



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

**CARRERA DE INGENIERÍA EN ELECTRONICA Y REDES DE
COMUNICACIÓN**

**PLAN DE NEGOCIOS PARA EL SISTEMA DE CONTEO DE
PASAJEROS DE LA EMPRESA INPRISE**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA
EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

BURBANO ROBLES SANTIAGO MANUEL

Director:

ROSERO PAÚL, ING

Ibarra – Ecuador

2014

CERTIFICACIÓN

Certifico que la Tesis "**PLAN DE NEGOCIOS PARA EL SISTEMA DE CONTEO DE PASAJEROS DE LA EMPRESA INPRISE**" ha sido realizada en su totalidad por el señor: **SANTIAGO MANUEL BURBANO ROBLES** portador de la cédula de identidad número: **1003073358**

A handwritten signature in blue ink, appearing to read 'Paul Rosero', is written over a horizontal dotted line. The signature is stylized and somewhat illegible due to the cursive nature of the writing.

Ing. Paul Rosero
Director de la Tesis

CERTIFICACIÓN

Ibarra, 14 de febrero de 2014
Señores
UNIVERSIDAD TÉCNICA DEL NORTE
Presente

De mis consideraciones.-

Siendo auspiciantes del proyecto de tesis del Egresado **SANTIAGO MANUEL BURBANO ROBLES** con CI: 1003073358 quien desarrolló su trabajo con el tema "**PLAN DE NEGOCIOS PARA EL SISTEMA DE CONTEO DE PASAJEROS DE LA EMPRESA INPRISE**", me es grato informar que se han superado con satisfacción las pruebas técnicas y la revisión de cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte del egresado **SANTIAGO MANUEL BURBANO ROBLES**. Una vez que hemos recibido la capacitación y documentación respectiva, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra empresa.

El egresado **SANTIAGO MANUEL BURBANO ROBLES** puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte

Atentamente,



LUIS BURBANO

Gerente

INPRISE





UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, SANTIAGO MANUEL BURBANO ROBLES, con cédula de identidad Nro. 1003073358, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **"PLAN DE NEGOCIOS PARA EL SISTEMA DE CONTEO DE PASAJEROS DE LA EMPRESA INPRISE"**, que ha sido desarrollada para optar por el título de Ingeniería en Electrónica y Redes de Comunicación, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.

Firma

Nombre: SANTIAGO MANUEL BURBANO ROBLES

Cédula: 1003073358

Ibarra a los 14 días del mes de Febrero del 2014



**UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR
DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad. Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente investigación:

	DATOS DE CONTACTO
CÉDULA DE IDENTIDAD	1003073358
APELLIDOS Y NOMBRES	SANTIAGO MANUEL BURBANO ROBLES
DIRECCIÓN	Ciudadela del Consejo Provincial Casa #9
EMAIL	santimbr@gmail.com
TELÉFONO FIJO	2651420
TELÉFONO MOVIL	0989589285

	DATOS DE LA OBRA
TÍTULO	"PLAN DE NEGOCIOS PARA EL SISTEMA DE CONTEO DE PASAJEROS DE LA EMPRESA INPRISE"
AUTOR	BURBANO ROBLES SANTIAGO MANUEL
FECHA	14 DE FEBRERO DEL 2014
PROGRAMA	PREGRADO
TÍTULO POR EL QUE	INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN
DIRECTOR	ING. PAÚL ROSERO

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, SANTIAGO MANUEL BURBANO ROBLES, con cédula de identidad Nro. 1003073358, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 143.

Firma

Nombre: SANTIAGO MANUEL BURBANO ROBLES
Cédula: 1003073358
Ibarra a los 14 días del mes de febrero del 2014

DEDICATORIA

El presente trabajo está dedicado a mis padres Maribel y Manuel por toda la entrega y dedicación puestas en mi formación como persona, siendo ellos el pilar fundamental en mi carrera universitaria.

AGRADECIMIENTO

Un agradecimiento especial a la empresa INPRISE por abrirme las puertas para la realización de este trabajo de titulación y su financiamiento para el mismo.

Agradecer a toda mi familia por el apoyo incondicional durante este periodo universitario.

A la Universidad Técnica del Norte por ser el ente de formación de mi carrera profesional.

Y a mis amigos y compañeros por ser parte de mi vida durante el periodo universitario.

Tabla de contenido

ANTECEDENTES	1
1.1. Problema	1
1.2. Objetivos	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	2
1.3. ALCANCE	3
1.4. Justificación	4
MARCO TEÓRICO	6
2.1. Plan de negocios	6
2.1.1. Definición de Producto	6
2.1.2. Desarrollo del Plan de Negocios	8
2.2. Sistemas Microcontrolados	10
2.2.1. Microcontroladores PIC	10
2.2.2. ARDUINO	11
2.3. Comunicación Serial (RS-232)	13
2.3.1. Velocidad de Transmisión (BAUD RATE)	15
2.4. Tratamiento de Señales Seriales	15
2.5. Programación en Sistemas Electrónicos	16
2.5.1. Lenguaje Máquina	17
2.5.2. Lenguajes de Bajo Nivel	18
2.5.3. Lenguaje de Alto Nivel	18
2.6. Sensores	19
2.6.1. Sensores de Proximidad	21
2.7. Tecnología GPRS – General Packet Radio Service	23
2.7.1. Arquitectura GPRS (España, 2003)	24
2.7.2. Funcionamiento GPRS	29
2.8. Tecnología GPS	30
2.8.1. Sistema de Posicionamiento Global GPS	30
2.8.2. Componentes del Sistema de Posicionamiento Global	31
2.8.3. FUNCIONAMIENTO GPS	35
2.9. Transmisión de Información en un Sistema GPS	37
2.9.1. Protocolo NMEA 0183	37
2.9.2. Trama GPRMC	38

2.10.	Comandos HAYES - Comandos AT	39
2.11.	Protocolo de Control de Transmisión TCP	40
2.11.1.	Formato de un Segmento TCP	41
2.12.	Socket de Comunicación.....	42
2.12.1.	Protocolo de Comunicación.....	43
2.12.2.	Dirección de Protocolo (DIRECCIÓN IP) y Número de Puerto	44
2.13.	Base de Datos	45
2.13.1.	POSTGRESQL	47
2.14.	Lenguaje de Programación PYTHON.....	49
2.14.1.	Lenguaje Interpretado o de Script.....	49
2.14.2.	SOCKET EN PYTHON.....	50
PLAN DE NEGOCIOS PARA EL SISTEMA DE CONTEO DE PASAJEROS Y ENVIÓ DE INFORMACIÓN MEDIANTE TECNOLOGÍA GPRS DE LA EMPRESA INPRISE.....		55
3.1.	Oportunidades del Negocio	55
3.1.1.	Análisis del Sector	55
3.1.2.	Análisis de la Industria.....	58
3.1.3.	Análisis De Factores	59
3.2.	Análisis del Mercado	60
3.2.1.	Diseño de la Investigación	60
3.3.	Realización de la Investigación de Mercados.....	62
3.3.1.	Identificación de la población	62
3.3.2.	Cálculo de la muestra.....	63
3.3.3.	Desarrollo de la encuesta	64
3.3.4.	Análisis de la Demanda.....	69
3.4.	Plan de Marketing.....	71
3.4.1.	Análisis de la situación	71
3.4.2.	Datos del mercado.....	71
3.4.3.	Estrategia de Marketing Local	74
3.5.	Plan de Fabricación y Operación	75
3.5.1.	EQUIPOS O HERRAMIENTAS A SER UTILIZADAS	77
3.5.2.	Materia prima para la construcción.....	78
3.5.3.	Número de personas para la producción	78
3.5.4.	Tareas del personal para producción	78

3.6.	Costos de Producción y Financiamiento.....	79
3.6.1.	Proyección del Precio de Venta.....	79
3.6.2.	Financiamiento.....	82
DISEÑO.....		84
4.1.	Descripción del Dispositivo.....	84
4.2.	Contador de Pasajeros.....	87
4.2.1.	Sensor SHARP modelo GP2Y0A21YK.....	87
4.2.2.	Microcontrolador PIC 16F88.....	89
4.2.3.	Conexión Sensores al Microcontrolador.....	91
4.2.4.	Conexión de Componentes.....	92
4.2.5.	Envío de Información.....	93
4.2.6.	Programación del Microcontrolador 16F88.....	94
4.3.	Controlador.....	97
4.3.1.	ARDUINO MEGA2560.....	97
4.3.2.	Conexión Eléctrica.....	99
4.3.3.	SHIELD para la placa ARDUINO Mega.....	100
4.3.4.	Tratamiento de la Información.....	104
4.3.5.	Programación de ARDUINO.....	107
4.4.	Módulo GPS / GPRMS.....	110
4.4.1.	SKYPATROL TT8750+.....	110
4.4.2.	Conexión.....	114
4.4.3.	Configuración.....	115
4.5.	Servidor.....	117
4.6.	Recepción de Información en Socket.....	118
4.7.	Almacenamiento de Información en la Base de Datos.....	120
PRUEBAS Y RESULTADOS.....		123
5.1.	Pruebas.....	123
5.1.1.	Contador.....	123
5.1.2.	Controlador.....	124
5.1.3.	Módulo SKYPATROL.....	125
5.1.4.	Representación de la información.....	125
5.2.	RESULTADOS.....	126
5.2.1.	ENSAMBLAJE.....	126

5.2.2. Representación en el Software QGIS.....	128
CONCLUSIONES.....	130
RECOMENDACIONES.....	132
REFERENCIAS BIBLIOGRAFICAS.....	133
ANEXOS.....	135

Índice de Figuras

Figura. 1 Microcontroladores PIC	11
Figura. 2 Ejemplos placas ARDUINO.....	12
Figura. 3 Instrucciones desde el programador hacia el microcontrolador	17
Figura. 4. Proceso de compilación de lenguaje de alto nivel a lenguaje máquina....	19
Figura. 5. Ejemplo de sensores	20
Figura. 6 Arquitectura GPRS.....	24
Figura. 7. Constelación de satélites	32
Figura. 8. Ejemplo de triangulación hacia el punto B	35
Figura. 9. Ejemplos de códigos pseudo-aleatorios	36
Figura. 10. Formato trama TCP	41
Figura. 11 Componentes PostgreSQL.....	47
Figura. 12 Código para crear un objeto	50
Figura. 13 Asignación de IP y puerto.....	51
Figura. 14 Código para aceptar conexiones.....	52
Figura. 15 Código para enviar y recibir por el socket	52
Figura. 16 Software QGIS	54
Figura. 17 Bloques de Funcionamiento	85
Figura. 18 Representación del conteo lógico.....	86
Figura. 19 Relación voltaje-distancia sensor SHARP	88
Figura. 20 Diagrama de pines.....	90
Figura. 21 Conexión de sensores SHARP al microcontrolador.....	91
Figura. 22 Conexión de componentes	92
Figura. 23 Conexión USART entre PIC y ARDUINO	93
Figura. 24. Flujograma principal	96
Figura. 25 Conexión alimentación ARDUINO	100
Figura. 26 Diseño PBC del controlador para ARDUINO.....	101
Figura. 27 Configuración max3232.....	102
Figura. 28 Conexión MAX3232 con Skypatrol.....	102
Figura. 29 Circuito de pulso de activación	104
Figura. 30 Conexión serial 0.....	104
Figura. 31 Conexión serial 1 y 2.....	105
Figura. 32 Configuración Serial del Microcontrolador	105
Figura. 33 IDE para programación de sketches.....	108
Figura. 34 Flujograma del controlador	109
Figura. 35 Imagen Skypatrol	111
Figura. 36 Conexión del socket SKYPATROL.....	114
Figura. 37 Configuración módulo SKYPATROL.....	115
Figura. 38 Configuración para PIN de envío	116
Figura. 39 Recepción de tramas en el servidor.....	118
Figura. 40 Editor Geany y script en ejecución	120
Figura. 41 Creación de campos en la tabla contador	121
Figura. 42 Panel de navegación de PgAdmin	122
Figura. 43 Datos almacenados en Postgres.....	122

Figura. 44 Pruebas de funcionamiento del contador.....	123
Figura. 45 Pruebas del controlador	124
Figura. 46 Funcionamiento del SKYPATROL	125
Figura. 47 Filtrado de datos en QGIS	126
Figura. 48 Ensamblaje Final modelo 1	127
Figura. 49 Ensamblaje final modelo 2	128
Figura. 50 Representación de datos en QGIS	129

Índice de Tablas

Tabla 1. Parámetros estándar NMEA	37
Tabla 2. Descripción campos trama GPRMC	38
Tabla 3. Numero unidades asignadas a las cooperativas de transporte público en Ibarra.....	56
Tabla 4. Matriz de relación de la investigación de negocios	61
Tabla 5. Número de socios de las cooperativas de transporte	62
Tabla 6. Número de encuestados por cooperativa	64
Tabla 7. Demanda actual.....	70
Tabla 8. Equipos a ser utilizados	77
Tabla 9. Materia prima para la construcción.....	78
Tabla 10. Personal necesario para la producción.....	78
Tabla 11. Tareas del personal de producción	78
Tabla 12. Costos materia prima	80
Tabla 13. Costos de mano de obra	81
Tabla 14. Costos por servicios	81
Tabla 15. Costos de servidores y transferencia de datos.....	82
Tabla 16. Especificaciones sensor SHARP.....	88
Tabla 17. Características PIC 16F88.....	89
Tabla 18. Características Mega2560.....	97
Tabla 19. Campos de la trama desarrollada	107
Tabla 20. Características SKYPATROL TT8750+	111
Tabla 21. Pines de conexión SKYPATROL TT8750+.....	112
Tabla 22. Características servidor de aplicaciones	118
Tabla 23. Comparación Manual y Contador Electrónico	128

RESUMEN

El presente proyecto tiene como objetivo describir el proceso de ingeniería para la elaboración de un contador de pasajeros y envío de la información mediante tecnología GPRS.

Este estudio comprende dos partes fundamentales, la primera parte describe un plan de negocios sobre el producto, en el cual se detallan todos los procesos por el cual se llegó a determinar que el producto salga al mercado, un análisis de marketing para determinar la posición de la empresa en el mercado, un plan de fabricación que contiene los procesos a ser desarrollados y un análisis de costos el cual determinara el valor del producto.

La segunda parte consiste en el detalle teórico práctico sobre el diseño y fabricación del dispositivo, se toma en cuenta la fundamentación teórica de todos los componentes necesarios, se analiza los requerimientos a ser desarrollados en el dispositivo obtenidos mediante análisis con los futuros clientes y se describe paso a paso los procesos para su fabricación, se adjunta diagramas de construcción, códigos de configuración y resultados obtenidos.

SUMMARY

This project describe the engineering process for the development of a counter passenger and sent the information through GPRS technology.

This study comprises two main parts, the first part describes a business plan for the product, in which all the processes by which it came to determine that the product comes to market, marketing analysis are described for determining the position of the company in the market, a manufacturing plan that contains the process has to be developed and a cost analysis which will determine the value of the product.

The second part is the practical theoretical detail on the design and manufacture of the device , taking into account the theoretical basis of all the necessary components , the requirements to be developed in the device obtained by analysis with future customers is analyzed and described step by step processes for manufacturing, construction diagrams , setup codes and results are attached.

ANTECEDENTES

1.1.Problema

La empresa INPRISE Inc. de la ciudad de Ibarra esta dedica al desarrollo de soluciones en el campo del control y administración de procesos para transportes de pasajeros y carga, actualmente la empresa tiene instalado en un sector del transporte público de la ciudad de Ibarra un sistema de geo-posicionamiento que envía información sobre posición y tiempo vía GPRS¹.

La necesidad de desarrollar productos innovadores que cubran los requerimientos de los clientes como es el conteo de pasajeros que ingresan a su unidad y la posición del ingreso de los mismos, se hace necesario el desarrollo de un sistema de conteo de pasajeros el cual debe disponer de un sistema autónomo de conteo, el mismo que proporciona al propietario de vehículo un control contable de los ingresos en su unidad y a la cooperativa un análisis de demanda de paradas en sus diferentes rutas.

La empresa INPRISE hace necesario un plan de negocios que garantice que el proyecto ofrecido cumpla con los requerimientos de mercado y justifique la inversión ofrecida para el mismo.

¹GPRS. General Packet Radio Service. Servicio general de paquetes vía radio (transmisión de información por la red celular).

1.2.Objetivos

1.2.1. Objetivo General

Desarrollar un plan de negocios para la construcción y comercialización del sistema de conteo de pasajeros y envío de información vía tecnología GPRS para la empresa INPRISE Inc.

1.2.2. Objetivos Específicos

- Realizar una investigación de mercado que justifique la necesidad de desarrollar un producto desde el punto de vista de comercialización.
- Elaborar un plan administrativo y de marketing para la puesta en marcha del proyecto.
- Estudiar sistemas autónomos microcontrolados y envío-recepción de información mediante tecnología GPRS.
- Diseñar el plan de ingeniería para la construcción del dispositivo como un producto final.
- Elaborar un plan financiero para determinar los ingresos y costos de este proyecto.

1.3. ALCANCE

El presente trabajo requerirá el estudio de mercado en el sector del transporte de pasajeros de la ciudad de Ibarra el mismo que determinará las necesidades concretas de los propietarios de vehículos de esta clase. Se desarrollará un plan administrativo y de marketing en cual se determine los procesos que debe seguir la empresa para desarrollar el producto y a la vez comercializarlo.

Se describirá todo el proceso de ingeniería para la construcción del sistema de conteo de pasajeros, el cual debe contener todos los aspectos analizados y recogidos en el estudio de mercado. El sistema de conteo de pasajeros estará construido a base de sensores de presencia que determinan el conteo solo con el cruce del pasajero a una distancia máxima del dispositivo de 110cm (centímetros), este dispositivo discriminará si la persona entra o sale del vehículo; por disponer de sensores de presencia no es factible diferenciar si un pasajero es adulto o niño solo se podrá diferenciar pasajeros de diferentes alturas, esta medida se la obtendrá al hacer una media entre las alturas de los pasajeros.

El módulo de control y comunicación debe disponer de 2 comunicaciones seriales para la recepción y transmisión de la información del contador de pasajeros, el módulo determinará la ubicación del vehículo y solo tomará información cuando el vehículo se haya desplazado una distancia mínima después del último conteo de pasajeros.

Una vez adquirido los datos se construirá una trama de información la cual disponga del número de pasajeros contados, tiempo (hora y fecha) y la posición geográfica del vehículo, esta trama será enviada vía GPRS como un dato más del campo de datos de la trama TCP² disponible en el módulo de Geolocalización para su tratamiento en el servidor de aplicaciones ya existente. Finalmente se realizará un estudio financiero que determine la inversión y costos del mismo los cuales nos servirán para determinar la factibilidad del producto.

1.4. Justificación

La empresa INPRISE Inc. necesita automatizar todos sus procesos ya que de ellos depende mantenerse en el mercado del control de transporte de pasajeros, la nueva normativa en el país hace referencia a que todas las unidades de transporte público deben contar con sistemas inteligentes de conteo de pasajeros. Para que un proyecto sea fiable en términos económicos es necesario que se desarrolle un plan de negocios el mismo que justifique y sustente la inversión del mismo, demostrando su rentabilidad.

Con un sistema de conteo de pasajeros, más el envío de la información vía GPRS hace que una unidad de transporte sea más rentable y controlable, con el módulo de

²TCP Transmission Control Protocol (Protocolo para la transmisión de datos en Internet)

comunicación y control se gana prestación a los dispositivos a desarrollarse para que sean tratados en conjunto, y los resultados además de ser representados en el servidor de aplicaciones servirán para que se haga un análisis del tráfico de pasajeros en las rutas donde el vehículo transite.

MARCO TEÓRICO

1.1. Plan de negocios

Según la Corporación El Plan de Negocios es un documento que esquematiza de manera clara la información necesaria para conocer si un producto o la ampliación de la actividad productiva de una empresa va ser exitosa y rentable.

Uno de los errores frecuentes en estas iniciativas es no determinar resultados alcanzables. Es imprescindible identificar las fortalezas y las necesidades para no incurrir en el desperdicio de recursos. (Corporación Financiera Nacional, 2010)

1.1.1. Definición de Producto

A continuación se describe la definición de producto que fue tomado de “Manual Básico para Elaborar Plan de Negocio para PYMEs” del Centro de Exportaciones e Inversiones Nicaragua del año 2012.

1.1.1.1. Tipos de producto.

1.1.1.1.1. *Por su naturaleza:*

Producto Primario; son productos de consumo inmediato que no necesitan ninguna transformación, se pueden consumir en su estado natural. Ejemplo; frutas, verdura, leche, carne, etc.

Producto Secundario: son productos que necesitan ser transformado o que sufren modificación en su estado natural antes de ser consumido.

Producto Terciario o Servicios: son intangibles. Ejemplo; servicios de salud, educación, transporte comercialización, etc.

1.1.1.1.2. *Por su destino:*

Materia prima: son utilizados para ser transformado. Ejemplo; naranja para hacer jugo, mango para hacer mermelada.

Bienes de capital; son aquellos que con su utilización, sin cambiar, permiten la preparación de otro producto. Ejemplo; maquinarias y equipos.

Bienes intermedios: son aquellos que han sido transformado, pero que requieren nuevas transformaciones para ser consumidos. Ejemplo; pulpa de frutas.

Bienes finales: son bienes dirigidos a la satisfacción de necesidades personales. Ejemplo; vinos, conservas, etc.

1.1.1.1.3. *Por su duración:*

Se clasifican en productos perecederos y no perecederos.

1.1.1.2. Tipos de Servicios:**1.1.1.2.1. Servicios de preventa:**

Son aquellos que se relacionan con la información sobre los usos y características del producto ofrecido. Ejemplo; catálogos de información, entrega directa de información técnica.

1.1.1.2.2. Servicios durante la venta:

Son aquellos relacionados con entrega y puesta en operación del producto vendido. Ejemplo; despacho, servicio de instalación.

1.1.1.2.3. Servicios post ventas:

Son aquellos relacionados con las atenciones que se prestan después de haber realizado la venta. Ejemplo; mantenimiento de un equipo.

Si queremos que nuestro producto se distinga entre un conjunto y se requiere de su introducción al mercado.

1.1.2. Desarrollo del Plan de Negocios

Para realizar el plan de negocios primero se realiza un análisis del negocio, que oportunidades se tiene con la realización del proyecto, perspectivas del mercado local, competencia y factores industriales que determinen la ejecución del proyecto.

Seguidamente se realiza un análisis del mercado, en el cual se desea ingresar con el producto, analizar sus requerimientos, con este estudio se puede determinar si realmente el producto tendrá cabida en el determinado sector y si este puede ser construido o no.

Se debe redactar además un plan en el que conste la forma de comercialización del producto, además de analizar a la empresa tomando en cuenta sus fortalezas y debilidades hacia la competencia y las oportunidades y amenazas del sector productivo.

A continuación se desarrolla el plan de fabricación, el cual consiste en documentar todos los procesos y técnicas necesarias para el diseño, fabricación y producción del producto. Aquí se describe todo el proceso de ingeniería y técnico para que el producto pueda ser lanzado al construido.

Y finalmente se realiza un análisis de costo de producción del producto el cual ayudará a determinar el valor neto del producto y cuánto debe ser la inversión de la empresa para el desarrollo del producto

Esta tesis está basado en la construcción del producto que se presenta en el plan de negocios, describir la fundamentación teórica, diseño y construcción del dispositivo. Por lo cual se presenta a continuación toda la fundamentación teórica a nivel de diseño electrónico y funcional.

1.2.Sistemas Microcontrolados

Un sistema microcontrolado es aquel que está programado para realizar una o varias tareas de una forma sistemática. Estos dispositivos encapsulan en un solo dispositivo las 3 partes esencial para el procesamiento y ejecución de una aplicación, estos son la unidad de procesamiento, memoria y periféricos de entrada y salida. Los sistemas microcontrolados pueden desarrollarse en varias plataformas ya sea de software o de hardware. (USERS, 2014)

1.2.1. Microcontroladores PIC

El desarrollo de los microcontroladores nace en el año 1975 en una división de la compañía General Instruments la cual desarrolla un microcontrolador denominado PIC1650 pero con propósitos diferentes y su nombre completo fue PICmicro (Peripheral Interface Controller), después de varios avances y al añadir una memoria EEPROM este circuito se convirtió en lo que hoy conocemos como microcontrolador. (USERS, 2014)



Figura. 1 Microcontroladores PIC
Fuente: diaweb.usal.es/diaweb/index.jsp

En la actualidad la empresa norteamericana Microchip es líder en la fabricación y venta de microcontroladores y semiconductores analógicos, microchip dependiendo de la aplicación y los recursos a ser controlados ha desarrollado varias gamas de microcontroladores de 8, 16 y 32 bits a la vez estos deben ser elegidos según su disposición física y condiciones de trabajo, tomando siempre en cuenta la aplicación donde van a ser usados. (Microchip Technology Inc. All rights reserved, 2008-2014)

1.2.2. ARDUINO

ARDUINO es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos. (Arduino, 2014)

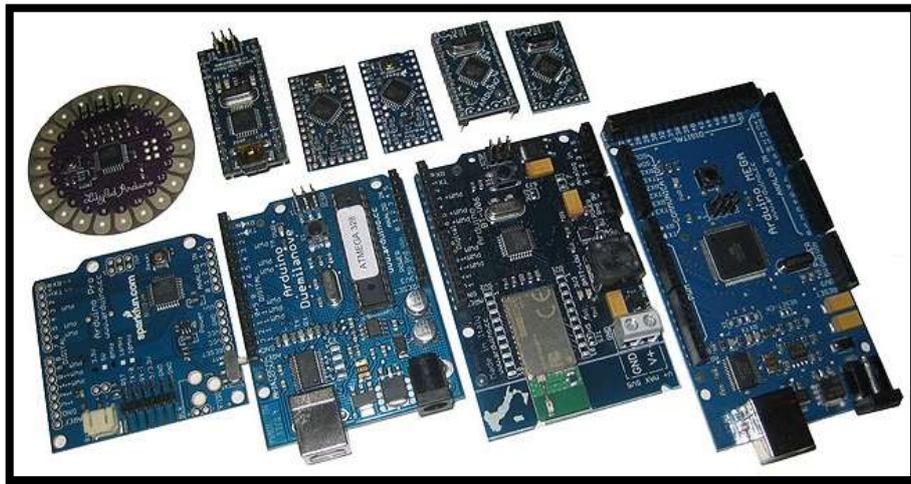


Figura. 2 Ejemplos placas ARDUINO

Fuente: <http://www.robotshop.com/blog/en/ARDUINO-robotics-projects-3666>.

El software ARDUINO consiste en un entorno de desarrollo (IDE³) y de librerías. El IDE está escrito en Java y basado en el entorno de desarrollo Processing⁴. Las librerías están escritas en C y C++ y compiladas usando las librerías avr-gcc y AVR. (Arduino, 2010).

El proyecto ARDUINO de desarrolla en un entorno de hardware basado en microcontroladores AVR, de los cuales dependiendo de las características del microcontrolador y las necesidades de la aplicación se han desarrollado varios diseños de placas en las cuales existen varias características especiales.

³ IDE Integrated development environment. Es un entorno de programación que ha sido empaquetado como un programa de aplicación.

⁴Processing. Lenguaje de programación basado en Java que puede heredar sus funcionalidades, herramienta muy poderosa al momento de encarar proyectos grandes.

1.3. Comunicación Serial (RS-232)

Para realizar una comunicación entre dispositivos electrónicos es necesario disponer de un estándar que norme el software y hardware para la transferencia de información, los dispositivos electrónicos son montados con varias interfaces de comunicación dependiendo de su utilidad y funcionamiento.

La manera de intercambio de información entre dispositivos electrónicos es de forma binaria “unos y ceros”, que representan estados de voltaje que son manejados por los dispositivos; existen 2 maneras de comunicación serial y paralela.

La comunicación paralela tiene como ventajas la velocidad de transferencia de bits ya que todos sus bits son transmitidos de manera simultánea pero esto conlleva a tener muchos pines para la transferencia y no se puede transmitir a largas distancias.

Existen dos tipos de comunicaciones seriales:

- **Síncrona y asíncrona**

En la comunicación Serial síncrona, se necesitan 2 líneas, una línea sobre la cual se transmitirán los datos y otra la cual contendrá los pulsos de reloj que indicaran cuando un dato es válido.

Ejemplos: de este tipo de comunicación son los protocolos:

- ✓ I2C (Inter Integrated Circuit)
- ✓ SPI (Serial Peripheral Interface)

En la comunicación Serial asíncrona, no son necesarios los pulsos de reloj. La duración de cada bit está determinada por la velocidad con cual se realiza la transferencia de datos. Como en toda comunicación serial, los datos viajan en grupos de bits. En este caso cada grupo o carácter consta de un bit Start, los bits de Datos (8 por lo general), un bit de Paridad (opcional) y finaliza con uno o dos bits de Stop.

