



**UNIVERSIDAD  
NACIONAL AUTÓNOMA  
DE MÉXICO**



Facultad de ingeniería

**Monitoreo y control remoto usando la  
red de telefonía celular GSM**

Tesis para obtener el grado de:

Ingeniero Eléctrico Electrónico

Por

Carlos Gutiérrez Mondragón

Director de Tesis

M.I. Benjamín Valera Orozco

MÉXICO, DF

2014



FACULTAD DE INGENIERIA





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



FACULTAD DE INGENIERIA



## Indice

Introducción	1
Definición del problema	1
Objetivo	1
Antecedentes	1
Descripción del problema a resolver	1
Relevancia y justificación	2
Alcance y limitaciones	2
Relación con otras áreas	2
Método	2
Inventario de materias	3
Resultados esperados	3
Resumen de la tesis	4
1. Conceptos básicos	5
1.1. Principios de telefonía móvil	5
1.1.1. Red de telefonía celular	6
1.1.2. Terminales móviles	9
1.1.3. Mensajes de texto SMS	13
1.1.4. Lenguaje de comandos “AT”	21
1.2. El sistema operativo Android	25
1.2.1. Características	25
1.2.2. Aplicaciones	26
1.2.3. Desarrollo de aplicaciones	26
1.3. Microcontrolador PIC18F97J60	34
1.4. Herramientas de desarrollo	37
1.4.1. MPLab	37
1.4.2. PIC-GSM	39
2. Descripción del proyecto	47
2.1. Esquema general	47
2.1.1. Esquema para la programación del sistema electrónico	48
2.1.2. Esquema para la depuración del programa en el sistema electrónico	48
2.2. Sistema electrónico	49
2.2.1. Etapa de control y comunicaciones	49
2.2.2. Etapas de monitoreo de señales	50
2.2.3. Etapas de control de actuadores	51
2.2.4. Algoritmo de monitoreo, control y comunicaciones	51
2.3. Software de operación	55
2.3.1. Esquema general	55
2.3.2. Descripción del software en el ámbito del operador	56
2.3.3. Descripción del software en el ámbito del programador	60
3. Resultados	73
3.1. Prueba 1	75
3.2. Prueba 2	76
3.3. Prueba 3	77
3.4. Prueba 4	77
3.5. Prueba 5	78

3.6. Discusión	80
3.7. Trabajo a futuro	80
Conclusiones	83
Referencias	85
Anexo A. Envío y recepción de un mensaje SMS	87
Anexo B. Tabla ASCII	88
Anexo C. Descripción de los pines del SIM300	92

## **Introducción**

### **Definición del problema**

Los sistemas de control con base en la tecnología GSM (*Global System for Mobile communications*, Sistema Global para las comunicaciones Móviles) constituyen un área emergente en donde se busca extender las capacidades naturales de comunicación en los dispositivos móviles hacia nuevas aplicaciones que permitan a los usuarios de esta tecnología operar aparatos o dispositivos electrónicos a distancia. Un ejemplo se puede encontrar en la domótica<sup>1</sup> y controles remotos a distancia que utilizan la red de telefonía móvil GSM para comunicar un terminal controlador instalado en un teléfono celular con dispositivos electro-domésticos o industriales. De esta manera es posible controlar salidas digitales (shwiches on/off) y obtener valores analógicos (presión, temperatura, etc.).

La industria se cuenta con diversos equipos, los cuales requieren de una continua inspección que consta de revisar valores y activar válvulas. Estas actividades son importantes porque de no hacerse puede cuásar un rápido deterioro en el equipo. En la domótica es posible controlar la iluminación y calefacción de un hogar, abrir y cerrar puertas, simular presencia para el control de mascotas y muchas aplicaciones más.

### **Objetivo**

Desarrollar hardware y software para monitorear variables y controlar dispositivos actuadores mediante una aplicación para teléfono celular usando la red de telefonía digital GSM como medio de comunicación entre el usuario y el sistema electrónico monitor y actuador.

### **Antecedentes**

El principal antecedente para el desarrollo del presente proyecto es el módulo de desarrollo PIC-GSM. Este módulo permite desarrollar aplicaciones de monitoreo y control remoto usando las redes celular GSM y Ethernet. El módulo contiene un microcontrolador PIC18F97J60, dos salidas de relevador de 240VAC/10A, dos entradas aisladas ópticamente y una entrada para sensor de temperatura.

Otro antecedente es el artículo “Implementation of ZigBee-GSM based Home Security Monitoring and Remote Control system” en donde se implementa un sistema de seguridad y control para casas habitación utilizando mensajes SMS sobre la red de telefonía GSM. Este sistema está compuesto por un microcontrolador Atmega128, un módulo GSM ZigBee y un teléfono celular Sony Ericsson T290i.

Adicionalmente, la red de telefonía celular GSM ha tenido un gran impacto en la sociedad y cuenta con un número grande de usuarios a nivel nacional.

### **Descripción del problema a resolver**

Los antecedentes descritos y nuestro interés particular por las interfaces de los sistemas electrónicos, nos conducen a proponer el desarrollo de un sistema de monitoreo y

<sup>1</sup> Conjunto de sistemas capaces de automatizar una vivienda.

control con base en mensajes SMS utilizando como medio de comunicación la red telefónica GSM entre el usuario y los dispositivos electrónicos. En nuestro concepto, una aplicación para teléfono celular del tipo “Smartphone” desarrollada en Android, ejecuta las instrucciones necesarias para acceder al sistema electrónico de forma remota mediante mensajes SMS. El sistema electrónico consiste de un microcontrolador PIC18F97J60 que se comunica con un circuito integrado encargado del envío y recepción de mensajes SMS SIM300.

### Relevancia y justificación

La relevancia del proyecto radica en la creación de infraestructura de desarrollo propio que puede ser ofrecida como una transferencia tecnológica.

### Alcance y limitaciones

El alcance del presente proyecto es la creación de una herramienta básica para el desarrollo de aplicaciones de instrumentación remota con base en mensajes SMS utilizando la red GSM.

Las limitaciones del prototipo por desarrollar radican en lo elemental de su principio de operación. Sin embargo, constituye un elemento valioso en el momento de transmitir el conocimiento básico que puede ser utilizado en aplicaciones futuras a partir de una herramienta básica.

### Relación con otras áreas

Existe una gran relación con las áreas de sistemas de comunicaciones, instrumentación virtual, programación de computadoras y microcontroladores y microprocesadores.

### Método

En este trabajo se propone desarrollar hardware y software para implementar un sistema de control remoto con base en mensajes SMS utilizando la red GSM para controlar un sistema electrónico mediante un teléfono celular. El esquema general se presenta en la figura 1.

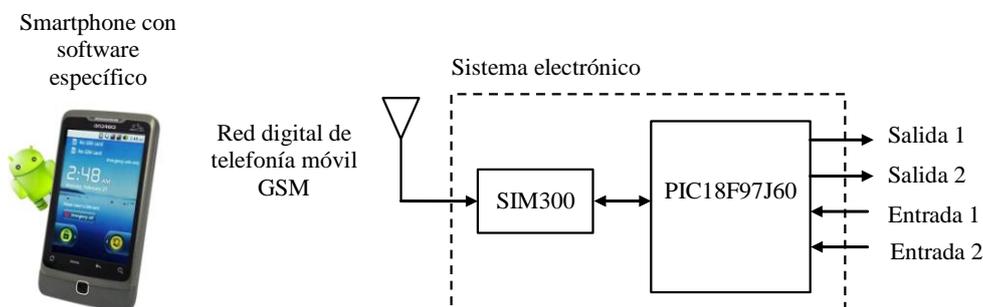


Figura 1: Método a emplear.

En el esquema de la figura 1, una aplicación para teléfono celular desarrollada en Android realiza las siguientes tareas.

- Al presionar sendos botones, genera y envía los mensajes SMS para instruir al sistema electrónico para activar o desactivar las salidas digitales 1 y 2 respectivamente. Espera la respuesta de confirmación para actualizar la interface gráfica.
- Al presionar sendos botones, genera y envía los mensajes SMS para instruir al sistema electrónico para digitalizar las entradas analógicas 1 y 2 respectivamente. Espera la respuesta de confirmación conteniendo los valores digitalizados y los despliega en pantalla.
- Al presionar opciones específicas del menú, da de alta o baja a usuarios registrados por el sistema electrónico.
- Al presionar la opción específica del menú realiza una llamada de voz para escuchar lo que sucede en la localidad del sistema electrónico.

El microcontrolador en el sistema electrónico se comunica con el circuito integrado SIM300 por medio de un enlace RS232 y comandos en protocolo AT. El circuito integrado SIM300 es un circuito integrado de radio frecuencia que soporta el enlace con la red GSM. Adicionalmente, el sistema electrónico cuenta con una memoria EEPROM en donde se administra una pequeña base de usuarios registrados para acceder al sistema. El programa en el microcontrolador, desarrollado en lenguaje C para PIC, espera a recibir mensajes SMS para de ahí extraer el número de celular que le envió el mensaje y el contenido del mensaje que le indicará la acción a ejecutar.

### **Inventario de materias**

- Análisis de circuitos eléctricos
- Microcontroladores y Microprocesadores
- Diseño digital
- Dinámica de sistemas físicos
- Fundamentos de control
- Procesamiento digital de señales
- Dispositivos electrónicos
- Sistemas electrónicos analógicos
- Circuitos integrados analógicos
- Amplificadores electrónicos
- Electrónica de potencia
- Medición e instrumentación
- Recursos y necesidades de México

### **Resultados esperados**

Desarrollar hardware y software para implementar un sistema de control remoto con base en mensajes SMS utilizando la red GSM.

## **Resumen de la tesis**

La tesis que se presenta esta estructura da de la siguiente forma:

En el capítulo 1 se da un panorama de lo que han sido las distintas generaciones de telefonía móvil, se continúa con una descripción general de la arquitectura de la red de telefonía móvil y los sistemas que la componen, posteriormente se hace referencia a los diferentes tipos de terminales móviles y sus sistemas operativos. En este capítulo también se explica cómo está organizada la arquitectura de la red de mensajería móvil, la estructura que conforma un mensaje de texto, los comandos AT usados en el proyecto y se da una introducción tanto al sistema operativo Android, como a las características del microcontrolador PIC 18F97J60.

En el segundo capítulo se describe el proyecto en general y cada uno de sus bloques, abarcando el sistema electrónico, el algoritmo de control, una descripción del funcionamiento de la aplicación desarrollada en Android y la descripción de la estructura del programa en Android.

Finalmente, en el tercer capítulo se exponen los resultados obtenidos, las conclusiones y el trabajo a futuro.

## 1. Conceptos básicos

En el presente capítulo se exponen los principios básicos para el desarrollo y entendimiento del proyecto.

### 1.1. Principios de telefonía móvil

A partir de 1946 surgieron diferentes sistemas de comunicaciones móviles de dos direcciones, estos presentaban desventajas tanto en la calidad de la comunicación como en la separación física requerida para que funcionaran.

En 1947, los laboratorios Bell introducen el concepto “celular” gracias a D.H. Ring y W.R. Young. El sistema celular que diseñaron propone colocar varios transmisores de baja potencia en diferentes puntos de un territorio, estos transmisores dividen un territorio en pequeñas áreas hexagonales denominadas celdas. También desarrollaron el concepto de reuso de frecuencia entre distintas celdas [Eugenio r, 1999].

El primer sistema de telefonía móvil comercial fue introducido en Japón en el año de 1979 por la compañía NTT (Nippon Telegraph and Telephone). A este sistema le siguieron el NMT (Nordic Mobile Telephony) que fue utilizado en Dinamarca, Noruega, Suecia y Finlandia, el AMPS (Advanced Mobile Phone Service) de EUA, el TACS (Total Access Communications System) del Reino Unido, C-Netz en Alemania occidental, Radiocom 2000 en Francia y RTMI/RTMS en Italia [Sánchez P, 2007].

A estos sistemas de telefonía móvil se les conoce como sistemas de primera generación (1G). Se caracterizaban por ser puramente analógicos y ofrecer solo servicios de voz.

Debido a los problemas de saturación en la capacidad de los sistemas de primera generación y la incompatibilidad de todos los sistemas existentes, se creó en 1982 el grupo GSM que en un inicio significaba Groupe Speciale Mobile y posteriormente cambio a Global System for Mobile Communications. En 1989 la recién formada ETSI (European Telecommunication Standards Institute) prosiguió con los avances en el proyecto de GSM logrando el lanzamiento comercial del sistema en 1992 dando lugar a lo que ahora se les conoce como sistemas de segunda generación (2G).

La principal diferencia entre los sistemas de primera generación y los de segunda generación es que pasaron de ser analógicos a digitales.

Los sistemas de segunda generación mostraron grandes mejoras en comparación a la primera generación, pero funcionalmente seguían siendo lo mismo. Esto dio paso a los sistemas 2.5G que hacen su aparición en 1998. Los sistemas de generación 2.5 se caracterizan por:

- Ofrecer capacidades adicionales a los sistemas 2G.

- Utilizar protocolos estándar similares a los utilizados en internet, que facilitan el uso del internet móvil como bluetooth y WAP.
- Utilizar comunicación y paquetización de voz y datos.

A partir de 2.5G las mejoras en los sistemas de telefonía móvil son para ofrecer más servicios y mejorar la calidad de los anteriores, los sistemas 3G llegan en el año 2001.

En la tercera generación el sistema no solo se utiliza para transferencia de información entre usuarios sino también entre dispositivos portátiles. Las características principales del sistema 3G son:

- Calidad de voz comparable con la calidad ofrecida por sistemas alámbricos.
- La convergencia de la voz y datos con acceso inalámbrico a Internet.
- Servicios de conmutación de paquetes y en modo circuito, tales como tráfico Internet (IP) y video en tiempo real.
- Cobertura global.
- Velocidades de transmisión de hasta 384 Kbps.

En el 2010 salen los sistemas de cuarta generación que ofrecen una velocidad de transmisión de hasta diez veces más rápida que la ofrecida por el 3G, esto para mejorar sus servicios de voz en banda estrecha a servicios multimedia en tiempo real y banda ancha, navegación por la World Wide Web (WWW), correo electrónico multimedia, comercio electrónico móvil, videoteléfono, videoconferencia interactiva, audio y música, aplicaciones multimedia especializadas como telemedicina y supervisión remota de seguridad.

Actualmente la tecnología 4G se extiende por el mundo y no hay datos concretos acerca de una quinta generación.

### **1.1.1. Red de telefonía celular**

Un sistema de telecomunicaciones consiste en una infraestructura física a través de la cual se transporta la información desde la fuente hasta el destino, y con base en esa infraestructura, se ofrecen a los usuarios los diversos servicios de telecomunicaciones. En lo sucesivo se denominará "RED DE TELECOMUNICACIONES" a la infraestructura encargada del transporte de la información.

Para recibir un servicio de telecomunicaciones, un usuario utiliza un equipo terminal a través del cual obtiene entrada a la red por medio de un canal de acceso. Cada servicio de telecomunicaciones tiene distintas características, puede utilizar diferentes redes de transporte, y por tanto, el usuario requiere de distintos equipos terminales.

En la figura 1.1 se muestra la arquitectura básica de una red de telefonía móvil, en donde MS representa a una estación móvil. El sistema está compuesto por los dos subsistemas, BSS y NSS, que se describen a continuación.

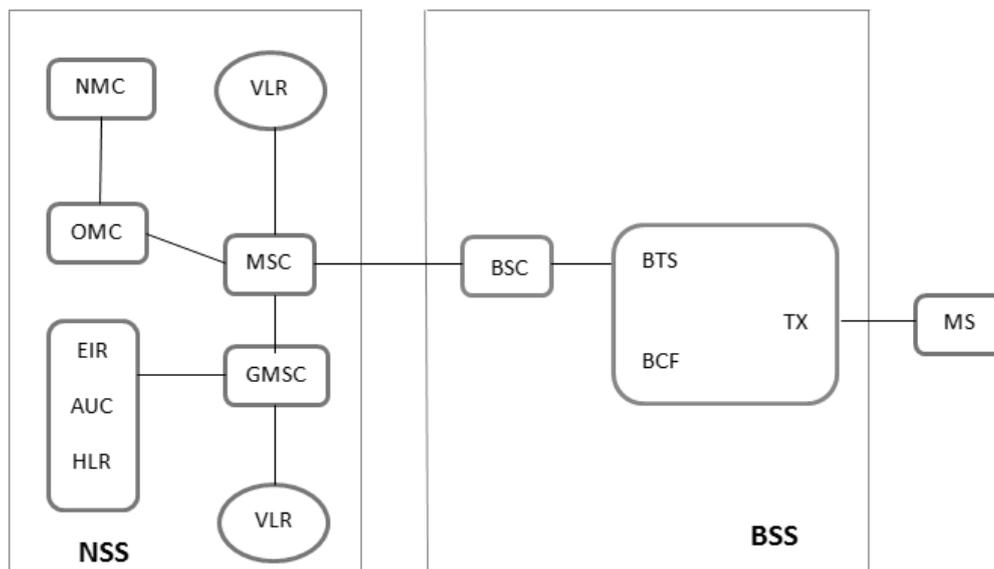


Figura 1.1: Arquitectura básica de un sistema de telefonía móvil (GSM).

### Base Station Subsystem (BSS)

El subsistema de estación base de la arquitectura de la red GSM que esta fundamentalmente asociado con la comunicación con los móviles de la red. Se compone de dos elementos [Parsons J, 1989].

- Base Transceiver Station (BTS): El transceptor de estación base es el elemento que se comunica con los móviles y define el área de cada celda. Sus principales funciones son: La adaptación de la señal, la medición de la señal en nivel y calidad y el cifrado de la información.
- Base Station Controller (BSC): El subsistema de conmutación y red controla un conjunto de BTS. Sus principales funciones son: La selección de canal, la configuración de los canales, la supervisión del enlace, el control de la potencia y el control de los traspasos.
- Base Station Control Function (BCF): El BCF proporciona operaciones y mantenimiento al sistema de red y maneja los estados operacionales de cada TX.
- Transceiver (TX): Proporciona la señal a diferentes frecuencias y sectores de la celda.

## Network Switching Subsystem (NSS)

El NSS contiene una variedad de diferentes elementos y con frecuencia se denomina la red núcleo. Se le proporciona el control principal y la interfaz para toda la red móvil. Sus elementos más importantes son:

- **Mobile Switching Center (MSC):** El MSC es el elemento principal, dentro de la red núcleo. El MSC actúa como nodo normal de conmutación y también proporciona una funcionalidad adicional, para habilitar los requisitos de un usuario móvil. Sus principales funciones son: el control del establecimiento de llamada, el control de los trasposos, el acceso a la base de datos del sistema y el acceso a otras redes (móvil, fija y de datos).
- **Home location register (HLR):** Es la base de datos que contiene información sobre la localización y características de los usuarios asociados a cada MSC. De esta manera la red GSM es capaz de encauzar las llamadas a la estación más relevante para la MS.
- **Visitor location register (VLR):** Contiene toda la información de un MS, que está de paso en el área controlada por la MSC. Esta información es necesaria para habilitar los servicios seleccionados por el suscriptor.
- **Equipment Identity Register (EIR):** Es la entidad que decide si un equipo móvil dado puede ser permitido en la red. Cada equipo tiene un número conocido como International Mobile Equipment Identity (IMEI), la EIR contiene todos esos datos y dependiendo de ellos, el móvil puede ser asignado a uno de estos tres estados, “permite acceso”, “acceso prohibido” o “acceso controlado”.
- **Authentication center (AUC):** Es la base de datos que contiene la clave secreta, también contenida en la tarjeta SIM del usuario, se utiliza para la autenticación y para el cifrado del canal de radio.
- **Operation and Management Center (OMC):** OMC supervisa todas las redes y entidades (srs, mss, interruptores, bases de datos, etc). Sus principales funciones son la gestión de los siguientes elementos: las alarmas, los fallos, la configuración, la seguridad y el rendimiento.
- **Mobile Gateway Switching Centre (GMSC):** El GMSC es el punto en el que una llamada se encamina, sin ningún conocimiento de la ubicación de la MS. El GMSC es por lo tanto responsable de la obtención de la MSRN (Mobile Station Roaming Number) desde el HLR con base en el (número RDSI de Estación Móvil, el "número de directorio" de un MS) MSISDN y el encaminamiento de la llamada a la MSC.

- Network Management Console (NMC): El NMC es utilizado para monitorear y controlar la conectividad de la red.

### **Ejemplo: Establecimiento de una llamada en una red GSM**

Primero se indica el tipo de servicio que se desea, en este caso voz. Se introduce el número destino y se envía. La estación móvil pasa la información a la BTS, esta la pasa a la MSC. Cuando la MSC recibe el mensaje de establecimiento, analiza la petición y comprueba si puede aceptarla. La aceptación de la llamada depende de la capacidad de la MSC y la VLR para proveer este servicio, de las características de suscripción del cliente (determinado de forma local gracias a la información del cliente que el HLR envió a la MSC y VLR en el proceso de registro) y de la disponibilidad de recursos. Si alguno de estos requisitos falla, se aborta la llamada, si todo está bien, la MSC comienza el establecimiento a través de la red y notifica a la estación móvil de este evento.

