

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA DE INGENIERIA ELECTRICA



**“Levantamiento topográfico basado en equipo  
GPS con comunicación inalámbrica y publicación  
de resultados en aplicativo WEB.”**

PRESENTADO POR:

**JOSÉ ROBERTO MARTÍNEZ MEJÍA**

PARA OPTAR AL TITULO DE:

**INGENIERO ELECTRICISTA**

CIUDAD UNIVERSITARIA, DICIEMBRE DE 2009

**UNIVERSIDAD DE EL SALVADOR**

**RECTOR :**

**MSc. RUFINO ANTONIO QUEZADA SÁNCHEZ**

**SECRETARIO GENERAL :**

**LIC. DOUGLAS VLADIMIR ALFARO CHÁVEZ**

**FACULTAD DE INGENIERIA Y ARQUITECTURA**

**DECANO :**

**ING. MARIO ROBERTO NIETO LOVO**

**SECRETARIO :**

**ING. OSCAR EDUARDO MARROQUÍN HERNÁNDEZ**

**ESCUELA DE INGENIERIA ELECTRICA**

**DIRECTOR :**

**ING. JOSE WILBER CALDERON URRUTIA**

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA DE INGENIERIA ELECTRICA

Trabajo de Graduación previo a la opción al Grado de:

**INGENIERO ELECTRICISTA**

Título :

**“Levantamiento topográfico basado en equipo  
GPS con comunicación inalámbrica y publicación  
de resultados en aplicativo WEB.”**

Presentado por :

**JOSÉ ROBERTO MARTÍNEZ MEJÍA**

Trabajo de Graduación Aprobado por :

Docente Director :

**ING. HUGO MIGUEL COLATO RODRÍGUEZ**

San Salvador, Diciembre de 2009

Trabajo de Graduación Aprobado por:

Docente Director:

ING. HUGO MIGUEL COLATO RODRÍGUEZ

## ACTA DE CONSTANCIA DE NOTA Y DEFENSA FINAL

En esta fecha, 17 de Noviembre de 2009, en la Sala de Lectura, a las 17:00 horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

1. Ing. José Wilber Calderón Urrutia  
Director
2. Ing. Salvador de Jesús German  
Secretario



Firma:

*Wilber Calderón*  
*Salvador de Jesús German*

Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:

Firma:

- 1- Ing. José Enrique Hernández Jeréz
- 2- Dr. Carlos Eugenio Martínez Cruz

*Enrique Hernández Jeréz*  
*Carlos Eugenio Martínez Cruz*

Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

“Levantamiento topográfico basado en equipo GPS con comunicación inalámbrica y publicación de resultados en aplicativo WEB”

A cargo del Bachiller:

MARTÍNEZ MEJÍA, JOSE ROBERTO

Habiendo obtenido el presente Trabajo una nota final, global de: 9.5

( nueve punto cinco )

## **AGRADECIMIENTOS**

Le agradezco en primer lugar a Dios por haberme dado todo lo necesario para salir adelante en esta vida, todo lo que a mí me parece malo y bueno porque me ayudaron a ser una mejor persona.

En segundo lugar le agradezco a mis padres por todo el apoyo que siempre me han brindado; le agradezco a toda mi familia por la unidad que hemos tenido en todos los momentos de la vida; a mis amigos y compañeros de estudio por todo lo que hemos pasado en cada etapa de la vida; al igual a todos los catedráticos que he tenido por ser pacientes y enseñarme lo necesario para el desarrollo de mi vida productiva.

También le agradezco a todas las personas que me ayudaron a desarrollar este proyecto de tesis, a las personas le digo: Por Todo Gracias.

**Martínez Mejía, José Roberto**

## **PREFACIO**

A medida que el tiempo va pasando la tecnología facilita más la vida diaria al igual que lo hace en todo nuestro entorno, como también en el trabajo que desarrollamos. Cuando se necesita un instrumento se toman las decisiones necesarias para desarrollarlo y ponerlo a disposición de las personas que lo necesitan.

Lo que se pretende es abrir una brecha para que se pueda desarrollar tecnología, que esté a nuestro alcance económico, para facilitar el trabajo a las personas hacia donde este dirigido el proyecto a desarrollar. En el caso de Ingeniería Eléctrica se tiene los conocimientos para desarrollar dispositivos que pueden facilitar el buen funcionamiento de proyectos que estamos ejecutando al igual que proyectos que se ejecutan en otras áreas como medicina al crear dispositivos que pueden detectar una enfermedad a tiempo, al igual que se pueden crear dispositivos para todas las áreas de la ingeniería.

Para el caso del presente proyecto se desarrolla un dispositivo que enfocado hacia ingeniería civil, como es el caso de realizar levantamientos topográficos, no siendo un sustituto de la tecnología existente sino una herramienta más para la implementación de este tipo de trabajo. Se pretende facilitar el envío de datos desde el lugar en que se está realizando el levantamiento y así ahorrar tiempo en la ejecución de este tipo de trabajo, al poder enviarse los datos tomados por medio de la red de telefonía celular y al final poder observar el trabajo realizado en el lugar donde se tomaron los datos sin tener que movilizarse hasta las oficinas.

## RESUMEN DEL TRABAJO

En la ejecución de un proyecto este se realiza por etapas se colocan las base y luego se coloca el resto. En este proyecto se ha implementado de la misma manera por secciones: el marco teórico, la sección de hardware y luego la sección de software.

El proyecto se puede desglosar en dos partes: la primera es un equipo móvil, que se encargara de recolectar la parte de la información para realizar un levantamiento topográfico, esta información son las coordenadas geográficas del terreno al que se quiere realizar el levantamiento; la segunda parte es la parte que se encuentra en oficina y que se encarga de guardar los dato recolectados por el equipo móvil, además de crear los archivos necesarios para posteriormente ser puestos a disposición de las personas correspondiente por medio de internet en un aplicativo Web.

La parte móvil, denominada Equipo Cliente, está conformada por un receptor GPS, el cual es el equipo del que se extraerán las coordenadas de posicionamiento del terreno. Otra parte del equipo móvil es un circuito electrónico que se basa en un microcontrolador, que es el que realiza la tarea de administración del equipo móvil; extrae los datos del receptor GPS para luego guardarlos en una memoria EEPROM y al terminar de realizar el levantamiento, enviará los datos almacenados al equipo de oficina por medio de un teléfono celular que el microcontrolador utiliza para tal fin; los datos enviados van a través de la red de telefonía móvil, específicamente por mensajes de texto.

La otra parte es la que se encuentra en oficina, denominada equipo Servidor, está conformada por un teléfono, celular que sirve para la comunicación inalámbrica con el equipo móvil o equipo Cliente. Además de un ordenador que funciona como servidor Web para poder acceder a los datos de los levantamiento realizados. En este ordenador se encuentra el software para la comunicación del teléfono celular que está conectado por medio de un puerto USB y así poder obtener los mensajitos que el equipo móvil a enviado. Estos mensajes de texto son guardados en una base de datos geoespacial la cual es POSTGRE con la extensión geoespacial POSTGIS. También se posee un programa ejecutándose como servicio desarrollado con tecnología JAVA el cual es el encargado de procesar los mensajitos para extraerle los datos latitud, longitud y altura; además el software intermedio, denominado Aplicativo Intermedio o Geodeamon, también crea los archivos en formato KML, para poder ser observados en el programa Google Earth y así ver el levantamiento realizado.

# TABLA DE CONTENIDOS

<b>OBJETIVOS.....</b>	<b>1</b>
Objetivo General.....	1
Objetivos Específicos .....	1
<b>CAPITULO I.....</b>	<b>2</b>
<b>DEFINICIONES Y DESCRIPCION DE PROTOCOLOS DE COMUNICACION .....</b>	<b>2</b>
<b>1.1.    Introducción. ....</b>	<b>2</b>
<b>1.2.    Levantamiento topográfico .....</b>	<b>2</b>
1.2.1.    Topografía. ....	2
1.2.2.    Levantamiento topográfico:.....	3
<b>1.3.    GPS.....</b>	<b>4</b>
1.3.1.    Orígenes del GPS .....	5
1.3.2.    Funcionamiento del sistema GPS .....	5
<b>1.4.    Protocolo NMEA.....</b>	<b>8</b>
1.4.1.    Requerimientos en la comunicación con NMEA.....	9
1.4.2.    Frasas de comunicación NMEA.....	9
<b>1.5.    Protocolo MODBUS.....</b>	<b>13</b>
1.5.1.    Modo RTU.....	14
1.5.2.    Modo ASCII .....	15
1.5.3.    Campo de dirección de la trama MODBUS .....	16
1.5.4.    campo de función de la trama MODBUS .....	16
1.5.5.    Campo de datos de la trama MODBUS.....	17
1.5.6.    Campo de error de la trama MODBUS.....	18
<b>1.6.    Telefonía celular.....</b>	<b>18</b>
1.6.1.    SMS.....	19
1.6.2.    Protocolo GSM para SMS.....	21
<b>1.7.    Comandos AT.....</b>	<b>23</b>
1.7.1.    V.250 .....	25
1.7.2.    Descripción de comandos AT en modo texto.....	26

<b>1.8. Demonio GAMMU y GNOKY.....</b>	<b>29</b>
1.8.1. GAMMU.....	29
1.8.2. GAMMU como servicio.....	30
<b>1.9. Bases de datos.....</b>	<b>32</b>
1.9.1. Características.....	33
1.9.2. Tipos de bases de datos.....	33
1.9.3. POSTGIS.....	36
<b>1.10. JAVA.....</b>	<b>37</b>
1.10.1. Hilos demonios (procesos). ....	38
 <b>CAPITULO II .....</b>	 <b>40</b>
 <b>DISEÑO E IMPLEMENTACION DE LA SOLUCION DEL SISTEMA CLIENTE.....</b>	 <b>40</b>
2.1. Introducción. ....	40
2.2. El sistema. ....	40
2.3. Criterios de selección del equipo cliente .....	42
2.3.1. Criterios para la selección del dispositivo receptor GPS.....	43
2.3.2. Criterios para la selección del microcontrolador.....	45
2.3.3. Criterios para la selección de la memoria externa. ....	47
2.3.4. Criterios para la selección del teléfono celular.....	47
2.4. Implementación del equipo cliente .....	49
2.4.1. Comunicación GPS y microcontrolador .....	50
2.4.2. Comunicación microcontrolador memoria EEPROM. ....	51
2.4.3. Comunicación microcontrolador celular.....	52
2.5. Funcionamiento del equipo cliente. ....	54
2.6. Flujo de información en el equipo cliente. ....	57
2.7. Conclusiones del capítulo. ....	63
 <b>CAPITULO III.....</b>	 <b>64</b>
 <b>DISEÑO E IMPLEMENTACION DEL SISTEMA SERVIDOR .....</b>	 <b>64</b>
3.1. Introducción. ....	64
3.2. Arquitectura del diseño de la solución al lado de la oficina.....	64

3.1.1. Criterios para el teléfono celular.....	65
3.1.2. Criterios para la selección del ordenador.....	66
3.3. Requerimientos de software.....	67
3.4. Funcionamiento del Equipo Servidor.....	67
3.5. Conclusiones del capítulo.....	80
<b>CONCLUSIONES.....</b>	<b>81</b>
<b>REFERENCIAS BIBLIOGRAFICAS.....</b>	<b>83</b>
<b>ANEXOS.....</b>	<b>84</b>
<b>A. DATASHETT de dispositivos utilizados.....</b>	<b>84</b>
A.1. Microcontrolador PIC 16F877.....	84
A.2. La memoria EEPROM 24LC512 .....	85
A.3. El MAX232 .....	86
A.4. LCD de NOKIA 3310.....	87
A.5. El teléfono celular Motorola k1 .....	87
A.6. El teléfono celular Nokia 3320 .....	88
<b>B. Manual de usuario.....</b>	<b>89</b>
B.1. Instalación de software.....	89
B.2. Utilización del Equipo Cliente.....	90
B.3. Utilización del Equipo Servidor.....	93
<b>C. Mejoras al sistema.....</b>	<b>95</b>
<b>D. Costos de elementos utilizados en la construcción del sistema.....</b>	<b>96</b>

## LISTA DE TABLAS

Tabla 1.1 Nombres de las diferentes tramas NMEA que podemos encontrar. ....	9
Tabla 1.2 Descripción de los campos de la frase NMEA GPGGA .....	11
Tabla 1.3 Nombres de las diferentes tramas NMEA de propiedad de GARMIN para el envío de información. ....	12
Tabla 1.4 Nombres de las diferentes tramas NMEA de propiedad de GARMIN para la recepción de información. ....	12
Tabla 1.5 formato de la latitud y longitud dentro de una trama NMEA. ....	13
Tabla 1.6 Descripción de una trama MODBUS en modo RTU. ....	15
Tabla 1.7 Descripción de una trama MODBUS en modo ASCII. ....	16
Tabla 1.8 Funciones respaldadas por MODBUS en dispositivos Modicom. ....	17
Tabla 1.9 Definición principales de secuencias de caracteres para comandos AT .....	25
Tabla 1.10 Descripción de los campos de la frase NMEA GGA Parámetros del comando +CMGF. ....	26
Tabla 1.11 Diferentes tipos de error en los comandos AT. ....	27
Tabla 1.12 Parámetros del comando +CNMI. ....	28
Tabla 1.13 Parámetros del comando +CMGS. ....	29
Tabla 2.1 Parámetros del comando +CMGS. ....	29
Tabla 2.2 Descripción de la función de la trama MODBUS. ....	29

## LISTA DE FIGURAS

Figura 1.1. Representación de un levantamiento topográfico con teodolito	4
Figura 1. 2 Representación de recepción de señal GPS en un punto de la tierra	6
Figura 1. 3 Representaciones de la señal de tres satélites GPS.	6
Figura 2. 1 Representación del Sistema de Levantamiento topográfico.	40
Figura 2. 2 Representación del Sistema en forma de Módulos del Levantamiento topográfico.	41
Figura 2. 3 Diseño de la solución del proyecto.	42
Figura 2. 4 Representación en forma de Módulos del Equipo Cliente.	43
Figura 2.4 Diagrama eléctrico implementado en el equipo móvil	49
Figura 2. 5 circuito integrado MAX232.	50
Figura 2. 6 Conexión de dispositivos en un bus I2C.	52
Figura 2. 7 Puerto de comunicación del celular (POP-PORT).	53
Figura 2. 8 Pantalla principal de receptor GPS.	54
Figura 2. 9 Flujo de información entre receptor GPS y microcontrolador.	58
Figura 2. 10 Flujo de información entre microcontrolador y teléfono celular.	59
Figura 2. 11 Diagrama de flujo de la rutina principal del microcontrolador.	59
Figura 2. 12-a Diagrama de flujo de la rutina de adquisición de datos georeferenciados.	60
Figura 2. 12-b Diagrama de flujo de la parte de la rutina para borrar un punto almacenado.	61
Figura 2. 13 Diagrama de flujo de rutina de recepción de mensajitos.	62
Figura 3. 1 Representación del equipo Servidor.	64
Figura 3. 2 Marcas de teléfonos soportados por Gammu.	66
Figura 3. 3 Representación de cómo interactúa el software en el equipo servidor.	68
Figura 3. 4 Diagrama de flujo del aplicativo intermedio (Geodeamon).	75
Figura 3. 5 Flujo de información del Equipo Servidor.	76
Figura 3. 6 Pagina principal del aplicativo Web.	77
Figura 3. 7 Pagina de resultados al observar los puntos de un levantamiento.	78
Figura 3. 8 Levantamiento topográfico visualizado en Google Earth.	79
Figura B.1 Equipo Cliente al ser encendido.	91
Figura B.2 Equipo Cliente listo para la toma de datos.	92
Figura B.2 Equipo Cliente listo para la toma de datos.	94

# OBJETIVOS

## Objetivo General

Diseñar y construir el software y hardware prototipo, que sea capaz de realizar procesos de levantamientos topográficos utilizando un dispositivo GPS, y publicar los resultados en un ordenador conectado a Internet o Intranet, a través de un aplicativo Web creado con tecnología N Capas.

## Objetivos Específicos

- Diseñar y construir el hardware necesario para la adquisición y envío de las coordenadas del GPS a una base de datos, por medio de un sistema inalámbrico.
- Desarrollar una aplicación de sub monitoreo para interpretar la información recibida en la base de datos que ha sido enviada por la estación móvil.
- Desarrollar el software en la plataforma Java para el procesamiento de las lecturas realizadas en campo, tal que sea capaz de construir el plano como un archivo en formato vector o una imagen, dejando estos disponibles para su acceso por la Web.
- Implementar protocolo MODBUS para la comunicación, a través de un medio inalámbrico, entre el equipo cliente y el equipo servidor.

# **CAPITULO I**

## **DEFINICIONES Y DESCRIPCION DE PROTOCOLOS DE COMUNICACION**

### **1.1. Introducción.**

En el primer capítulo se presentan las bases teóricas necesarias para la ejecución del proyecto desarrollado; se presentan definiciones y explicaciones. Se da una pequeña explicación de los protocolos implementados. Se presenta la información necesaria de lo que es un levantamiento topográfico. También está la información del sistema de posicionamiento GPS orígenes y desarrollo, además de la forma en cómo se comunican los dispositivos receptores GPS, mediante el protocolo NMEA. Al igual que se encuentra la forma en que este proyecto puede ser exportado hacia otro sistema y no quedar aislado, este es el Protocolo MODBUS, que es una forma de comunicación entre dispositivos donde se encuentra la forma maestro-esclavo. Además una explicación de la forma de comunicación inalámbrica que posee el proyecto, que es a base de telefonía celular y más específicamente: mensajitos de texto (SMS). Se encuentra la explicación y forma de configuración del software necesario para la comunicación entre un teléfono celular y el ordenador, este software es llamado GAMMU, el cual se utiliza para la extracción de los mensajitos para ser guardados en una base de datos dentro de un ordenador. Esta la parte de cómo son tratada la información que se ha recolectado del levantamiento topográfico, esto se realiza en el ordenador en base a un programa implementado en lenguaje JAVA con características de programación de procesos residentes en memoria.

### **1.2. Levantamiento topográfico**

#### **1.2.1. Topografía.**

La topografía es un conjunto de principios y procedimientos que tienen por objeto la representación gráfica de una parte de la superficie terrestre, con sus formas y detalles tanto naturales como artificiales. Procede de topo (lugar) y grafos (descripción). Esta representación tiene lugar sobre superficies planas, limitándose a pequeñas extensiones de terreno, utilizando la denominación de geodesia para áreas mayores. De manera muy simple, puede decirse que para un topógrafo la Tierra es plana, mientras que para un geodesta no lo es.

Para eso se utiliza un sistema de coordenadas tridimensional, siendo la X y la Y competencia de la planimetría, y la Z de la altimetría.

Los mapas topográficos utilizan el sistema de representación de planos acotados, mostrando la elevación del terreno utilizando líneas que conectan los puntos con la misma cota respecto de un plano de referencia, denominadas curvas de nivel, en cuyo caso se dice que el mapa es

*hipsográfico*. Dicho plano de referencia puede ser o no el nivel del mar, pero en caso de serlo se hablará de altitudes en lugar de cotas.

La topografía es aplicada a la descripción de la realidad física inmóvil circundante, es plasmar en un plano topográfico la realidad vista en campo, en el ámbito rural ó natural, de la superficie terrestre; en el ámbito urbano, es la descripción de los hechos existentes en un lugar determinado: muros, edificios, calles, entre otros.

Actualmente el método más utilizado para la toma de datos se basa en el empleo de una estación total, con la cual se pueden medir ángulos horizontales, ángulos verticales y distancias. Conociendo las coordenadas del lugar donde se ha colocado la Estación es posible determinar las coordenadas tridimensionales de todos los puntos que se midan.

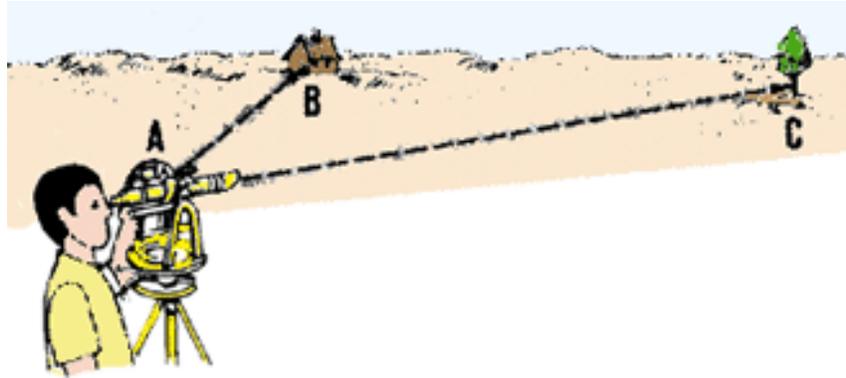
Procesando posteriormente las coordenadas de los datos tomados es posible dibujar y representar gráficamente los detalles del terreno considerados. Con las coordenadas de dos puntos se hace posible además calcular las distancias o el desnivel entre los mismos puntos aunque no se hubiese estacionado en ninguno.

Se considera en topografía como el proceso inverso al replanteo, pues mediante la toma de datos se dibuja en planos los detalles del terreno actual. Este método está siendo sustituido por el uso de GPS, aunque siempre estará presente pues no siempre se tiene cobertura en el receptor GPS por diversos factores (ejemplo: dentro de un túnel). El uso del GPS reduce considerablemente el trabajo, pudiéndose conseguir precisiones buenas de 2 a 3 cm si se trabaja de forma cinemática y de incluso 2 mm de forma estática.

La topografía tiene aplicaciones en las diferentes ramas de la ingeniería; dentro de ingeniería agrícola, tanto en levantamientos como en trazos, deslindes, divisiones de tierra (agrodesia) determinación de área, etc. En la ingeniería eléctrica: en los levantamientos previos y los trazos de líneas de transmisión, construcción de plantas hidroeléctricas, en instalación de equipos para plantas nucleoeeléctricas, etc. En ingeniería mecánica e industrial: para la instalación precisa de maquinas y equipos industriales, configuración de piezas metálicas de gran precisión, etc. En la ingeniería civil: en ella es necesario realizar trabajos topográficos antes, durante y después de la construcción de obras tales como carreteras, ferrocarriles edificios, puentes, canales, presas, etc.

### **1.2.2. Levantamiento topográfico:**

Se denomina levantamiento al conjunto de operaciones necesarias para representar topográficamente un terreno. Aunque en general todo levantamiento ha de hacerse con precisiones ya establecidas, hay ocasiones en que, por la índole del trabajo, puede aligerarse aún cuando lleguen a cometerse errores sensibles en el plano, e incluso, a veces, solo es necesario un ligero bosquejo, con rápidas medidas, constituyendo un croquis. La figura 1.1 muestra la representación de cómo se efectúan las mediciones con un teodolito.



**Figura 1.1.** Representación de un levantamiento topográfico con teodolito

Los levantamientos topográficos en cuanto a su calidad se dividen en:

- **PRECISOS**, que se ejecutan por medio de triangulaciones o poligonales de precisión. Se emplean para fijar los límites entre naciones o estados, en el trazo de ciudades, etc.
- **REGULARES**, los cuales se realizan por medio de poligonales, levantamiento de tránsito y cinta. Se usan para levantar linderos de propiedades, para el trazo de caminos, vías férreas, canales, etc.
- **TAQUIMÉTRICOS**, en los cuales las distancias se miden por procedimientos indirectos. Generalmente se ejecutan por tránsito y estadía, y se emplean en trabajos previo al trazo de vías de comunicación, en trabajos de configuración y rellenos, y también para la formación de planos a pequeña escala.
- **EXPEDITIVOS**, efectuados con aparatos portátiles, poco precisos, como: brújula, sextante, podómetro, telemetro, etc., y cuando no se dispone de aparatos se ejecutan a ojo o por informe proporcionados por los habitantes de la región. Estos levantamientos se emplean en reconocimiento del terreno o en las exploraciones militares.

### 1.3. GPS

El **Global Positioning System (GPS)** o **Sistema de Posicionamiento Global** (más conocido con las siglas *GPS*, aunque su nombre correcto es **NAVSTAR-GPS**) es un Sistema Global de Navegación por Satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave, con una precisión hasta de centímetros, usando GPS diferencial, lo habitual son unos pocos metros. Aunque su invención se atribuye a los gobiernos francés y belga, el sistema fue desarrollado e instalado, y actualmente es operado, por el Departamento de Defensa de los Estados Unidos.

### 1.3.1. Orígenes del GPS

El lanzamiento del satélite espacial estadounidense *Vanguard* en 1959 puso de manifiesto que la transmisión de señales de radio desde el espacio podría servir para orientarnos y situarnos en la superficie terrestre o, a la inversa, localizar un punto cualquiera en la Tierra.

En 1993 el Departamento de Defensa de los Estados Unidos de América, basado en la experiencia recogida del satélite *Vanguard* (en un principio para uso exclusivamente militar) puso en funcionamiento un sistema de localización por satélite conocido por las siglas en inglés GPS.

En sus inicios el propio Departamento de Defensa programó errores de cálculo codificados en las transmisiones de los satélites GPS para limitarlo solamente a la actividad militar que sí contaba con decodificadores para interpretar correctamente las señales, pero a partir de mayo de 2000 esta práctica quedó cancelada y hoy en día el sistema GPS se utiliza ampliamente en muchas actividades de la vida civil, aunque no está exento de ser reprogramado de nuevo en caso de cualquier conflicto bélico.

Este sistema permite conocer la posición y la altura a la nos encontramos situados en cualquier punto de la Tierra en todo momento, ya sea que estemos situados en un punto fijo sin desplazarnos, e incluso en movimiento, tanto de día como de noche.

El sistema GPS permite rastrear también, en tiempo real, la ubicación de una persona, animal, vehículo, etc., desde cualquier sitio y prestar auxilio si fuera necesario, con la condición que esté equipado con un dispositivo que pueda emitir algún tipo de señal, ya sea de radio o telefonía, que permita su localización.

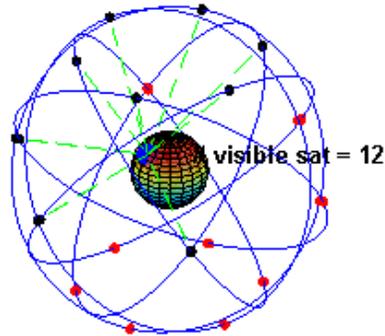
La primera prueba exitosa del sistema GPS desde el punto de vista práctico como instrumento de ayuda a la navegación, la realizó el trasbordador espacial *Discovery* en el propio año que se puso en funcionamiento el sistema. Actualmente los satélites GPS pertenecen a una segunda generación denominada *Block II*.

### 1.3.2. Funcionamiento del sistema GPS

El sistema se compone de 24 satélites distribuidos en seis órbitas polares diferentes, situadas a 2,169 kilómetros (11 000 millas) de distancia de la Tierra; cada satélite la circunvala dos veces cada 24 horas. Por encima del horizonte siempre están “visibles” para los receptores GPS por lo menos 4 satélites, de forma tal que puedan operar correctamente desde cualquier punto de la Tierra donde se encuentren situados. Una representación de la cantidad de satélites vista por los receptores GPS está en la figura 1.2 que en ese punto de la tierra el receptor GPS recibe la señal de 12 satélites.

Los receptores GPS detectan, decodifican y procesan las señales que reciben de los satélites para determinar el punto donde se encuentran situados y son de dos tipos: **portátiles** y **fijos**. Los

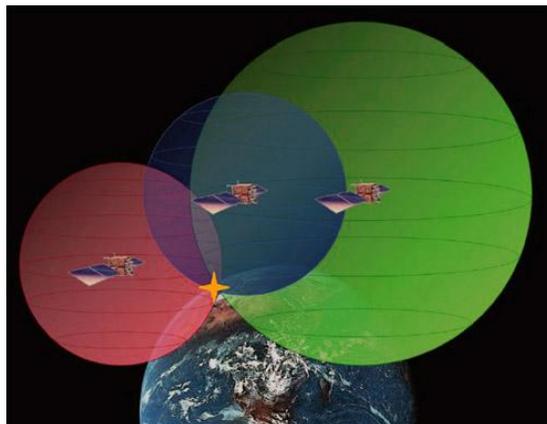
portátiles pueden ser tan pequeños como algunos teléfonos celulares o móviles. Los fijos son los que se instalan en automóviles o coches, embarcaciones, aviones, trenes, submarinos o cualquier otro tipo de vehículo.