- **Bit Start.** Es la transición de 1 a 0 e indica el inicio de una transferencia. En la lógica RS-232 podría significar una transición de -15V a +15V y en lógica TTL es una transición de 5V a 0V.
- **Bits de Datos.** Forman los datos en sí que se desean transmitir. Cada dato puede ser de 5, 6, 7 u 8 bits. Por supuesto, siempre preferimos trabajar con 8 bits (1 byte). El primer bit a transmitir es el menos significativo o LSbit (Least Significant Bit).
- **Bit de Paridad.** Este bit es opcional y se puede enviar después de los bits de datos. Sirve para ayudar a detectar posibles errores en las transferencias de datos. Es muy raramente usado, primero, porque es poco efectivo (solo podría detectar errores, no corregirlos) y, segundo, porque hay mejores formas de tratamiento de errores.

- **Bits Stop.** Los bits de Stop son estados de 1 lógico. El Estándar dice que puede haber 1, 1.5 o 2 bits de Stop al final de los datos (o del bit de paridad si lo hubiera).
(CursoMicros)

1.3.1. Velocidad de Transmisión (BAUD RATE)

El Baud Rate es el número de bits que se transmiten por segundo. En la transmisión asíncrona, no existe una señal de reloj que sincronice los bits de datos. Para que los dispositivos transmisor y receptor se entiendan correctamente también es necesario que operen con el mismo baud rate. Los valores más comunes que fija el Estándar RS-232 son: 1200, 2400, 4800, 9600, 19200, 38400, 56000, 57600, 115200, 128000, 256000. Aunque las versiones más recientes del Estándar ponen un límite de 20 kbits, es común emplear los valores altos como 115200 (siempre que sea posible). (Verduguez Muñoz, 2014)

1.4. Tratamiento de Señales Seriales

El estándar RS-232 define para una comunicación serial niveles de voltaje de -12V para un uno lógico y 12V para un cero lógico, estos niveles de voltaje eran usados hasta hace poco entre módems y computadores que disponían de un puerto serial nativo por hardware el cual soportaba estos niveles y sus fuentes de alimentación disponían estos voltajes, actualmente la mayoría de dispositivos que disponen de una señal serial para

transmitir información disponen de señales TTL⁵, las computadoras de escritorio y portátiles ya no disponen de puertos seriales nativos por lo que para su conexión se ha hecho común el usar un cable conversor de señales seriales a USB.

La comunicación entre dispositivos que adoptaron este cambio se la hace directamente pero con dispositivos que no han mantenido parámetros originales es necesario aparte de nivelar los niveles de voltaje invertir las señales para su correcto uso, básicamente esto se puede realizar de varias maneras, con el uso de transistores inversores, una compuerta lógica NOT o un circuito integrado dedicado; las 2 primeras opciones son válidas pero para una solución óptima con un tiempo de conversión adecuada es necesario invertir las señales con un circuito integrado.

1.5.Programación en Sistemas Electrónicos

Los sistemas microcontrolados para que puedan realizar tareas específicas y sistemáticas necesitan ser programados con rutinas y configuraciones que el microcontrolador almacena en su memoria para luego efectuar estas rutinas sistemáticamente.

⁵ TTL Transistor-Transistor Logic Es una tecnología de construcción de circuitos digitales, Los niveles lógicos vienen definidos por el rango de tensión comprendida entre 0,0V y 0,8V para el estado L (bajo) y los 2,4V y Vcc para el estado H (alto).

Un microcontrolador al ser un sistema electrónico solo reconoce niveles de voltaje que representan niveles lógicos de “unos” y “ceros”, el proceso para llegar con las instrucciones al microcontrolador es el siguiente:



Figura. 3 Instrucciones desde el programador hacia el microcontrolador

Fuente: Autor

El programador escribe sus instrucciones en un lenguaje de programación, este programa es compilado es decir transformado a “unos” y “ceros” para que el programador reciba las cadenas de datos binarios y los envíe al microcontrolador.

Existen varios lenguajes de programación para microcontroladores, de los cuales para su uso y estudio se han clasificado en 3 grandes grupos. En (MikroElektronica, 2008-2014) se describe estos grupos de programación a continuación se describe.

1.5.1. Lenguaje Máquina

Este lenguaje no necesita un compilador porque sus procesos son escritos directamente en secuencias binarias, este lenguaje es de fácil procesamiento para la máquina y el microcontrolador pero muy difícil y extenso para el programador.

1.5.2. Lenguajes de Bajo Nivel

Este tipo de lenguaje ya comprende un código fuente que normalmente es escrito en lenguaje ensamblador, dispone de varias instrucciones que hacen referencia a registros del microcontrolador para su programación, este código necesita un compilador⁶. Este lenguaje depende de la capacidad de la máquina que procese el compilado del programa fuente y de la capacidad del programador ya que sigue siendo muy complicado.

1.5.3. Lenguaje de Alto Nivel

Este tipo de lenguaje es el más usado actualmente ya que consiste en formar un programa con una serie de palabras conocidas y de fácil memorización, este lenguaje se asemeja al lenguaje humano, es necesario contar con un compilador que traduzca estas instrucciones a un lenguaje de bajo nivel.

Dependiendo de las necesidades y gustos del programador hoy en día se puede disponer de una gama alta de lenguajes de alto nivel, entre los más usados en la electrónica están en lenguaje C y Basic, además dependiendo del compilador que se use algunas de las sentencias pueden variar en el mismo lenguaje.

⁶ Compilador. Es un programa informático que traduce un lenguaje de programación a un lenguaje máquina.

1.5.3.1. Lenguaje C

El lenguaje C dispone de todas las ventajas de un lenguaje de programación de alto nivel (anteriormente descritas) y le permite realizar algunas operaciones tanto sobre los bytes como sobre los bits (operaciones lógicas, desplazamiento etc.). Las características de C pueden ser muy útiles al programar los microcontroladores. Además, C está estandarizado (el estándar ANSI), es muy portable, así que el mismo código se puede utilizar muchas veces en diferentes proyectos. Lo que lo hace accesible para cualquiera que conozca este lenguaje sin reparar en el propósito de uso del microcontrolador. C es un lenguaje compilado, lo que significa que los archivos fuentes que contienen el código C se traducen a lenguaje máquina por el compilador. Todas estas características hicieron al C uno de los lenguajes de programación más populares. (MikroElektronika, 2008-2014)

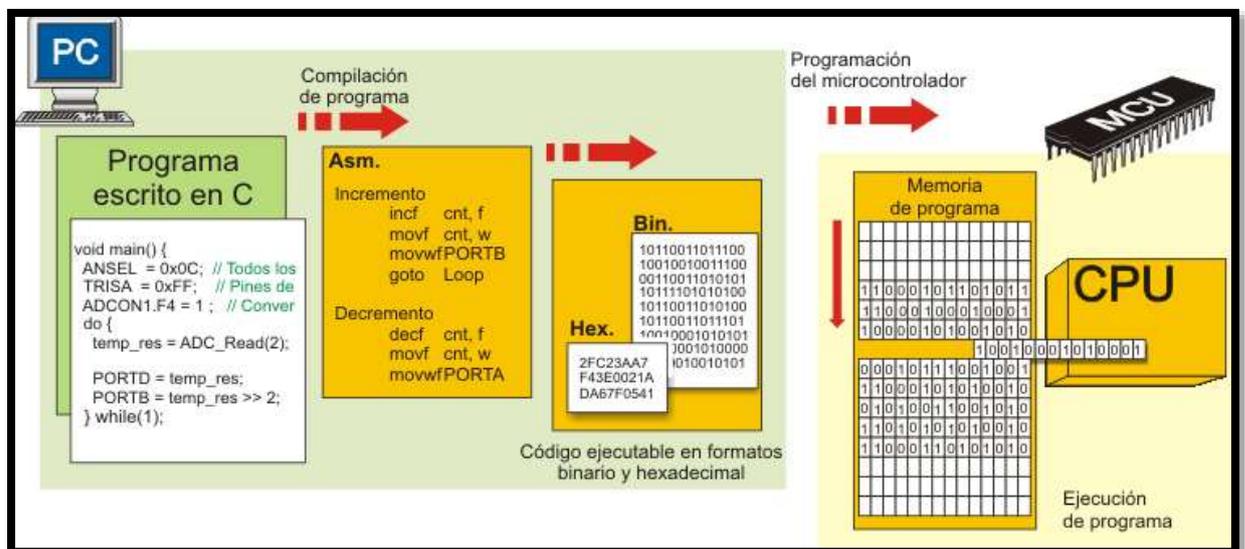


Figura. 4. Proceso de compilación de lenguaje de alto nivel a lenguaje máquina
Fuente: MiKroElektronika

1.6.Sensores

Un sensor es un dispositivo capaz de convertir una magnitud física a una señal

eléctrica, estos sensores normalmente entregan su respuesta de forma analógica (niveles de voltaje) por lo que es necesario un proceso de análisis de dicha señal para su posterior representación.



Figura. 5. Ejemplo de sensores

Fuente: <http://jaimejoelgo.blogspot.com/2009/02/act-6-sensores.html>

Los sensores se pueden clasificar dependiendo de la respuesta en activos y pasivos, los activos obtiene su respuesta por efecto de la energía otorgada por la entrada (Ej. Sensores piezoeléctricos y magnéticos) y los pasivos que solo se limitan a variar alguna condición (resistencia, capacitancia, inductancia) necesitando una fuente de alimentación.

1.6.1. Sensores de Proximidad

Este tipo de sensores detecta señales que se encuentran cerca del elemento sensor o determina la presencia de algún objeto que cause interferencia en su señal, según el principio físico que comande el sensor se clasifican en:

1.6.1.1. Sensor capacitivo

Este tipo de sensor tiene como función detectar un cambio de estado basado en la variación de un estímulo de un campo eléctrico, los sensores capacitivos detectan objetos metálicos y no metálicos midiendo el cambio de capacitancia, la cual depende de la constante dieléctrica del material a detectar, su masa, tamaño y distancia hasta la superficie sensible del detector.

1.6.1.2. Sensor inductivo

Este tipo de sensores consiste en generar un campo magnético y detectar las pérdidas de corriente de dicho campo cuando se introducen o interfieren objetos férricos y no férricos en su campo. Al aproximarse un objeto metálico o no metálico, se inducen corrientes de histéresis en el objeto. Debido a ello hay una pérdida de energía y una menor amplitud de oscilación. El sensor reconoce un cambio específico de amplitud y genera una señal que conmuta un estado “ON” y “OFF”.

1.6.1.3. Sensores Infrarrojos

Los sensores infrarrojos son una clasificación de los sensores fotoeléctricos ya que estos son sensibles a la radiación luminosa y en este caso a la luz infrarroja. Estos sensores presentan ventajas en cuanto a inmunidad a las interferencias pero por disponer de un transmisor y receptor es necesario que la superficie refleje la luz, a menos que la configuración sea directa entre el emisor y receptor. Su funcionamiento se basa en recolectar la señal producida por el emisor del sensor, estos sensores pueden ser de barrera (emisor y receptor en línea de vista) o de reflexión los cuales en el mismo encapsulamiento disponen de emisor y receptor.

1.6.1.4. Sensores de ultrasonido

Este sensor basa su funcionamiento en la emisión y recolección de sonidos ultrasónicos, el sensor emite impulsos y estos se reflejan en un objeto, el objeto recibe el eco y lo transforma a señales eléctricas, las cuales dependiendo del valor obtenido y relacionando con la velocidad del sonido se puede determinar la distancia desde el sensor hacia el objeto donde se reflejó los impulsos. Este sensor no necesita contacto físico con el objeto y ofrece la posibilidad de detectar objetos frágiles.

1.7. Tecnología GPRS – General Packet Radio Service

GPRS hace referencia a la tecnología desarrollada para transmitir datos mediante una red GSM desarrollado por el Instituto de Estandarización de Telecomunicaciones Europeo (ETSI, European Telecommunications Standards Institute) entre los años 1994 y 1997 como un sistema complementario a GSM por lo que se lo nombro como sistema 2.5G.

El aspecto primordial de una red GPRS es su funcionamiento en base a la conmutación de paquetes y no de circuitos como en GSM para realizar llamadas telefónicas. GPRS al transmitir su información mediante conmutación de paquetes se tarifa por tráfico enviando y no por tiempo establecido además que el canal de comunicación solo está disponible mientras la información se esté transmitiendo y luego puede ser usado por otro usuario.

Al estar sobre una red GSM se puede realizar una llamada y transmitir información en el mismo instante, el servicio GPRS permite que un terminal este siempre conectado “always on”, lo cual facilita conexiones instantáneas. Estas características permiten disponer varios tipos de transmisión como pueden ser tipo ráfaga, transmisión frecuente de volúmenes pequeños de datos, transmisión infrecuente de volúmenes grandes de datos.

La tecnología GPRS por estar sobre una red GSM reutiliza varios componentes de la infraestructura de la red ya existente, por ende la cobertura de una red GPRS es la misma que una red GSM.

1.7.1. Arquitectura GPRS (España, 2003)

La siguiente imagen describe todos los componentes que deben estar presentes para realizar una conexión GPRS, para lograr eso se añaden dos componentes a la red GSM que sirven para realizar la conmutación por paquetes.

Serving GPRS Support Node (SGSN)

Gateway GPRS Support Node (GGSN)

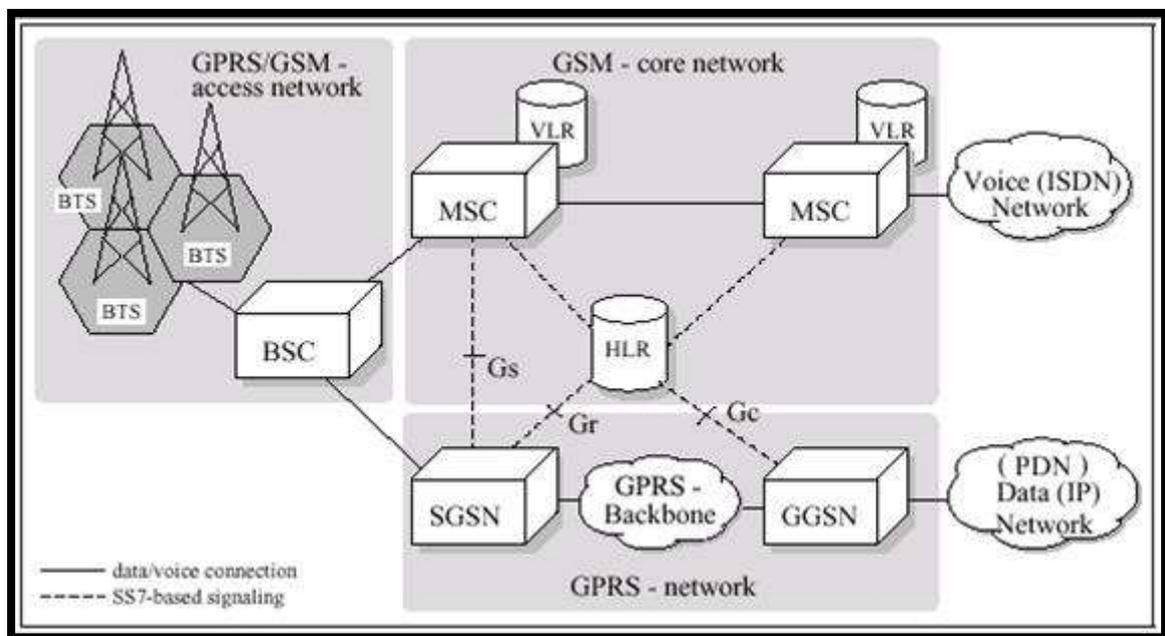


Figura. 6 Arquitectura GPRS

Fuente: <http://cecilia-urbina.blogspot.com/2012/02/gsm.html>

Los nodos GSN son los responsables de la conmutación y el encaminamiento de los paquetes entre los terminales móviles (MS) y las redes de datos externas (PDN). Estos nodos interoperan estrechamente con el HLR, con el MSC/VLR y con el BSS de GSM para lograr cumplir sus funciones.

1.7.1.1. Nodo Soporte de Servicio GPRS (SGSN, Serving GPRS Support Node).

SGSN es el nodo de conmutación de paquetes y se sitúa en el mismo nivel jerárquico que las MSCs en GSM. Este nodo es el responsable de la gestión y de la conexión del terminal móvil a la red GPRS lo que implica funciones de:

- Entrega de paquetes desde y hacia los móviles que se encuentran dentro del área de servicio.
- Control de acceso a la red GPRS mediante el intercambio de información con el HLR donde se encuentra el perfil de suscripción del usuario.
- Gestión de la localización y de la movilidad del usuario, tiene asociado un Local Register similar al VLR.
- Selección del nodo GGSN más apropiado para iniciar una sesión con la red de datos (Internet, red corporativa, etc.). El paso previo al establecimiento de la sesión es la activación del denominado contexto PDP (Estructura de datos presente en el

SGSN y GGSN). Durante esta fase el SGSN y el GGSN negocian los parámetros necesarios para que la conexión entre el terminal móvil y la PDN pueda establecerse. La sesión permanecerá mientras el contexto PDP esté activo.

- Encaminamiento y transferencia de paquetes entre las MSs y el GGSN, para esto utiliza en el interior del Backbone IP (GPRS) el Protocolo Tunelización GPRS (GTP, GPRS Tunneling Protocol) que permite establecer los túneles apropiados para la comunicación con el GGSN.
- Generación de registros de tarificación, denominados CDR.

1.7.1.2. Nodo Pasarela de Soporte GPRS (GGSN, Gateway GPRS Support Node).

GGSN actúa como interfaz entre la red troncal GPRS y con la red externa de datos como IP. Si se toma como referencia uno de los dos sentidos el GGSN convierte los paquetes GPRS provenientes del SGSN en el formato correspondiente a la red externa de datos que en nuestro caso será IP, efectuando después el envío de los mismos. En lo que respecta al sentido contrario traduce las direcciones IP que llegan y los envía al SGSN que corresponda para alcanzar el móvil destino. En general se puede decir que realiza funciones de enrutamiento de paquetes entre la red interna y externa. GGSN realiza las siguientes funciones:

- Realiza funciones de Pasarela de Borde (BG, Border Gateway) que permiten una interfaz con las redes de otros operadores GPRS.
- Funciones de autenticación de usuarios móviles con las redes de paquetes externas.
- Genera también registro de tarificación CDR.

1.7.1.3. Central de conmutación móvil (MSC, Mobile Switching Center).

Realiza la conmutación telefónica del sistema, sirve como puerta de enlace hacia el mundo exterior (PSTN) o hacia diferentes bases de datos necesarias para administrar a los usuarios de la red. También captura la información de facturación de una llamada.

1.7.1.4. Registros de Ubicación Base (HLR, Home Location Register).

Es una base de datos estática, que contiene detalles de la suscripción de cada abonado, puede manejar información de cientos de miles de abonados, para localizar a un abonado esta almacena la información del área en la que el abonado se registró por primera vez. Un HLR está conectado al MSC y VLR.

1.7.1.5.Registro de ubicación de visitante (VLR, Visitor Location Register).

Es una base de datos que contiene información dinámica de los usuarios que están asociados a la red móvil, incluyendo la ubicación geográfica. Una VLR está asociada con una sola MSC. Cuando un usuario se mueve de un lugar a otro los datos del VLR “vieja” se pasan al VLR de la ubicación a la cual entra “nueva”, este procedimiento se lo realiza usando el procedimiento de actualización de ubicación.

1.7.1.6.Subsistema de Estaciones Base (BSS, Base Station Subsystem).

EL subsistema de radio base permite conectar la estación móvil con el subsistema de red y conmutación (NSS). Consta de dos elementos Transceptor de Estación Base (BTS) y Controlador de estación base (BSC)

1.7.1.6.1. Transceptor de Estación Base (BTS, Base Station Transceiver).

Consta de dispositivos de transmisión y recepción por radio frecuencia (transceivers), incluyendo las antenas y todo el procesamiento de las señales en el interfaz aire. Una estación puede alcanzar un radio de cobertura a su alrededor desde varios cientos de metros (Zonas urbanas) hasta un máximo de 35 km (zonas rurales) dependiendo de la potencia y la orografía del entorno, a esta área geográfica que brinda cobertura se le conoce como celda o célula.

1.7.1.6.2. *Controlador de Estación Base (BSC, Base Station Controller).*

Asigna y administra los recursos de radio a una estación móvil, para una o varias BTS, maneja el handoff de una estación móvil de una celda a otra dentro del BSS y controla el paging. La BSC es la interfaz entre la estación móvil y la MSC, puede estar ubicada físicamente en el mismo lugar que la BTS.

1.7.2. Funcionamiento GPRS

Un dispositivo para que pueda recibir y transmitir datos sobre una red GPRS este debe primero asociarse a un SGSN y activar su dirección PDP (Packet Data Protocol). La asociación de estos registros se los conoce como contexto PDP.

1.7.2.1. *Activación del contexto PDP*

El contexto PDP contiene una dirección IP, sea IPv4 o IPv6, clase de QoS requerido y la dirección del GGSN conocida como APN (Access Point Name); este contexto es almacenado en la estación móvil, este contexto varía entre los operadores de servicio.

- La MS envía un mensaje “Active PDP Context Request” hacia el SGSN, este mensaje incluye el APN de la red a la que desea conectarse, QoS solicitada, y el tipo de direccionamiento sea este dinámico o estático.

- Este mensaje es enviado por el SGSN al DNS para determinar a cual GGSN debe enviar la solicitud.
- El SGSN envía la petición al GGSN.
- El GGSN valida al usuario y le asigna una dirección IP dependiendo del tipo de direccionamiento al que se haya asignado, si es la primera vez estos datos se guardan en el HLR en caso de ser un una MS que se encuentra en movilidad estos datos se almacenan en la VLR.
- El SGSN envía un mensaje de aceptación a la estación móvil "Activate PDP Context Accept".

1.8.Tecnología GPS

1.8.1. Sistema de Posicionamiento Global GPS

El sistema de posicionamiento global GPS nace como un proyecto militar estadounidense en el año 1973, la responsabilidad y desarrollo del mismo estaba a cargo del Departamento de Defensa de los Estados Unidos y su uso era estrictamente militar. GPS es un sistema que tiene como objetivo la determinación de las coordenadas espaciales de puntos de un sistema de referencia mundial. Los puntos pueden estar ubicados en cualquier lugar del planeta, pueden permanecer estáticos o en movimiento y las observaciones pueden realizarse en cualquier momento del día.

Para la obtención de coordenadas el sistema se basa en la determinación simultánea de las distancias a cuatro satélites (como mínimo) de coordenadas conocidas. Estas distancias se obtienen a partir de las señales emitidas por los satélites, las que son recibidas por receptores especialmente diseñados. Las coordenadas de los satélites son provistas al receptor del sistema. (Huerta, Mangiaterra, & Noguera, 2005)

A partir del año 1984 por un incidente aéreo entre aerolíneas se permitió hacer uso de este sistema pero se tenía muchas restricciones en su exactitud por el tema de seguridad estadounidense, con la evolución de la electrónica y procesos de cambio en el año 2000 de la mano del Presidente Bill Clinton se decidió eliminar todas los errores y restricciones a las mediciones en los relojes atómicos de los satélites con lo cual toda la población civil dispondría de un sistema de navegación confiable. (NEOTEO)

1.8.2. Componentes del Sistema de Posicionamiento Global

Para que el sistema de posicionamiento cumpla con su objetivo es necesario disponer de estos 3 elementos:

- Segmento espacial o Constelación de satélites.
- Segmento de control o Estaciones terrenas.
- Segmento de usuario o Receptores GPS.

1.8.2.1.Segmento Espacial

El segmento espacial consiste en una constelación de satélites que transmiten señales de radio hacia los usuarios. Los Estados Unidos se compromete a mantener disponibles al menos 24 satélites GPS operativos, el 95% del tiempo. Para garantizar este compromiso, la Fuerza Aérea Norteamericana mantiene en órbita 31 satélites GPS operacionales en los últimos años.

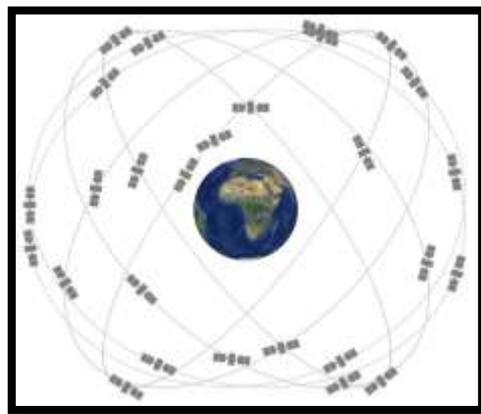


Figura. 7. Constelación de satélites

Fuente: <http://www.gps.gov/multimedia/images/constellation.jpg>

Los satélites vuelan en la órbita media de la tierra (MEO) a una altura aproximada de 20200Km, cada satélite gira alrededor de la tierra 2 veces al día. La constelación de los satélites GPS están organizados en 6 planos orbitales iguales circundante a la Tierra. Cada plano contiene cuatro "slots" ocupados por los satélites de base. El arreglo de 24 slots asegura que un usuario pueda mirar como mínimo cuatro satélites desde cualquier parte del planeta.

La Fuerza Aérea norteamericana normalmente opera más de 24 satélites GPS para mantener cobertura cada vez que los satélites bases están en mantenimiento o son sacados de servicio. Los satélites extras pueden incrementar el rendimiento de GPS pero no son considerados parte de la constelación base.

En Junio de 2011, la Fuerza Aérea norteamericana completo satisfactoriamente la expansión de la constelación GPS llamada como " Expandable 24". Tres de los 24 slot se ampliaron, y seis satélites fueron reubicados, por lo que tres de los satélites extras son ahora parte de la base de la constelación. Como resultado, GPS ahora opera efectivamente en una constelación de 27 slots con mejora de cobertura en más partes del mundo.

1.8.2.2.Segmento de control

Las estaciones terrestres de control consisten en una red global de tierra que realizan un seguimiento a los satélites GPS, monitorean las transmisiones, realizan análisis y envían comandos y datos a la constelación. La actual operación de las estaciones terrenas disponen de una estación de control master, una estación master alterna, 12 antenas de mando y control, y 16 sitios de monitoreo.

- La estación de control master (MSC) de Colorado provee mando y control de la constelación GPS. La MSC genera y carga mensajes de navegación y asegura la

estabilidad y precisión de la constelación GPS. Recibe información de navegación desde las estaciones de monitoreo, utiliza esta información para calcular la posición precisa de los satélites GPS en el espacio y entonces cargar estos datos en los satélites.

- Las estaciones de monitoreo rastrean a los satélites GPS en su paso por las estaciones y envían esta información a la estación central, estas estaciones recolectan datos atmosféricos, señales de navegación. Estas estaciones disponen de sofisticados receptores GPS. Las 16 estaciones de monitoreo están ubicadas alrededor del mundo, incluyendo 6 de la Fuerza Aere norteamericana y 10 estaciones de la Agencia Nacional de Inteligencia Geoesacial (NGA) alrededor del mundo, una de estas estaciones se encuentra ubicada en la ciudad de Quito.
- Las antenas en tierra son usadas para comunicarse con los satélites GPS con fines de mando y control. Estas antenas con enlaces de comunicación en la banda S transmiten datos de navegación y cargan programas, además recolectan datos de telemetría.

1.8.2.3.Segmento de usuario

Este componente consiste en el dispositivo de usuario el cual es el encargado de recolectar la información enviada por los satélites y procesarla para determinar la posición del dispositivo. GPS al ser un sistema abierto cualquier empresa puede desarrollar

dispositivos capaz de obtener estas señales y además usar esta información para cualquier propósito sea este comercial o particular. Las aplicaciones en las que se usan actualmente son muy variadas en la actualidad siendo las más utilizadas la geolocalización de medios de transporte.

1.8.3. FUNCIONAMIENTO GPS

Para determinar la ubicación de un objeto se utiliza el método de triangulación, el cual consiste en medir las distancias desde el objeto hasta 3 satélites (como mínimo), aunque es necesario el medir un 4 satélite para tener exactitud y descartar mediciones incorrectas.

Cada medición desde el objeto hacia el satélite posiciona al objeto en un lugar en la tierra y la siguiente medición hará que esta posición sea más exacta; estas mediciones se logran calculando el tiempo que se demora en viajar la señal hasta el satélite.

