tipos, pero se componen principalmente de actividades. Los componentes no pueden hacer funcionar la aplicación por sí mismos, para ello es

necesario usar “Intents”, los cuales transmiten mensajes al sistema de una actividad de destino o servicio, indicando su intención de realizar una acción.¹⁹

Todos ellos deben declararse en el AndroidManifest.

3.4.1 Actividades

Una actividad es el componente principal encargado de mostrar al usuario la interfaz gráfica, es el medio de comunicación entre la aplicación y el usuario. Los elementos mostrados dentro de la interfaz deben ser definidos en un fichero “.xml” que llevan asociado para poder ser tratados en la clase `NameActivity.class`.

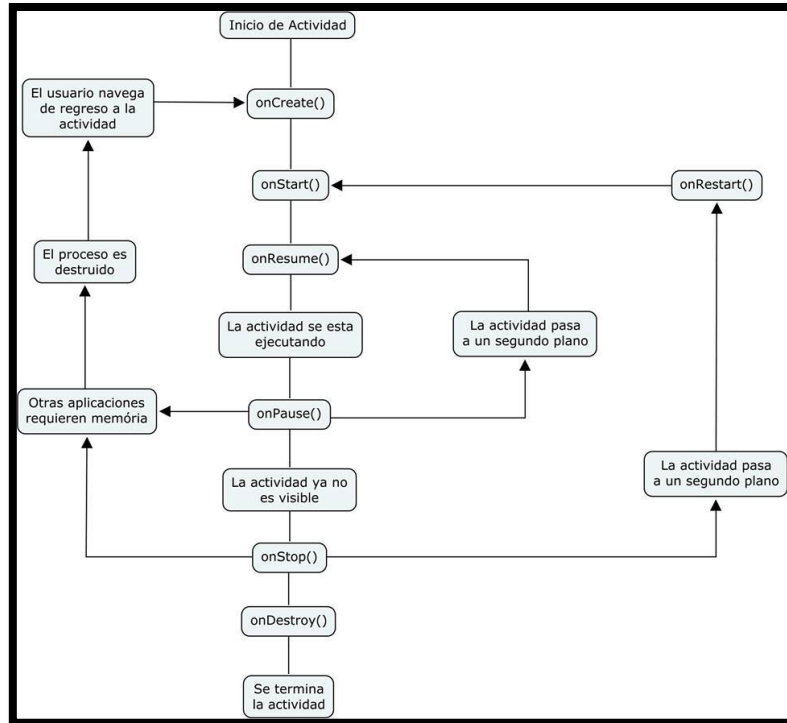
Dentro del fichero “.xml” se definen elementos como la ubicación en la pantalla, botones, textos, etc. Todas las actividades poseen un ciclo de vida, activo, pausado y parado. En el primero la actividad está en ejecución. En el segundo la actividad se encuentra semi-suspendida es decir no es una tarea principal y en el tercero la actividad está detenida totalmente, por lo tanto no es visible para el usuario.

Una vez definido que es un ciclo de vida, debemos tener en cuenta cuales son los métodos importantes en cada uno de ellos:

- `OnCreate(Bundle savedInstanceState)`: Es el método que crea la actividad.
- `OnRestart()`: Reinicia la actividad tras haber sido detenida. La actividad es iniciada desde cero.
- `OnStart()`: Inmediatamente después de `OnCreate` o de `OnRestart()` esta muestra al usuario la actividad.
- `OnResume()`: En este apartado se establece la interactividad entre el usuario y la aplicación. Solo es ejecutado cuando la actividad está en el primer plano.
- `OnPause()`: Es ejecutada cuando una actividad va a dejar de estar en el primer plano para dar paso a otra actividad.
- `OnStop()`: La actividad pasa a un segundo plano por un largo periodo. El sistema puede prescindir de esta para liberar espacio en el caso en que sea requerido.
- `OnDestroy()`: Este es el método final de una actividad.

¹⁹ Desarrolladores de Android. Get the Android SDK. 2013. Página Web. (Fecha de consulta 18 de Julio de 2013). Disponible en: <http://developer.android.com/>

Ilustración 19. Ciclo de vida de una actividad.



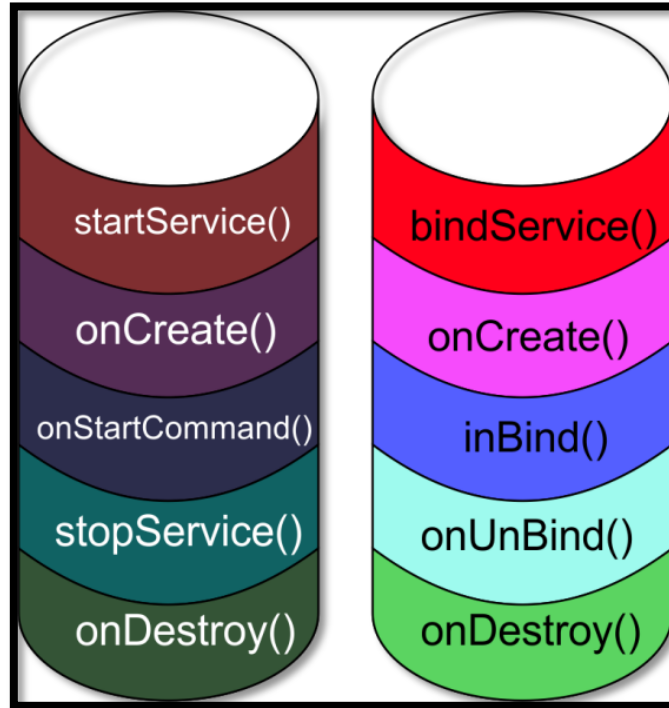
3.4.2 Servicios

Los servicios son tareas no visibles que se ejecutan por debajo, cuando la actividad asociada no está en primer plano. No necesita una interfaz, a menos que se pida explícitamente.

El ciclo de vida de un servicio se inicia con el método `onCreate(Bundle)` y se finaliza con el método `onDestroy()`.

- Si se inicia el método `startService()`, querrá decir que el servicio ejecutará todo ciclo vital. El siguiente método a ejecutar es el `onStartComand(Intent,int)`. Y finalmente el servicio se termina con `stopService()`
- Si se inicia el método `bindService()`, el usuario podrá interactuar a través de la interfaz gráfica, luego se ejecutará el método `onBind(intent)`. En ese caso, el servicio se termina llamando el método `onUbind(Intent)`

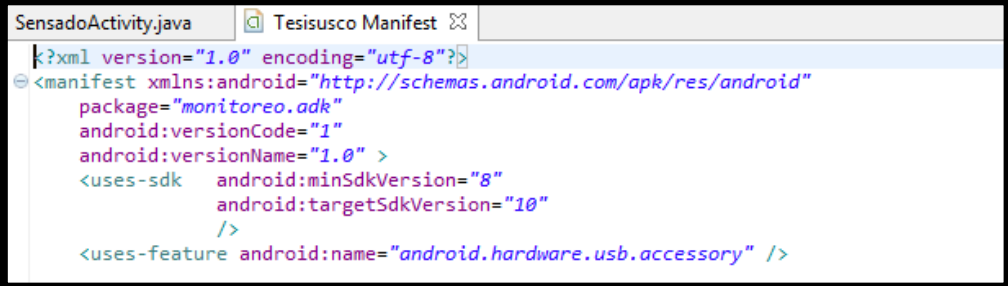
Ilustración 20. Ciclo de vida de un servicio.



- **Receptores de mensajes:**
Son también llamados notificaciones o broadcast, y son los encargados de reaccionar ante los eventos ocurridos en el dispositivo móvil. No disponen de una interfaz pero son capaces de mandar un “activity” por medio de un evento. El ciclo de vida de este componente es muy corto ya que solo está activo mientras se ejecuta el método `onReceive(Context,Intent)`.
- **Proveedores de contenidos:**
Llamados content provider y son los encargados de que la aplicación pueda acceder a la información que necesita. Para obtener los datos necesarios, es necesario conocer la URL del dato, los campos que tiene y los tipos de esos campos.
- **Intents:**
Los intents contienen los datos que describen la operación que desarrollará el componente a quien va dirigido. Son declarados en el “manifest”, pueden ser explícitos es decir, que especifican el componente al que van destinados, mientras los implícitos no.

- **Intent-filters:**
Los intent-filters son únicamente usados por los intents implícitos. Definen que tipos de intent puede lanzar la actividad. Se definen en el AndroidManifest con la etiqueta <intent-filter>. La información que pasa por los intents se compone de tres campos:
 - **Action:** Es una cadena de caracteres que informa el tipo de acción que es llevada a cabo.
 - **Data:** Informa del identificador del dato que se asocia la acción y de qué tipo de dato es.
 - **Category:** Es una cadena de caracteres que contiene información adicional sobre el tipo de componente al que va dirigido el intent.²⁰
- **AndroidManifest:**
El AndroidManifest es un documento .xml que contiene los elementos declarados de una aplicación, además de contener sus permisos, restricciones, procesos, accesos de datos entre otros.

Ilustración 21. Código generado al crear el proyecto en Android Manifest.



```

SensadoActivity.java | Tesisusco Manifest
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="monitoreo.adk"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="8"
        android:targetSdkVersion="10"
        />
    <uses-feature android:name="android.hardware.usb.accessory" />
  
```

Este código se genera al crear el proyecto. Se declara con la etiqueta <manifest> y dentro de este se incluyen los paquetes en que se encuentra la aplicación. Se define la aplicación en la etiqueta <application>, es necesario también incluir todos los componentes en esta. Las actividades se declaran en la etiqueta <activity>, luego se especifican los intent-filters asociados con esta actividad. Se declara con la etiqueta <Service> los servicios, después irán los intent-filters. Los proveedores de contenido se declaran con la etiqueta <provider> y son los únicos componentes que no se declaran intent-filters ya que no son necesarios.²¹

²⁰ Báez Manuel, Zapata Álvaro, Introducción a Android, ISBN: 978-84-96285-39-5, 2000. Segunda edición E.M.E. editorial (16). Madrid, España.

²¹ Android Developers, Activities. Página Web. (Fecha de consulta: 22 de julio de 2013) disponible en: <http://developer.android.com/>.

Para preparar la aplicación Android para el protocolo “open accessory”, no es necesario escribir ningún código. Primero se hacen cambios en el AndroidManifest.xml. Sin embargo, para la comunicación vía USB es necesario declarar el uso de las librerías USB en el “manifest” para dispositivos con versiones anteriores a la 3.1.

Ilustración 22. Código generado librería usb.

```
<uses-library android:name="com.android.future.usb.accessory" />
<activity
    android:name="monitoreo.adk.SensadoActivity"
    android:label="@string/app_name" >
```

La librería USB fue adherida en la versión de Android 2.3.4 y fue llamada com.android.future.usb. Para las versiones superiores a la 3.1 la librería fue declarada dentro del paquete android.hardware.usb.

Ahora es necesario declarar otro intent-filter que es el encargado de iniciar la aplicación cuando el dispositivo Android es conectado con el accesorio.

Ilustración 23. Etiquetas Intent Filter.

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
<intent-filter>
    <action android:name="android.hardware.usb.action.USB_ACCESSORY_ATTACHED" />
</intent-filter>