**Figura 1. 2** Representación de recepción de señal GPS en un punto de la tierra

El monitoreo y control de los satélites que conforman el sistema GPS se ejerce desde diferentes estaciones terrestres situadas alrededor del mundo, que rastrean su trayectoria orbital e introducen las correcciones necesarias a las señales de radio que transmiten hacia la Tierra. Esas correcciones benefician la exactitud del funcionamiento del sistema, como por ejemplo las que corrigen las distorsiones que provoca la ionosfera en la recepción de las señales y los ligeros cambios que introducen en las órbitas la atracción de la luna y el sol.

El funcionamiento del sistema GPS se basa también, al igual que los sistemas electrónicos antiguos de navegación, en el principio matemático de la triangulación. Por tanto, para calcular la posición de un punto será necesario que el receptor GPS determine con exactitud la distancia que lo separa de los satélites. Con la aplicación del principio matemático de la triangulación podemos conocer el punto o lugar donde nos encontramos situados, e incluso rastrear y ubicar el origen de una transmisión por ondas de radio. El sistema GPS utiliza el mismo principio, pero en lugar de emplear círculos o líneas rectas crea esferas virtuales o imaginarias para lograr el mismo objetivo.



**Figura 1. 3** Representaciones de la señal de tres satélites GPS.

Desde el mismo momento que el receptor GPS detecta una señal de radiofrecuencia transmitida por un satélite desde su órbita, se genera una esfera virtual o imaginaria que envuelve al satélite. El propio satélite actuará como centro de la esfera cuya superficie se extenderá hasta el punto o lugar donde se encuentre situada la antena del receptor; por tanto, el radio de la esfera será igual a la distancia que separa al satélite del receptor. A partir de ese instante el receptor GPS medirá las distancias que lo separan como mínimo de dos satélites más. Para ello tendrá que calcular el tiempo que demora cada señal en viajar desde los satélites hasta el punto donde éste se encuentra situado y realizar los correspondientes cálculos matemáticos. En la figura 1.3 se mostro la forma de como se interceptan las esferas imaginarias para determinar la posición del receptor GPS.

Todas las señales de radiofrecuencias están formadas por ondas electromagnéticas que se desplazan por el espacio de forma concéntrica a partir de la antena transmisora. Debido a esa propiedad las señales de radio se pueden captar desde cualquier punto situado alrededor de una antena transmisora. Las ondas de radio viajan a la velocidad de la luz, es decir, 300 mil kilómetros por segundo (186 mil millas por segundo) medida en el vacío, por lo que es posible calcular la distancia existente entre un transmisor y un receptor si se conoce el tiempo que demora la señal en viajar desde un punto hasta el otro.

Para medir el momento a partir del cual el satélite emite la señal y el receptor GPS la recibe, es necesario que tanto el reloj del satélite como el del receptor estén perfectamente sincronizados. El satélite utiliza un reloj atómico de cesio, extremadamente exacto, pero el receptor GPS posee uno normal de cuarzo, no tan preciso. Para sincronizar con exactitud el reloj del receptor GPS, el satélite emite cada cierto tiempo una señal digital o patrón de control junto con la señal de radiofrecuencia. Esa señal de control llega siempre al receptor GPS con más retraso que la señal normal de radiofrecuencia. El retraso entre ambas señales será igual al tiempo que demora la señal de radiofrecuencia en viajar del satélite al receptor GPS.

La distancia existente entre cada satélite y el receptor GPS la calcula el propio receptor realizando diferentes operaciones matemáticas. Para hacer este cálculo el receptor GPS multiplica el tiempo de retraso de la señal de control por el valor de la velocidad de la luz. Si la señal ha viajado en línea recta, sin que la haya afectado ninguna interferencia por el camino, el resultado matemático será la distancia exacta que separa al receptor del satélite.

Las ondas de radio que recorren la Tierra lógicamente no viajan por el vacío sino que se desplazan a través de la masa gaseosa que compone la atmósfera; por tanto, su velocidad no será exactamente igual a la de la luz, sino un poco más lenta. Existen también otros factores que pueden influir también algo en el desplazamiento de la señal, como son las condiciones atmosféricas locales, el ángulo existente entre el satélite y el receptor GPS, etc. Para corregir los efectos de todas esas variables, el receptor se sirve de complejos modelos matemáticos que guarda en su memoria. Los resultados de los cálculos los complementa después con la información adicional que recibe también del satélite, lo que permite mostrar la posición con mayor exactitud.

Para ubicar la posición exacta donde nos encontramos situados, el receptor GPS tiene que localizar por lo menos 3 satélites que le sirvan de puntos de referencia. Para detectar también la altura a la que se encuentra situado el receptor GPS sobre el nivel del mar, tendrá que medir

adicionalmente la distancia que lo separa de un cuarto satélite y generar otra esfera virtual que permitirá determinar esa medición. Si por cualquier motivo el receptor falla y no realiza las mediciones de distancias hasta los satélites de forma correcta, las esferas no se interceptan y en ese caso no podrá determinar, ni la posición, ni la altura.

El DGPS (Differential GPS), o GPS diferencial, es un sistema que proporciona a los receptores GPS correcciones de los datos recibidos de los satélites GPS, con el fin de proporcionar una mayor precisión en la posición calculada. Se concibió fundamentalmente debido a la introducción de la disponibilidad selectiva (SA).

El fundamento radica en el hecho de que los errores producidos por el sistema GPS afectan por igual (o de forma muy similar) a los receptores situados próximos entre sí. Los errores están fuertemente correlacionados en los receptores próximos.

Un receptor GPS fijo en tierra (referencia) que conoce exactamente su posición basándose en otras técnicas, recibe la posición dada por el sistema GPS, y puede calcular los errores producidos por el sistema GPS, comparándola con la suya, conocida de antemano. Este receptor transmite la corrección de errores a los receptores próximos a él, y así estos pueden, a su vez, corregir también los errores producidos por el sistema dentro del área de cobertura de transmisión de señales del equipo GPS de referencia.

## **1.4. Protocolo NMEA**

**NMEA 0183** (o **NMEA** de forma abreviada) es una especificación combinada eléctrica y de datos entre aparatos electrónicos marinos y, también, más generalmente, receptores GPS.

La comunicación de receptores GPS se define dentro de esta especificación. La mayoría programas de computadora que proporcionan información de posición de tiempo real entienden y esperan datos por estar en formato de NMEA. Estos datos incluyen el PVT completo (posición, velocidad, tiempo) solución computada por el receptor de GPS. La idea de NMEA es enviar una línea de datos llamada una frase que es totalmente autosuficiente e independiente de otras frases. Hay frases estándar para cada categoría de dispositivos y también está la habilidad de definir frases propietario para el uso por una compañía individual.

Cada frase empieza con un "\$" y el final con un retorno/línea y no pueden ser más largo que 80 caracteres de texto visible (más el fin de línea). Los datos están dentro de esta línea, con los ítems separados por comas.

Hay una provisión para un checksum al final de cada frase que puede o no puede ser verificada por la unidad que lee los datos. El campo del checksum consiste de un "\*" y dos dígitos hexadecimales que representan 8 bits de OR exclusivo de todos los caracteres entre, pero no incluyendo, el "\$" y el "\*".

### 1.4.1. Requerimientos en la comunicación con NMEA

La interface del hardware para las unidades de GPS se diseña para reunir los requisitos de NMEA. Ellos también son compatibles con la mayoría de los puertos serie de la computadora que usan protocolos RS232, sin embargo estrictamente hablando la norma de NMEA no es RS232. Ellos recomiendan conformidad a EIA-422. La velocidad de la interface puede ajustarse en algunos modelos, pero la norma de NMEA es de 4800 b/s (bits por segundo) con 8 bits de datos, sin paridad, y un bit de stop. Todas las unidades que apoyan NMEA deben apoyar esta velocidad.

### 1.4.2. Frases de comunicación NMEA

NMEA consiste en frases, la primera palabra de que, llamó un tipo de los datos, define la interpretación del resto de la frase. Cualquier dispositivo o programa que leen los datos, pueden mirar para la frase de los datos de que está interesado y simplemente ignora otras frases que no son de su interés.

Algunas frases NMEA las podemos observar en la tabla 1.1, donde se encuentra la cabecera de la trama y el nombre que esta posee.

**Tabla 1.1 Nombres de las diferentes tramas NMEA que podemos encontrar.**

Cabecera	Nombre de la trama
\$GPAAM	Waypoint Arrival Alarm
\$GPALM	GPS Almanac Data
\$GPAPA	Autopilot format "A"
\$GPAPB	Autopilot format "B"
\$GPASD	Autopilot System Data
\$GPBEC	Bearing & Distance to Waypoint, Dead Reckoning
\$GPBOD	Bearing, Origin to Destination
\$GPBWC	Bearing & Distance to Waypoint, Great Circle
\$GPBWR	Bearing & Distance to Waypoint, Rhumb Line
\$GPBWW	Bearing, Waypoint to Waypoint
\$GPDBT	Depth Below Transducer
\$GPDCN	Decca Position
\$GPDPT	Depth
\$GPFSI	Frequency Set Information
\$GPGGA	Global Positioning System Fix Data

\$GPGLC	Geographic Position, Loran
\$GPGLL	Geographic Position, Latitude/Longitude
\$GPGRS	GPS Range Residuals
\$GPGSA	GPS DOP and Active Satellites
\$GPGST	GPS Pseudorange Noise Statistics
\$GPGSV	GPS Satellites in View
\$GPGXA	TRANSIT Position
\$GPHDG	Heading, Deviation & Variation
\$GPHDT	Heading, True
\$GPHSC	Heading Steering Command
\$GPLCD	Loran
\$GPMSK	Control for a Beacon Receiver
\$GPMSS	Beacon Receiver Status
\$GPMTA	Air Temperature (to be phased out)
\$GPMTW	Water Temperature
\$GPMWD	Wind Direction
\$GPMWV	Wind Speed and Angle
\$GPOLN	Omega Lane Numbers
\$GPOSD	Own Ship Data
\$GPR00	Waypoint active route (not standard)
\$GPRMA	Recommended Minimum Specific Loran
\$GPRMB	Recommended Minimum Navigation Information
\$GPRMC	Recommended Minimum Specific GPS/TRANSIT Data
\$GPROT	Rate of Turn
\$GPRPM	Revolutions
\$GPRSA	Rudder Sensor Angle
\$GPRSD	RADAR System Data
\$GPRTE	Routes
\$GPSFI	Scanning Frequency Information
\$GPSTN	Multiple Data ID
\$GPTRF	TRANSIT Fix Data
\$GPTTM	Tracked Target Message
\$GPVBW	Dual Ground/Water Speed
\$GPVDR	Set and Drift
\$GPVHW	Water Speed and Heading

\$GPVLW	Distance Traveled through the Water
\$GPVPW	Speed, Measured Parallel to Wind
\$GPVTG	Track Made Good and Ground Speed
\$GPWCV	Waypoint Closure Velocity
\$GPWNC	Distance, Waypoint to Waypoint
\$GPWPL	Waypoint Location
\$GPXDR	Transducer Measurements
\$GPXTE	Cross
\$GPXTR	Cross
\$GPZDA	UTC Date / Time and Local Time Zone Offset
\$GPZFO	UTC & Time from Origin Waypoint
\$GPZTG	UTC & Time to Destination Waypoint

Un ejemplo de una frase NMEA es la siguiente:

**\$GPGGA,170834,4124.8963,N,08151.6838,W,1,05,1.5,280.2,M,-34.0,M,,,\*75**

La tabla 1.2 muestra la descripción de los diferentes campos de frase NMEA anterior, como el nombre de cada campo, la forma en que se representa dentro de la frase NMEA y la descripción.

**Tabla2.2 Descripción de los campos de la frase NMEA GPGGA**

Nombre	Dato	Descripción
Identificador de frase	\$GPGGA	Datos fijos del sistema de posicionamiento global (Global Positioning System Fix Data)
Tiempo	170834	17:08:34 UTC
Latitud	4124.8963, N	41d 24.8963' N ó 41d 24' 54" N
Longitud	08151.6838, W	81d 51.6838' W ó 81d 51' 41" W
Calidad fijada: - 0 = Invalido - 1 = GPS fijado - 2 = DGPS fijado	1	Los datos desde un GPS fijo
Numero de Satélites	05	5 Satélites son vistos
Horizontal Dilution of Precision (HDOP)	1.5	Exactitud relativa de posición horizontal
Altitud	280.2, M	280.2 metros sobre el nivel del mar
La altura de geoid sobre WGS84 elipsoide	-34.0, M	-34.0 metros
Tiempo desde última actualización de DGPS	blank	No hay actualización

Id de la estación de referencia DGPS	blank	No hay id de la estación
Checksum	*75	Usado por el programa para verificar errores en la transmisión.

Frases NMEA propiedad de GARMIN. Para el envío están en la tabla 1.3 y para la recepción en la tabla 1.4

**Tabla 1.3 Nombres de las diferentes tramas NMEA de propiedad de GARMIN para el envío de información.**

Cabecera	Nombre de la trama
\$HCHDG	Compass Heading
\$PGRMB	DGPS Beacon Information
\$PGRMC	Sensor Configuration Information
\$PGRMC1	Additional Sensor Configuration Information
\$PGRME	Estimated Position Error
\$PGRMF	GPS Position Fix Data
\$PGRMI	Sensor Initialization Information
\$PGRMM	Map Datum
\$PGRMT	Sensor Status Information
\$PGRMV	3D Velocity Information
\$PGRMZ	Altitude Information
\$PSLIB	Tune DPGS Beacon Receiver

**Tabla 1.3 Nombres de las diferentes tramas NMEA de propiedad de GARMIN para la recepción de información.**

Cabecera	Nombre de la trama
\$PGRMC	Sensor Configuration Information
\$PGRMCE	Sensor Configuration Information Enquiry
\$PGRMC1	Additional Sensor Configuration Information
\$PGRMC1E	Additional Sensor Configuration Information Enquiry
\$PGRMI	Sensor Initialization Information
\$PGRMIE	Sensor Initialization Information Enquiry
\$PGRMO	Output Sentence Enable / Disable
\$PSLIB	Tune DPGS Beacon Receiver

## FORMATO DE LATITUD Y LONGITUD

Cuando una latitud o longitud es dada, los dos dígitos inmediatamente a la izquierda del punto decimal son minutos enteros, y los de la derecha del punto decimal son las fracciones de los minutos, los restantes dígitos a la izquierda son los grados. Un ejemplo de esto es la tabla 1.5 donde se muestra la forma en que la trama NMEA coloca cada campo y la forma de cómo debe interpretarse.

**Tabla 1.5 formato de la latitud y longitud dentro de una trama NMEA.**

Campo		Grados	minutos
Latitud	4533.35	45	33.35
Longitud	08151.6838	081	51.6838

## 1.5. Protocolo MODBUS

Es un protocolo de comunicaciones situado en el nivel 7 del Modelo OSI, basado en la arquitectura maestro/esclavo o cliente/servidor, diseñado en 1979 por Modicon para su gama de controladores lógicos programables (PLCs).

Protocolo MODBUS es una estructura de mensajería, ampliamente utilizados para establecer comunicación maestro-esclavo entre dispositivos inteligentes. Un mensaje MODBUS enviado por un maestro a un esclavo contiene la dirección del esclavo, el comando (por ejemplo, " leer registro "o" escribir registro "), los datos, y una suma de verificación (LRC o CRC).

Dado que el protocolo Modbus es sólo una estructura de mensajería, es independiente de la capa física subyacente. Es tradicionalmente aplicado mediante RS232, RS422 o RS485

### ➤ **La consulta**

La función de código en la consulta le dice al esclavo direccionado de qué tipo de acción se desea ejecutar. Los bytes de datos contiene toda la información adicional que el esclavo necesita para llevar a cabo la función. Por ejemplo, la función código 03 se consulta al esclavo a leer los registros de las explotaciones y responder con su contenido. El campo de datos debe contener la información diciendo al esclavo cual es el registro para iniciar y los registros a leer. El campo de error de verificación proporciona un método al esclavo para validar la integridad del contenido de los mensajes.

### ➤ **La respuesta**

Si el esclavo hace una respuesta normal, el código de la función de la respuesta es un eco de la función de código en la consulta. Los bytes de datos contienen los datos recopilados por el esclavo, tales como registro de valores o condición. Si se produce un error, el código de la función se ha modificado para indicar que la respuesta es una respuesta de error, y los bytes de datos contienen un código que describe el error. El error permite comprobar al maestro que el contenido de los mensajes es válido.

Controladores puede ser configurado para comunicarse en redes Modbus estándar utilizando cualquiera de los dos modos de transmisión: RTU o ASCII.

### 1.5.1. Modo RTU

Cuando los controladores se configuran para comunicarse en una red usando Modbus RTU Unidad de terminal remota (Remote Terminal Unit), cada ocho bits (1 byte) en un mensaje contiene cuatro caracteres hexadecimales de dos bits. La principal ventaja de esta modalidad es que su carácter de mayor densidad permite un mejor rendimiento que los datos ASCII para la misma velocidad en baudios. Cada mensaje debe ser transmitido en un flujo continuo.

➤ **Sistema de Codificación**

Ocho bits binario, hexadecimal 0-9, A-F.

Dos caracteres hexadecimales contienen en cada campo ocho bits del mensaje.

➤ **Bits por Byte**

1 bit de arranque

8 bits de datos, el bit menos significativo se envía primero.

1 bit para paridad Par / Impar --- sin bits para no paridad.

1 bit de parada, si la paridad se utiliza---2 bits si no hay paridad.

➤ **Verificación de error**

De redundancia cíclica (CRC).

➤ **Trama RTU**

En modo RTU, los mensajes comienzan con un intervalo de silencio de al menos 3,5 veces el carácter. Esto es más fácil de implementar como un múltiplo de veces en la tasa de baudios que se utiliza en la red. El primer campo es transmitido la dirección del dispositivo.

La transmisión de caracteres permitidos para todos los campos es hexadecimal 0-F. Cuando el primer campo (el campo de la dirección) es recibido, cada dispositivo lo decodifica para averiguar si se trata de abordar el dispositivo.

Después de la última transmisión de caracteres, un intervalo de al menos 3,5 veces el carácter que marca el final del mensaje; un nuevo mensaje puede comenzar después de este intervalo.

La trama del mensaje debe ser transmitido completamente como un flujo continuo. Si un intervalo de silencio de más de 1,5 veces el carácter se produce antes de la terminación de la trama, el dispositivo receptor vuelca el mensaje incompleto y asume que el próximo byte será el campo de la dirección de un nuevo mensaje.

Del mismo modo, si un mensaje nuevo comienza antes de 3,5 veces el carácter anterior a raíz de un mensaje, el dispositivo receptor lo considera una continuación del mensaje anterior. De esta forma, establecerá un error, ya que el valor final en el campo CRC no será válida para el

conjunto de mensajes La descripción de los diferentes campos de la trama se muestra a en la tabla 1.6.

**Tabla 1.6 Descripción de una trama MODBUS en modo RTU.**

Start	Address	Function	Data	CRC	End
3.5 Char time	8 Bit	8 Bit	N * 8Bit	16 Bit	3.5 Char time

### 1.5.2. Modo ASCII

Cuando los controladores son configurados para comunicarse en una red usando Modbus ASCII (Código Estándar Americano para Intercambio de Información), cada ocho bits (1 byte) en un mensaje es enviado como dos caracteres ASCII. La principal ventaja de esta modalidad es que permite intervalos de tiempo de hasta un segundo que se produzca entre caracteres sin causar un error.

➤ **Bits por Byte**

1 bit de arranque.

7 bits de datos, el bit menos significativo se envía primero.

1 bit para paridad Par / Impar --- sin bits para no paridad.

1 bit de parada, si la paridad se utiliza---2 bits si no hay paridad.

➤ **Comprobación de errores**

Longitudinal de redundancia (LRC).

➤ **Trama ASCII**

En el modo ASCII, los mensajes comienzan con un signo de dos puntos (:) (3A hexadecimal), y termina con un retorno de carro-avance de línea (CR LF) (0D y 0A hexadecimal).

➤ **Sistema de Codificación**

Caracteres hexadecimales ASCII imprimibles 0-9, A-F.

La transmisión de caracteres permitidos para todos los demás campos es hexadecimal 0-9, A-F

Cuando un dispositivo recibe una trama, cada dispositivo decodifica el siguiente campo (el campo de la dirección) para averiguar si se trata de abordar el dispositivo Intervalos de hasta un segundo puede transcurrir entre caracteres en el mensaje. En caso de que un mayor intervalo se produce, el dispositivo receptor se da por supuesto se ha producido un error. Una típica trama de este modo se muestra en la tabla 1.7.

**Tabla 1.7 Descripción de una trama MODBUS en modo ASCII.**

Start	Address	Function	Data	LRC	End
:	2 Chars	2 Chars	N Chars	2 Chars	CR LF

### **1.5.3. Campo de dirección de la trama MODBUS**

El campo de la dirección de una trama o frase del mensaje contiene dos caracteres (ASCII) u ocho bits (RTU). Direcciones validas de dispositivos esclavos están en el rango de 0-247 decimal. Los dispositivos esclavos individuales se asignan las direcciones en el rango de 1-247. Un maestro direcciona a un esclavo al colocar la dirección en el campo de dirección del mensaje. Cuando el esclavo envía su respuesta, coloca su propia dirección en este campo de dirección de la respuesta para que el maestro reconozca cual esclavo responde.

Dirección 0 se usa para la dirección de broadcast, lo cual todos los dispositivos esclavo reconocen. Cuando el protocolo Modbus se utiliza en redes de nivel superior, los broadcast no podrán ser autorizados o podrán ser sustituidos por otros métodos. Por ejemplo, Modbus Plus utiliza una base de datos compartidos a nivel mundial que puede ser actualizado con cada modo de rotación.

### **1.5.4. campo de función de la trama MODBUS**

El campo función de una trama de mensaje contiene dos caracteres (ASCII) u ocho bits (RTU). Los códigos son válidos en el rango de 1-255 decimal. De estos, algunos códigos son aplicables a todos los controladores Modicon, mientras que algunos códigos se aplican sólo a ciertos modelos, y otros están reservados para uso futuro.

Cuando se envía un mensaje de un maestro a un dispositivo esclavo, el campo función le dice al esclavo qué tipo de acción que desea ejecutar. Ejemplos de ello son la lectura ON / OFF estados de un grupo de bobinas discretas o de insumos para leer el contenido de los datos de un grupo de registros; para leer el diagnóstico de la condición de esclavo; a escribir a determinados registros o bobinas, o para permitir la carga, Grabación, o verificar el programa en el esclavo.

Cuando el esclavo responde al maestro, utiliza el código de función para indicar o bien una respuesta normal (sin errores) o que la respuesta con algún tipo de error producido (llamado una respuesta de excepción). Para una respuesta normal, el esclavo, simplemente se hace eco de la función original de código. Para una respuesta de excepción, el esclavo devuelve un código que es equivalente a su código de función original con el bit más significativo seteado a lógica 1.

El dispositivo maestro tiene la responsabilidad del manejo de las respuestas de excepción.

En la tabla 1.8 se encuentran las funciones que están respaldadas por Modbus en dispositivos Modicom:

**Tabla 1.8 Funciones respaldadas por MODBUS en dispositivos Modicom.**

función	Descripción.
1	leer el estatus de bobina
2	leer condición de entrada
3	leer los registros de las explotaciones
4	leer registros de entrada
5	escribir bobina simple
6	escribir registro único
15	escritura de rollos múltiples
16	escribir múltiples registros

### 1.5.5. Campo de datos de la trama MODBUS

El campo de datos se construye con conjuntos de dos dígitos hexadecimales, en el rango de 00 a FF hexadecimal. Estos se pueden realizar desde un par de caracteres ASCII, o de un carácter RTU, de acuerdo a la red del modo de transmisión en serie.

El campo de datos de mensajes enviados desde un maestro a esclavo contiene información adicional que el esclavo debe utilizar para tomar la acción definida por el código de función. Esto puede incluir temas como discretos y direcciones de registro, la cantidad de temas que se manejan, y el número de bytes de datos reales sobre el campo.

Por ejemplo, si el maestro pide un esclavo a leer un grupo de los registros de las explotaciones (código de función 03), el campo de datos especifica el registro de inicio y cuántos registros se van a leer. Si el maestro escribe a un grupo de registros en el esclavo (función código 10 hexadecimal), el campo de datos especifica el registro de inicio, el número de registros a escribir, el número de bytes de datos a seguir en el campo de datos, y los datos que se escrito en los registros.

Si no se produce error, el campo de datos da una respuesta de un esclavo a un maestro que contiene los datos solicitados. Si se produce un error, el campo contiene un código de excepción que el maestro puede utilizar para determinar las próximas medidas que deben adoptarse.

El campo de datos puede ser inexistente (de longitud cero) en ciertos tipos de mensajes. Por ejemplo, en una solicitud de un dispositivo maestro un esclavo para responder a sus comunicaciones con el registro de sucesos (función 0B código hexadecimal), el esclavo no requiere ninguna información adicional. La función solo se especifica código de la acción.

### 1.5.6. Campo de error de la trama MODBUS

Dos tipos de error métodos de control se utilizan para redes estándar Modbus. La comprobación de errores del contenido de los campos dependerá del método que se utiliza.

#### ➤ ASCII

Cuando el modo ASCII se utiliza para la elaboración de carácter, el campo de control de error es de dos caracteres ASCII. La verificación de error de caracteres son el resultado de una redundancia longitudinal (LRC) que el cálculo se realiza sobre el mensaje contenido, excluidas los caracteres del comienzo y terminación de la trama CRLF.

Los caracteres LRC se adjuntan al mensaje como el último campo que precede a los caracteres CRLF (fin de trama).

#### ➤ RTU

Cuando el modo de RTU se utiliza para la elaboración de carácter, el error de control contiene un campo de 16-bit de valor aplicado como dos de ocho bits bytes. El error de verificación valor es el resultado de un cálculo de redundancia cíclica (CRC) realizado sobre el contenido de los mensajes.

El campo CRC se adjunta al mensaje como el último campo en el mensaje. Cuando esto se hace, el byte-bajo sobre el campo se incluye en primer lugar, seguido por el byte de alto orden. El byte de alto orden del CRC es el último byte que se enviará en el mensaje.

## 1.6. Telefonía celular.

La comunicación inalámbrica más utilizada en los últimos tiempos es la telefonía celular, que básicamente está formada por dos grandes partes: una red de comunicaciones (o red de telefonía móvil) y los terminales (o teléfonos móviles) que permiten el acceso a dicha red.

El **teléfono celular o móvil** es un dispositivo inalámbrico electrónico que permite tener acceso a la red de telefonía celular o móvil. Se denomina celular debido a las antenas repetidoras que conforman la red, cada una de las cuales es una célula, si bien existen redes telefónicas móviles satelitales. Su principal característica es su portabilidad, que permite comunicarse desde casi cualquier lugar. Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado otras funciones como son cámara fotográfica, agenda, acceso a Internet, envío de mensajes de texto, reproducción de video e incluso GPS.

El primer antecedente respecto al teléfono celular es de la compañía Motorola, con su modelo DynaTAC 8000X. El modelo fue diseñado por el ingeniero de Motorola Rudy Krolopp en 1983. El modelo pesaba poco menos que un kilo y un valor de casi 4.000 dólares. Krolopp se incorporaría posteriormente al equipo de investigación y desarrollo de Motorola liderado por Martin Cooper. Tanto Cooper como Krolopp aparecen como propietarios de la patente original.