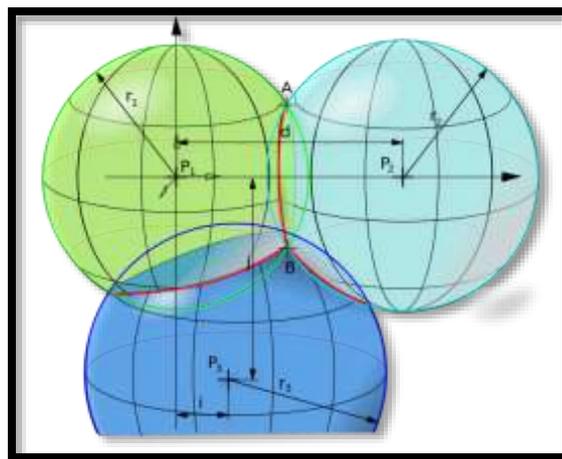


Figura. 8. Ejemplo de triangulación hacia el punto B
Fuente: Heriberto Arribas Abato

Cada satélite dispone de un código [Código Pseudo Aleatorio] para identificarse con el receptor y sincronizarse. Dado que cada uno de los satélites tiene su propio y único Código Pseudo Aleatorio, esta complejidad también garantiza que el receptor no se confunda accidentalmente de satélite. De esa manera, también es posible que todos los satélites transmitan en la misma frecuencia sin interferirse mutuamente. Esto también complica a cualquiera que intente interferir el sistema desde el exterior al mismo. El Código Pseudo Aleatorio le da la posibilidad al Departamento de Defensa de EEUU de controlar el acceso al sistema GPS.

Pero hay otra razón para la complejidad del Código Pseudo Aleatorio, una razón que es crucial para conseguir un sistema GPS económico.

El código permite el uso de la "teoría de la información" para amplificar las señales de GPS. Por esa razón las débiles señales emitidas por los satélites pueden ser captadas por los receptores de GPS sin el uso de grandes antenas. (Pedro Gutovnik)

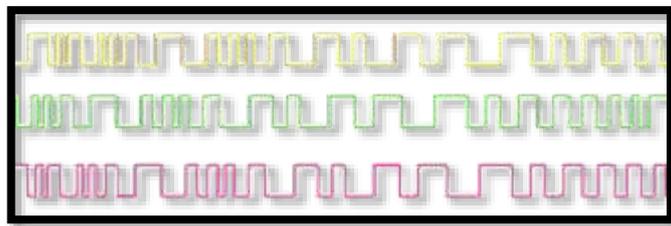


Figura. 9. Ejemplos de códigos pseudo-aleatorios
Fuente: <http://gutovnik.com/>

1.9. Transmisión de Información en un Sistema GPS

1.9.1. Protocolo NMEA 0183

NMEA es la abreviatura de National Marine Electronics Association, es una asociación sin fines de lucro de fabricantes, distribuidores, instituciones educativas y otros interesados en dispositivos electrónicos marítimos. El estándar NMEA define una interfaz eléctrica y un protocolo de datos para comunicaciones entre la instrumentación marítima.

Los dispositivos NMEA están diseñados como transmisores o receptores (algunos con las dos funcionalidades), empleando una interfaz serial asincrónica con los siguientes parámetros: (NMEA)

Tabla 1. Parámetros estándar NMEA

Velocidad en baudios	4800
Numero de bits	8 (bit 7 a 0)
Bits de parada	1
Paridad	No
Handshake	No

Fuente: Protocolo NMEA 0183

Dependiendo del dispositivo el estándar NMEA define tramas de información con identificadores, para los dispositivos receptores GPS hay una serie de tramas NMEA [Anexo] que muestran varios datos y parámetros calculados; los cuales dependiendo del propósito deben ser filtrados y analizados, el identificador es \$GPxxx donde las xxx varían dependiendo de la información que contenga la trama.

1.9.2. Trama GPRMC

Esta trama contiene los datos específicos mínimos recomendados en una transmisión GPS, el siguiente es un ejemplo de la trama y la descripción de todos sus campos. (Sparkfun)

\$GPRMC,220516,A,5133.82,N,00042.24,W,173.8,231.8,130694,004.2,W*70

Tabla 2. Descripción campos trama GPRMC

NOMBRE	EJEMPLO	UNIDAD	DESCRIPCIÓN
ID del mensaje	\$GPRMC		
Hora UTC	220516		hora minutos segundos
Estatus	A		A= datos validos V= datos inválidos
Latitud	5133.82		N = Norte S= Sur
Indicador N/S	N		
Longitud	00042.24		
Indicador E/W	W		E = Este W= Oeste
Velocidad	173.8	Nodos	
Rumbo	231.8	grados	Posición hacia donde se dirige
Fecha	130694		día mes año
Variación	004.2		
Indicador E/W	W		
Checksum	*70		Suma de verificación
<CR> <LF>			Termina la trama con retorno de carro y salto de línea.

Fuente: Ésta tabla fue tomada de <http://aprs.gids.nl/nmea/>

1.10. Comandos HAYES - Comandos AT

El conjunto de comandos Hayes es un lenguaje desarrollado por la compañía Hayes Communications que prácticamente se convirtió en estándar abierto de comandos para configurar y establecer parámetros en los módems. Los caracteres «AT», que preceden a todos los comandos, significan «Atención», e hicieron que se conociera también a este conjunto de comandos como comandos AT.

Un aparato que implemente el conjunto de comandos Hayes se considera compatible Hayes. Parte del conjunto de comandos Hayes fue incluido por la ITU-T⁷ en el protocolo V.25ter, actual V.250. La adopción de este estándar hizo el desarrollo de controladores específicos para distintos módems superfluo.

Como se dijo anteriormente la mayoría de los comandos Hayes empieza con la secuencia “AT”, siendo las excepciones el comando “A/” que repite el último comando introducido y la secuencia triple del carácter de escape. Los otros comandos luego de la secuencia “AT” siguen con las letras del alfabeto. Además, muchos de ellos necesitan a continuación un valor numérico, que en el caso que no se escriba se tomará como que

⁷ ITU-T Unión Internacional de las Telecomunicaciones sector de Normalización. Es la encargada de redactar “recomendaciones” acerca de estándares desarrollados por la UIT.

dicho valor es cero.

Ante cualquier comando enviado al módem, este responde con el resultado de la operación: "OK", "ERROR", CONNECT, etc. El operador de la terminal puede elegir que el resultado, en vez de aparecer en forma alfabética, aparezca de forma numérica donde cada número corresponde a una respuesta en particular. Por ejemplo "OK" tiene por valor cero.

En los últimos años el uso de comandos AT han sido utilizados especialmente en la configuración y conexión en equipos móviles los mismos que disponen de módems GSM/GPRS; en la práctica estos comandos sirven para asignar parámetros de configuración y para realizar acciones específicas como hacer llamadas, enviar mensajes, transmitir datos, sin usar la interfaz física o lógica por defecto del dispositivo. (Universidad de Buenos Aires).

1.11. Protocolo de Control de Transmisión TCP

El "protocolo de control de transmisión" ('Transmission Control Protocol', TCP) está pensado para ser utilizado como un protocolo 'host' a 'host' muy fiable entre miembros de redes de comunicación de computadoras por intercambio de paquetes y en un sistema interconectado de tales redes, el propósito principal de TCP consiste en proporcionar un servicio de conexión o circuito lógico fiable y seguro entre pares de procesos.

1.11.1. Formato de un Segmento TCP

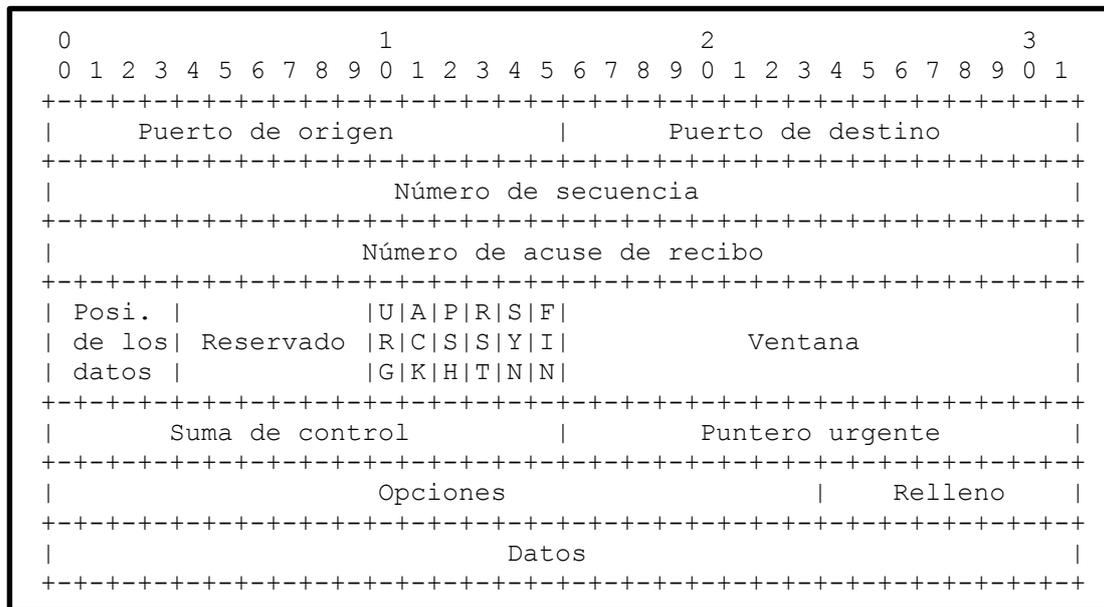


Figura. 10. Formato trama TCP
Fuente: RFC 793

El módulo de TCP debe poder recuperar los datos que se corrompan, pierdan, dupliquen o se entreguen desordenados por el sistema de comunicación del entorno de internet. Esto se consigue asignando un número de secuencia a cada octeto transmitido, y exigiendo un acuse de recibo (ACK, N.T.:del inglés 'acknowledgment') del módulo de TCP receptor. Si no se recibe un ACK dentro de un cierto plazo de expiración prefijado, los datos se retransmiten. En el receptor, se utilizan los números de secuencia para ordenar correctamente los segmentos que puedan haber llegado desordenados y para eliminar los duplicados. La corrupción de datos se trata añadiendo un campo de suma de control ('checksum') a cada segmento transmitido, comprobándose en el receptor y descartando los segmentos dañados. En tanto en cuanto los módulos de TCP continúen funcionando

adecuadamente y el sistema de internet no llegue a quedar particionado de forma completa, los errores de transmisión no afectarán la correcta entrega de datos. TCP se recupera de los errores del sistema de comunicación de internet. (Grupo de Trabajo de Ingeniería de Internet)

1.12. Socket de Comunicación

En los inicios de la computación en la universidad de Berkeley los programadores tuvieron la necesidad de interconectar sus programas, es de ahí que nacen los socket como un mecanismo de comunicación como una variación del sistema Unix llamándolo BDS Unix.

Una forma de conseguir que dos programas se transmitan datos, basada en el protocolo TCP/IP, es el uso de sockets. Un socket no es más que un "canal de comunicación" entre dos programas que corren sobre ordenadores distintos o incluso en el mismo ordenador. Para que exista una comunicación efectiva se necesitan tres componentes, el protocolo de comunicación, la dirección del protocolo (dirección IP) y el número de puerto.

1.12.1. Protocolo de Comunicación

Existen básicamente dos tipos de "canales de comunicación" o sockets, los orientados a conexión y los no orientados a conexión.

En el primer caso ambos programas deben conectarse entre ellos con un socket y hasta que no esté establecida correctamente la conexión, ninguno de los dos puede transmitir datos. Esta es la parte TCP del protocolo TCP/IP, y garantiza que todos los datos van a llegar de un programa al otro correctamente. Se utiliza cuando la información a transmitir es importante, no se puede perder ningún dato y no importa que los programas se queden "bloqueados" esperando o transmitiendo datos. Si uno de los programas está atareado en otra cosa y no atiende la comunicación, el otro quedará bloqueado hasta que el primero lea o escriba los datos.

En el segundo caso, no es necesario que los programas se conecten. Cualquiera de ellos puede transmitir datos en cualquier momento, independientemente de que el otro programa esté "escuchando" o no. Es el llamado protocolo UDP, y garantiza que los datos que lleguen son correctos, pero no garantiza que lleguen todos. Se utiliza cuando es muy importante que el programa no se quede bloqueado y no importa que se pierdan datos. (<http://www.chuidiang.com>)

1.12.2. Dirección de Protocolo (DIRECCIÓN IP) y Número de Puerto

La dirección de protocolo o dirección IP es una etiqueta numérica única que representa o identifica una interfaz de red de un dispositivo dentro de una red que utilice el protocolo IP (internet protocolo) que corresponde al nivel red del modelo OSI.

Para comunicaciones locales se hace uso de direcciones IP privadas mientras que para transferencias de información en el internet es necesario usar direcciones públicas.

Los socket pueden realizar varias conexiones hacia una misma dirección IP gracias al uso de un número de puerto el cual puede estar comprendido entre el cero y 65535, estos puertos se han dividido en puertos bien conocidos y no bien conocidos. Los puertos bien conocidos se encuentran entre 0 y 1024 y están destinados a las conexiones de protocolos estándares comúnmente conocidos como telnet, ssh, ftp, etc. Los puertos no bien conocidos se encuentran desde 1025 a 65535 y sirven para aplicaciones personales o de carácter temporal.

1.13. Base de Datos

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos inter-relacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente. Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información.

La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos.

Dado que la información es tan importante en la mayoría de las organizaciones, los desarrolladores informáticos han desarrollado un amplio conjunto de conceptos y técnicas para la gestión de los datos. (Silberschatz, Korth, & Sudarshan, 2006)

Un sistema de gestión de base de datos (SGBD) otorga a una aplicación varias ventajas las cuales otorgan superioridad a cualquier otro tipo de almacenamiento de datos:

- Control sobre redundancia de datos
- Consistencia de datos
- Compartición de datos
- Integridad
- Seguridad
- Accesibilidad y recurrencia

Todas estas ventajas deben ser administradas mediante software y hardware en una organización en el cual es vital el manejo de datos, no solo depende del motor de procesamiento de la base de datos sino también de elementos extras que complementan el trabajo, todos estos recursos deben medirse como inversión en una aplicación.

Actualmente en el mercado existen varios sistemas de gestión de base de datos los cuales han sido desarrollados por empresas privadas y otras por grupos de desarrollo en el que son financiadas por el soporte y mantenimiento personalizados a usuario finales. Las bases de datos al igual que cualquier software dispone de licencias para su funcionamiento sean libres o privativas, este aspecto no merma el rendimiento de una base de datos siendo el sistema como tal el que aporta el rendimiento a la solución.

1.13.1. POSTGRESQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos* para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (Martinez) . La imagen a continuación describe de manera general los componentes de un sistema PostgreSQL.

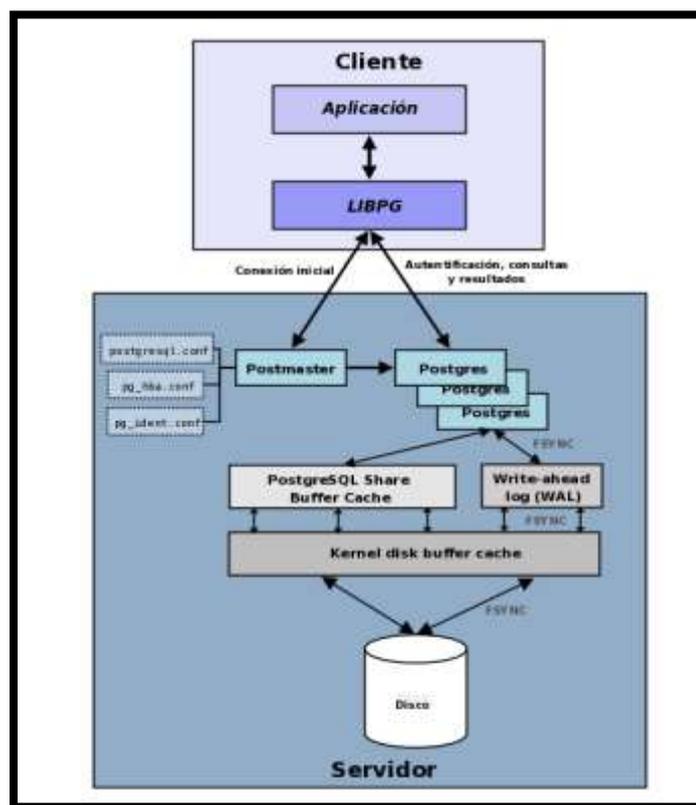


Figura. 11 Componentes PostgreSQL

Fuente: http://www.postgresql.org/es/sobre_postgresql

- **Aplicación cliente:** Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir via TCP/IP o sockets locales.
- **Demonio postmaster:** Este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autentificar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes
- **Ficheros de configuración:** Los 3 ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg_hba.conf y pg_ident.conf
- **Procesos hijos postgres:** Procesos hijos que se encargan de autentificar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes
- **PostgreSQL share buffer cache:** Memoria compartida usada por PostgreSQL para almacenar datos en caché.
- **Write-Ahead Log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO)
- **Kernel disk buffer cache:** Caché de disco del sistema operativo
- **Disco:** Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione

1.14. Lenguaje de Programación PYTHON

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible.

Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos. (Gonzales)

1.14.1. Lenguaje Interpretado o de Script

Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (lenguajes compilados). La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo los lenguajes interpretados son más flexibles y más portables.

Python tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. En Python, como en Java y muchos otros lenguajes, el código fuente se traduce a un pseudo código máquina intermedio llamado bytecode la primera vez que se ejecuta, generando archivos .pyc o .pyo (bytecode optimizado), que son los que se ejecutarán en sucesivas ocasiones.

1.14.2. SOCKET EN PYTHON

Los tipos y funciones necesarios para trabajar con sockets se encuentran en Python en el módulo socket, como no podría ser de otra forma.

Los sockets se clasifican en sockets de flujo (socket.SOCK_STREAM) o sockets de datagramas (socket.SOCK_DGRAM) dependiendo de si el servicio utiliza TCP, que es orientado a conexión y fiable, o UDP, respectivamente.

Para crear un socket se utiliza el constructor socket.socket() que puede tomar como parámetros opcionales la familia, el tipo y el protocolo. Por defecto se utiliza la familia AF_INET y el tipo SOCK_STREAM.

1. Lo primero que tenemos que hacer es crear un objeto socket para el servidor ver Figura. 12.

```
socket s = socket.socket()
```

Figura. 12 Código para crear un objeto
Fuente: Autor

2. En la figura 13 se indica en qué puerto se va a mantener a la escucha nuestro servidor utilizando el método bind. Para sockets IP, como es nuestro caso, el argumento de bind es una tupla⁸ que contiene el host y el puerto. El host se puede dejar vacío, indicando al método que puede utilizar cualquier nombre que esté disponible.

```
socket s.bind(("localhost", 9999))
```

Figura. 13 Asignación de IP y puerto

Fuente: Autor

3. Por último utilizamos listen para hacer que el socket acepte conexiones entrantes y accept para comenzar a escuchar ver Figura 14. El método listen requiere de un parámetro que indica el número de conexiones máximas que queremos aceptar; evidentemente, este valor debe ser al menos 1. El método accept se mantiene a la espera de conexiones entrantes, bloqueando la ejecución hasta que llega un mensaje. Cuando llega un mensaje, accept desbloquea la ejecución,

⁸ **Tupla.** Es una secuencia de valores agrupados. Sirve para agrupar, como si fueran un único valor, varios valores que, por su naturaleza, deben ir juntos.

devolviendo un objeto socket que representa la conexión del cliente y una tupla que contiene el host y puerto de dicha conexión.

```
socket s.listen(10)
socket c, (host c, puerto c) = socket s.accept()
```

Figura. 14 Código para aceptar conexiones
Fuente: Autor

4. Una vez que tenemos este objeto socket podemos comunicarnos con el cliente a través suyo, mediante los métodos `recv` y `send` (o `recvfrom` y `sendfrom` en UDP) que permiten recibir o enviar mensajes respectivamente. El método `send` toma como parámetros los datos a enviar, mientras que el método `recv` toma como parámetro el número máximo de bytes a aceptar como se indica en la Figura 15.

```
recibido = socket c.recv(1024)
print "Recibido: ", recibio
socket c.send(recibido)
```

Figura. 15 Código para enviar y recibir por el socket
Fuente: Autor

1.15. Sistema de Información Geográfica QGIS

Sistema de Información Geográfica (QGIS) es de código abierto. El proyecto nació en mayo de 2002 y se estableció como un proyecto en SourceForge en junio del mismo año. Se ha trabajado duro para hacer que el software SIG (tradicionalmente software propietario caro) esté al alcance de cualquiera con acceso básico a un ordenador personal. QGIS actualmente funciona en la mayoría de plataformas Unix, Windows y OS X. QGIS se desarrolla usando el kit de herramientas Qt (<http://qt.digia.com>) y C++. Esto significa que QGIS es ligero y tiene una interfaz gráfica de usuario (GUI) agradable y fácil de usar.

QGIS pretende ser un SIG amigable, proporcionando funciones y características comunes. El objetivo inicial del proyecto era proporcionar un visor de datos SIG. QGIS ha alcanzado un punto en su evolución en el que está siendo usado por muchos para sus necesidades diarias de visualización de datos SIG. QGIS admite diversos formatos de datos ráster y vectoriales, pudiendo añadir nuevos formatos usando la arquitectura de complementos.

QGIS se distribuye bajo la Licencia Pública General GNU (GPL). El desarrollo de QGIS bajo esta licencia significa que se puede revisar y modificar el código fuente y garantiza que el usuario siempre tendrá acceso a un programa de SIG que es libre de costo y puede ser libremente modificado. (QGIS , 2014)

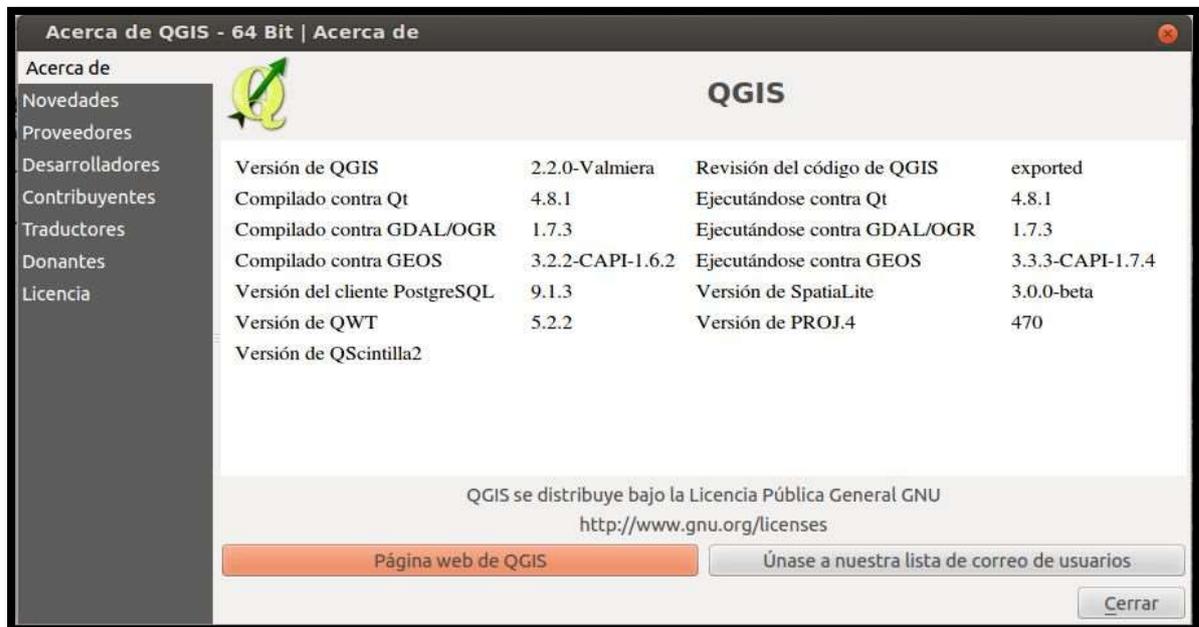


Figura. 16 Software QGIS
Fuente: Autor

PLAN DE NEGOCIOS PARA EL SISTEMA DE CONTEO DE PASAJEROS Y ENVIÓ DE INFORMACIÓN MEDIANTE TECNOLOGÍA GPRS DE LA EMPRESA INPRISE.

3.1. Oportunidades del Negocio

Esta sección describe las principales características por las cuales se desarrollará este proyecto en la ciudad de Ibarra y por qué está enfocada al sector de transporte público de pasajeros.

El proyecto está cimentado sobre un nicho de mercado limitado y específico lo que lo hace más fácil de delimitar su procedimiento pero más difícil que concretar un rendimiento óptimo.

3.1.1. Análisis del Sector

La ciudad de Ibarra ubicada en el norte del país cuenta con una población de 181.175 habitantes [según INEC censo del 2010], el principal medio de transporte de la ciudad de Ibarra es el transporte público, actualmente el transporte público está regulado y

sus normas están otorgadas por el Gobierno Autónomo Descentralizado (GAD) de la Ciudad de Ibarra, el cual es el encargado de disponer de permiso de operación y asignación de rutas a todas las cooperativas registradas para ofrecen el servicio en la ciudad.

Actualmente la ciudad de Ibarra cuenta con 4 Cooperativas de Transporte público legalmente acreditadas que cubren el acceso a la mayoría de parroquias de la ciudad.

Tabla 3. Numero unidades asignadas a las cooperativas de transporte público en Ibarra

Cooperativa	Unidades
28 de Septiembre	160
San Miguel de Ibarra	127
La Esperanza	24
Campesinor	7

Fuente: Dirección de tránsito GAD-Ibarra

La unidad de transito del GAD de Ibarra requirió que las cooperativas cuenten con un sistema de rastreo para controlar sus diferentes rutas y turnos en tiempo real, además disponer un historial sobre los recorridos de las unidades y sus velocidades. El 90% de las unidades de transporte en la ciudad cuentan con un sistema de control y ubicación mediante dispositivos GPS-GPRS, de los cuales el 100% están bajo el control de la empresa INPRISE, la cual oferto los equipos y el software de gestión. Aunque este sistema fue requerido por el GAD de Ibarra todos los rubros referentes a equipamiento físico y lógico fueron asumidos por las cooperativas contratantes hacia la empresa INPRISE, además el costo operativo mensual es responsabilidad de cada cooperativa.

El sistema que actualmente está en funcionamiento además de ofrecer los servicios requeridos por el GAD de Ibarra cuenta con una función específica en la cooperativas que adquirieron este sistema ya que a través del software se realiza asignación automática de rutas y tiempos a todas las unidades y contabilización de atrasos, a nivel de propietarios de los vehículos estos poseen accesos individuales para que puedan localizar y controlar su vehículo desde cualquier lugar a través de acceso vía Internet.

Un 75% (datos obtenidos de la Base de datos de las Cooperativas) de las unidades de transporte público no son conducidas por sus propietarios, de lo cual nace la necesidad de disponer dispositivos que ayuden a controlar el ingreso de pasajeros y más aún cuando se puede ofertar una solución que se adhiere a su sistema de monitoreo.

El uso de contadores de pasajeros o dispositivos que determine la cantidad de pasajeros que ingresan a una unidad no están contemplados en la normativa del transporte público en la ciudad de Ibarra, esto se debe a que todo el rédito económico de las unidades de transporte es exclusivo de los propietarios de las unidades de transporte, por este motivo cualquier dispositivo o equipamiento para este cometido es exclusivo de los propietarios de las unidades de transporte, tomando en cuenta las decisiones en conjunto de la cooperativa a cual pertenecen.

INPRISE por ofrecer un servicio adicional al ya prestado y garantizar la satisfacción de los clientes desarrolla un sistema de conteo de pasajeros el cual transmitirá información por el actual sistema GPRS que cuentan las unidades de transporte, este servicio al igual que el servicio ofertado de control y de rastreo está destinado al total de las unidades de las cooperativas.

3.1.2. Análisis de la Industria

El producto a ser diseñado y construido por la empresa INPRISE está dirigido especialmente al sector de la transportación pública de pasajeros, es un producto para un segmento específico del mercado ya que su aplicación se basa en el conteo de pasajeros en vehículos de transporte público y la transmisión de información por un sistema GPRS del cual el 90% de las unidades de transporte público de la ciudad ya disponen. Este 90% de vehículos lo comprenden las 2 cooperativas más grandes de la ciudad por lo cual este nuevo dispositivo estará enfocado hacia la comercialización del total de vehículos.

El conteo de pasajeros a nivel nacional ha sido regulado solo de tal forma que no interfiera en el acceso hacia la unidad de transporte, refiriéndose a equipos mecánicos (Ej. torniquete) los cuales están prohibidos su funcionamiento por tratarse de dispositivos que necesitan intervención del usuario y que en algunos casos se hace difícil para personas discapacitadas, adultos mayores o niños, el sistema a ofertar debe ser autónomo, independiente e imperceptible hacia el usuario.

En la ciudad de Ibarra existen varias franquicias que se dedican a la instalación de sistemas de rastreo por medio de dispositivos GPS-GPRS, estas empresas no desarrollan sus sistemas localmente y no ofertan servicios a medida. En cuanto al desarrollo de hardware electrónico a medida y con una línea de producción no existen empresas registradas para tal propósito ya que solo existen tiendas menores de comercio de elementos electrónicos y desarrollo de proyectos universitarios.