<meta-data android:name="android.hardware.usb.action.USB_ACCESSORY_ATTACHED"
    android:resource="@xml/accessory_filter" />
```

El segundo intent-filter es un disparador de la actividad para que sea iniciada cuando el dispositivo Android sea vinculado con la tarjeta ADK. El “meta-data” hace referencia a recursos adicionales, que pueden ser proporcionados por los intent-filter para refinar aún más el mecanismo de filtrado.

4 SENSORES

4.1 Velocímetro

El sistema del velocímetro emplea un engranaje que encaja con el eje principal de la transmisión. La velocidad del vehículo es transmitida por el engranaje el cual es comunicado a través del cable al velocímetro ubicado en el grupo de medidores, a fin de indicar la velocidad del vehículo.²²

Un sensor genera un pulso por cada giro de la rueda, el cual simboliza la frecuencia del pulso que varía proporcionalmente a la velocidad del vehículo. En el velocímetro se muestra una interpretación de la frecuencia de los pulsos para mostrar la velocidad instantánea.

Antes que todo existen dos tipos de velocímetros: el velocímetro mecánico y el velocímetro electrónico.

El velocímetro electrónico funciona en base a un sensor de campo magnético, que al rotar produce una tensión que varía proporcionalmente en amplitud y en frecuencia con la velocidad del vehículo.

Esta tensión se alimenta a la unidad de control electrónico del automóvil, que la transforma en una indicación de voltaje para mostrarla en el medidor de velocidad.²³

El velocímetro mecánico va acoplado a la caja de cambios. De esta caja sale un cable de acero el cual llega hasta el indicador de velocidad. Cuando la salida de la caja gira, el cable de acero gira transmitiendo el movimiento rotatorio a un sistema de embrague que hay dentro del velocímetro, esta gira siempre, pero esta descansado sobre una lámina de acetato con la cual roza, al rozar hace que la lámina de acetato se mueva, marcando la velocidad en el indicador.

4.1.1 Efecto hall

Consiste en la aparición de un campo eléctrico en un conductor cuando es atravesado por un campo magnético. A este campo eléctrico se le llama campo Hall. Este efecto fue descubierto en 1879 por el físico estadounidense Edwin Herbert Hall.

4.1.2 Sensor de efecto Hall

Los sensores constan de un elemento conductor o semiconductor y un imán. Cuando un objeto ferro-magnético se aproxima al sensor, el campo que provoca el

²² Mitsubishi, Manual de taller chasis Canter `90, 2013. (Manual de instrucciones). PUB #TWSS 9014.

²³Daewoo. Auto daewoo spark, ¿Cómo funciona el velocímetro? 2013. manual de instrucciones. (Fecha de consulta: junio de 2013). Disponible en: <http://www.autodaewoospark.com>.

imán en el elemento se debilita. Así se puede determinar la proximidad de un objeto, siempre que sea ferro-magnético.²⁴

En este caso simulamos la salida de un sensor de velocidad para alimentar el velocímetro durante diferentes escenarios con la ayuda de un generador de pulsos cuadrados. La frecuencia de funcionamiento del indicador de velocidad (Denso 637REV=1KM) oscila desde 0 Hz hasta 114 Hz que equivalen respectivamente 0 Km/h a 160Km/h.

Ilustración 24. Indicador de Velocidad.



4.1.3 Circuito acondicionador del sensor de velocidad

4.1.3.1 Convertidor de frecuencia a voltaje LM2907

La serie LM2907 de circuitos tacómetro está diseñada para utilizar el mínimo de partes externas y máxima versatilidad. Con el fin de aprovechar sus características y ventajas. La primera etapa del circuito es un amplificador diferencial llevado a una realimentación positiva del flip-flop. La tensión umbral de la entrada es la cantidad de tensión de entrada diferencial a la que la salida de esta etapa cambia de estado.

Existen dos opciones (LM2907-8, LM2917-8) ambos tienen una entrada conectada internamente a tierra de modo que una señal de entrada debe oscilar por encima y debajo de la tierra para superar los umbrales de entrada y producir una salida. Esto se brinda específicamente para recolectar la reluctancia variable magnética que normalmente proporciona una sola salida AC. Esta única entrada está totalmente protegida contra cambios de voltaje a ± 28 V, que son fáciles de lograr con este tipo de pastillas. Las opciones de entrada diferencial (LM2907, LM2917) ofrecen al usuario la opción de establecer su propio nivel de conmutación de entrada y aun así tener la histéresis alrededor de ese nivel para un excelente rechazo de ruido en cualquier aplicación. Con el fin de permitir que las entradas alcancen tensiones de modo común por encima de tierra, se elimina la protección de entrada y tampoco debe ser tomada fuera de los límites de la tensión de alimentación que se está utilizando. Es muy importante que una entrada no vaya por debajo de tierra sin cierta resistencia para limitar la corriente en el diodo.

²⁴ Robótica CRYA, Sensor de efecto hal. 2013. Disponible en: <http://www.crya.com.mx/>.

Después de la etapa de entrada está la CHARGE PUMP, donde la frecuencia de entrada se convierte a un voltaje de corriente continua. Para hacer esto se requiere un condensador de temporización, una resistencia de salida y un integrador o un condensador de filtro. Cuando los cambios en la etapa de entrada indican que el condensador de temporización está bien cargado o descargado linealmente entre dos tensiones cuya diferencia está $V_{CC} / 2$. Luego, en ciclo medio de la frecuencia de entrada o un tiempo igual a $\frac{1}{2} F_{IN}$ el cambio de la carga en el condensador de temporización es igual a $V_{CC} / 2 \times C1$. La cantidad promedio de corriente suministrada dentro o fuera del condensador es:

Ecuación 1. Cantidad de corriente en el condensador.

$$\frac{\Delta Q}{T} = I_{c(AVG)} = C1 \times \frac{V_{CC}}{2} \times (2f_{IN}) = V_{CC} \times f_{IN} \times C1$$

El circuito de salida de esta corriente refleja con gran precisión en la resistencia de carga R1, conectada a tierra, de tal manera que si los pulsos de corriente se integran con un condensador de filtro, a continuación, $V_O = I_C \times R1$, y la ecuación de conversión total se convierte en:

Ecuación 2. Voltaje de salida del conversor.

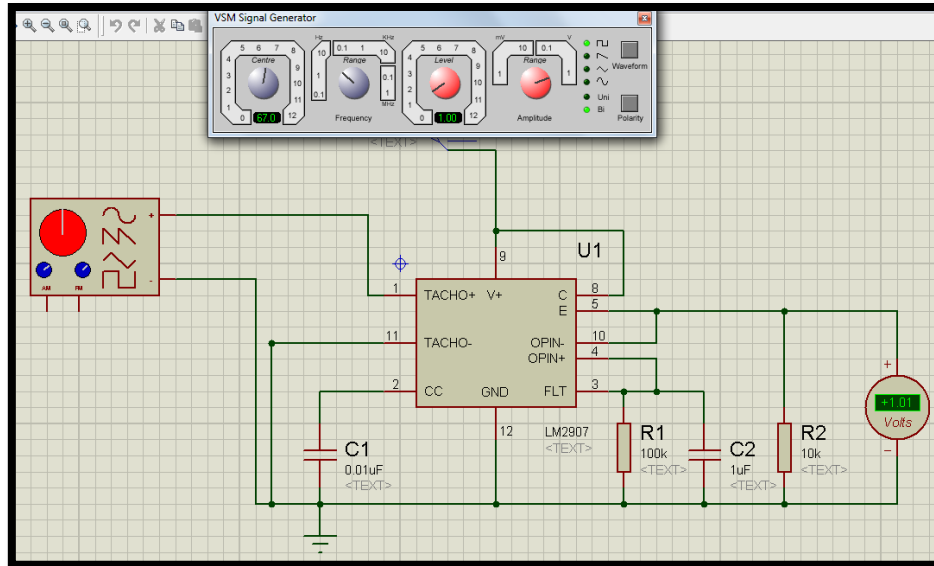
$$V_O = V_{CC} \times f_{IN} \times C1 \times R1 \times K$$

Donde K es típicamente 1, puede variar de 0.9 a 1.1 debido a la diferencia entre el voltaje de salida en alto y el voltaje de salida en bajo que es igual a $V_{CC}/2$, además la ganancia de corriente de salida es igual a la unidad, estos dos factores causan que el tacómetro tenga una ganancia constante.

El tamaño de C2 depende de la cantidad de voltaje de rizo permitida y el tiempo de respuesta requerido.²⁵

²⁵ Texas Instruments, LM2907/LM2917 Frequency to Voltage Converter. Marzo de 2013. (Manual de instrucciones). Disponible en: <http://www.ti.com/lit/ds/symlink/lm2907-n.pdf>

Ilustración 25. Simulación LM2907.



El indicador de velocidad se alimenta con el voltaje de la batería que suele ser 12v, puede variar dependiendo del fabricante.

La alimentación será tomada de la tarjeta Arduino por lo que $V_{CC} = 4.7V$, la frecuencia de salida oscila entre 0Hz a 114 Hz por lo tanto, los valores de C1, C2 y R1 son:

Ecuación 3. Calculo de resistancia y capacitancia.

$$5V = 4.7V \times 114Hz \times R1 \times C1$$

$$0.00933184 = R1 \times C1$$

$$R1 \cong 1k\Omega$$

$$C1 \cong 10\mu F$$

El diseño del circuito impreso ver **Anexo E**.

4.2 Medidor de combustible

El sistema medidor de combustible consiste en una unidad medidora de gasolina instalada en el tanque de combustible, que detecta el nivel de gasolina remanente a fin de mostrar este nivel en el indicador.

El flotador de la unidad medidora de combustible efectúa un movimiento ascendente y descendente conforme a las variaciones del nivel de combustible en

el tanque. Este movimiento hace que el resistor variable se mueva a través del brazo.²⁶

Ilustración 26. Sistema medidor de combustible.

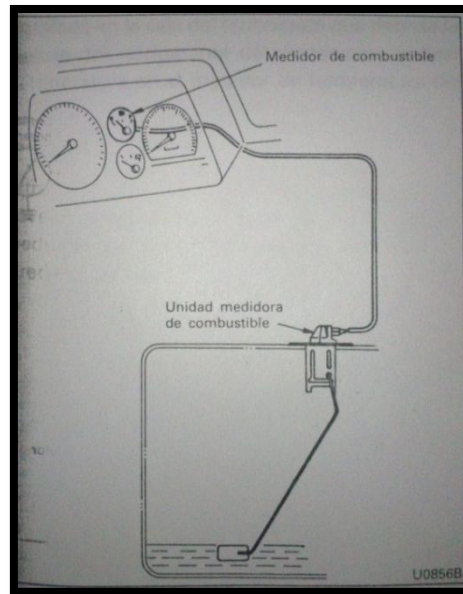
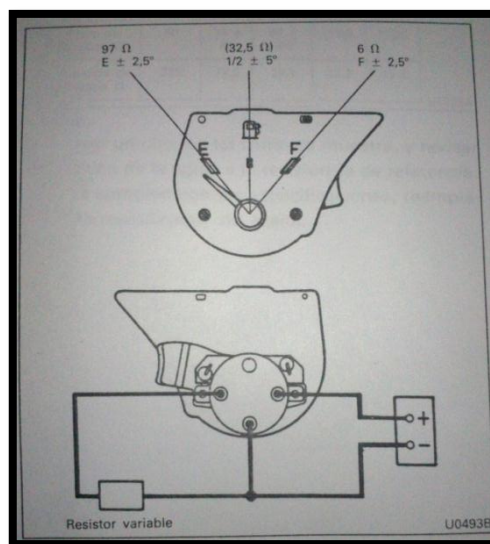


Ilustración 27. Comprobación de deflexión de la aguja.



Fuente: Mitsubishi, Manual de taller chasis Canter `90

²⁶ Mitsubishi, Manual de taller chasis Canter `90, PUB #TWSS 9014.

El medidor de combustible usado se activa al alterar la resistencia del sensor que posee una resistencia interna que varía de 0Ω (4/4) a 300Ω (0). Estos valores no son estándar para todos los indicadores de combustible la resistencia de sensor varía de acuerdo al fabricante.

Para ajustar la aguja del medidor de combustible se utilizó una resistencia variable (Potenciómetro) para demostrar los posibles valores que puede tomar un flotador.

El voltaje de alimentación del indicador de combustible es 12v que salen de la batería del vehículo.

Los voltajes que llegan al medidor para convertir, son:

Tabla 2. Voltaje en los diferentes niveles de combustible.

100% Lleno	0v
50%	3.9v
0% Vacío	5v

Ilustración 28. Medidor de combustible.



4.3 Acelerómetro

La fuerza G no es una medida de fuerza sino una medida intuitiva de aceleración. Está basada en la aceleración que produciría la gravedad terrestre en un objeto cualquiera en condiciones ideales (sin atmósfera u otro rozamiento). Una aceleración de 1G es generalmente considerado como igual a la gravedad estándar, que es de 9.80665 metros por segundo cuadrado (m/s^2).²⁷

²⁷ Colaboradores de Wikipedia. Fuerza G. 2013. La enciclopedia libre. (Fecha de consulta: julio de 2013). Disponible en: http://es.wikipedia.org/wiki/Fuerza_G, Wikipedia (20 jul 2013).

Ilustración 29. Ejemplos de aceleraciones.

	Valor (o rango)
Satélites en órbita y pruebas de gravedad cero ⁵	0 G
Superficie de la luna en el Ecuador	0,1654 G
Superficie de la tierra en el Ecuador al nivel del mar.	1 G
Cohete lunar <i>Saturno V</i> recién despegando	1,14 G
Máximo de una <i>Lanzadera espacial</i> durante el despegue y el reingreso a la atmósfera.	3 G
Montañas rusas más veloces ⁶	3,5–5 G
<i>Apolo 16</i> en reingreso ⁷	7,19 G
Viraje máximo normal en un avión acrobático o un jet de combate.	9 G
Máximo para un humano en un trineo cohete (<i>John Stapp</i>)	46,2 G
Misil Sprint	100 G
Corta exposición de una persona a un choque ⁸	>100 G
Capacidad de absorción de relojes mecánicos ⁹	5.000–7.500 G
Munición 9 × 19 Parabellum de mano (Largo promedio de una munición) ¹⁰	31.000 G
Munición de mano 9 × 19 Parabellum , máximo ¹¹	190.000 G

Al detectar 3Gs saltan los airbag de los vehículos.

Para alimentar el acelerómetro de tres ejes, conecte 2,2v - 16v de fuente de alimentación al pin de V_{IN} . Tenga en cuenta que los pines no toleran 5v, y se requieren componentes externos como divisores de tensión, para conectar la placa a sistemas de 5v. Las conexiones a los pines g-Select y Test son opcionales, la placa trabajará con estos pines desconectados.

Los pines X, Y, y Z son tres salidas de voltajes analógicos. Aceleraciones positivas aumentan la tensión de salida del eje por encima de la tensión de centro y aceleraciones negativas disminuyen la tensión de salida por debajo de la tensión de centro. Las salidas serán siempre dentro de la gama de 0 a la tensión del pin de 3V3.

El pin de selección de sensibilidad, g-Select, es internamente puesto en bajo, se selecciona por defecto una sensibilidad de $\pm 1.5g$ (800 mV/g) en el MMA7361L y $\pm 3g$ (440 mV/g) en el MMA7341L. Poniendo el pin en alto, se selecciona una sensibilidad de $\pm 6g$ (206 mV/g) en el MMA7361L y $\pm 11g$ (118 mV/g) en el MMA7341L

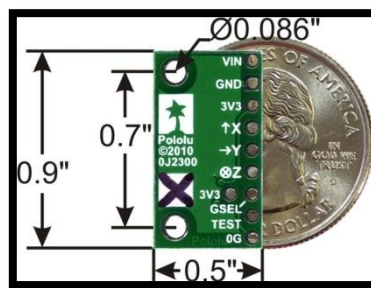
El pin 0g-Detect se pone en alto cuando los 3 ejes simultáneamente detectan 0g, lo cual pasa cuando la placa se encuentra en caída libre. Este pin solo está documentado en el datasheet para mayor sensibilidad MMA7361L (en el MMA7341L este pin no se usa), pero encontramos que funciona en el MMA7341L

por defecto $\pm 3g$ de sensibilidad. Esta salida no funciona con una sensibilidad de $\pm 11g$.

El pin Test es referenciado en tierra en la placa y puede ser desconectado.

Para aplicaciones de muy baja potencia, los tres ejes del acelerómetro pueden ser alimentados entre los 2.2v a 3.6v conectado al pin 3V3. Esto te permite obviar el regulador evitando esta pequeña corriente de fuga y permitiendo que el microcontrolador controle la energía de la placa y la apague cuando no esté en uso.²⁸

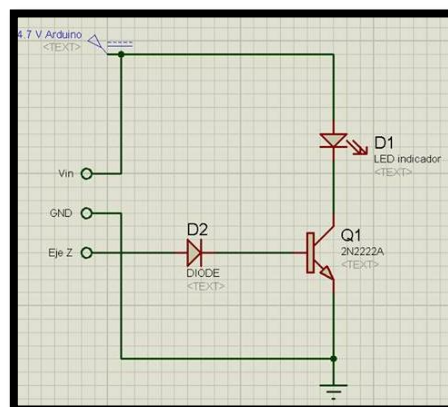
Ilustración 30. Acelerómetro de 3 ejes MMA7361L con regulador de voltaje, vista con dimensiones.



Fuente: <http://b.pololu-files.com/picture/0J2682.250.jpg?cbbda1f8f313d4ff2d5e715a9e3239ad>

El acelerómetro MMA7361L se ajusta perfectamente a nuestras necesidades de medir la vibración en la parte trasera del vehículo también por su capacidad de regulación de voltaje. Se realiza un circuito que indique por medio de un LED ubicado en el tablero del vehículo, la situación en la parte trasera del automotor.

Ilustración 31. Simulación circuito indicador de vibraciones.



²⁸POLOLU, Robotics & Electronics, MMA7361L 3-Axis Accelerometer $\pm 1.5/6g$ with Voltage Regulator. Recurso Web. (Fecha de consulta 03 de mayo de 2013). Disponible en: <http://www.polulo.com>

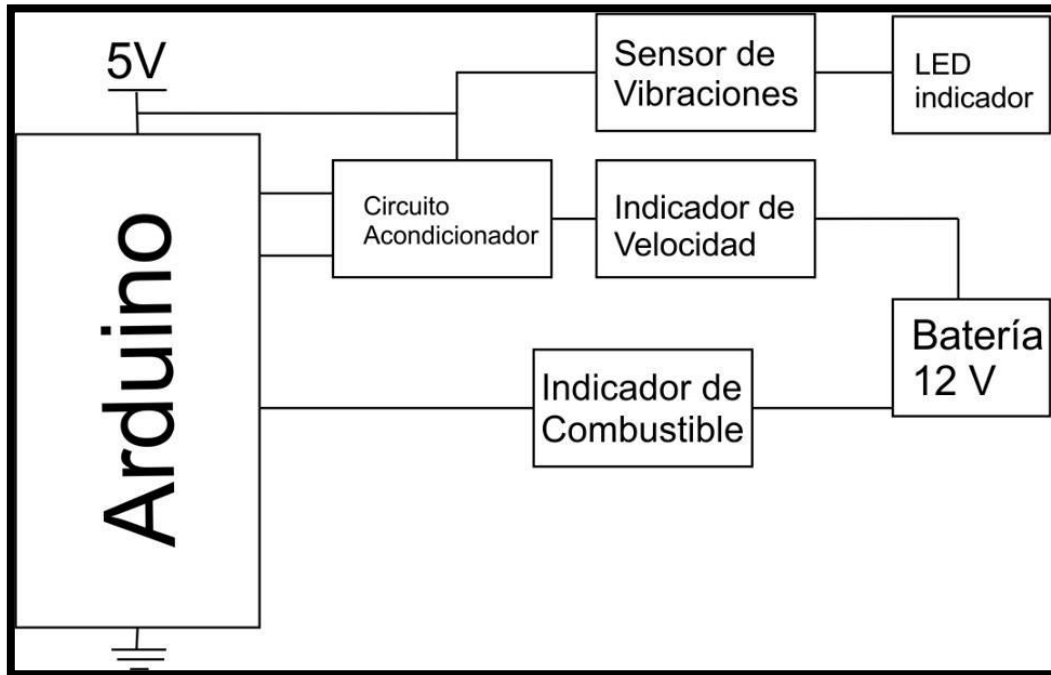
El transistor Q1 es usado como interruptor para dejar pasar el voltaje de la tarjeta hacia el LED cuando el voltaje de base, supera los 0.7v. El voltaje análogo del pin Z, varía entre 0.3v y 0.9v cuando existe movimiento (Vibración) en el eje z. El diodo es usado para asegurar un voltaje de activación de 0.7 V en la base del transistor Q1.

El montaje de los medidores utilizados se puede visualizar en el anexo F (**ver Anexo F**).

5 RECEPCIÓN Y TRANSMISIÓN DE ARDUINO, ANDROID Y PHP

5.1 Recepción Arduino

Ilustración 32. Transmisión de variables a Arduino.



Todas las entradas analógicas de la Arduino permiten entradas de 0v a 5v para su procesamiento digital, los circuitos acondicionadores entregan valores en este rango permitido. En el caso específico del circuito del velocímetro se deben tomar los voltajes máximos y mínimos que están relacionados directamente con las variables de entrada y los convertimos en unidades reales es decir kilómetros /Hora.

Es necesaria una reconversión para la velocidad, la tarjeta interpreta que 1023 muestras (128 bits) es el voltaje máximo y 0 el voltaje mínimo, pero esto acarrea un problema, el indicador de velocidad tiene un máximo de 160 Km/h que equivalen a 1023 muestras entonces, se realiza una reconversión de valores para que el voltaje máximo sea 160; como medida de protección fue preciso tomar solo la mitad de ese rango, por lo que el límite de voltaje máximo fue el doble de la velocidad o sea 320 Km/h equivalen a 5v.

En el caso de la adquisición del nivel de combustible, la medida en el tanque no es totalmente lineal, entonces se diseña un comparador para linealizar la señal correctamente. De 0% a 50% de combustible el nivel de voltaje representa 5v a

3.9v que en muestras son de 800 a 1023 y cuando el nivel de combustible pasa de 50% a 100% representa 3.89v a 0v que en muestras son 0 a 799.

Se alimenta el acelerómetro, el LED indicador y el circuito de acondicionamiento con el pin 5v que proporciona 4.7v de la tarjeta Arduino.

5.2 Transmisión Arduino a celular

El paquete de referencia ADK contiene dos librerías, que establecerán la comunicación USB. Se trata de una versión modificada de la librería *USB_Host_Shield*, originalmente creado por Oleg Mazurov de Circuitos @ Home. La librería fue originalmente diseñada para trabajar con un *Host Shield USB* Arduino. Sólo algunas modificaciones de menor importancia se han realizado en la librería ya que el chip USB es compatible con las placas ADK que se hace equivalente al *USB Host Shield*. La segunda librería es la librería *AndroidAccessory*, que es responsable de la aplicación del Protocolo de accesorio Abierto.

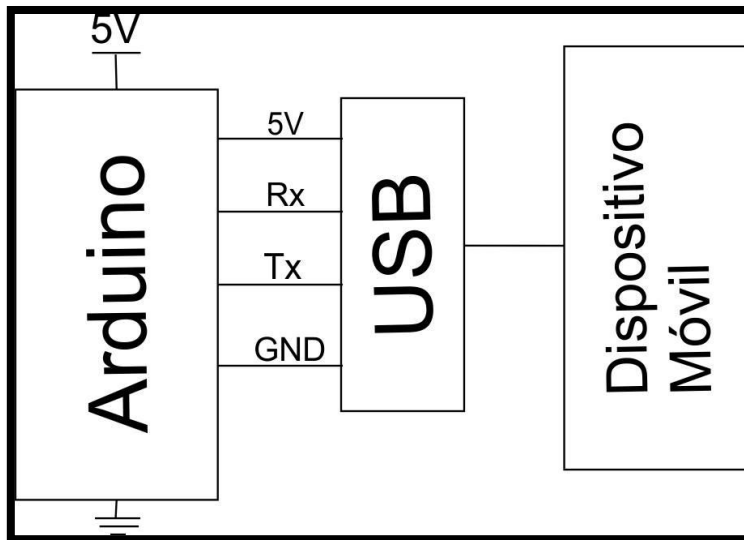
```
#include <Max3421e.h>
#include <Usb.h>
#include <AndroidAccessory.h>
```

Como se puede apreciar, fueron necesarios hacer tres cambios en el sketch para la comunicación de *open accesory*. Lo primero que se debe hacer es inicializar un objeto *AndroidAccessory*, que implemente los protocolos *open accesory*. Los parámetros más importantes son el fabricante, modelo y versión que serán utilizados en la aplicación Android para verificar que se está comunicando con la placa ADK correcta.

En el *void setup()* se configura el objeto en un estado activo con el método *power on*. La rutina *loop* comprueba en cada bucle si algo está conectado al “accesory” y sólo a continuación, ejecuta el código dentro.²⁹

²⁹ Böhmer Mario, *Beginning Android ADK with Arduino*, (2010). Editorial Frensoft Apress. (Página 55). EE.UU.

Ilustración 33. Conexión USB Arduino a un dispositivo móvil.



Como primer paso se emplea un *switch case* en donde se hace la matriz de bytes a ser enviados. De acuerdo con el protocolo auto-definido, el primer byte establece el comando tipo constante, el segundo byte establece el destino de la constante y el tercer byte es el valor que va a ser transmitido hacia el objetivo. Cuando la matriz de bytes está lista, el método de escritura del objeto *AndroidAccessory* transmitirá los datos a través del flujo de salida para el dispositivo Android. Igualmente se administran tiempos de retardo para enviar datos cada 2 minutos para darle el tiempo al GPS de triangular su posición y obtener un valor preciso.

Ilustración 34. Código switch case para la transmisión de Arduino.