Transcurrido un tiempo, la MSC recibirá de la red exterior información sobre la petición de llamada realizada, tal y como lo ve la central a cargo de la persona llamada. Tal información puede indicar que la terminal de la persona llamada está siendo alertado, o que la llamada ha sido abortada por cualquier motivo (congestión, ocupado, no localizable). Esta información es transferida directamente al usuario móvil y en su caso la MSC abortará la llamada.

Cuando el cliente destino responde a la llamada, la MSC recibe un mensaje indicándolo. Cuando esto ocurre, se establece un camino de voz entre los usuarios (hasta ahora todo había sido señalización). Entonces la estación móvil interrumpe la indicación de llamada, responde a la red y establece el circuito a través de la interfaz de radio.

#### **1.1.2. Terminales móviles**

Existen diferentes tipos de terminales móviles tales como teléfonos celulares, computadoras del tipo tableta o reproductores multimedia. Todos estos cuentan con diferentes tipos de accesorios y sistemas operativos. Para no hablar de marcas, las terminales móviles han sido separadas en dos grupos: las terminales accesibles y las terminales inteligentes (Smartphone y Tablet).

##### **Terminales accesibles**

Las terminales accesibles están limitadas en funciones y accesorios en comparación con las de nueva generación. Estas cumplen con la función de comunicar a los usuarios por medio de voz o mensajes de texto. Generalmente se distinguen por su pantalla fácilmente legible, teclas grandes y una buena ergonomía.

## Terminales inteligentes

Todos los teléfonos de última generación son llamados Smartphone o teléfonos inteligentes, porque no se limitan solamente a comunicar a los usuarios como lo hacen las terminales accesibles, incluyen servicios y programas que los hacen parecer más unos ordenadores portátiles que teléfonos móviles. Una de las principales características de los teléfonos inteligentes es permitir la conexión a internet.

Estas terminales funcionan regidos por un sistema operativo, que tiene el mismo propósito que en los ordenadores, gestionar y regular el funcionamiento del aparato.

Los sistemas operativos que actualmente se encuentran en el mercado para este tipo de dispositivos son SYMBIEN, ANDROID, IOS, WINDOWS PHONE 7, BADA 1.1, BLACKBERRY OS V6.0 y Firefox OS.

Algunas características de los sistemas operativos son las siguientes.

- iOS



*Ventajas:* La integración de las aplicaciones con el hardware es perfecta, debido a que solo los dispositivos de Apple pueden contar con este sistema operativo. Es uno de los sistemas con más aplicaciones y normalmente el primero en recibir las nuevas creaciones.

*Inconvenientes:* Depende de un ordenador con iTunes instalado para realizar tareas como la configuración inicial, pasar contenido multimedia al móvil o actualizaciones, mientras que con otros sistemas operativos esto se hace vía Wi-Fi. iTunes decide por el usuario dónde se almacenan todos los elementos, algo que para los usuarios menos expertos no es trivial.

*Aplicaciones:* App Store es la tienda más completa y de mayor calidad de todas las existentes. Sin duda, es una de las claves del rotundo éxito del iPhone. Ofrece aplicaciones diseñadas específicamente para disfrutar de la informática portátil y las hay para todos los gustos, desde las más útiles hasta las más frívolas.

*Teléfonos:* No hay mucho donde elegir. Se tendrá que comprar un iPhone para disfrutar de este sistema, un smartphone de precio elevado pero gran calidad. También se usa en otros dispositivos de Apple, como el iPod Touch y el iPad

- Android



*Ventajas:* Destaca por su facilidad para transferir ficheros al ordenador: se conecta el puerto USB al PC, se arrastran los ficheros y ya está. Con otros teléfonos, esta sencilla operación requiere de un programa de sincronización. También destaca por su navegador web, el único de su categoría 100% compatible con Flash Player, por lo que con los móviles que corran con Android se podrán ver la mayoría de las páginas y animaciones web.

*Inconvenientes:* No es capaz de sincronizar con Outlook y a la hora de recibir las actualizaciones del software depende de los fabricantes (LG, HTC, Samsung, etc.).

*Aplicaciones:* Se podrá navegar en el renovado Android Market. Cuanto más se extiende este sistema entre los usuarios, más son los desarrolladores que se animan a crear aplicaciones: cantidad y variedad están garantizadas.

*Teléfonos:* Son muchas las marcas que equipan sus móviles con Android: LG, Samsung, Sony Ericsson. Destaca HTC, que incorpora un interfaz mucho más personalizable que el resto (HTC Sense). Este añadido permite que algunas tareas sean más rápidas y sencillas. También está HTC Sync, un software para PC con el que sincronizar contactos y calendarios de Outlook, así como hacer copias de seguridad de todos los datos del teléfono.

- Windows Phone



*Ventajas:* Su interfaz es sencilla de utilizar y está muy bien optimizada. Dispone de un servicio propio de almacenamiento en la nube como Skydrive y la integración con el resto de equipos Windows 8 es realmente buena.

*Inconvenientes:* No es compatible con Outlook y para transferir ficheros al móvil es necesario instalar un programa (Zune) en el PC.

*Aplicaciones:* El Market Place es una de las tiendas de aplicaciones más nuevas, pero también de las más logradas. Es fácil de usar, con muchos detalles de cada aplicación y, aunque podría tener un repertorio mayor, lo que hay sirve para cubrir bien el mercado.

*Teléfonos:* Muchos fabricantes entre los que elegir. Samsung, LG, HTC, Nokia etc.

- Symbian



*Ventajas:* Funciona correctamente y es fácil de usar.

*Inconvenientes:* Es el que menos opciones ofrece en el correo electrónico, una característica importante para un smartphone.

*Aplicaciones:* De buena calidad, aunque sin llegar a despertar el mismo interés que las de sus competidores.

*Teléfonos:* En todos los que actualmente fabrica Nokia, aunque parece que dentro de poco irá dejando sitio al sistema de Windows.

- Bada



*Ventajas:* Junto con iOS, es el único compatible con Outlook. Suele funcionar en teléfonos que son más baratos.

*Inconvenientes:* Es inestable y da muchos errores, especialmente en el correo electrónico. Limitada compatibilidad en formatos de vídeo. No hay ninguna razón para comprar un modelo que funcione con Bada en lugar de uno que lo haga con Android, salvo quizás el precio del smartphone.

*Aplicaciones:* Tiene una tienda de mala calidad y en la que hay muy poco de donde elegir. Samsung ha desarrollado algunas aplicaciones, pero no es nada si comparamos con la competencia. La peor de todas las tiendas existentes.

*Teléfonos:* Solo en la serie de modelos Samsung Wave.

- Blackberry OS



*Ventajas:* Un teclado físico muy cómodo para escribir. En España, los operadores de telefonía móvil se hacen cargo de todos los trámites y se lo ponen fácil al usuario: algunos operadores tienen tarifas específicas para Blackberry.

*Inconvenientes:* Tiene una configuración inicial realmente complicada, en especial la del correo electrónico, que depende de servidores propios de Blackberry e incluye una cuota mensual. Su teclado físico no es lo mejor para navegar.

*Aplicaciones:* El App World es una de las tiendas peor valoradas debido a que tiene pocas aplicaciones y de mala calidad.

*Teléfonos:* Es un sistema exclusivo para la marca Blackberry. Su teclado físico es característico, pero ya empieza a haber modelos con pantalla táctil como el Blackberry Torch 9800.

- Firefox OS



*Ventajas:* Está apoyado por grandes empresas como Telefónica. El sistema es totalmente abierto y admite la modificación completa por parte de los usuarios, desarrolladores y operadoras y está totalmente sobre la base de HTML5.

*Desventajas:* El hecho de ser un sistema totalmente nuevo, en el que tendremos que esperar a ver el soporte de la comunidad de desarrolladores y de las operadoras.

### 1.1.3. Mensajes de texto SMS

El servicio de mensajes cortos o SMS permite transmitir mensajes de texto entre una terminal MS y otros sistemas como pueden ser un elemento fijo que solo atienda ordenes u otra terminal móvil.

Al enviar un mensaje, éste pasa por diferentes etapas para ser validado y re-direccionado, como se vio en el capítulo 1.1.1 en el ejemplo de establecimiento de una llamada. La diferencia más relevante al mandar un mensaje es que no se necesita crear un canal de comunicación entre los usuarios, sino que el mensaje es enviado a una entidad llamada SC y después la terminal SC reenvía el mensaje a su destino. Ver ejemplo de envío y recepción de un mensaje en el anexo A.

### Servicio SMS

El servicio SMS se divide en dos partes básicas, SM MT Y SM MO, como se muestra en la figura 1.2.

- SM MT: Servicio de entrega de un mensaje desde el SC hasta una SME (sistema que puede enviar o recibir mensajes).
- SM MO: Servicio de envío de un mensaje desde una MS hasta un SC.

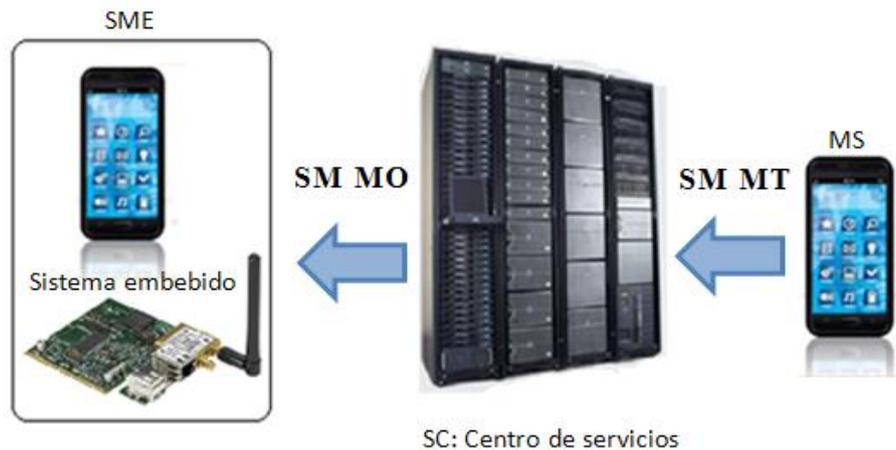


Figura 1.2: Ejemplo del servicio de SMS.

Los mensajes SMS, según lo especifica el Etsi (European Telecommunications Standards Institute), pueden contener hasta 160 caracteres de longitud, donde cada carácter es de 7 bits de acuerdo a un alfabeto especial. En el caso de mensajes de ocho bits tiene un máximo de 140 caracteres y no son generalmente visibles para los teléfonos como mensajes de texto, sino que se utilizan para los datos, por ejemplo la mensajería inteligente (imágenes y señales de llamada).

Para mensajes de 16 bits el máximo es de 70 caracteres, se utilizan para mensajes de texto Unicode (UCS2) y se pueden ver en la mayoría de los teléfonos [ETSI, 1996].

### Arquitectura de una red SMS

La arquitectura básica de una red SMS es la que se muestra en la figura 1.3.

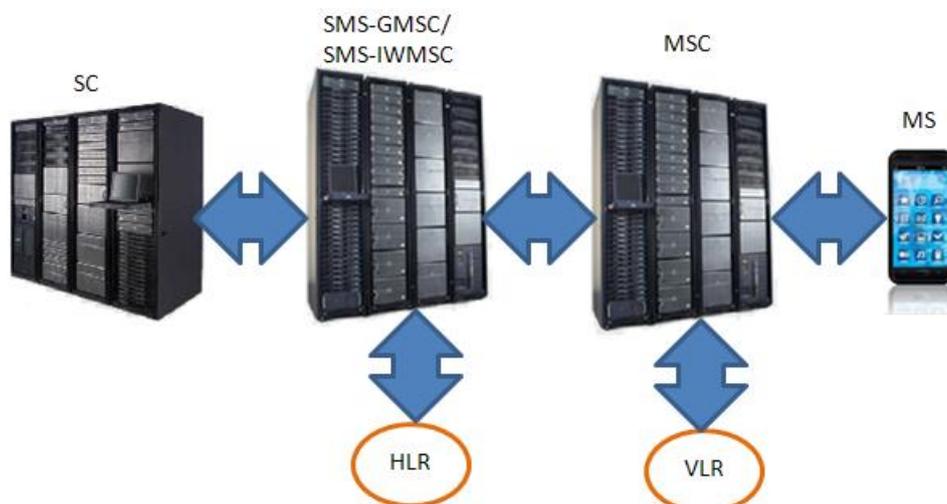


Figura 1.3: Arquitectura de una red SMS.

Los componentes de la figura 1.3 son los siguientes.

- SC: Centro de servicio. Es responsable de dirigir o almacenar y transmitir el mensaje desde un MS a un MS, este puede ser interno o externo a la red GSM. Sus principales funciones son: Almacenar un determinado mensaje que no ha podido entregarse hasta que el MS esté disponible y disponer de una tabla de reintentos donde se almacenan los intervalos de tiempo para volver a transmitir un mensaje.
- SMS-GMSC: MSC de interconexión entre PLMN y el SC. Es el nodo que realiza la pasarela para el servicio SMS-MT y puede ser cualquier MSC. Sus principales funciones son: Comprueba que todos los parámetros son correctos, envía el mensaje hasta la MSC y VLR donde está registrado el teléfono móvil e informa al HLR del resultado de entrega.
- SMS-IWMSC: MSC pasarela para el servicio de mensajes cortos. Es el nodo que realiza la pasarela para el servicio SMS-MO, esto quiere decir que solo interfiere cuando un usuario ha generado un mensaje. Su principal función es informa al SC que el MS, para el que tenía un mensaje, se ha encontrado.
- HLR: Registro de usuarios locales. (ver capítulo 1.1.1)
- VLR: Registro de usuarios visitantes. (ver capítulo 1.1.1)
- MSC: Centro de conmutación.
- MS: Estación móvil.

## Formatos de los mensajes SMS

Normalmente se habla de dos formas de envío y recepción de mensajes SMS que son el modo texto y el modo PDU. El modo texto es solo una codificación de una secuencia de bits en formato PDU, por lo que no se puede hablar de este como un método de transferencia de mensajes.

La cadena PDU por la que viaja el mensaje no sólo contiene el mensaje, sino también una gran cantidad de información sobre el remitente, el centro de servicio de SMS, la marca de tiempo, etc. Todo está en la forma de octetos hexadecimales o decimales.

En SMS hay seis tipos de cadenas PDU que dependen del tipo de servicio, como muestra la figura 1.4.

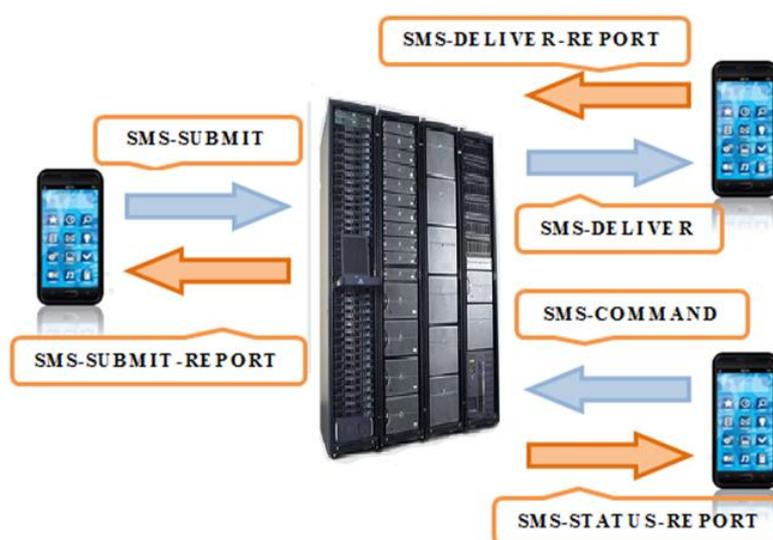


Figura 1.4: Tipos de PDU'S.

Los tipos de servicio de la figura 1.4 se describen en la tabla 1.1.

Tipo de servicio	Descripción
SMS-DELIVER	Transmitir un mensaje desde el SC al MS.
SMS-DELIVER-REPORT	Error en la entrega.
SMS-SUBMIT	Transmitir un mensaje corto desde el MS al SC.
SMS-SUBMIT-REPORT	Error en la transmisión.
SMS-STATUS-REPORT	Transmitir un informe de estado desde el SC al MS.
SMS-COMMAND	Transmitir un comando desde el MS al SC.

Tabla 1.1: Tipos de servicios y cadenas PDU.

La estructura más común del formato PDU en el envío de mensajes SMS es la que se muestra en la figura 1.5 y se describe en la tabla 1.2.

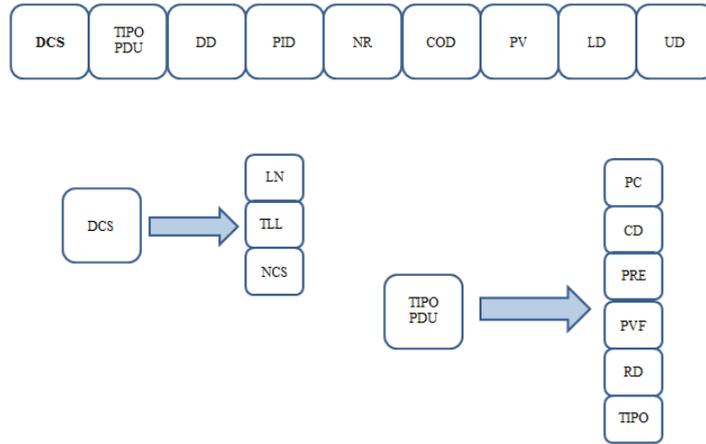


Figura 1.5: Envío de SMS en formato PDU.

Campo		Descripción
DCS	LN	Longitud del número del centro de servicio
	TLL	Tipo de llamada: 81 nacional 91 internacional
	NCS	Número de teléfono del centro de servicios de mensajes cortos
TIPO DE PDU	PC	Indica si existe camino de respuesta: 0 SI 1 NO
	CD	Cabecera de datos. Indica si el campo UD contiene solo el mensaje. 0 SI 1 NO
	PRE	Informe de estado. Dicho reporte lo enviara el SC: 0 no solicitado 1 si solicitado
	PVF	En qué forma está presente el formato: Formato Relativo Formato Mejorado Formato Absoluto
	RD	Indica si existen más mensajes de texto esperando en el SC
	TIPO	Identificador del tipo de mensaje: SMS Recibido SMS Enviado SMS Estado
DD	Dirección de destino	
PID	Protocolo de identificación	
NR	Número de referencia	
COD	Tipo de identificación dentro de los datos de usuario	
PV	Periodo de valides del mensaje	
LD	Longitud de del campo UD	
UD	Datos de usuario	

Tabla 1.2: Campos de la estructura de envío PDU.

La trama de recepción tiene una estructura similar a la estructura de envío, lo único que cambia es el contenido en las variables DD, LD y cómo podemos ver en la figura 1.6, se sustituye PV por TR. En la tabla 1.3 se describe la función de los nuevos campos.

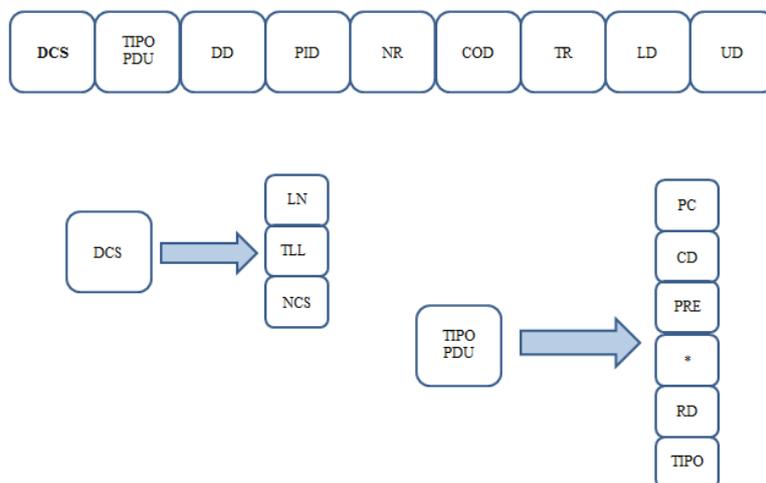


Figura 1.6: Recepción de SMS en formato PDU.

Campo	Descripción
DD	Dirección de origen
TR	Indica el momento en que el centro de servicio recibió el mensaje
LD	Longitud de datos
PVF	No se utiliza en la recepción

Tabla 1.3: Campos de la estructura de recepción PDU.

A manera de ejemplo, a continuación se presentan las figuras 1.7 y 1.8 que despliegan el mensaje almacenado en un chip SIM de teléfono celular, una en modo texto y otra en modo PDU, mediante el comando at+cmgr=5. En la figura 1.7 se configura el modo texto con el comando at+cmgf=1 y en la figura 1.8 se configura el modo PDU con el comando at+cmgf=0. En la sección 1.1.4 se describirá con mayor detalle el lenguaje de comandos AT.

```

aatt++ccmnggf==1
OK
aatt++ccmnggr==55
+CMGR: "REC READ","5534083640",,"13/08/09,23:44:33-20"
analog1;0

OK
    
```

Figura 1.7: Mensaje en modo texto.

```

aatt++ccmmggff==00
OK
aatt++ccmmggr==55
+CMGR: 1,,26
0891254901001014F0240A81554380630400003180903244330A09617798FD3EC77630

OK
    
```

Figura 1.8: Mensaje en modo PDU.