La empresa INPRISE se encuentra radicada en la ciudad de Ibarra y es la única empresa local que ofrece servicios de localización y control vehicular a medida, además de desarrollar dispositivos electrónicos dependiendo de los requerimientos de sus clientes, la empresa tiene entre sus clientes el 90% de vehículos de transporte público de la ciudad de Ibarra instalados y operativamente funcionando su sistema del cual se maneja como un producto por servicios.

3.1.3. Análisis De Factores

La empresa INPRISE ofertará el dispositivo para su venta directamente con los directivos de las cooperativas, todo el hardware necesario será adquirido por el propietario y garantizando la transmisión de datos a su sistema actual de localización. Como el ingreso de pasajeros se verá reflejado en el actual sistema de rastreo se hace necesario aumentar la cuota mensual de pago por el servicio de transmisión de nuevos datos.

Para la fabricación del prototipo del producto se utilizará materia prima local, dependiendo de la producción inicial y la demanda se importará estos elementos del país que mejor ventaja económica brinde a la empresa. Todo el proceso de construcción y ensamblaje del dispositivo estará a cargo de INPRISE y solo se dependerá del stock de materia prima en las empresas de distribución de componentes electrónicos. El software se desarrollará como un módulo extra al sistema actualmente utilizado en el cual indique los datos recolectados.

3.2. Análisis del Mercado

En esta sección se describe todo el proceso de identificación de oportunidades del negocio, su tamaño y composición en el mercado, perspectivas de crecimiento, patrones de consumo y un pronóstico de demanda y ventas.

3.2.1. Diseño de la Investigación

El tipo de desarrollo de investigación para el presente estudio es de tipo **Descriptivo** ya que se necesita describir y analizar factores y variables que permitan relacionar el requerimiento de los potenciales clientes con el producto final para su adecuación al producto.

El análisis de estos factores ayudará a determinar los elementos del dispositivo (tamaños, modelos, peso, estructura), la producción y comercialización del mismo, para llegar de una manera sistemática y competente hacia los clientes.

3.2.1.1. Matriz de Relación de la investigación de negocios

A continuación se describe la matriz de la investigación de negocios que sirve para determinar el tipo de investigación.

Tabla 4. Matriz de relación de la investigación de negocios

FACTOR	VARIABLE	INDICADORES	TÉCNICA	FUENTE
Determinar factores del contador de pasajero en una unidad de transporte público.	Demanda	Rendimiento Control Ubicación Instalación	Encuesta	Socios transporte público de pasajeros
Determinar la competencia hacia el producto por empresas similares en la ciudad.	Oferta	Dispositivo Ubicación Control	Observación Directa	Empresas similares de la ciudad

Fuente: Información recopilada por el autor

3.2.1.2. Segmentación del mercado

La empresa INPRISE al desarrollar este dispositivo, su segmentación primaria del mercado está destinado al transporte público urbano de la ciudad de Ibarra, no obstante esto puede replicarse a varias ciudades de las provincias y el país, donde no se cuente con el sistema en las unidades de transporte.

3.2.1.3. Mercado objetivo

Por tratarse de un dispositivo específico para el transporte público de pasajeros, el mercado está destinado a los socios de las cooperativas de transporte público de la ciudad de Ibarra. El mercado se encuentra cerrado por su aplicabilidad y escalabilidad hacia los componentes ya instalados.

3.3. Realización de la Investigación de Mercados

3.3.1. Identificación de la población

La ciudad de Ibarra cuenta actualmente con 5 cooperativas de servicio público para el transporte de pasajeros de los cuales en el área urbana funcionan dos cooperativas las mismas que servirán para identificar la población a analizar.

Tabla 5. Número de socios de las cooperativas de transporte

COOPERATIVA	NÚMERO DE SOCIOS
28 de Septiembre	160
San Miguel de Ibarra	127

Fuente: Base de datos de las Cooperativas de Transporte Urbano de la Ciudad

Con estos datos se obtiene un universo de 287 clientes hacia el producto.

Con respecto al segundo factor, competencia de otras empresas, la cantidad de empresas dedicadas a la producción y venta de dispositivos electrónicos es mínima (menos de 5), el análisis se la realizara en su totalidad de manera descriptiva.

3.3.2. Cálculo de la muestra

Para la obtención de la muestra nos basamos en la siguiente formula

$$n = \frac{Z^2 d^2 N}{(e)^2 (N-1) + Z^2 d^2}$$

Fórmula 1. Cálculo de la muestra
Fuente: Tomado del Libro de Interaprendizaje de Estadística Básica

Simbología

N = Tamaño de la población

n = tamaño de la muestra

e = error máximo admisible en la muestra

d = heterogeneidad en la muestra

Z= nivel de confianza

Para nuestra población de 287 (total de socios de las cooperativas a ser encuestadas) clientes se ha dispuesto un error admisible de 0.1 una heterogeneidad de 0.5 y un nivel de confianza del 95% (1,96); se tiene como resultado que nuestra muestra a ser encuestada será de 73 personas.

El número de personas encuestas por cooperativa de transporte dependerá del porcentaje al que corresponden según su universo

Tabla 6. Número de encuestados por cooperativa

COOPERATIVA	NUMERO DE SOCIOS	PORCENTAJE	NUMERO DE ENCUESTADOS
28 de Septiembre	160	55.7 %	41
San Miguel	127	44.3 %	32
TOTAL	287	100 %	73

Fuente: Información recopilada por el autor

3.3.3. Desarrollo de la encuesta

A continuación se plantean las preguntas que serán evaluadas en la encuesta, este documento está basado en requerimientos del dispositivo hacia las normas técnicas de órganos de regulación y planteamientos de las directivas del transporte en la ciudad.

El formato es único y la encuesta es la misma sin diferencias a que cooperativa corresponda un socio.

ENCUESTA INPRISE

Soluciones tecnológicas empresariales

La siguiente encuesta nos servirá para desarrollar un producto que satisfaga las necesidades del transporte público relacionado con el sistema de conteo de pasajeros.

1. ¿Es necesario conocer la cantidad de pasajeros que ingresan a una unidad de transporte para determinar el rendimiento económico del mismo?

Si No

2. ¿Es necesario que el dispositivo funcione de forma independiente (sin intervención humana)?

Si No

3. ¿Cree necesario una señal sonora para identificar que una persona ingresa a la unidad?

Si No

4. ¿Cree necesario contar con una pantalla que indique el número de personas que ingresan a la unidad?

Si No

5. ¿Es necesario una señal sonora para indicar que una persona baja de una unidad de transporte?

Si No

6. ¿El reporte de personas que ingresan a la unidad deben ser?

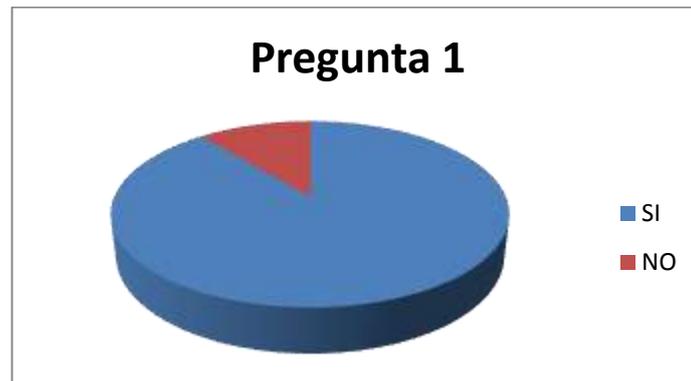
Diario Semanal Mensual

7. Cree necesario que el dispositivo sea compacto y desmontable con facilidad?

Si No

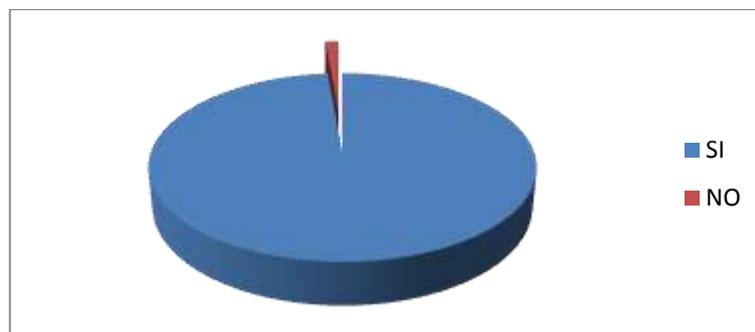
3.3.3.1. Resultados de la investigación de mercados

1. Del universo encuestado se obtiene un SI en un **89%** y como respuesta negativa un **11%**



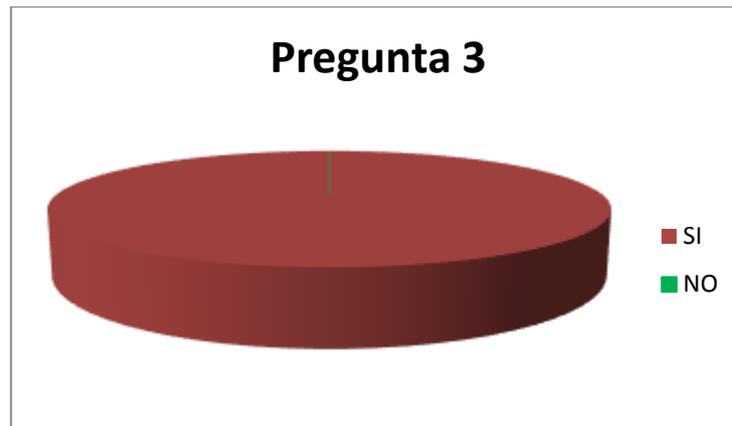
La muestra indica el nivel de prioridad que tienen los propietarios de los vehículos para saber cuántas personas ingresan a su vehículo y determinar el monto económico del mismo. Su respuesta es preponderante para la ejecución del proyecto ya que está determinada por la aceptación hacia el dispositivo.

2. A la segunda respuesta se obtiene una repuesta afirmativa en un **98,6%** mientras que un NO en **1,4%**



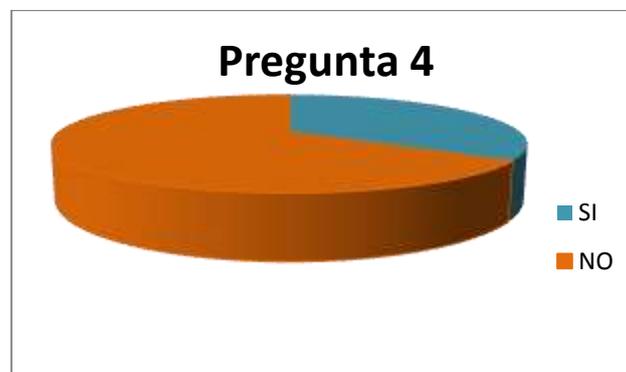
Por conversación directa hacia la pregunta y sabiendo el método de acceso a las unidades de transporte la respuesta es muy clara en que el dispositivo no tenga intervención humana y todos sus funciones las realice independientemente.

3. En la tercera pregunta se tiene una aceptación total 100% afirmativa



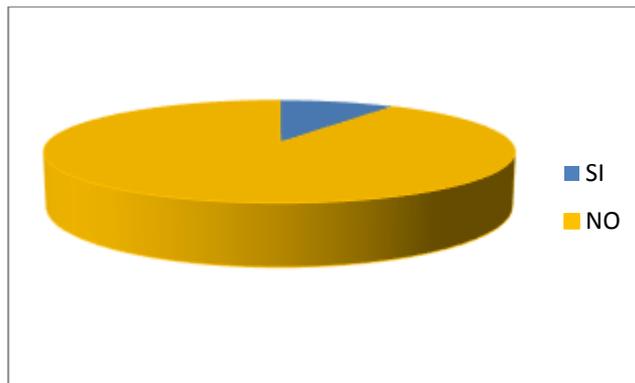
La respuesta afirmativa a esta respuesta responde a que el conductor del vehículo podría estar atento a señales sonoras para saber que el dispositivo está trabajando correctamente y a la vez detectar cuando el contador esta obstruido.

4. En la cuarta pregunta se tiene una respuesta afirmativa del 34,2% y negativa del 65,8%



La respuesta a esta pregunta describe el hecho de que con el reporte final de conteos en la unidad no es muy recomendable disponer de una pantalla, a la vez que abarataría el coste del precio final del dispositivo.

5. En la quinta pregunta se tiene una respuesta afirmativa del 9,5% y negativa del 90,5%



Con la señal sonora de ingreso son suficiente las advertencias sonoras por lo cual no tiene aceptación esta pregunta.

6. Para la sexta pregunta se tiene una aceptación al resultado diario de 91,7 % semanal del 5,5% y mensual del 2,8%



El sistema económico de pagos de los propietarios de los vehículos se lo realiza diariamente (pago de multas por atrasos y aportaciones a la cooperativa), por lo cual se debe tomar en cuenta para que los reportes sean entregados diariamente.

7. En la séptima pregunta se tiene una respuesta afirmativa del 94.5% y negativa del 5.5%



Por requerimiento de los propietarios de los vehículos. El dispositivo deberá ser compacto y de fácil montaje porque muchas veces el vehículo puede entrar en mantenimiento o el sistema deberá ser instalado en otra unidad.

3.3.4. Análisis de la Demanda

Con los resultados obtenidos se puede determinar que si existe demanda por la utilización de un contador de pasajeros en las unidades de transporte público, los resultados obtenidos demuestran el gran grado de aceptación por el producto tomando en cuenta que en las encuestas se describió algunas características que dispondrá el dispositivo.

3.3.4.1. Demanda Actual

La demanda actual consta del número total de socios de las cooperativas de transporte público tomando en cuenta la pregunta uno de la encuesta realizada (sección 3.3.3) en el cual se hace referencia al uso del dispositivo.

Tabla 7. Demanda actual

DEMANDA POTENCIAL Socios de cooperativas	% DE ACEPTACION DEL SERVICIO	DEANDA REAL
287	98.6%	283

Esta demanda es relativa ya que como se analizará en la sección de marketing el producto se ofrecerá a todo la cooperativa como un producto por servicios hacia todos los socios por disponer ya instalada una parte del sistema y no como un producto individual.

3.3.4.2. Pronóstico de ventas

El proyecto está destinado a un sector específico de mercado en una localidad geográfica por lo cual la proyección de venta será evaluada como única y solo se explotará el costo por servicio indefinido, para futuras ventas se pude analizar otro sector geográfico del mismo mercado específico para aplicar los resultados del presente informe.

3.4. Plan de Marketing

En esta sección se describe de una forma estructurada los objetivos de comercialización en un periodo de tiempo, tomando en cuenta los factores externos e internos de la empresa y su enfoque con el producto.

3.4.1. Análisis de la situación

El mercado para la comercialización del sistema de conteo de pasajeros está limitado a propietarios de vehículos de transporte público de pasajeros. Este mercado está fraccionado dependiendo de la asociación entre cooperativas en una determinada ciudad.

La comercialización del producto se la realiza a nivel de cooperativas como una solución integral para sus unidades y no como un producto individual por socios, cabe recalcar que este producto es un complemento al sistema de control y localización ofertado por la empresa.

3.4.2. Datos del mercado

La comercialización inicial del producto como se ha planteado anteriormente está enfocada a las cooperativas de transporte público de pasajeros de la ciudad de Ibarra ya que estas disponen actualmente del sistema de control y localización vehicular.

Las cooperativas actualmente por el servicio de control y localización pagan una

cantidad mensual hacia la empresa, pues esta ofrece el servicio de almacenamiento y desarrollo de las aplicaciones que utilizan las cooperativas más el servicio de soporte técnico hacia el software y los equipos instalados en las unidades.

El producto ofertado comprende dos fases de comercialización, la primera es la venta del dispositivo como hardware el cual adquiere el propietario de la unidad de transporte y se convierte en el propietario del dispositivo; la segunda es el pago por el servicio de transmisión de los datos, para este punto como las unidades del mercado objetivo principal ya cuentan con el servicio base se hará un reajuste de precio para el nuevo dispositivo.

3.4.2.1. Amenazas y Oportunidades (Análisis A/O)

Oportunidades de la empresa

- Con el cambio de procesos en el manejo de recursos de las cooperativas de transporte público se hace más fácil y manejable el ofertar un complemento hacia su actual sistema.
- Llevar las directivas de la empresa al lineamiento que demanda el transporte público y las normativas de los GADs.

Principales amenazas de la empresa

- Ofertas otras de empresas de otras ciudades.
- Cambio de normativas en el sector público de transporte

3.4.2.2. Fortalezas y Debilidades (Análisis F/D)

Las principales fortalezas son:

- Nombre de la empresa reconocida localmente por proveedor servicios informáticos y electrónicos.
- Servicio de control y localización instalado y funcionando correctamente.
- Servicio Técnico local garantizado; avalado por los actuales clientes
- Comercialización directa entre la empresa y los dirigentes de las cooperativas.

Las principales debilidades que se afronta la empresa son:

- Poca publicidad como empresa hacia productos actuales
- Producción limitada de hardware electrónico.

3.4.3. Estrategia de Marketing Local

La comercialización del producto se realizará de manera personalizada en la ciudad de Ibarra, el producto se comercializa como un complemento al actual sistema de rastreo que cuentan las unidades de transporte urbano de la ciudad.

Al ser un complemento del sistema que actualmente poseen, las condiciones de venta del producto consisten en un pago inicial por instalación del hardware nuevo y el aumento en el coste mensual por servicio del que ya vienen cancelando.

La oferta del producto y el detalle de todos sus componentes y beneficios se la realizarán en 2 etapas, la primera etapa será una explicación detallada a los consejos administrativos de las cooperativas, previa a su aprobación, se realizará la siguiente etapa que consiste en la explicación funcional del sistema a todos los socios de las cooperativas.

Los elementos de promoción hacia los clientes por tratarse de una venta directa se consideran la explicación mediante exposiciones en la que el mayor recurso será el audiovisual, además de brindar catálogos con las especificaciones técnicas del producto.

3.5. Plan de Fabricación y Operación

El plan de fabricación a ser mencionado describe el proceso de venta directa del sistema a un sector específico de mercado por lo cual la producción es limitada, esta será dimensionada para este propósito; pero podrá servir para próximas producciones; se considerara la producción inicial de 100 dispositivos.

El desarrollo y fabricación del sistema será elaborado en las oficinas de la empresa INPRISE, los diferentes procesos pueden ser elaborados en su totalidad por el personal de la empresa, se analizará la posibilidad de disminuir procesos y costos con la reducción de procesos ya que pueden ser adquiridos ya ensamblados.

Los diferentes procesos que se desarrollaran para la obtención del producto final se describen a continuación:

- **Diseño del sistema electrónico del conteo de pasajeros**

En este proceso se desarrollará el software y hardware necesario para que el sistema a ser instalado en los vehículos funcione autónomamente. El hardware debe contener las interfaces de comunicación y alimentación necesarias para que se adapte a los vehículos.

Los componentes deben estar montados sobre una placa de circuito impreso para

que el dispositivo sea compacto y resistente a movimientos. El software que será grabado en los componentes electrónicos que manejan los periféricos también deberá ser desarrollado por la empresa.

- **Fabricación de placas de circuito impreso**

Una vez determinado el diseño final del hardware del sistema el siguiente paso es construir las placas con los circuitos impresos y los componentes necesarios, este proceso por tratarse de pasos consecutivos y repetitivos se deberá automatizar de manera eficiente para ahorrar tiempo y dinero.

- **Software de gestión remota y representación de conteos.**

Se desarrollara un módulo extra al actual sistema de monitoreo y control para que este muestre los datos adquiridos por el contador de pasajeros, el desarrollo del software se realizara con las mismas bases y aplicaciones del software que se encuentra en funcionamiento.

- **Ensamblaje del sistema**

El sistema para ser instalado necesita estar compactado y listo para funcionar, para esto se procederá a ensamblar todos los componentes en una sola caja. En ensamblaje

consiste en la compactación de todos los componentes y su interfaz eléctrica para la alimentación de la corriente del vehículo.

- **Instalación del sistema**

La instalación del dispositivo comprenderá la ubicación del dispositivo en el vehículo y la adaptación a las conexiones eléctricas del vehículo, este pasó será el último proceso de hardware en el sistema. Cabe mencionar que para la instalación los vehículos deberán trasladarse hacia las oficinas de la empresa, el sistema debe ser instalado por personal de la empresa para certificar su correcto funcionamiento y garantía.

No existe proceso de transporte del sistema ya que solo la materia prima será ingresada a la empresa.

3.5.1. EQUIPOS O HERRAMIENTAS A SER UTILIZADAS

Tabla 8. Equipos a ser utilizados

EQUIPOS O HERRAMIENTAS	CANTIDAD (UNIDADES REQUERIDAS)
Computador	2
Cortadora	3
Estación de soldadura	3

Fuente: Información recopilada por el autor

3.5.2. Materia prima para la construcción

Tabla 9. Materia prima para la construcción

MATERIA PRIMA INSUMOS	CANTIDAD (UNIDADES REQUERIDAS)
Láminas de baquelita PCB	25
Estaño	20 m

Fuente: Información recopilada por el autor

3.5.3. Número de personas para la producción

Tabla 10. Personal necesario para la producción

PUESTO DE TRABAJO	NUMERO
Desarrollador software	1
Desarrollador hardware	3
Instalador	2

Fuente: Información recopilada por el autor

3.5.4. Tareas del personal para producción

Tabla 11. Tareas del personal de producción

PUESTO DE TRABAJO	TAREAS DEL PUESTO
Desarrollador software	Desarrollo de la aplicación de recepción de datos del contador. Desarrollo del firmware para los componentes electrónicos
Desarrollador hardware	Construcción de placas PCB del contadores Ensamblaje de componentes
Instalador	Instalación del dispositivo en el vehículo

Fuente: Información recopilada por el autor

3.6. Costos de Producción y Financiamiento

A continuación se detalla todos los ingresos, egreso, estados financieros para el desarrollo y ejecución del proyecto, como se describió en anteriores secciones no existe proyección de venta sino se ejecutara como una sola construcción y venta del dispositivo.

3.6.1. Proyección del Precio de Venta

Los siguientes datos describen todos los valores necesarios para la fabricación del dispositivo el cual nos ofrecerá el precio final del mismo. Cabe mencionar que el proceso de instalación para los vehículos tendrá un tiempo de duración de 2 meses, en el que se requerirá mano de obra extra.

- **Costos materia prima**

En la tabla a continuación se muestra los materiales necesarios y su precio final con un coste por unidad; para representación de análisis posteriores se calculara el precio total del proyecto hacia una venta directa.

Tabla 12. Costos materia prima

DESCRIPCION	VALOR
Elementos electrónicos	200
Sensores infrarrojos Sharp Placa Arduino Mega 2560 PIC 16F88 Baquelita de fibra A4 Regulador de voltaje 7805 Relay 5V Transistor 2N3904 MAX 232 Borneras de 2 contactos Espadines machos Potenciómetro 5K Condensadores 10uF Socalo 18 pines Socalo 16 pines Resistencias 10K Buzzer 12V Leds	
Materiales de ensamblaje	60
Materiales de instalación	100
TOTAL	360

Fuente: Información recopilada por el autor

- **Costo de mano de obra**

En esta tabla se describe la mano de obra que se necesitó para desarrollar el dispositivo en su fase de diseño, programación y construcción y el tiempo necesario para el proceso de instalación.

Tabla 13. Costos de mano de obra

DESCRIPCIÓN	CANTIDAD	TIEMPO / DIAS	COSTO
Persona de diseño	1	20	500
Persona de construcción	1	10	50
Persona de instalación	1	1	20
TOTAL			570

Fuente: Información recopilada por el autor

Esta tabla de igual manera representa el valor unitario de los costos del dispositivo, se debe tomar en cuenta que el costo no es directo hacia el empleado ya que este obtiene un salario total por esta y otras actividades realizadas en la empresa.

- **Costo por servicios básicos.**

Se toman en cuenta un valor porcentual al mes de los servicios utilizado por toda la empresa.

Tabla 14. Costos por servicios

DESCRIPCION	TIEMPO/MESES	COSTO
Agua	3	6
Luz	3	15
Internet	3	30
	TOTAL	51

Fuente: Información recopilada por el autor

- **Costos por mantenimiento de servidores y transferencia de datos.**

Este coste es un adicional al servicio que actualmente se cobra a las unidades, los valores a continuación son determinados como servicios por un mes para la aplicación en el proyecto serán valores proporcionales ya que la infraestructura de comunicación donde se desarrolla este proyecto se encuentra operativa.

Tabla 15. Costos de servidores y transferencia de datos

DESCRIPCION	TIEMPO/MESES	COSTO (\$)
Servidor de Aplicaciones	1	20
Hosting	1	15
Transmisión GPRS	1	10
	TOTAL	45

Fuente: Información recopilada por el autor

De acuerdo a estos análisis se puede deducir que el costo de fabricación, producción e instalación del dispositivo es de \$1026 dólares, a este valor se debe añadir un margen de error y el porcentaje de ganancia. En este documento no se detalla el precio final del dispositivo, la empresa se reserva los costos finales del sistema por futuras negociaciones con las cooperativas de transporte.

3.6.2. Financiamiento

La empresa INPRISE toma la decisión de invertir en este dispositivo por los resultados obtenidos de las conversaciones con dirigentes y socios de las cooperativas y su proceso de innovación en el campo tecnológico expuesto en anteriores páginas. Con este

antecedente todo el costo necesario de materiales y tiempo para el diseño del prototipo del dispositivo son financiadas por la empresa y luego con la venta del producto y el cobro por servicios indefinidos se compensará la inversión.

Una vez que se negocie la venta del dispositivo con las cooperativas se firmará un contrato en el cual se debe detallar que es necesario el 60% de anticipo por razones de materia prima y el 40% a contra entrega cuando el sistema este correctamente instalado y funcionando en los vehículos.

DISEÑO

4.1.Descripción del Dispositivo

El contador de pasajeros es un dispositivo electrónico autónomo, el cual se encarga de determinar si una persona ingresa o sale de un vehículo mediante sensores de presencia, este contador estará conectado a un dispositivo GPS/GPRS el cual estará encargado de enviar los datos adquiridos por el contador a un servidor de aplicaciones tomando en cuenta una referencia de ubicación para enviar los datos, los datos no serán enviados individualmente cada vez que sean registrados, sino una sumatoria de los mismo dependiendo de la distancia que haya recorrido el vehículo.

En la figura 17 se detalla el funcionamiento del sistema empezando desde el conteo de pasajeros de forma autónoma, este dispositivo se comunicará a través de comunicación serial con un controlador el cual almacenará el número de conteos y determinara el momento en que estos datos deban ser enviados, para enviar los datos el controlador se comunicara igualmente por comunicación serial hacia el dispositivo GPS/GPRS ya instalado en el vehículo el cual enviará los datos del conteo mediante la red de telefonía celular; en el servidor de aplicaciones mediante un socket de comunicación filtrara los resultados y los mostrara en la aplicación ya existente en el cual además de mostrar la ubicación del vehículo mostrara la cantidad de pasajeros ingresos a la unidad.

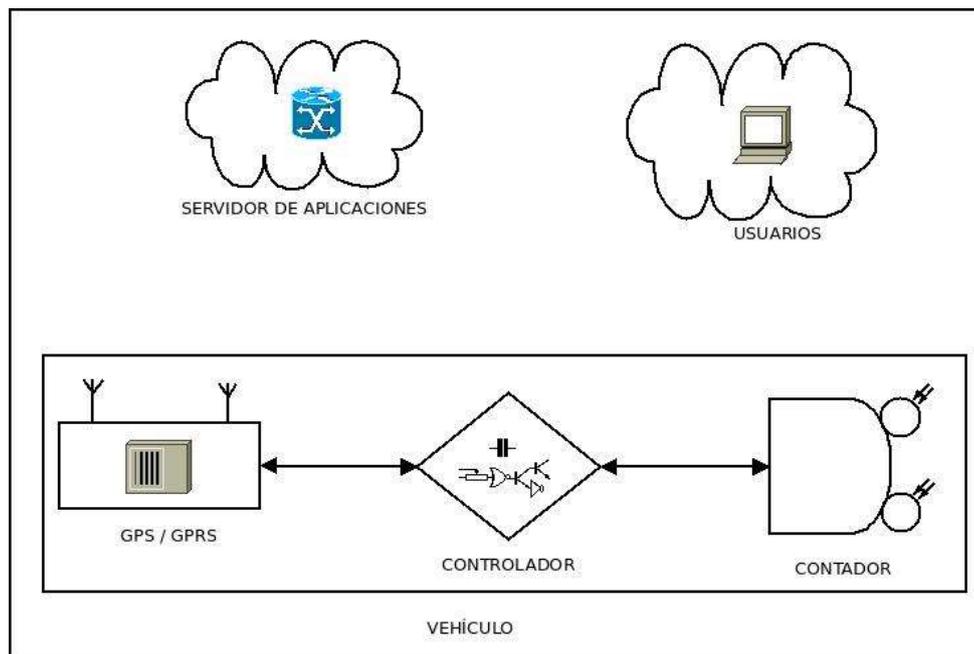


Figura. 17 Bloques de Funcionamiento
Fuente: Autor

En la figura 16 se encuentra representado la parte lógica del conteo y transmisión de los datos de un vehículo que se traslade de la ubicación A hacia la B ; suponiendo que un vehículo se encuentre estacionado en la ubicación E1 y mientras no tenga movilidad él envió de información no se realizará, pero sí se estará guardando el conteo de pasajeros que suban mientras el vehículo este estacionado; una vez que el vehículo comience a desplazarse y llegue al punto D1 donde ha completado un distancia o una velocidad X enviará información hacia el servidor la cual tendrá el número de pasajeros más las coordenadas del punto D1, la distancia recorrida y el contador de envió volverá a cero mas no el contador general; el vehículo se desplazará recogiendo pasajeros por su trayectoria en los puntos P1, P2, P3 en estos puntos sumará la cantidad de pasajeros y espera hasta una

distancia X (llegar al punto D2) para enviar la nueva sumatoria de pasajeros y la ubicación; este proceso se repetirá mientras el vehículo este en movimiento; para el caso que el vehículo se desplace de D2 hacia D3 en el cual existe movimiento pero no conteo de pasajeros el dispositivo no enviará información de conteo ni ubicación del punto D3.