```
envio_datos_final | Arduino 1.0.2
Archivo Editar Sketch Herramientas Ayuda
envio_datos_final

switch (count++ % 0x10) {
  case 0:
    currentValue = analogRead(INPUT_PIN_0);
    sntmsg[0] = VELOCIDAD;
    sntmsg[1] = INPUT_PIN_0;
    sntmsg[2] = map(currentValue, 0, 1023, 0, 320);
    acc.write(sntmsg, 3);
    // Serial.println(map(currentValue, 0, 1023, 0, 320));
    delay(40000);
    break;
  case 0x2:
    currentValue = a; //
    sntmsg[0] = USUARIO;
    sntmsg[1] = IDEUS;
    sntmsg[2] = currentValue;
    acc.write(sntmsg, 3);
    delay(40000);
}
```

5.3 Recepción al celular

Ahora que se ha asegurado que ambos dispositivos se reconocen entre sí debemos comunicarlos. Los mensajes se envían y reciben a través de cadenas de bytes. En una aplicación Android esto se hace mediante la lectura y la escritura en el flujo de entrada y de salida de un archivo especial. En el lado Arduino, la clase *AndroidAccessory* proporciona métodos para leer y escribir mensajes.³⁰

Los dos primeros bytes del mensaje tienen que coincidir con los bytes definidos en el sketch de Arduino, por lo que ellos definen como lo siguiente:

Ilustración 35. Bytes de conexión.

```
private static final byte COMBUSTIBLE = 0x4;  
private static final byte VELOCIDAD = 0x5;
```

CombustibleView, como su nombre lo indica, es una vista personalizada que se extiende al sistema de clases Android view. Después de leer el mensaje recibido, se debe convertir la matriz de bytes de nuevo a su valor entero original.

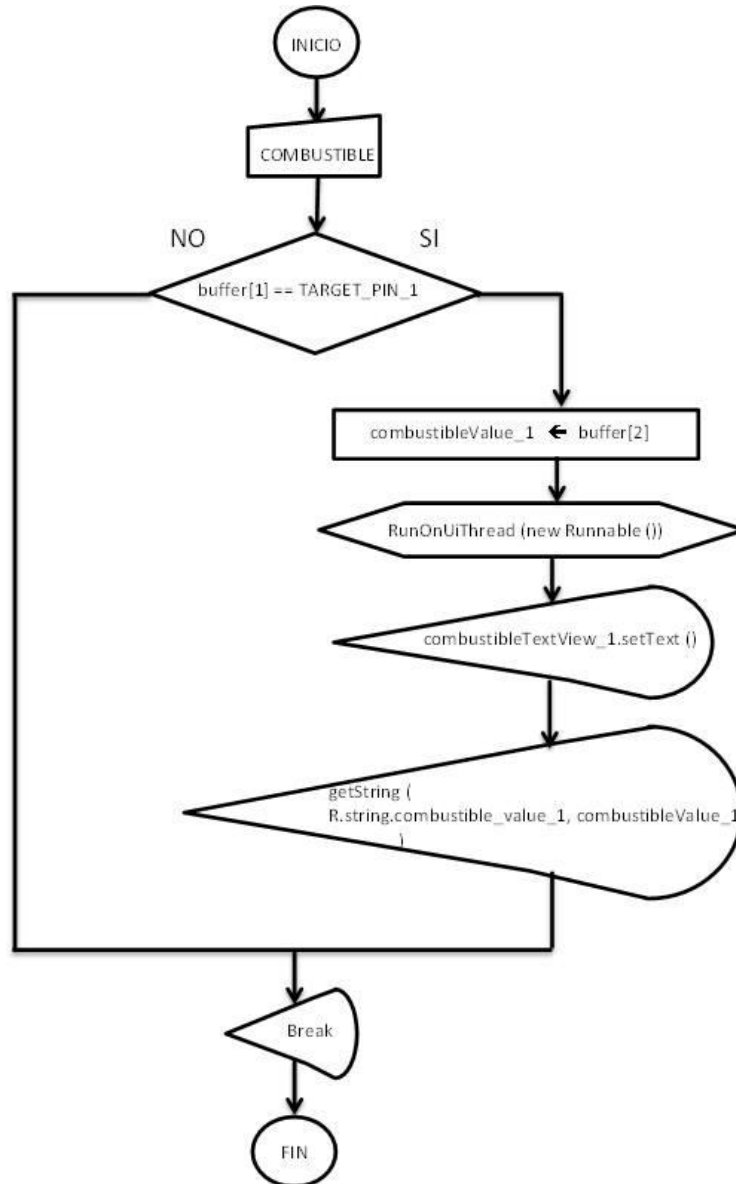
Ilustración 36. Etiquetas de texto.

```
private TextView velocidadTextView;  
private TextView combustibleTextView_1;
```

La última cosa por hacer es transferir el valor a la *CombustibleTextView_1* para que pueda establecer una visualización del combustible en la pantalla del dispositivo. Se aplica un caso para recepción de múltiples datos, esto se logra empleando un comparador de buffer con el byte 2 que establece el destino.

³⁰ Böhmer Mario, *Beginning Android ADK with Arduino*, (2010). Editorial Frensoft Apress ,(Página 56).Estados Unidos.

Ilustración 37. Comparador buffer recepción.



5.4 Recepción del GPS

El GPS funciona mediante una red de 24 satélites en órbita sobre el planeta tierra, a 20,200 kilómetros, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos. Con base en estas señales, el dispositivo sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo, y de tal modo mide la distancia al satélite mediante "triangulación", la cual se basa en determinar la

distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites.³¹

5.4.1 Software GPS móvil

La recepción del posicionamiento global se hace a través de una clase llamada *LocationManager* la cual provee acceso al servicio de localización. Este servicio permite a las aplicaciones obtener actualizaciones periódicas de la geolocalización del dispositivo.³²

Ilustración 38. Clase *LocationManager*.

```
// Getting LocationManager object
LocationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
```

En el *Manifest* es necesario crear los permisos para que la aplicación acceda al uso de los sensores GPS del dispositivo móvil y a los servicios de localización por internet. Para la posición por internet se debe determinar la ubicación según la disponibilidad de antena de telefonía móvil y puntos de acceso Wi-Fi.

Ilustración 39. Permisos para acceder a los servicios de localización.

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
```

La sentencia *.getBestProvider()*, devuelve el nombre del proveedor que mejor cumple con los criterios dados. Sólo los proveedores que están permitidos para ser visitados por la actividad de la llamada serán devueltos. Si varios proveedores cumplen con los criterios, se devuelve el que tenga la máxima precisión. Los criterios son los siguientes:

- Requerimientos de potencia
- Precisión
- Aceleración
- Altitud

³¹ Colaboradores de Wikipedia. GPS. La enciclopedia libre.2013. (Fecha de consulta, 13 de junio de 2013).
Extraído de <https://es.wikipedia.org>. Wikipedia.

³² Android Developers, *LocationManager*. 2013. Página Web. (Fecha de consulta 16 de Julio de 2013).
Disponible en: <http://developer.android.com/>

Ilustración 40. Código de obtención de mejor proveedor.

```
// Creating an empty criteria object
Criteria criteria = new Criteria();
// Getting the name of the provider that meets the criteria
provider = locationManager.getBestProvider(criteria, true);
```

5.5 Recepción de la hora

Los *widgets* son elementos visuales básicos que, combinados en una aplicación, mantienen todos los datos procesados por la aplicación y las interacciones disponibles en estos datos.³³ Eclipse provee de un *widget* llamado *DigitalClock* que toma la hora del dispositivo móvil y la presenta en la interfaz de usuario por medio de un *TextView*

5.6 Proceso de comunicación entre celular y PHP

5.6.1 JSON

Acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON.³⁴

Podemos utilizar JSON y HTTP como un intermedio de comunicación entre Android y PHP. Se resume en los siguientes pasos:

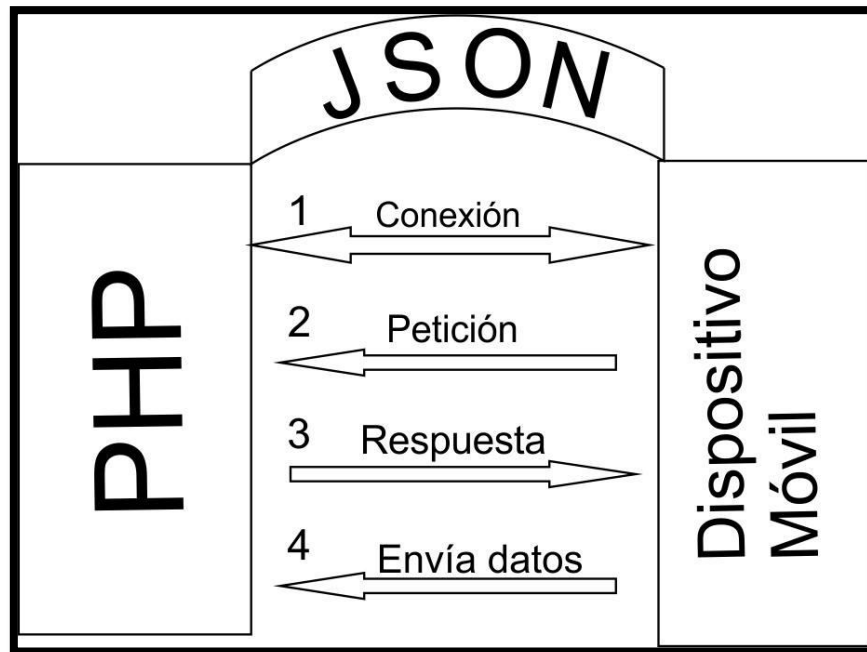
- Una url para enviar los datos a una página php.
- Una solicitud de HTTP que se envía desde Android (Cliente) a PHP (Servidor).
- Tomar los datos de las etiquetas *TextView* de la interfaz gráfica.

³³Colaboradores de Wikipedia, GUI widget , 2011. la enciclopedia libre. (Fecha de consulta, 27 de Abril 2013). Disponible <https://en.wikipedia.org>

³⁴Colaboradores de Wikipedia, JSON. 2011. La enciclopedia Libre. (Fecha de consulta, 6 de Julio 2013). Disponible en: <https://es.wikipedia.org>,

- Construir los parámetros a enviar y etiquetarlos con sus respectivos nombres.
- PHP interpreta los códigos para hacer la conexión con la base de datos y enviar la solicitud a MySQL con los datos etiquetados.

Ilustración 41. Comunicación entre Android y PHP.



6 CONFIGURACIÓN DEL SERVIDOR

6.1 XAMPP

X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. La instalación de un servidor web Apache no es fácil y se complica aún más si se desea agregar MySQL, PHP y Perl.

XAMPP Es realmente simple de instalar y usar; basta descargarlo, extraerlo y comenzar.

XAMPP es una compilación de software libre (comparable a una distribución Linux), es gratuito y libre para ser copiado conforme los términos de la licencia GNU General Public License. Sin embargo, sólo la compilación de XAMPP está publicada bajo la licencia GPL. Desde el punto de vista de XAMPP, el uso comercial es también libre.