La secuencia de octetos del mensaje PDU de la figura 1.8 se compone de tres partes: Un octeto inicial que indica la longitud de la información SMSC ("08"), la propia información SMSC ("91254901001014"), y la parte SMS-DELIVER.

La tabla 1.4 divide de izquierda a derecha la cadena PDU mostrada en la figura 1.8, en sus diferentes partes que la componen.

PDU	Nombre	Descripción
08	SCA	LN Longitud del número del centro de servicio.
91		TLL El valor "81" indica que se trata de un número nacional. El valor "91" indica que se trata de un número internacional.
254901001014F0		NCS número del centro de servicio
24	PDU TYPE Tipo de PDU.	
0A	ADDRESS LENGTH Longitud del número de teléfono.	
81	TIPO DE DIRECCION El valor "81" indica que se trata de un formato nacional. El valor "91" indica que se trata de un formato internacional.	
5543806304	NUMERO DE TELEFONO Cuando el número de teléfono es impar se añade una F al último octeto. En esta ocasión solo es necesario invertir cada uno de los valores de los octetos. Ejemplo: 5534080304	
00	PID Protocolo de identificación.	
00	DCS Tipo de codificación de los datos del mensaje.	
3180903244330 <sup>a</sup>	TR Hora de recepción del mensaje 09/08/12 23:44:55-0A.	
09	UD Numero de octetos que forman el mensaje.	
617798FD3EC77630	UD Datos codificados a 7 bits.	

Tabla 1.4: Secuencia PDU dividida en todos sus componentes.

A continuación se mostrara un ejemplo de cómo obtener el mensaje de texto contenido en la cadena PDU. Para decodificar el mensaje se rescriben los octetos de izquierda a derecha en su equivalente binario, como se muestra en la figura 1.9.

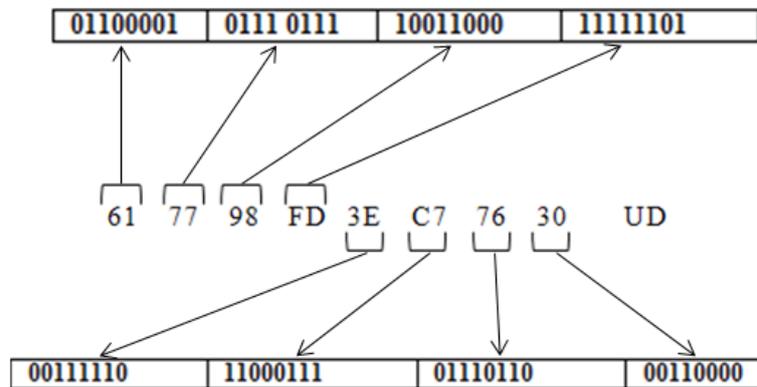


Figura 1.9: Conversión de hexadecimal a binario.

Posterior mente se convierten a septetos, quitándoles en el mismo orden, los bits más significativos que le sobren y colocándolos como bits menos significativos en el siguiente octeto, como se muestra en la figura 1.10.

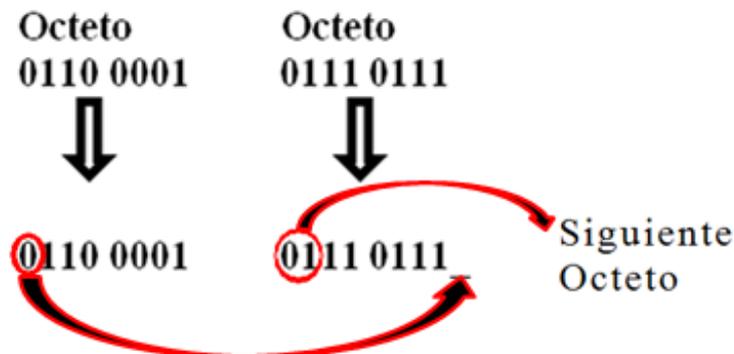


Figura 1.10: conversión de octetos a septetos.

Una vez obtenidos los septetos, éstos se convierten a su equivalente en hexadecimal o en su equivalente decimal, para tener mayor facilidad al buscar los caracteres, a que corresponde cada número en el alfabeto de 7 bits, especificado en el documento gsm 03.38 [ETSI, 1995].

La tabla 1.5 muestra el mensaje contenido en la cadena PDU, después de haber obtenido los septetos y compararlos con la tabla contenida en este documento en el anexo B.

7 bits	Hexadecimal	Texto
1100001	61	A
1101110	6E	N

1100001	61	A
1101100	6C	L
1101111	6F	O
1100111	67	G
0110001	31	1
0111011	3B	;
0110000	30	0

Tabla 1.5: conversión de septetos a su equivalente en caracteres.

#### 1.1.4. Lenguaje de comandos “AT”

Los comandos AT son un grupo de instrucciones codificadas que fueron creadas para comunicarse con las terminales módems. Fueron desarrollados en 1977 por Dennis Hayes como un interfaz de comunicación con un MODEM para así poder configurarlo y proporcionarle instrucciones, tales como marcar un número de teléfono. Más adelante, con el avance del baudio, fueron las compañías Microcomm y US Robotics las que siguieron desarrollando y expandiendo el juego de comandos hasta universalizarlo.

Posteriormente la tecnología móvil GSM adoptó este lenguaje para poder comunicarse con sus terminales móviles, más en concreto con la tarjeta SIM de un dispositivo móvil. De esta forma todos los teléfonos celulares poseen los mismos comandos AT, que sirven de interfaz para poder configurar las terminales y permitirle realizar acciones, tales como mandar mensajes de texto o de voz, realizar una llamada entre otras cosas. A continuación se presenta un resumen de los comandos AT [Sim com, 2006], que permiten, en particular, comunicar un microprocesador con un chip de radio frecuencia SIM 300 en el interior de un dispositivo móvil.

*AT*: El comando más básico es *AT*. Este ayuda a comprobar si la conexión entre el equipo ETD (equipo terminal de datos) y el ECD (equipo control de datos) esta lista. La figura 1.11 muestra un ejemplo de este comando.

```
at
ERROR
at
OK
```

Figura 1.11: Comando AT.

*AT+CMGS*: Este comando permite enviar un mensaje. Ver figura 1.12.

```
AT+CMGS="5534498951" → Número del destinatario
> Se enviara el mensaje al oprimir control+Z
+CMGS: 212

OK
```

Figura 1.12: Comando AT+CMGS.

Para indicar al sistema electrónico que se terminó de escribir el mensaje es necesario oprimir “Ctrl+Z” para dar un salto de línea basta con oprimir “enter”.

*AT+CMGW*: Permite escribir un mensaje SMS en memoria. Ver figura 1.13.

```
AT+CMGW="5534498951"
> Hola! a que hora sales hoy?
+CMGW: 3
```

Figura 1.13: Comando AT+CMGW.

*AT+CMSS*: permite enviar un mensaje que se encuentra almacenado en memoria. Ver figura 1.14.

```
AT+CMSS=3
+CMSS: 211

OK
```

Figura 1.14: Comando AT+CMSS.

Al mandar un mensaje que se encuentra en memoria, como muestra la figura 1.14, solo se tiene que especificar la localidad en la que se encuentra éste. Cuando el sistema realice la instrucción responderá con +CMSS: y el número de mensajes mandados hasta ese momento.

*ATD*: permite realizar una llamada sea ésta de datos o de voz. Ver figura 1.15.

```
ATD5534498951;
OK
ATH
OK
```

Figura 1.15: Comando ATD.

Se tiene que indicar el número de celular al que se llamará justo después del comando ATD como se muestra en la figura 1.15. El comando ATH termina la llamada.

*AT+CMGR*: Permite leer mensajes SMS de la bandeja de entrada o del chip SIM. Ver figura 1.16.

```
AT+CMGR=1
+CMGR: "STO UNSENT","5534498951",
Tenemos un regalo para ti! Gratis 20 SMS para mensajear a tus amigos telcel al r
ecargar $50 o mas. Vigencia al 16 de octubre. mas inf cac telcel
OK
```

Figura 1.16: Comando AT+CMGR.

*AT+CMGD*: Permite borrar mensajes de la memoria. Ver figura 1.17.

```
OK
AT+CMGD=?
+CMGD: (1-20)

OK
AT+CMGD=10
OK
```

Figura 1.17: Comando AT+CMGD.

La sintaxis *AT+CMGD=?* mostrada en la figura 1.17, proporciona las capacidades de la memoria.

*AT+CMGF*: Permite configurar el formato en el que serán mostrados los mensajes, normalmente cuenta con dos formatos; estos pueden ser PDU o texto. La figura 1.18 muestra cómo utilizar este comando.

```
AT+CMGF?
+CMGF: 0

OK
AT+CMGF=?
+CMGF: (0-1)

OK
AT+CMGF=1
OK
```

Figura 1.18: Comando AT+CMGF.

La sintaxis *AT+CMGF?* indica el formato actual, *AT+CMGF=?* indica los formatos soportados en donde 0 es formato PDU y 1 formato texto.

*AT+CMGL*: Permite leer los datos contenidos en la memoria del chip SIM, los cuales estarán divididos de la siguiente manera: "REC UNREAD", "REC READ", "STO UNSENT", "STO SENT", "ALL". La figura 1.19 muestra un ejemplo de esta instrucción.

```
aatt+ccmngl1=="AALLL"
+CMGL: 1,"REC READ","UNONOTICIAS",,"13/08/02,22:13:23-20"
MANANA SUBEN GASOLINAS 11 CTVS/EU EMITE ALERTA POR AL QAEDA; CIERRA EMBAJADAS/BE
NITEZ ESTABA CONDENADO A MORIR: FEDERACION ECUATORIANA.VER http://bit.ly/1cwwVT0

+CMGL: 4,"REC UNREAD","UNONOTICIAS",,"13/10/03,21:27:41-20"
VISITAR ACAPULCO,UNA FORMA DE APOYAR GRO:EPN/MUERE UNO EN PERSECUCION EN CAPITOL
IO DE EU.VER http://bit.ly/18VBPFN /LLAMAN A 26 AL TRI.VER http://bit.ly/18zqex8
OK
```

Figura 1.19: Comando AT+CMGL.

*AT+CBC*: Devuelve el estado de carga de la batería. Ver figura 1.20.

```
AT+CBC=?
+CBC: (0-1),(1-100),(voltage)

OK
AT+CBC
+CBC: 2,96,4156

OK
```

Figura 1.20: Comando AT+CBC.

El comando *AT+CBC=?* muestra la respuesta con el siguiente formato. El primer dato indica si la batería está conectada, el segundo es el porcentaje de carga de la batería y por último el voltaje.

*AT+CSQ*: Devuelve el estado de calidad de la señal de cobertura. Ver figura 1.21

```
AT+CSQ
+CSQ: 22,7

OK
```

Figura 1.21: Comando AT+CSQ.

El número que regresa éste comando puede ser interpretado por medio de la tabla 1.6.

Respuesta	Valor en dB
0	-113 dBm o menos
1	-111 dBm
2 a 30	-109..-53 dBm
31	-51dBm o más
99	Valor desconocido

Tabla 1.6: Valor en dBm de la respuesta a CSQ.

*AT+CPAS*: Informa del estado de actividad de la tarjeta SIM. Ver figura 1.22.

```

AT+CPAS=?
+CPAS: (0,2,3,4)

OK
AT+CPAS
+CPAS: 0
    
```

Figura 1.22: Comando AT+CPAS.

La sintaxis AT+CPAS=? mostrada en la figura 1.22, responde con los estados en los que puede estar el sistema electrónico, la tabla 1.7 explica a que corresponde cada número.

Número	Descripción
0	Ready (Encendido pero inactivo)
2	Unknown (Desconocido)
3	Ringing (Llamada entrante en proceso)
4	Call in progress (Llamada saliente en proceso)

Tabla 1.7: Descripción de los estados en el comando AT+CPAS.

## 1.2. El sistema operativo Android

Con la aparición de la nueva generación de teléfonos inteligentes o Smartphones, Google presenta a finales del 2008 su primer dispositivo móvil con sistema operativo Android, aunque es un sistema operativo joven con apenas 5 años de vida, este se encuentra en constante evolución y responde a las necesidades de los usuarios.

### 1.2.1. Características

Android es una plataforma para el desarrollo de software libre, para dispositivos móviles, creada por una pequeña empresa llamada Android Inc, que posteriormente fue comprada por Google. Actualmente Android recibe el apoyo de diversas empresas que conforman el consorcio Open Handset Alliance.

Android ha tenido gran aceptación en el mercado tanto por usuarios como por la industria. Como se puede ver en la figura 1.23, se ha convertido en uno de los sistemas operativos móviles que predominan el mercado.

Android no fue diseñado orientado exclusivamente para su uso en dispositivos móviles, éste es flexible y su potencial de desarrollo puede llegar a diferentes tipos de hardware como tabletas, cámaras, relojes, electrodomésticos, TV e incluso la domótica.

Esto presenta una gran ventaja en comparación del IOS de Apple, el cual tiene que desarrollar una aplicación para cada hardware sea Smartphone o Tablet.

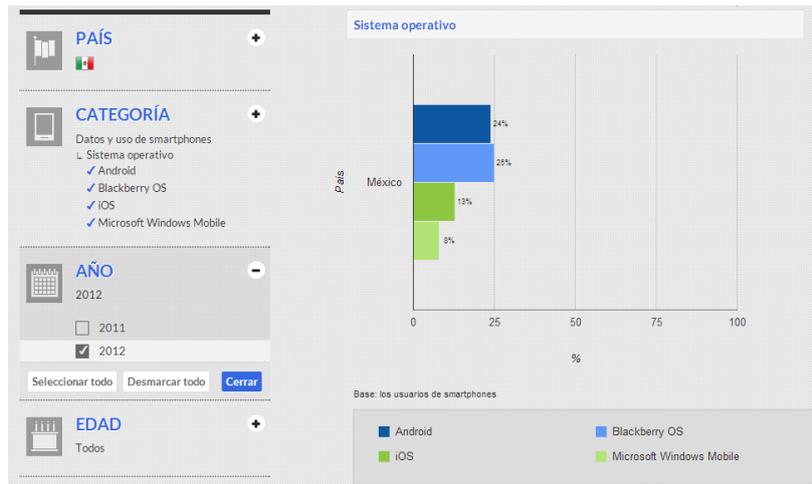


Figura 1.23: Penetración de los sistemas operativos en México [Google, 2013].

### 1.2.2. Aplicaciones

Todas las aplicaciones creadas con la plataforma Android, incluirán como base un correo electrónico, calendario, programa de SMS, mapas, navegador, contactos, y algunos otros servicios mínimos. Todas ellas escritas en el lenguaje de programación Java.

Actualmente existen diversas aplicaciones creadas por los usuarios y expertos de Android que se pueden descargar desde Google Play. Para crear una cuenta como desarrollador Android y poder subir las aplicaciones se cobra un pago único de 25 dólares.

### 1.2.3. Desarrollo de aplicaciones

Existen diferentes software para programar en Android, muchos de ellos tratan de hacer las cosas más sencillas y están orientados para personas que no están interesadas en un trasfondo técnico.

El entorno de desarrollo de Eclipse es el software más utilizado para crear aplicaciones Android, en éste se programa en Java, pero también en conjunto con el SDK creado por Google, cuenta con otras herramientas que ayudan en el diseño visual como botones y cuadros de texto. También se cuenta con un emulador con el cual se pueden ver preliminarmente las aplicaciones en tiempo real en diferentes tipos de teléfonos celulares.

El software se puede descargar desde la página oficial de Android [Android Developers, 2009].

Una vez que se descomprima el fichero descargado, se tendrá el aspecto de la figura 1.24.

Nombre	Fecha de modifica...	Tipo	Tamaño
eclipse	21/09/2013 11:14 ...	Carpeta de archivos	
sdk	04/02/2013 10:02 ...	Carpeta de archivos	
SDK Manager	06/12/2012 11:14 a...	Aplicación	350 KB

Figura 1.24: Estructura interna del paquete SDK.

El siguiente paso será descargar las diferentes API, para ello se ejecuta el archivo *SDK Manager* que se muestra en la figura 1.24. La pantalla de la aplicación ejecutada se muestra en la figura 1.25.

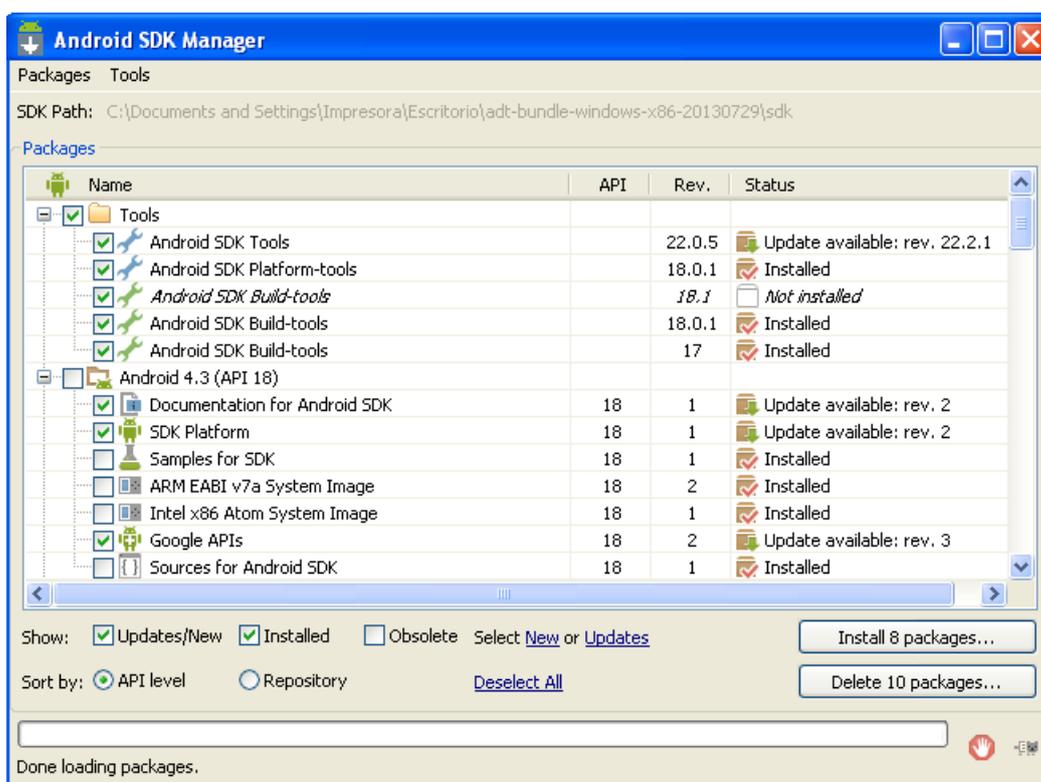


Figura 1.25: SDK Manager.

Para crear un nuevo proyecto se tiene la opción del menú *archivo/nuevo/proyecto/Android*. Se indican los parámetros iniciales como nombre del proyecto, entre otros. Hecho esto, se obtiene como resultado un esqueleto de programa como el indicado en la figura 1.26.

Eclipse proporciona dos formas de organizar las vistas de la aplicación, una por código XML y otra visual en la cual se arrastran los elementos que se requerirán, a una simulación de pantalla como la que se muestra en la figura 1.26. De esta forma eclipse generará el código XML correspondiente a la ubicación y tipo de elementos insertados en la pantalla.