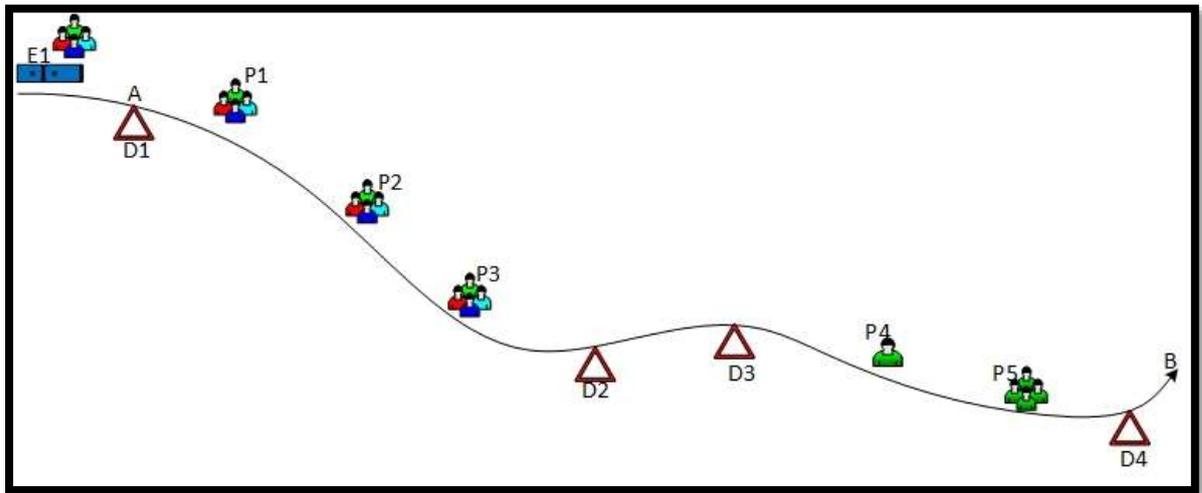


Figura. 18 Representación del conteo lógico
Fuente: Autor

El servidor de aplicaciones será el encargado de realizar las sumatorias y reportes de los conteos, además del filtrado de tiempos de periodos validos de conteo y ubicaciones, tomando él cuenta la velocidad de procesamiento del módulo ARDUINO, además para no sobrecargar el tamaño de la información que se envía por el módulo GPRS.

A continuación se describe el proceso lógico y físico a realizarse en el sistema, el mismo que ha sido separado por etapas, las cuales comprenden el diseño y construcción del contador de pasajeros, diseño y construcción de controlador, acople al módulo

GSM/GPRS y recepción y almacenamiento de resultados.

4.2. Contador de Pasajeros

El diseño del contador de pasajeros basa su funcionamiento en determinar la distancia de un objeto hacia el sensor, cuando esta medida se cumple se determina que hay un objeto en el sensor. El contador debe determinar el sentido en el que una persona pasa por el contador por lo cual es necesario contar con 2 sensores infrarrojos que ayudaran a determinar el sentido de paso dependiendo de la secuencia de activación, el proceso está considerado para una sola puerta, pero este puede ser replicado en el número de puertas necesarias ya que su funcionamiento no varía.

4.2.1. Sensor SHARP modelo GP2Y0A21YK

Este sensor es pasivo y sirve para determinar la distancia desde el sensor hacia un objeto, dispone de un solo pin para la transferencia de información siendo este de tipo análogo el cual entregara un nivel de voltaje dependiendo de la medición realizada.

Este sensor puede medir entre 10 y 80 cm dando niveles de voltaje de 3.1V a 10cm hasta 0.4V a 80cm, funciona con una alimentación de 5V y sus pines pueden ser conectados directamente hacia el sistema microcontrolado para su análisis.

Tabla 16. Especificaciones sensor SHARP

ESPECIFICACIONES				
Alimentación (recomendado)	4.5 a 5.5 V			
Temperatura de operación	-10 a 60 oC			
	MIN.	TYP.	MAX.	UNID
Rango de medición de distancia	10	--	80	cm
Pin de salida de medición L=80cm	0.25	0.4	0.55	V
Diferencia de salida de voltaje	1.65	1.9	2.15	V
Consumo promedio de corriente L=80cm	--	30	40	mA

Fuente: Hoja de datos del sensor

Gráfico de relación distancia-voltaje

El grafico a continuación describe la relación entre el nivel de voltaje y la distancia del objeto a ser medido con respecto al sensor, la relación es inversamente proporcional a mayor distancia menor nivel de voltaje y a menor distancia el nivel de voltaje aumenta a su punto máximo de operación.

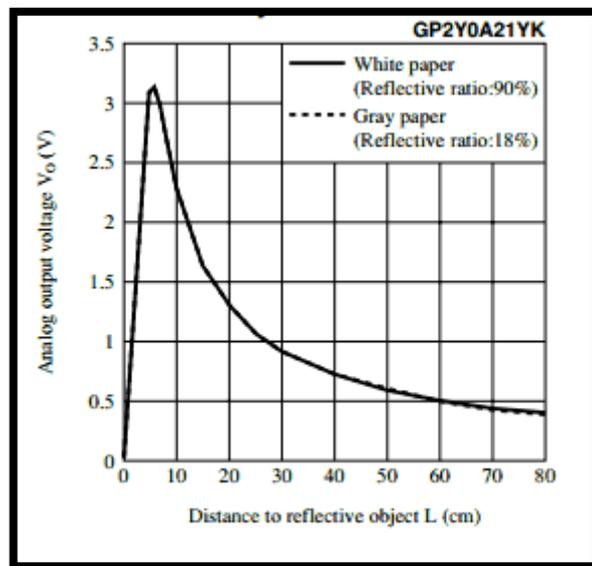


Figura. 19 Relación voltaje-distancia sensor SHARP

Fuente: Hoja de datos del sensor

4.2.2. Microcontrolador PIC 16F88

Para la elección del microcontrolador se tomó en cuenta los conocimientos adquiridos en clase, determinando requerimientos mínimos y costo.

El PIC 16f88 es un microcontrolador de microchip de gama baja de 8bits, las características más especiales se describen a continuación:

- Arquitectura Harvard; memoria de código de 14bits.
- 2 puertos digitales; puerto A de 5bits y puerto B de 8bits
- Oscilador LP,XT y HS hasta 20Mhz
- Módulo PWM/CCP
- ADC de 10bits 7 canales
- USART
- Características de bajo consumo (tecnología nanoWatt)

Tabla 17. Características PIC 16F88

Memoria de programa		Memoria de datos		I/O pines	Timmers 8/16bit
Flash (bytes)	Instrucciones	SRAM (bytes)	EEPROM (bytes)		
7168	4096	368	256	16	2/1

Fuente: Información recopilada por el autor

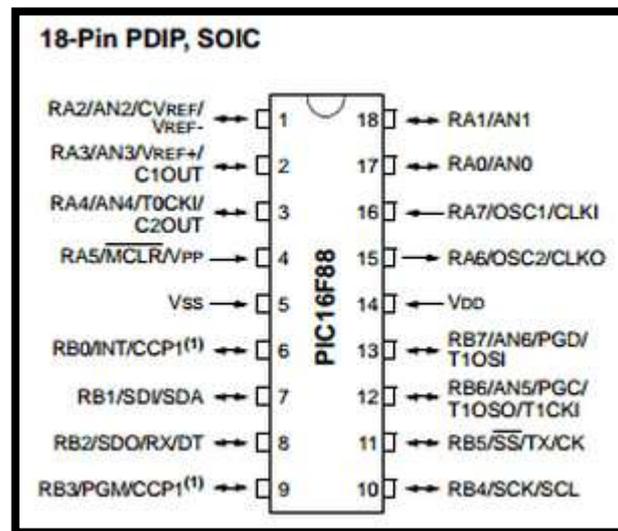


Figura. 20 Diagrama de pines
Fuente: Datasheet PIC 16F88

4.2.2.1. Entradas analógicas del microcontrolador

El microcontrolador 16f88 dispone de 7 entradas analógicas de 10 bits, suficientes para este propósito las cuales son configuradas en las sentencias de programación. Cada entrada dispone de un canal independiente de toma de datos, estos datos son almacenados como una variable de tipo “float”, con este valor se procede a realizar el cálculo de conversión para determinar la distancia.

Primero se determinó el voltaje que envía el sensor y luego con los datos proporcionados en el datasheet del sensor calculamos la distancia dependiendo del nivel de voltaje con una regla de 3 inversas.

$A/D = 10 \text{ bits} = 1023$ // número máximo de bits

$5V = 1023 \text{ bits}$ // número de bits para el voltaje máximo

$0V = 0 \text{ bits}$ // número de bits para el voltaje mínimo

$\text{voltaje} = (\text{lectura_del_conversor} * 5) / 1023$ //regla de tres inversa

$10\text{cm} = 3,1V$ // valores proporcionados por la hoja de datos del sensor

$80\text{cm} = 0.4V$ // para la lectura de máxima y mínima distancia

$\text{distancia} = (0,4 * 80) / \text{voltaje}$ // formula descrita en la hoja de datos del sensor para determinar la distancia

4.2.3. Conexión Sensores al Microcontrolador

El sensor infrarrojo SHARP va conectado directamente al microcontrolador el mismo que se encarga de recibir la información analógica por un pin e interpretar las mediciones que realice el sensor. El sensor es alimentado con 5V los mismos que se le suministran al microcontrolador y de los cuales se hace referencia para los cálculos respectivos.

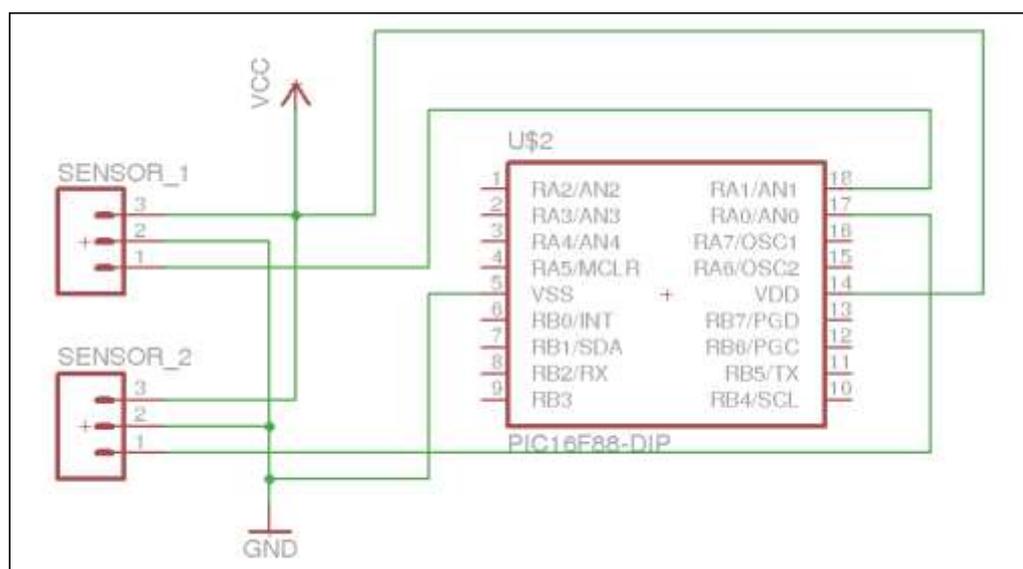


Figura. 21 Conexión de sensores SHARP al microcontrolador
Fuente. Autor

4.2.4. Conexión de Componentes

El nivel de voltaje en un vehículo es de 12V por lo cual se debe conectar un regulador de voltaje para que el nivel sea de 5V, para eliminar ruido provocado por el vehículo se debe colocar 2 capacitores de 10uF a la entrada y salida del regulador los mismos que son recomendados en la hoja de datos del regulador de voltaje.

Para la señal sonora se conectará un buzzer en una entrada digital del microcontrolador, la corriente que maneja un pin del microcontrolador no es suficiente para que la señal del buzzer sea la adecuada para esto se dispone de un transistor 2N3904 que amplificara la señal para que el sonido sea el correcto.

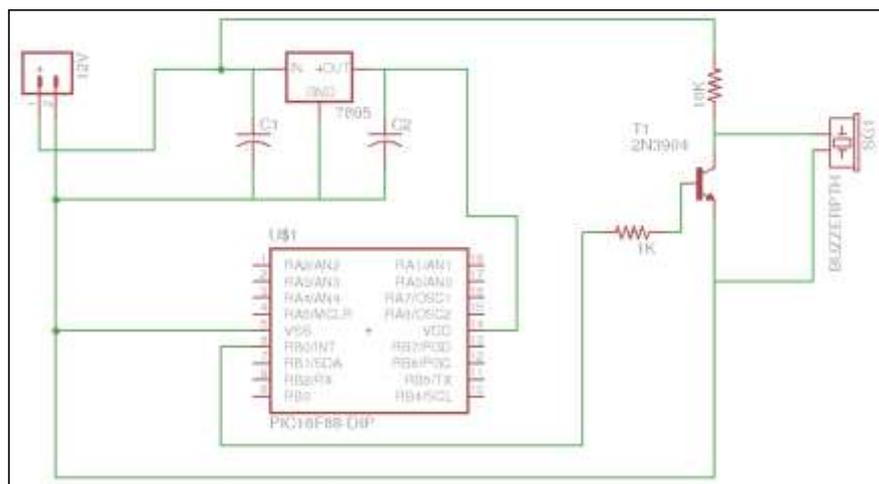


Figura. 22 Conexión de componentes
Fuente: Autor

4.2.5. Envío de Información

Para el envío de información hacia el controlador ARDUINO se utilizara comunicación serial, mediante el módulo USART disponible en el microcontrolador la conexión es directa entre los pines RX y TX.

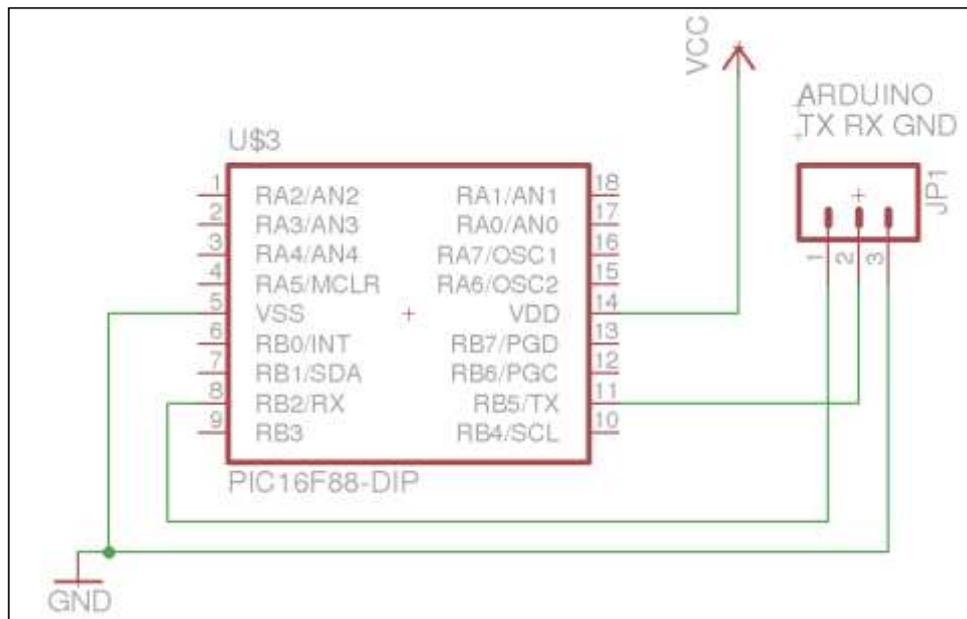


Figura. 23 Conexión USART entre PIC y ARDUINO

Fuente: Autor

Se configura una conexión con los siguientes parámetros

```

velocidad = 9600 baudios
    bits = 8
    paridad = N
    bits de parada = 1

```

En el microcontrolador uso la directiva del compilador en la sección de configuración.

```
#use RS232 (baud=9600, xmit = pin_c6, rcv = PIN_c7, parity=N, bits=8)
```

4.2.6. Programación del Microcontrolador 16F88

La programación se la realizará en lenguaje C con el compilador SDCC, el IDE de programación es PikLab y está basado en software libre, el compilador SDCC de igual manera es software libre por lo que no es necesario adquirir una licencia y su uso es de propósito general.

4.2.6.1. Algoritmo del programa

Para la actuación sistemática de la lectura de sensores y la información hacia arduino se debe crear un algoritmo que ejecute pasos de la siguiente manera:

- Configuración de puertos, variables e interrupciones
- Configuración de comunicación serial
- Lectura del puertos analógicos para determinar que sensor se activo
- Dependiendo del sensor que se active esperara la activación del siguiente sensores
- Si solo se activa un solo sensor por un determinado tiempo, envía mediante comunicación serial la indicación de que el sensor esta obstruido.
- Determina si el pasajero ingresa o sale del vehículo

- Dependiendo de la activación de los sensores se procede a enviar la información con la indicación si el pasajero ingresa o sale del vehículo
- Una vez enviado la información los sensores vuelven a modo de lectura
- El proceso se repite cada vez que se active la medición de los sensores.

4.2.6.2. Flujograma principal

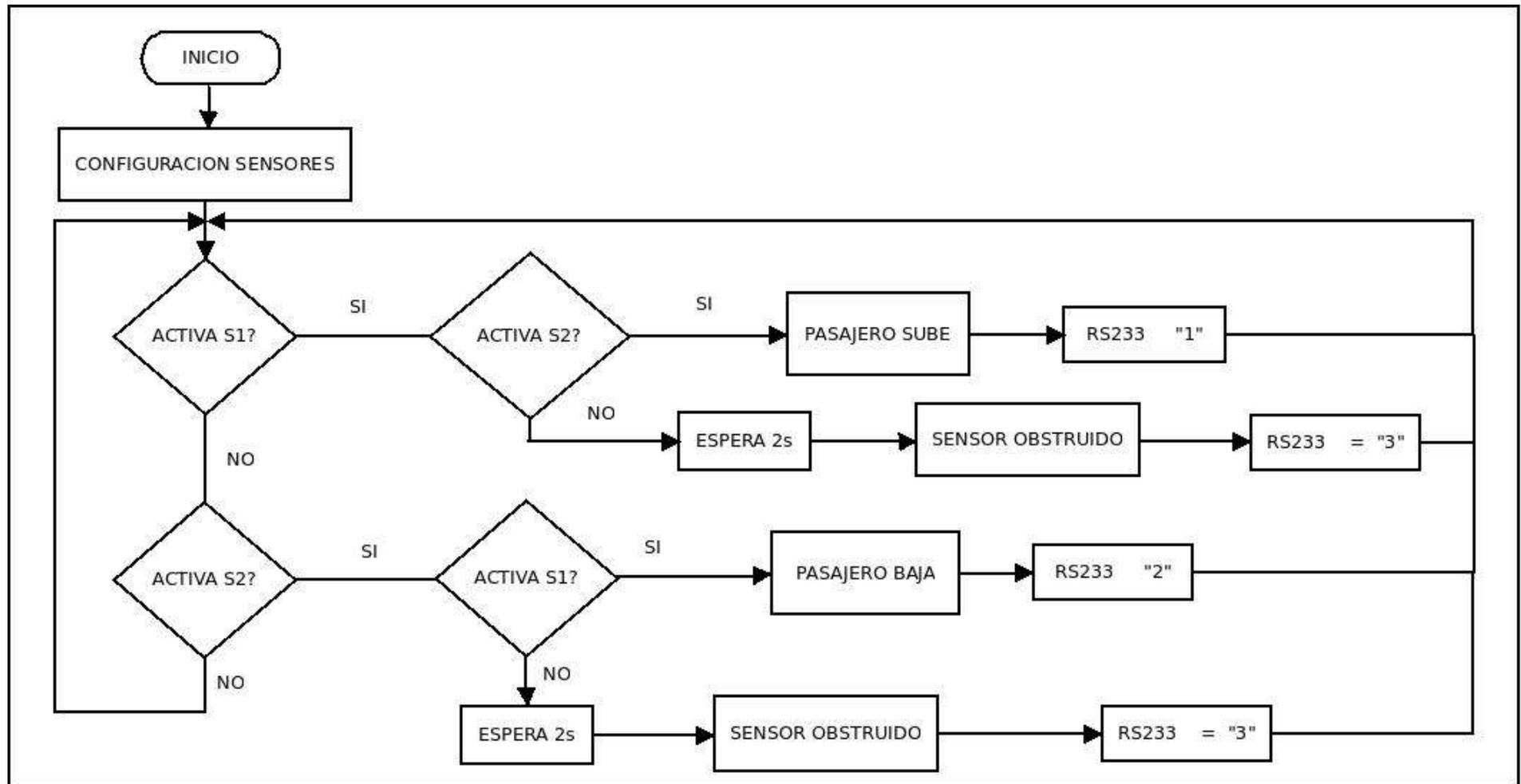


Figura. 24. Flujograma principal

4.3. Controlador

Este dispositivo será el encargado de almacenar la información del contador de pasajero y analizarla según los datos adquiridos del dispositivo GPS, una vez que se cumplan las condiciones para él envío de información el dispositivo se encargara de enviar la información al módulo GPRS para que este envíe los dato al servidor de aplicaciones.

4.3.1. ARDUINO MEGA2560

El ARDUINO Mega es una placa microcontrolada basada en el ATmega2560. Tiene 54 entradas/salidas digitales (de las cuales 14 proporcionan salida PWM), 16 entradas digitales, 4 UARTS (puertos serie por hardware), un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reset. El Mega2560 es compatible con la mayoría de shields diseñados para el ARDUINO Duemilanove o Diecimila.

Tabla 18. Características Mega2560

Resumen de características	
Microcontrolador	Atmega 2560
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limite)	5-20V
Pines E/S digitales	54 (15 proporcionan salida PWM)
Pines de entrada analógica	16
Intensidad por pin	40 mA
Intensidad en pin 3.3V	50 mA
Memoria FLASH	256 Kb (8Kb usados por el bootloadear)
SRAM	8 Kb
EEPROM	4 Kb
Velocidad del reloj	16 Mhz

Fuente: Información recopilada por el autor

El ARDUINO Mega puede ser alimentado vía la conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente.

La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7V el pin de 5V puede proporcionar menos de 5 Voltios y la placa puede volverse inestable, si se usan más de 12V los reguladores de voltaje se pueden sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

4.3.1.1. Memoria

El ATmega2560 tiene 256KB de memoria flash para almacenar código (8KB son usados para el arranque del sistema (bootloader). 8 KB de memoria SRAM y 4KB de EEPROM , a la cual se puede acceder para leer o escribir con la librería EEPROM.

4.3.1.2. Entradas y Salidas

Cada uno de los 54 pines digitales en el Mega pueden utilizarse como entradas o como salidas usando las funciones `pinMode()`, `digitalWrite()`, y `digitalRead()` . Las E/S operan a 5 voltios. Cada pin puede proporcionar o recibir una intensidad máxima de 40mA y tiene una resistencia interna (desconectada por defecto) de 20-50kOhms. Además, algunos pines tienen funciones especializadas:

4.3.1.3. Comunicaciones

EL ARDUINO Mega2560 facilita en varios aspectos la comunicación con el ordenador, otro ARDUINO u otros microcontroladores. El ATmega2560 proporciona cuatro puertos de comunicación vía serie UART TTL (5V). Los LEDS RX y TX de la placa parpadearan cuando se detecte comunicación transmitida través del chip FTDI y la conexión USB (no parpadearan si se usa la comunicación serie a través de los pines 0 y 1).

La libreria SoftwareSerial permite comunicación serie por cualquier par de pines digitales del ARDUINO Mega. (Arduino - Traducción)

4.3.2. Conexión Eléctrica

La placa ARDUINO funciona con una alimentación de 12V, que serán proporcionados directamente por la alimentación del vehículo. Para asegurarnos que la alimentación no exceda los 12V soportados por la placa, los voltajes en los vehículos no son estables y sufren variaciones constantemente por lo cual se conectará un regulador de 12V y a continuación el conector hacia la placa.

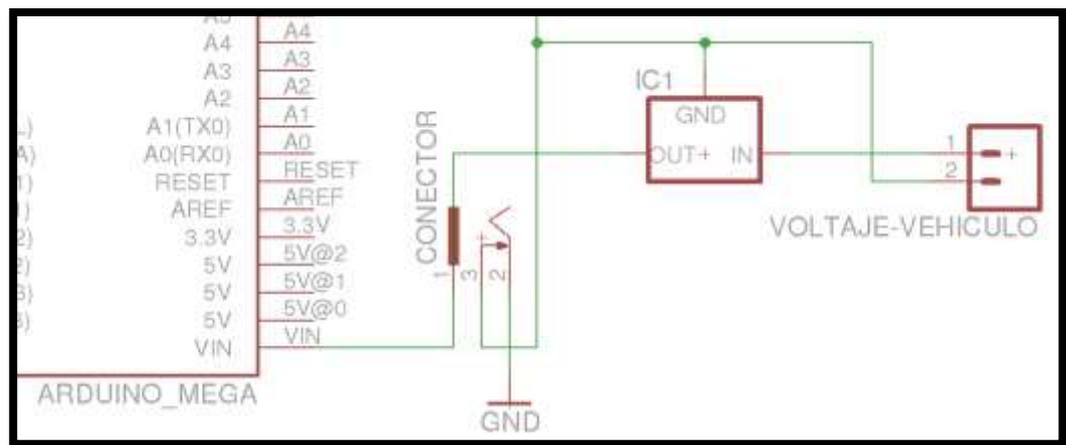


Figura. 25 Conexión alimentación ARDUINO

Fuente: Autor

4.3.3. SHIELD para la placa ARDUINO Mega

Para facilitar la conexión con los elementos a controlarse es necesario desarrollar una placa que se adapte a la placa ARDUINO (comúnmente conocida como shield) que nos sirva de interfaz física entre los dispositivos a controlarse, y disponer de elementos electrónicos como el circuito integrado max3232 necesario para la comunicación entre el ARDUINO y SKYPATROL, el circuito acondicionador del pulso para activar envío de información y borneras de conexión.

Este shield es una placa de circuito impreso que se adaptará a los pines del módulo ARDUINO, estos pines se conectarán a las interfaces seriales y a los pines digitales para activar la emisión.

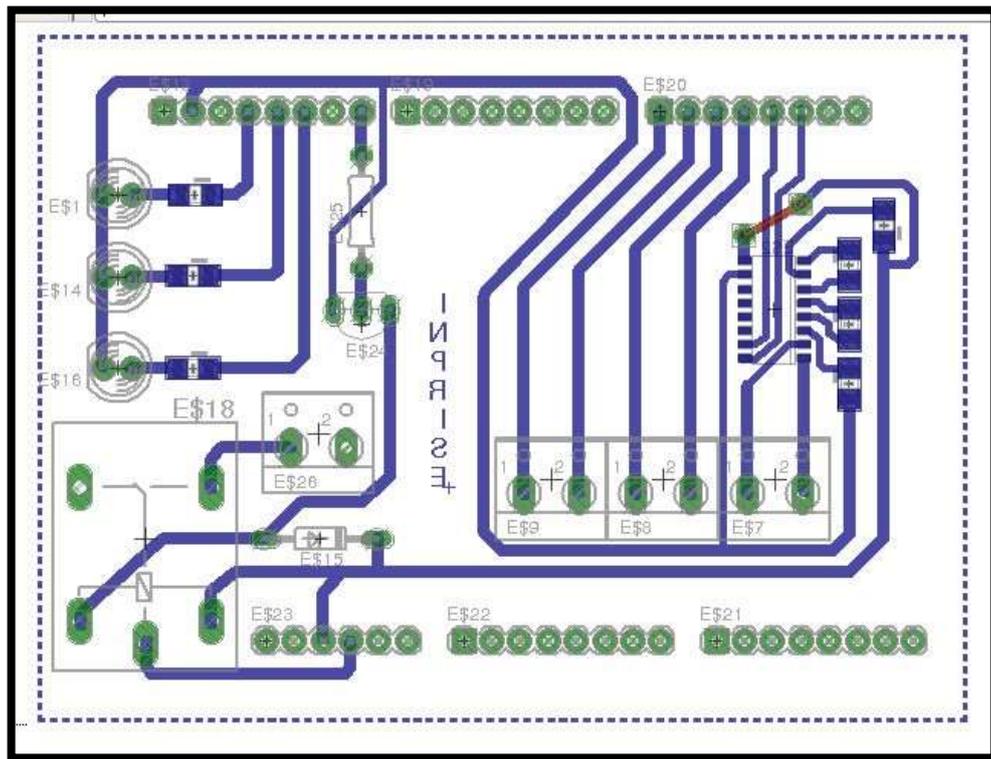


Figura. 26 Diseño PBC del controlador para ARDUINO
Fuente: **Autor**

4.3.3.1. Circuito Integrado MAX 3232

Este circuito sirve para invertir las señales seriales para que sean procesadas por el módulo ARDUINO ya que el equipo SKYPATROL envía la información serial de forma invertida. Este circuito necesita de 4 condensadores de 0.1uF para su funcionamiento los mismos que son recomendados por el fabricante descritos en su hoja de datos, este circuito integrado solo sirve para este propósito por lo cual no es configurable, su conexión es estándar y es detalla en el datasheet del producto como se muestra en la figura.