XAMPP 1.8.3 contiene:

- Apache 2.4.4
- MySQL 5.6.11
- PHP 5.5.0
- phpMyAdmin 4.0.4
- FileZilla FTP Server 0.9.41
- Tomcat 7.0.41 (with mod_proxy_ajp as connector)
- Strawberry Perl 5.16.3.1 Portable
- XAMPP Control Panel 3.2.1 (from hackattack142)³⁵

La instalación del Xampp permite todos estos servicios, para la tesis se activa el servidor Apache y MySQL

6.2 PHP

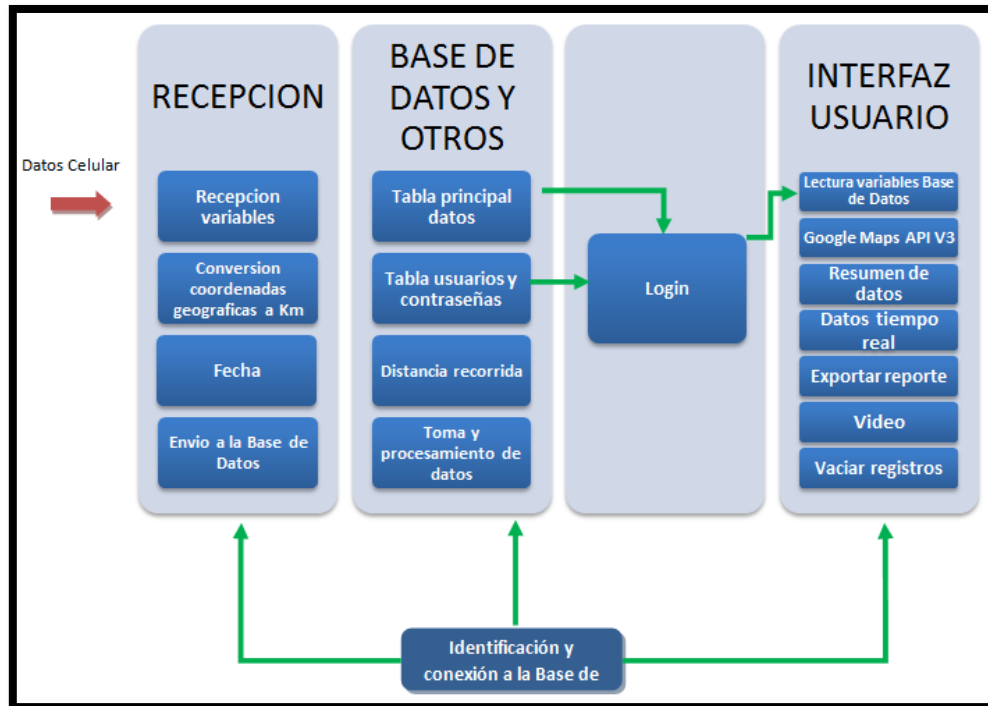
PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, y es fácil de aprender.³⁶

³⁵ Colaboradores de Apache friends. XAMPP Control Panel 3.2.1 (from hackattack142). 2010, (fecha de consulta, 3 de julio de 2013). Disponible en: <http://www.apachefriends.org>,

³⁶Colaboradores php. Net. PHP. (Página Web). Extraído de <http://www.php.net/>

La aplicación de PHP a nuestro sistema se distribuye en bloques que contienen los diferentes códigos que trabajaran dentro del servidor.

Ilustración 42. Diagrama de bloques de la programación PHP.



6.2.1 Bloque de recepción

Este bloque contiene los archivos PHP que recibe los datos provenientes del celular. Este bloque se ejecuta en tiempo real y por ende no impide la realización de otras actividades.

- **Recepción variables:** Este código guarda en variables independientes cada uno de los datos recibidos desde el celular con el método “\$_POST”. El celular hace una conexión al servidor y se encarga de mandar los datos de manera separada, de esta forma los datos se mantienen hasta cierto punto privados y tenemos la ventaja de que podemos enviar una mayor cantidad de datos en la solicitud.³⁷
- **Conversión de coordenadas geográficas a Km:** Un método para describir la posición de una ubicación geográfica en la superficie de la tierra consiste en utilizar mediciones esféricas de latitud y longitud. Estas son mediciones

³⁷ Camacho.J. POST y GET, 2007. bien explicado. [Mensaje en Foro]. (Fecha de consulta: 02 de junio de 2013). Extraído de: [http://www.svcommunity.org/forum/web/\\$_post-y-\\$_get-bien-explicados/](http://www.svcommunity.org/forum/web/$_post-y-$_get-bien-explicados/)

de los ángulos (en grados) desde el centro de la tierra hasta un punto en la superficie de la tierra. Este sistema de referencia generalmente se denomina sistema de coordenadas geográficas.³⁸

Se determina la distancia entre los puntos con la aproximación esférica de la superficie terrestre mediante la fórmula Haversine.

Ilustración 43. Formula Haversine.

$$\text{havrsin} \left(\frac{d}{R} \right) = \text{havrsin}(\varphi_1 - \varphi_2) + \cos(\varphi_1) \cos(\varphi_2) \text{havrsin}(\Delta\lambda).$$

Dónde

havrsin es la función haversine, $\text{havrsin}(\theta) = \frac{1 - \cos(\theta)}{2}$

d es la distancia entre los dos puntos (a lo largo de un círculo máximo de la esfera,

R es el radio de la esfera,

φ_1 es la latitud del punto 1,

φ_2 es la latitud del punto 2, y

$\Delta\lambda$ es la diferencia de longitud.³⁹

Al tener calculada la distancia, se le asigna una variable que posteriormente será enviada a la base de datos.

- Fecha: Es obtenida con la función “date” que toma la fecha del servidor, le asigna una variable y después se envía a la base de datos.
- Envío a la base de datos: Al hacer la respectiva identificación y conexión a la base de datos con los archivos PHP anteriormente creados, se hace una petición en la cual va el nombre de cada una de las columnas y posteriormente se especifica el nombre de las variables que contienen los valores que se guardaran en ella.

6.2.2 Bloque base de datos y otros

El siguiente bloque explica las diferentes tablas y programas independientes que no están en ningún bloque específico pero que si cumplen una función en la interfaz final.

³⁸ ArgicsExplorer. Proyecciones del Mapa. Contenido Web. (Fecha de consulta 78 de mayo de 20013). Disponible en: http://webhelp.esri.com/arcgisexplorer/900/es/map_projections.htm

³⁹ Colaboradores de Wikipedia. Fórmula de Haversine. 2013. Wikipedia la enciclopedia libre. (Fecha de consulta 16 de Julio de 2013). Disponible en: <https://es.wikipedia.org>.

- Tablas: En Mysql se usaron dos tablas, la principal que guarda todos los datos recibidos del bloque anterior en 9 columnas:

Ilustración 44. Base de datos.

The screenshot shows the MySQL interface for a database named 'practica' and a table named 'bus02'. The table structure is as follows:

#	Columna	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
1	id	int(11)			No	Ninguna	AUTO_INCREMENT	Cambiar Eliminar Más
2	ruta	varchar(50)	utf8_spanish_ci		No	Ninguna		Cambiar Eliminar Más
3	velocidad	float			No	Ninguna		Cambiar Eliminar Más
4	combustible	float			No	Ninguna		Cambiar Eliminar Más
5	latitud	double			No	Ninguna		Cambiar Eliminar Más
6	longitud	double			No	Ninguna		Cambiar Eliminar Más
7	suma	double			No	Ninguna		Cambiar Eliminar Más
8	hora	varchar(10)	latin1_swedish_ci		No	Ninguna		Cambiar Eliminar Más
9	fecha	varchar(10)	latin1_swedish_ci		No	Ninguna		Cambiar Eliminar Más
10	acelerometro	varchar(10)	latin1_swedish_ci		No	Ninguna		Cambiar Eliminar Más

- id: Identificación numérica secuencial de cada fila de la tabla.
- Ruta: Identificación de la ruta del vehículo monitoreado.
- Velocidad: Valores de la velocidad medida.
- Combustible: Porcentaje del combustible recibido.
- Latitud y longitud: Estas dos columnas guardan las coordenadas geográficas enviadas por el GPS.
- Suma: Los valores de distancia en Km convertidos desde las coordenadas geográficas.
- Hora: Hora enviada desde el celular.
- Fecha: La fecha en formato D.M.Y.

La otra tabla de la base de datos contiene los usuarios y contraseñas registrados para el acceso a la página.

- Programas independientes: Existen cinco programas independientes que no están ligados a ningún bloque en especial, pero cumplen funciones específicas de seguridad y estabilidad de la interfaz.
 - Identificación a la base de datos: Guarda la información de usuario y contraseña de la base de datos. Es esencial aislar esta información de los otros programas ya que por razones de seguridad estos datos no deben ser accesibles desde la web.
 - Conexión a la base de datos: Crea la conexión a la base de datos con la información suministrada en el programa anterior, lo cual agiliza las

consultas y funciones utilizadas en todas las otras aplicaciones en la que interviene.

- Distancia recorrida: Es una sub-rutina la cual da como respuesta la distancia total recorrida por el vehículo. Opera todos los valores de la columna “suma” de la base de datos.
- Procesamiento de datos: Se analizan los datos guardados en la base de datos para hallar el combustible consumido, distancia recorrida, la cantidad de veces que sobre paso la velocidad limite que son 60Km/H en zonas urbanas y tiempo transcurrido. Esta información es enviada al siguiente bloque para su visualización.⁴⁰
- Login: Hace la identificación del usuario para ingresar a la interfaz principal comparando los datos asociados a la tabla de usuarios en la base de datos.

6.2.3 Bloque interfaz de usuario

Para el usuario final se desarrolló un portal web que dispone de la visualización de los datos recopilados en el transcurso de los bloques anteriores.

- Lectura de datos: Se inicia llamando los archivos de identificación y conexión a la base de datos mencionados en el apartado anterior. Desde la base de datos se toma el último dato guardado de la columna de latitud y longitud para asignarlas a sus respectivas variables.
- GOOGLE MAPS API V3: La API Google Maps te permite embeber mapas directamente en tu sitio web. Sólo se necesita un poco de JavaScript, y algo de estilos CSS para embellecer.⁴¹

La versión 3 de esta API está especialmente diseñada para proporcionar una mayor velocidad y que se pueda aplicar más fácilmente tanto a móviles como a las aplicaciones de navegador de escritorio tradicionales.

El API proporciona diversas utilidades para manipular mapas (como la de la página <http://maps.google.com>) y para añadir contenido al mapa mediante diversos servicios, permitiéndote crear sólidas aplicaciones de mapas en tu sitio web.

⁴⁰ Código de tránsito. Disponible en: <http://www.colombia.com/noticias/codigotransito/t3c11.asp>

⁴¹ Jordán. M, Insertar y personalizar Google Maps en tu sitio web. 2009. (Fecha de consulta 01 de junio de 2013). Recurso en línea. Extraído de <http://www.elwebmaster.com/general/insertar-y-personalizar-google-maps-en-tu-sitio-web>

La versión 3 del API de JavaScript de Google Maps es un servicio gratuito disponible para cualquier sitio web que sea gratuito para el consumidor.⁴²

El mapa se crea en un ambiente HTML pero como se manejan variables externas tomadas de la base de datos se crea un archivo PHP con el HTML embebido.

Primero que nada es necesario crear el puerto de visualización y obtener el archivo Javascript del código Google.

Se añade esta línea entre <head> y </head>:

Ilustración 45. Archivo JavaScript de Google.

```
<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false"></script>
```

Luego de la URL, se coloca sensor=false dado que no se utiliza ningún sensor, para localizar el lugar, esto se programa en false.

Ilustración 46. Configuraciones básicas del mapa.

```
function initialize() { //configuraciones basicas del mapa
  var latlng = new google.maps.LatLng(latitud,longitud); //variable que incluye latitud y longitud
  var settings = { //crea variable setting
    zoom: 14,

    center: latlng,
    mapTypeControl: true, //habilita os botones de vista del mapa(satelital o mapa)
    mapTypeControlOptions: {style: google.maps.MapTypeControlStyle.DROPDOWN_MENU}, //hace la seleccion de la vista un menu desplegable
    navigationControl: true,
    //navigationControlOptions: {style: google.maps.NavigationControlStyle.SMALL},
    mapTypeId: google.maps.MapTypeId.ROADMAP}; //define que nuestro mapa inicie con vista mapa
  var map = new google.maps.Map(document.getElementById("map_canvas"), settings); //crea la variable map con las configuraciones creadas
```

Dentro de esta función se definen las configuraciones básicas del mapa. Las cuales son:

google.maps.LatLng(latitud, longitud): Representa la posición en latitud y longitud del centro de nuestro mapa.

⁴² Desarrolladores Google. Developer guide. (Fecha de consulta 01 de junio de 2013). Disponible en: <https://developers.google.com/maps/documentation/javascript/?hl=es>

Luego de esto se crea la variable *settings*.

- *Zoom*: Especifica cuánto zoom tendrá el mapa.
- *Center*: Especifica nuestro centro.
- *mapTypeId*: Define que tipo de mapa se desea mostrar al inicio.
- *scrollwheel*: Habilita o inhabilita el poder hacer “zoom” con la rueda del ratón.
Valores: *True/false*, por defecto *true*.
- *mapTypeControlOptions*: Son las opciones de visualización iniciales del control de tipo de mapa.
- *navigationControl*: Habilita o inhabilita el control de navegación.
Valores: *True/False*, por defecto *true*.
- *scaleControl*: Habilita o inhabilita el control de escala.
Valores: *True/False*, por defecto *true*.⁴³

Este código crea la variable “*map*”, y define que el mapa debe utilizar las configuraciones que se han creado.

Ilustración 47. Código variable map.

```
var map = new google.maps.Map(document.getElementById("map_canvas"), settings);
```

Se añadió un marcador personalizado:

Ilustración 48. Icono personalizado.



Ilustración 49. Código para ubicar el marcador en la posición deseada.

```
var companyPos = new google.maps.LatLng(latitud, longitud); //posicion del marcador  
var companyMarker = new google.maps.Marker({
```

⁴³Jordán. M, Insertar y personalizar Google Maps en tu sitio web. 2009. (Fecha de consulta 01 de junio de 2013). Recurso en línea. Extraído de <http://www.elwebmaster.com/general/insertar-y-personalizar-google-maps-en-tu-sitio-web>

new google.maps.Marker: Se inserta el marcador con su respectiva posición las cuales están cargadas en las variables latitud y longitud.

Ilustración 50. Código para cargar el icono personalizado.

```
var companyLogo = new google.maps.MarkerImage('googlemaps/images/buslogo05.png', //apunta al nombre de la imagen
```

new google.maps.MarkerImage("ruta de imagen"); se agregó el icono deseado en la posición del marcador.

Posteriormente se dibujó una *PolyLine* que indicara el recorrido que debe seguir el vehículo.

Ilustración 51. Código para dibujar una *PolyLine*.

```
var ruta = [ //array de coordenadas donde pasa la linea  
new google.maps.LatLng(2.9412083, -75.2977166), //punto 1  
new google.maps.LatLng(2.9418777, -75.2974777), //punto 2
```

Var ruta = []: Crea un array con cada uno de los puntos (latitud y longitud) del recorrido de la línea para su trazo en el mapa.

- Datos tiempo real: Muestra los últimos valores recibidos de la base de datos.

Ilustración 52. Captura de imagen Últimos datos recibidos.

Últimos datos recibidos:	
Combustible:	0 %
Ruta:	50
Hora:	16:54:03
Velocidad:	83 Km/H

- Resumen de datos: Se insertan los archivos del bloque procesamiento de datos para su visualización en una tabla que sintetiza la jornada de trabajo.

Ilustración 53. Captura de imagen Resumen de la jornada

Resumen de la jornada:					
Combustible consumido:	28 %	Distancia recorrida:	1.871Km	Sobrepaso de velocidad:	225 veces
Hora inicial:	16:36:21	Hora final:	16:54:03	Tiempo transcurrido:	0:17:42

- Exportar reporte: El código se inicia con un botón que nos permite ver toda la información de la base de datos que estén en el momento en un archivo formato .PDF para mantener y guardar el registro de trabajo en cualquier momento del día.
- Vaciar registros: Esta parte de código hace que se borren todos los datos almacenados en la tabla principal de la base de datos, esta función se encuentra implementada en un botón.
- Video: Para el video se implementó un banner con el link de la dirección de la cámara

Ilustración 54. Captura de imagen Banner video.