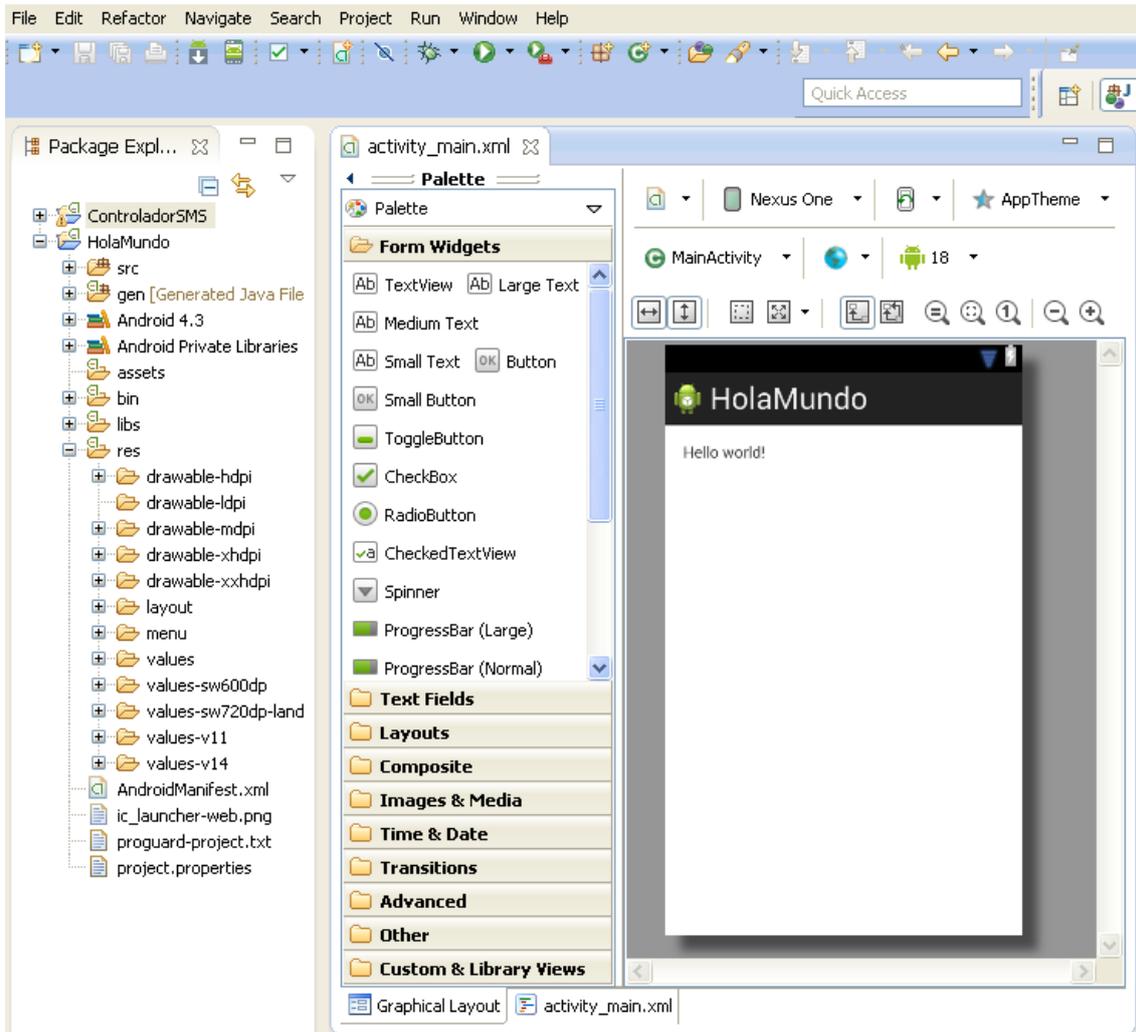


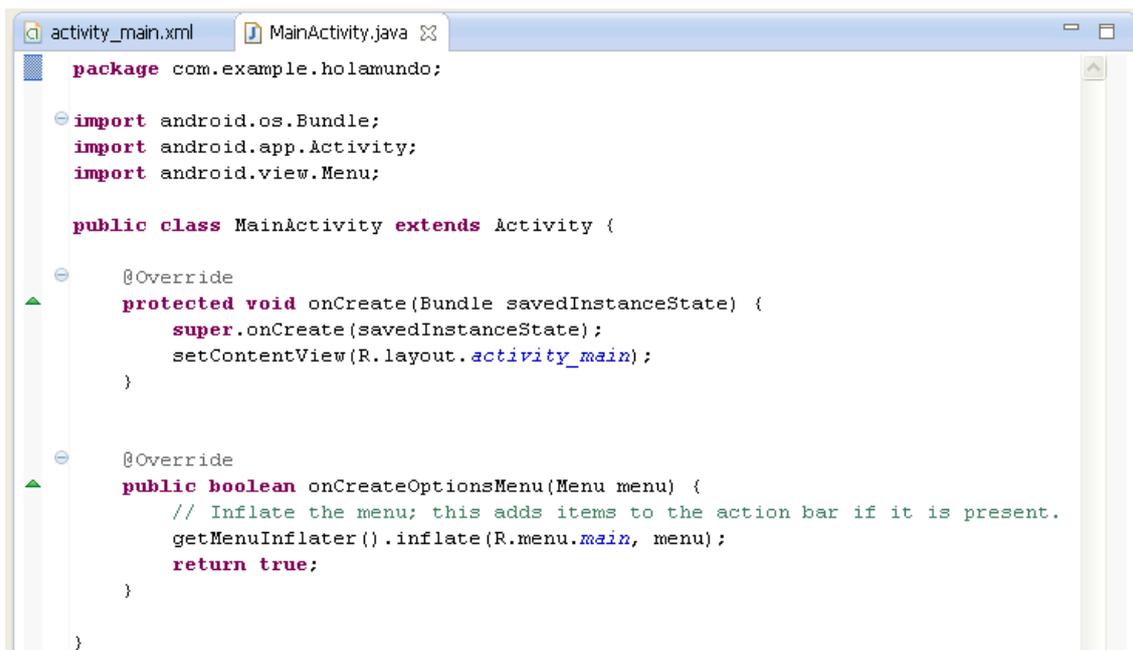
Figura 1.26: Pantalla principal Eclipse.

Como se aprecia en la figura 1.26, en el lado izquierdo del ambiente se ha creado una serie de carpetas con diferentes ficheros. Estas carpetas se organizan en una carpeta que contiene todo el programa, dentro de ella se encuentran las carpetas que contienen el código, los recursos y al final un par de ficheros que son los descriptores de la aplicación. En general, el contenido de las carpetas es como se describe a continuación.

- *Src*: Esta carpeta contiene el programa principal, en el caso de la figura 1.26, *HolaMundo*.
- *Gen*: Contiene todos los ficheros de recursos necesarios para el proyecto: imágenes, vídeos, cadenas de texto, etc., que se encuentran distribuidos en las carpetas de recursos.
- *Assets*: contiene recursos pero a diferencia de *Res*, a estos no se accede desde la clase *R*, que se describirá a continuación.
- *Res*: Se almacenan los recursos o ficheros de la aplicación.
- *Drawable*: Contiene los elementos (imágenes) que se utilizan en la aplicación

- *Layout*: Contiene un *main.xml* que es la pantalla que se va a visualizar por defecto.
- *Values*: Se encuentra el archivo *string.xml* que contiene todas las cadenas de caracteres que utiliza la aplicación.

Para explicar cómo está compuesto un programa de Android en el entorno de Eclipse, se empezará describiendo a la clase principal, en este caso *MainActivity.java*, que se muestra en la figura 1.27.



```
activity_main.xml | MainActivity.java
package com.example.holamundo;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

Figura 1.27: Clase *MainActivity*.

En la figura 1.27, lo primero que se encuentra es un *package* o paquete con el nombre del programa. Este contendrá todas las clases en el programa. Después se encuentran los *import* que ayudan a especificar que librerías se utilizarán, para no tener que indicar en cada clase a que librería pertenece. Al final se dispone de la clase que tiene como nombre *MainActivity* y como se observa, se va a extender a una actividad (*Activity*). Una actividad es el elemento básico de toda interfaz de usuario en Android, se puede decir que es la pantalla de la aplicación.

Dentro de la clase se cuenta con el método *onCreate*, éste es utilizado para reiniciar la actividad, en este caso la interfaz de usuario, para esto el programa indica que se utilizará la vista almacenada en recursos dentro de la carpeta *layout* que tiene el nombre de *activity\_main*, que se muestra en la figura 1.28.

El término *inflate* se refiere comúnmente a tomar elementos existentes definidos como recursos y exponerlos a la vista del usuario.



```

activity_main.xml MainActivity.java
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
    
```

Figura 1.28: Estructura *activity\_main*.

A su vez, dentro del archivo *activity\_main*, como se aprecia en la figura 1.28, se trata de una descripción en código XML que contiene diversas especificaciones *layout* del tipo página web. En este caso lo más relevante es el *Textview* que hace referencia a *@string/hello\_world*, lo que muestra que no se despliega el texto, sino que se hace uso de una cadena almacenada en el recursos que contiene la información a mostrar en pantalla, ver figura 1.29. Las restantes especificaciones del *layout* proporcionan el formato a los elementos que se despliegan en pantalla.



```

activity_main.xml MainActivity.java strings.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">HolaMundo</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>

</resources>
    
```

Figura 1.29: Archivo Strings.

En el archivo *strings.xml*, mostrado en la figura 1.29, que también es un XML, encontramos todas las cadenas de caracteres a utilizar en la aplicación, en el ejemplo sólo se muestra el nombre de la aplicación y el texto que aparece en pantalla. Esto permite una separación entre el texto y el código, para hacer más entendible el programa da una aplicación.

Para acceder a los ficheros contenidos en la carpeta de recursos desde el código del programa, se utiliza la clase *R*, usando la sintaxis que se muestra a continuación.

```

setContentView(R.layout.activity_main)
    