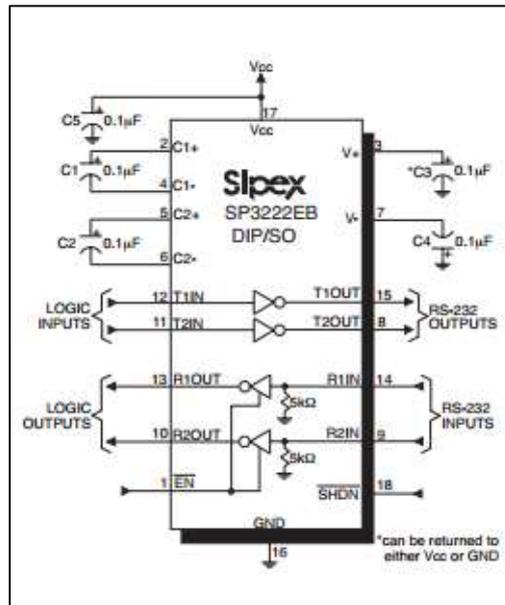


Figura. 27 Configuración max3232
Fuente: Datasheet MAX 3232

Existen en el mercado varias versiones de este circuito integrado, se ha optado por esta versión por el tipo de encapsulamiento SMD ya que reduce la disposición física en el shield además que los condensadores son de igual manera SMD.

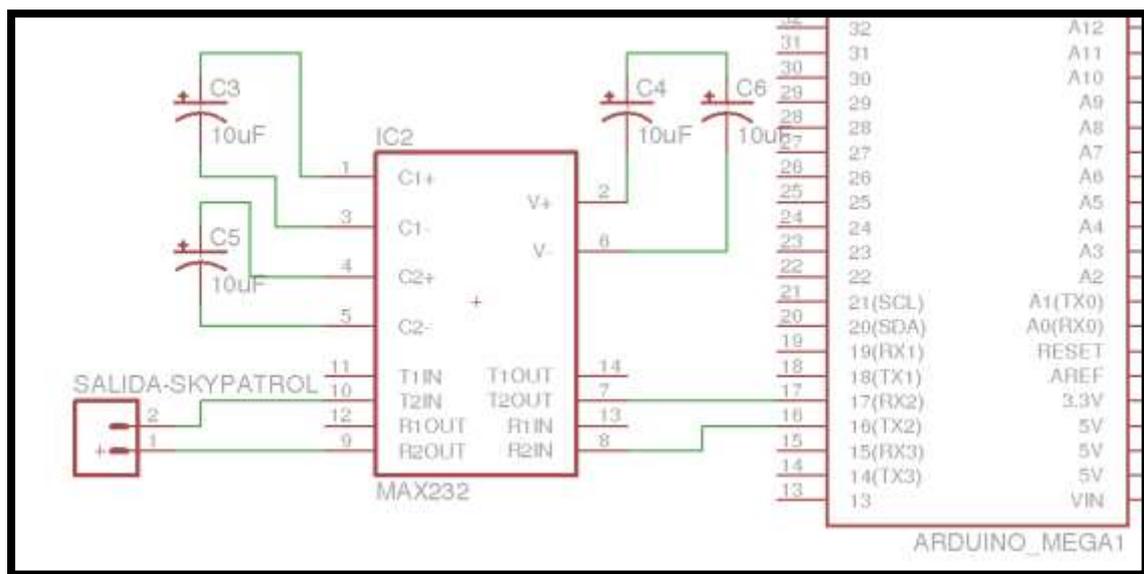


Figura. 28 Conexión MAX232 con Skypatrol
Fuente: Autor

4.3.3.2. Pulso de activación

El módulo de transmisión SKYPATROL para enviar la información realiza eventos en su configuración; para que estos eventos se realicen es necesario que ocurra un evento externo, este evento externo será activado mediante un pulso negativo que lo entregara el módulo ARDUINO mediante un pin digital, para activar este evento es necesario que el pulso sea cero lógico.

Los pines digitales del módulo ARDUINO no disponen de la suficiente corriente para activar la entrada digital del módulo SKYPATROL por lo cual se ha desarrollado un circuito con un relé para que disponga de la suficiente corriente.

A continuación se ilustra el esquema del circuito, figura 29, la configuración del transistor 2N3904 esta como una compuerta NOT, el cual al disponer de un nivel alto en la entrada (Base) del transistor, en el colector se obtendrá un señal de 0V o GND, esta señal activa el relé y hace pasar la señal L(low) hacia el SKYPATROL.

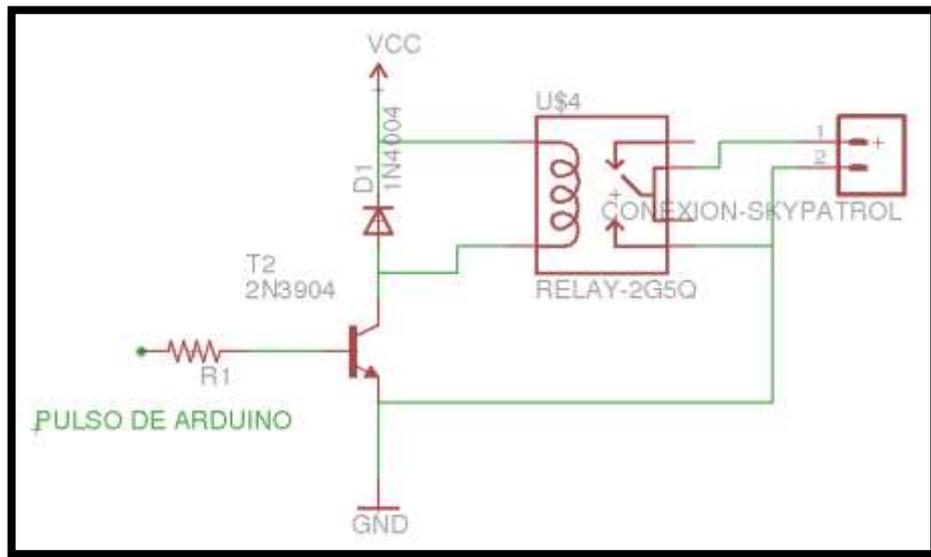


Figura. 29 Circuito de pulso de activación
Fuente: Autor

4.3.4. Tratamiento de la Información

ARDUINO Mega dispone de 4 comunicaciones UART de las cuales utilizaremos 3, la comunicación 1 para recibir información desde el contador de pasajeros, la comunicación 2 para recibir y enviar información desde el equipo SKYPATROL y la comunicación 0 para grabar cambios en el firmware directamente sin desmontar el circuito.

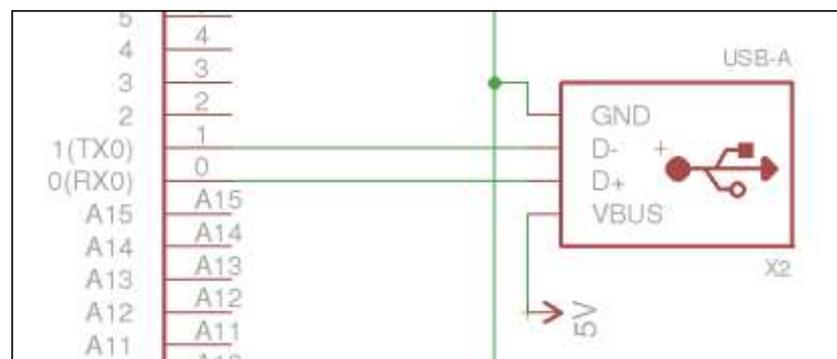


Figura. 30 Conexión serial 0
Fuente: Autor

El contador de pasajeros envía los siguientes valores los cuales serán almacenados en variables para luego ser enviados por el dispositivo SKYPATROL.

“1” Cuenta pasajero ingresa.

“2” Cuenta pasajero sale.

“3” Obstrucción del Sensor.

Cada segundo el controlador solicita la trama GPRMC para recolectar datos los datos necesarios cuando se cumple las condiciones dadas en el controlador construye una trama, la cual es enviada al equipo SKYPATROL y será activada mediante un evento externo para su envío hacia el servidor.

Para la interpretación en el servidor de aplicación la trama estará separada por comas, sus componentes se describen a continuación.

Ej. CONT,712839298772842,123412,2,234,1,190,3751.65,S,14507.36,260414

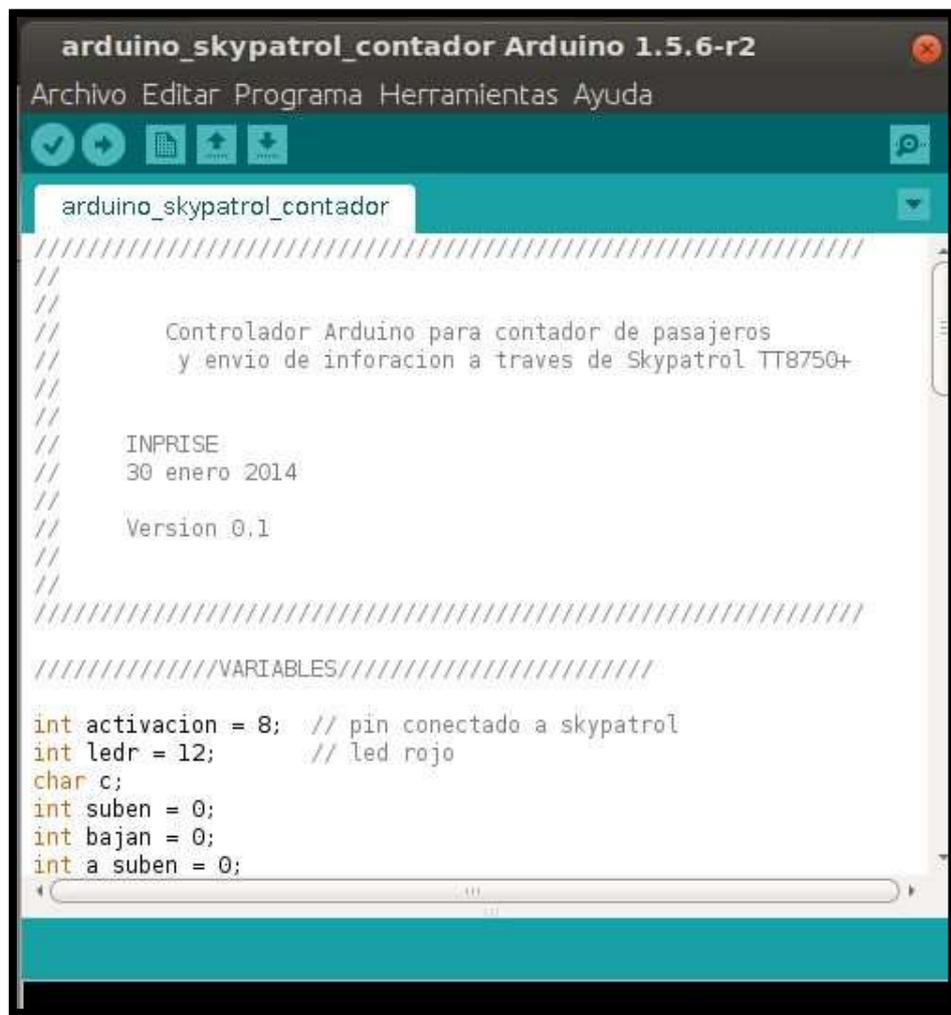
Tabla 19. Campos de la trama desarrollada

CONT	inicio de trama
712839298772842	IMEI del dispositivo
123412	Hora (hora, minutos, segundos)
2	Pasajeros suben
234	Conteo suben general
1	Pasajeros bajan
190	Conteo bajan general
3751.65,S	Latitud
14507.36,E	Longitud
260414	Fecha (día, mes, año)

Fuente: Información recopilada por el autor

4.3.5. Programación de ARDUINO

El proyecto ARDUINO cuenta con un IDE propio para la elaboración, compilación y grabado de sus programas normalmente llamados “sketchs”, la programación está basada en lenguaje C y la grabación hacia la placa se hace directamente a través de un cable USB. Este IDE al igual que todo el proyecto ARDUINO es software libre por lo que su descarga y uso es totalmente libre.



```
arduino_skypatrol_contador Arduino 1.5.6-r2
Archivo Editar Programa Herramientas Ayuda
arduino_skypatrol_contador
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
//      Controlador Arduino para contador de pasajeros
//      y envio de inforacion a traves de Skypatrol TT8750+
//
//
//      INPRISE
//      30 enero 2014
//
//      Version 0.1
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////VARIABLES//////////////////////////////////////////////////////////////////

int activacion = 8; // pin conectado a skypatrol
int ledr = 12;     // led rojo
char c;
int suben = 0;
int bajan = 0;
int a suben = 0;
```

Figura. 33 IDE para programación de sketches
Fuente: Autor

El entorno de programación dispone de varias librerías las cuales pueden ser usadas por otros programas a la vez, además que se puede desarrollar rutinas para acciones repetitivas. Para esta aplicación se desarrolló rutinas para la decodificación de la trama GPRMC, captura de valores del contador, construcción de la trama y envío de la información.

El algoritmo utilizado en el controlador realizara las siguientes actividades:

- Configurar las comunicaciones seriales con el contador y el modulo GPS/GPRS
- Almacenar los datos enviados por el contador (interrupción serial)
- Obtener trama GPRMC del módulo GPS
- Determinar distancia de recorrido o movimiento del vehículo
- Enviar los datos por el módulo GPRS

4.3.5.1. Flujograma

A continuación se esquematiza el Flujograma que sirve para la creación del código del programa.

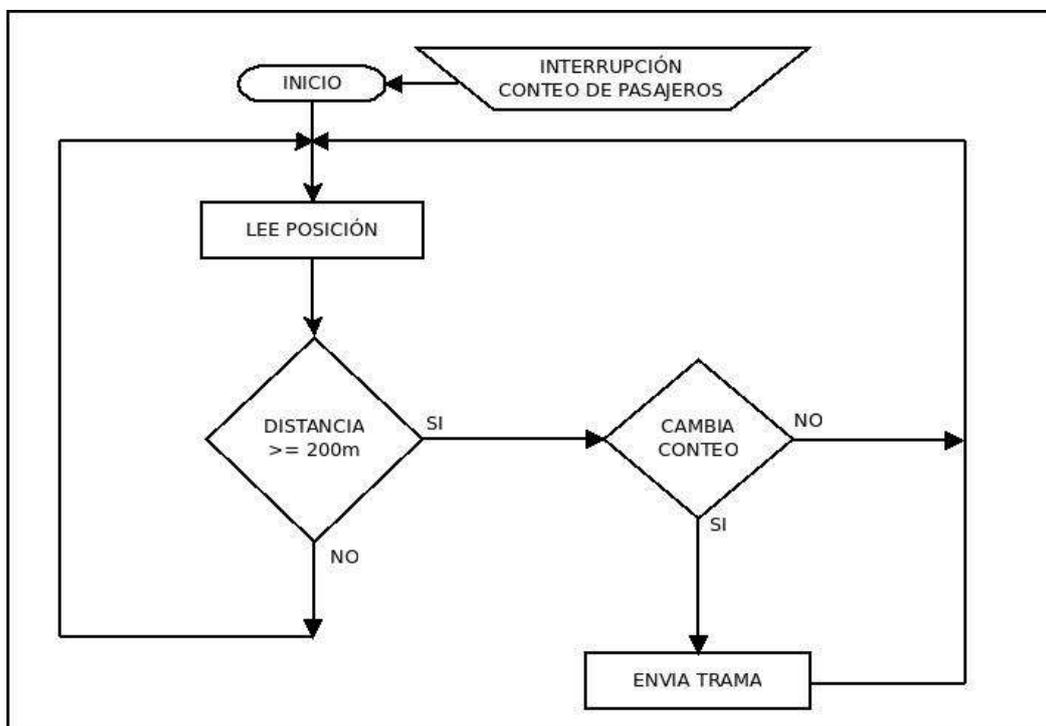


Figura. 34 Flujograma del controlador

Fuente: Autor

4.4. Módulo GPS / GPRMS

Se usa el dispositivo que los vehículos cuentan actualmente, este dispositivo es un equipo SKYPATROL TT8750+ el cual dispone de un módulo GPS y GPRS el mismo que puede ser configurado mediante comandos AT.

Este dispositivo servirá para dotar de información sobre posición y tiempo al controlador ARDUINO y a la vez de interfaz de comunicación con el servidor de aplicaciones.

4.4.1. SKYPATROL TT8750+

Es un sistema electrónico de rastreo que combina la tecnología GPS y GSM/GPRS y las encapsula en un solo dispositivo, el cual determina la posición del mismo. Este dispositivo dispone de entradas y salidas que pueden controlar dispositivos externos al sistema de rastreo.

Características principales del TT8750+:

- Poderoso sistema de GPS
- Modem de cuatribandas
- Sensor de movimiento

- Batería de repuesto
- Estuche compacto y resistente
- Bajo consume de energía
- Compatible con Garmin



Figura. 35 Imagen Skypatrol
Fuente: Autor

- **Características SKYPATROL TT8750+**

Tabla 20. Características SKYPATROL TT8750+

I/O Entrada digital, salidas, entrada analógica	2,3,2
Canales GPS	50
Antena GSM	Interna
Antena GPS	Interna / Externa
Contadores (Timers)	32
Bateria Interna	Si

Fuente: Información recopilada por el autor

Pines de conexión

Tabla 21. Pines de conexión SKYPATROL TT8750+

PIN	NOMBRE	FUNCION
1	MICP	Micrófono 2.2 K
2	AGND	Masa para entrada análoga
3	IGN	Entrada de ignición, activación positiva
4	RXD	Rx RS232
5	TXD	TX RS232
6	GND	Masa digital
7	OUT 3	Salida 3, 150mA máximo
8	OUT 2	Salida 2, 150mA máximo
9	EARNP	Parlante 1/4 W
10	EARN	
11	PWR	Fuente externa de 8 a 32V
12	IN 2	Entrada digital, activación negativa
13	IN 1	Entrada digital, activación negativa
14	OUT 1	Salida 2, 150mA máximo, con circuito latch
15	AD1	Entrada Analógica 0-16.8 V
16	AD2	Entrada Analógica 0-16.8 V

Fuente: Información recopilada por el autor

4.4.1.1. Configuración

Los parámetros de configuración para realizar aplicaciones específicas con este dispositivo se realizan mediante comunicación serial y el uso de comandos AT, el proveedor del dispositivo incluye a sus clientes una guía de todos los comandos y parámetros que pueden ser utilizados.

La comunicación serial por defecto para configurar el dispositivo SKYPATROL mediante comandos AT está asignada a una velocidad de 115200 baudios, 8bits, sin paridad, 1 bit de parada (8N1); cualquier software de comunicación serial puede servir de host para asignar los comandos AT de configuración; para cada comando de configuración

el módulo SKYPATROL enviara OK, ERROR, o parámetros de consulta, dependiendo del comando o la petición.

El dispositivo SKYPATROL por realizar la función de rastreo para el control de tiempos en las unidades de transporte dispone de una configuración preestablecida que realiza acciones para el propósito actual, estos parámetros son:

- Emisión de posición, tiempo y fecha cada 30 segundos
- Alertas de eventos externos: desconexión alimentación externa, desconexión batería interna, botón externo

La configuración que realizaremos al equipo SKYPATROL se complementara al script de configuración inicial y luego para adquirir la información que necesitamos haremos una consulta vía comunicación serial con los comandos AT respectivos.

La configuración extra que se realiza en el equipo SKYPATROL no afecta en ninguna circunstancia el propósito primario del equipo, su funcionamiento es exactamente igual que al inicio.

4.4.2. Conexión

La alimentación del dispositivo SKYPATROL funciona dentro de un rango de 7 a 35 V por lo cual la alimentación con los demás dispositivo sigue siendo común a 12V, hay que tomar en cuenta que para que las comunicaciones entre dispositivos funcionen correctamente las masas (GND) deben ser comunes.

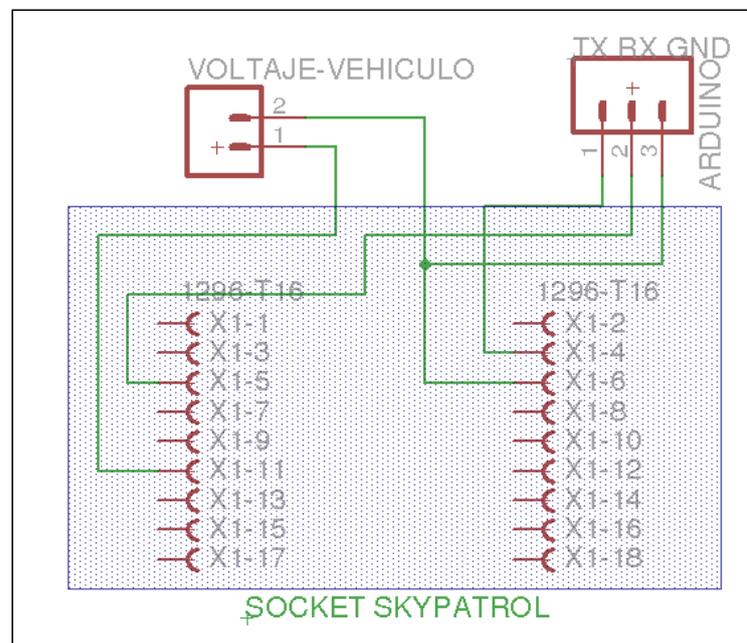


Figura. 36 Conexión del socket SKYPATROL

Fuente: Autor

4.4.3. Configuración

El equipo SKYPATROL actualmente se encuentra configurado para que emita información para el sistema de control y monitoreo cada 30 segundos, la configuración extra que se va a realizar no altera la actual configuración, además que el programa del controlador ARDUINO enviara comandos para pedir información al equipo SKYPATROL.

4.4.3.1. Script de configuración

Se debe configurar hacia donde se enviara la información, el APN de la operadora celular; se describe 2 timer los cuales se activan de pendiendo de la condición del pin de censado del encendido del vehículo.

```
at+cgdcont=1,"ip","internet.claro.com.ec" // APN del operador
AT$TTSRVDST=1,1,"198.50.158.43",22000,1 // ip, puerto y
protocolo
AT$TTCNT=1,2,60 //timer 1 tiempo 60 segundos
AT$TTCNT=2,2,30 //timer 2 tiempo 30 segundos
AT$TTMSGMASK=01,1675462 // formato de trama
GPRMC

AT$TTFNT=11,1,14,2,0 // evento censado de encendido del
vehículo
AT$TTFNT=11,2,9,1,1 // activación timer 1
AT$TTFNT=11,3,26,11,1 // envío de trama con timer 1
AT$TTFNT=12,1,14,1,0 // evento censado del vehículo
apagado
AT$TTFNT=12,2,9,0,0 // activación timer 2
AT$TTFNT=12,3,26,12,1 // envío de trama timer 2
```

Figura. 37 Configuración módulo SKYPATROL

Fuente: Autor

Esta configuración hace que el servidor de aplicaciones dependiendo del timer activado lleguen tramas de información conteniendo la trama GPRMC más datos adicionales de identificación del dispositivo y numero de evento ejecutándose.

El dispositivo para realizar acciones basa su funcionamiento en la ejecución de eventos, él envió de la trama de conteo de pasajeros será un evento no continuo por lo cual no necesitaremos un timer para su envió sino un evento externo el cual active el envió, para lograr nuestro propósito crearemos un evento en el cual se censará el estado de un pin digital del módulo SKYPATROL, este pin al sensar un estado LOW (0V) activa una acción, esta acción será la encargada de enviar la información al servidor de aplicaciones.

```
AT$TTSTOCMD=1,  
AT$TTSNDMG=1,"Trama de información de conteo" // comando  
// almacenamiento  
AT$TTFNT=13,1.0.0.0 // censa estado input 1 a LOW  
AT$TTFNT=13.3,28.1.0 // ejecuta comando almacenado 1  
AT$TTFNT=13,3.26,13,1 //confirmación de estado del input 1
```

Figura. 38 Configuración para PIN de envío

Fuente: Autor

4.4.3.2. Comandos de consulta para pedir información

Para obtener la trama GPRMC del dispositivo GPS debemos enviar desde el controlador el comando "AT\$TTGPSQRY=10" , este comando envía en una sola línea toda la trama GPRMC seguido del comando de confirmación "OK"; en el controlador es necesario construir un método el cual separe todos los valores enviados para su análisis.

4.5. Servidor

Para ofrecer un servicio con un alta disponibilidad localmente de la aplicación; las condiciones ofrecidas por los proveedores de servicio de Internet y suministro eléctrico no es el adecuado o en su defecto es de excesivo costo, se ha optado por contratar un servidor con acceso remoto el cual es 100% administrable desde cualquier parte del mundo y ofrece una disponibilidad del 99.999% ya que se encuentra bajo una infraestructura de datacenter.

El servidor que se encuentra contratado es de gama alta con capacidades de virtualización, aloja varios servidores virtuales que hacen funcionar todos los sistemas e infraestructura de INPRISE. La distribución instalada en el servidor es Proxmox v3.0 64bits, distribución basada en Debian Wheezy con una interfaz web para el manejo de las máquinas virtuales.

Tabla 22. Características servidor de aplicaciones

Ítem	Descripción
Procesador	Intel(R) Core(TM) i3-2130 CPU @ 3.40GHz
RAM	8GB DDR3
Discos Duros	2x 1TB - SATA2
RAID	Sin RAID
NIC	GigaEthernet
ipv4	soporta
ipv6	soporta

Fuente: Información recopilada por el autor

4.6. Recepción de Información en Socket

El script realizado con lenguaje PYTHON será el encargado de mantener activa la conexión TCP en el servidor, el mismo que estará escuchando las 24 horas del día toda la información que envíen los dispositivos asociados al sistema.

```

-----
861074021943181 18|014 TT0750 PLUS 1459 2014-02-14 08:59:30 |0.348585 -78.1234916667| 0km/h 0°
rost 0.93
-----
861074021903351 18|006 TT0750 PLUS 1459 2014-02-14 08:59:35 |0.348976166667 -78.1204773333| 19km/h 10°
rost 0.882
-----
861074021825362 18|016 TT0750 PLUS 1459 2014-02-14 08:59:36 |0.345211666667 -78.1345978333| 0km/h 0°
rost 0.611
-----
861074021967347 18|013 TT0750 PLUS 1459 2014-02-14 08:59:36 |0.346514166667 -78.1212869| 0km/h 0°
rost 0.571
-----
861074021795748 18|001 TT0750 PLUS 1459 2014-02-14 08:59:38 |0.344076333333 -78.1328365| 28km/h 79°
rost 0.667
-----
861074021922193 18|012 TT0750 PLUS 1459 2014-02-14 08:59:38 |0.361485666667 -78.1300948333| 26km/h 123°
rost 0.889
-----
861074021927648 18|008 TT0750 PLUS 1459 2014-02-14 08:59:45 |0.337468833333 -78.1693743333| 0km/h 0°
rost 0.73

```

Figura. 39 Recepción de tramas en el servidor

Fuente: Autor

El script una vez que lee dos datos transmitidos envía la información a la base de datos, para el contador de pasajeros se guardara el ID del dispositivo, la posición y el número de conteos; los demás datos necesarios ya son guardados por el anterior script que maneja los datos de ubicación y posición del vehículo.

Se debe toma en cuenta que en la base de datos se tiene un registro único del IMEI del dispositivo y a este dato se relacionan todos los eventos y valore que llegan, ya que para la posterior aplicación se hará referencia al IMEI del dispositivo y se recuperara lo datos necesarios.