Sirve con cualquier navegador y dispositivo móvil que soporte aplicaciones .jar; para aprovechar el máximo funcionamiento a la interfaz de la cámara se recomienda usar el internet Explorer.

6.3 Cámara IP

Una Cámara IP (también conocidas como cámaras Web o de Red) son videocámaras especialmente diseñadas para enviar las señales (video, y en algunos casos audio) a través de Internet desde un explorador (por ejemplo el Internet Explorer) o a través de concentrador (un HUB o un SWITCH) en una Red Local (LAN)

En las cámaras IP pueden integrarse aplicaciones como detección de presencia (incluso el envío de mail si detectan presencia), grabación de imágenes o secuencias en equipos informáticos (tanto en una red local o en una red externa

WAN), de manera que se pueda comprobar por qué ha saltado la detección de presencia y se graben imágenes de lo sucedido.⁴⁴

Ilustración 55. Cámara IP.



Fuente: <http://thebesthomesecurity.org/wp-content/uploads/2011/01/Foscam-FI8918W.jpg>

Tabla 3. Especificaciones cámara IP.

ITEMS		FI9818W
Image Sensor	Image Sensor	CMOS Sensor
	Display Resolution	1280*720 Pixels
	Lens	f: 2.8mm, F:2.4
	Min. Illumination	0 Lux
	Viewing Angle	70°
Audio	Input/Output	Linear Input/Output, which can realize two-way voice the intercom
	Audio Compression	G.711/G.726, Audio sampling Bit rate 128kbps ~ 2.048Mbps

⁴⁴ Todo en domótica, Inmótica y control, ¿Qué es una cámara IP?. 2012. Página Web. (Fecha de consulta: 01 de julio de 2013). Disponible en : <http://www.domodesk.com/>,

Video	Image Compression	H.264
	Light Frequency	50Hz, 60Hz
	Image Resolution	720p (1280*720) , VGA (640*480) , QVGA (320*240)
	Image frame rate	25fps(50Hz), 30fps(60Hz),Down adjustable
	View	H.264 dual streaming
	Image adjust	The brightness, contrast, saturation, chromaticity, is adjustable
	White balance	Automatic, Indoor, outdoor , is adjustable
	Infrared mode	Automatic , manual (on/off)
	Night visibility	5mm 11 IR LEDs, Night visibility up to 8 meters
Network	Ethernet	One 10/100Mbps RJ-45
	Supported Protocol	TCP/IP、 UDP/IP、 HTTP、 SMTP、 FTP、 DHCP、 DDNS、 UPNP
	Wireless Standard	WIFI(IEEE802.11b/g/n)
	Data Rate	802.11b: 11Mbps(Max.) 802.11g: 54Mbps(Max.) 802.11n: 150Mbps (Max.)
	Support IP address	static IP address, dynamic IP address
Alarm	Alarm detection	The motion detecting, detection threshold Settings
	Alarm inform	Support Email、 FTP etc alarm inform way
Hardware Interface	Memory	/
	POWER Interface	DC 5V/2.0A (EU,US,AU adapter or other types optional) , Power Consumption < 8 Watts
	Audio Input/Output Interface	One audio input jack,One audio outputt jack.
	Reset Buttons	One Reset
	Ethernet	One 10/100Mbps RJ-45
	Wireless Interface	WIFI antenna
Other	Pan/Tilt Angle	Horizontal:300° & Vertical: 120°
	LED Explain	One SD CARD Instructions, One power Instructions
Environment	Operate Temp	0° ~ 40°C (32°F ~ 104°F)
	Operate Humidity	10% ~ 80% non-condensing
	Storage Temp	-10°C ~ 60°C(14°F ~ 140°F)
	Storage Humidity	0% ~ 90% non-condensing

Physical	Dimension(LxWxH)	110.5*103*127mm
	Gross Weight	768g(color box size:200*120*175mm)
	Net Weight	420g
PC Requirements	CPU	2.0GHz or above (suggested 3.0GHz)
	Memory Size	256MB or above (suggested 1.0GHz)
	Display Card	64MB or above
	Supported OS	Microsoft Windows 2000/XP, Vista, Windows7, Mac OS, Linux
	Browser	IE6 and above version or compatible browser,Firefox or other standard browsers
Certification	CE,FCC	
Warranty	Limited 2-year warranty	

Para instalar la cámara, dentro de un vehículo, es necesario el uso de un router 3G Wi-Fi.

Ilustración 56. Router 3G Wi-Fi.



Fuente:http://img2.mlstatic.com/router-wifi-n-con-puerto-para-modem-3g-tplink-mr3420-300mbps_MLA-O-109305390_4691.jpg

El router 3G Wi-Fi funciona de manera que irradia internet dentro el vehículo para la transmisión de datos, fue necesario contratar un servicio de internet móvil para conectarlo al modem 3G.

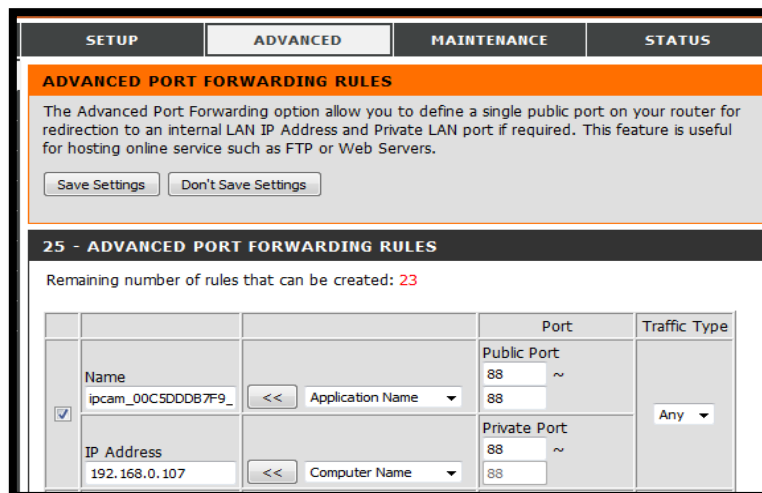
Ilustración 57. Internet móvil Tigo.



Fuente: http://bimg2.mlstatic.com/internet-movil-tigo-prepago-acepto-cambios_MCO-F-4220165836_042013.jpg

La cámara requiere que por medio de las configuraciones de administrador del modem, se abra un puerto para la transmisión por internet, de igual forma se asignó un puerto fijo a la cámara para que no se reasigne una nueva dirección IP al desconectar el router.

Ilustración 58. Abrir puertos del router.



ADVANCED PORT FORWARDING RULES				
The Advanced Port Forwarding option allow you to define a single public port on your router for redirection to an internal LAN IP Address and Private LAN port if required. This feature is useful for hosting online service such as FTP or Web Servers.				
<input type="button" value="Save Settings"/>		<input type="button" value="Don't Save Settings"/>		
25 - ADVANCED PORT FORWARDING RULES				
Remaining number of rules that can be created: 23				
	Name	IP Address	Port	Traffic Type
<input checked="" type="checkbox"/>	ipcam_00C5DDDB7F9_	192.168.0.107	Public Port: 88 Private Port: 88	Any

Para que el video pueda transmitirse a internet, se hace necesario el uso de un DNS, FOSCAM ofrece un servicio gratuito de DNS para cada una de sus cámaras.

Para suministrar energía a todos los dispositivos (Cámara, Router 3G, Tarjeta Arduino) se debe usar una fuente inversora de corriente que va conectada en el cenicero que suministra los 12v necesarios para hacerlo funcionar.

Ilustración 59. Inversor de corriente 12v a 110v.

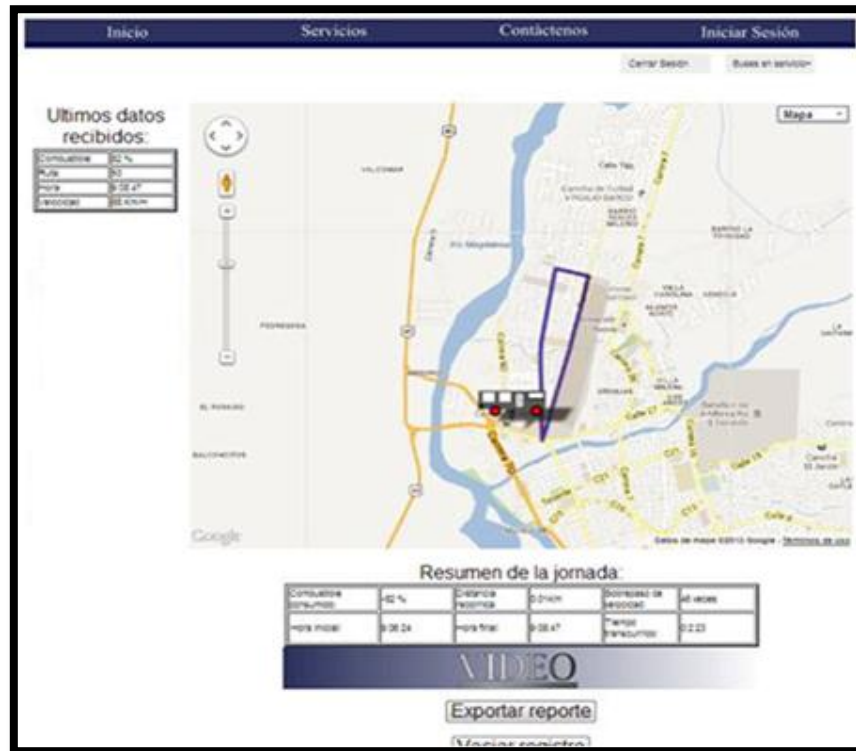


Fuente: http://img1.mlstatic.com/conversor-inversor-corriente-carro-12v-dc-110v-ac-200watts_MCO-O-3509921598_122012.jpg

6.4 Portal web

6.4.1 Interfaz de usuario portal Web

Ilustración 60. Captura interfaz de usuario portal web.



La ilustración muestra el portal web que contiene el mapa con una ruta predefinida, la posición actual del vehículo, los últimos datos recibidos y el resumen de la jornada laboral.

Ilustración 61. Captura reporte de la jornada.

Reporte Monitores de Bus
información de la base de datos

ID	Ruta	Distancia	Combinatoria	hora
1	50	86	81	9:06:26
2	50	-118	82	9:06:45
3	50	82	83	9:06:46
4	50	107	84	9:06:51
5	50	102	85	9:06:52
6	50	117	86	9:06:55
7	50	117	87	9:06:55
8	50	117	88	9:06:57
9	50	126	89	9:06:59
10	50	126	90	9:07:01
11	50	-122	91	9:07:02
12	50	-122	92	9:07:06
13	50	-126	93	9:07:07
14	50	-126	94	9:07:08
15	50	-126	95	9:07:11
16	50	-126	96	9:07:13
17	50	-126	97	9:07:14
18	50	-126	98	9:07:16
19	50	-126	99	9:07:17
20	50	-126	100	9:07:19
21	50	118	101	9:07:23
22	50	118	102	9:07:25
23	50	82	103	9:07:27
24	50	82	104	9:07:28
25	50	126	105	9:07:31
26	50	126	106	9:07:33
27	50	126	107	9:07:36
28	50	126	108	9:07:37
29	50	87	109	9:07:38
30	50	87	110	9:07:41
31	50	86	111	9:07:43
32	50	86	112	9:07:46
33	50	109	113	9:07:47
34	50	109	114	9:07:49
35	50	126	115	9:07:51
36	50	-126	116	9:07:56
37	50	126	117	9:07:53
38	50	-126	118	9:07:57
39	50	-126	119	9:07:58
40	50	-126	120	9:08:01
41	50	129	121	9:08:03
42	50	129	122	9:08:06
43	50	119	123	9:08:07
44	50	119	124	9:08:09
45	50	117	125	9:08:11
46	50	117	126	9:08:13
47	50	124	127	9:08:16
48	50	124	128	9:08:17
49	50	-126	129	9:08:19
50	50	-126	130	9:08:21
51	50	121	131	9:08:23
52	50	121	132	9:08:26
53	50	86	133	9:08:27
54	50	86	134	9:08:29
55	50	86	135	9:08:31
56	50	86	136	9:08:33
57	50	103	137	9:08:36
58	50	103	138	9:08:37
59	50	109	139	9:08:39
60	50	109	140	9:08:41
61	50	103	141	9:08:43
62	50	103	142	9:08:46
63	50	86	143	9:08:47

Resumen de la jornada:

-52 %	Cantidad recorrida:	0.01km	Comprobos de velocidad:	46 veces
9:06:24	hora final:	9:20:47	Tiempo transcurrido:	0:14:23

VIDEO

Exportar reporte

Vaciar registro

Por medio del uso del botón “exportar reporte”, se crea un documento PDF que contiene un registro los valores más relevantes. Para reiniciar este registro, se utiliza el botón de “vaciar registro”.

7 PRESUPUESTO

Tabla 4. Presupuesto.

RECURSOS	PRECIO
ACELERÓMETRO	43000
INDICADOR DE COMBUSTIBLE	30000
INDICADOR DE VELOCIDAD	120000
CAMARA IP	175000
CELULAR CON S.O. ANDROID	400000
MODULO ARDUINO MEGA ADK	215000
PLAN DE DATOS	50000 Mensual
CIRCUITERIA EXTERNA	50000
EQUIPO DE COMPUTO	1500000
ROUTER 3G	90000
FUENTE INVERSORA DE CORRIENTE	75000
IMPREVISTOS	200000
PAPELERIA	150000
TOTAL:	3098000

8 CONCLUSIONES

- Se logró diseñar, simular y probar un sistema de monitoreo y registro en tiempo real de las variables aceleración, combustible, velocidad, fecha, hora, posición, kilometraje integrados en una caja. Este dispositivo permite a las empresas dar un mayor control de sus vehículos y conductores. SMYRT es un sistema poco invasivo, de fácil instalación y mantenimiento.
- Todo el sistema se basó en software y hardware libre lo que nos permitió reducir costos y tener un mayor acceso a la información para desarrolladores. Esta tecnología sirve como medio para prevención de futuros accidentes ya que sus características ayudan a determinar las fallas en cuanto a la seguridad en la prestación del servicio.
- El entorno de desarrollo Arduino es agradable, fácil de entender y por ser libre ofrece varias ventajas, entre ellas que los usuarios avanzados compartan sus librerías y conocimientos de desarrollo de manera educativa.
- Se realizó un acondicionamiento del sensor de velocidad, ya que entregan un tren de pulsos que aumentaba su frecuencia al aumentar la velocidad para ello se utilizó un circuito conversor de frecuencia a voltaje. La tarjeta Arduino efectuó el proceso de digitalizar las diferentes señales analógicas de los indicadores y linealizar la variable del medidor de combustible.
- Se realizó una aplicación móvil en el sistema operativo Android para realizar la recepción de los valores anteriormente procesados de Arduino y la posterior transmisión a la base de datos del servidor.
- Fue necesario el uso de la versión 2.3.7 de Android (*Gingerbread*) porque es la más comercial, además de ser compatible con los sistemas Arduino. No se garantiza el funcionamiento en otras versiones de Android.
- Para la programación en eclipse fue utilizado el API 10 de Google ya que para el desarrollo generaba menos errores.
- Existe un error al iniciar la aplicación debido a que la base de datos se encuentra vacía, se soluciona reconectando el dispositivo, reiniciando la aplicación.
- Se profundizó en los lenguajes de programación: Java, C++, C, PHP, Javascript, HTML, CCS y Arduino.

- La posición no solo se toma del sensor GPS sino también de internet y así el programa selecciona el valor de mayor precisión.
- Se optó que el sistema envíe datos cada 2 minutos para darle el tiempo al GPS de triangular su posición y obtener un valor preciso, se puede modificar este tiempo dependiendo de las necesidades del usuario.
- Se diseñó un portal web con la información básica del proyecto y un acceso restringido para ingresar a la interfaz principal que contendrá todos los datos monitoreados además se acondiciono una cámara IP para una transmisión de video y audio.
- Se logró exportar los valores de la base de datos en un documento PDF para tener un registro de lo sucedido en la jornada. Existe un error en la creación del reporte al exceder 250 valores en la base de datos.
- Hay una optimización en el control actual de los autobuses ya que pasan de solo 2 controles de una variable a 30 controles de 6 variables por ruta (ruta de prueba 1 hora 30 min).