```

Este comando está pidiendo acceso a *activity\_main*, que está contenido en la clase *layout*, que a su vez está contenida en la clase *R*.



Figura 1.30: Clase R.

En la carpeta *gen* se encuentra el archivo *R.java* y como se aprecia en la figura 1.30, la clase *R* contiene las direcciones de las clases catalogadas como recursos *attr*, *drawable*, *layout*, *string* etc. (Tomas J, 2011).

Para el despliegue, depuración y evaluación preliminar de aplicaciones, como ya se mencionó, el SDK de Android cuenta con un emulador de dispositivos virtuales, en donde se simulan terminales con características diferentes tanto en software como de hardware.

Para acceder al emulador se da clic en el icono *Run* que se encuentra en la barra de herramientas y desplegará la pantalla de la figura 1.31.

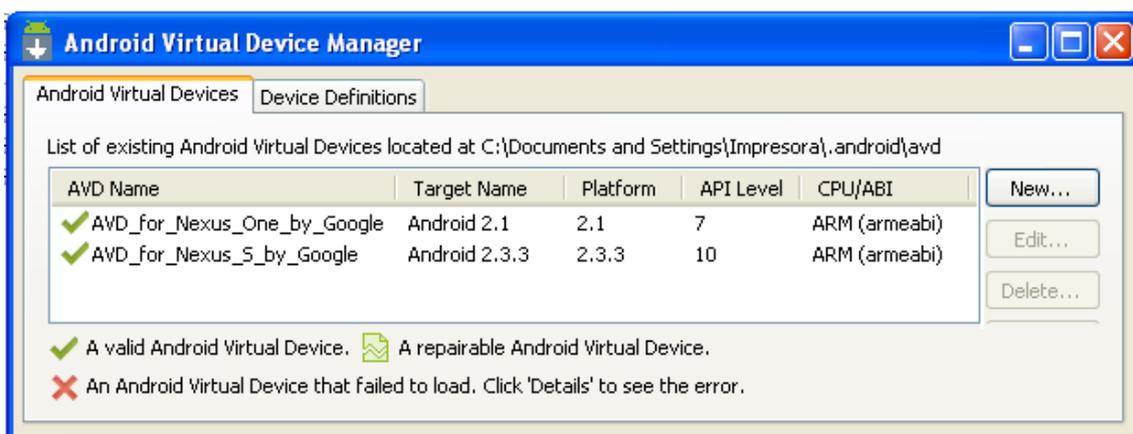


Figura 1.31: Dispositivos virtuales.

Desde el gestor de dispositivos móviles se aprecia una lista de terminales ya configuradas. Para configurar una nueva, se pulsa el botón *New* y se dan valor a los siguientes parámetros que se piden en la ventana emergente de la figura 1.32.

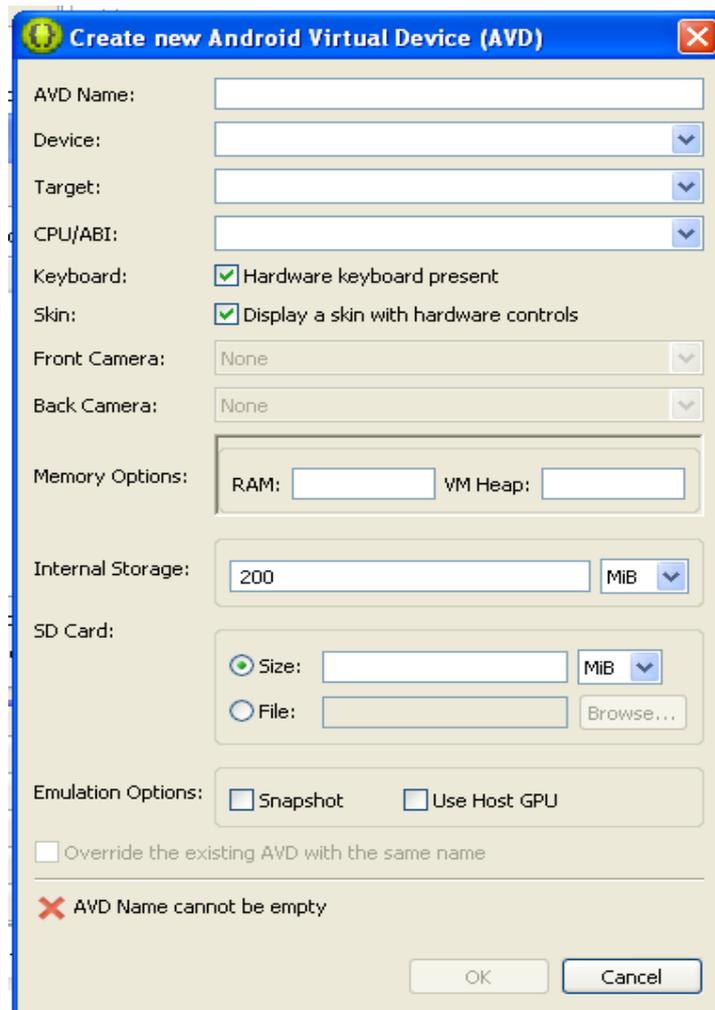


Figura 1.32: Configuración del dispositivo.

- **AVD Name:** Nombre del nuevo dispositivo.
- **Device:** Tipo de dispositivo
- **Target:** Nivel de API que correrá en el dispositivo.
- **CPU/ABI:** Tipo de procesador que incorpora el dispositivo.
- **Keyboard:** Hardware o integrado a la pantalla.
- **Skin:** Opción para asignarle una resolución de pantalla personalizada.
- **Memory options:** Capacidad de memoria RAM
- **Internal Storage:** Cantidad de memoria del teléfono.
- **SD Card:** Tamaño de almacenamiento que tendrá el dispositivo.
- **Snapshot:** Salva el estado en que se quedó el dispositivo antes de cerrarlo. Esto acelera el tiempo de ejecución del emulador.

Una vez configurado el dispositivo, se presiona OK y estará listo para usarse.

Hasta ahora estas son las herramientas principales para el desarrollo de una aplicación, no obstante cabe destacar la herramienta *Outline* también contenida en Eclipse. Esta herramienta se sincroniza con la ventana principal y muestra cómo están organizadas las

clases y funciones en el caso del programa en Java (ver figura 1.33) o los elementos de la vista en el caso del programa en XML (ver figura 1.34).

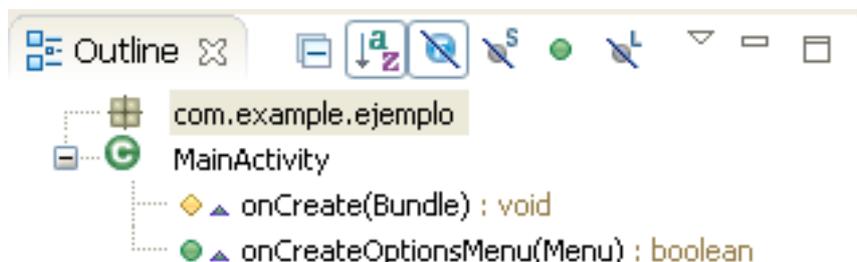


Figura 1.33: *Outline* sincronizado con java.



Figura 1.34: *Outline* sincronizado con XML.

Como se puede ver en las figuras 1.33y 1.34 a cada elemento se le antepone una figura de un color y forma diferente, que depende de las propiedades de estos, como se explica a continuación en las tablas 1.8 y 1.9:

Figura	Descripción
	Indica una clase
	Indica una subclase
	Indica un método privado
	Indica un método público
	Indica un método protegido

Tabla 1.8: Iconos del programa principal en *Outline*.

Figura	Descripción
	Indica una ventana.
	Indica que se inserta una imagen.
	Indica que se inserta un texto.
	Indica que se inserta un custom view.
	Indica que se inserta un botón.
	Indica que se inserta un cuadro de texto.

Tabla 1.9: Iconos de los *layout's* en *Outline*.

*Outline* es muy útil en programas o vistas complejas, porque permite navegar fácilmente entre los elementos del programa (Roberto M, 2012).

### 1.3. Microcontrolador PIC18F97J60

La principal característica de la familia de los microcontroladores PIC 18 de Microchip, es que cuentan con un periférico para comunicación Ethernet de 10 Mbps integrado. Esta integración lo convierte, en el chip ideal para aplicaciones que requieren un control y monitoreo remoto. La figura 1.35, muestra el diagrama electrónico del PIC18F96J60.

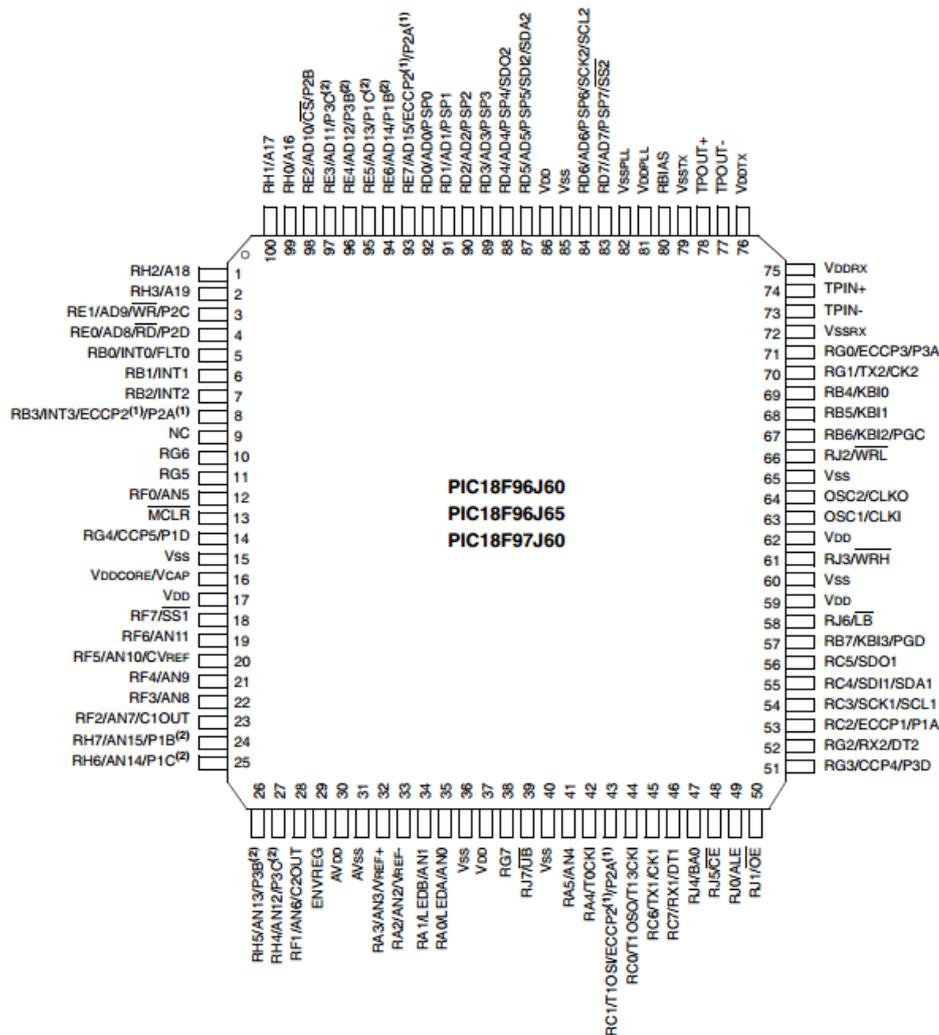


Figura 1.35: Diagrama del microcontrolador.

Las características principales de éste microcontrolador se listan a continuación:

- Frecuencia de operación hasta 25 MHz.
- Hasta 3808 bytes en memoria RAM.
- Hasta 128K bytes en memoria de programa.
- La capacidad de la memoria en instrucciones es de 65532 bytes.

- 29 Interrupciones.
- Los puertos A, B, C, D, E, F, G, H, J son de entrada y salida.
- Tiene 70 pines de entrada y salida.
- 5 timers.
- 2 módulos de Captura/Comparación/PWM.
- módulos de comunicación serial MSSP.
- 2 módulos de comunicación serial USART mejoradas.
- Comunicación con Ethernet mediante la configuración 10-BaseT.
- Comunicación por puerto paralelo en modo esclavo.
- 16 canales ADC de 10 bits cada uno.
- Capacidad de Auto-adquisición.
- Conversión disponible durante el sueño.
- Resets POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT (PWRT, OST).

Las características eléctricas de éste microcontrolador se muestran en la tabla 1.10.

<b>Características eléctricas</b>	<b>Grados máximos absolutos</b>
Temperatura ambiente por debajo a 100° C	
Temperatura de almacenamiento	-65 ° C a +150 ° C
Tensión en cualquier pin de entrada de sólo digital o MCLR con respecto a VSS (excepto VDD)	0.3V a 6.0V
Voltaje en cualquier pin digital y analógica combinada con respecto a VSS	-0.3V a (VDD + 0.3V)
Voltaje en VDDCORE con respecto a Vss	-0.3V a 2.75V
Voltaje en VDD con respecto a VSS	-0.3V a 3.6V
Disipación de energía total	1.0 W
Corriente máxima de salida en el pin VSS	300 mA
Máxima corriente en pin VDD	250 mA
Corriente máxima de salida por cualquier PORTB y PORTC pins de E / S	25 mA
Corriente de salida máxima hundido por cualquier PORTD, PORTE y alfileres PORTJ E / S	8 mA
Corriente de salida máxima hundido por cualquier PORTA, PORTF, PORTG y alfileres Porth E / S	2 mA
Nota 1: Las excepciones son RA0: RA1, que son capaces de conducir directamente a los LED de hasta 8 mA.	

Tabla 1.10: Características eléctricas del microcontrolador PIC18F97J60.

El diagrama de bloques del PIC18f97J60 se muestra en la figura 1.36.

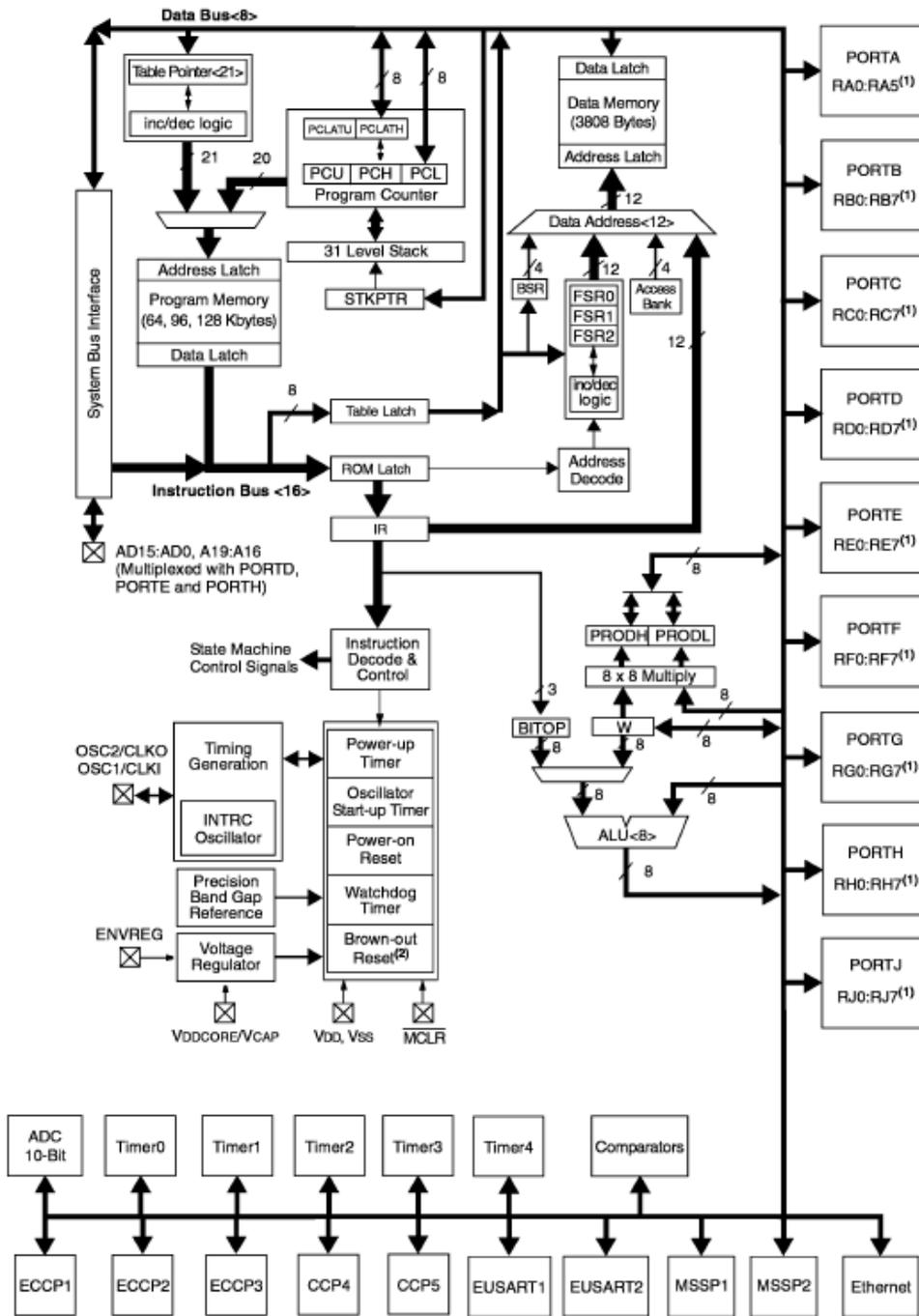


Figura 1.36: Diagrama de bloques.

El mapa de memoria PIC18F97J60 se muestra en la figura 1.37.

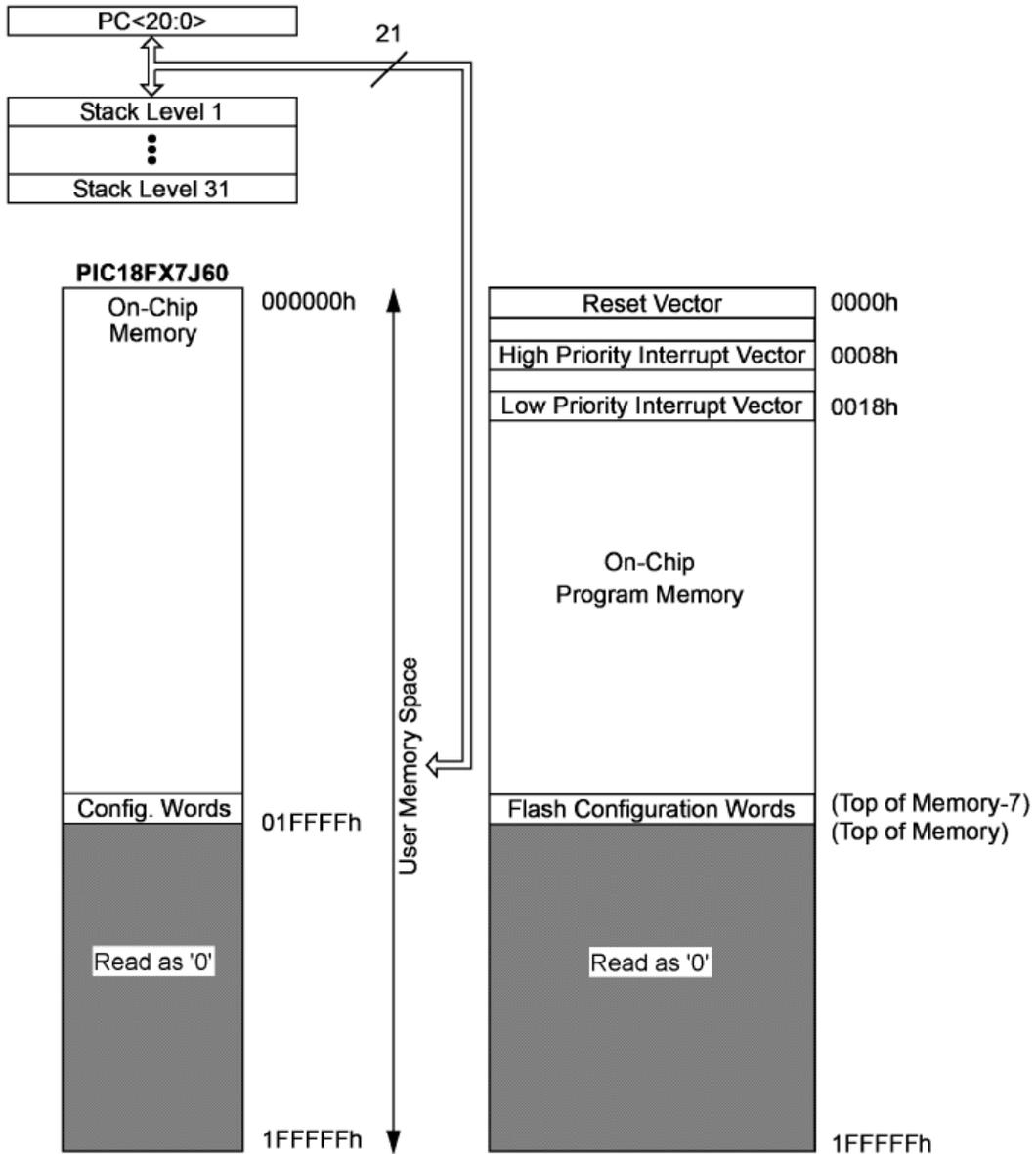


Figura 1.37: Mapa de memoria.

#### 1.4. Herramientas de desarrollo

Como parte del proyecto, se utilizaron las siguientes herramientas para el desarrollo y depuración de aplicaciones.

##### 1.4.1. MPLab

El software que se utilizó para elaborar el programa en el microcontrolador del sistema electrónico desarrollado es MPLab, un ambiente creado por Microchip en el que se puede trabajar tanto en lenguaje C como en ensamblador [García E, 2008]. A continuación se mostrará como configurar el software para crear un nuevo programa.

Se ejecuta el programa y se crea un nuevo proyecto definiendo la versión del microcontrolador PIC que se va a utilizar. Para este caso, es el PIC18F97J60, ver figura 1.38.

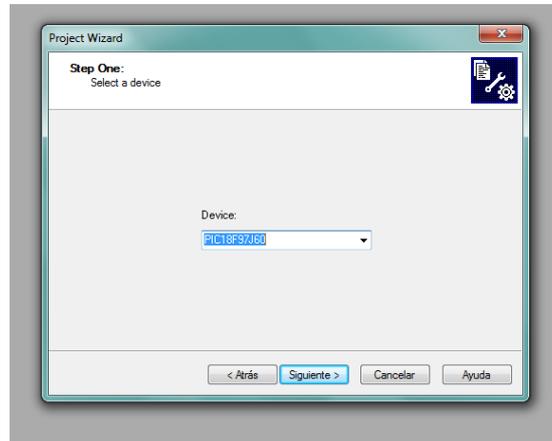


Figura 1.38: Configuración de MPLab.

El siguiente paso es la selección del lenguaje de programación ya sea Assembler o CCS, en este proyecto de tesis se utilizó CCS, ver figura 1.39.

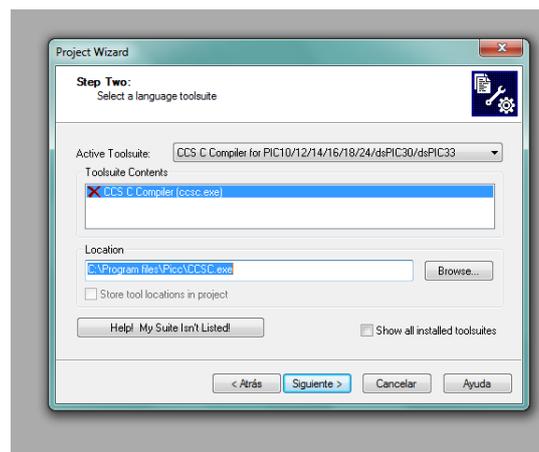


Figura 1.39: Configuración de MPLab.

Hecho esto, se configura el dispositivo para realizar la compilación con la opción del menú *Debugger/Select Tool*, ver figura 1.40.

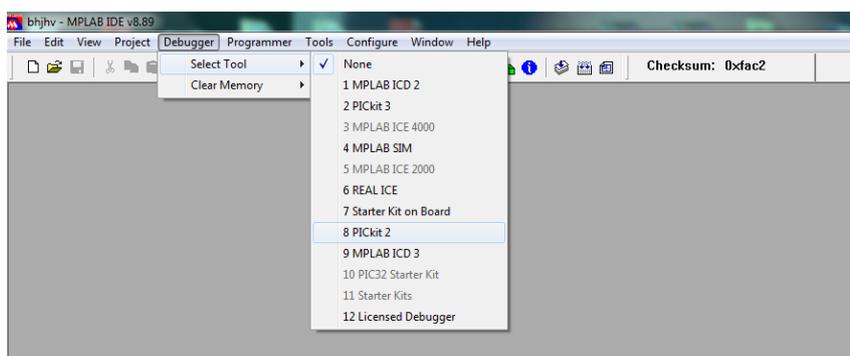


Figura 1.40: Configuración de MPLab.

Las opciones del menú *Debugger* y *Programmer* insertan iconos que permiten compilar, programar, leer, borrar y ejecutar el programa con herramientas como *PICkit2*, también utilizada en este proyecto de tesis.

Finalmente se crea el proyecto de programación como el mostrado en la figura 1.41.

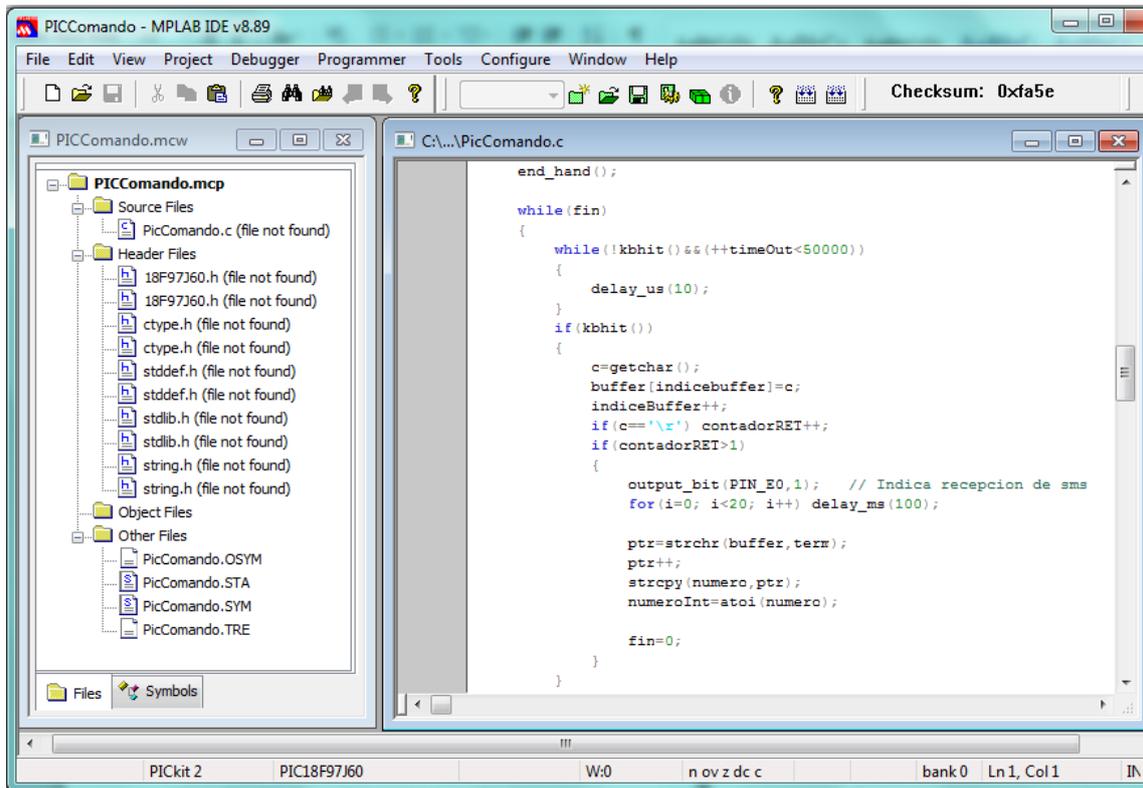


Figura 1.41: Proyecto de MPLab.

El ambiente consta de una ventana principal donde se escribirá el programa y una ventana secundaria donde se encuentran las librerías para la interpretación de las instrucciones. MPLAB cuenta con las herramientas típicas para el desarrollo de programas y no es necesario tener el dispositivo conectado mientras se depura el programa.

#### 1.4.2. PIC-GSM

La principal herramienta de desarrollo para este proyecto es el módulo PIC-GSM. Este módulo permite desarrollar aplicaciones de monitoreo y control remoto usando las redes celular GSM y Ethernet, ver figura 1.42.