Para lograr todo este propósito se desarrolla el script en el editor Geany en cual dispone de herramientas de edición y ejecución basadas en PYTHON, el script desarrollado realiza el proceso que será detallado a continuación.

- Activa conexión UDP.
- Escucha información por el puerto seleccionado.
- Busca información necesaria, decodifica la trama tomando en cuenta la separación por comas.
- Abre una conexión con la base de datos.
- Guarda los datos en la tabla de nuestra base de datos.
- Cierra la conexión a la base de datos.
- Vuelve a escuchar el socket de comunicación.

The image shows a screenshot of the Geany text editor with a Python script named 'contador.py' open. The script is a simple counter that reads a file line by line, counts the number of lines starting with 'Suben' and 'Bajan', and prints the totals along with the current date and time. The script also prints the coordinates (Lat/Lon) and a TrackID for the first line that starts with 'Dato Encontrado'. The terminal window shows the output of the script, which is executed twice. The first execution shows a total of 4911 'Suben' and 3210 'Bajan' lines, with a date of 2014-05-07 16:19:45 and coordinates 0.347255 -78.1107728333. The second execution shows a total of 4912 'Suben' and 3211 'Bajan' lines, with a date of 2014-05-07 16:20:10 and coordinates 0.348205166667 -78.1101873333. The terminal also shows the program exiting with code 0 and a prompt to press return to continue.

```

contador.py
2  #!/usr/bin/python
3
4  import datetime
5  import psycopg2
6
7
8  dsn = "dbname=dbcont
9  # trama ['CONT','86
10     # 0,
11
12 def leertxt():
13     archi=open('cont
14     linea=archi.read
15     while linea!="
16         if 'CONT' i
17             try:
18
19                 pri
20                 tc =
21                 s =
22                 st =

```

```

Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Suben: 1
Bajan: 2
Suben Total: 4911
Bajan Total: 3210
Date: 2014-05-07 16:19:45
Lat/Lon: 0.347255 -78.1107728333
Dato Encontrado
#####
TrackID: 861074023497681
#####
Suben: 1
Bajan: 1
Suben Total: 4912
Bajan Total: 3211
Date: 2014-05-07 16:20:10
Lat/Lon: 0.348205166667 -78.1101873333
-----
(program exited with code: 0)
Press return to continue

```

Figura. 40 Editor Geany y script en ejecución

Fuente: Autor

4.7. Almacenamiento de Información en la Base de Datos

El script desarrollado en PYTHON permite guardar automáticamente los datos en la base de datos, por lo cual se debe crear una tabla en Postgres que contenga los campos que vamos a obtener de la decodificación de la trama.

La creación de tablas en postgres se puede hacer de dos maneras mediante líneas de comandos utilizando sentencias SQL y de forma gráfica con el asistente PgAdmin; para la facilidad de uso y comprensión se ha usado el entorno grafico PgAdmin en el que se ha

creado una nueva tabla con los campos a llenarse.

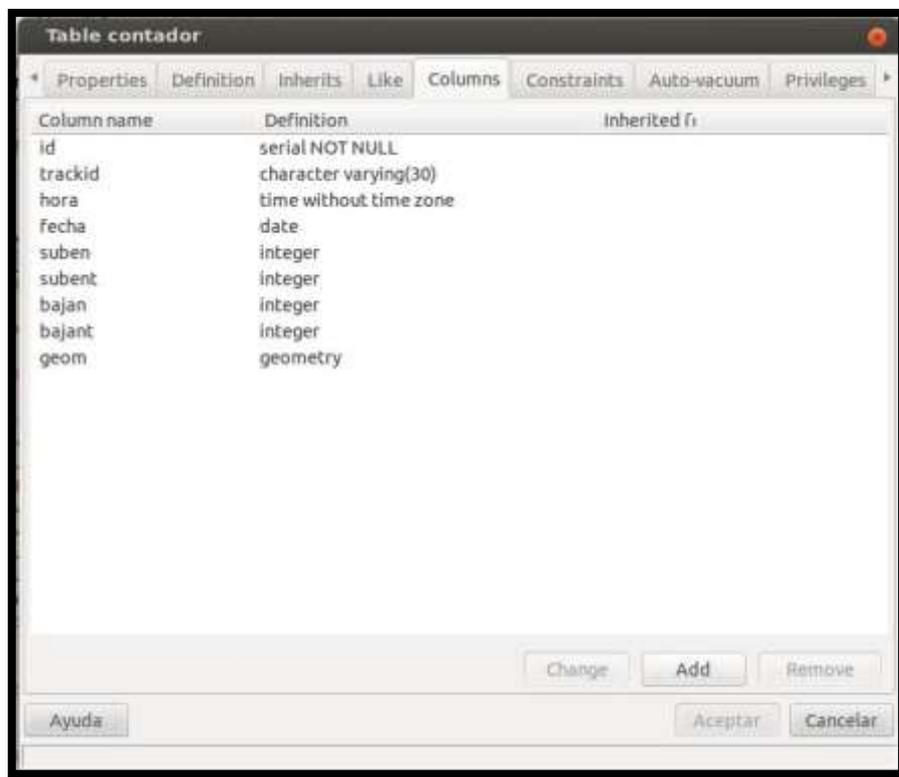


Figura. 41 Creación de campos en la tabla contador

Fuente : Autor

Una vez creada la tabla y ejecutado el script los campos en la tabla comenzaran a llenarse automáticamente, para visualizar estos datos los podemos hacer desde el panel de navegación de las tablas de PgAdmin.

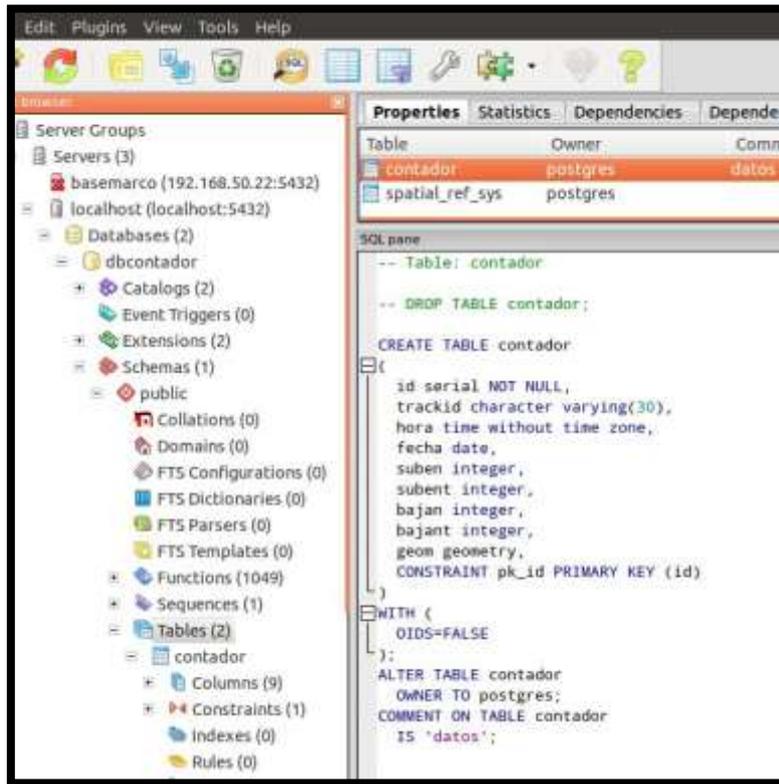


Figura. 42 Panel de navegación de PgAdmin

Fuente: Autor

id [PK] serial	trackid character varying(30)	hora time without time zone	fecha date	suben integer	subest integer	bajan integer	bajant integer	geom geometry
1	861074023497681	20:33:47	2014-04-25	2	2	5	9	D10100020651000006000380CC84753C0444015E16200053F
2	861074023497681	20:36:36	2014-04-25	9	11	4	9	D1010002064100000471A6EEDC30753C088C00229A280052F
3	861074023497681	21:04:32	2014-04-25	18	29	10	18	D101000206810000049952870CBE753C027CF0ADC6700033F
4	861074023497681	21:07:37	2014-04-25	3	31	2	21	D10100020641000000558ACF1CB8753C0048FC73C0203053F
5	861074023497681	21:45:25	2014-04-25	5	36	3	24	D1010002065100000064A0F39C853CF450AD53D230063F
6	861074023497681	08:00:22	2014-04-26	10	48	10	34	D1010002060100000272F32018F853CDAF706F4951043F
7	861074023497681	08:03:02	2014-04-26	1	47	0	34	D101000206010000005568188124753C0800080F044D43F
8	861074023497681	08:14:00	2014-04-26	4	51	1	35	D10100020641000005C568849738753C0C938F0C6754043F
9	861074023497681	08:18:34	2014-04-26	1	52	0	35	D10100020601000000241640C908753C0C38836E06A043F
10	861074023497681	08:23:04	2014-04-26	7	59	6	41	D1010002060100000A18504C8E8753C0A0C04B07854F043F
11	861074023497681	08:38:22	2014-04-26	5	64	1	42	D1010002064100000CF52C371778753C00F6C46E2880063F
12	861074023497681	08:39:46	2014-04-26	1	65	0	42	D101000206010000000003F96C8753C0D0906E1CA72F1043F
13	861074023497681	08:40:57	2014-04-26	1	68	0	42	D10100020601000007043A42B888753C050F3CE38FA03073F
14	861074023497681	08:43:03	2014-04-26	8	68	2	44	D10100020641000001934766A238753C02F294CF05230373F
15	861074023497681	09:48:03	2014-04-26	1	70	2	46	D1010002064100000B6AF67080F6753C0558998E809E4D43F
16	861074023497681	09:48:32	2014-04-26	0	70	3	49	D101000206010000008B71D0118753C0330510001806043F
17	861074023497681	08:52:14	2014-04-26	0	70	2	51	D1010002060100000CE78838048753C05073F2212F0053F
18	861074023497681	08:54:18	2014-04-26	1	71	1	52	D10100020641000008670928F8753C0C06ACFFC57C3053F
19	861074023497681	08:55:00	2014-04-26	3	74	2	54	D10100020641000006628D867158753C0C4E28D348485033F
20	861074023497681	08:55:56	2014-04-26	1	75	0	54	D101000206010000005F0A1141A8753C048C08045F8C9053F
21	861074023497681	08:56:17	2014-04-26	1	76	0	54	D10100020641000005F476856108753C0002571544C3053F

Figura. 43 Datos almacenados en Postgres

Fuente: Autor

PRUEBAS Y RESULTADOS

5.1. Pruebas

Las pruebas realizadas fueron en base a condiciones de programación y visualización en terminales seriales de depuración que determinaron que todos los pasos y procesos sean ejecutados

5.1.1. Contador

Las pruebas realizadas en el sensor se realizaron observando el envío de información de conteo en un terminal serial para determinar fallas y corregir errores. Además de comparar el conteo manual con los resultados.



Figura. 44 Pruebas de funcionamiento del contador

Fuente: Autor

5.1.2. Controlador

Se verifica el correcto funcionamiento verificando los datos que este envía cuando se cumplen las condiciones de posición más la información que llega al servidor.

The image shows a screenshot of the Arduino IDE interface. The title bar at the top reads 'Archivo Editar Sketch Herramientas Ayuda'. Below the title bar is a toolbar with icons for checkmark, refresh, file, upload, and download. The main editor area contains the following code:

```
arduino_skypatrol_contador
int ledr = 12;      // led rojo
char c;
int suben = 0;
int bajan = 0;
int a_suben = 0;
int a_bajan = 0;

//////////////////CONFIGURACION INICIAL//////////////////

void setup(){

  Serial.begin(9600);    // conexion PC
  Serial1.begin(115200); // conexion skypatrol
  Serial2.begin(4800);   // conexion contador
  pinMode(activacion,OUTPUT); //pin de activacion de envio
  pinMode(ledr,OUTPUT);  //pin de activacion de envio

  digitalWrite(activacion,LOW); //pin inicializa en alto
  digitalWrite(ledr,HIGH); //pin inicializa en alto
}

//////////////////PROGRAMA PRINCIPAL//////////////////

void loop(){
```

Figura. 45 Pruebas del controlador

Fuente: Autor

5.1.3. Módulo SKYPATROL

El módulo skyptrol tiene que seguir enviando información de manera continua sobre posición y tiempo como lo hacía anteriormente y enviara la nueva trama del evento de conteo.



Figura. 46 Funcionamiento del SKYPATROL
Fuente: Autor

5.1.4. Representación de la información

La información almacenada en la base de datos se la representara para propósitos de presentación local en el software QGIS el cual realiza una conexión directa con la base de datos y a la vez puede filtrar contenidos para su análisis. El software QGIS permite descargar capas de mapas para la visualización de los datos con imágenes reales del entorno.

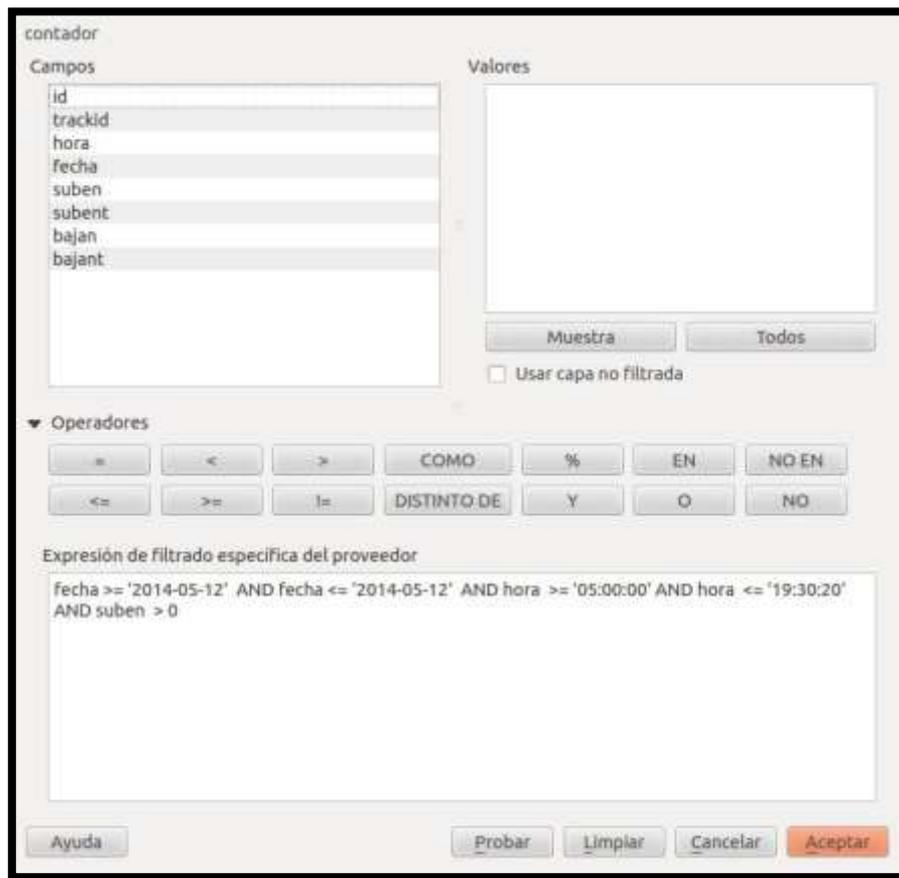


Figura. 47 Filtrado de datos en QGIS

Fuente: Autor

5.2. RESULTADOS

5.2.1. ENSAMBLAJE

Todos los componentes extras que se necesitaron para construir el dispositivo fueron contruidos sobre placas de circuito impreso las cuales garantiza fiabilidad y duración. El dispositivo esta ensamblado en una caja compacta de metal la cual está adherida a soportes para su instalación en el vehículo.



Figura. 48 Ensamblaje Final modelo 1
Fuente: Autor

Los soportes o tubos son de igual manera de metal para asegurar el dispositivo a los soportes del vehículo, estos soporte pueden ir remachados o ajustados con tuercas dependiendo del vehículo.



Figura. 49 Ensamblaje final modelo 2
Fuente: Autor

5.2.2. Comparación de mediciones

A continuación se presenta una tabla comparativa en la que se muestra el error porcentual de las mediciones manuales con respecto al dispositivo electrónico.

Tabla 23. Comparación Conteo Manual y Contador Electrónico

COMPARACIÓN ENTRE CONTEO MANUAL Y UTILIZANDO CONTADOR ELECTRÓNICO.

	Conteo manual	Conteo electrónico	% error
Muestra 1	122	113	7.4
Muestra 2	98	90	8.2
Muestra 3	129	120	7

Fuente: Información recopilada por el autor

5.2.3. Representación en el Software QGIS

La representación final de los datos de conteo puede ser vistas en el software QGIS los mismos que representan las ubicaciones el número de conteos que fueron almacenados en la base de datos.

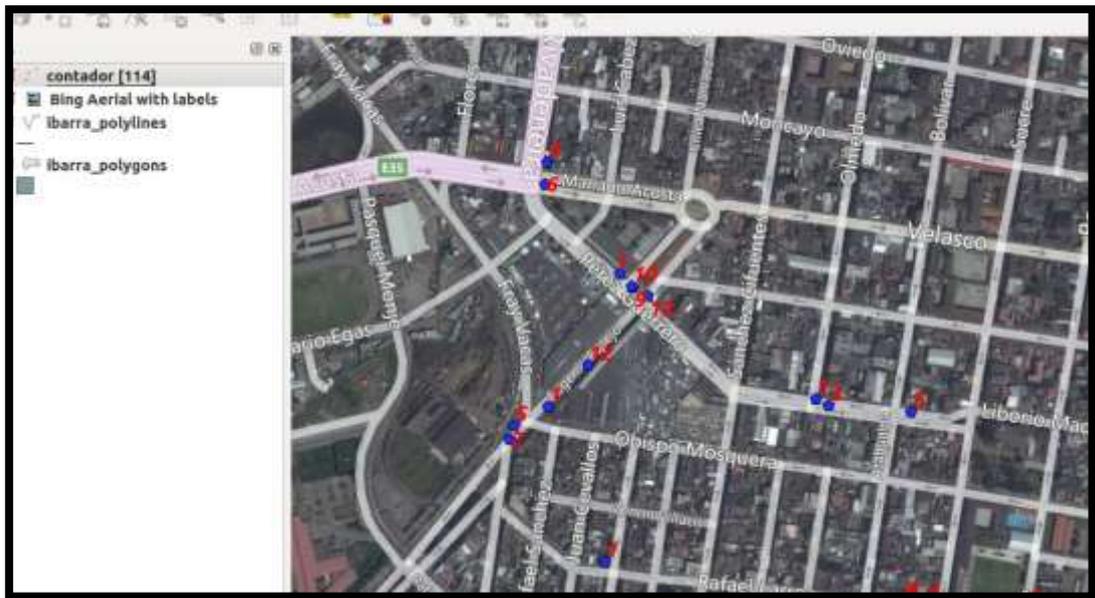


Figura. 50 Representación de datos en QGIS
Fuente: Autor

CONCLUSIONES

- La redacción de un plan de negocios antes de desarrollar un producto que se desea sacar al mercado marca una perspectiva del entorno y la viabilidad del producto, además que garantiza operaciones o acciones que se dejan por alto y luego resultan fundamentales en el proceso.
- La inversión en un producto es fundamental en una empresa de esto depende su presencia en el mercado, pero siempre tomando en cuenta que el nuevo producto sea innovador y resuelva una necesidad del mercado.
- El conteo de pasajero se puede realizar de varias maneras, depende de los factores en los que se encuentra el dispositivo para seleccionar el tipo de sensor a utilizar.
- Los sistemas microcontrolados permiten la creación sistemas embebidos versátiles y de bajo costo, el análisis de factores externos ayuda a diseñar de manera adecuada las interfaces necesarias para la conexión eléctrica.
- El desarrollo de proyectos como ARDUINO ayuda a reducir tiempos en el diseño de dispositivos ya que la circuitería se encuentra probada y solo se adhiere componentes; el mayor porcentaje de tiempo está destinado al desarrollo de software el cual es el motor de los procesos.

- Lo elementos electrónicos básicos ayudan a resolver problemas que aparentemente necesitan mayores componentes, es siempre necesario recordar conceptos básicos teóricos que otorgan soluciones rápidas y de bajo costo.
- Para transferencias pequeñas de datos en entornos móviles, la transmisión GPRS es una solución muy factible y de bajo costo, actualmente en el país la cobertura de GPRS es aceptable y de fácil acceso.
- La geolocalización en la actualidad se ha vuelto una herramienta indispensable en el campo de la transportación, la utilización de sistemas de posicionamiento hace que el servicio sea más ágil y seguro.
- Para la adquisición de datos es necesario disponer de un servidor el cual mediante una conexión a la red almacene los datos enviados por el dispositivo GPRS.
- La manera más confiable y fiable de almacenar los datos obtenidos es mediante el uso de una base de datos estructural, para cualquier aplicación posterior la extracción de datos se la realizara de forma sencilla y rápida.
- Un producto debe ser analizado en su entorno de funcionamiento y los factores en el cual va a operar, de ello depende el diseño para el ensamblaje y su posterior instalación.
- Se utilizó dos sistemas microcontrolados distintos, PIC y Arduino los cuales cumplen sus propósitos y fueron usados por conocimiento adquirido y facilidad de uso, para futuras versiones la placa Arduino puede y soporta la implementación completa del sistema.

RECOMENDACIONES

- Se debe realizar las pruebas de conteo en el ambiente de luminosidad que el vehículo va a trabajar ya que las mediciones pueden variar por los sensores infrarrojos.
- Por la gran gama de microcontroladores se debe analizar cuál es el más factible para el diseño de nuestro dispositivo tomando en cuenta funciones, rendimiento y costo.
- Para la construcción en serie de las placas de circuito impreso se deberá encargar la fabricación a máquinas de ese propósito para reducir errores, costos y optimización del tiempo.
- Actualmente en el país existen 3 operadoras celulares que ofrecen servicio GPRS/GSM es necesario analizar los costos de transferencia para seleccionar la operadora que sirva de transporte de datos.
- Los soportes para la instalación deben ser de un material resistente y a la vez la instalación debe estar bien fijada ya que los vehículos ofrecen mucha vibración.

REFERENCIAS BIBLIOGRAFICAS

LIBROS

España, M. C. (2003). Acceso a internet a través de GPRS/UMTS. En M. C. Boquera, *Sistemas Avanzados de Telecomunicaciones*. Madrid: Diaz de Santos.

Grupo de Trabajo de Ingeniería de Internet. (s.f.). *RFC 793*.

Huerta, E., Mangiaterra, A., & Noguera, G. (2005). GPS Posicionamiento Satelital. En E. Huerta, A. Mangiaterra, & G. Noguera, *GPS Posicionamiento Satelital*. Argentina: UNR.

Gonzales, R. (s.f.). *Python para todos*. España.

Silberschatz, A., Korth, H., & Sudarshan, S. (2006). *Fundamentos de bases de datos* (Quinta ed.). (McGraw-Hill, Ed.) España: Interamericana de España S.A.

DOCUMENTOS ELECTRONICOS

- ARDUINO - Traducción. (s.f.). *ARDUINO*. Obtenido de <http://ARDUINO.cc/en/Main/ARDUINOBoardMega2560>
- ARDUINO. (s.f.). *Grupo de desarrollo Arduiino*. Obtenido de <https://code.google.com/p/ARDUINO/>
- ARDUINO. (s.f.). *Sitio web de ARDUINO*. Obtenido de <http://www.ARDUINO.cc/es/>
- CursoMicros. (s.f.). *Curso de microcontroladores*. Obtenido de Formato transferencia rs232: <http://www.cursomicros.com/avr/usart/formato-transferencia-rs232.html>
- <http://www.chuidiang.com>. (s.f.). <http://www.chuidiang.com>. Obtenido de Programación de sockets en c de Unix/Linux: <http://www.chuidiang.com>
- Martinez, R. (s.f.). *PostgreSQL- es*. Obtenido de http://www.postgresql.org/es/sobre_postgresql
- MikroElektronika. (s.f.). *mikroC*. Obtenido de Programación de microcontroladores: <http://www.mikroe.com/chapters/view/80/>
- NEOTEO. (s.f.). *Neoteo*. Obtenido de Sitio web de Neoteo: <http://www.neoteo.com/historia-del-gps-como-el-mundo-dejo-de-perderse/>
- NMEA. (s.f.). *NMEA*. Obtenido de Estandar NMEA 0183: <http://www.nmea.org>
- Pedro Gutovnik. (s.f.). <http://gutovnik.com/>. Obtenido de <http://gutovnik.com/>: http://gutovnik.com/como_func_sist_gps.htm
- PostgreSQL. (s.f.). *PostgreSQL*. Obtenido de PostgreSQL: <http://www.postgresql.org/es/>
- Sparkfun. (s.f.). *NMEA Manual de referencia Rev 2.1*. Obtenido de Sitio web de Sparkfun: <https://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual1.pdf>
- Universidad de Buenos Aires. (s.f.). *Facultad de Ingenieria*. Obtenido de https://docs.google.com/viewer?url=http%3A%2F%2Fwww.fi.uba.ar%2Fmaterias%2F6637%2Ftps%2Fpractica_hayes.doc

ANEXOS

RESUMEN EJECUTIVO

La empresa INPRISE tomando en cuenta la necesidad de desarrollar productos innovadores que cubran los requerimientos de sus actuales clientes, ve como necesario el desarrollo de un sistema de conteo de pasajeros y transmisión de la información mediante tecnología GPRS disponible actualmente en los vehículos de sus clientes.

El mercado objetivo para este producto esta específicamente enfocado al transporte urbano de pasajeros los mismos que disponen actualmente de un sistema de localización y control con tecnología GPS/GSM, este será el mercado con el cual se inicie el desarrollo del proyecto y que puede ser replicado en cualquier ciudad donde se disponga de los mismos factores.

A continuación se ponen a consideración todos los aspectos económicos que intervienen en la ejecución del proyecto.

Costo diseño y software del dispositivo a la empresa	\$500
Costo hardware del dispositivo a la empresa	\$360
Costo final del dispositivo	\$860
Costo mensual de transmisión de datos, representación y mantenimiento	\$47

El valor de 860 dólares es el valor neto del producto que es desarrollado en la empresa, el modelo de negocios de la empresa es la rentabilidad por servicios, mas no las

utilidades por la venta de un equipo, de ahí se deben tomar en cuenta 2 aspectos fundamentales.

La venta del dispositivo no debe generar un costo elevado hacia el cliente, ya que este valor debe ser cancelado en su totalidad al momento de la instalación.

La rentabilidad del negocio está determinada en el cobro de las mensualidades por transmisión de información, representación, reportes y soporte técnico.

Tomando en cuenta estos detalles, a continuación se pone a consideración el valor mínimo por el cual se debe ofertar el producto, los valores que se detallan a continuación están basados en la aceptación de una de las 2 Cooperativas como empresa inicial de venta.

Universo inicial de clientes	127
Precio venta mínimo del dispositivo	\$1000
Ganancia de 16,27% del valor a invertir	

El valor arriba descrito (47 dólares) para el cobro por transferencia de información, reportes y soporte técnico se realizó para un solo vehículo, tomando en cuenta la escalabilidad del servidor y los servicios web en conjunto, se puede determinar que el valor mínimo por cobro de servicios para el universo inicial de clientes que se estima es de 10 dólares.

Este valor representa el pago mensual por unidad de transporte, siendo este rubro el que determina la rentabilidad del proyecto ya que se deberá negociar el contrato mínimo por un año de servicio, bajo estos términos se puede disponer de una ganancia del 66.6% del valor recauda al cabo de un año.

Inversión Inicial

Tomando en cuenta las anteriores negociaciones de venta de soluciones con las empresas, el contrato consistirá en recibir un anticipo del 60% para iniciar el proyecto, con estos antecedentes la empresa deberá invertir el 40% restante que consiste en un valor de 43686 dólares.

Esta inversión será recuperada al finalizar la fase de instalación del sistema en todas las unidades en un estimado de 25 días laborables.

Ingresos mensuales por servicio

Una vez que el proyecto comience su funcionamiento, el valor que debe cancelar la empresa será de 1127 dólares mensuales, de los cuales se debe debitar el pago por el consumo de datos de transferencia (megas de navegación) por cada vehículo, el pago de servidores y personal de soporte técnico; el restante será el margen de ganancia con un porcentaje del 66.6%.

Modelo de Negocios CANVAS



Sensor Infrarrojo Sharp

GP2Y0A21YK0F

Distance Measuring Sensor Unit
Measuring distance: 10 to 80 cm
Analog output type



■ Description

GP2Y0A21YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IRED (infrared emitting diode) and signal processing circuit.

The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.