A partir del DynaTAC 8000X, Motorola desarrollaría nuevos modelos como el Motorola MicroTAC, lanzado en 1989, y el Motorola StarTAC, lanzado en 1996 al mercado.

### 1.6.1. SMS

El servicio de mensajes cortos o **SMS** (*Short Message Service*) es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos (también conocidos como mensajes de texto, o más coloquialmente, textos o mensajitos) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano. SMS fue diseñado originariamente como parte del estándar de telefonía móvil digital GSM en 1985 como medio de envío de mensajes de hasta 160 caracteres (espacios incluidos), desde y hacia teléfonos móviles GSM. Desde entonces, el apoyo para el servicio se ha ampliado para incluir otros móviles, como las normas ANSI redes CDMA y AMPS digital, así como por satélite y redes terrestres. La mayoría de los mensajes SMS son mensajes de texto de móvil a móvil, aunque el estándar es compatible con otros tipos de emisión de mensajes, pero en la actualidad está disponible en una amplia variedad de redes, incluyendo las redes 3G.

Debido a que los mensajes SMS son recibidos prácticamente de inmediato por el destinatario y son un medio de comunicación muy personal, muchos ya los están utilizando como el mejor medio para comunicarse con una comunidad para invitar a eventos, dar avisos, enviar alarmas, coordinar evacuaciones, confirmar transacciones bancarias, enviar confirmaciones de compra y muchas cosas más.

Además de convertirse en una herramienta de comunicación cada vez más popular, se debe de pensar que empieza a utilizarse para la recepción de mensajes de alerta (alarma) de sistemas y para la emisión de mensajes de control hacia dispositivos (domótica), algo que era de esperar después de pensar que el teléfono móvil es una herramienta cada vez más extendida.

Para la correcta gestión de los mensajes SMS se hace necesario introducir en el sistema GSM un nuevo elemento: el **centro de mensajes cortos** o **SMSC** (*Short Message Service Center*). Las funciones del SMSC son:

- Recibir y almacenar los mensajes cortos enviados por los usuarios o por otras fuentes (avisos del operador, buzón de voz, sistemas de publicidad, alertas de correo electrónico...) hasta que puedan ser enviados;
- Verificar los permisos del usuario para enviar mensajes, en comunicación con el VLR de origen;
- Verificar si el usuario al que se envía el mensaje está operativo o no, mediante consulta al HLR de destino; si está operativo, el mensaje se envía, y si no se almacena temporalmente en el SMSC;
- Verificar periódicamente el estado de los usuarios que tienen mensajes pendientes.

Cuando un usuario envía un SMS, o lo recibe, se incluyen con su payload (carga útil o cuerpo del mensaje) al menos los siguientes parámetros:

- Fecha de envío (también llamada *timestamp*);
- Validez del mensaje, desde una hora hasta una semana;
- Número de teléfono del remitente y del destinatario;
- Número del SMSC que ha originado el mensaje;

De este modo se asegura el correcto procesamiento del mensaje en el SMSC y a lo largo de toda la cadena.

Los mensajes cortos hacen un uso extremadamente eficaz de la red de radio, y además pueden ser enviados y recibidos en cualquier momento, incluso durante una llamada. La explicación es que, debido a su pequeño tamaño, los SMS no necesitan que se asigne un canal de radio al usuario, como ocurre durante una llamada, sino que se insertan en la información de señalización de la propia red, en los time slots reservados para este fin.

Algunos operadores han implementado el transporte de los mensajes SMS a través del protocolo de paquetes GPRS en lugar del canal de señalización, incrementando la velocidad de transmisión y la capacidad del sistema, pero este cambio opcional en el transporte no se encuentra muy extendido.

Cuando un usuario de la red genera un mensaje corto (MO-SM *Mobile Originated*, originados en el terminal del usuario), se producen los siguientes acontecimientos:

- El VLR donde está registrado el usuario decide si puede o no enviar mensajes; si todo está en orden,
- El MSC al que está conectado el usuario recibe el mensaje, envía la información necesaria al VLR para su posterior tarificación y después lo remite al SMSC de origen;
- El SMSC de origen envía el mensaje al SMSC de destino (etiquetado SME). Una vez allí, se convierte y se procesa como a continuación veremos.
- El SMSC de destino informa del estado del mensaje y devuelve un informe de recepción al MSC y al usuario. En la pantalla del usuario se advierte: “mensaje enviado”.
- Si el usuario lo ha solicitado, recibirá posteriormente un mensaje de estado confirmándole si el usuario de destino ha recibido el mensaje o no, y un mensaje de error en caso de que caduque.

Por tanto, el método de envío de los SMS tiene un pequeño defecto: los mensajes se tarifican y confirman inicialmente al usuario **cuando son enviados a la red, no al destino final**, incluso aunque el cliente que los envía haya solicitado confirmación de envío. Un mensaje podría no llegar por problemas en la red destino, caducidad de la validez o cualquier otro motivo, pero sin embargo ser cobrado igualmente por el operador.

Una vez hay un mensaje preparado para su envío en el SMSC de destino, originado por un usuario o por cualquier otra circunstancia, se le denomina MT-SM (*Mobile Terminated-Short Message*, es decir, mensajes que llegan al terminal del usuario) y el proceso de entrega es el siguiente:

- El SMSC que ha recibido el mensaje lo almacena en su base de datos y solicita al HLR (registro de localización de datos de las terminales móviles) del usuario la información de localización;
- Si el usuario destino está disponible, el SMSC envía al MSC el mensaje, indicando en que parte del BSS debe ser entregado; si no lo está, se almacena en el SMSC durante su periodo de vigencia;
- Si el usuario destino está disponible, el MSC envía un aviso al VLR (registro de localización de visitante: indica la zona en la que se encuentra registrado el terminal móvil destino) al que está conectado el usuario destino (que puede ser o no de su operador) para indicarle que va a entregarse un mensaje;
- El VLR avisa al terminal del usuario y verifica si está conectado a la red (en zona de cobertura);
- El VLR responde al SMSC con el estado del usuario y, si está operativo, con la información de localización (parte del BSS en que se encuentra conectado);
- El MSC envía el mensaje al usuario;
- El MSC informa al SMSC de que el mensaje se ha entregado y puede ser borrado de su base de datos;
- Opcionalmente, el SMSC de destino responde a quien originó el mensaje (normalmente, el SMSC origen) con un aviso de entrega del mensaje.

## 1.6.2. Protocolo GSM para SMS

El *Short Message Service - Punto a Punto (SMS-PP)* se define en la recomendación GSM 03.40. GSM 03.41 define el *servicio de mensajes cortos - de difusión celular (SMS-CB)*, que permite que los mensajes (publicidad, información pública, etc.) para ser difundidos a todos los usuarios de móviles en una zona geográfica concreta. Los mensajes se envían a un Centro Servicio de Mensajes Cortos (SMSC), que proporciona un almacenamiento y mecanismo de traspaso de fondos. Se trata de enviar mensajes a sus destinatarios, si un destinatario no está accesible, el SMSC encolas el mensaje para que vuelva a intentar enviarlo más tarde. SMSCs también provee una opción "adelante y olvidar" donde la transmisión es enviado sólo una vez. La entrega de mensajes es el mejor esfuerzo, por lo que no hay garantías de que un mensaje realmente sea entregado a su destinatario y el retraso o pérdida completa de un mensaje es frecuente, sobre todo cuando el envío es entre las redes. Los usuarios pueden optar por solicitar los informes de entrega (simplemente añade 0 # \* o \* N # al principio de su mensaje de texto).

La transmisión de mensajes cortos entre el SMSC y el destinatario se realiza a través de la parte aplicación móvil (MAP) del protocolo SS7. Los mensajes se envían con el MAP y operaciones mo- y mt-ForwardSM, cuya longitud de carga útil está limitada por las limitaciones del protocolo de señalización precisamente a 140 octetos (140 octetos = 140 \* 8 bits = 1120 bits). Mensajes se pueden codificar utilizando una variedad de alfabetos: el valor por defecto GSM 7 – alfabeto de bits (que se muestra a continuación), los 8-bits de alfabeto de datos, y de alfabeto de 16-bit UTF-16/UCS-2. Dependiendo de qué alfabeto, ha configurado el abonado en el terminal, esto lleva al máximo tamaño individual de mensajes cortos de 160 caracteres de 7 bits, 140 de 8 bits de caracteres, o 70 de 16 bits caracteres (incluidos espacios) El soporte del alfabeto GSM de 7 bits

es obligatorio para teléfonos GSM y elementos de la red, pero caracteres en idiomas como el árabe, chino, coreano, japonés o alfabeto cirílico (Rusia, por ejemplo) deben ser codificados utilizando el de 16-bit UCS -2. Enrutamiento de datos y otros metadatos es adicional al tamaño de la carga útil.

### ➤ **Tamaño del mensaje**

Un mayor contenido (concatenado SMS, varios SMS o segmentado "tamaño sms") puede ser enviado a través de varios mensajes, en cuyo caso cada mensaje se iniciará con una cabecera de datos de usuario (UDH), que contiene información de segmentación. Cuando UDH está dentro de la carga útil, el número de caracteres por segmento es menor: 153 de 7-bits de codificación, 134 de 8-bits de codificación y 67 para 16-bits de codificación. El teléfono que recibe los SMS es responsable de re ensamblar los mensajes y presentar al usuario como un mensaje largo. Si bien en teoría, la norma permite hasta 255 segmentos, en la práctica es de 6 al 8 de segmento como máximo, y los mensajes largos se facturarán a menudo como equivalente a múltiples mensajes SMS.

### ➤ **Proveedores Gateway SMS**

Gateway SMS facilitan el tráfico de SMS entre las empresas y los abonados a servicios móviles, siendo el principal responsable de la ejecución de misión crítica los mensajes SMS para las empresas, entrega de contenido y servicios de entretenimiento con SMS. Teniendo en cuenta el rendimiento de mensajería SMS y los costes, así como el nivel de los servicios de mensajería, Gateway SMS pueden clasificarse como agregadores o proveedores SS7.

El modelo agregador se basa en acuerdos con varias telefonías móviles para el intercambio de 2 forma de tráfico de SMS de entrada y salida de la plataforma del operador de SMS (SMSC), también conocido como **modelo de terminación local**. El modelo de Agregadores falta de acceso directo en el protocolo SS7, que es el protocolo que el **SMS** se intercambian mensajes. Los mensajes SMS son entregados en el operador de SMSC, pero no al teléfono del abonado, el SMSC se ocupa de su posterior manipulación del mensaje a través de la red SS7.

Otro tipo de proveedor de Gateway SMS se basa en la conectividad SS7, para enrutar los mensajes SMS, también conocidos como **modelo de terminación internacional**. La ventaja de este modelo es la capacidad para enrutar los datos directamente a través de SS7, lo que da al proveedor es control total y la visibilidad de la ruta completa durante el enrutamiento de SMS. Esto significa que los mensajes SMS pueden ser enviados directamente a los beneficiarios y sin tener que pasar por el SMSC de otros operadores móviles. Por lo tanto, es posible evitar demoras y pérdidas mensaje, que ofrece plena garantía de entrega de los mensajes y el enrutamiento optimizado. Este modelo es especialmente eficaz cuando se utiliza en misión crítica y de mensajes SMS utilizadas en las comunicaciones corporativas.

## 1.7. Comandos AT

La comunicación entre un teléfono celular y un dispositivo cercano se realiza por medio de comandos AT o comandos Hayes.

AT (attention command), son un conjunto de comandos necesarios para el control del modem integrado del teléfono. Ya sea para control infrarrojo, inalámbrico o por cable.

Los Comandos AT pueden ser usados a grandes rasgos para ajustar propiedades avanzadas del teléfono:

- Configurar el teléfono para una conexión inalámbrica, a través de infrarrojos o por el sistema de bus o cable.
- Configurar el modem interno del teléfono para una conexión inalámbrica, a través de infrarrojos o por el sistema de bus o cable.
- Solicitar información sobre la configuración actual o estado operativo del teléfono o modem.
- Probar la disponibilidad del teléfono o modem.
- Envío de mensajes.

El conjunto de órdenes Hayes puede subdividirse en cuatro grupos:

- conjunto básico de comandos - Un carácter de capital seguido de un dígito. Por ejemplo, M1.
- comandos extendidos - Una "&" (símbolo de unión) y un carácter de capital seguida de un dígito. Así se amplía la base de comandos. Por ejemplo, &M1. Tenga en cuenta que M1 es diferente de la &M1.
- comando propietarios - Por lo general, empezando con una barra invertida ("\") o con un signo de porcentaje ("%"); estos comandos varían ampliamente entre los fabricantes de módem.
- Comandos de registro - Sr = n donde r es el número del registro a ser cambiado, y n es el nuevo valor que se asigna.

Un registro representa una ubicación física en la memoria. Módems tienen pequeñas cantidades de memoria a bordo. El cuarto grupo de comandos sirve para introducir valores en un registro especial (de memoria). El registro de un almacén particular variable (alfa-numérico de la información) que el módem y el software de comunicaciones pueden utilizar. Por ejemplo, S7 = 60 instruye a la computadora para "Configurar el registro # 7 para el valor 60".

Aunque la sintaxis del conjunto de comando define la mayoría de los comandos por medio de una combinación de números y letras (L0, L1 etc.), el uso de un cero es opcional. En este ejemplo, "L0" equivale a una simple "L".

Algunos de los más importantes caracteres que pueden aparecer en la cadena de inicialización del módem son los mostrados a continuación. Normalmente uno no debe cambiar estos caracteres.

- AT "Atención": dice al módem comandos-módem seguirán. Este debe comenzar en cada línea de comandos.
- Z - Restablece el módem a su estado predeterminado.
- , (Una coma) - Hace que el software de una pausa por un segundo. Más de una coma puede aparecer en la sucesión. Por ejemplo, ",,,," dice al software para hacer una pausa en cuatro segundos. (La configuración de registro S8 regula la duración de la pausa.).
- ^ M (Control-M) - Envía la terminating Carriage Return al módem. Se trata de un código de control que la mayoría de software de comunicación se traduce en un retorno de carro. (Nota, cuando este se envía al módem, es enviado como un solo byte, ASCII CR (0x0D), o "Control-M" no los dos caracteres ^ y M.).
- ; (Un punto y coma) - Regreso a modo de comando inmediatamente después de marcar. Esto hace que sea posible, por ejemplo, para marcar más de 45 dígitos de los números, o navegar a través de menús interactivos.

Cuando se está en modo de datos, una secuencia de escape puede devolver el módem al modo de comando. La secuencia de escape normal es de tres signos más ("+++").

## Definiciones de sintaxis

Las siguientes definiciones aplican la sintaxis:

- <CR> carácter de retorno, es la línea de comandos y el resultado de código de caracteres de terminación, valor que, en ASCII decimal entre 0 y 255, se especifica en el parámetro S3. El valor por defecto es 13.
- <LF> carácter salto de línea, el carácter es reconocido como línea de alimentación de caracteres. Su valor, en ASCII decimal entre 0 y 255, se especifica en el parámetro S4. El valor por defecto es 10. La línea de alimentación de salida es de carácter retorno de carro después de carácter verbose, si se utilizan códigos (V1 opción utilizada) de otro modo, si se utilizan el formato numérico de códigos de resultado (V0 opción utilizada), no aparecerán en los códigos de resultado.
- <...> encerrado en paréntesis angulares es un elemento de sintáctica. Ellos no aparecen en la línea de comandos.
- [...] Subparámetro opcional de un comando o una parte opcional de la respuesta es la información entre corchetes. corchetes en sí mismos no aparecen en la línea de comandos. Cuando no se da Subparametros en los comandos AT que tienen un comando "read", el nuevo valor es igual a su valor anterior. En comandos AT que no almacenan los valores de cualquiera de sus subparametros, por lo que no tienen un comando "read", que se llaman comandos tipo de acción, la acción debe hacerse sobre la base de la configuración recomendada por defecto de los subparametros.

### 1.7.1. V.250

La UIT-T estableció un estándar en sus recomendaciones Series-V, V.25 ter, en 1995 en un intento de establecer un estándar para el set de comando. Se rebautizó V.250 en 1998 con un anexo que no era relativa al conjunto de comandos Hayes rebautizado como V.251. Un módem compatible con V.250 implementa la A, D, E, H, I, L, M, N, O, P, Q, T, V, X, Z, y C, y D, y F y los comandos en la forma especificada por la norma. También debe aplicar registros S y debe utilizar los registros S0, S3, S4, S5, S6, S7, S8 y S10 para los fines que figuran en la norma. Por último también se debe aplicar cualquier comando que comienza con el signo más "+" seguido de cualquier letra "A" a la "Z", sólo en conformidad con recomendaciones de la UIT. Los fabricantes de Módem son libres de ejecutar otros comandos y los registros de S si lo consideran conveniente, y puede añadir opciones a los comandos estándar.

**Tabla1.9 Definición principales de secuencias de caracteres para comandos AT**

Caracteres principales	Incluye comandos relacionados
+A	Control de llamada (direccionamiento de red), comunes, PSTN, ISDN, ITU-T Rec. X.25, conmutación digital
+C	Extensión de celulares digitales
+D	Compresión de datos
+E	Control de error
+F	Faxes
+G	cuestiones genéricas como la identidad y la capacidad
+I	Interfaces
+M	modulación
+P	Comandos PCM DCE
+S	conmutación o simultánea tipos de datos
+T	cuestiones de prueba
+V	Extensiones de voz
+W	Extensiones Wireless

La ETSI GSM 07.07 (3GPP TS 27.007) especifica en el estilo de comandos para el control de un teléfono GSM o módem.

Casi todos los comandos extendidos tienen un comando de prueba "=?" para probar la existencia de comando y para dar información de los subparámetros. Casi todos los comandos tienen un comando de lectura "?" para verificar los valores actuales de los subparámetros. Los comandos de tipo de acción no guardan los valores de cualquiera de su posible subparámetros, y por consiguiente ellos no tienen una orden lectura.

Las respuestas son habilitadas con el comando ATV1; si el comando se ha efectuado con éxito, el código de resultado <CR><LF>OK <CR><LF> se envía al ordenador o equipo Terminal. Si se habilitan contestaciones numéricas con el comando ATV0, el código de resultado 0 <CR> es el que se envía. Si el comando no se efectúa correctamente se devuelve el código ERROR.

Hay dos maneras de enviar y recibir mensajes SMS: modo de texto y de PDU (unidad de descripción de protocolo) modo de espera. El modo texto (no disponible en algunos teléfonos) no es más que una codificación del flujo de bits representado por el modo PDU. Pueden ser diferentes alfabetos y existen varias alternativas de codificación cuando se muestra un mensaje SMS. Las opciones más comunes son "PCCP437", "PCDN", "8859-1", "IRA" y "GSM". Todos estos son fijados por el comando at "AT + CAC", cuando haya leído el mensaje en una aplicación informática. Si lee el mensaje en su teléfono, el teléfono escoge una correcta codificación. Una aplicación capaz de leer los mensajes SMS, por lo tanto, puede utilizar en modo texto o modo PDU. Si se utiliza el modo texto, la aplicación se ve obligada a (o limitado por) el conjunto de opciones de codificación preestablecido. Si se utiliza el modo PDU, cualquier codificación puede ser implementada.

Un ejemplo de la forma en que se utilizan los comandos es la mostrada en el ejemplo siguiente donde se describe el comando y como el modem responde a ese comando.

```
AT+CGMI<CR>           ← línea entera de un comando
Nokia                 ← información de respuesta
OK                    ← código de resultado final
```

### 1.7.2. Descripción de comandos AT en modo texto

Algunos comandos utilizados son para el seteo del teléfono celular, de los cuales se tiene:

#### FORMATO DEL SMS (+CMGF)

Parámetros del comando:

**Tabla 1.10 Descripción de los campos de la frase NMEA GGA Parámetros del comando +CMGF.**

COMANDO	POSIBLE RESPUESTA
+CMGF=[<CODIGO>]	
+CMGF?	+CMGF: <MODO>
+CMGF=?	+CMGF: LISTA DE <MODOS> SOPORTADOS

Descripción:

Comando que indica al equipo cual es el formato de entrada y salida a usar en los mensajes.

<modo> indica el formato de los mensajes utilizados cuando se envía, lista, lee y escribe comandos y códigos de resultado no solicitadas como consecuencia de los mensajes recibidos. El comando retorna los valores soportados como un valor compuesto.

<mode>:

- 0 PDU modo (implementado por defecto)
- 1 modo texto

### CÓDIGO DE RESULTADO DE FRACASOS (+CMS)

Código de resultado final +CMS error: <err>. Indica un error relacionado con los equipos móviles o de red. El funcionamiento es similar al código de resultado ERROR. Ninguno de los siguientes comandos en la misma línea de comando es ejecutado. Ni bien ni ERROR resultado código será devuelto. Error se produce normalmente cuando el error se relaciona con la sintaxis o parámetros no válidos. la tabla 1.11 Da a conocer los diferentes tipos de error que se pueden encontrar al estar utilizando los comandos AT.

**Tabla 1.11 Diferentes tipos de error en los comandos AT.**

<err>	valores comunes utilizados por los comandos de mensajería
0-127	GSM 04.11
128-255	Valores GSM 03.40 subclausula 9.2.3.22
300	Error en equipo móvil(ME)
301	
302	Operación no permitida
303	Operación no soportada
304	Parámetro no valido modo PDU
305	Parámetro no valido modo TEXTO
310	SIM no insertada
311	PIN de SIM necesario
312	PIN PH-SIM necesario
313	Error en SIM
314	SIM ocupado
315	SIM equivocado
316	PUK de SIN necesario
317	PIN2 de SIM necesario
318	PUK2 de SIM necesario
320	Fallo de memoria
321	Índice de memoria invalido

322	Memoria llena
330	Dirección desconocida de SMSC
331	No hay servicio de red
332	Fuera de tiempo en red
340	No espera reconocimiento +CNMA
500	Error desconocido
...511	Otros valores en rango de 256-511 son reservados
512....	Específicos del fabricante

## INDICACIÓN DE NUEVOS MENSAJES (+CNMI)

Parámetros del comando AT+CNMI son mostrados en la tabla 1.12:

**Tabla 1.12** Parámetros del comando +CNMI.

COMANDO	POSIBLE RESPUESTA
+CNMI=[<mode>[,<mt>[,<bm>[,<ds>[,<bfr>]]]]]	
+CNMI?	+CNMI: <mode>,<mt>,<bm>,<ds>,<bfr>
+CNMI=?	+CNMI: (lista de <mode> soportados) , (lista de <mt> soportados) , (lista <bm> soportados) , (lista de <ds> soportados) , (lista de <bfr> soportados)

### Descripción:

Comando que selecciona el procedimiento, cómo la recepción de nuevos mensajes de la red cuando el equipo terminal (TE siglas en ingles) está activo, por ejemplo, DTR señal está en ON. TE si está inactivo (por ejemplo, es señal DTR OFF), mensaje que reciben se debe hacer tal como se especifica en GSM 03.38.

NOTA: Cuando la señal DTR no está disponible o el estado de la señal se ignora (V.25ter comando & D0) y es fiable de transferencia de mensajes se puede garantizar mediante el uso de + CNMA que es el procedimiento de reconocimiento.

<mode> controla los procesos de códigos de resultados no solicitados dentro de este comando,  
 <mt> establece la indicación del código de resultado de enrutamiento para el envío de SMS y  
 <ds> para reporte de estatus de SMS.  
 <bfr> define el método para la manipulación de resultado de buffer cuando <modo> códigos 1, 2 o 3 está habilitado.

## ENVIO DE SMS (+CMGS)

Parámetros del comando AT+CMGS son mostrados en la tabla 1.13:

Tabla 1.13 Parámetros del comando +CMGS.

COMANDO	POSIBLE RESPUESTA
+CMGS=<da>[, <toda>]<CR> <b>TEXTO A ENVIAR</b> <ctrl-Z/ESC>	+CMGS: <mr>[, <scts>] <b>Si el envío falla:</b> +CMS ERROR: <err>

Descripción:

Ejecutar el comando envía un mensaje de TE a la red. El valor de referencia del mensaje <mr> es devuelto al TE cuando el mensaje es entregado. Si falla el envío en una red o un error en ME (equipo móvil por sus siglas en ingles), el código de resultado final +CMS ERROR: <err> es devuelto. Este comando debe ser abordable.

<da> GSM 03.40 campo de dirección destino o valor de dirección en formato string; número BCD (o caracteres por defecto GSM)

## 1.8. Demonio GAMMU y GNOKY

Gammu es un software de libre distribución y multiplataforma, que puede manejar muchas funciones en teléfonos celulares y ofrece funciones no disponibles en otros proyectos similares; esta aplicación fue llamada anteriormente como Gnokii. Esta herramienta de línea de comandos nos permite administrar una amplia selección de marcas y modelos de teléfonos móviles, sin importar la forma de enlace de comunicación: cable de datos como serial (RS-232) o USB, infrarrojo y Bluetooth. Existe una herramienta de software que proporciona una interfaz gráfica intuitiva que hace el trabajo más fácil a la hora de usar Gammu, llamada Wammu; solamente que dicha herramienta carece hasta este momento de muchas de las inmensas funcionalidades que posee Gammu. El antecesor de esta herramienta de software, se llama Gnokii, el cual se puede seguir utilizando y con muchas de las funcionalidades que ahora posee Gammu. Gnokii se trata de un proyecto liberado bajo licencia GNU GPL 2, inicialmente desarrollado como reemplazo para el software original de Nokia llamado “Nokia Cellular Data Suite”. Se inicio el 21 de octubre de 1998 por Hugh Blemings y creado para el Nokia 3110/8110. Históricamente surgió el Gammu posteriormente.

### 1.8.1. GAMMU

Gammu permite acceder al sistema de ficheros del celular y funciones especiales de control, como radios y cámaras integradas. Incluso se puede realizar pruebas para verificar la alarma vibratoria y la luz del teclado o la pantalla. Gammu puede instalarse en sistemas operativos tales como Windows, Linux, Mac OS, etc. Este software fue más temprano (hasta la versión 0.58)

llamado MyGnokii2, actualmente llamado Gammu no tiene ninguna relación con Gammu de “Heretics of Dune” escrito por Frank Herbert. El programa Gammu puede trabajar con los contactos, mensajes (SMS, EMS y MMS), calendario, listas de tareas, sistema de archivos, radio, cámara, etc. Este también soporta el modo de servicio para enviar y recibir solamente mensajes SMS. Gammu por si mismo funciona en línea de comandos o como un demonio.

## **Características**

Un resumen de las características soportadas por el software Gammu, son las siguientes:

1. Envío y recepción de mensajes SMS.
2. Agenda telefónica.
3. Marcado y recepción de llamadas.
4. Calendario.
5. Notas.
6. Manipulación del Sistema de Ficheros.

## **Teléfonos soportados**

1. Muchos modelos de teléfonos Nokia.
2. Teléfonos Alcatel BE5 (501/701), BF5 (715), BH4 (535/735).
3. Teléfonos con capacidades de comandos AT (Siemens, Nokia, Alcatel, Motorola, Sony-Ericsson, IPAQ).
4. Teléfonos Symbian mediante gnapplet.

Para una lista más detallada de las marcas, modelos, tipos de conexión y funcionalidades soportadas por los teléfonos móviles, se puede consultar en el anexo o en la página de internet, <http://cihar.com/gammu/phonedb/>

### **1.8.2. GAMMU como servicio.**

Gammu tiene la habilidad de enviar y recibir mensajes SMS en modo servicio. Para ejecutarlo se necesita ajustar la configuración en un archivo especial para dicha función, además se necesita tener instalado previamente el paquete servidor de MySQL, crear la base de datos y las tablas respectivas, y finalmente ejecutar Gammu en línea de comandos con los parámetros adecuados.