Con el PIC-GSM se pueden monitorear y controlar de manera remota variables desde cualquier lugar mediante un teléfono celular.

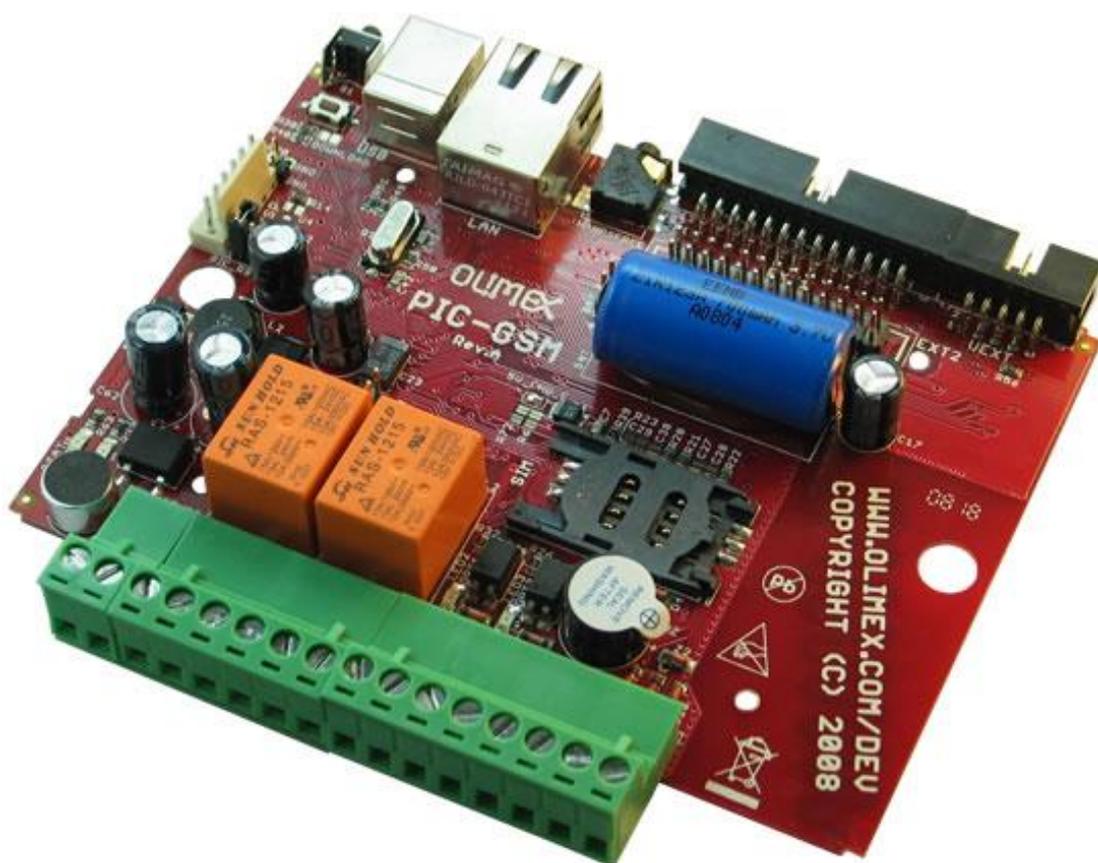


Figura 1.42: PIC-GSM OLIMEX.

Este módulo contiene un conector ICSP / ICD para la programación y la depuración con PICICD2, PIC-ICD2-POCKET, PIC-ICD2-TINY, conector USB 2.0 tipo B para ser conectado a la PC anfitriona, batería Li-ion de reserva, PIC18F97J60-I/PT, conector Ethernet RJ45 aislado, audio GSM de entrada y salida, Conector RS232, cristal de cuarzo de 20 Mhz, dos relés 10A/250VAC, dos entradas aisladas ópticamente, sensor de temperatura, regulador de voltaje de 5V, conector EXT disponibles para GPIO. Las dimensiones de la tarjeta son 124x90 mm o 4,880 x 3,540 milésimas de pulgada.

El circuito de alimentación se puede conectar de dos formas diferentes:

- 1) Conectado a una fuente de 12VDC sin utilizar la batería interna de reserva. En este caso el jumper BAT\_E debe de estar abierto, el 4V\_E debe de estar cerrado, 4V\_VBAT debe de estar cerrado y el 5V\_CHG\_E debe de estar abierto.
- 2) Conectado a una fuente de 12VDC, con batería interna de respaldo. En este caso el jumper BAT\_E debe de estar cerrado, 4V\_E debe de estar abierto, 4V\_VBAT debe de estar abierto y el %V\_CHG\_E debe de estar cerrado.

La descripción de los conectores disponibles en el PIC-GSM se presenta a continuación.

### Conector ICSP

El ICSP mostrado en la figura 1.43, permite programar y depurar a través de PIC-ICD2, PICICD2-POCKET, PIC-ICD2-TINY y otras herramientas compatibles Microchip.

Pin #	Signal Name
1	RST
2	VCC
3	GND
4	PGD
5	PGC
6	NC

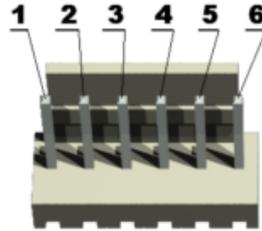


Figura 1.43: Conector ICSP.

### Conector USB

El conector USB mostrado en la figura 1.44, está bajo el estándar USB tipo B. Este está conectado al circuito FT232RL que es un convertidor de comunicación USB a UART. El microcontrolador PIC18F97J60 puede controlar si la comunicación será entre la terminal PC y el módulo GSM o entre el PIC18F97J60 y el módulo GSM.

Pin #	Signal Name
1	VCC
2	USBDM
3	USBDP
4	GND



Figura 1.44: Conector USB.

### Conector SIM

La figura 1.45 muestra al conector para tarjeta SIM estándar, que permitirá operar el módulo PIC-GSM, debe de ser una tarjeta SIM válida para la red del operador.

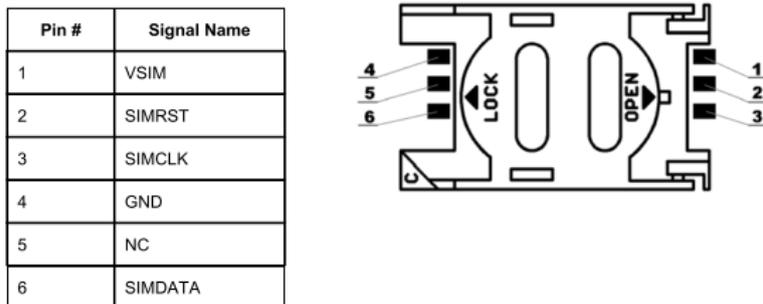


Figura 1.45: Conector de la tarjeta SIM.

### Conector de polarización

La figura 1.46, muestra las terminales de polarización del módulo PIC- GSM en el conector PWR.

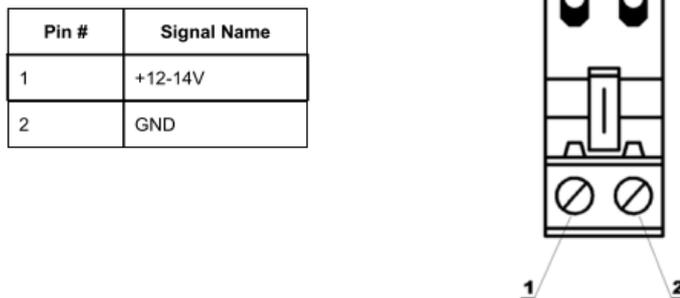


Figura 1.46: Conector PWR.

### Conector para los relevadores

La figura 1.47, indica cómo están ordenadas las salidas de los dos relevadores que tiene la tarjeta.

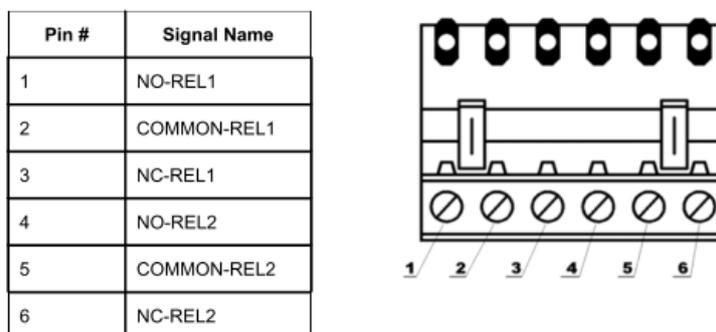


Figura 1.47: Salida de los relevadores.

En donde:

NO – REL, contacto normalmente abierto

NC – REL, contacto normalmente cerrado

COMMON - Relé de contacto común

Los rangos máximos permitidos son: 15A/125VAC, 10A/250VAC y 15A/24VDC.

### Conector de entradas opto-acopladas

La figura 1.48, muestra cómo están ordenadas las 2 entradas ópticamente aisladas. El nivel de voltaje de entrada debe de estar entre 5 y 12 VDC.

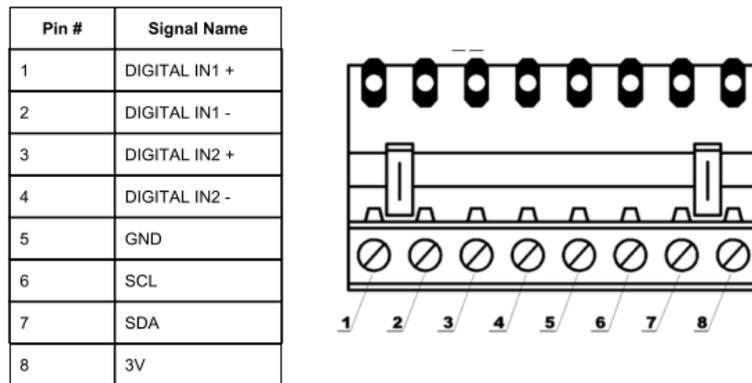


Figura 1.48: Entradas opto-acopladas.

### Conector de audio

En la figura 1.49, se muestra al conector de audio de 2.5 mm, para las funciones de audio y micrófono combinado o manos libres.

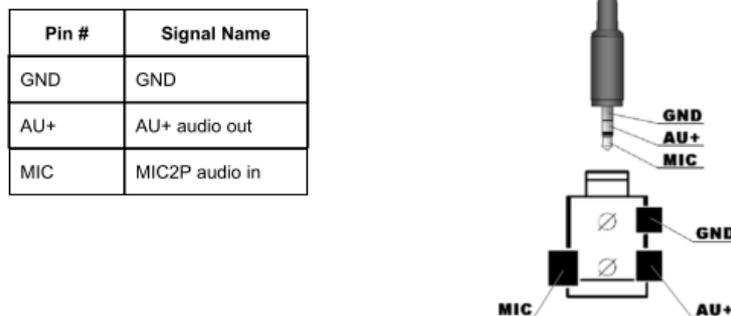


Figura 1.49: Manos libres.

### Conector del altavoz

La figura 1.50, muestra la ubicación de las terminales para el altavoz externo que debe ser de 32 ohms.

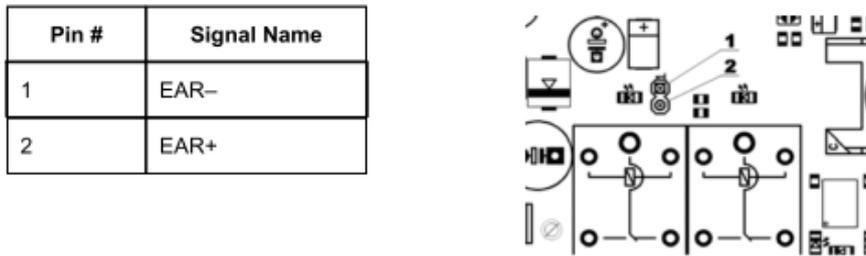


Figura 1.50: Conector para el alta voz.

### Conector EXT 1

En la figura 1.51, se muestra la función de las terminales de la extensión 1, EXT 1. En la tabla 1.9 se describe detalladamente cada una de las funciones.

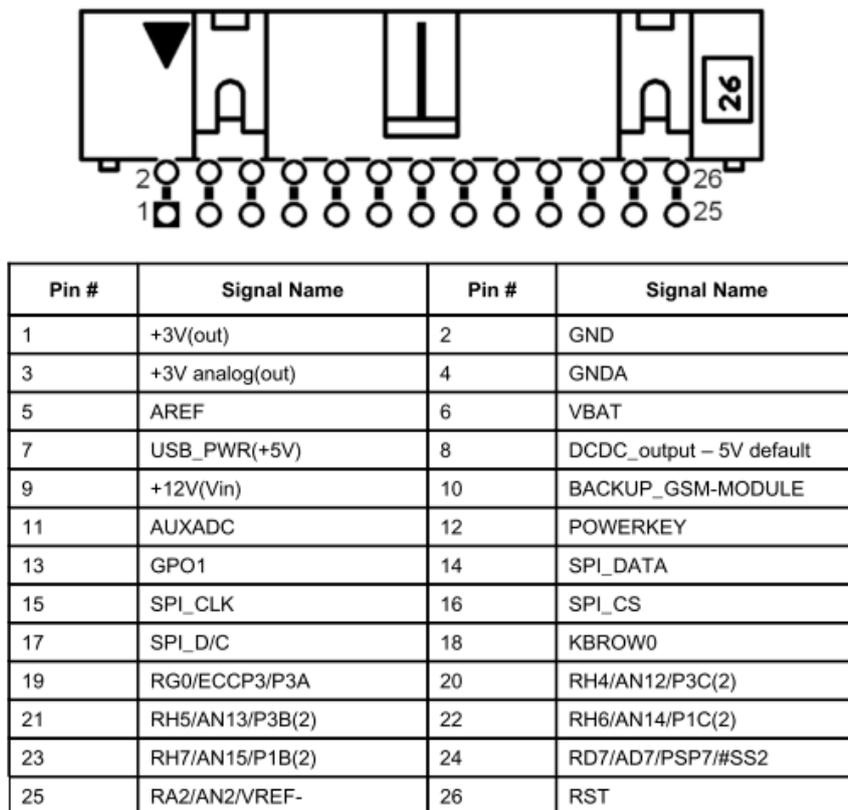


Figura 1.51: Conector EXT 1.

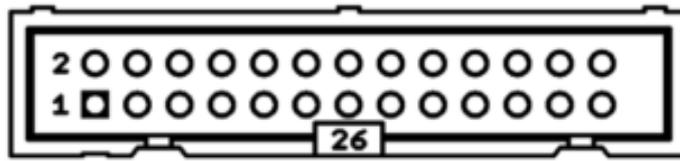
Nombre	Descripción
3V	Esta es la salida 3VDC para los módulos digitales externos
GND	Tierra digital
3VA	Esta es la salida 3VDC que puede ser

	utilizada para los módulos analógicos externos
GNDA	Puede ser utilizado para la grabación analógica de circuitos externos
AREF	Entrada de referencia analógica del microcontrolador PIC18F97J60
VBAT	Dedicado a conectar la batería principal de iones de litio
USB -PWR	Salida de 5 V directamente conectada a +5 V del conector USB
DCDC _output-5V default	Salida de 5V o 4 V (hasta 2 A ) del convertidor DCDC dependiendo del puente de configuración
Backup	Fuente de alimentación de respaldo de RTC para el módulo de reloj en tiempo real GSM y RAM
AUXADC	Auxiliar del convertidor analógico a digital
PowerKey	Botón de encendido / apagado del módulo GSM. Cuando el módulo está encendido, si pulsa durante más de 3 segundos, el módulo se apagará. Si el módulo está apagado y se mantiene pulsado este botón durante más de 1 segundo, el módulo se encenderá. <b>GPO1:</b> puerto de entrada y salida del chip sim300
SPI_DATA SPI_CLK SPI_CS SPI_D/C	Esto se reserva puerto SPI del módulo GSM
KBROW0	Pin de entrada de teclado externo del módulo GSM
RG0/ECCP3/P3A RG0/ECCP3/P3A RH5/AN13/P3B RH6/AN14/P1C RH7/AN15/P1B RD7/AD7/PSP7/# SS2 RA2/AN2/VREF	Son pines del PIC18F97J60
RST	Pin de reset del PIC18F97J60

Tabla 1.11. Descripción del conector EXT 1.

### Conector EXT 2

La figura 1.52, muestra a la extensión 2, EXT 2, que es un conector que está encaminado al resto de los pines del PIC.



Pin #	Signal Name	Pin #	Signal Name
1	RE0/AD8/#RD/P2D	2	RE1/AD9/#WR/P2C
3	RE2/AD10/#CS/P2B	4	RE3/AD11/P3C(2)
5	RE4/AD12/P3B(2)	6	RE5/AD13/P1C(2)
7	RE6/AD14/P1B(2)	8	RE7/AD15/ECCP2(1)/P2A(1)
9	RF0/AN5	10	RF1/AN6/C2OUT
11	RF2/AN7/C1OUT	12	RF3/AN8
13	RF4/AN9	14	RF5/AN10/CVREF
15	RF6/AN11	16	RJ0/ALE
17	RJ1/#OE	18	RJ2/#WRL
19	RJ3/#WRH	20	RJ4/BA0
21	RJ5/#CE	22	RJ6/#LB
23	RJ7/#UB	24	PB3/INT3/ECCP2(1)/P2A(1)
25	RB4/KB10	26	RB5/KB11

Figura 1.52: Conector EXT 2.

### Conector UEXT

La terminal UEXT mostrada en la figura 1.53, es un conector universal con fuente de alimentación 3,3 V, UART, SPI y la interfaz I2C.

Pin #	Signal Name
1	VCC
2	GND
3	TX2
4	RX2
5	SCL
6	SDA
7	SDI1
8	SDO1
9	SCK1
10	SS1

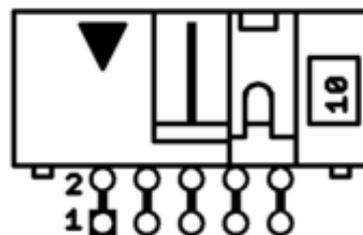


Figura 1.53: Conector UEXT.

## 2. Descripción del proyecto

En este trabajo se propone desarrollar hardware y software para implementar un sistema de control remoto con base en mensajes SMS utilizando la red GSM para controlar un sistema electrónico mediante un teléfono celular.

### 2.1. Esquema general

El esquema consta de dos partes una aplicación desarrollada en Android para dispositivos móviles (Smartphone) y un sistema embebido, que tiene como principales funciones controlar salidas digitales y proporcionar datos analógicos. Este sistema de control remoto utiliza la red GSM para mandar mensajes desde el dispositivo móvil a un sistema electrónico y viceversa.

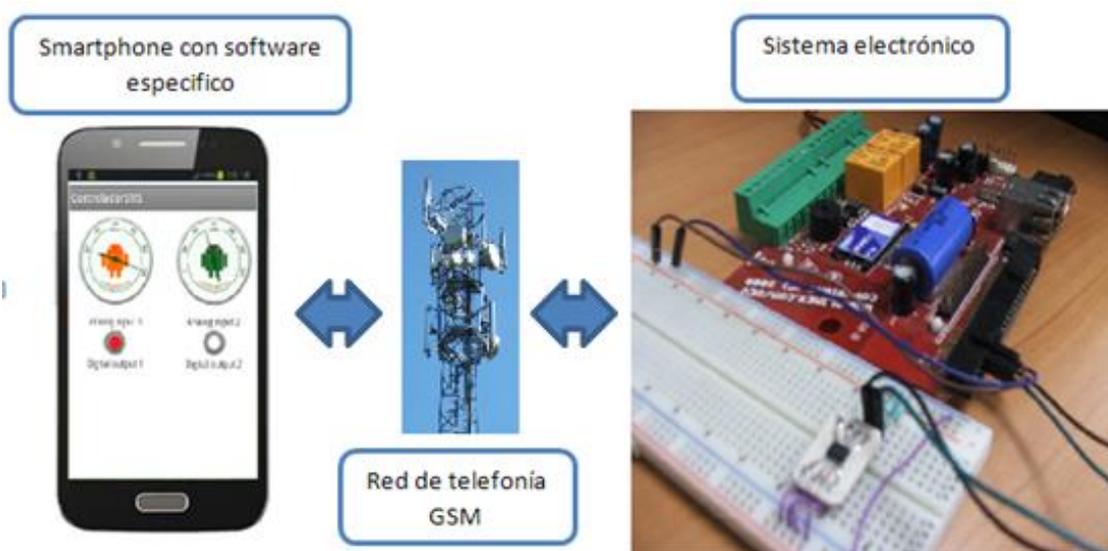


Figura 2.1: Esquema general para el control remoto con base en mensajes SMS.

En la figura 2.1, se muestra el esquema del sistema completo en el cual una aplicación desarrollada en Android realiza las siguientes funciones.

- Al presionar botones específicos, genera y envía los mensajes SMS para instruir al sistema electrónico para activar o desactivar las salidas digitales 1 y 2 respectivamente. Espera la respuesta de confirmación conteniendo los valores actualizados de las salidas digitales y los despliega en pantalla en forma de indicadores digitales: encendido=rojo, apagado=blanco, indeterminado=?
- Al presionar botones específicos, genera y envía los mensajes SMS para instruir al sistema electrónico para digitalizar las entradas analógicas 1 y 2 respectivamente. Espera la respuesta de confirmación conteniendo los valores digitalizados y los despliega en pantalla en forma de indicadores analógicos.
- Al presionar opciones específicas del menú, da de alta o baja a usuarios registrados por el sistema electrónico.