This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■ Features

1. Distance measuring range : 10 to 80 cm
2. Analog output type
3. Package size : 29.5×13×13.5 mm
4. Consumption current : Typ. 30 mA
5. Supply voltage : 4.5 to 5.5 V

■ Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

■ Applications

1. Touch-less switch
(Sanitary equipment, Control of illumination, etc.)
2. Robot cleaner
3. Sensor for energy saving
(ATM, Copier, Vending machine)
4. Amusement equipment
(Robot, Arcade game machine)

Microcontrolador PIC 16F88



PIC16F87/88

18/20/28-Pin Enhanced Flash MCUs with nanoWatt Technology

Low-Power Features:

- Power-Managed modes:
 - Primary Run: RC oscillator, 76 μ A, 1 MHz, 2V
 - RC_RUN: 7 μ A, 31.25 kHz, 2V
 - SEC_RUN: 9 μ A, 32 kHz, 2V
 - Sleep: 0.1 μ A, 2V
- Timer1 Oscillator: 1.8 μ A, 32 kHz, 2V
- Watchdog Timer: 2.2 μ A, 2V
- Two-Speed Oscillator Start-up

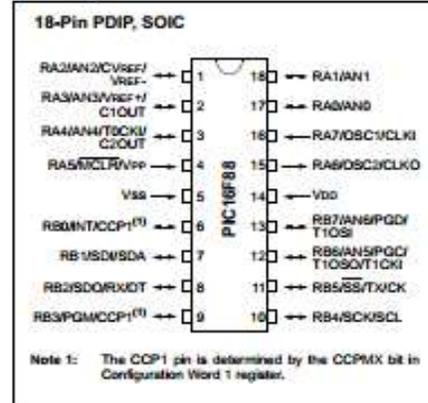
Oscillators:

- Three Crystal modes:
 - LP, XT, HS: up to 20 MHz
- Two External RC modes
- One External Clock mode:
 - ECIO: up to 20 MHz
- Internal oscillator block:
 - 8 user selectable frequencies: 31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz

Peripheral Features:

- Capture, Compare, PWM (CCP) module:
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit, 7-channel Analog-to-Digital Converter
- Synchronous Serial Port (SSP) with SPI (Master/Slave) and I²C™ (Slave)
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART/SCI) with 9-bit address detection:
 - RS-232 operation using internal oscillator (no external crystal required)
- Dual Analog Comparator module:
 - Programmable on-chip voltage reference
 - Programmable input multiplexing from device inputs and internal voltage reference
 - Comparator outputs are externally accessible

Pin Diagram



Special Microcontroller Features:

- 100,000 erase/write cycles Enhanced Flash program memory typical
- 1,000,000 typical erase/write cycles EEPROM data memory typical
- EEPROM Data Retention: > 40 years
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Processor read/write access to program memory
- Low-Voltage Programming
- In-Circuit Debugging via two pins
- Extended Watchdog Timer (WDT):
 - Programmable period from 1 ms to 268s
- Wide operating voltage range: 2.0V to 5.5V

Device	Program Memory		Data Memory		IO Pins	10-bit A/D (ch)	CCP (PWM)	AUSART	Comparators	SSP	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)							
PIC16F87	7168	4096	368	256	16	N/A	1	Y	2	Y	2/1
PIC16F88	7168	4096	368	256	16	1	1	Y	2	Y	2/1

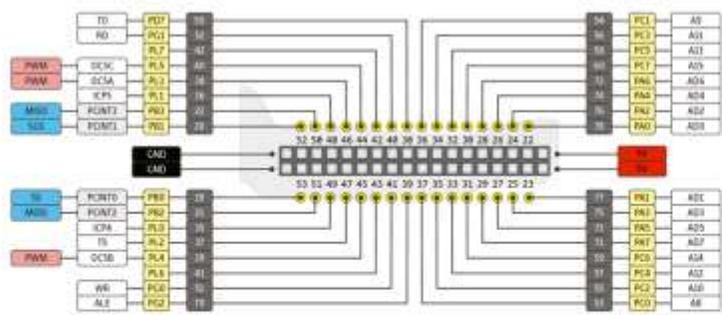
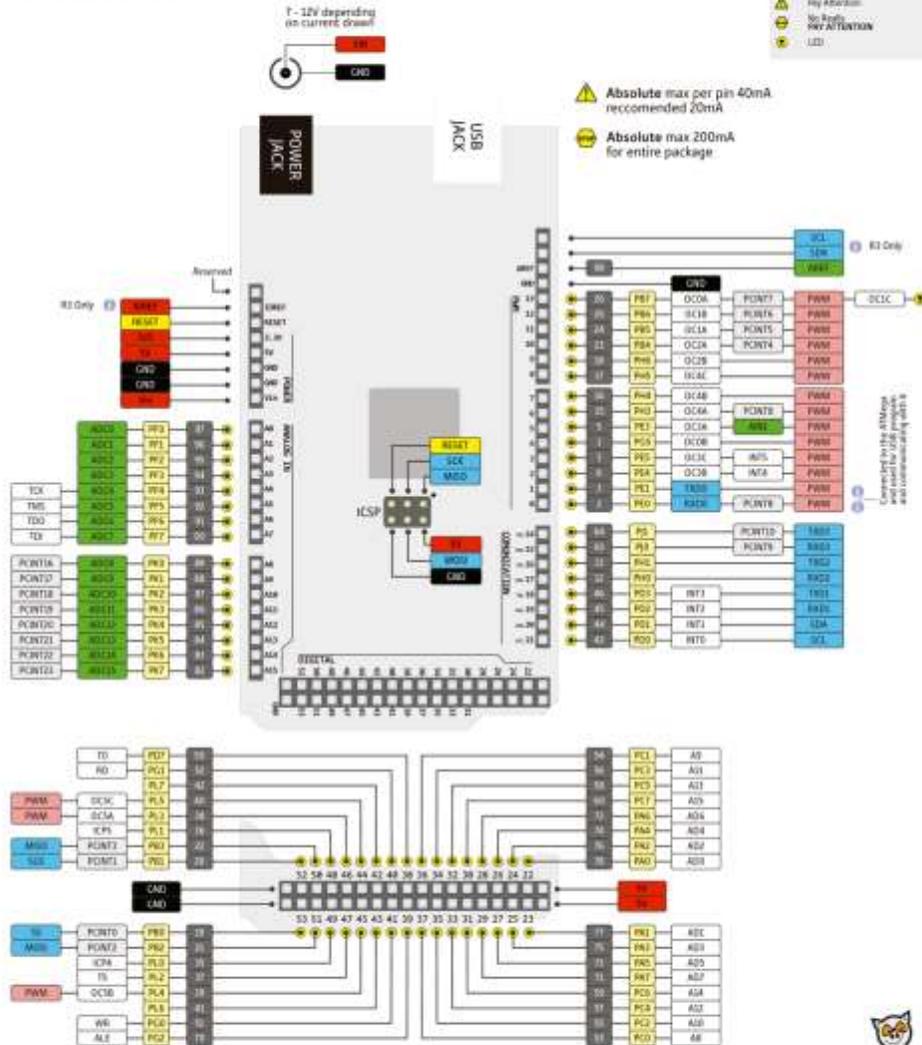
ARDUINO mega 2560

THE UNOFFICIAL
ARDUINO MEGA
PINOUT DIAGRAM

LEGEND

Black	GROUND
Red	POWER
Yellow	CONTROL
Light Blue	PHYSICAL PIN
Light Green	PORT PIN
Light Purple	ATMEGA PIN FUNC
Light Orange	DIGITAL PIN
Light Pink	ANALOG-RELATED PIN
Light Blue	PWM PIN
Light Green	SERIAL PIN

General information
 Pay attention
 No Reply
 PAY ATTENTION
 LED



Código del programa para el shield de Arduino Mega

```
////////////////////////////////////  
// $GPRMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E*68  
//  
// 0 $GPRMC  
// 1 225446 Time of fix 22:54:46 UTC  
// 2 A Navigation receiver warning A = OK, V = warning  
// 3 4916.45,  
// 4 N Latitude 49 deg. 16.45 min North  
// 5 12311.12,  
// 6 W Longitude 123 deg. 11.12 min West  
// 7 000.5 Speed over ground, Knots  
// 8 054.7 Course Made Good, True  
// 9 191194 Date of fix 19 November 1994  
// 10 020.3,E Magnetic variation 20.3 deg East  
// *68 mandatory checksum  
//  
//  
// 5 de junio 2014  
//  
////////////////////////////////////
```

```
#define TEM 500
```

```
int activacion = 8; // pin conectado a skypatrol  
int ledr = 12; // led rojo  
int ledv = 11; // led verde  
int leda = 10; // led amarillo  
  
int buffSize = 0; // lectura del buffer en cero  
int c; //variable auxiliar  
char buffer[100]; // tamaño maximo del buffer de lectura  
int i = 0; //  
long tiempo = 0; // variable tiempo  
char separador[] = ","; // variable separador asignada el carácter coma [,]  
  
int velocidad; // variable para la velocidad en nudos  
int velocidadkm; // variable para la velocidad en kilometros  
long fecha; // variable para fecha
```

```

int distancia;           // variable distancia (km por segundo)
int distanciam;         // variable distancia en metros
long distancia_total;   // variable distancia

char *resultado = NULL; // variable para determinar caracteres vacios
int index = 0;          // índice que recorre el buffer
char campos[20][15];   // matriz de almacenamiento
long hora;              // variable hora

unsigned long millisantes = millis();           // rutina para esperar un determinado
unsigned long millisdespues = millisantes;     // tiempo la respuesta del modulo
unsigned long time;                             // skypatrol

int suben = 0;           // inicializacion de variables el cero
int bajan = 0;
long subeng = 0;
long bajang = 0;

void setup(){           // configuraciones iniciales

  Serial.begin(9600);    // conexión PC
  Serial1.begin(115200); // conexión skypatrol
  Serial2.begin(4800);   // conexión contador
  pinMode(activacion,OUTPUT); //pin de activación de envío
  pinMode(ledr,OUTPUT);  //pines para leds en modo salida
  pinMode(ledv,OUTPUT);
  pinMode(leda,OUTPUT);

  time = millis();

  Serial.println("INICIO"); // depuración comunicación serial
}

void loop(){

  if (tiempo == 65000){ // espera tiempo para pedir la trama
    digitalWrite(ledv,HIGH);
    Serial1.println("AT$TTGPSQRY=10"); // petición de trama GPRMC al modulo Skypatrol
    //Serial.println("AT$TTGPSQRY=10");
    buffSize = 0;
  }
}

```

```

millisantes = millis();
millisdespues = millisantes;
while((millisdespues - millisantes) < TEM){ // espera para respuesta
  while (Serial1.available()>0){
    c= Serial1.read(); // lectura de respuesta del módulo skypatrol
    buffer[buffSize] = c;
    buffSize++; // llenado del vector buffer con la información obtenida
  }
  millisdespues = millis();
}

for(i=0;i<buffSize+1;i++){ // depuración serial impresión de trama
  Serial.print(buffer[i]);
}
decodificar(); // lleva al vector para ser decodificado
delay(50);
tiempo = 0;
}

if(Serial2.available()){ // lectura del contador
  c = Serial2.read();
  if(velocidadkm < 16){
    if( c == 49){ // si llega "1" aumenta variable suben
      suben++;
      subeng++;
    }else if( c == 50){ // si llega "2" aumenta variable bajan
      bajan++;
      bajang++;
    }
    Serial.println(suben,DEC); // depuración impresión de conteos
    Serial.println(bajan,DEC);
    delay(100);
  }
}
digitalWrite(ledv,LOW);
tiempo++;

if ((velocidadkm >= 16) && (suben > 0 || bajan > 0)){ // condición para el envío al modulo
  enviar(); // rutina para el envío
  suben = 0;
  bajan = 0;
}
}

```

```

void decodificar(){ // rutina para separar los campos del buffer tomando en cuenta la coma
resultado = strtok(buffer,separador);
index = 0;
while(resultado != NULL){
//Serial.println(resultado);
strcpy(campos[index], resultado);
index++;
resultado = strtok(NULL, separador);
}

```

```

char *prueba = campos[7];
velocidad = atoi(campos[7]);
velocidadkm = velocidad * 1.852; // transformar velocidad de nudos a km
fecha = atol(campos[9]);

```

```

Serial.println("/////");
Serial.println(prueba);
Serial.println(velocidad,DEC);
Serial.println(velocidadkm,DEC);

```

```

}

```

```

void enviar(){ // rutina para envio de la trama al modulo skypatrol

```

```

Serial.print(" ENVIO ");
Serial1.print("AT$TSTOCMD=1,AT$TTSNDMG=2,");
Serial1.write((byte)0x22);
Serial1.print("CONT,861074023497681,");
Serial1.print(campos[1]); //hora
Serial1.print(",");
Serial1.print(suben);
Serial1.print(",");
Serial1.print(subeng);
Serial1.print(",");
Serial1.print(bajan);
Serial1.print(",");
Serial1.print(bajang);
Serial1.print(",");
Serial1.print(campos[3]); //latitud
Serial1.print(",");
Serial1.print(campos[4]); //

```

```

Serial1.print(",");
Serial1.print(campos[5]); //longitud
Serial1.print(",");
Serial1.print(campos[6]); //
Serial1.print(",");
Serial1.print(campos[9]); //fecha
Serial1.print(",");
Serial1.write((byte)0x22);
Serial1.println();
delay(500);
Serial1.println("AT&W");
delay(1000);

digitalWrite(activacion,HIGH);          // activación del relé

delay(2500);
digitalWrite(activacion,LOW);           // desactivación del relé
digitalWrite(ledr,LOW);

}

```

Código de PYTHON para decodificación del archivo obtenido del socket

```

# -*- coding: utf-8 -*-
#!/usr/bin/python

import datetime
import psycopg2

dsn = "dbname=dbcontador host=localhost port=5432 user=postgres password=postgres"
#
['CONT','861074023497681',013547.00',2',2,5,5,0020.45132,N,07807.29647,W,'260414','\n'] trama
# 0, 1, 2, 3,4,5,6, 7 ,8,9 ,10,11

def leertxt():
    archi=open('contador.log','r') // abrir archivo plano de datos almacenados
    linea=archi.readline().strip()
    while linea!="":
        if 'CONT' in linea: // busca cabecera CONT
            try:

```

```

print 'Dato Encontrado'
tc = linea.split(',') // Separar resultados tomando en cuenta la coma
s = tc[3]
st = tc[4]
b = tc[5]
bt = tc[6]
trackid = tc[1]
print '#####'
print 'TrackID: ' + trackid
print '#####'
print 'Suben: ',s
print 'Bajan: ',b
print 'Suben Total: ',st
print 'Bajan Total: ',bt
h = tc[2] #horas
horas = int(h[0:2])
mins = int(h[2:4])
segs = int(h[4:6])

fecha = tc[11] # fecha
dia = int(fecha[:2])
mes = int(fecha[3:4]) #+ 10

anio = int(fecha[4:])
anio = anio + 2000
fecha_hora = datetime.datetime(anio,mes,dia,horas,mins,segs)
fecha_hora = fecha_hora - datetime.timedelta(hours = 5) // UTC - 5
fecha = str(fecha_hora.date())
horas = str(fecha_hora.time())
date = fecha
time = horas
print 'Date: ' + date + ' ' + time

#latitud
lat_dir = tc[8]
sec = float('0' + tc[7][4:]) * 60
lat_deg = float(tc[7][:2])
lat_min = float(tc[7][2:4])
lat_sec = sec
lat = (((lat_sec/60) / 60) + (lat_min/60) + lat_deg)
lat = ((-1, 1)[lat_dir != 'S']) * lat
lat = str(lat)

#longitud

```

```

lon_dir = tc[10]
sec = float('0' + tc[9][5:]) * 60
lon_deg = float(tc[9][:3])
lon_min = float(tc[9][3:5])
lon_sec = sec
lon = (((lon_sec / 60) / 60) + (lon_min / 60) + lon_deg)
lon = ((-1, 1)[lon_dir != 'W']) * lon
lon = str(lon)
print 'Lat/Lon: ' + lat + ' ' + lon

con=psycopg2.connect(dsn)    // conectar con postgres
cur=con.cursor()
geom = "GeomFromEWKT('SRID=4326; POINT(" + lon + ' ' + lat + ")'"
sql = """
INSERT INTO contador(trackid, hora, fecha, suben, subent,bajan,bajant,
geom)
VALUES('%s', '%s', '%s', %d, %d, %d, %d, %s );
""" % (trackid, time,date, int(s),int(st),int(b),int(bt), geom)

try:
    cur.execute(sql)
    con.commit()
except:
    print 'Error insert!!!'
cur.close()
con.close()
#break
except:
    print 'Error decoding trama!!!'
linea=archi.readline()
archi.close()

leertxt()

```

Script para lectura del socket en el servidor

```

# -*- coding: utf-8 -*-import socket
import socket
import time
import datetime
#import psycopg2
import os

```

```

dsn = "dbname=pantana3 host=localhost port=5432 user=postgres password=postgres"
#dsn      =      "dbname=pantana      host=localhost      port=5432      user=postgres
password=in1010postgis"

```

```

if __name__ == '__main__':
    #cabecera para envio de configuracion
    header = str(chr(0x00)) + str(chr(0x01)) + str(chr(0x04)) + str(chr(0x00))
    # A UDP server
    # Set up a UDP server
    UDPSock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
    # Listen on port 21567
    # (to all IP addresses on this system)21571
    listen_addr = ("",22000)
    UDPSock.bind(listen_addr)
    # def hex2dec(s):
    #     """return the integer value of a hexadecimal string s"""
    #     return int(s, 16)
    #trama='??                                9                                123456789,DE  12
$GPRMC,165606.00,A,0020.459976,N,07807.319924,W,0.0,0.0,240212,2.5,E,A*2B'
    prev_map = ""
    while True:
        print '\n...'
        data,addr = UDPSock.recvfrom(1024)
        #Envia comando al gps
        #comando = header + 'ATI'
        #UDPSock.sendto(header, addr)
        trama = data.strip()
        print trama
        UDPSock.sendto(trama, ('162.243.103.85', 7777))
        os.system('echo "' + trama[5:] + '" >> contador.log')

```

Código para skypatrol tt8750+

```

at+cgdcont=1,"ip","internet.claro.com.ec" // apn

AT$TTSRVDST=1,1,"198.50.158.43",22000,2 // puerto ip

AT$TTCNT=1,2,60 // timer 1

AT$TTCNT=2,2,30 // timer 2

```

AT\$TMSGMASK=01,1675462 // formato gprmc

AT\$TTSTOCMD=1,AT\$TTSNDMG=2,"Prueba de pasajeros" // comando
almacenamiento

AT\$TTFNT=11,1,14,2,0 // evento activation timer 2
AT\$TTFNT=11,2,9,1,1 // evento sensado de igniion on
AT\$TTFNT=11,3,26,11,1 // envio de trama

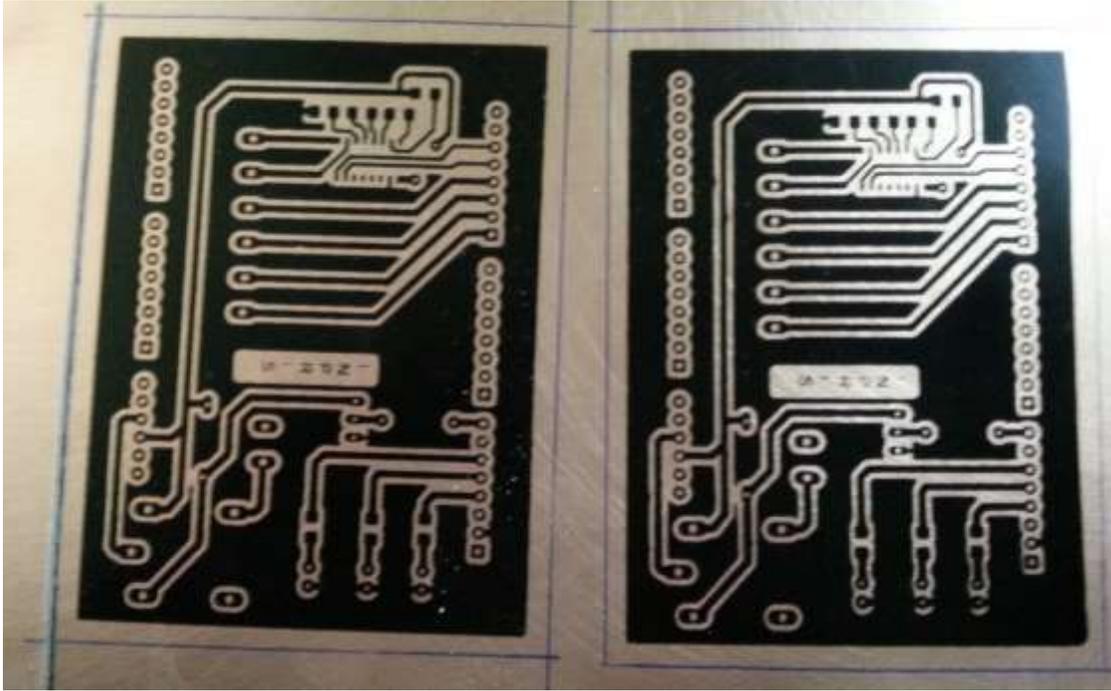
AT\$TTFNT=12,1,14,1,0 // evento timer 1
AT\$TTFNT=12,2,9,0,0 // ignicion off
AT\$TTFNT=12,3,26,12,1 // envio de trama

AT\$TTFNT=13,1,0,0,0 // sensa estado input 1 a LOW
AT\$TTFNT=13,3,28,1,0 // ejecuta comando almacenado 1
AT\$TTFNT=13,3,26,13,1 // confirmacion de estado del input 1

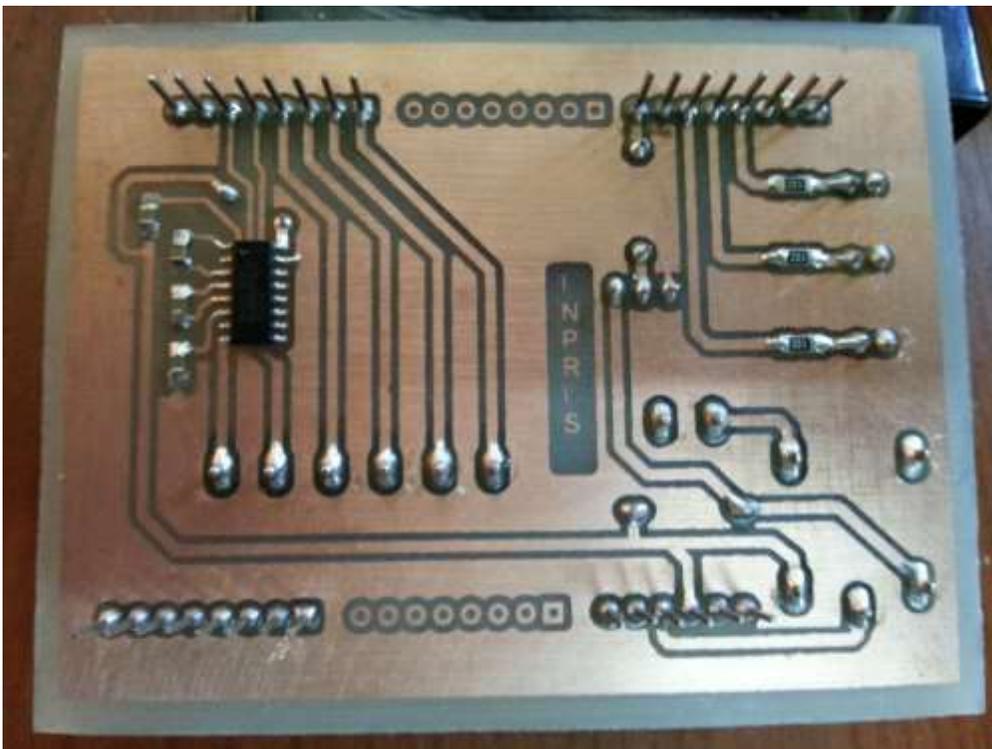
AT\$TTSTOCMD=2,AT\$TTSNDMG=0,"distancia" // comando almacenamiento
AT\$TTCNT=3,1,10,0
AT\$TTFNT=14,1,14,3,1 // sensa estado input 1 a LOW
AT\$TTFNT=14,3,28,2,0 // ejecuta comando almacenado 1
AT\$TTFNT=14,3,26,14,1 // confirmacion de estado del input 1

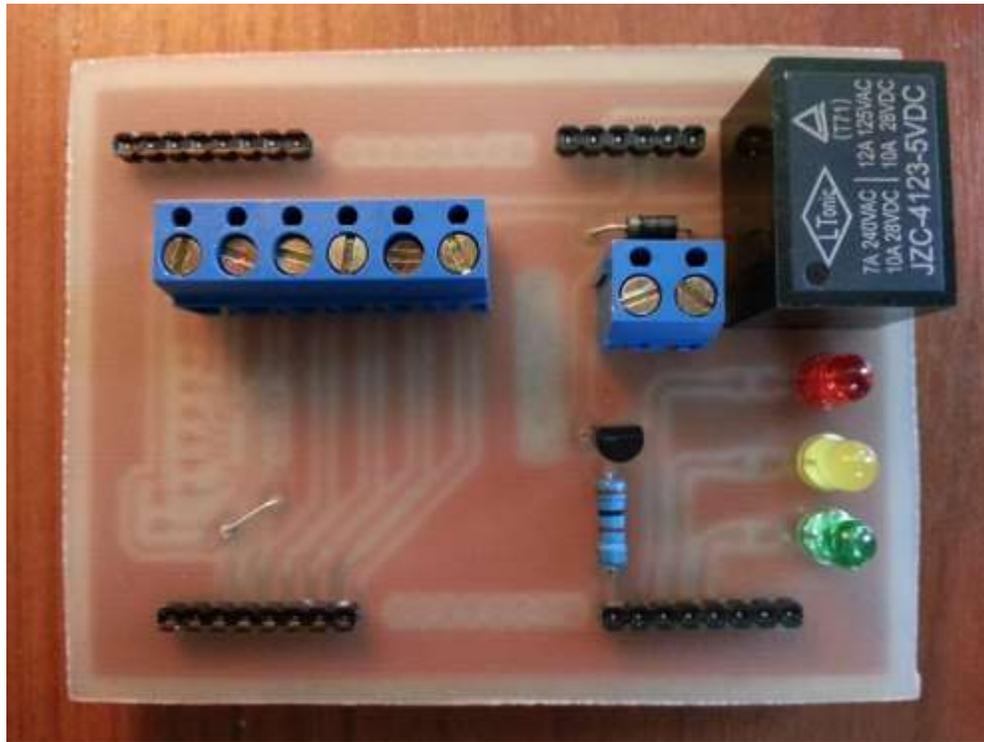
AT\$TTSMSDST=1,"+593984672008"
AT\$TTFNT=14,3,30,14,1
AT\$TTARG=2
at&w

Diseño final placas de circuito impreso

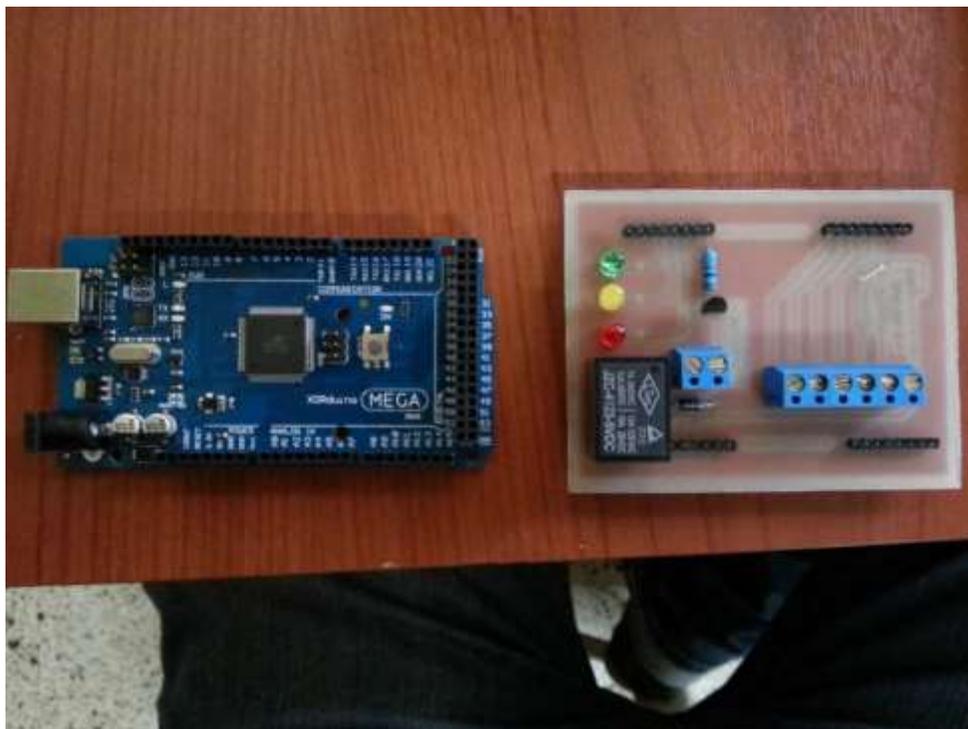


Soldado y Montaje de componentes





Arduino Mega con Shield



Montaje Final



Funcionamiento en el vehículo