Existen dos modos en los que puede trabajar el servicio, el primero es con el parámetro FILES, en el cual se almacenan los mensajes como ficheros de texto en el disco de almacenamiento; y el segundo es con el parámetro MYSQL donde los SMS son almacenados en una base de datos MySQL o en su defecto en cualquier otra base de datos como POSTGRE.

Este servicio se puede iniciar manualmente desde una ventana de consola Linux y esperar, sin embargo las instrucciones contenidas smsdrc que explican como cambiar inittab en Ubuntu no son funcionales, por lo que se puede iniciar Gammu en event.d así:

Crear un fichero guión de instrucciones de inicio de Gammu y se ubica en la carpeta /usr/local/bin

```
# cd /usr/local/bin
# vi gammu.sh
```

El contenido del fichero es como el siguiente:

```
#!/bin/sh
export LANG=es_ES
/usr/local/bin/gammu --smsd MYSQL /etc/smsdrc
```

Crear el fichero guión de instrucciones en la carpeta /etc/event.d/

```
root@laptop:/etc/event.d# nano gammu1
```

Complete las siguientes líneas en este archivo:

```
start on runlevel 2
start on runlevel 3
start on runlevel 4
start on runlevel 5
respawn
exec gammu.sh
```

Guarde los cambios antes de salir. Inicie el servicio Gammu:

```
root@laptop:/etc/event.d# start gammu1
gammu1 (start) waiting
gammu1 (start) starting
gammu1 (start) pre-start
gammu1 (start) spawned, process 11626
gammu1 (start) post-start, (main) process 11626
gammu1 (start) running, process 11626
```

Ver el estado de Gammu:

```
root@laptop:/etc/event.d# status gammu1
gammu1 (start) running, process 11626
```

Parar el proceso gammu:

```
root@laptop:/etc/event.d# stop gammu1
gammu1 (stop) running, process 11626
gammu1 (stop) pre-stop, (main) process 11626
gammu1 (stop) stopping, process 11626
gammu1 (stop) killed, process 11626
gammu1 (stop) post-stop
gammu1 (stop) waiting
```

Ver si Gammu se está ejecutando:

```
root@laptop:/etc/event.d# ps ax | grep gammu
11820?    Ss    0:00/usr/local/bin/gammu --smsd MYSQL /etc/smsdrc
11931 pts/0  R+   0:00 grep gammu
```

Adicionalmente se puede hacer que el servicio de Gammu con el Sistema Operativo Ubuntu, añadiendo el comando “/usr/bin/gammu POSTGRESQL /etc/smsdrc” al fichero “/etc/rc.local” por delante de “exit 0”.

## 1.9. Bases de datos.

Base de datos se define como un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Análogamente, una biblioteca puede considerarse una base de datos compuesta generalmente por documentos y textos impresos en papel e indexados para su posterior consulta. Actualmente, debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos se encuentran en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Existen programas llamados sistemas gestores de bases de datos, abreviado SGBD, que permite almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGDB, así como su utilización y administración, se estudian dentro del ámbito de la informática. Las aplicaciones más utilizadas son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

### 1.9.1. Características.

Entre las principales características de los sistemas de base de datos se puede mencionar:

1. Independencia lógica y física de los datos.
2. Redundancia mínima.
3. Acceso concurrente por parte de múltiples usuarios.
4. Integridad de datos.
5. Consultas complejas optimizadas.
6. Seguridad de acceso y auditoría.
7. Respaldo y recuperación.
8. Acceso a través de lenguajes de programación.

### 1.9.2. Tipos de bases de datos.

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación:

#### Según la variabilidad de los datos almacenados

##### **Bases de datos estáticas**

Estas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

##### **Bases de datos dinámicas**

Estas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, etc.

#### Según el contenido

##### **Bases de datos bibliográficas**

Solo contiene una representación de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no se estaría en presencia de una base de datos a texto completo (o de fuentes primarias). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio.

##### **Bases de datos de texto completo**

Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

## **Directorio**

Un ejemplo son las guías telefónicas en formato electrónico.

## **Bases de datos de información Biológica**

Son base de datos que almacenan diferentes tipos de información proveniente de las ciencias de la vida o médicas. Se pueden considerar en varios subtipos.

### *Modelos de bases de datos*

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

### **Bases de datos jerárquicas**

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

### **Base de datos de red**

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico). Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

### **Base de datos relacional**

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de

los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

### **Bases de datos multidimensionales**

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de Cubos OLAP. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos multidimensional podría serlo también en una base de datos relacional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.

### **Bases de datos orientadas a objetos**

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signature) de una operación incluye el nombre de la operación y los

tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

### **Bases de datos documentales**

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Taurus es un sistema de índices optimizado para este tipo de bases de datos.

### **Base de datos deductivas**

Un sistema de base de datos deductivos, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática.

### **Gestión de bases de datos distribuida**

La base de datos está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así a distintas universidades, sucursales de tiendas, etc.

## **1.9.3. POSTGIS.**

PostGIS: Es una extensión al sistema de base de datos objeto-relacional PostgreSQL. Permite el uso de objetos *GIS* (Geographic information systems). PostGIS incluye soporte para índices GiST basados en R-Tree, y funciones básicas para el análisis de objetos GIS.

Esta creado por Refrations Research Inc, como un proyecto de investigación de tecnologías de bases de datos espaciales. Está publicado bajo licencia GNU. Con PostGIS podemos usar todos los objetos que aparecen en la especificación OpenGIS como puntos, líneas, polígonos, multi-líneas, multipuntos, y colecciones geométricas.

Los objetos GIS soportados por PostGIS son de características simples definidas por OpenGIS. Actualmente PostGIS soporta las características y el API de representación de la especificación OpenGIS pero no tiene varios de los operadores de comparación y convolución de esta especificación.

Ejemplos de la representación en modo texto:

- POINT(0 0 0)
- LINESTRING(0 0,1 1,1 2)
- POLYGON((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0))
- MULTIPOINT(0 0 0,1 2 1)
- MULTILINESTRING((0 0 0,1 1 0,1 2 1),(2 3 1,3 2 1,5 4 1))

- MULTIPOLYGON(((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0)),((-1 -1 0,-1 -2 0,-2 -2 0,-2 -1 0,-1 -1 0)))
- GEOMETRYCOLLECTION(POINT(2 3 9),LINESTRING((2 3 4,3 4 5))

En los ejemplos se pueden ver características con coordenadas de 2D y 3D (ambas son permitidas por PostGIS). Podemos usar las funciones `force_2d()` y `force_3d()` para convertir una característica a 3d o 2d.

## 1.10. JAVA.

Java nació en 1991 cuando un conjunto de ingenieros de Sun Microsystems intentaron diseñar un novedoso lenguaje de programación destinado a electrodomésticos. La reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar código de tamaño reducido.

Debido a la existencia de distintos de Unidades Centrales de Procesos (UCP) y a los continuos cambios, era importante conseguir una herramienta independiente del tipo de UCP utilizada. Desarrollaron un código “neutro” que no dependía del tipo de electrodoméstico, el cual se ejecutaba sobre una “máquina hipotética o virtual” denominada Máquina Virtual Java (siglas en inglés JVM). Era la JVM quien interpretaba el código neutro convirtiéndolo a código particular de la UCP utilizada. Esto permitía lo que luego se ha convertido en el principal lema del lenguaje: “Escríbelo una vez, ejecútalo donde sea”. A pesar de los esfuerzos realizados por sus creadores, ninguna empresa de electrodomésticos se interesó por el nuevo lenguaje.

Como lenguaje de programación para ordenadores, Java se introdujo a finales de 1995. La clave fue la incorporación de un intérprete Java en la versión 2.0 del programa Netscape Navigator, produciendo una verdadera revolución en Internet. Java 1.1 apareció a principios de 1997, mejorando sustancialmente la primera versión del lenguaje. Java 1.2, más tarde rebautizado como Java 2, nació a finales de 1998.

Al programar en Java no se parte de cero. Cualquier aplicación que desarrolle se apoya, en un gran número de clases preexistentes. Algunas de ellas las ha podido hacer el propio usuario y otras pueden ser comerciales, pero siempre hay un número muy importante de clases que forman parte del propio lenguaje (el API o Application Programming Interface de Java). Java incorpora en el propio lenguaje muchos aspectos que en cualquier otro lenguaje son extensiones propiedad de empresas de software o fabricantes de ordenadores (hilos, ejecución remota, componentes, seguridad, acceso a bases de datos, etc.). Por eso muchos expertos opinan que Java es el lenguaje ideal para aprender la informática moderna, porque incorpora todos estos conceptos de forma estándar, mucho más sencillo y claro que con las mencionadas extensiones de otros lenguajes.

El principal objetivo del lenguaje Java es llegar a ser el “nexo universal” que conecte a los usuarios con la información, éste está ubicado en el ordenador local, en un servidor de WEB, en una base de datos o en cualquier otro lugar.

### 1.10.1.Hilos demonios (procesos).

Un proceso es un programa ejecutándose dentro de su propio espacio de direcciones. Java es un sistema multiproceso, esto significa que soporta varios procesos corriendo a la vez dentro de sus propios espacios de direcciones. Estamos más familiarizados con el término multitarea, el cual describe un escenario muy similar al multiproceso. Por ejemplo, consideremos la cantidad de aplicaciones que corren a la vez dentro de un mismo entorno gráfico. Mientras escribo esto, está corriendo Microsoft Word además de Internet Explorer, Windows Explorer, CD Player y el Volumen Control. Estas aplicaciones son todos procesos ejecutados dentro de Windows. De esta forma, se puede pensar que los procesos son análogos a las aplicaciones o a programas aislados, pero cada proceso tiene asignado espacio propio de ejecución dentro del sistema.

Un hilo es una secuencia de código en ejecución dentro del contexto de un proceso. Los hilos no pueden ejecutarse ellos solos; requieren la supervisión de un proceso padre para correr. Dentro de cada proceso hay varios hilos ejecutándose. Por ejemplo, Word puede tener un hilo en *background* chequeando automáticamente la gramática de lo que estoy escribiendo, mientras otro hilo puede estar salvando automáticamente los cambios del documento en el que estoy trabajando. Como Word, cada aplicación (proceso) puede correr varios hilos los cuales están realizando diferentes tareas. Esto significa que los hilos están siempre asociados con un proceso en particular.

Los hilos a menudo son conocidos o llamados procesos ligeros. Un hilo, en efecto, es muy similar a un proceso pero con la diferencia de que un hilo siempre corre dentro del contexto de otro programa. Por el contrario, los procesos mantienen su propio espacio de direcciones y entorno de operaciones. Los hilos dependen de un programa padre en lo que se refiere a recursos de ejecución.

Java es un lenguaje de programación que incorpora hilos en el corazón del mismo lenguaje. Comúnmente, los hilos son implementados a nivel de sistema, requiriendo una interfaz de programación específica separada del núcleo del lenguaje de programación.

Java se presenta como ambos, como lenguaje y como sistema de tiempo de ejecución (*runtime*), siendo posible integrar hilos dentro de ambos. El resultado final es que se pueden usar hilos Java como standard, en cualquier plataforma.

En Java, los hilos comparten el mismo espacio de memoria. Incluso comparten gran parte del entorno de ejecución, de modo que la creación de nuevos hilos es mucho más rápida que la creación de nuevos procesos. La ventaja que proporcionan los hilos es la capacidad de tener más de un camino de ejecución en un mismo programa. Así, con un único proceso, ejecutándose una JVM (*Java Virtual Machine*), habrá siempre más de un hilo, cada uno con su propio camino de ejecución. En cuanto al proceso de creación de hilos, son dos los mecanismos que nos permiten llevarlo a cabo en Java: implementando la interfaz Runnable, o extendiendo la clase Thread, esto es, creando una subclase de esta clase.

Un proceso demonio es un proceso que debe ejecutarse continuamente en modo *background* (en segundo plano), y generalmente se diseña para responder a peticiones de otros procesos a través de la red. La palabra “*daemon*” (proveniente de la palabra griega “*ghost*”) es propia de UNIX, pero no se utiliza de este mismo modo en Windows. En Windows, los demonios se denominan “servicios”. Cuando los servicios atienden peticiones, se conocen como la parte “Servidor” de una arquitectura Cliente/Servidor.

Los hilos demonio también se llaman servicios, porque se ejecutan, normalmente, con prioridad baja y proporcionan un servicio básico a un programa o programas cuando la actividad de la máquina es reducida. Un ejemplo de hilo demonio que está ejecutándose continuamente es el recolector de basura (*garbage collector*). Este hilo, proporcionado por la Máquina Virtual Java, comprueba las variables de los programas a las que no se accede nunca y libera estos recursos, devolviéndolos al sistema. Un hilo puede fijar su indicador de demonio pasando un valor “true” al método `setDaemon()`. Si se pasa “false” a este método, el hilo será devuelto por el sistema como un hilo de usuario. No obstante, esto último debe realizarse antes de que se arranque el hilo con el método `start()`.

## CAPITULO II

# DISEÑO E IMPLEMENTACION DE LA SOLUCION DEL SISTEMA CLIENTE

### 2.1. Introducción.

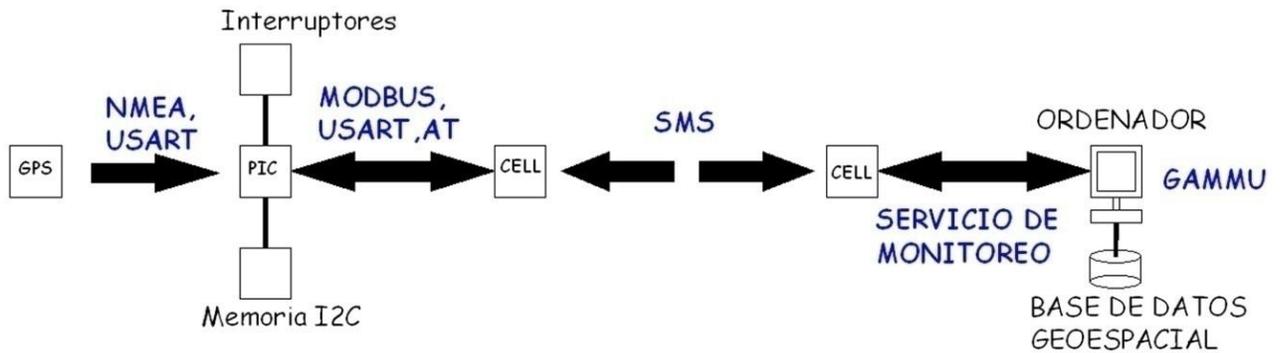
En el segundo capítulo se muestran los criterios de selección de dispositivos y la forma de utilización de la parte móvil del proyecto, que es la que va a estar tomando las mediciones, en campo, del levantamiento topográfico para luego ser enviadas en forma inalámbrica hacia un equipo que se encuentra en oficina. Este equipo móvil recoge datos georeferenciados desde un equipo receptor GPS y es con estos datos que se realiza el levantamiento topográfico.

### 2.2. El sistema.

El sistema está conformado en dos partes: el **equipo cliente**, que será la parte del sistema que se encargará de obtener las coordenadas georeferenciadas del lugar del cual se realizará el levantamiento topográfico; y el **equipo servidor** que es donde se guardarán los datos, extraídos por parte del equipo cliente, en una base de datos; además de realizar en forma automática la tarea construir la representación grafica del levantamiento topográfico para ser mostrados en un aplicativo correspondiente (web). La figura 2.1 muestra la representación de cómo estará conformado el sistema: un equipo GPS, el circuito electrónico basado en microcontrolador, la comunicación inalámbrica que se realizará con dos teléfonos celulares, además de un ordenador. En la figura 2.2 está representado el sistema, en forma de módulos, con sus diferentes protocolos para la comunicación entre estos equipos.



**Figura 2. 1** Representación del Sistema de Levantamiento topográfico.



**Figura 2. 2** Representación del Sistema en forma de Módulos del Levantamiento topográfico.

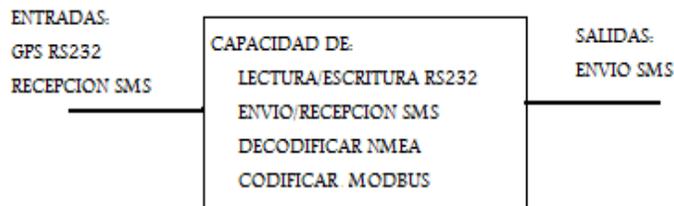
El diseño de la solución al sistema es el mostrado en la figura 2.3 en el cual se observa las entradas que debe tener las dos partes principales del proyecto al igual que los procedimientos a realizar por cada parte.

COMPONENTES:

PLANTEAMIENTO DE LA SOLUCION:

SISTEMA PROTOTIPO:

EQUIPO MOVIL (CAJA NEGRA) CON ENTRADAS Y SALIDAS:



EQUIPO SERVIDOR EN OFICINA CON ENTRADAS Y SALIDAS:

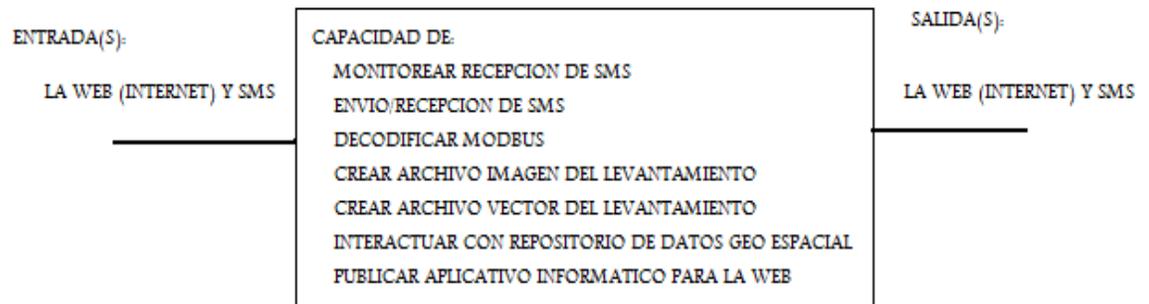
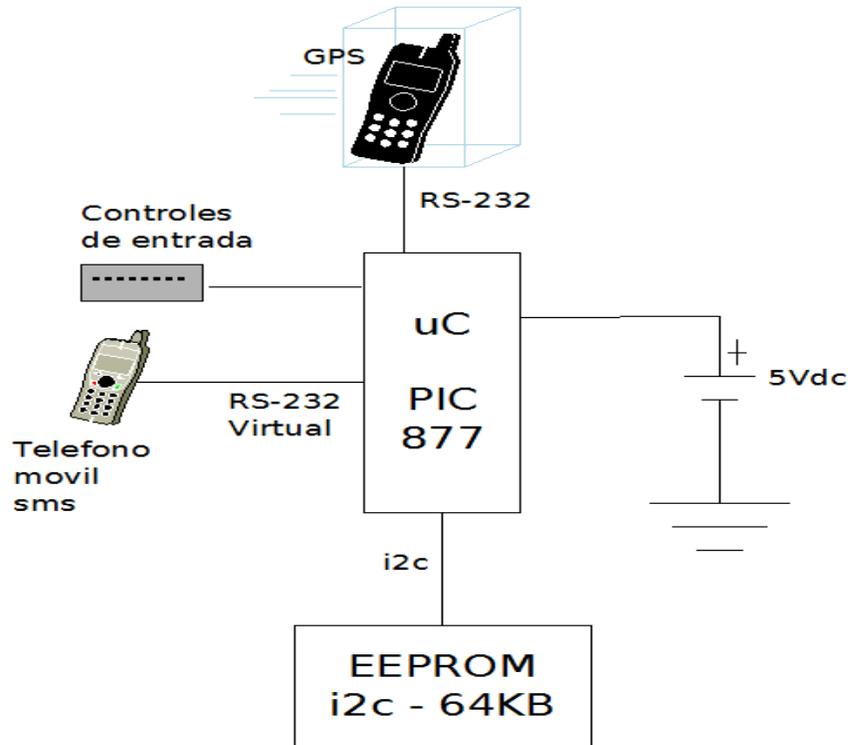


Figura 2. 3 Diseño de la solución del proyecto.

### 2.3. Criterios de selección del equipo cliente

Esta es la parte móvil de sistema, y donde se encuentra la mayor parte del hardware a utilizar. La figura 2.4 muestra la representación en módulos del equipo cliente y la forma en cómo se comunican entre sí los diferentes partes del que está constituido.

Los módulos centrales del equipo cliente son: el GPS, un microcontrolador y un teléfono celular.



**Figura 2. 4** Representación en forma de Módulos del Equipo Cliente.

Para la extracción de las coordenadas del GPS se realizan por el puerto serie que este dispositivo posee, hacia un puerto USART del microcontrolador, el cual guardará en una memoria EEPROM externa mientras se finalizan las mediciones. Cuando estén finalizados las toma de coordenadas o la memoria EEPROM este a un 80% de su capacidad, el microcontrolador enviara las coordenadas al equipo servidor por medio de mensajes de texto; esto será realizado por el celular conectado al microcontrolador por otro puerto USART, la comunicación entre el microcontrolador y el celular se hace mediante los comandos AT y se codifican los datos con la trama del protocolo MODBUS.

Cada parte del sistema cliente se tuvo que seleccionar de una gran gama de opciones disponibles en el mercado de acuerdo a las necesidades requeridas por el proyecto, además de la disponibilidad en el mercado y no obviando el valor económico que estos poseen.

### 2.3.1. Criterios para la selección del dispositivo receptor GPS

Al realizar un levantamiento topográfico se deben de detallar los niveles de error y precisión que se pueden tener en las coordenadas geodésicas que nos dan los dispositivos GPS.

En el mercado existen una gran variedad de dispositivos GPS con diferentes aplicaciones y estilos, entre los cuales encontramos modelos que nos muestran mapas a color y en escala de

grises, otros que no pueden mostrar un mapa en pantalla; además de diferentes niveles de precisión de las coordenadas georeferenciadas y que pueden trabajar en los sistemas americanos y europeo.

Los dispositivos GPS que tiene un mayor grado de precisión en sus coordenadas son aquellos que su valor económico es mayor.

Los requerimientos que demanda este proyecto para el dispositivo GPS son que debe de poseer una precisión de milímetros, para la realización de levantamientos topográficos en zonas urbanas, donde se requiere tal precisión; pero el costo económico se eleva con los dispositivos que poseen las características requeridas.

Para la solución del problema de la precisión del equipo se busco un dispositivo GPS que tiene en sus características precisiones de menos de 10 metros para no elevar los costos y demostrar un funcionamiento aceptable para la ejecución del proyecto.

El dispositivo GPS seleccionado es de la marca GARMIN y modelo ETREX LEGEND que posee características aceptables para la ejecución del proyecto y además el costo económico es bajo en comparación de los equipos con mayor precisión.

Las características técnicas del dispositivo GPS ETREX LEGEND las podemos apreciar en la tabla 2.1 Las cuales muestran características como la precisión del equipo, características físicas entre otras.

**Tabla 2.1 Parámetros del comando +CMGS.**

Navegación	
Waypoints / Favoritos - Nº	1000
Track log (registro de trayecto)	10.000 pts. + 20 tracks de 500 ptos. c.u.
Rutas	50 reversibles (250 wpts./ruta)
Presentación datos de trayecto	Odómetro, Altitud GPS, Veloc. media en movimiento, Tiempo hasta destino, Distancia hasta destino, Distancia recorrida, etc.
Registro datos de trayecto	Sí
Rendimiento	
Receptor	Chipset GPS de alta sensibilidad
T' de arranque en frío	39 seg. promedio
T' de arranque en caliente	3 seg. promedio
Precisión de posicionamiento GPS	< 10 metros el 95% en condiciones óptimas

Precisión de posicionamiento DGPS	3 metros el 95% en condiciones óptimas
Nº de canales	12
WAAS / EGNOS	Sí
<b>Características Físicas</b>	
Tamaño del receptor	10.7 x 5.6 x 3 cm (ancho x alto x grosor)
Peso	156 g. (con baterías)
Tamaño de pantalla	3.3 x 4.3 cm (ancho x alto)
Resolución de pantalla	176 x 220 píxeles
Tipo de pantalla	TFT-LCD de 256 colores
Orientación de pantalla	Vertical
Pantalla táctil	No
Joystick / Botón de navegación por menús	Sí
Antena	Integrada tipo parche
Conexión de antena externa / tipo	No
Memoria interna	No
Expansión de memoria	No
Conectividad	RS-232, serial
Voltaje usado / alimentación externa	- / 12-24V
Batería	2 x AA
Duración de la batería	hasta 25 horas

### 2.3.2. Criterios para la selección del microcontrolador.

Para obtener el microcontrolador óptimo al desarrollar proyectos se deben de tener en cuenta diversos factores, de los cuales podemos mencionar:

#### Recursos comunes a todo mC

- ✓ Procesador
- ✓ Memoria de datos y de instrucciones
- ✓ Líneas de E/S
- ✓ Oscilador de reloj
- ✓ Módulos controladores de periféricos

#### El tipo de tecnología que utiliza:

- ✓ Gama baja: 4, 8 y 16 bits. Dedicados fundamentalmente a tareas de control (electrodomésticos, cabinas telefónicas, smart-cards, algunos periféricos de ordenadores, etc.) Generalmente son mC.
- ✓ Gama media: 16 y 32 bits. Tareas de control con cierto grado de procesamiento (control en automóvil, teléfonos móviles, PDA, etc). Suelen ser P o C + periféricos integrados, y memoria externa.
- ✓ Gama alta: 32, 64 y 128 bits. Fundamentalmente procesamiento (ordenadores, videoconsolas, etc.) Casi en su totalidad son mP + circuitería periférica + memoria

Micros de 4 bits:

Pocos bytes de RAM, sin SO, todo el software en ensamblador, cada vez menos usados

Micros de 8 bits:

RAM de unos pocos bytes a unos cientos de KB, usan ensamblador, pero también C, C++, Java. Pueden llevar SO específico

Micros de 16 y 32 bits

RAM de pocos KB a muchos MB, generalmente llevan un SO de tiempo-real, pueden o no tener cachés

Micros de 32 o 64 bits

Básicamente un PC en un envase pequeño, llevan Win XP, Linux; relativamente caros

### **Lenguajes de programación**

- ✓ Ensamblador

Ocupa el espacio mínimo, código eficiente, no es portable

- ✓ C

Pequeña sobrecarga de espacio y tiempo, más o menos portable, existen buenos compiladores

- ✓ C++

A menudo empleado como un 'C mejorado', poca sobrecarga de espacio y tiempo si se emplea con cuidado.

- ✓ Java

Más portable, requiere mucha RAM, J2ME popular en dispositivos móviles, no es bueno para tiempo real.

### **Otros factores**

- ✓ Experiencia y soporte
- ✓ Compatibilidad
- ✓ Disponibilidad y segundas fuentes

Para el proyecto realizado, tomando en cuenta los factores anteriormente expuestos se escogió el microcontrolador de la empresa MICROCHIP. El microcontrolador PIC16F877 debido a que ya se tiene la experiencia con estos dispositivos cuando se curso la materia Microcontroladores, de la currícula de la Escuela de Ingeniería Eléctrica.

### **2.3.3. Criterios para la selección de la memoria externa.**

Cuando se realizan trabajos a la intemperie no se tiene las facilidades que se tienen en un laboratorio como lo son la energía eléctrica, previendo esto el equipo cliente puede tener una falta de energía al estar siendo utilizado en un periodo largo de tiempo, si este inconveniente llegara a ocurrir y los datos extraídos del dispositivo GPS que no se han enviado se perderán y se tendría que realizar el trabajo nuevamente. Una solución a este problema es almacenar los datos en una memoria EEPROM.