- Al presionar la opción específica del menú realiza una llamada de voz para escuchar lo que sucede en la localidad del sistema electrónico.

El microcontrolador en el sistema electrónico se comunica con un circuito integrado SIM300 por medio de un enlace RS232 y comandos en protocolo AT. El circuito integrado SIM300 es el encargado de la etapa de radiofrecuencia para el servicio de mensajes SMS. El programa en el microcontrolador, desarrollado en lenguaje C para PIC, espera a recibir mensajes SMS para de ahí extraer el número de celular que le envió el mensaje y el contenido del mensaje que le indicará la acción a ejecutar.

Las entradas analógicas pueden representar cualquier variable por determinar que esté en el alcance del sistema electrónico, por ejemplo temperatura, presión, PH, etc. De igual manera, las salidas digitales pueden representar cargas que se deseen activar o desactivar como motores, lámparas, bombas de agua, resistencias calefactoras, etc.

### 2.1.1 Esquema para la programación del sistema electrónico

El software del micro controlador realizo en MPLab, este se instaló en la memoria del pic por comunicación I2C con ayuda del PIC-kit2 el cual se conecta a través del puerto ICSP de la tarjeta electrónica. Ver figura 2.2.

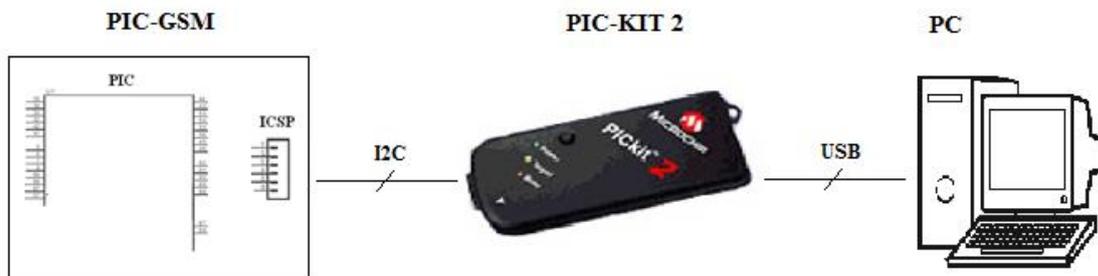


Figura 2.2: Esquema general para el control remoto con base en mensajes SMS.

### 2.1.2 Esquema para la depuración del programa en el sistema electrónico

Para depurar el programa se habilito el convertidor serie-USB en forma de esclavo, el cual está conectado en serie al chip sim300 y en forma paralela, al micro controlador esto permite convertir los datos de la comunicación entre el sim300 y el PIC, de protocolo serie a USB para que puedan ser observados por la hyperterminal de Windows. Ver figura 2.3.

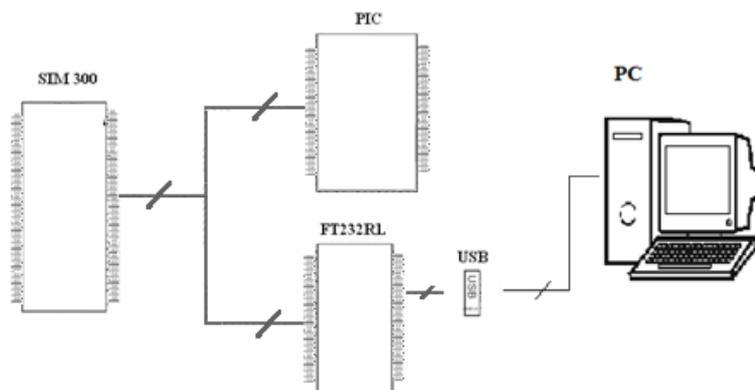


Figura 2.3: Esquema general para el control remoto con base en mensajes SMS.

## 2.2. Sistema electrónico

El sistema electrónico cuenta con una etapa de comunicaciones, una de control de actuadores y una etapa de monitoreo de señales como muestra la figura 2.4. El sistema electrónico esta sobre la base de la tarjeta PIC-GSM descrita en la sección 1.4.2.

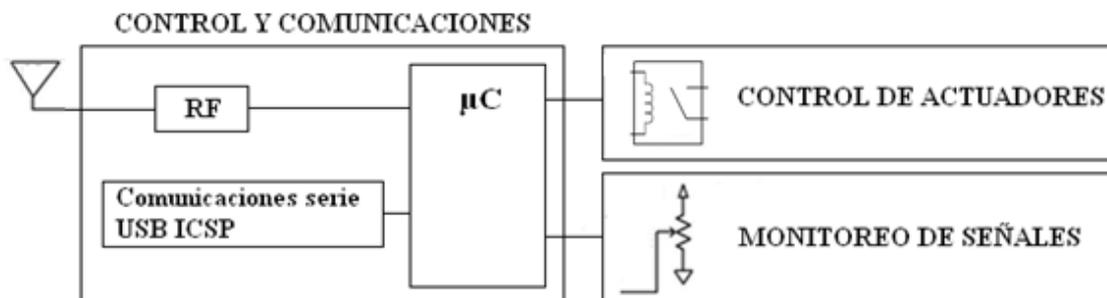


Figura 2.4: Sistema eléctrico

La etapa de comunicaciones está compuesta por un microcontrolador PIC18F97J60 que controla la parte de radio frecuencia, la cual permite la comunicación a través de la red GSM.

Esta etapa también cuenta con los circuitos necesarios para convertir de USB a RS232 y viceversa, utilizada con fines de depuración desde una PC.

Los actuadores en este caso son 2 relevadores que se encuentran integrados en la tarjeta PIC-GSM. Los actuadores tienen integrados 2 leds que se utilizan para observar el estado actual en el relevador.

Las señales de entrada son los valores de la tensión en 2 potenciómetros externos que se digitalizan en el microcontrolador.

A continuación una descripción detallada de las etapas del sistema electrónico mostrado en la figura 2.4.

### 2.2.1. Etapa de control y comunicaciones

En la figura 2.5 se muestra la conexión entre el PIC y las partes principales que conforman la etapa de control.

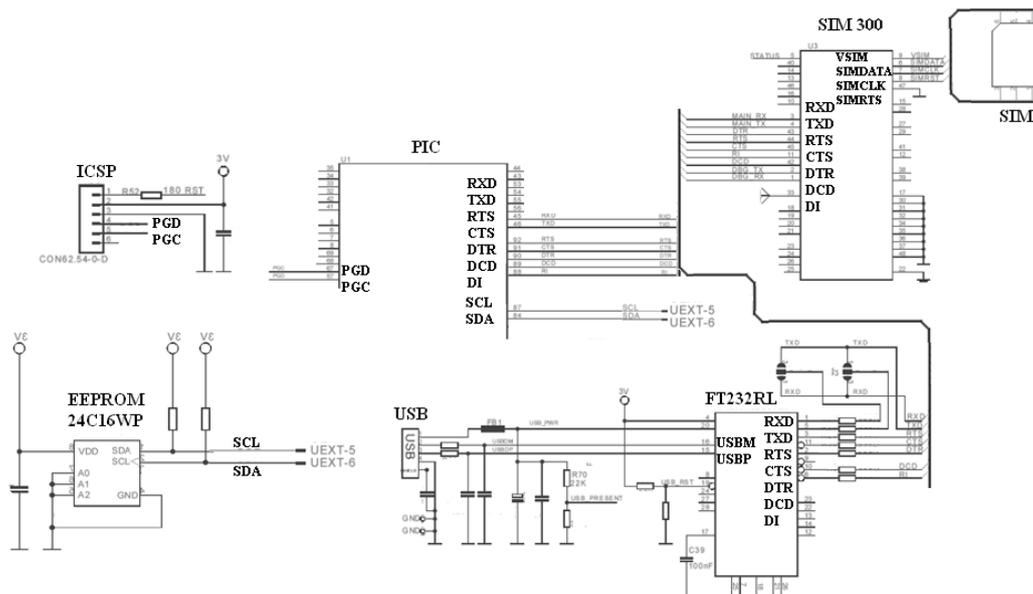


Figura 2.5: diagrama del sistema electrónico.

La etapa de radio frecuencia está compuesta por el chip SIM300 [Sim com, 2005] y una tarjeta SIM normal, en este caso de la marca telcel. El chip de telcel almacena la información del usuario como por ejemplo el saldo, mientras que la sim300 permite que el sistema de control, se comuniquen con SIM y la red de telefonía móvil.

El chip SIM300 puede ser controlado internamente por medio del PIC18F97J60 por medio de los puertos C y D o externamente mediante un ordenador a través del puerto USB. La conexión entre el PIC y el SIM300 es mediante el protocolo RS232, por lo que es necesario pasar de USB a RS232, para esto el modulo cuenta con el chip FT232RL [FTDI chip, 201]. Para evitar problemas en la comunicación, el PIC puede interrumpir el funcionamiento del chip FT232LR por medio de los pines denominados USB\_RTS y el USB\_PRESENT. En el anexo C se muestra la descripción de los pines del chip SIM300.

El sistema electrónico incluye una memoria EEPROM de 16 Kbits 24C16WP [ST, 1999] para almacenar los números telefónicos de los usuarios autorizados para operar el sistema. La memoria se conecta al microprocesador por medio de la interface serial I2C. El bloque ICSP nos permite grabar el programa en la memoria flash del PIC, a través de comunicación serie.

### 2.2.2. Etapas de monitoreo de señales

Para el monitoreo se utilizó el convertidor analógico a digital de 10 bits, que contiene el PIC. Los pines utilizados fueron RF0 Y RF1, accediendo a ellos a través de las extensiones EXT 2 pin 9 y EXT 2 pin 10, como se muestra en la figura 2.6.

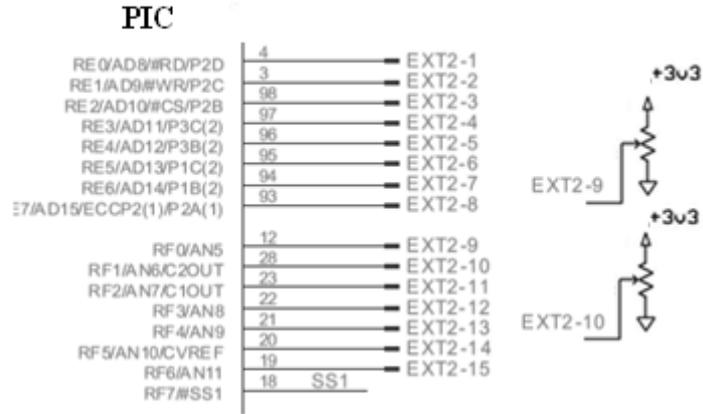


Figura 2.6: Monitoreo de señales

### 2.2.3. Etapas de control de actuadores

En esta etapa se controlan 2 relevadores los cuales están conectados al puerto G en los pines R5 y R6, la figura muestra las conexiones entre el PIC y el relevador.

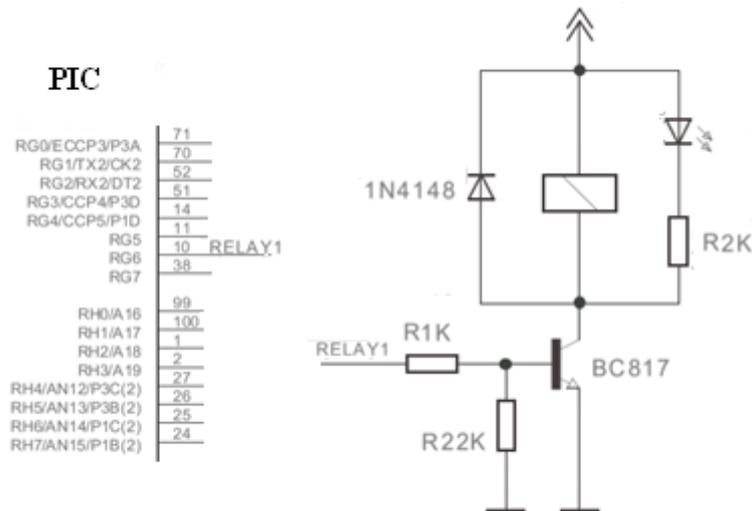


Figura 2.7: Control de actuadores

El transistor BC817 se encuentra en configuración pull up, lo cual sirve para adecuar la tensión a la salida de un circuito lógico a la carga. Como el relevador es una carga inductiva, se conecta un diodo en paralelo a este, para evitar un pico en la tensión de salida lo cual le causaría daños al transistor.

### 2.2.4. Algoritmo de monitoreo, control y comunicaciones

El programa almacenado en la memoria del microcontrolador PIC18F97J60 consiste básicamente en controlar un chip SIM300 mediante un microcontrolador PIC con ayuda

del protocolo RS232 y comandos AT. La figura 2.8 muestra el diagrama de flujo del programa en el PIC.

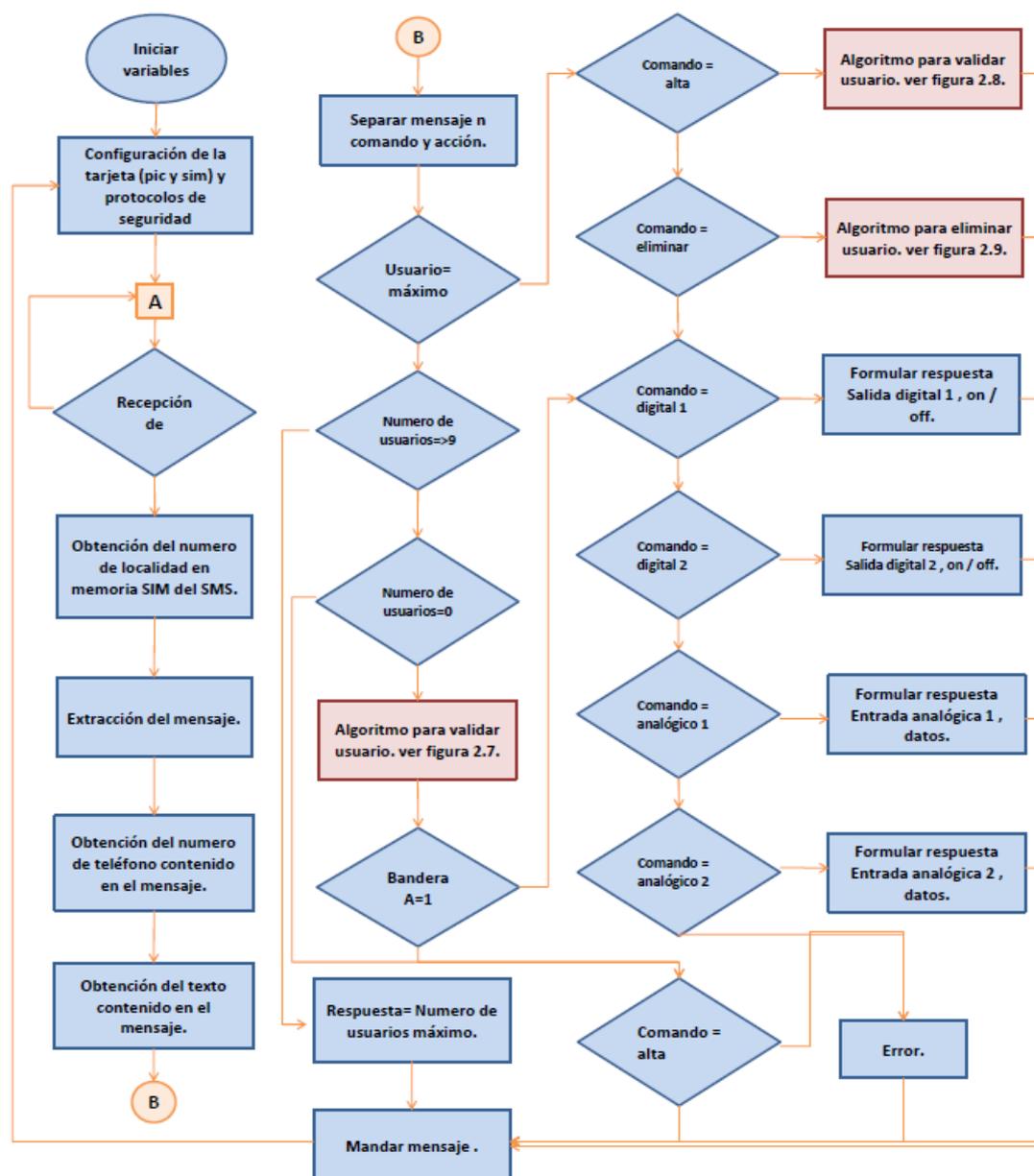


Figura 2.8: Diagrama de flujo controlador SMS

El programa en el microcontrolador espera a recibir la indicación de que arribó un mensaje, cuando esto sucede extrae el número de localidad donde se encuentra almacenado el mensaje, llama a todo lo almacenado en esa localidad y separa el número telefónico y el cuerpo del mensaje. Posteriormente se procesa el texto para extraer dos cadenas claves *Comando* y *Acción*: *Comando* contiene el proceso a realizar y *Acción* valores que se requieren durante el proceso. Al final del proceso seleccionado, se formula la respuesta que se enviara al usuario que solicito la información.

El programa principal utiliza tres subrutinas importantes que han sido resaltadas en la figura 2.8. La descripción detallada de estas tres subrutinas se presenta a continuación.

### Diagrama de flujo para dar de alta a un usuario

Para habilitar a un usuario se necesita guardar su número de celular en la memoria externa mediante I2C. Para que el algoritmo cumpla con el proceso son necesarios dos valores: el número del celular a guardar y el número de usuarios registrados, ver figura 2.9.

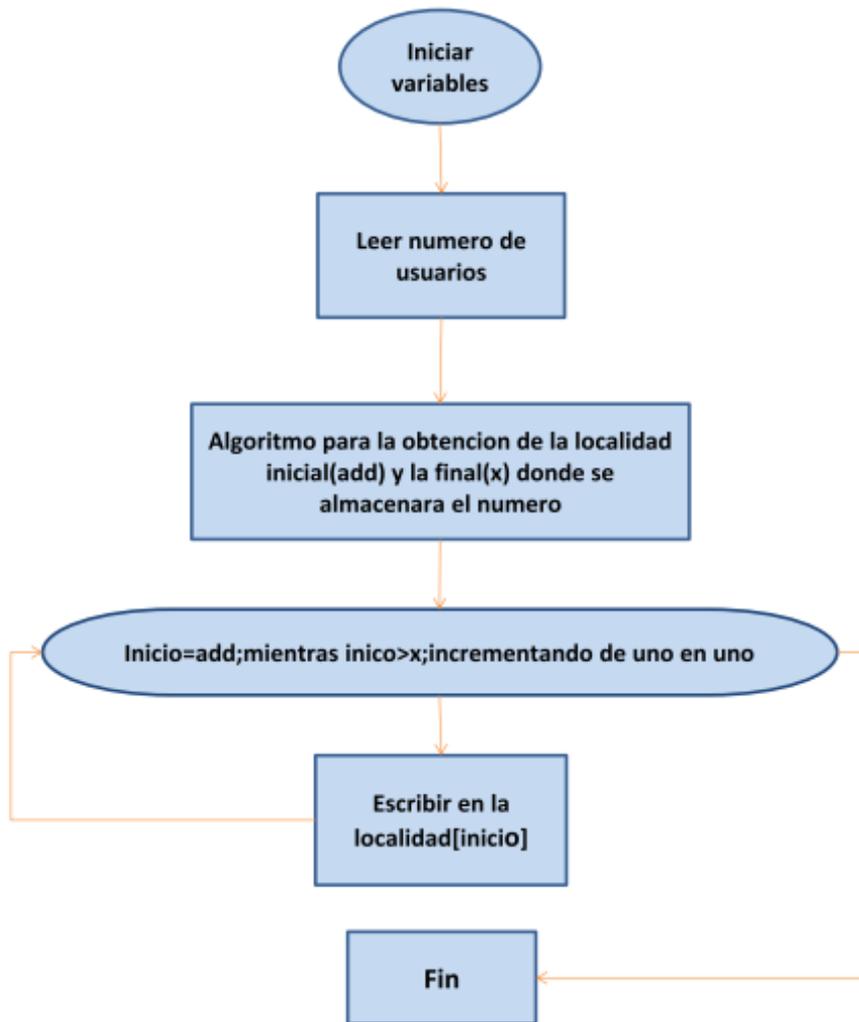


Figura 2.9: Algoritmo para habilitar a un usuario.

El algoritmo escribe al nuevo usuario a partir del inicio calculado a partir de las localidades de memoria previamente ocupadas por otros usuarios registrados. El último registro escrito en memoria se calcula a partir de la longitud de la cadena que representa el número del celular del nuevo usuario.

### Diagrama de flujo para validar a un usuario

Para validar un usuario se compara uno por uno el número de celular entrante con los números almacenados en la memoria externa; si el número coincide con alguno de los almacenados, se activa una bandera especial para este proceso que le permitirá el acceso a diferentes comandos, figura 2.10.

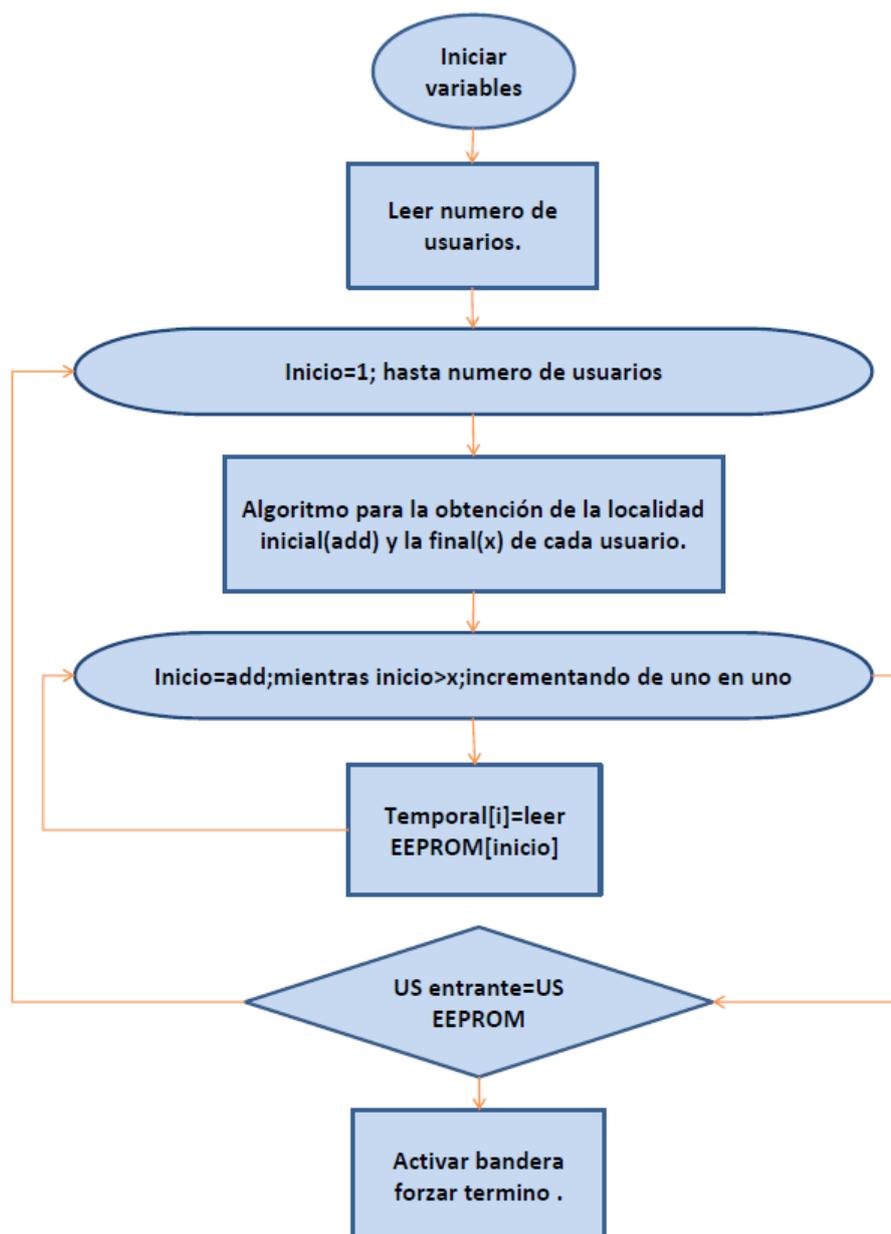


Figura 2.10: Algoritmo para validar a un usuario.

Si alguno de los caracteres que componen el número del usuario a validar es diferente al número de los usuarios registrados, entonces se obliga el término de la búsqueda y se procede con otro número de usuario registrado.