9 RECOMENDACIONES

- Se recomienda un sensor de choque o sensor de fuerza G para poder sentir mejor estas fuerzas.
- Se recomienda usar un dispositivo Android no necesariamente un celular (Tablet Android 2.3 con wi-fi) para minimizar costes.
- Se recomienda usar un hosting para que el portal web y la base de datos sean accesibles desde internet y no localmente.
- Es necesario el uso de un router administrable, que facilite la apertura de los puertos necesarios para la transmisión de video por internet.
- Es necesario asignar una IP fija a la cámara para que al momento de reconectarse no haya errores al abrir los puertos.
- La interfaz de la cámara permite la conexión de más terminales de video al mismo tiempo para aumentar el rango de visión.
- Se recomienda una cámara con las mismas características, además de SD Card para guardar el video.
- La información recopilada se puede usar para estadísticas nacionales de seguridad vial
- Se recomienda usar la última versión del Api de Google Maps ya que se encuentran liberadas y son de acceso público. Las antiguas versiones requerían un registro de aplicación, una clave y eran muy limitados.
- Se recomienda utilizar otros dispositivos o medios para medir la velocidad del vehículo ya que no siempre se usan los mismos sensores y el sistema sea poco invasivo.
- Se recomienda que el dispositivo Android quede fuera de la caja, ya que al ocurrir algún error en la conexión se facilita el cambio y la configuración del terminal.
- En el caso de utilizar un dispositivo celular, se recomienda dejar el modo avión para evitar interrupciones.
- Se recomienda vaciar la base de datos al iniciar la jornada laboral para evitar el error de exportación.

- Se recomienda exportar los datos en otro formato para realizar un análisis más detallado.
- Se recomienda complementar los protocolos de seguridad en el portal web.
- Para pulir el sistema, se considera necesario agregar una alerta en caso de que el vehículo sobrepase un rango estimado en el acelerómetro que nos indicaría un choque.

10 FUENTES DE CONSULTA

10.1 Bibliográficas

- Pastrana John, Polanco Camilo y otros. diseño e implementación de un sistema de monitoreo por radiofrecuencia para el registro de horarios de buses de transporte. Neiva 2005. Tesis de grado. Universidad Surcolombiana. Facultad de ingeniería. (Fecha de consulta, 22 de Julio de 2013).
- Morales, L y Trujillo, O. MONITOREO DE VEHICULOS DE TRANSPORTE DE SERVICIO PÚBLICO. Neiva 2005. Tesis de grado. Universidad Surcolombiana, Facultad de Ingeniería. (Fecha de consulta, 22 de Julio de 2013).
- Báez; Manuel, Zapata Álvaro, Introducción a Android, ISBN: 978-84-96285-39-5, 2000. Segunda edición E.M.E. editorial (16). Madrid, España. (Fecha de consulta, 22 de Julio de 2013).
- Böhmer Mario, Beginning Android ADK with Arduino, (2010). Editorial Frensoft Apress. (Página 55). EE.UU. (Fecha de consulta, 22 de Julio de 2013).

10.2 Medios virtuales

- Instituto español de Comercio Exterior (ICEX), Sistemas Inteligentes de Transporte en EEUU, 18 de octubre 2011. Extraído de: WWW.NAVACTIVA.COM. (Fecha de consulta, 22 de Julio de 2013).
- Negocio tecnológico. Nuevos autobuses europeos para un transporte público eficiente. 15 de noviembre 2012. Extraído de <http://www.negociotecnologico.com> . (Fecha de consulta, 22 de Julio de 2013).
- INDRA. INDRA implanta su tecnología inteligente para la gestión del tráfico y el transporte público de Medellín por 9 m€. (14 de agosto 2012). Extraído de <http://www.indracompany.com> . (Fecha de consulta, 22 de Julio de 2013).

- Ministerio de Transporte. Plan de Seguridad Vial 2011-2016. Bogotá D.C. Extraído de: <https://www.mintransporte.gov.co/descargar.php?idFile=8>. (Fecha de consulta, 22 de Julio de 2013).
- Creative Commons CC, Attribution-ShareAlike. 2012. Página Web. Fuente: <http://creativecommons.org> (Fecha de consulta, 01 de Julio de 2013).
- Arduino Team, Arduino Homepage. 2013. Página Web. . DISPONIBLE EN: <http://www.arduino.cc/>. (Fecha de consulta, 01 de julio de 2013).
- Arduino Team, Arduino ADK.2013. Página Web. Disponible en: <http://www.arduino.cc/>. (Fecha de consulta, 04 de mayo de 2013).
- Arduino Team, Download the Arduino software.2013. Página Web.. Disponible en: www.arduino.cc/. (Fecha de consulta, 22 de julio 2013)
- Processing programming language, Homepage. Página Web. Disponible en: <http://www.processing.org/>. (Fecha de consulta, 06 de mayo de 2013).
- Source Forge, WinAVR. 2010. En línea. Extraído de: <http://www.winavr.sourceforge.net/>. (Fecha de consulta, 19 de mayo de 2013).
- Arduino c.c. Arduino playground, Libraries for Arduino. 2011. Extraído de: <http://www.arduino.cc/> (Fecha de consulta, 03 de junio de 2013).
- Colaboradores de Wikipedia, Historial de versiones de Android. 2013. Wikipedia la enciclopedia libre Disponible en: <https://es.wikipedia.org>. (Fecha de consulta, 16 de Julio de 2013).
- Desarrolladores de Andorid. Get the Android SDK. 2013. Página Web. Disponible en: <http://developer.android.com/>. (Fecha de consulta, 18 de Julio de 2013).
- Android Developers, Activities. Página Web. Disponible en: <http://developer.android.com/>.(Fecha de consulta, 22 de julio de 2013)
- Daewoo. Auto daewoo spark, ¿Cómo funciona el velocímetro? 2013. manual de instrucciones. Disponible en: <http://www.autodaewoospark.com>. (Fecha de consulta, 22 de junio de 2013).

- Texas Instruments, LM2907/LM2917 Frequency to Voltage Converter. marzo de 2013. (Manual de instrucciones). Disponible en: <http://www.ti.com/lit/ds/symlink/lm2907-n.pdf> (Fecha de consulta, 22 de junio de 2013).
- Colaboradores de Wikipedia. Fuerza G. 2013. La enciclopedia libre. (Fecha de consulta 06 julio de 2013). Disponible en: http://es.wikipedia.org/wiki/Fuerza_G, Wikipedia (Fecha de consulta, 20 julio 2013).
- POLOLU, Robotics & Electronics, MMA7361L 3-Axis Accelerometer $\pm 1.5/6g$ with Voltage Regulator. Recurso Web. Disponible en: <http://www.polulo.com>. (Fecha de consulta, 03 de mayo de 2013).
- Colaboradores de Wikipedia, GUI widget, 2011. la enciclopedia libre). Disponible <https://en.wikipedia.org>. (Fecha de consulta, 27 de Abril 2013).
- Colaboradores de Wikipedia, JSON. 2011. La enciclopedia Libre Disponible en: <https://es.wikipedia.org>. (Fecha de consulta, 06 de Julio 2013).
- Colaboradores de Apache friends. XAMPP Control Panel 3.2.1 (From hackattack142). 2010, Disponible en: <http://www.apachefriends.org>. (Fecha de consulta, 3 de julio de 2013).
- Colaboradores php. Net. PHP. (página Web). Extraído de <http://www.php.net/> . (Fecha de consulta, 3 de julio de 2013).
- Camacho, J. POST y GET, bien explicado. 2007. [Mensaje en Foro]. Extraído de: [http://www.svcommunity.org/forum/web/\\$_post-y-\\$_get-bien-explicados/](http://www.svcommunity.org/forum/web/$_post-y-$_get-bien-explicados/). (Fecha de consulta, 02 de junio de 2013).
- ArcGisExplorer. Proyecciones del Mapa. Contenido Web. Disponible en: http://webhelp.esri.com/arcgisexplorer/900/es/map_projections.htm. (Fecha de consulta 78 de mayo de 20013).
- Colaboradores de Wikipedia. Fórmula de Haversine. 2013. Wikipedia la enciclopedia libre. Disponible en: <https://es.wikipedia.org>. (Fecha de consulta, 16 de Julio de 2013).
- Jordán. M, Insertar y personalizar Google Maps en tu sitio web. 2009. Recurso en línea. Extraído de

<http://www.elwebmaster.com/general/insertar-y-personalizar-google-maps-en-tu-sitio-web>. (Fecha de consulta, 01 de junio de 2013).

- Desarrolladores Google. Developer guide. Disponible en: <https://developers.google.com/maps/documentation/javascript/?hl=es>. (Fecha de consulta, 01 de junio de 2013).

11 ANEXOS

Anexo A. Placas oficiales de Arduino:

- **Arduino UNO.**
Esta placa Arduino se conecta al computador a través de un cable estándar USB. Esta conexión permite alimentar la placa además de otorgar una vía para la programación.
- **Arduino Mega**
Esta placa Arduino es similar a la UNO, pero tiene mayor capacidad de procesamiento además de poseer un número de pines de entradas y salidas tanto digitales como analógicas.
- **Arduino Mega ADK**
La placa Arduino ADK es similar a la Arduino Mega, pero con una interface USB incorporada para conectarse con teléfonos basados en el sistema operativo Android.
- **Arduino Pro**
La placa Arduino Pro es similar al Arduino UNO, pero diseñada con la intención de ser instalada semipermanentemente. La placa no cuenta con los distintos conectores, es compatible con algunas extensiones Arduino y se puede alimentar con baterías de 3.3 Voltios.
- **Arduino Ethernet**
Esta placa es similar al Arduino UNO, pero no tiene soporte USB, pero si cuenta con un conector RJ-45 para dar soporte Ethernet. En algunas versiones, es posible alimentar la placa a través de la propia Ethernet.
- **Arduino Fio**
Esta placa está orientada para usarlo como un nodo inalámbrico. Posee conectores para un módulo Xbee (módulo inalámbrico) [], un conector para una batería LiPo (polímeros de litio) y un circuito para cargar la batería.
- **Arduino LilyPad**
Es una placa Arduino con forma circular, reducida al máximo, está diseñada para ser cosida a la ropa cualquier otro tipo de fibra flexible.
- **Arduino BT**
La placa Arduino BT contiene un módulo bluetooth que facilita las comunicaciones móviles.
- **Arduino Nano**
Es una placa Arduino todo en uno, tiene un diseño compacto para poder ser usada en cualquier placa prototipo.
- **Arduino Serial**
Las placas de Arduino Serial se vendieron en kits no ensamblados para ser montados por uno mismo, principalmente usadas en el aprendizaje.
- **Arduino Mini**

El Arduino Mini es una placa muy compacta de Arduino. Para poder reducir espacio, la comunicación con el PC se hace a través de adaptadores USB.

- **Arduino Pro Mini**
Al igual que el Arduino Mini, es una placa compacta, pero no consta de los cabezales ni conectores montados, ya que está diseñada con la intención de ser instalada permanentemente.

Anexo B. Shields producidos por usuarios.

- **RobotShopRover**
Es una plataforma móvil diseñada para el entorno Arduino. Los usuarios pueden personalizarla a gusto añadiendo funcionalidades.
- **Arduino Motor Shield**
Este Shield permite a Arduino controlar motores CC, servos y motores Step.
- **Arduino Proto Shield**
Este Shield ofrece al usuario un área para soldar componentes.
- **Arduino WirelessProto Shield**
Este Shield le da a una placa Arduino la posibilidad de comunicarse de manera inalámbrica, además de ofrecer una pequeña área para soldar componentes.
- **Arduino Wireless SD Shield**
Al igual que la Arduino WirelessProto Shield da comunicación inalámbrica pero dando un soporte para acceder a una memorita tipo SD Card.
- **Arduino Ethernet Shield**
Este Shield da la posibilidad a una placa Arduino de conectarse a una red Ethernet y tener acceso a internet.

Anexo C. Operadores y funciones

Sintaxis

- ; (limitador de línea de código)
- // (comentarios)
- {} (limitador de bloques de código)
- #define (definición de precompilador)
- #include (inclusión de código externo)

Estructuras de control

- if (comparador si - entonces)
- ifelse (comparador si - si no)
- for (bucle contador)
- switch case (comparador múltiple)
- while (bucle por comparación booleana)

- break (salida de bloque de código)
- return (devuelve el valor al programa)

Operadores booleanos

- && (y)
- || (o)
- ! (negación)

Operadores de comparación

- == (igual a)
- != (distinto de)
- < (menor que)
- > (mayor que)
- <= (menor o igual que)
- >= (mayor o igual que)

Los tipos de datos que usa el IDE de Arduino son muy similares a los de otros lenguajes de programación, tales como el C o Java.

Las variables pueden ser declaradas en cualquier parte del código y si no se inicializan en algún valor, estas valdrán cero.

- char (Carácter, ocupa 8 bits en la memoria)
- unsignedchar (Carácter sin signo).
- boolean (Booleano, puede ser cierto o falso, ocupa 8 bits en la memoria)
- byte (Dato de 8 bits)
- int (Entero de 16 bits con signo)
- word (Palabra sin signo)
- long (Entero de 32 bits con signo)
- float (Valor en coma flotante de 32 bits)
- double (Valor en coma flotante)
- array (Vector)
- void (Vacío)
- string (Cadena de caracteres)

Algunas de las funciones manejadas por el entorno Arduino simplifican en gran medida la solución de problemas comunes. Muchas de esas librerías están limitadas a ser usadas en los pines determinados y pueden necesitar de argumentos previos que no se detallaran en el siguiente apartado.

E/S Analógica

- analogReference(tipo) // configura el tipo de referencia analógica
- analogRead(pin) // devuelve una lectura analógica

- analogWrite(pin, valor) // escritura analógica

E/S Digital

- pinMode(pin, modo) // configura el pin en modo de entrada o salida
- digitalWrite(pin, valor) // escritura digital
- digitalRead(pin) // devuelve una lectura digital

Tiempo

- millis() // Tiempo de arranque en ms
- micros() // Tiempo de arranque en μ s
- delay(ms) // Retardo activo en ms
- delayMicroseconds(μ s) // Retardo activo en μ s

Cálculo

- min(x, y) //Devuelve el mínimo entre los dos números
- max(x, y) // Devuelve el máximo entre los dos números
- abs(x) // Devuelve el valor absoluto
- map(value, fromLow, fromHigh, toLow, toHigh) //devuelve el valor de un “re-mapeo” de un rango hacia otro.
- Pow(base, exponente) /devuelve el valor de un número elevado a otro numero
- Sqrt(x) // Devuelve la raíz cuadrada

Anexo D. Versiones de Android.

Android 1.5 *Cupcake*

- Mejoras notorias en la velocidad de la cámara.
- Disminuyeron los tiempos de búsqueda de satélites GPS.
- Se incluyó un teclado en la pantalla
- Fueron rediseñados todos los elementos de la interfaz gráfica.

Android 1.6 *Donut*

- Conexión a redes VPN.
- Motor de texto a voz.
- Se incluyó una nueva pantalla para el control de la batería.

Android 2.0/2.1 *Eclair*

- Se rediseño la interfaz del navegador.
- Bluetooth 2.1.
- Mejoras notorias en el Google Maps.
- Mejoras en la duración de la batería.

- Soporte flash de la cámara.

Android 2.2 *Froyo*

- Actualizaciones automáticas para las aplicaciones.
- Soporte WiFi IEEE 802.11n.
- Soporte API grafica OpenGL.
- Se creó el Android market.

Android 2.3 *Gingerbread*

- Refinamientos en la interfaz del usuario.
- Se optimizo el teclado táctil.
- Soporte para reproducción de videos web.
- Soporte nativo para la telefonía VoIP SIP
- Administrador de aplicaciones.