**EEPROM** son las siglas de *Electrically-Erasable Programmable Read-Only Memory* (ROM programable y borrable eléctricamente). Los datos almacenados en un chip EEPROM son permanentes, por lo menos hasta que el usuario decida que va a borrar o sustituir la información que contiene. Además, los datos almacenados en un chip EEPROM no se pierden aun cuando el dispositivo está apagado. Estos dispositivos suelen comunicarse mediante protocolos como I2C, SPI y Microwire

Por las características que posee la memoria EEPROM es una opción óptima para almacenar datos sin pérdidas, por falta de energía, además de poder sustituir los datos almacenados a la hora de realizar un nuevo levantamiento. Otra ventaja es que se comunicación que se realiza mediante protocolo I2C el cual también posee el microcontrolador.

La memoria seleccionada es una memoria EEPROM 24LC512 de la marca Microchip la cual posee una capacidad de almacenaje de 512 kbits que se traduce en 64 kBytes; con tamaño de paginas de 128 bytes y una frecuencia máxima de reloj de 400 kHz

### **2.3.4. Criterios para la selección del teléfono celular.**

La comunicación entre los equipos cliente y servidor se debe de realizar por medios inalámbricos, para que el equipo cliente se mueva con facilidad y poder ser transportado a grandes distancias del equipo servidor. Los medios inalámbricos son diversos, entre los cuales está la comunicación por medio de ondas de radio, teléfonos celulares, que son medios inalámbricos que abarcan grandes distancias. Existen otros medios de comunicación inalámbricos como WIFI y BLUETOOTH, pero existe una limitante en estas tecnologías que es la poca distancia en la que operan estas redes.

La comunicación por medio de radio complica nuestro proyecto porque se deben de adquirir el equipo transmisor, receptor y algunos repetidores, por tanto la opción más factible para la comunicación entre el equipo cliente y el equipo servidor es por medio de la red celular. La red

de telefonía celular ya posee la estructura física para la cobertura casi en su totalidad de todo el país, por ende es la mejor opción para realizar este proyecto.

Ya teniendo la comunicación inalámbrica establecida por medio de la telefonía celular; la forma de enviar datos por medio del teléfono celular es por medio de mensajes de texto en los cuales hoy en día se puede enviar archivos multimedia imágenes, video y sonidos.

El los mensajes de texto solo enviaremos caracteres que serán las diferentes coordenadas que están guardadas en la memoria externa EEPROM que fueron extraídas del receptor GPS, además de otros caracteres para la codificación del mensaje en una trama MODBUS.

La comunicación entre el teléfono celular y el microcontrolador se hace en forma serial utilizando el modulo USART del microcontrolador con las características de 9600 baudios, 8 bits de datos, sin bit de parada y un bit de stop. La forma en que se comunican el microcontrolador y el teléfono celular es por medio de comandos AT, por los cuales el microcontrolador le dirá al teléfono celular cuando enviar el mensaje al celular del equipo servidor, además de la forma en que se recibirán los mensajes provenientes del equipo servidor.

Un pequeño ejemplo de cómo enviar un mensaje de texto por medio de comandos AT es el mostrado a continuación, el cual se realizo por medio de la hiperteminal de Windows.

```
1      AT
2      OK
3      AT+CMGF?
4      +CMGF: 0
5      OK
6      AT+CMGF=1
7      OK
8      AT+CMGS="numero de celular"
9      > AQUI ESCRIBO EL MENSAJE      [CTRL-Z]
10     OK
```

En la línea 1 está el comando básico, y el teléfono celular responde OK en la línea 2; la forma de preguntar una configuración del teléfono celular se muestra en la línea 3, y la respuesta a esta pregunta está en la líneas 4 y 5, lo que quiere decir es que este parámetro tiene esa configuración y se traduce en que la comunicación con el teléfono es vía RTU, la cual se explico en los apartados anteriores; en la línea 6 esta como se cambian los parámetros, para el caso se configura la comunicación con el teléfono en modo TEXTO; para el envío de un SMS está el comando en la línea 8 y 9 del ejemplo, en la línea 8 se envía el comando incluyendo el número del teléfono celular al que se pretende enviar el SMS, después el teléfono debe responder con el carácter “mayor que” (>) el cual nos indica que podemos escribir el mensaje a enviar. Para indicar el final del mensaje se realiza por medio de [CTRL+Z], que es lo que se muestra en la línea 9; y el celular responde con un código de aceptación del comando, ya sea con un “OK” el cual confirma el envío del SMS o un código de error, que nos dice que no se ha podido enviar el mensaje.

De la forma vista en el ejemplo anterior es como el microcontrolador debe enviar los SMS en conjunto con el teléfono celular, pero los SMS deben estar con el formato MODBUS; a las coordenadas enviadas se le deben añadir la cabecera y cola de la trama MODBUS.

## 2.4. Implementación del equipo cliente

La comunicación entre los diferentes dispositivos del equipo cliente es una es la parte más importante de realizar: comunicar el dispositivo GPS con el microcontrolador, el microcontrolador con la memoria EEPROM y la comunicación entre el microcontrolador con el teléfono celular.

Cada dispositivo se comunica con los otros a través del microcontrolador que es el que administrará los diferentes eventos y tomara decisiones de cómo realizar las diferentes operaciones a ejecutar.

En la figura 2.4 Se muestra el circuito eléctrico implementado en este proyecto, se puede apreciar el microcontrolador, la pantalla LCD, la memoria EEPROM, además de los pulsadores ocupados; en este caso para efectos de representación se encuentran dos módulos que representa al teléfono celular y al receptor GPS. En el anexo B se encuentra el diagrama del circuito impreso que se implementado.

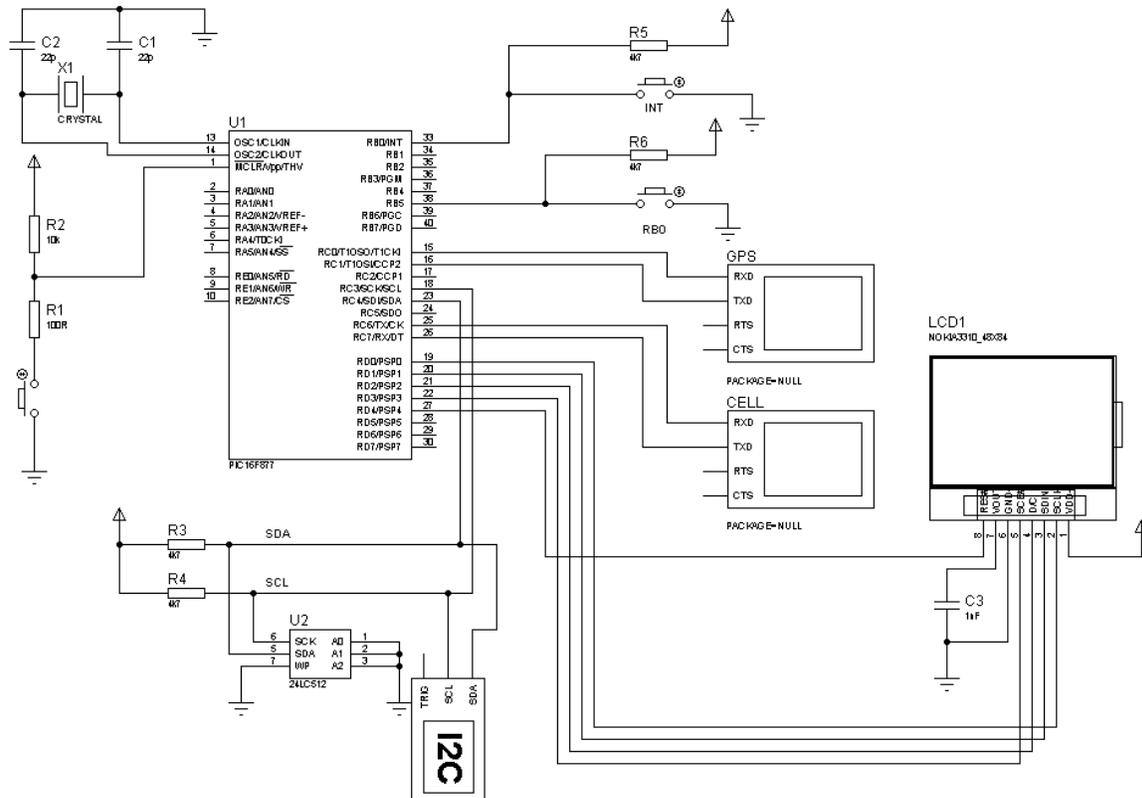


Figura 2.4 Diagrama eléctrico implementado en el quipo móvil

## 2.4.1. Comunicación GPS y microcontrolador

La forma en que los dispositivos GPS tienen para comunicarse con otros dispositivos externos es a través de puertos seriales ya sea RS232 o USB. El dispositivo utilizado en este proyecto es un ETREX LEGEND de la marca GARMIN el cual posee un puerto serie con comunicación RS232; en este puerto se puede configurar para que el dispositivo GPS se comunique con diferentes formatos de datos entre los cuales está el protocolo NMEA, el cual es para comunicaciones entre dispositivos de navegación, también se puede configurar la comunicación en modo ASCII, y otras configuraciones propias del fabricante.

El formato a utilizar en la comunicación entre el dispositivo GPS y el microcontrolador es el protocolo NMEA con estas especificaciones: 4800 b/s (bits por segundo) con 8 bits de datos, sin paridad, y un bit de stop. Se utiliza este protocolo para estandarizar la comunicación con dispositivos GPS, esto quiere decir que se puede utilizar cualquier dispositivo GPS que utilice el protocolo y el equipo cliente trabajará de la misma manera (podemos sustituir el dispositivo GPS utilizado por otro de mayor precisión).

Los niveles de voltaje del protocolo RS232 están entre -9 voltios a +9 voltios los cuales se deben adecuar a niveles TTL que es con los que trabaja el microcontrolador; para solventar este inconveniente se utiliza un integrado MAX232, el cual podemos apreciar en la figura 2.4.

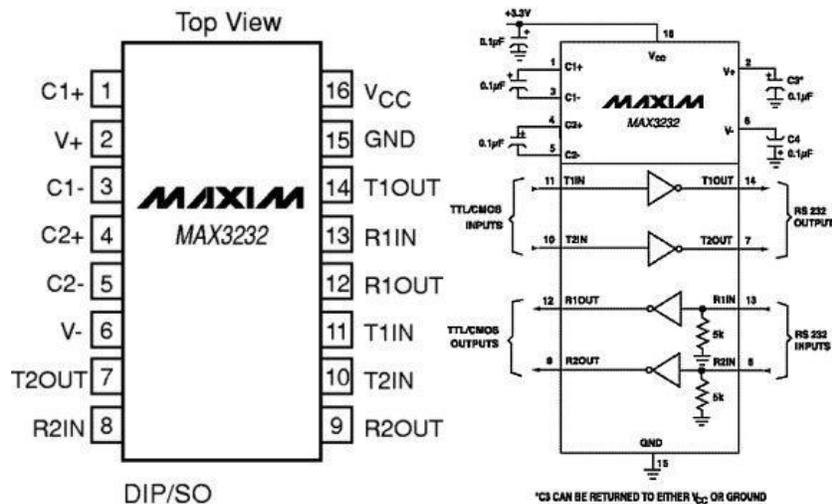


Figura 2. 5 circuito integrado MAX232.

Al colocar el circuito integrado MAX232 se tiene todo el hardware necesario para la comunicación entre el receptor GPS y el microcontrolador, ahora se le debe decir al microcontrolador la tarea que debe realizar con los datos que el receptor GPS le enviará.

Al colocar el puerto RS232 del receptor GPS en una computadora comunicándose con el hiperterminal, configurada para el protocolo NMEA, los resultados se muestran a continuación y se pueden apreciar las diferentes frases NMEA que el receptor GPS coloca en su puerto de comunicación.

```

$GPRMC,201604,A,1339.4312,N,08910.9173,W,0.0,185.5,280708,2.0,E,A*06
$GPRMB,A,,,,,,,,,A,A*0B
$GPGGA,201604,1339.4312,N,08910.9173,W,1,04,3.3,771.9,M,-2.2,M,,*40
$GPGSA,A,2,03,,,13,19,,23,,,,,3.4,3.3,1.0*3D
$GPGSV,3,2,12,13,47,330,38,19,65,113,45,20,25,203,00,23,83,329,44*7E
$GPGLL,1339.4312,N,08910.9173,W,201604,A,A*54
$GPBOD,,T,,M,,*47
$GPVTG,185.5,T,183.5,M,0.0,N,0.0,K*48
$PGRME,13.4,M,25.8,M,29.1,M*1D
$PGRMZ,2533,f,2*2D
$PGRMM,WGS 84*06
$GPRTE,1,1,c,*37

```

Estas son las frases NMEA que el receptor GPS envía por el puerto de comunicación que posee; la frase que nos interesa es la frase GGA, que donde se encuentran las coordenadas georeferenciadas: latitud, longitud y la altura. Para el caso mostrado:

Latitud	1339.4312,N	13 grados, 39.4312 minutos, latitud norte.
Longitud	08910.9173,W	89 grados, 10.9173 minutos, longitud oeste.
Altura	771.9,M	771.9 metros sobre el nivel del mar.

Para capturar estos datos el microcontrolador debe seleccionar la frase requerida (GPGGA) y luego dentro de esta frase seleccionar los datos necesarios, recordando que los caracteres se encuentran en formato ASCII.

## 2.4.2. Comunicación microcontrolador memoria EEPROM.

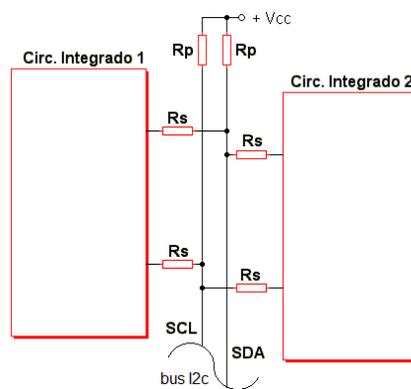
La forma en que se comunica la memoria EEPROM seleccionada es por el puerto serial con formato I2C, el cual debe ser configurado en el microcontrolador en el modulo MSSP para una frecuencia de 100 kHz.

El protocolo I2C es implementado en forma de maestro-esclavo, en este caso el maestro es el microcontrolador y el esclavo es la memoria EEPROM.

Los requerimientos de hardware para establecer la comunicación solo es un bus de dos líneas, una para el reloj y otra para la transmisión de los datos; estas líneas deben poseer un par de resistencias (resistencias de pull-up) conectadas a Vcc para establecer una tensión en las líneas; la forma en que se deben de conectar los dispositivos al bus I2C es como se muestra en la figura 2.5 En la cual se muestran las líneas del bus y las resistencias colocadas en el bus.

Dependiendo de los requerimientos del circuito integrado se colocaran las resistencias en serie.

Existen dos operaciones que se realizan con la memoria: la escritura en la memoria y la lectura de los datos guardados. Para la escritura en una posición en la memoria el maestro, en este caso el microcontrolador, debe generar la condición de Start, después debe colocar la dirección del esclavo (memoria EEPROM) de tamaño de 1 byte, con el séptimo bit en cero, con esto indica que es una operación de escritura; el maestro debe esperar un bit de ACK como respuesta del esclavo al haber una coincidencia en la dirección. Después el maestro envía la dirección de la posición de la memoria donde se guardará el dato, el tamaño de la dirección es de dos bytes para esta memoria, y el maestro debe esperar los correspondientes ACK para cada byte. En estos momentos la memoria esta seleccionada y tiene la dirección donde empezará a guardar los datos que es lo que debe enviar el maestro, esperando el ACK por cada byte enviado. Al finalizar el envío de los datos el maestro debe poner en el bus una condición de STOP y así será finalizada la comunicación con la memoria EEPROM en la operación de escritura de datos.



**Figura 2. 6 Conexión de dispositivos en un bus I2C.**

Debemos tener cuidado a la hora de hacer la escritura en la memoria, por el tamaño de la paginación que posee la memoria, para nuestro caso cada página está constituida por 128 bytes. Si en una escritura secuencial se sobrepasa el tamaño de la página, la dirección de la memoria desbordará hasta la posición cero de la página actual.

La lectura de la memoria es similar a la escritura salvo que el byte de dirección de esclavo el séptimo bit esta a 1, y el maestro es el que genera los ACK después de leer los datos almacenados en la memoria; para finalizar la lectura de datos el maestro debe contestar con un NO-ACK y poner una condición de STOP.

### **2.4.3. Comunicación microcontrolador celular.**

Una parte importante del Equipo Cliente es la forma en que este se comunica con su contraparte, el Equipo Servidor. La comunicación se realiza a través de dos teléfonos celulares por medio de SMS, y cada mensaje en formato MODBUS, el cual llevara las coordenadas que están guardadas en la memoria EEPROM.

Para el envío de SMS hacia el Equipo Servidor, el usuario debe de intervenir lo menos posible y así evitar errores de escritura, por esto la mejor opción es que los SMS se escriban y envíen en forma automática, la manera de controlar el teléfono celular es a través de los Comandos AT.

Los Comandos AT los debe de generar el microcontrolador a través de un puerto serie configurado a 9600 baudios. Por medio de este puerto se enviaran los comandos hacia el Teléfono celular y así controlar el envío y recepción de SMS hacia y desde el Equipo Servidor.

Además por el bus serie que ha establecido entre el microcontrolador y el teléfono celular se deben enviar las coordenadas que se encuentran almacenadas en la memoria EEPROM.

La comunicación puede realizarse por medio de cables entre el teléfono celular y el microcontrolador, también puede realizarse por medio inalámbrico como BLUETOOTH, dependiendo de la características que posea el teléfono celular a utilizar. Para nuestro caso la comunicación es realizada por medio de cables conectados en el puerto de comunicaciones del teléfono celular.

El puerto de comunicación del teléfono celular se llama POP-PORT, y se muestra en la figura 2.7, La configuración de pines es la siguiente:

1-Vin	6-Rx fbus	11-HS ear n
2-Gnd	7-Tx fbus	12-HS ear p
3-ACI	8-GND	13-HS ear R n
4-Vout	9-X mic N	14-HS ear R p
5- --	10-Xmic P	



**Figura 2. 7 Puerto de comunicación del celular (POP-PORT).**

Los pines utilizados son Rx, TX y GND que son los pines 6,7,8 respectivamente del puerto de comunicación y con estos controlaremos el flujo de información para el control del teléfono celular mediante los comandos AT, afín de enviar y recibir SMS para la aplicación de este proyecto.

## 2.5. Funcionamiento del equipo cliente.

Al realizar el trabajo de campo de un levantamiento topográfico se utilizan diferentes equipos para realizarlo, en este caso solo se estará ocupando el Equipo Cliente para realizar esta función, por autónomo y fácil de manipular.

La primera operación a realizar, esto por parte de la persona que opera el equipo, es la verificación de que el receptor GPS está obteniendo las señales de los satélites, como se muestra en la figura 2.8, y así el microcontrolador pueda guardar las coordenadas georeferenciadas; de lo contrario se estarán extrayendo del receptor GPS datos no validos.

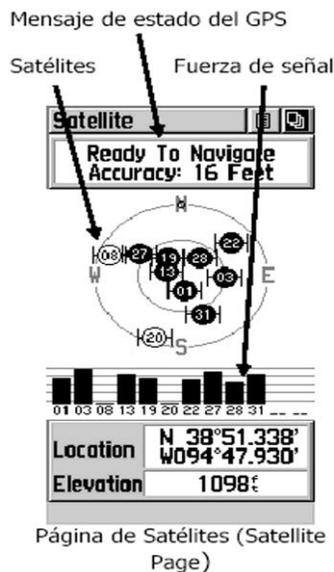


Figura 2. 8 Pantalla principal de receptor GPS.

El Equipo Cliente posee diferentes indicadores para la realización del trabajo de campo, el primer indicador es si el aparato muestra que si se ha establecido la comunicación entre el microcontrolador y el teléfono celular aceptando los comandos AT para el seteo del teléfono en forma adecuada para el envío y recepción de los SMS.

Una vez la persona que opera el equipo ha verificado la señal óptima del receptor GPS, y por medio de un indicador led, la comunicación entre el microcontrolador y el teléfono celular, el equipo está listo para realizar la toma de datos del levantamiento en cuestión.

Al realizar la toma de datos georeferenciados el Equipo Cliente nos muestra en pantalla el número de punto que tiene almacenado en su memoria. Si por un evento no imprevisto se toma una medición equivocada, el equipo posee la capacidad de borrar un punto guardado o borrar todos los puntos.

El envío de las coordenadas almacenadas en el Equipo Cliente se puede realizar por dos formas: una porque el usuario termino de realizar el levantamiento y la otra porque el equipo ya tiene un 80% de memoria utilizada en ese levantamiento.

El número máximo, en hexadecimal, que puede ser direccionado en la memoria EEPROM es 0XFFFF que es equivalente a 65,535 bytes. E cada punto guardado del levantamiento se almacena 24 bytes, esto está conformado por la latitud, longitud y altura en formato de gramos y minutos, por tanto el número total de puntos guardados en la memoria es de 65,535/24 que da un resultado de 2730.63 puntos. El 80% de esta cantidad es de 2184.5 puntos.

Si por alguna razón se llegara a almacenar el 80% de la memoria y se enviaran los datos almacenados se tendrían que enviar aproximadamente 365 mensajitos, porque en cada mensaje se pueden enviar 6 puntos, que quiere decir que son 144 bytes. Al enviar 365 mensajitos con un costo de \$0.05 de dólar (tarifa regular) el teléfono celular debería de tener aproximadamente \$37 dólares de saldo; y esto con la suposición de que la compañía que presta el servicio a los teléfonos celulares no se saturaría por la cantidad de mensajitos enviados en periodo corto de tiempo.

Antes de enviar las coordenadas al Equipo Servidor se procede a enviar una petición de ID de levantamiento, esto para dar oportunidad a otros equipo, porque puede existir más de un Equipo Cliente realizando levantamientos, para obtener un identificador de levantamiento realizado y así ser almacenado en la base de datos del Equipo Servidor.

La tabla 2.2 Nos muestra la descripción de cada función de la trama MODBUS, los cuales están implementados en el proyecto.

**Tabla 2.2 Descripción de la función de la trama MODBUS.**

FUNCION	DESCRIPCION
00	Petición de ID de levantamiento por parte del Equipo Cliente
01	Envío de coordenadas de un levantamiento realizado por parte de Equipo Cliente.
02	Función de error por parte del Equipo Cliente indicando que no posee o no son suficientes las coordenadas que se encuentra almacenadas en el Equipo Cliente

La forma en que se envía el SMS desde el Equipo Cliente hacia el Equipo Servidor es cómo se muestra a continuación, lo que se observa es el comando AT para envío de mensajes, en la fila 7 del ejemplo, y el mensaje de petición de número de levantamiento en la fila 8. Las primeras filas son los comandos AT, y la respuesta del celular, para la configuración del teléfono celular. El primer comando limpia a los valores por defecto toda la configuración anterior; el segundo

comando configura el teléfono celular en formato ASCII para el envío de SMS; y el tercer comando setea el teléfono para no almacenar los mensajitos en la memoria del teléfono. Después de esta configuración cada mensaje enviado o recibido se traslada inmediatamente al puerto de salida del teléfono y así no se saturaría la memoria del celular después de varios mensajitos enviados y recibidos; esto se realiza para que el usuario no se preocupe de estar limpiando la memoria del teléfono.

```
ATZ
OK
AT+CMGF=1
OK
AT+CNMI=1,2,0,0,0
OK
AT+CMGS="72308163"
> :01000000010000000000000000000000FE
>
→
OK
```

Al enviar este mensaje el Equipo servidor le indica al Equipo Cliente el numero de mensajes que se enviaran con coordenadas que es de 1 mensaje (son los caracteres 9 y 10 del mensaje a partir del carácter “:”)

Cuando el Equipo Servidor responde al mensaje enviado anteriormente lo hace de la siguiente forma:

```
+CMT: "+50372308163", , "09/11/11,11:00:27-24"
:01000411111111B7
```

Esto es la respuesta del celular del Equipo Cliente el cual indica la recepción de un SMS por parte del Equipo Servidor; se observa el número de teléfono del cual procede el mensaje recibido, la fecha y hora de recepción del mensaje y en la segunda fila, el mensajito de texto el cual contiene el numero de levantamiento que le ha asignado el Equipo Servidor, para el caso se le ha asignado el levantamiento numero 04 (caracteres 5 y 6 después del carácter “:” en la segunda fila).

Cuando el Equipo Servidor ha recibido el SMS con el número de levantamiento asignado el microcontrolador debe de procesarlo desglosándolo y averiguando si viene con formato MODBUS, además de encontrar el checksum de la trama, comparándolo con el revido en el mensajito.

Una vez se ha determinado que el mensajito es correcto y proviene del numero de celular correspondiente, se procede a crear y enviar el o los mensajitos que llevarán las coordenadas del levantamiento realizado; en este proceso el microcontrolador se encarga de extraer los datos que se encuentran almacenados en la memoria EEPROM de 64KBTS. Cuando se extraen los datos de la memoria se almacenan los datos en la memoria RAM del microcontrolador porque de esta forma los datos son más accesibles a la hora de estar enviándolos al celular.

La forma en que se envían los mensajitos con las coordenadas del levantamiento es como se observa a continuación:

```
AT+CMGS="72308163"  
> :010104000113394423A089109207A7589B13394420A089109207A7592B13394413A089109207A  
7602B13394403A089109211A7601BF5  
>  
→  
OK
```

Lo primero que se observa es el comando AT, enviado por el microcontrolador hacia el modem del teléfono celular, que indica el envío de mensaje con el correspondiente número al cual se envía el mensajito. La segunda y tercera fila es la trama MODBUS el cual lleva las coordenadas del levantamiento realizado. Y al final la respuesta del celular el cual es un OK, que nos indica que se ha enviado el SMS.

Cuando se envían dos o más SMS, el microcontrolador los realiza con un retardo de tiempo de 5 segundos, esto porque en algunos teléfonos celulares se obtiene una confirmación de que se está enviando un SMS y/o que se ha enviado un SMS. Después de este tiempo el celular está listo para enviar otro mensajito por lo tanto el microcontrolador procede a enviar los siguientes SMS.

Desglosando el mensajito del ejemplo anterior, se tiene que el mensaje esta con formato MODBUS del cual se puede determinar que 01 es la dirección de equipo al que está dirigido el mensaje; el siguiente 01 es la función a realizar, para el caso es el envío de los datos almacenados en memoria; lo siguiente es 04 que es el ID de levantamiento realizado; lo que sigue es el numero de SMS enviado el cual es: 0001; el resto del mensajito son las coordenadas del levantamiento realizado, para el caso son cuatro coordenadas las que se están enviando, esto se observa porque cada punto del levantamiento está separado por un carácter "B" dentro del mensajito, la latitud, longitud y altura van separado por un carácter "A"; por último se encuentra el checksum de la trama MODBUS , que en este mensaje es de F5, que son los dos últimos caracteres del mensajito.

## 2.6. Flujo de información en el equipo cliente.

El flujo de información o de datos del Equipo Cliente lo determina el microcontrolador. Es capaz de direccionar los datos que están fluyendo por los puertos que están conectados, además de desechar los datos que no son útiles en el proyecto.

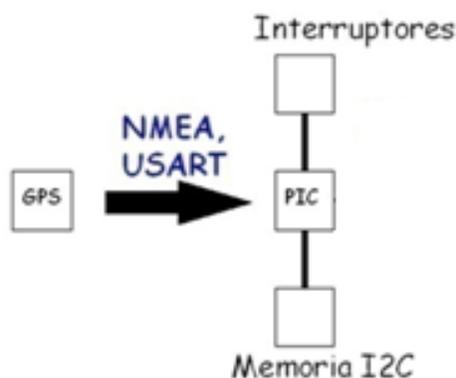
Al realizar un levantamiento se debe de obtener los datos que el receptor GPS está colocando en su puerto de salida, estos datos están en formato ASCII y están disponibles en el puerto serial del receptor GPS a través del protocolo NMEA. Una muestra de los datos que el Receptor GPS coloca en su puerto de salida es como se muestra a continuación:

```

$GPRMC,180018,A,1339.4293,N,08910.9211,W,0.0,0.0,141109,2.0,E,A*08
$GPRMB,A,1.88,L,,San Salvado 84*06
$GPRTE,1.1,c,*37
$GPRMC,180022,A,1339.4293,N,08910.9211,W,0.0,0.0,141109,2.0,E,A*01
$GPRMB,A,1.88,L,,San Salvador,1342.9858,N,08912.0447,W,3.727,342.9,,V,A*2A
$GPGGA,180022,1339.4293,N,08910.9211,W,1.03,2.4,751.9,M,-2.2,M,,*44
$GPGSA,A,2,,04,,17,,28,,2.6,2.4,1.0*38
$GPGSV,3,1,11,02,39,234,00,04,72,265,45,07,01,155,00,09,12,314,00*7C
$GPGLL,1339.4293,N,08910.9211,W,180022,A,A*53
$GPBOD,132.7,T,130.7,M,San Salvador,*09
$GPVTG,0.0,T,358.0,M,0.0,N,0.0,K*40
$PGRME,21.4,M,50.0,M,54.4,M*19

```

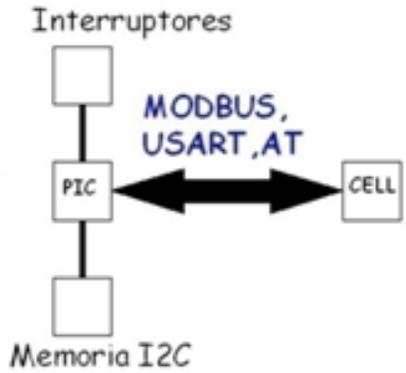
El microcontrolador debe de determinar la trama correcta para extraer los datos de latitud, longitud y altura. La trama que el microcontrolador busca es la que tiene cabecera GPGGA de la cual se ha explicado en los capítulos anteriores.



**Figura 2. 9 Flujo de información entre receptor GPS y microcontrolador.**

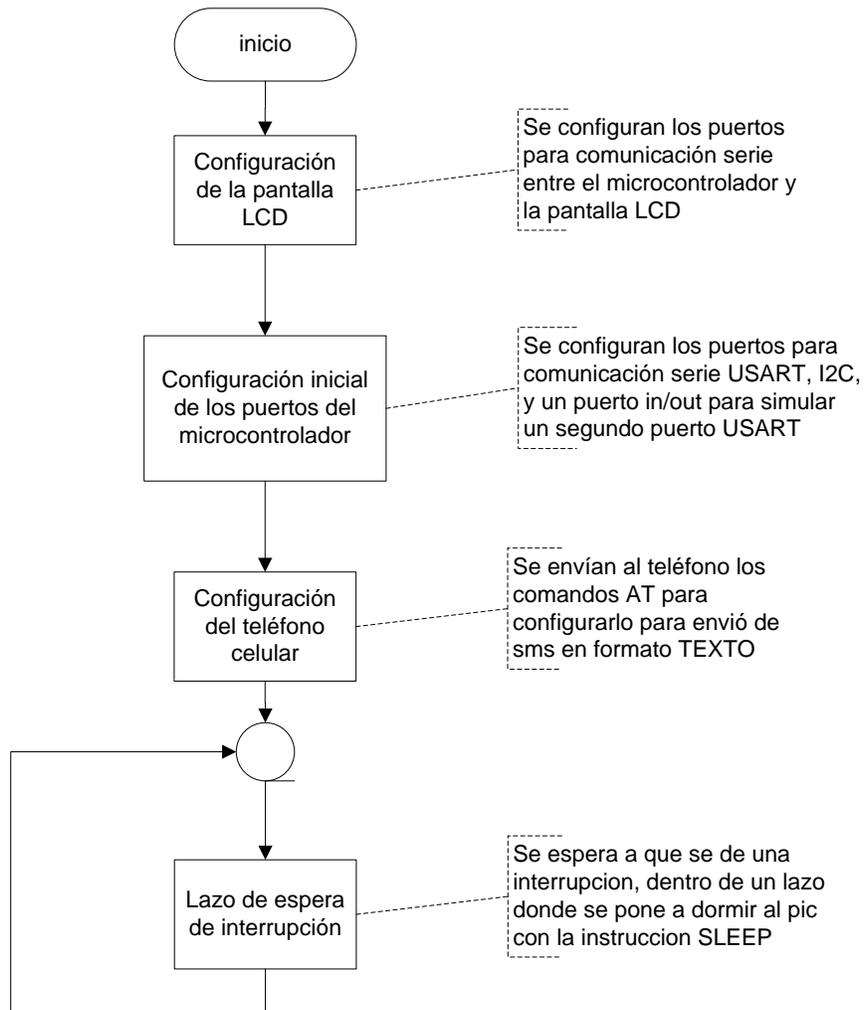
La figura 2.9 muestra como fluye la información entre el receptor GPS y el microcontrolador es en un solo sentido desde el receptor GPS hacia el microcontrolador. El puerto que se utiliza en el microcontrolador es un puerto USART virtual; el puerto USART Virtual no es un puerto físico del microcontrolador se realiza por medio de Software, esto porque el microcontrolador PIC16F877 solo posee un puerto UASRT físico, y este puerto está conectado el teléfono celular.

La otra parte de Equipo Cliente es la comunicación entre el teléfono celular y el microcontrolador, esta comunicación se realiza en forma serial por el puerto USART del microcontrolador; la información entre estos dos dispositivos es en ambos sentidos desde y hacia el teléfono celular como se muestra en la figura 2.10 los comandos AT viajan de esta forma para la configuración, recepción y envío de SMS.



**Figura 2. 10 Flujo de información entre microcontrolador y teléfono celular.**

El diagrama de flujo del programa principal que el microcontrolador implementa es la que se muestra en la figura 2.11



**Figura 2. 11 Diagrama de flujo de la rutina principal del microcontrolador.**

El diagrama de flujo de la rutina de interrupción para la captura de las coordenadas GPS es la que se muestra en la figura 2.12 se debe destacar que en esta rutina es donde se termina si la memoria ya ha alcanzado el 80% de datos almacenados, si se ha alcanzado el porcentaje requerido se envían los SMS sino termina la interrupción. La rutina esta desglosada en dos partes, la figura 2.12.a muestra la parte donde se realiza la captura de datos desde el receptor GPS para luego ser almacenados en la memoria externa.

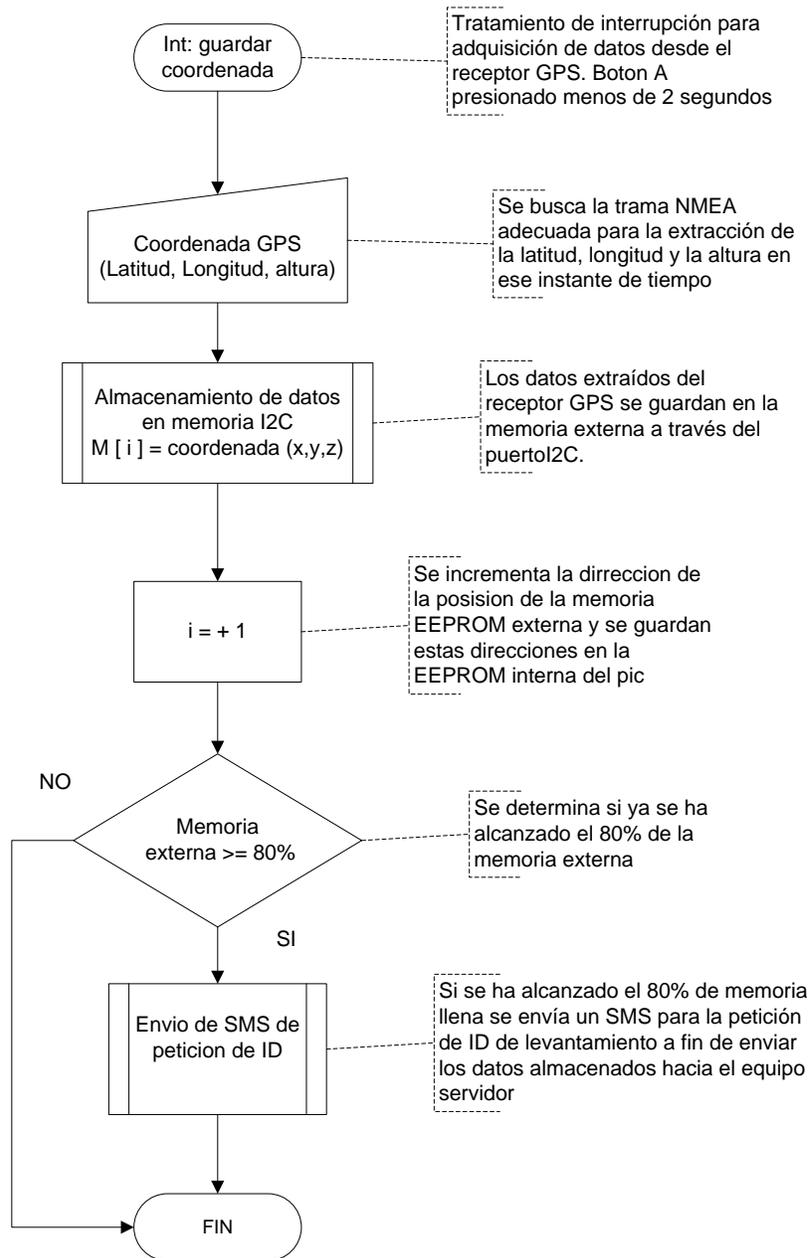
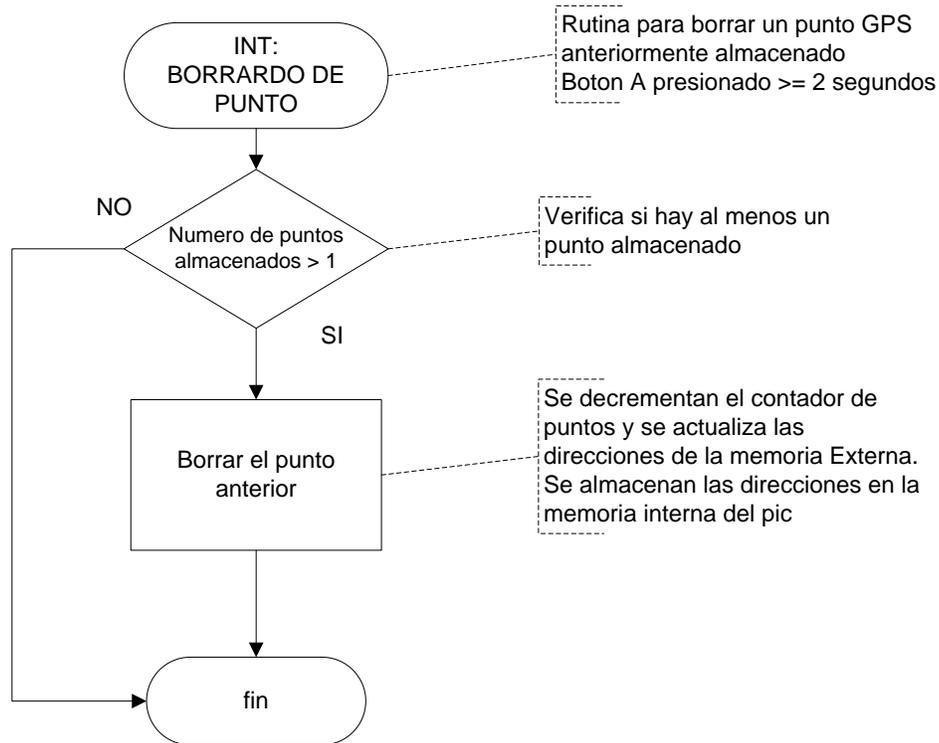


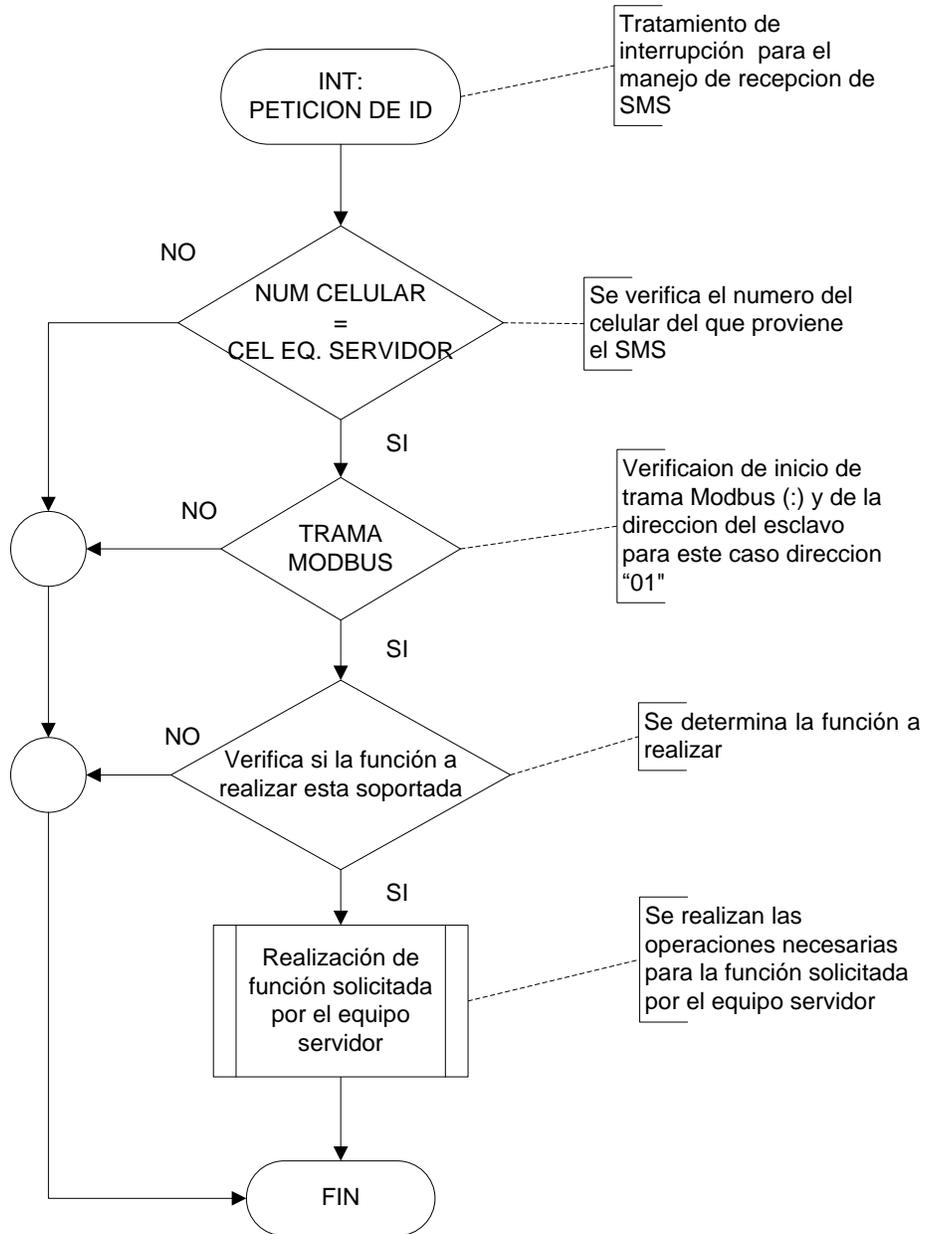
Figura 2. 12-a Diagrama de flujo de la rutina de adquisición de datos georeferenciados.

La figura 12.2.b muestra la parte de la rutina donde se borra un punto georeferenciado que ha sido almacenado previamente, y que por alguna razon es necesario eliminar, la selección de una parte de la rutina o de la otra es cuando el usuario presiona el botón “A”. el botón “A” presionado durante un tiempo menor a dos segundos, hace que se entre a la parte de almacenamiento de datos. El botón presionado durante un tiempo igual o mayor a dos segundos hace que se entre a la parte de borrado de un punto anteriormente almacenado.



**Figura 2. 12-b Diagrama de flujo de la parte de la rutina para borrar un punto almacenado.**

La figura 2.13 nos muestra el diagrama de flujo de la rutina de recepción de SMS por parte del equipo Cliente, esta verifica si el mensajito recibido es del número telefónico correcto, este es el número de teléfono que posee el Equipo Servidor, después verifica si el mensaje recibido posee las características de la trama MODBUS como el inicio de trama y la dirección del que proviene el mensaje, después verifica que si la función que posee el mensaje esta soportada por el Equipo Cliente, si lo está realiza la función sino se sale de la rutina de interrupción.



**Figura 2. 13 Diagrama de flujo de rutina de recepción de mensajitos.**

## 2.7. Conclusiones del capítulo.

- ✓ Se desarrolla la captura de coordenadas georeferenciadas desde un receptor GPS, al escanear las tramas NMEA que el receptor GPS coloca en su puerto. Esto se realiza por medio de un puerto USART virtual en el microcontrolador a 4800 baudios.
- ✓ Se guardan los datos extraídos en una memoria EEPROM externa, a través del puerto I2C, para su posterior lectura en el envío de las coordenadas georeferenciadas.
- ✓ Se manipula un teléfono celular por medio de los comandos AT, por medio del puerto USART que posee el microcontrolador a una tasa de 9600 baudios, para el envío de mensajitos de texto (SMS) los cuales van codificados en formato MODBUS.

## CAPITULO III

### DISEÑO E IMPLEMENTACION DEL SISTEMA SERVIDOR

#### 3.1. Introducción.

El tercer capítulo muestra el funcionamiento y desarrollo de la parte que se encuentra en oficina, el cual consiste en un teléfono celular, que por medio de este se reciben los datos que fueron tomados para realizar el levantamiento topográfico. La otra parte de este equipo es un ordenador que presta servicio Web para que se accedan los datos del levantamiento y se ejecute el software necesario para la creación de los archivos necesarios en la ejecución de este proyecto.

#### 3.2. Arquitectura del diseño de la solución al lado de la oficina.

El Equipo Servidor es la parte del sistema donde tiene la menor cantidad de hardware a utilizar, y posee un teléfono celular y también un ordenador en donde está instalado el software requerido para el correcto funcionamiento del proyecto. La representación del equipo servidor lo podemos observar en la figura 3.1.

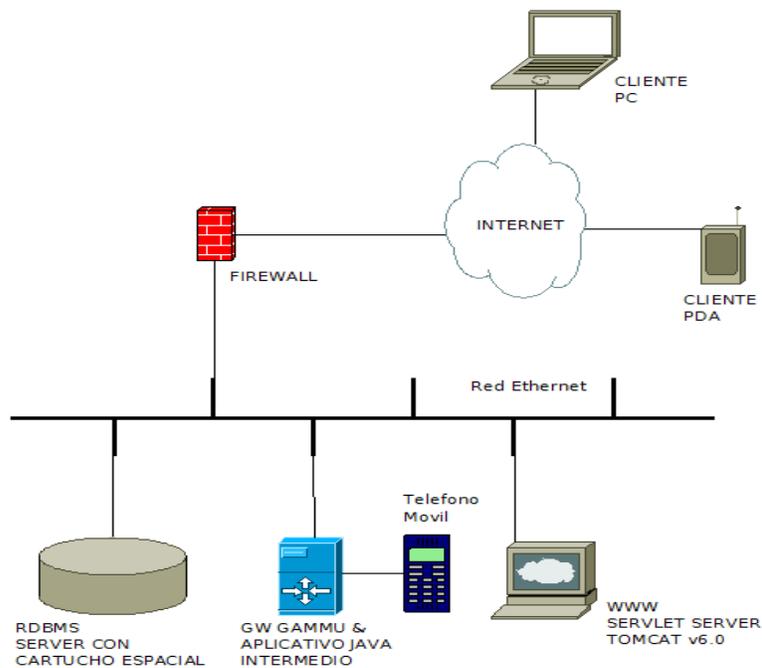


Figura 3. 1 Representación del equipo Servidor.

### **3.1.1. Criterios para el teléfono celular.**

La comunicación entre el Equipo Servidor y el Equipo Cliente es por medio de mensajes de texto por lo cual el uso de un teléfono celular es indispensable al lado del Equipo Servidor. Hoy en día todos los teléfonos celulares poseen la característica de recibir y enviar mensajes de texto; lo que deja solo la opción de buscar teléfonos que tengan compatibilidad con el programa residente GAMMU.

Para la selección de un teléfono celular acorde al sistema implementado se debe de considerar lo siguiente:

- El teléfono celular debe de soportarlo por el proyecto GAMMU, esto es primordial para la adquisición de teléfono celular, el cual también debe de poseer la característica que el se puedan enviar y recibir mensajes de texto por medio del aplicativo GAMMU.
  
- El conector de la carga de batería puede ser independiente del puerto de datos del teléfono, si la comunicación entre el teléfono celular y el ordenador es por un medio inalámbrico.
  
- El conector de la carga de batería del teléfono celular puede ser el mismo que el conector para datos, como en el caso de los teléfonos con tecnología USB, si es que se tiene una comunicación por medio de este puerto.

En la documentación de Gammu se puede obtener una infinidad de marcas y modelos de teléfonos celulares que soportan la compatibilidad con dicho programa. Una de las características que se requieren para el establecimiento de la comunicación entre el teléfono celular y Gammu es la configuración en Gammu debe de ser a una determinada velocidad y configuración, dependiendo del modelo de celular.

En el mercado se pueden encontrar diversas marcas y modelos de teléfonos de los cuales que tienen compatibilidad con el programa Gammu entre los cuales podemos observar en la figura 3.2:

Alcatel (17)	Apple (1)	BenQ-Siemens (10)	Elson (1)
Emgeton (0)	Falcom (4)	Fly (1)	Gionee (1)
HTC (14)	Huawei (15)	Hughes (1)	Jinpeng (0)
Lenovo (1)	LG (53)	Mitsubishi (2)	Motorola (92)
Nokia (257)	Onda (0)	Openmoko (1)	Option (7)
PalmOne (1)	Philips (4)	Sagem (37)	Samsung (136)
Sanyo (0)	SciPhone (7)	Sharp (6)	Siemens (56)
Sierra Wireless (5)	Simcom (2)	Sony Ericsson (170)	Teltonika (1)
Toshiba (1)	Vodafone (0)	Wavecom (5)	ZTE (4)

**Figura 3. 2 Marcas de teléfonos soportados por Gammu.**

El numero que aparece a la par de la arca del teléfono es la cantidad de modelos soportados por Gammu. Estas son las marcas de teléfonos soportados por Gammu, aunque no todos los modelos soportan el envío y recepción de SMS. La marca que mas modelos soportan el programa Gammu es Nokia, por ser el fabricante que con el cual se inicio el proyecto Gammu.

El teléfono seleccionado para el proyecto es un celular marca Motorola y modelo K1 el cual soporta el envío y recepción de SMS desde el proyecto GAMMU. Además este modelo posee un puerto USB por el cual se realiza la conexión con el ordenador y también posee conectividad bluetooth para realizar futuras mejoras al establecer una comunicación inalámbrica y la conexión para la carga de batería del teléfono celular se realizaría por medio del puerto USB que este dispositivo posee. Si es posible de poseer un Sim Card de la misma compañía que el del teléfono del Equipo Cliente, aunque no es indispensable este requerimiento.

### **3.1.2. Criterios para la selección del ordenador.**

La selección de un ordenador para el proyecto no requiere de especificación que no posee cualquier ordenador en venta en el mercado salvadoreño.

Lo que se requiere es un ordenador donde se encuentren instalados los programas necesarios como: la maquina virtual de java, la base de datos PostgreSQL junto con el repositorio Postgis, JDK, Tomcat, entre otros.

### **3.3. Requerimientos de software.**

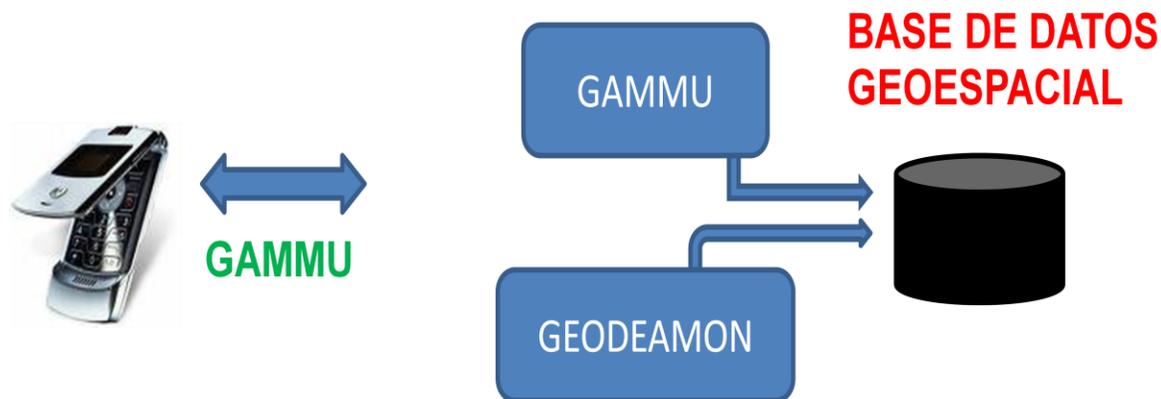
El software necesario para la ejecución del proyecto por parte del Equipo Servidor en el sistema Operativo Linux Ubuntu 9.04 es:

- ✓ Conexión entre el teléfono celular y el ordenador: el programa Gammu corriendo como servicio.
- ✓ Base de Datos: la base de datos PostgreSQL con la extensión Postgis, para el almacenamiento de los datos de los levantamientos. Las tablas principales que se tienen son: la tabla llamada 'INBOX' que es donde Gammu almacena los SMS que lee del teléfono celular; la tabla 'OUTBOX' que es donde se deben de colocar los SMS que se envían hacia el teléfono destinatario; la tabla 'coordenadasgeo' es donde están almacenados las coordenadas espaciales (latitud, longitud y altura); una tabla adicional es la tabla 'coordenadas', esta tabla almacena la misma información de la tabla anterior solo que no en forma espacial.
- ✓ La maquina virtual de java y JDK, para la compilación y ejecución de un programa intermedio para el gestiona miento entre el programa Gammu, la base de datos y la pagina Web.
- ✓ El servidor de aplicaciones web Tomcat, para la ejecución del aplicativo Web diseñado en n capas.

### **3.4. Funcionamiento del Equipo Servidor.**

La forma en que interactúan los diferentes programas dentro del ordenador debe ser en forma automática para que intervenga el usuario lo menos posible y asegurar la menor perdida de datos que se pueda dar. Al correr programas como servicio (daemon) se garantiza la poca intervención del usuario del ordenador en la ejecución de los programas residentes en memoria.

El programa Gammu puede trabajar en modo monitor que está constantemente revisando el teléfono celular pero a nivel que interactúa con el usuario, en cambio en modo servicio, no se observa cuando se está interactuando con el teléfono celular solo se observan los resultados en la base de datos, al guardar un SMS. En la figura 3.3 Se observan los programas instalados en el ordenador en forma de módulos y el flujo de información.



**Figura 3. 3 Representación de cómo interactúa el software en el equipo servidor.**

Al llegar un SMS al teléfono celular, Gammu toma ese SMS y lo coloca en la base de datos en la tabla “inbox”, con los parámetros adecuados, como la fecha y hora de llegada del SMS, el número de celular del que proviene el SMS, entre otros.

Un ejemplo de un mensaje de texto que se extrae del teléfono celular y el cual se almacena en la base de datos es el que se muestra a continuación:

```
:010104000113394299A089109247A7872B13394254A089109232A7774B13394217A0891092
24A7735B13394202A089109217A7685B13394189A089109214A7662B0D
```

Este mensaje de texto posee las características de la trama Modbus, con su cabecera, datos y checksum. Este mensaje es el que el Equipo Cliente ha enviado con las coordenadas georeferenciadas de un levantamiento que ha terminado de tomar las mediciones en campo y las envió con la solicitud de ID previamente realizado.