Android 3.0/3.1/3.2 *Honeycomb*

- Gran avance en tablets.
- Escritorios 3D con widgets rediseñados.
- Mejoras en el soporte de redes WiFi.
- Soporte para periféricos de salida.

Android 4.0 *IceCreamSandwich*

- Soporte para lápiz táctil Stylus.
- Reconocimiento facial.
- Reconocimiento de voz de usuario.
- Android Beam para compartir contenido entre teléfonos.
- Mejoras notables en la creación de directorios.

Android 4.1 *JellyBean*

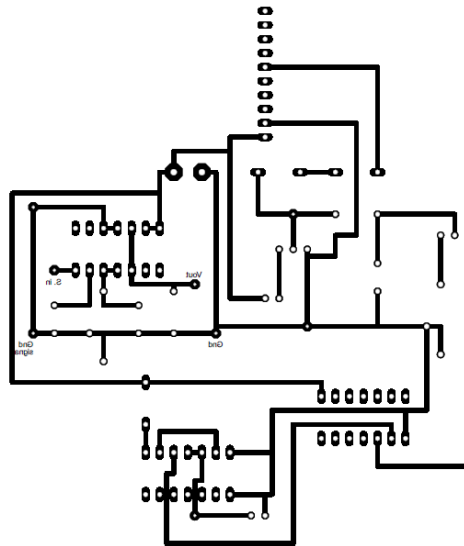
- Transferencia de datos Bluetooth para Android Beam.
- Se implementaron accesos directos y widgets que se reorganizan automáticamente para permitir nuevos ítems en la pantalla de inicio.
- Dictado de voz offline.
- El navegador de serie es reemplazado por la versión móvil de Google Chrome.
- Audio multicanal.

Android 4.2 *JellyBean*

- Se agregó el software para agregar gamepads y joysticks como dispositivos HID.

- Se mejoró notablemente el rendimiento.
- Ahora se pueden tener varias cuentas de usuario (solo para tablets).
- Soporte para pantallas inalámbricas.
- Teclado con escritura gestual.⁴⁵

Anexo E. Diseño de la placa.



Anexo F. Montaje de los indicadores.



⁴⁵ Colaboradores de Wikipedia, Historial de versiones de Android. 2013. Wikipedia la enciclopedia libre (fecha de consulta 16 de Julio de 2013). Disponible en: <https://es.wikipedia.org>.

Sistema de caja negra para monitoreo y registro de automotores tipo buseta

Black box system for monitoring and registration of bus type vehicles

Alexander Gordillo¹, José Felipe Zambrano² y Mauricio Fernando Gutiérrez³

Resumen

En este artículo se da a conocer el proyecto que tiene como base tomar información en tiempo real de una serie de parámetros los cuales serán monitoreados por sensores y esta información será enviada vía internet a un host desde el cual podrá ser supervisado por un ente privado. Es necesario implementar una cámara para monitorear y registrar lo que pasa en la cabina del conductor, con el fin de determinar posibles causas de culpabilidad en caso de accidentes de tránsito o irregularidades en la prestación del servicio además de facilitar alguna seguridad tanto al conductor como a los usuarios.

El proyecto se ha dividido en las siguientes etapas: la digitalización de todas las salidas análogas de los distintos sensores, el procesamiento de todas las salidas digitalizadas y su transformación en valores contables y de fácil interpretación, el envío de estos datos ya procesados a un dispositivo Android por cable USB y de este a un host donde se guardan y se disponen para la visualización de estos datos a través de un portal web en tiempo real.

Palabras clave: Monitoreo vehículos; Monitoreo buses; GPS; Arduino; Software Libre; Seguridad vial; Seguridad Vehículos; Mysql; PHP; Android; Registro tiempo real; Open Source; SMYRT.

Abstract

This paper discloses the project which is based on real-time information to take a number of parameters which will be monitored by sensors, and this information will be sent via Internet to a host from which will be supervised by a private entity. You must implement a camera to monitor and record what happens in the driver's cab, in order to determine possible causes of guilt in case of traffic accidents or irregularities in service delivery as well as providing some security for both the driver and users.

The project has been divided into the following stages: the digitalization of all the analog outputs of the various sensors, processing of all outputs digitized and its transformation into accounting and easy to interpret, the sending of the data already processed to a device Android by USB cable and from this to a host where it is stored and is available to visualize this data through a web portal in real time.

Keywords: Vehicle Monitoring, Monitoring buses, GPS, Arduino, Free Software, Road safety, Security Vehicles, Mysql, PHP, Android, Real time recording; Open Source; SMYRT.

¹ Ingeniero Electrónico. Universidad Surcolombiana-Neiva. Av. Pastrana-Carrera 1. gorlexx@hotmail.com

² Ingeniero Electrónico. Universidad Surcolombiana-Neiva. Av. Pastrana-Carrera 1. plpe007@hotmail.com

³ Ingeniero Electrónico. Universidad Surcolombiana-Neiva. Av. Pastrana-Carrera 1. mlcho@hotmail.com

1. Introducción

Al iniciar el proyecto se buscó un dispositivo que pudiera solucionar los problemas más mediatos que padece el transporte urbano, en especial los automotores tipo buseta, que además de los beneficios que potencialmente podría traer a las empresas de transporte público, también, pudiera constituir un servicio de prevención de accidentes viales que, junto a una plataforma tecnológica y una interfaz amigable con el usuario podría prevenir en tiempo real, posibles accidentes monitoreando la velocidad del vehículo, combustible, además de videograbaciones durante los turnos de las diferentes rutas, que en caso de posibles accidentes podrían corregir un camino para evitar futuros imprevistos.

Por lo anterior se identificaron tres problemáticas fundamentales a las que se les pretende dar solución:

- a. La existencia de un sistema de transporte público, en el que los buses se quedan estáticos a pesar de que la luz del semáforo este en verde; no respetan las señales de tránsito, golpean vehículos a diario y ponen en riesgo los usuarios del servicio y peatones. Este problema, está asociado a una negligencia de los conductores de buses y no existe un registro físico que pueda determinar la culpabilidad en un accidente de tránsito.
- b. Al no haber un control del seguimiento de las rutas, los autobuses toman diferentes vías que no son permitidas por la ley, también el modo de manejar de algunos conductores no es el más apropiado porque exceden los límites de velocidad permitidos dentro de la ciudad. A pesar de los controles impuestos por las diferentes empresas de transporte público, estos no son suficientes.
- c. En la ciudad de Neiva, el vigente método de registro de puntos viales de los diferentes automotores de servicio público se hace manualmente por medio de operadores, lo que produce una baja fiabilidad de la información. Esta técnica pudo haber funcionado para el momento en el que existían pocos automotores de servicio público, pero con el aumento de la población se hizo necesario incrementar la cantidad de automotores y mejorar la calidad del servicio, esto acarrea un aumento en la cantidad de la información que se debe registrar y sistematizar.

Se tomaron como referente el caso de España donde se encuentra ubicada Indra, la empresa líder en reformas tecnológicas en el campo del transporte (Instituto español de Comercio Exterior ICEX, 2012). Así mismo Indra en medio de su proceso de expansión, el pasado mes de agosto firmó un contrato con Medellín para instalar una red más eficiente y moderna (INDRA, 2012), y Estados Unidos, que en el momento son líderes en el control de transporte público, entendiéndose por control, el manejo del tiempo en el que se realiza una ruta de transporte, la especificación de dicha ruta y la forma en que efectivamente son transportados los pasajeros, es decir las aptitudes de los empleados de estas empresas al manejar dichos vehículos (Negocio tecnológico, 2012).

Este trabajo, no solo aporta un sistema totalmente sofisticado y de alta tecnología como los de otros países a nivel mundial sino que realiza un enfoque en lograr una mayor seguridad vial. Todas las grandes tecnologías a lo largo de la historia no solo buscan comodidad y simplicidad a la hora de afrontar alguna tarea sino que también busca proteger la vida humana en todas sus formas, la actividad de conducción es considerada como de alto riesgo debido a que en el desenvolvimiento de esta se generan innumerables accidentes que involucran a millares de vidas humanas todos los días, para demostrar tal afirmación citaremos a continuación las estadísticas arrojadas por el estudio del Plan Nacional de Seguridad Vial en Colombia 2011 a 2016, en lo referente a las pérdidas humanas generadas por la conducción de Vehículos de transporte público:

- En Colombia, en general, por cada 100mil habitantes el 12.3 mueren a causas de accidentes de tránsito, En el 2010, esta estadística llegó a su máximo representando el 22% de las muertes en el país.
- En nuestro departamento, considerado dentro del ámbito de “otros”, los accidentes terrestres provocaron el 35% de muertes violentas.

- A su vez, en términos de vidas cuantificables, y aunque en menor escala que vehículos como motocicletas, se contó con 338 muertes para el 2010 a manos del servicio público.
- A causa del transporte público a cargo de vehículos tipo buseta generaron el 9% de las muertes en accidentes de tránsito en el país, de las cuales el 79% de estas suceden en un entorno urbano, como es el caso de nuestra ciudad (Ministerio de Transporte, 2012).

De acuerdo a lo anterior se quiso desarrollar un dispositivo que fuera no solo novedoso, sino además útil y necesario, a través de la utilización de software libre el cual permite una documentación amplia y de fácil acceso.

A nivel institucional, se han propuesto en años anteriores proyectos con características y propósitos similares, los que aquí contemplamos, los cuales incluyen implementaciones en el servicio de transporte consistente en las siguientes características:

- Una interface de usuario conectada a un equipo transreceptor, el cual se instala en la cabina de cada vehículo.
- Un módulo transmisor (control) que le proporcione la ubicación temporal al vehículo; esta ubicación es transmitida desde el vehículo hacia la central cuando esta lo indique.
- “Un equipo transreceptor que se encarga de hacer el monitoreo de los vehículos y a su vez enviar los datos adquiridos al computador.” (Pastrana *et al*, 2005)
- “Un sistema de seguimiento y rastreo de un vehículo del servicio público conformado por un hardware y un software que permiten llevar la información desde el receptor GPS hasta un computador, a través de una página W.A.P. (protocolo de aplicaciones inalámbricas) usando la tecnología G.P.R.S. (Servicio general de paquetes vía radio), mediante el protocolo N.M.E.A. (Asociación nacional de electrónica marina) y utilizando comunicación serial”. (Morales *et al*, 2011)

En el siguiente trabajo, se plantearan las principales definiciones de los distintos equipos, tanto software como hardware, que nos ayudaron a implementar el dispositivo, así como los pasos que se deben seguir para tal fin.

2. Metodología

2.1 Lectura de variables

Para este trabajo se tomaron como base tres variables sensadas: *Velocidad (V)*, *Combustible(C)* y *Choque (G)*, para luego ser digitalizadas. La variable *Velocidad* tuvo un proceso de acondicionamiento, en el que fue convertida de variación de frecuencia a valores de voltaje, a la variable *Choque* por medio de un acelerómetro se le aplicó un comparador para tomar el valor más alto en las diferentes variaciones de aceleración y así asignar un único valor booleano a la variable y finalmente la variable *Combustible* debido a su linealidad parcial se realizó un proceso de acondicionamiento. Todo mantiene una estructura de adquisición como vemos en la figura 1.

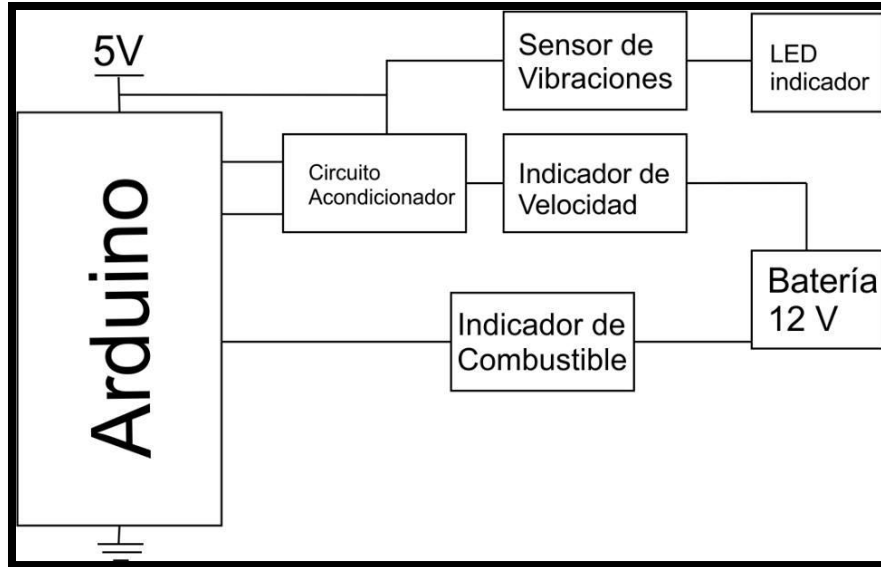


Fig. 1. Bloque de lectura de variables

2.2 Digitalización y procesamiento de variables con Arduino

Arduino es una plataforma de hardware libre (OSHW) que se basa en una placa con un microcontrolador y un entorno de desarrollo.

El microcontrolador usado en la placa es un Atmel AVR y varios puertos I/O, tanto análogos como digitales, así como salidas PWM y de comunicación, los cuales controlan objetos físicos (Arduino Team, 2013).

Las variables V, C y G anteriormente convertidas en impulsos eléctricos llegan a la tarjeta Arduino quien es la encargada de realizar todas las operaciones necesarias para transformar estos impulsos eléctricos en valores numéricos, que permiten identificar el nivel de combustible y el kilometraje del vehículo.

Debido a que la variable C no es lineal, se delimitaron unas condiciones que le permiten a la tarjeta Arduino linealizar los rangos del nivel de combustible, de tal forma que 0 es vacío y 100 lleno.

2.3 Transmisión Arduino-Android

Android es un sistema operativo basado en Linux con la interfaz de programación Java. Android SDK (Software Development Kit) provee todas las herramientas necesarias para desarrollar aplicaciones Android. El kit incluye un compilador, un emulador de dispositivo (en caso de no tener alguno) que sirve para correr todos los programas Android (Báez *et al*, 2010).

En este punto la tarjeta Arduino transmite los datos de las variables al dispositivo Android vía USB, con el fin de garantizar la estabilidad de la conexión y por ende la fiabilidad de los datos recibidos por el dispositivo como vemos en la figura 2.

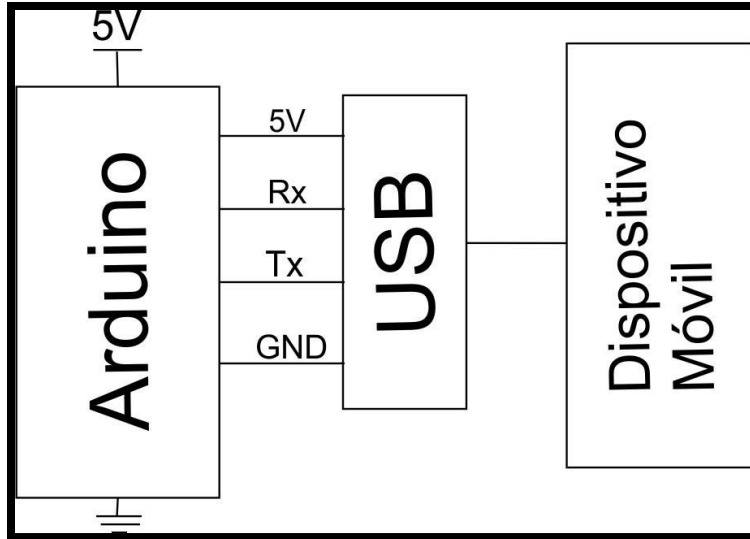


Fig. 2. Bloques transmisión Arduino-Android

2.4 Procesos en el Android

El dispositivo Android recibe las variables V, C y G convertidas en datos compresibles, a su vez el GPS se encargara de dar las coordenadas de latitud y longitud que permitirán establecer la ubicación del vehículo.

Con los datos anteriores se forman paquetes a los que aparte de la información sobre velocidad, el combustible y ubicación del vehículo se les agrega la hora que se obtendrá del dispositivo Android.