Los primeros dos caracteres después de los dos puntos de inicio de la trama Modbus son la dirección del cual proviene el mensaje en este caso es “01”; los segundos dos caracteres son la función a realizar por parte del Equipo Servidor, el cual en este ejemplo es “01”; los siguientes caracteres son los datos en la trama Modbus, para el caso son las coordenadas georeferenciadas de un levantamiento topográfico realizado. Antes de colocar las coordenadas, en el campo de datos se coloca el número de ID de levantamiento que ha sido solicitado anteriormente, para el ejemplo anterior es el ID de levantamiento “04”; luego se coloca el número de mensaje de texto de ese levantamiento, por si son varios mensajes de texto para un levantamiento, en el ejemplo aparece en número “0001”; y por último, siempre en el campo de datos, se encuentran las coordenadas del levantamiento realizado; el último campo es el checksum de la trama Modbus, que son los dos últimos caracteres del mensaje de texto. Cabe mencionar que existen dos

caracteres adicionales en este mensaje de texto, los cuales no son apreciados en el mensaje del ejemplo anterior; estos caracteres son el retorno de carro y salto de línea, que son los caracteres de fin de trama Modbus.

El programa intermedio, que se le ha llamado “Geodeamon” se encarga de desglosar el SMS que se ha recibido, el cual determina si proviene del Equipo Cliente. Todos los mensajes de texto que recibe el teléfono celular son almacenados en la base de datos, incluyendo los mensajes que la compañía telefónica envía los cuales pueden ser de notificación o de promociones; estos mensajes que no provienen del Equipo Cliente pueden ser filtrados en la configuración de GAMMU, pero se ha dejado en esta condición para efectos de demostración en la recepción de mensajes y la posterior decodificación.

Si el SMS viene del Equipo Cliente se revisa el mensaje de texto si posee la cabecera MODBUS respectiva, sino se toma este mensaje como invalido y solo se actualiza la base de datos en la tabla “inbox” como que ha sido revisado al colocar en la columna “proceded” el término “true”; cuando se posee la cabecera MODBUS y sigue revisando el SMS se determina la función que el equipo Cliente pide realizar.

En el capítulo 2 en la tabla 2.2 apreciamos la explicación de las funciones que el Equipo Servidor debe realizar cuando se están revisando los SMS que provienen del Equipo Cliente, esta función es parte de la trama MODBUS.

Cuando se hace una petición de ID de levantamiento por parte del Equipo Cliente, función 00, el aplicativo Geodeamon accede a la base de datos y determina cual es el numero de levantamiento mayor que se encuentra almacenado y guarda un SMS en la tabla Outbox con el numero de levantamiento máximo, incrementado en uno, para que el Equipo Cliente lo coloque al enviar las correspondiente coordenadas y así identificar el levantamiento realizado, en cada mensajito enviado por parte del Equipo Cliente debe de traer este identificador, con esto logramos agrupar todas las coordenadas del levantamiento realizado. El mensaje de texto con función 00, viene con un grupo de caracteres ASCII, el cual es una serie de “0” los cuales también son revisados por el aplicativo Geodeamon. El mensaje de contestación del Equipo Servidor lleva una serie de caracteres “1” para que el Equipo Servidor revise y valide este mensaje, además de la validación que se realiza por medio de Modbus.

Cuando se está ejecutando el aplicativo Geopolygon y se colocan las impresiones en pantalla, para efectos de que el programador verifique el estado del aplicativo se tiene lo siguiente, en un mensaje de petición de ID:

```
los SMS han sido procesados
el cell recibido:      +50372814168
el cell esperado:     +50372814168
el id de la fila es:   39
la fila que reviso es: 39
Hay datos sin procesar!!!
Entro
```



```
bytes de dato a enviar: 49
bytes de dato a enviar: 66
bytes de dato a enviar: 56
bytes de dato a enviar: 13
bytes de dato a enviar: 10
mande el sms de peticion
```

El mensaje enviado aparece en la primera fila el cual lleva el número de ID de levantamiento como uno de sus parámetros, para el caso es el ID “03”. Lo siguiente es la codificación en formato ASCII del mensaje enviado, lo interesante son las ultima líneas, en las cuales podemos apreciar el fin de la trama modbus, que son los caracteres de fin de línea (ASCII 0x13) y retorno de carro (ASCII 0x10), los cuales no son apreciados en los mensajes de texto, pero siempre van incluidos en el mensaje (SMS).

La función 01 indica que es un SMS con las coordenadas de un levantamiento, las cuales deben de extraerse del mensajito y adecuarse para ser almacenadas en la base de datos, en sus correspondiente tablas, además de realizar los archivos: render (imagen) y kml (archivo utilizado por Google Earth). El aplicativo Geodeamon es el encargado de esta tarea al desglosar el mensaje de texto; primeramente verifica el campo Checksum de la trama Modbus el cual debe de coincidir con el que trae el mensaje de texto, al coincidir este campo pasa a extraer las coordenadas del levantamiento separándolos en puntos georeferenciados, cada punto georeferenciado posee latitud longitud y altura, estos son separados; un punto georeferenciado está provisto por las tres coordenadas espaciales X, Y, Z que son la latitud longitud y la altura las cuales vienen en el mensaje de texto, de donde sede extraer en forma adecuada, porque no vienen con el formato requerido sino como una serie de caracteres ASCII.

La latitud y longitud tiene que ser separada en grados y minutos, la altura es una sola cifra pero se deben de poner en forma de decimales. Un ejemplo de cómo viene una coordenada georeferenciada es de la siguiente manera:

```
13394299A089109247A7872B
```

La latitud llega hasta el primer carácter ASCII “A” la longitud está entre los dos caracteres “A” y la altura está entre el segundo carácter “A” y el carácter “B”. la separación de la latitud se realiza tomando los primeros dos caracteres, los cuales son los grados de esta coordenada; los siguientes caracteres son los minutos de la latitud los cuales se le debe de colocar el punto decimal entre la posición cuatro y cinco de la serie mostrada. Al realizar las operaciones mencionadas anteriormente la latitud extraída del mensaje de texto es la siguiente:

```
13 39.4299
```

Los que se puede leer como 13 grados con 39.4299 segundos, como latitud de un punto georeferenciado de un levantamiento topográfico.

La separación y adaptación de la longitud es similar a la latitud solo que para los grados se toman tres caracteres y no dos como en la latitud; al realizar la separación y colocación del punto decimal en la posición adecuada se tiene lo siguiente:

```
89 10.9247
```

Lo cual puede leerse como 89 grados con 10.9247 grados de longitud de un punto georeferenciado del levantamiento realizado. A la altura solo debe de extraerse y colocar un punto decimal antes del último carácter, lo cual quedaría de la siguiente forma:

```
787.2
```

Esta altura son los metros sobre el nivel del mar donde se está realizando la medición. Así se realiza la extracción de las diferentes coordenadas en cada mensaje de texto.

Un ejemplo de parte de una corrida del aplicativo Geodeamon cuando encuentra un mensaje de texto que no ha sido procesado y que tiene las coordenadas georeferenciadas es el siguiente:

```
El dato es:
'13432511A089120173A7226B13432505A089120170A7199B13432500A089120
167A7168B13432497A089120165A7153B13432494A089120164A7140B1343249
1A089120163A7129B8'
en el driver
en la coneccion
voy a entrar al lazo
Latitud en grados '0' es: '13.0'
Longitud en minutos '43.2511
Minutos de latitud '0' es: '43.2511'
Latitud en grados '0' es: '89.0'
Minutos de latitud '0' es: '12.0173'
Altitud '0' es: '722.6'
voy a insertar a la base de datos
inserte en 2d
inserte en 3d en prueba geo
Latitud en grados '1' es: '13.0'
Longitud en minutos '43.2505
Minutos de latitud '1' es: '43.2505'
Latitud en grados '1' es: '89.0'
```

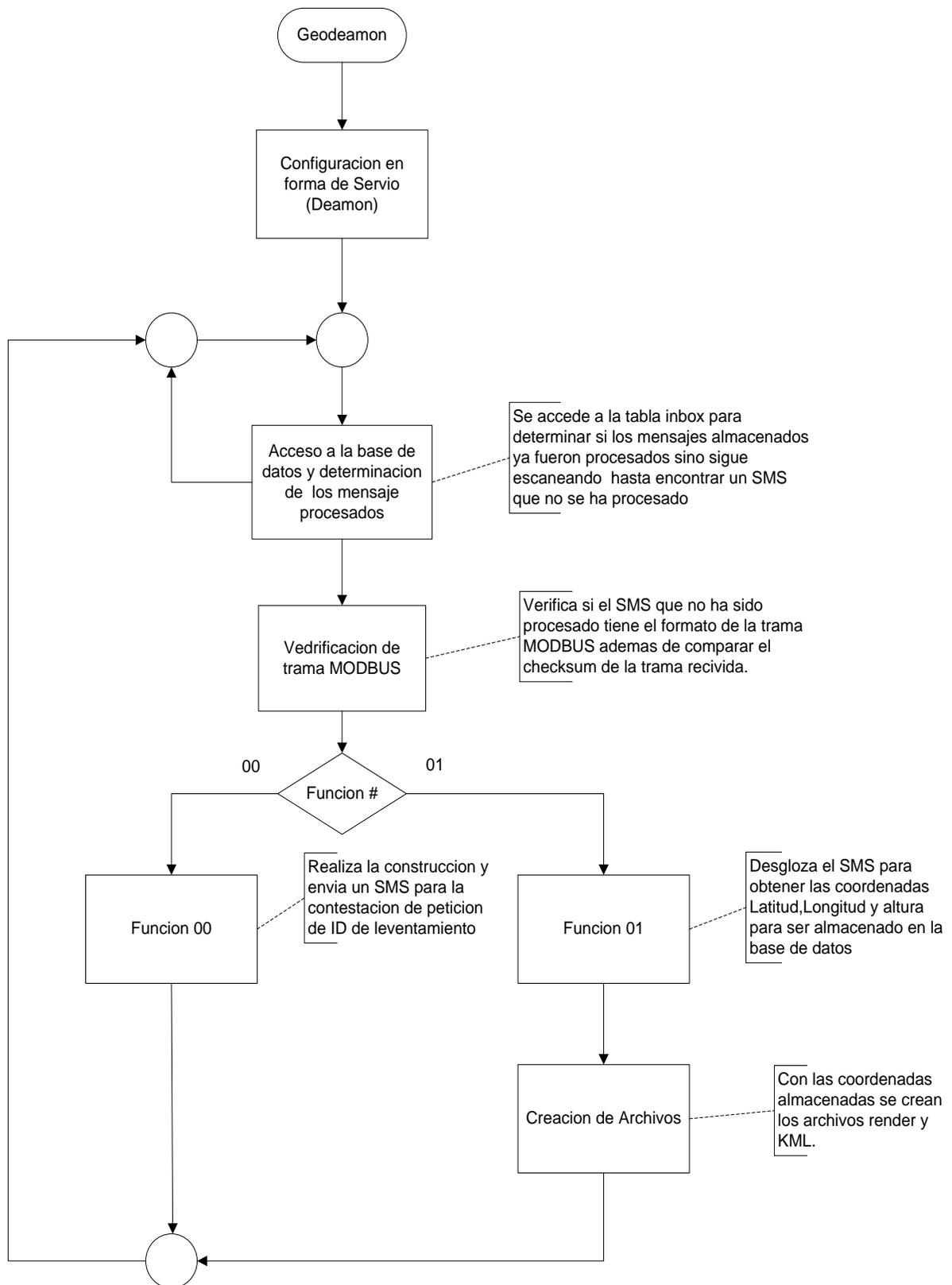
```
Minutos de latitud '1' es: '12.017'  
Altitud '1' es: '719.9'  
voy a insertar a la base de datos  
inserte en 2d  
inserte en 3d en prueba geo  
Latitud en grados '2' es: '13.0'  
Longitud en minutos '43.2500  
Minutos de latitud '2' es: '43.25'  
Latitud en grados '2' es: '89.0'  
Minutos de latitud '2' es: '12.0167'  
Altitud '2' es: '716.8'  
voy a insertar a la base de datos  
inserte en 2d  
inserte en 3d en prueba geo  
Latitud en grados '3' es: '13.0'  
Longitud en minutos '43.2497  
Minutos de latitud '3' es: '43.2497'  
Latitud en grados '3' es: '89.0'  
Minutos de latitud '3' es: '12.0165'  
Altitud '3' es: '715.3'  
voy a insertar a la base de datos  
inserte en 2d  
inserte en 3d en prueba geo  
Latitud en grados '4' es: '13.0'  
Longitud en minutos '43.2494  
Minutos de latitud '4' es: '43.2494'  
Latitud en grados '4' es: '89.0'  
Minutos de latitud '4' es: '12.0164'  
Altitud '4' es: '714.0'  
voy a insertar a la base de datos  
inserte en 2d  
inserte en 3d en prueba geo  
Latitud en grados '5' es: '13.0'  
Longitud en minutos '43.2491  
Minutos de latitud '5' es: '43.2491'  
Latitud en grados '5' es: '89.0'  
Minutos de latitud '5' es: '12.0163'  
Altitud '5' es: '712.9'  
voy a insertar a la base de datos  
inserte en 2d  
inserte en 3d en prueba geo
```

Lo primero que se aprecia es el campo de datos solo con las coordenadas georeferenciadas del levantamiento o parte de este que han sido extraídas del mensaje de texto. Luego se observa cómo se han separado las coordenadas en latitud, longitud y altura con su respectivo formato,

para luego ser almacenadas en la base de datos geoespacial, en sus respectivas tablas, como son las tablas “coordenadasgeo”, “coordenadas” y “puntosgeo”.

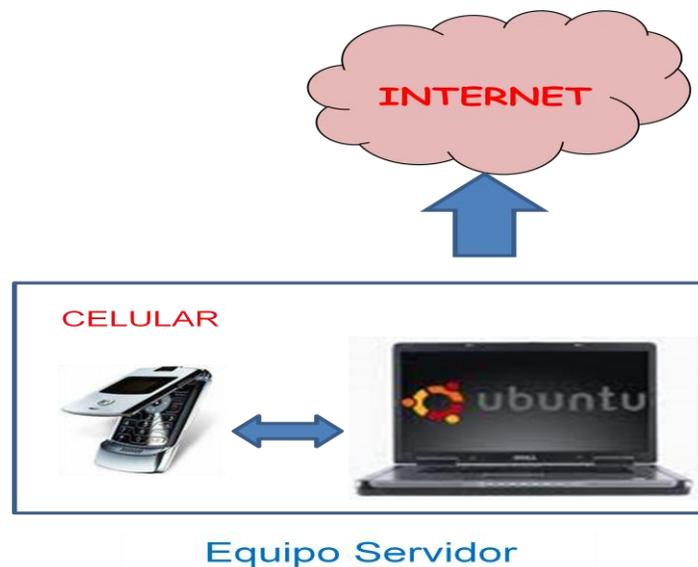
Después de esto es cuando se crean los archivos KML, esto se realiza al leer lo que está almacenado en la base de datos, para el respectivo levantamiento; esto asegura que si un mensaje de texto que pertenece a un levantamiento tarda en llegar al Equipo Servidor siempre va a construir el polígono respectivo al llegar el último mensaje de texto del levantamiento realizado.

En la función 02 no se realizan procedimiento porque es una contestación que indica que el Equipo Cliente no posee los suficiente punto almacenados para crear un levantamiento. El diagrama de flujo del aplicativo intermedio Geodeamon es el mostrado en la figura 3.4 el cual nos da la idea del funcionamiento de este.



**Figura 3. 4** Diagrama de flujo del aplicativo intermedio (Geodeamon).

La otra parte de la cual está constituido el Equipo Servidor es el despliegue de resultados en un aplicativo Web como lo vemos en la figura 3.5



**Figura 3. 5** Flujo de información del Equipo Servidor.

El aplicativo Web está constituido en base a la tecnología JAVA aplicado a una estructura en N Capas, esta estructura nos permite tener un programa que con ligeros cambios se puede desplegar los resultados en un ambiente diferente a internet, puede ser que se requieran acceder los datos localmente, lo cual se puede desarrollar.

Las clases utilizadas para la construcción de aplicativo Web están divididas en tres partes, las cuales son: la capa de negocio, la capa de datos y a capa de dominio.

En la capa de negocio se encuentran las clases: “CoordenadasNGC”, “outboxNGC” y “UrlNGC”, que son las encargadas de crear los objetos necesarios para la gestión con la base de datos. La capa de dominio se encuentran las clases: “AbstractCoordenadas”, “AbstractOutBox” y “AbstractUrl” las cuales se encargan de proveer la base de la definición de persistencia para la aplicación. Y en la capa de datos se encuentran las clases: “CoordenadasDAO”, “OutBoxDAO” y “UrIDAO” las cuales proveen la persistencia a la base de datos junto con Hibernate y dan el soporte a las entidades de la capa de negocio. El proyecto creado para desarrollar esta aplicación se encuentra en el DVD adjunto a este documento para poder apreciar los archivos java que se ejecutan el tal aplicación. Las tres capas anteriormente mencionadas se encuentran en el paquete sv.edu.ues.fia.eie.tesis.geopolygon que sirve para identificar el proyecto.

Las clases mencionadas anteriormente son las que se utilizan junto con las tablas principales de la base de datos desarrolladas en este proyecto, estas tablas son: “Coordenadas”, “OutBox”, “Url”. Otra tabla principal que utiliza el proyecto es la tabla “inbox”, que es donde se almacenan los mensajes de texto que el aplicativo GAMMU extrae desde el teléfono celular.

Además se encuentran unas tablas secundarias que se utilizan en el proyecto y que deben de existir para el correcto funcionamiento de la aplicación. Entre esta tablas están la tabla “phones” que es donde el aplicativo GAMMU tiene identificado algunos parámetros de teléfono celular, como es la intensidad de la señal percibida por el dispositivo telefónico, al igual que se tiene el porcentaje de carga de batería del teléfono móvil, entre otros; otra tabla secundaria y que es utilizada por el aplicativo GAMMU es la tabla “sentItem” que es donde se almacenan los mensajes que han sido enviados y da un reporte de cómo se efectuó el envío, si lo realizo con éxito o tubo algún error.

La parte final del aplicativo Web es la capa de presentación, que es donde se encuentran las páginas JSP y HTML que son vistas por el usuario. Las cuales dan diversas opciones y que interactúan con las capas de negocio y de dominio del sistema en cuestión.

La pantalla principal de la aplicación Web es la que se muestra en la figura 3.6 de la cual podemos apreciar las diferentes funcionalidades que posee la aplicación.



Figura 3. 6 Pagina principal del aplicativo Web.

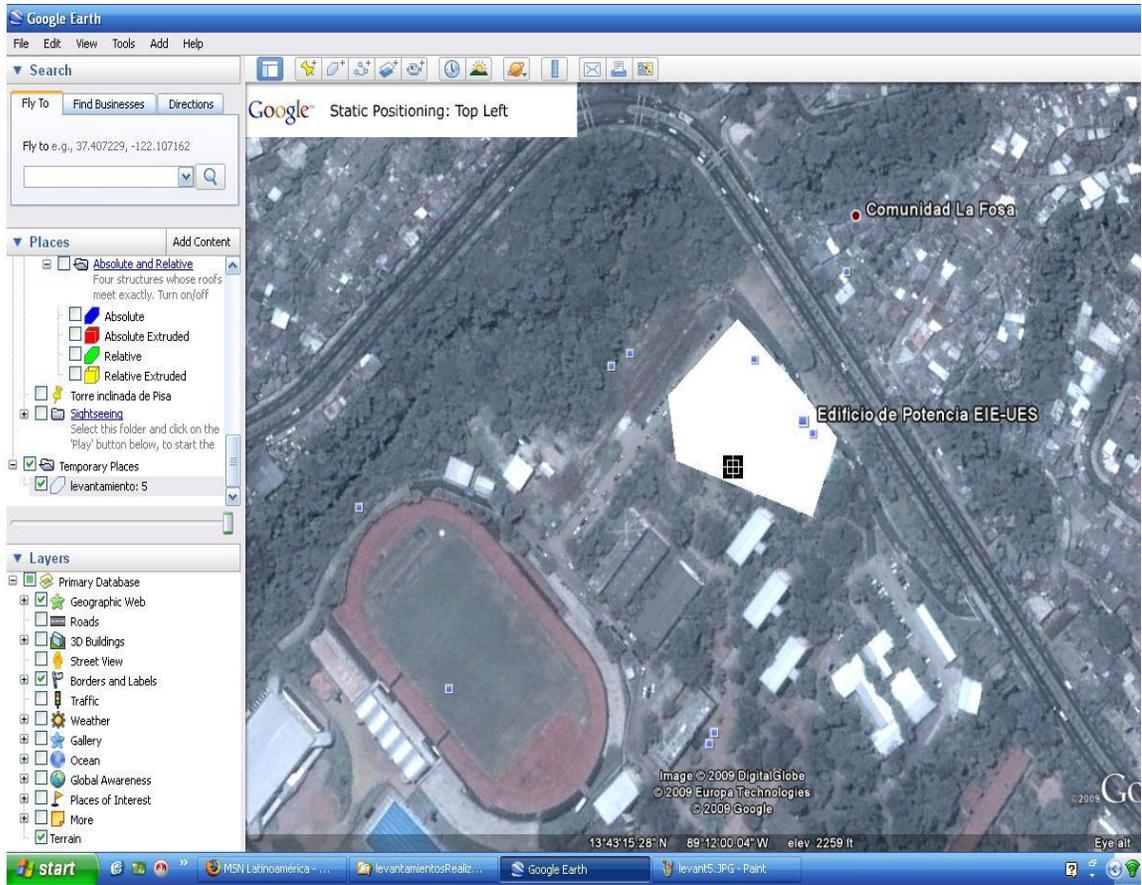
La primera funcionalidad es poder apreciar las coordenadas del levantamiento en forma de puntos georeferenciados dentro de una tabla, como se observa en la figura 3.7. Esto se realiza accediendo a la base de datos, primero para localizar el número total de levantamientos almacenados en la base de datos, lo segundo es ya el acceso a la base de datos extrayendo las coordenadas de latitud, longitud y altura que se mostraran en una página Web.



**Figura 3. 7 Pagina de resultados al observar los puntos de un levantamiento.**

La segunda funcionalidad es la descarga del archivo KML correspondiente al levantamiento realizado. El archivo KML se genero en el aplicativo intermedio. Este archivo el usuario debe de abrirlo en el Google Earth para poder ser apreciado el levantamiento. El archivo tiene lo necesario para que en forma automática Google Earth se posicione en las coordenadas necesaria para la visualización del levantamiento; si es necesario, debido a que el levantamiento es en una pequeña área, el usuario debe de manipular el programa para hacer una visualización más cerca de la superficie y así apreciar el levantamiento realizado.

Al descargar el archivo KML del levantamiento realizado y se abre en Google Earth puede apreciarse como se muestra en la figura 3.8 la cual es un levantamiento realizado alrededor de la Escuela de Ingeniería Eléctrica de la UES. En esta imagen se puede notar el error que posee el receptor GPS ocupado el cual ronda los 10 metros, pero cerca del follaje de los arboles el error incrementa, es por eso que el polígono formado no es exacto.



**Figura 3. 8 Levantamiento topográfico visualizado en Google Earth.**

### 3.5. Conclusiones del capítulo.

- ✓ Se guardan en la base de datos los mensajes en la tabla llamada “inbox”, que recibe el teléfono celular que está conectado al Equipo Servidor, la base de datos donde se almacenan los mensajes de texto recibidos es PostgreSQL por medio del servicio Gammu.
  
- ✓ El aplicativo intermedio (Geodeamon) procesa los mensajes que han sido guardados por Gammu, con anterioridad, para poder ser almacenados en forma espacial por medio del complemento PostGis en la tabla denominada “coordenadasgeo”.
  
- ✓ El aplicativo intermedio crea el archivo KML para que pueda ser publicado en el , posteriormente, en el sistema web y así ser accesible a los usuarios.
  
- ✓ El aplicativo Web permite al usuario descargar el archivo KML para que el levantamiento sea sobre puesto en el programa Google Earth.

## CONCLUSIONES.

- Se desarrollo el sistema para la toma de datos georeferenciados por medio de un receptor GPS, el cual lo realiza la parte móvil del sistema denominada “Equipo Cliente”, esta tarea es efectuada por el microcontrolador por medio de un puerto USART virtual a 4800 baudios, esta tarea es ralizada al escanear las tramas NMEA que el receptor GPS coloca en su puerto de salida, al identificar la trama correspondiente se extraen la latitud, longitud y la altura, que son las correspondiente coordenadas X,Y,Z de un punto georeferenciado.
- Se establece la comunicación inalámbrica entre las dos partes del sistema, Equipo Cliente y Equipo Servidor, por medio de dos teléfonos celulares. Esta tarea la realiza el microcontrolador por medio del puerto físico USART a 9600 baudios, en la parte del Equipo Cliente, esta comunicación se realiza utilizando los comandos AT para el control del modem interno del teléfono celular, y es así como se envían mensajes de texto a Equipo Servidor; en la parte del Equipo Servidor el teléfono celular se controla por el aplicativo GAMMU a través del puerto USB del ordenador; tanto el Equipo Cliente como el Equipo Servidor tienen la capacidad de envío de mensajes de texto y recepción de dichos mensajes.
- El envío de los datos se realiza a través de mensajitos de texto codificados con el formato de la trama MODBUS. Esta funcionalidad es para la validación de la información que está siendo enviada desde y hasta las dos partes del sistema, Equipo Servidor y Equipo Cliente. Además contribuye para la conexión de un sistema externo.
- Los datos obtenidos al realizar el levantamiento topográfico quedan almacenados en la base de datos PostgreSQL, que tiene añadido el repositorio Postgis. Esta tarea es efectuada en la parte de oficina del sistema que es el Equipo Servidor. Cuando se realiza esta operación se realiza en dos partes, la primera cuando se extraen los datos del teléfono celular que vienen en un mensaje de texto, la segunda parte es cuando se determina que se ha almacenado un mensaje de texto con las coordenadas georeferenciadas de un levantamiento, se selecciona la información necesaria para ser almacenada como datos geoespaciales
- Se crea el archivo para la visualización en forma grafica del levantamiento topográfico, por medio del aplicativo en lenguaje JAVA. El archivo creado es de Extencion KML para poder ser sobre puesto en el programa Google Earth, y así observar el levantamiento realizado y poder concluir de que se realizo una aproximación grafica del objeto físico

del cual se tomo la medición, este objeto es un terreno. El aplicativo intermedio (GeoDeamon) es el encargado de realizar el archivo cuando se tiene un mensaje guardado en la base de datos y que no ha sido procesado por el aplicativo en cuestión; el aplicativo almacena el archivo en los directorios que se pueden acceder desde el aplicativo Web y guarda la URL en una tabla de la base de datos.

- El usuario puede acceder a los datos de los levantamientos por medio del aplicativo Web que está desarrollado en base al sistema de N capas. Este aplicativo accesa a la base de datos para extraer los archivos KML correspondientes a ser descargados o para que se puedan ver los puntos georeferenciados, en forma de listas, en una página JSP.

## REFERENCIAS BIBLIOGRAFICAS

- [www.wikipedia.com](http://www.wikipedia.com)
  
- [www.monografias.com](http://www.monografias.com)
  
- [www.gammu.org](http://www.gammu.org)
- [www.modbus.org](http://www.modbus.org)
- [www.modicon.com](http://www.modicon.com)
- [www.itu.int/rec/dologin\\_pub.asp?lang=e&id=T-REC-V.250-200307-I!!PDF-E&type=items](http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-V.250-200307-I!!PDF-E&type=items)
- [www.ctiforum.com/standard/standard/etsi/0705.pdf](http://www.ctiforum.com/standard/standard/etsi/0705.pdf)
- 
  
- Curso básico de topografía  
Fernando García Marques  
Editorial Pax México, 2003

# ANEXOS

## A. DATASHETT de dispositivos utilizados.

### A.1. Microcontrolador PIC 16F877



# PIC16F87X

## 28/40-Pin 8-Bit CMOS FLASH Microcontrollers

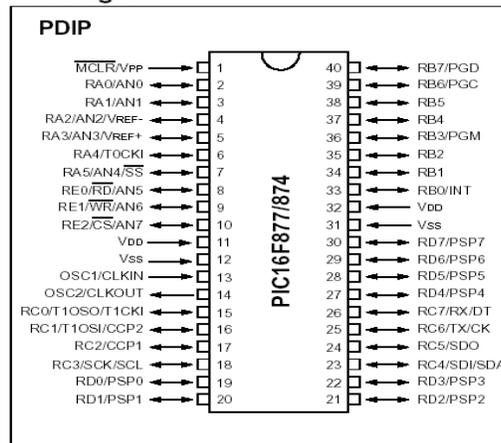
### Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

### Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,  
Up to 368 x 8 bytes of Data Memory (RAM)  
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and  
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC  
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM  
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two  
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature  
ranges
- Low-power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

### Pin Diagram



### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,  
can be incremented during SLEEP via external  
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period  
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - 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 multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master  
mode) and I<sup>2</sup>C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver  
Transmitter (USART/SCI) with 9-bit address  
detection
- Parallel Slave Port (PSP) 8-bits wide, with  
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for  
Brown-out Reset (BOR)

## A.2. La memoria EEPROM 24LC512



# MICROCHIP 24AA512/24LC512/24FC512

## 512K I<sup>2</sup>C™ CMOS Serial EEPROM

### Device Selection Table

Part Number	Vcc Range	Max. Clock Frequency	Temp. Ranges
24AA512	1.7-5.5V	400 kHz <sup>(1)</sup>	I
24LC512	2.5-5.5V	400 kHz	I, E
24FC512	1.7-5.5V	1 MHz <sup>(2)</sup>	I

**Note 1:** 100 kHz for Vcc < 2.5V  
**Note 2:** 400 kHz for Vcc < 2.5V

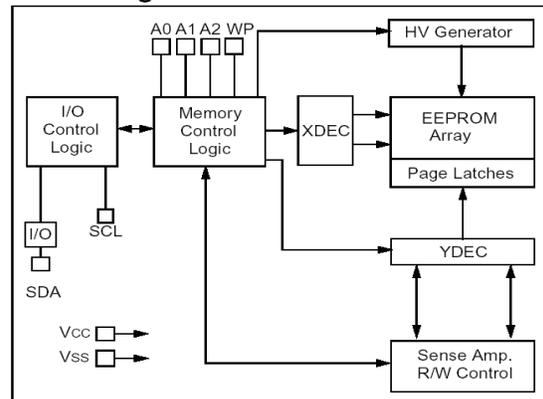
### Features:

- Single Supply with Operation Down to 1.7V for 24AA512 and 24FC512 Devices, 2.5V for 24LC512 Devices
- Low-Power CMOS Technology:
  - Active current 400 uA, typical
  - Standby current 100 nA, typical
- 2-Wire Serial Interface, I<sup>2</sup>C™ Compatible
- Cascadable for up to Eight Devices
- Schmitt Trigger Inputs for Noise Suppression
- Output Slope Control to Eliminate Ground Bounce
- 100 kHz and 400 kHz Clock Compatibility
- Page Write Time 5 ms max.
- Self-Timed Erase/Write Cycle
- 128-Byte Page Write Buffer
- Hardware Write-Protect
- ESD Protection >4000V
- More than 1 Million Erase/Write Cycles
- Data Retention > 200 years
- Packages Include 8-lead PDIP, SOIJ and DFN
- Pb-Free and RoHS Compliant
- Temperature Ranges:
  - Industrial (I): -40°C to +85°C
  - Automotive (E): -40°C to +125°C

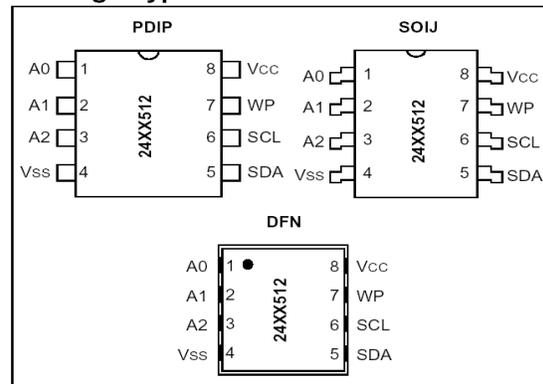
### Description:

The Microchip Technology Inc. 24AA512/24LC512/24FC512 (24XX512\*) is a 64K x 8 (512 Kbit) Serial Electrically Erasable PROM, capable of operation across a broad voltage range (1.7V to 5.5V). It has been developed for advanced, low-power applications such as personal communications and data acquisition. This device also has a page write capability of up to 128 bytes of data. This device is capable of both random and sequential reads up to the 512K boundary. Functional address lines allow up to eight devices on the same bus, for up to 4 Mbit address space. This device is available in the standard 8-pin plastic DIP, SOIJ and DFN packages.

### Block Diagram



### Package Type



\* 24XX512 is used in this document as a generic part number for the 24AA512/24LC512/24FC512 devices.

### A.3. EI MAX232

## +5V-Powered, Multichannel RS-232 Drivers/Receivers

**MAX220-MAX249**

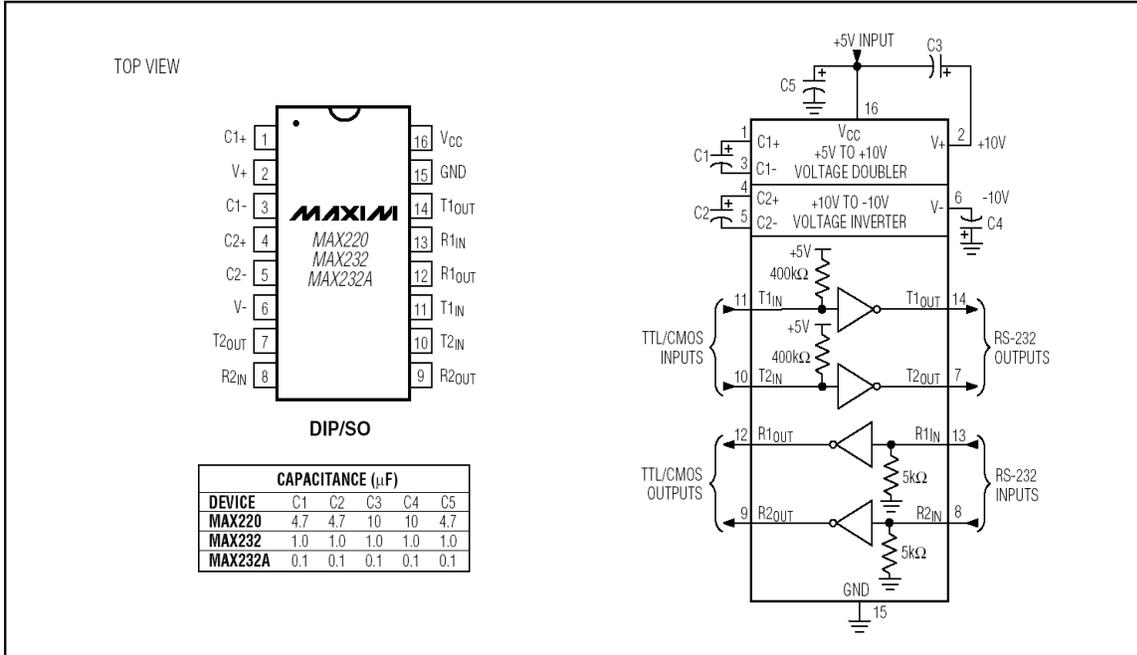


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

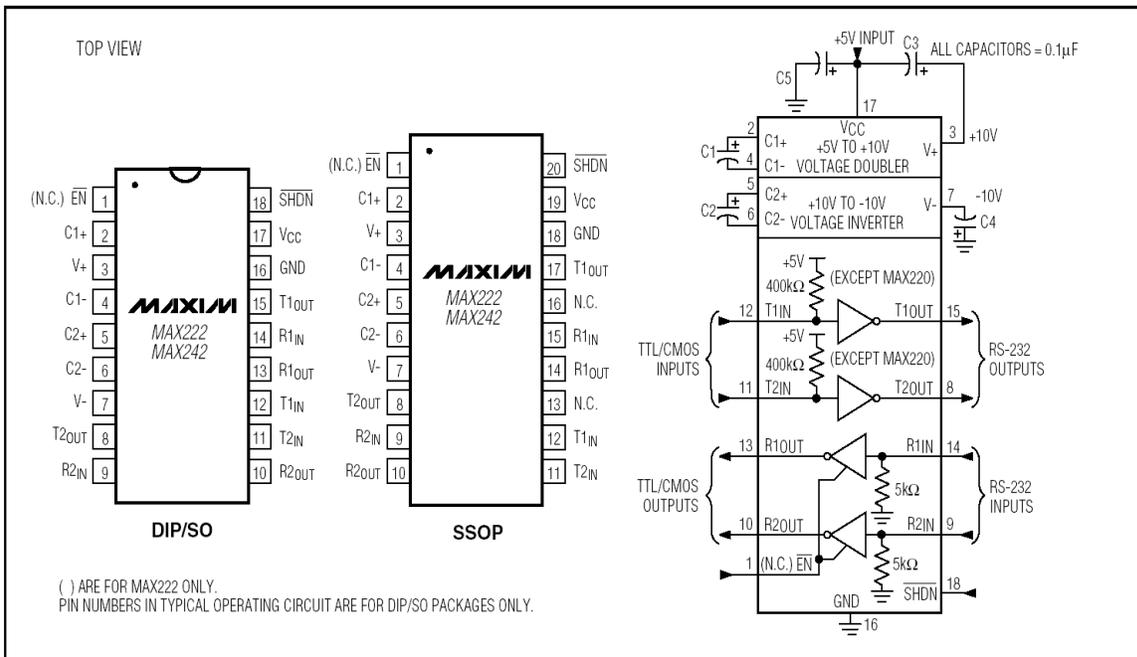
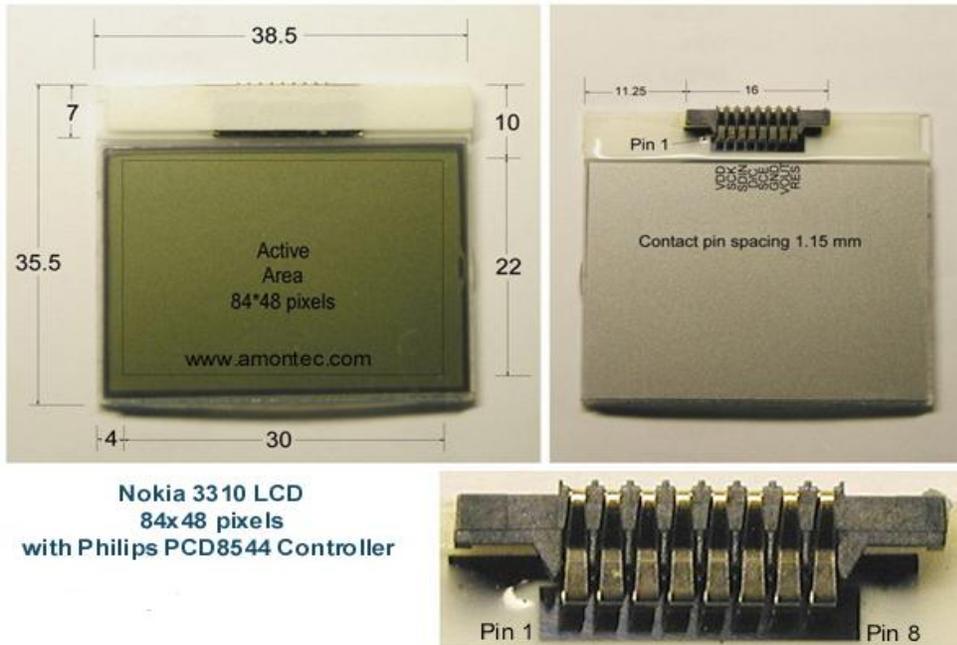


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

#### A.4. LCD de NOKIA 3310



#### A.5. El teléfono celular Motorola k1

Presentamos el nuevo teléfono inalámbrico **MOTOKRZR™ k1** GSM. Aquí le brindamos una rápida lección acerca de su anatomía.



Para conocer detalles de Push to Talk (PTT), consulte la página 17.

## A.6. El teléfono celular Nokia 3320



### Application Features

Digital rights management	Yes
Java support	J2ME
Over the air configuration (OTA)	Yes
SIM toolkit release	1998

### Browsing

WAP	Yes
WAP version	2
WTLS key encryption	1024
WTLS support	Class 3

### Communication Classes

EDGE	Yes
Extended GSM (EGSM)	Yes
GSM bands	GSM 900/1800/1900

### GPRS

GPRS	Yes
GPRS PBCCH	Yes
GPRS coding scheme	CS1 CS2 CS3 CS4
GPRS functionality	Class B
GPRS timeslots	Class 6 (3+2)

### Data Connectivity

Data & fax	Data only
Data cable	Available
USB port support	Yes

### PIM Synchronisation

Calendar Synchronisation	MS Outlook and Lotus Notes
Contact Synchronisation	MS Outlook and Lotus Notes

### Dimensions

Size depth (mm)	18.7
Size length (mm)	104.5
Size width (mm)	44.2
Weight with std batt (g)	86

### Display & Imaging

#### Camera

Camera option	Integrated
Integrated camera res X	640
Integrated camera res Y	480
Integrated video camera res X	128
Integrated video camera res Y	96
Video capture	Yes

#### Primary Display

Colour depth	65536
Colour type	Colour
Resolution X	128
Resolution Y	128
Viewable size X (mm)	27.69
Viewable size Y (mm)	27.69

#### Environmental

SAR @ 10g avg (W/kg)	0.78
----------------------	------

### Key Device Features

Calculator	Yes
Carkit option	Portable or fixed
Changeable covers	Front & back
Clock	Alarm clock
Download tones/pics	Ringtones & pictures
Event organiser	Calendar & organiser
Games	Pre-installed & downloadable
Handsfree	Integrated + headset incl.
Rapid text input	T9 standard
Ringtone complexity	Polyphonic
Timers	Stopwatch & countdown
Vibrate alert	Yes
Voice activation	Vox dial & command

### Messaging

Concatenated SMS	Yes
E-mail client	POP3 & IMAP4
SMTP authentication	Yes

### MMS

MMS	Yes
MMS download notify	Yes
Message size max rx (Kb)	100
Message size max tx (Kb)	100

### Audio

AMR format	Send & receive
MIDI format	Send & receive
MP3 format	No
SP-MIDI format	Receive only
WAV format	No
WMD format	No
i-melody format	No

### Image

BMP display oversize	Resize
BMP format	Send & receive
GIF animation	Loop supported

## B. Manual de usuario.

### B.1. Instalación de software.

Todo el software a instalar es el utilizado por el Equipo Servidor, el cual consiste en tener instalado la base de datos Postgre con la extensión Postgis, el servicio de monitoreo Gammu, la maquina virtual de java, JDK, Tomcat 6, además de los IDE's necesario: Myeclipse o en su defecto Eclipse. Los cuales se encuentran en el DVD del proyecto.

Para la instalación de Gammu dentro del sistema operativo Linux, específicamente la versión de Ubuntu 9.04, se puede realizar desde el Gestor de Paquetes Sinactic o desde la consola con el comando:

```
$ sudo install apt-get nombre_paquete_instalar
```

Después de instalar Gammu, se copian en la carpeta “/etc” los archivos “smsdrc” y “gammurc”, que son los archivos de configuración de Gammu. Estos archivos se encuentran en el DVD del proyecto de tesis.

También se debe de instalar la base de datos Postgre creando la base de datos “sms” con usuario “tesis” y clave “admin” como se muestra a continuación:

```
postgres@roberto-laptop:/home/roberto$ createuser
Ingrese el nombre del rol a agregar: tesis
¿Será el nuevo rol un superusuario? (s/n) s
postgres@roberto-laptop:/home/roberto$ psql
Bienvenido a psql 8.3.7, la terminal interactiva de PostgreSQL.
```

```
Teclee: \copyright para ver los términos de distribución
        \h para ayuda sobre órdenes SQL
        \? para ayuda sobre órdenes psql
        \g o punto y coma (<<,>>) para ejecutar la consulta
        \q para salir
```

```
postgres=# alter user tesis with password 'admin';
ALTER ROLE
postgres=#
```

Para la instalación de Postgis por consola es:

```
sudo apt-get install postgresql-8.3-postgis
```

Con eso ya tenemos instalado el PostGIS.

Nos hace falta crear ahora el *template\_postgis*. Nos logeamos en PostgreSQL al *template1*:

```
sudo -u postgres psql template1
```

Creamos una base de datos, *template\_postgis*, a partir de este template:

```
create database template_postgis with template = template1;
```

Actualizamos la tabla *pg\_database* para indicarle que la nueva base de datos es un template:

```
UPDATE pg_database SET datistemplate = TRUE where datname = 'template_postgis';
```

Nos conectamos a la nueva base de datos:

```
\c template_postgis
```

Agregamos las extensiones de PostGIS y les damos acceso a todos, a las tablas espaciales:

```
CREATE LANGUAGE plpgsql;
\i /usr/share/postgresql-8.3-postgis/lwpostgis.sql;
\i /usr/share/postgresql-8.3-postgis/spatial_ref_sys.sql;
GRANT ALL ON geometry_columns TO PUBLIC;
GRANT ALL ON spatial_ref_sys TO PUBLIC;
```

Restringimos futuros cambios en la base de datos:

```
VACUUM FREEZE;
```

Y con esto tenemos creada la base de datos con la extensión Postgis.

Luego tenemos que crear las tablas que Gammu necesita para su funcionamiento. Estas tablas se crean del archivo “*pgsql.sql*”, lo cual se realiza de la siguiente forma:

```
postgres@roberto-laptop:/etc$ psql -d sms -f psql.sql
```

una vez instalados todos los paquetes necesarios, el sistema debe de funcionar y, se procede a realizar el Levantamiento Topográfico.

## **B.2. Utilización del Equipo Cliente.**

La forma en cómo interactúa el Equipo Cliente o móvil con el usuario es a través de la pantalla LCD y de los diferentes pulsadores que posee. La tabla B.1 nos da a conocer las diferentes acciones que se tienen al presionar cada pulsador.

Antes de empezar a obtener los datos del levantamiento se debe de configurar el receptor GPS para la comunicación NMEA en el puerto de salida. Esto se realiza al desplazarnos por las

diferentes pantallas de receptor GPS y en la pantalla principal de menú, se accede al menú de “interface” y allí se setea el receptor GPS con salida NMEA.

**Tabla B.2 funciones de pulsadores del Equipo Cliente.**

Interruptor/Pulsador	FUNCION
ON/OFF	Enciende y apaga el Equipo Cliente.
A	Toma un punto georeferenciado del levantamiento.
A (mantenerlo presionado)	Al mantenerlo presionado durante unos segundos borra el punto anterior del levantamiento.
A (inicio)	Al encender el Equipo Cliente, se continúa con el levantamiento y no se pierden los puntos anteriores.
B	Envía un SMS para petición de ID de levantamiento.
B (inicio)	Al encender el Equipo Cliente, borra todos los puntos guardados.
RESET	Reset general del sistema.

Al encender el Equipo Cliente, el equipo espera que el usuario decida si quiere seguir realizando el levantamiento anterior o realizar un nuevo levantamiento, como se observa en la figura B.1. Si el usuario decide que va a continuar con el levantamiento anterior debe de presionar el pulsador “A” y enseguida continuará realizando el levantamiento anterior. Si el usuario decide de realizar un nuevo levantamiento debe de presionar el pulsador “B”, y se borrarán los datos del levantamiento anterior. En la figura B.2 se observa al Equipo Cliente listo para la toma de datos para realizar un levantamiento topográfico. En realidad solo se borran las direcciones de donde se encuentran los datos almacenados de la memoria externa.



**Figura B.1 Equipo Cliente al ser encendido.**

Los datos del levantamiento se recolectan al presionar el pulsador “A”, hasta terminar de realizar el levantamiento. Una vez terminado el levantamiento se presiona el pulsador “B” y se envía un mensaje de texto con la petición de ID de levantamiento, hacia el Equipo Servidor.



**Figura B.2 Equipo Cliente listo para la toma de datos.**

Cuando el Equipo Servidor contesta enviando el ID de levantamiento, el Equipo Cliente envía los datos que tiene almacenados en la memoria EEPROM por medio de mensajes de texto codificados en MODBUS.

La forma en cómo el Equipo Cliente envía el mensaje de petición de ID es en la forma siguiente:

```
ATZ
OK
AT+CMGF=1
OK
AT+CNMI=1,2,0,0,0
OK

+CMT: "+50372308163", "09/11/16,15:07:57-24"
:01000911111111B2
```

Y la forma en cómo recibe el mensaje con el cual viene el ID de levantamiento es de la siguiente manera:

```
ATZ
OK
AT+CMGF=1
OK
AT+CNMI=1,2,0,0,0
OK

+CMT: "+50372308163",,"09/11/16,15:10:47-24"
:0100091111111B2
```

El Equipo Cliente al reconocer el mensaje anterior envía los puntos guardados, lo cual lo realiza de la siguiente forma:

```
AT+CMGS="72308163"
>:010104000113394299A089109247A7872B13394254A089109232A7774B13394217A
089109224A7735B13394202A089109217A7685B13394189A089109214A7662B0D
>

OK
```

### B.3. Utilización del Equipo Servidor.

Al tener instalado el software que necesita el sistema se puede hacer funcionar. Lo primero es revisar si Gammu está corriendo como un servicio sino hacerlo funcionar desde la consola, y esto se realiza de la siguiente forma: primero revisando el puerto donde está conectado el teléfono celular, el cual debe ser ACM0ñ luego se revisa si es el teléfono correcto, en este caso es un Motorola K1, para el cual se tiene la configuración en el archivo "gammurc"; luego se hace correr gammu como servicio, por algún inconveniente Gammu coloca la salida en el archivo llamado "smslog", el cual esta almacenado en la carpeta personal de Ubuntu.

```
roberto@roberto-laptop:~$ sudo su
[sudo] password for roberto:
root@roberto-laptop:/home/roberto# ls /dev/ttyACM0
/dev/ttyACM0
root@roberto-laptop:/home/roberto# gammu --identify
Manufacturer      : Motorola
Model              : unknown (K1)
Firmware           : "R4527_G_08.42.0BR_B"
IMEI               : 359412002393423
Product code      : K1
SIM IMSI           : 706030006028817
```

```

root@roberto-laptop:/home/roberto# gammu --smsd PGSQL /etc/smsdrc
Log filename is "smsdlog"
Press Ctrl+C to stop the program ...
^Croot@roberto-laptop:/home/roberto#

```

Para finalizar el programa se presiona Ctrl+C al estar corriendo de esta forma.

Se debe de poner a trabajar el Aplicativo Intermedio denominado Geodeamon como se observa en las siguientes líneas:

```

roberto@roberto-laptop:~$ java -cp ./usr/share/java/postgresql-jdbc3-8.2.jar DemonioPgSql
d.isDaemon() = true
debe de presionar ENTER para terminar!!!!!!
estoy en un lazo infinito 0
los SMS han sido procesados
estoy en un lazo infinito 1
los SMS han sido procesados

roberto@roberto-laptop:~$

```

Esto se realiza desde la carpeta donde se encuentran las clases creadas para el proyecto. Para finalizar el programa debe de presionar ENTER.

La forma en que interactúa el Aplicativo Web con el usuario se observa en la figura B.3 la cual nos muestra las diferentes opciones que puede realizar, entre estas están la de observar los puntos de un levantamiento, la descarga del archivo KML para poder ser observado en Google Earth; además de que se puede realizar la petición del levantamiento que está realizando el Equipo Cliente, si este tiene guardados puntos en su memoria.



Figura B.2 Equipo Cliente listo para la toma de datos.

## **C. Mejoras al sistema.**

El sistema posee características para realizarle mejoras tanto en hardware como en software.

### **C.1. Mejoras en el Equipo Cliente.**

- Una de las mejoras que pueden realizarse en el Equipo Cliente es la captura de varios levantamientos, sin que se envíen los datos al Equipo Servidor. Esto debe de mejorarse porque actualmente solo se posee la característica de realizar el levantamiento y luego se envían los datos del levantamiento actual.
- Otra mejora puede ser en cambio de conexión entre el microcontrolador y el teléfono celular, lo cual puede realizarse por medio de un puerto USB o por medio de tecnología Bluetooth; esto con el fin de tener un mayor número de teléfonos celulares a disposición para la comunicación inalámbrica. La comunicación USB se logra con un circuito basado en un chip FT232R, este dispositivo convierte señales serie en USB y viceversa. Un dispositivo que venden en el mercado es The BlueSnap de serialio.com para la comunicación por medio de bluetooth. Otro dispositivo para comunicación /bluetooth es EYMF2CAMM-XX que una mejor opción para desarrollar proyectos. Aunque algunas personas recomiendan los dispositivos que vende la empresa Parallax los cuales se pueden controlar por medio de comandos AT.

### **C.2. Mejoras en el Equipo Servidor.**

- Una de las mejoras que se pueden realizar en el Equipo Servidor es, la creación de archivos Vector como DWG o DXF para ser mostrados en el Aplicativo Web, con la librería Java llamada YCAD. Esta librería posee características por las cuales se pueden ver los archivos vector por medio de una página Web, esto mejoraría la presentación de resultados.

## D. Costos de elementos utilizados en la construcción del sistema.

Los costos económicos de las diferentes partes del sistema se muestran en la tabla 4.1 donde se desglosan todos los elementos de la parte móvil y la parte de oficina del sistema.

ITEM	ELEMENTO	DESCRIPCION	CANTIDAD	COSTO \$	TOTAL
<b>EQUIPO CLIENTE O MOVIL.</b>					
1	PIC16F877	microcontrolador	1	18	18
2		Memoria EEPROM de 512kBits	1	1.85	1.85
3	MAX232		1	3.75	3.75
4	XTAL	Cristal de 4 MHz	1	0.45	0.45
5	C22pF	Capacitores de 22 pF	2	0.35	0.7
6	C1 $\mu$ F	Capacitores de 1 $\mu$ F	5	0.35	1.75
7	GPS	receptor GPS marca Garmin Etrex Legend (usado)	1	85	85
8	DB9	Conector placa para RS232	2	2.15	4.3
9	LED	Diodos Led		0.3	0
10	R 300 $\Omega$	Resistencias de 300 $\Omega$	5	0.25	1.25
11	CELL	Teléfono celular marca NOKIA modelo 3220 (usado)	1	35	35
12	LCD	Pantalla LCD	1	7	7
13	C 47 pF	Capacitores de 47 p F	4	0.35	1.4
14	'7805	referencia de 5 voltios	1	0.75	0.75
15	cable usb cell	cable usb para conexión celular nokia 3220	1	9	9
16	tableta cobre	tableta de cobre dos caras	1	17.5	17.5
17	Software creado	Horas trabajo para la codificación del firmware del Equipo Cliente	1000	10	10000
<b>EQUIPO SERVIDOR O DE OFICINA.</b>					
18	CELL	Teléfono celular marca Motorola modelo K1 (usado)	1	70	70
19	Software creado	Horas trabajo para la codificación del software creado en el lado del Equipo Servidor	400	10	4000
<b>TOTAL DEL SISTEMA</b>					<b>\$ 14,257.7</b>