2.5 Transmisión Android-servidor

El dispositivo Android toma los paquetes de datos anteriormente mencionados y los envía a internet o a un servidor local a través del protocolo JSON (Camacho, 2007) (ver figura 3), el cual permite el intercambio de datos en internet fijando un url de un servidor determinado.

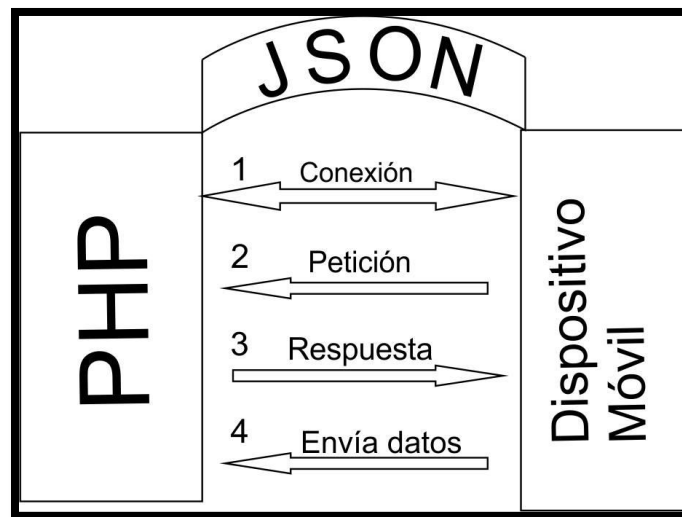


Fig. 3. Transmisión Android-PHP

2.6 Servidor

Para efectos de este trabajo se utilizó un servidor local por lo que se hizo necesaria la utilización de un paquete de software libre llamado XAMPP V.1.8.3, el cual instala en un computador los programas necesarios para que funcione como un servidor de manera local, dentro de los que se encuentran:

- Apache 2.4.4
- MySQL 5.6.11
- PHP 5.5.0
- phpMyAdmin 4.0.4
- FileZilla FTP Server 0.9.41
- Tomcat 7.0.41 (with mod_proxy_ajp as connector)
- Strawberry Perl 5.16.3.1 Portable
- XAMPP Control Panel 3.2.1 (Apache Friends, 2010)

De los cuales se usaron fundamentalmente:

- Apache 2.4.4: Es fue el encargado de simular el servidor web en el computador.
- MySQL 5.6.11: Permite la organización de la información a través de una base de datos.
- PHP 5.5.0: Acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, y es fácil de aprender. (Net. PHP, 2013)
- PhpMyAdmin 4.0.4: Es una interfaz para la gestión de la base de datos.

La aplicación de PHP a nuestro sistema se distribuye en bloques que contienen los diferentes códigos que trabajaran dentro del servidor, esa distribución se aprecia en la figura 4.

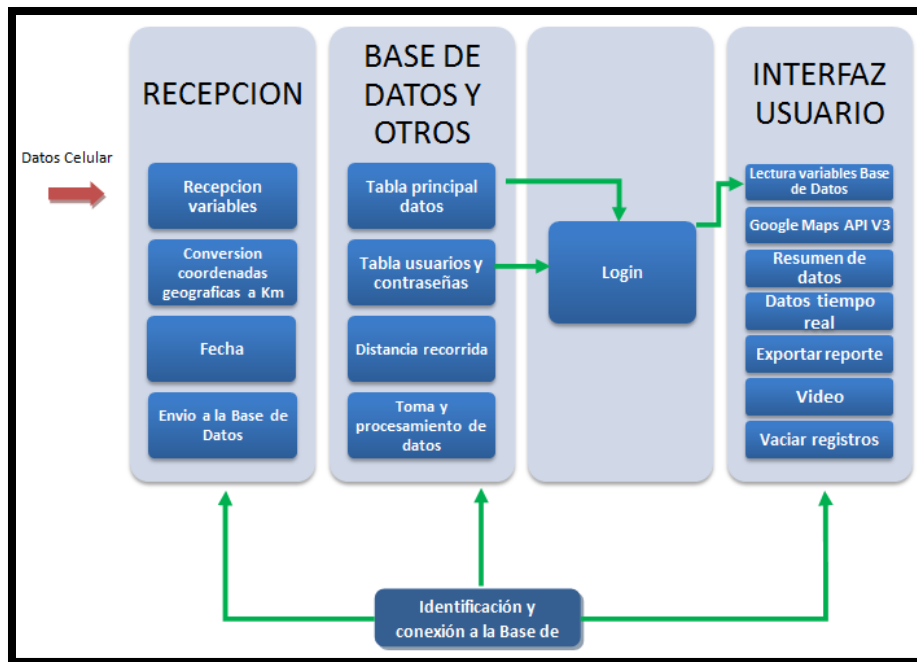


Fig. 4. Distribución de códigos en el servidor

2.6.1 Bloque recepción

Al llegar los paquetes de datos al servidor, este toma las variables y las organiza independientemente, transformando las coordenadas geográficas emitidas por el GPS a kilómetros y tomando la fecha actual del servidor, para finalmente enviar estos datos a la tabla principal en la base de datos.

2.6.2 Bloque base de datos y otros.

En una base de datos MySQL se diseñaron dos tablas, la primera es considerada la principal porque guarda todos los datos recibidos del bloque anterior en 9 columnas y la segunda tabla de la base de datos contiene los usuarios y contraseñas registrados para el acceso a la página.

Además de esto Existen cinco programas independientes que no están ligados a ningún bloque en especial, pero cumplen funciones específicas de seguridad y estabilidad de la interfaz.

- Identificación a la base de datos: Guarda la información de usuario y contraseña de la base de datos. Es esencial aislar esta información de los otros programas ya que por razones de seguridad estos datos no deben ser accesibles desde la web.
- Conexión a la base de datos: Crea la conexión a la base de datos con la información suministrada en el programa anterior, lo cual agiliza las consultas y funciones utilizadas en todas las otras aplicaciones en las que interviene.
- Distancia recorrida: Es una sub-rutina la cual da como respuesta la distancia total recorrida por el vehículo.
- Procesamiento de datos: Se analizan los datos guardados en la base de datos para hallar el combustible consumido, distancia recorrida, la cantidad de veces que sobre paso la velocidad limite que son 60Km/H en zonas urbanas y tiempo transcurrido. Esta información es enviada al siguiente bloque para su visualización. (Código de tránsito, 2013)
- Login: Hace la identificación del usuario para ingresar a la interfaz principal comparando los datos asociados a la tabla de usuarios en la base de datos.

2.6.3 Bloque de interfaz

Para el usuario final se desarrolló un portal web que dispone de la visualización de los datos recopilados en el transcurso de los bloques anteriores.

- Se realiza una lectura de datos a través de los archivos de identificación y conexión a la base de datos mencionados en el apartado anterior. Desde la base de datos se toma el último dato guardado de la columna de latitud y longitud para usarlas dentro del código de Google Maps.
- Datos en tiempo real: También se insertan los archivos del bloque procesamiento de datos para su visualización en una tabla que sintetiza la jornada de trabajo.
- Exportar reporte: El código se inicia con un botón que nos permite ver toda la información de la base de datos que estén en el momento en un archivo de formato .PDF para mantener y guardar el registro de trabajo en cualquier momento del día.
- Vaciar registros: Esta parte de código hace que se borren todos los datos almacenados en la tabla principal de la base de datos, esta función se encuentra implementada en un botón.
- Video: Para el video se implementó un banner con el link de la dirección de la cámara

2.7 Cámara IP

El vehículo contara con una cámara IP Foscam FI18918W (ver figura 5) que estará conectada inalámbricamente al router desde el cual el usuario podrá supervisar el recorrido en tiempo real y mantener una comunicación bilateral de audio. La cámara tiene un DNS gratuito con lo que nos permite acceder a ella desde internet con cualquier navegador.



Fig. 5. Cámara IP Foscam FI18918W

2.8 Portal Web

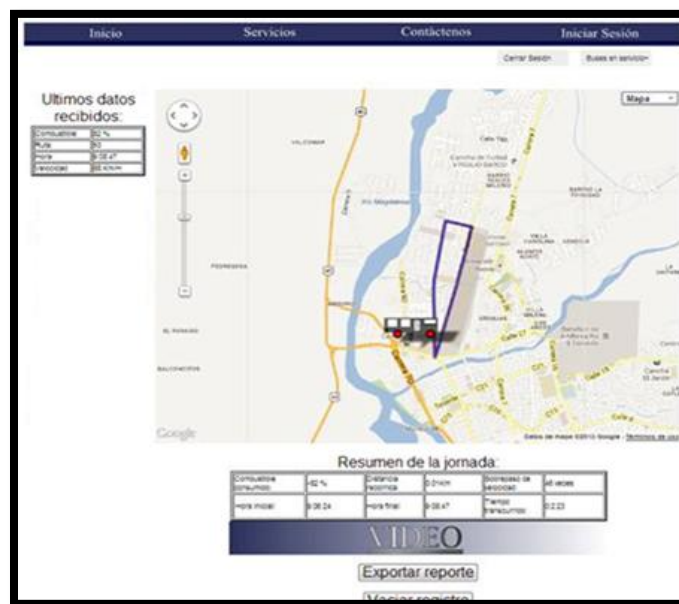


Fig. 6. Captura portal web

La figura 6 muestra el portal web que contiene el mapa con una ruta predefinida, la posición actual del vehículo, los últimos datos recibidos y el resumen de la jornada laboral.

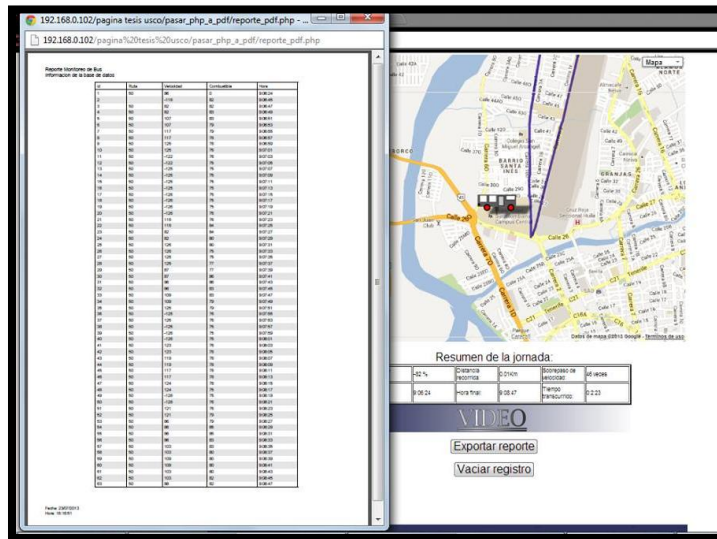


Fig. 7. Captura reporte

Por medio del uso del botón “exportar reporte”, se crea un documento PDF como vemos en la figura 7 el cual contiene un registro de los valores más relevantes; para reiniciar este registro, se utiliza el botón de “vaciar registro”.

3. Limitaciones

Existe un error en la creación del reporte al exceder 250 valores en la base de datos, también existe un pequeño error de GPS debido a condiciones climáticas y a la precisión del sensor.

La digitalización de la velocidad y el combustible puede variar, la resistencia interna del flotador y el tren de pulsos pueden no ser los mismos en diferentes vehículos.

4. Conclusiones

Se logró diseñar, simular y probar un sistema de monitoreo y registro en tiempo real de las variables aceleración, combustible, velocidad, fecha, hora, posición, kilometraje y choque integrados en una caja. Este dispositivo permite a las empresas dar un mayor control de sus vehículos y conductores. SMYRT es un sistema poco invasivo, de fácil instalación y mantenimiento.

Todo el sistema se basó en software y hardware libre lo que nos permitió reducir costos y tener un mayor acceso a la información para desarrolladores. Esta tecnología sirve como medio para prevención de futuros accidentes ya que sus características ayudan a determinar las fallas en cuanto a la seguridad en la prestación del servicio.

Se cumplió un acondicionamiento del sensor de velocidad, ya que entregan un tren de pulsos que aumentaba su frecuencia al aumentar la velocidad para ello se utilizó un circuito conversor de frecuencia a voltaje. La tarjeta Arduino efectuó el proceso de digitalizar las diferentes señales analógicas de los indicadores y linealizar la variable del medidor de combustible.

El entorno de desarrollo Arduino es agradable, fácil de entender y por ser libre ofrece varias ventajas, entre ellas que los usuarios avanzados compartan sus librerías y conocimientos de desarrollo de manera educativa.

Fue necesario el uso de la versión 2.3.7 de Android (*Gingerbread*) porque es la más comercial, además de ser compatible con los sistemas Arduino. No se garantiza el funcionamiento en otras versiones de Android.

Para la programación en eclipse fue utilizado el API 10 de Google ya que para el desarrollo generaba menos errores.

Se realizó una aplicación móvil en el sistema operativo Android para realizar la recepción de los valores anteriormente procesados de Arduino y la posterior transmisión a la base de datos del servidor.

Existe un error al iniciar la aplicación debido a que la base de datos se encuentra vacía, se soluciona reconectando el dispositivo y reiniciando la aplicación.

La posición no solo se toma del sensor GPS sino también de internet y así el programa selecciona el valor de mayor precisión.

Se profundizó en los lenguajes de programación: Java, C++, C, PHP, Javascript, HTML, CCS y Arduino.

Optamos porque el sistema envíe datos cada 2 minutos para darle el tiempo al GPS de triangular su posición y obtener un valor preciso, se puede modificar este tiempo dependiendo de las necesidades del usuario.

Se diseñó un portal web con la información básica del proyecto y un acceso restringido para ingresar a la interfaz principal que contendrá todos los datos monitoreados además se acondicionó una cámara IP para una transmisión de video y audio.

Logramos exportar los valores de la base de datos en un documento PDF para tener un registro de lo sucedido en la jornada.

Hay una optimización en el control actual de los autobuses ya que pasan de solo 2 controles a 30 por ruta (ruta de prueba 1 hora 30 min).

5. Referencias Bibliográficas

Negocio tecnológico, 2012. Nuevos autobuses europeos para un transporte público eficiente.. Consultado el 30 de abril del 2013. <http://www.negociotecnologico.com>

INDRA, 2012. INDRA implanta su tecnología inteligente para la gestión del tráfico y el transporte público de Medellín por 9 m€.. Consultado 14 de junio del 2013. Extraído de <http://www.indracompany.com>

Instituto español de Comercio Exterior (ICEX), 2012. Sistemas Inteligentes de Transporte en EEUU, Consultado el 18 de octubre 2012. www.navactiva.com

Ministerio de Transporte. Plan de Seguridad Vial 2011-2016, 2012. 72p. Consultado el 1 de julio del 2013. <https://www.mintransporte.gov.co/descargar.php?idFile=8>

Pastrana J.; Polanco C., 2005. Diseño e implementación de un sistema de monitoreo por radiofrecuencia para el registro de horarios de buses de transporte. Tesis de grado. Universidad Surcolombiana. Facultad de ingeniería.

Morales, L.; Trujillo, O., 2011. MONITOREO DE VEHICULOS DE TRANSPORTE DE SERVICIO PÚBLICO. Tesis de grado. Universidad Surcolombiana, Facultad de Ingeniería.

Arduino Team, 2013. Arduino Homepage. Consultado el 1 de julio de 2013. <http://www.arduino.cc/>.

Báez, M.; Zapata, A., 2000. Introducción a Android, ISBN: 978-84-96285-39-5. Segunda edición E.M.E. editorial. Madrid, España. 121 p.

Camacho, J. 2007. POST y GET. Consultado el 02 de junio de 2013 [http://www.svcommunity.org/forum/web/\\$_post-y-\\$_get-bien-explicados/](http://www.svcommunity.org/forum/web/$_post-y-$_get-bien-explicados/)

Apache friends, 2010. XAMPP Control Panel 3.2.1. Consultado el 3 de julio de 2013 <http://www.apachefriends.org>.

Net. PHP, 2013. Manual PHP. Consultado el 15 de julio de 2013 <http://www.php.net/>

Código de tránsito, 2013. Consultado el 1 de julio de 2013 <http://www.colombia.com/noticias/codigotransito/t3c11.asp>