**Diagrama de flujo para eliminar a un usuario**

La función para eliminar un usuario solo está permitida para el usuario de mayor jerarquía, esta requiere el número de celular a eliminar, posteriormente el algoritmo busca el número de usuario en memoria que representa a este celular, si es un número valido se activa una bandera denominada EUS y se guardan los valores necesarios para el siguiente proceso. En el siguiente proceso se sobrescribe el número de teléfono que se encuentra en la localidad superior del número a eliminar y así sucesivamente hasta llegar el último usuario registrado en memoria, ver figura 2.11.

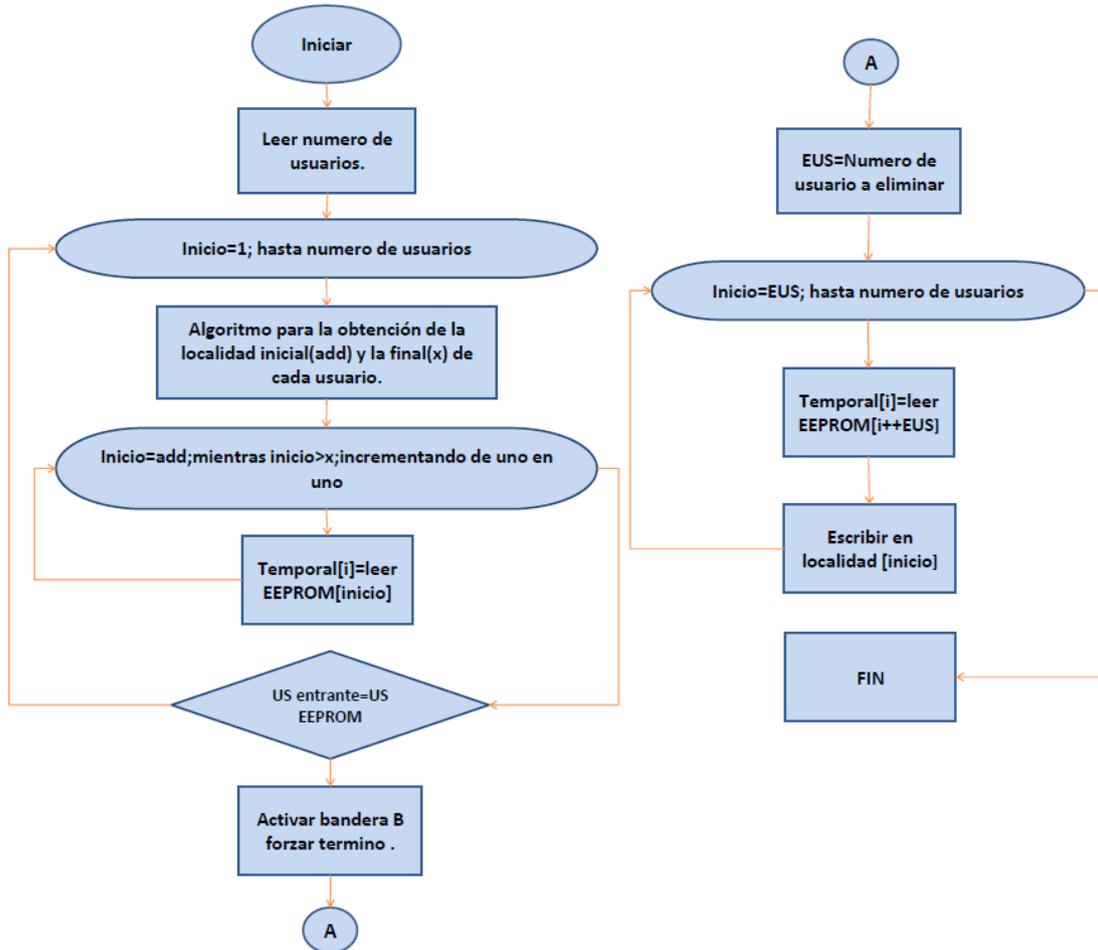


Figura 2.11: Algoritmo para eliminar usuario.

**2.3. Software de operación**

En la presente sección se hace referencia al software de operación desarrollado para dispositivos Android. Se explica cómo fue desarrollado y su descripción en los ámbitos de operación y programación.

**2.3.1. Esquema general**

El software de operación es ejecutable desde dispositivos Android. Se utilizó lenguaje Java para su desarrollo y Eclipse como ambiente integrado de desarrollo. El software puede ser depurado en el emulador incluido con Eclipse o directamente en un

dispositivo Android conectado al puerto USB de la computadora. Al final del proceso, el software de operación es cargado directamente en un dispositivo Android 2.3 o superior, como se muestra en la figura 2.12.

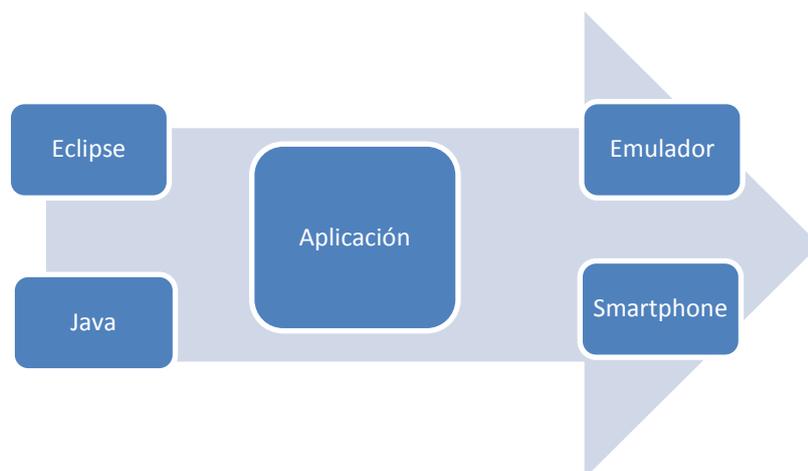


Figura 2.12: Esquema general del software de operación.

### 2.3.2. Descripción del software en el ámbito del operador

La pantalla principal de la aplicación cuenta con 2 despliegues analógicos y dos círculos que muestran las salidas digitales. Al iniciar la aplicación no se muestran las agujas de los indicadores analógicos y los indicadores digitales muestran un signo de interrogación, debido a que se desconoce el estado de estas variables en el sistema remoto, ver figura 2.13.



Figura 2.13: Pantalla de la actividad principal.

Al oprimir el botón *MENU*, figura 2.14, se muestra una pantalla con las opciones que se pueden realizar, las cuales son entrada analógica uno y dos, prender salida digital 1 o 2, apagar salida digital 1 o 2, eliminar usuario, dar de alta usuario y realizar llamada. Cuando se selecciona alguna de las opciones, se genera un mensaje en la pantalla principal, notificando que se ha mandado un mensaje al sistema electrónico como se muestra en la figura 2.15.

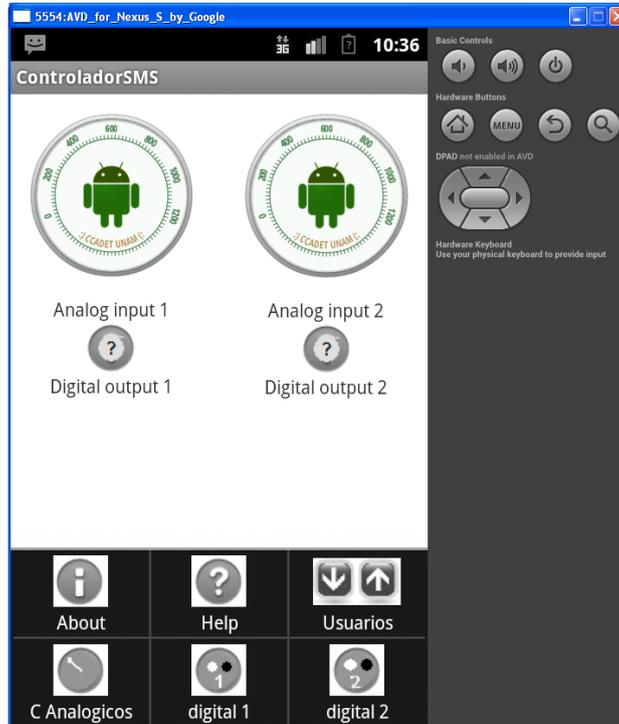


Figura 2.14: Menú.



Figura 2.15: Mensaje enviado

Para dar de alta o de baja a un usuario, se tiene que ingresar el número de teléfono del usuario, como se muestra en la figura 2.16. Al oprimir *Send*, la aplicación mostrará un mensaje informando lo que se ha solicitado: alta de usuario figura 2.17 o baja de usuario figura 2.16. Si todo está correcto y no ha sucedido ningún error, se mostrará un segundo mensaje en la pantalla principal como se ilustra en las figuras 2.17 y 2.18, de lo contrario se mostrara el mensaje “comando no valido” que se presenta en la figura 2.19.

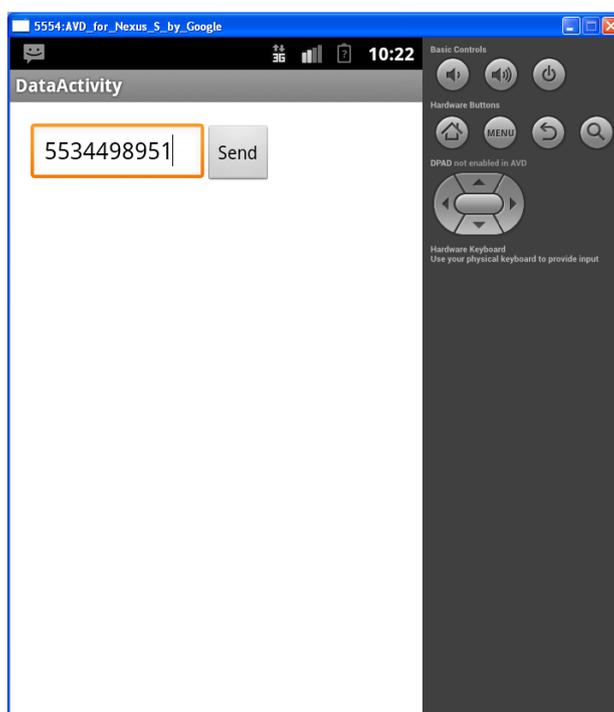


Figura 2.16: Pantalla secundaria.

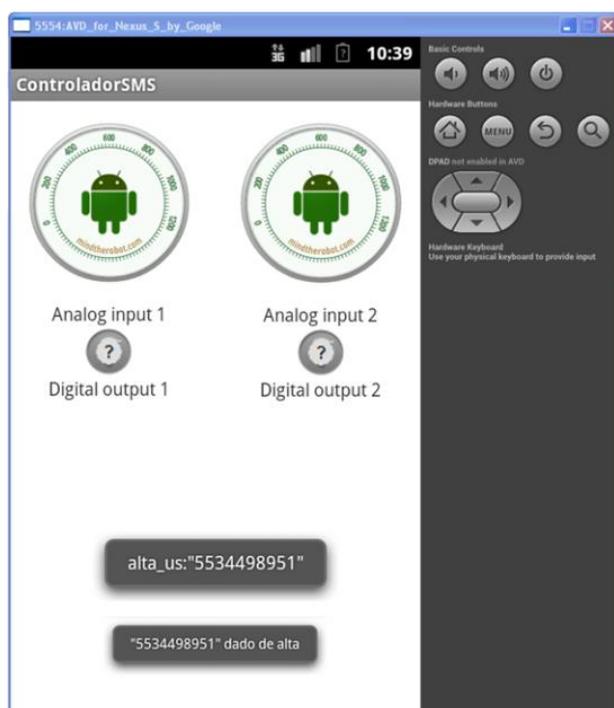


Figura 2.17: Mensaje alta usuario.



Figura 2.18: Mensaje elimina usuario.



Figura 2.19: Mensaje comando no valido.

En la figura 2.20, se muestran las 4 funciones principales de la aplicación; en los indicadores digitales el color rojo indica encendido y el blanco apagado, los indicadores analógicos cambian de color dependiendo del valor recibido, aparte estos cuentan con una función de acercamiento o zoom como se muestra en la figura 2.21. El color azul en los despliegues analógicos indica la cercanía al valor mínimo o cero. Por otra parte, el color rojo indica la cercanía al valor máximo o  $2^{10}=1024$ .



Figura 2.20: Aplicación en funcionamiento.

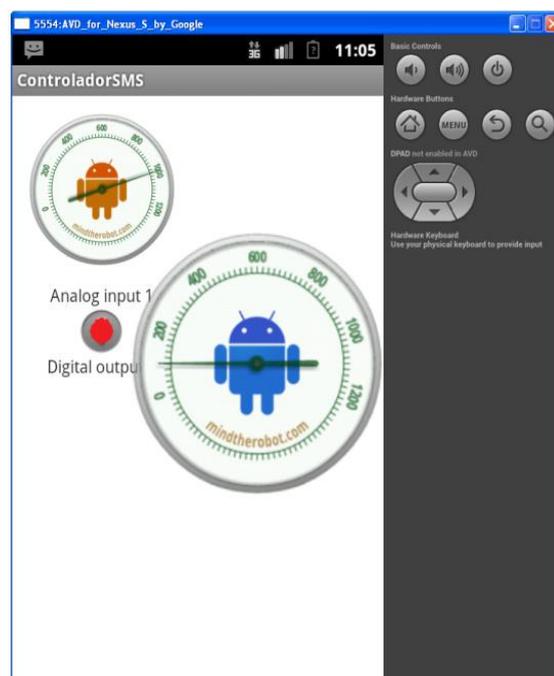


Figura 2.21: Zoom.

### 2.3.3. Descripción del software en el ámbito del programador

Como ya se mencionó en la sección 1.2, un proyecto Android está compuesto de diferentes carpetas para separar los diferentes archivos que se utilizan al crear la aplicación. En esta sección se explica cómo está compuesto y que función realiza cada componente en la aplicación desarrollada para este proyecto de tesis.

El programa principal se encuentra contenido en *com.example.controladorsms* y está compuesto principalmente por cuatro *clases*, dos *layouts*, un *menú*, un archivo de *manifiesto* y un archivo de *cadena*s.

Las *clases* contienen el código fuente que implementa la funcionalidad del programa que a su vez se encuentran dentro de la carpeta *src*, como se puede ver en la figura 2.22. Las clases se implementan en archivos que contienen el código Java.

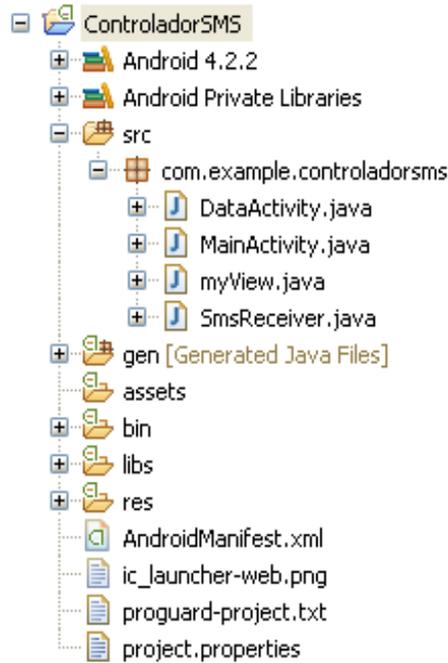


Figura 2.22: Clases del programa en Android.

Los *layouts* o vistas son almacenadas como recursos dentro de la carpeta *layout*, como se muestra en la figura 2.23, la aplicación solo cuenta con dos pantallas diferentes. Los *layouts* se encuentran en archivos con código XML los cuales se pueden introducir en forma de código o como recursos gráficos en el ambiente de Eclipse.

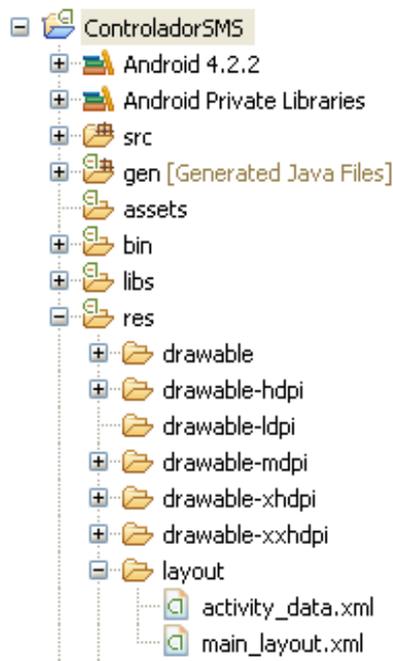


Figura 2.23: Layouts.

El *menú* se encuentra dentro de la *res/menu* cómo se muestra en la figura 2.24. Los menús son un componente de la interfaz común en muchos tipos de aplicaciones. Para proporcionar una experiencia familiar, se recomienda utilizar los menús APIs. El menú de opciones es el conjunto de los elementos del menú para una actividad. Este es el lugar donde se deben ingresar las acciones que tienen un impacto en la aplicación. Si el sistema está desarrollado para Android 2.3 o inferior, los usuarios pueden ingresar al menú pulsando el botón *menú*. Para Android 3.0 o superiores es necesario emigrar a la barra de acciones debido a que algunos dispositivos no cuentan con un botón de menú [AndroidDevelopers, 2009].

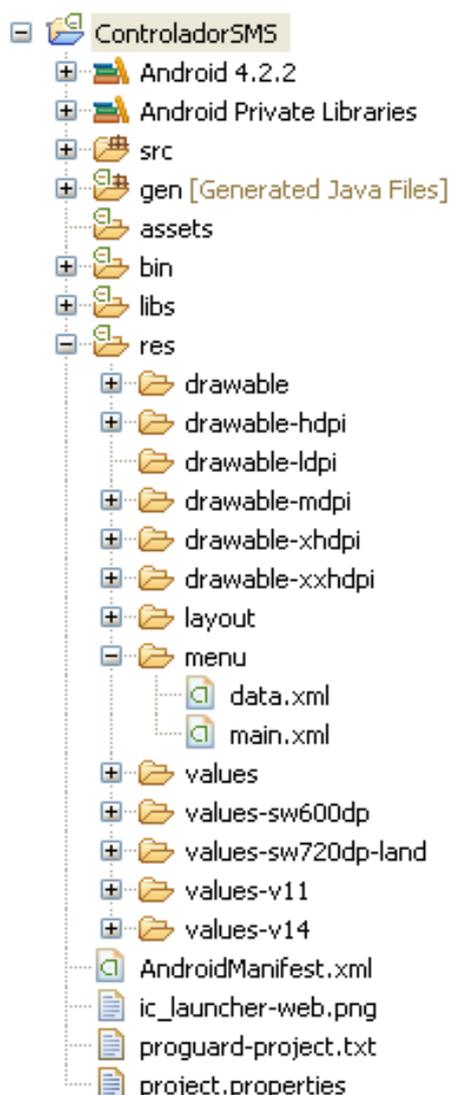


Figura 2.24: Menú.

El archivo de *manifiesto* se encuentra en la raíz del proyecto como se muestra en la figura 2.25. Para el presente proyecto, en el archivo *AndroidManifest* se indican todas las actividades y recursos que utilizará la aplicación.

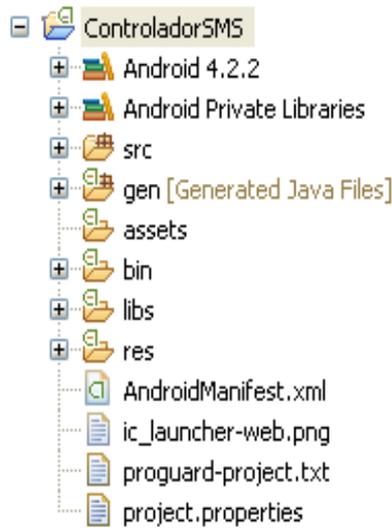


Figura 2.25: Archivo de manifiesto.

Las *cadena*s o *strings* se encuentran en la carpeta *res/values* como se muestra en la figura 2.26. Cada cadena se conforma por un nombre al que se hará referencia dentro del programa y su contenido que se mostrará en pantalla.

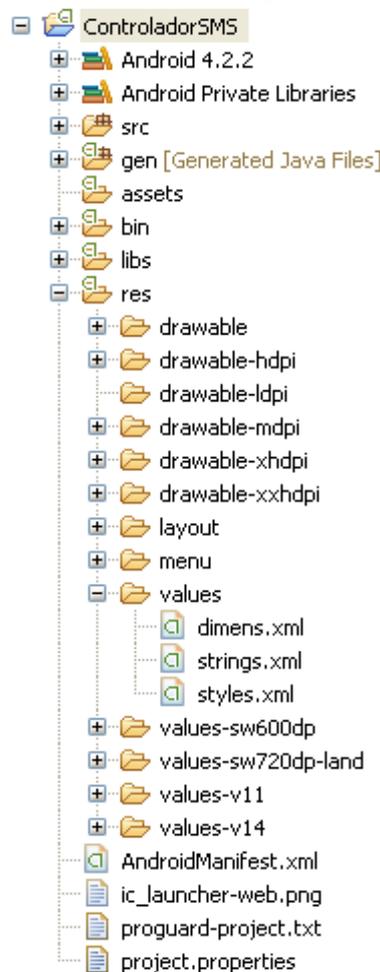


Figura 2.26: Archivo strings o cadena.

En resumen, los recursos utilizados se presentan en la tabla 2.1 y se describen con mayor detalle en las siguientes secciones.

Recurso	Nombre	Descripción
Clase	DataActivity.java	Es la actividad que despliega y administra la pantalla en donde se ingresan los datos para dar de alta y baja usuarios del sistema.
Clase	MianActivity.java	En esta actividad se generan las acciones que responden a los eventos por enviar comandos al sistema electrónico. También se atienden las respuestas del sistema electrónico para actualizar la pantalla principal de la aplicación.
Clase	myView.java	Tiene contenidos todos los métodos usados para dibujar en pantalla.
Clase	SmsReceiver.java	Es el código que permanece pendiente para recibir e interceptar los mensajes SMS con los cuales responde el sistema electrónico.
Layout	activity_data.xml	Contiene el diseño de la vista secundaria que se despliega cuando el usuario quiere validar o eliminar un usuario.
Layout	main_layout.xml	Contiene el diseño de la pantalla principal.
Menú	main.xml	Contiene las opciones que ejecutan las acciones propias de la aplicación
Manifiesto	AndroidManifest.xml	Contiene los parámetros de operación de la aplicación.
Cadena	strings.xml	Contiene las constantes del tipo <i>string</i> que utiliza la aplicación.

Tabla 2.1: Recursos del software de operación para dispositivos Android.

### Clase DataActivity

La clase *DataActivity* se muestra en la figura 2.27 y sus métodos son los siguientes.

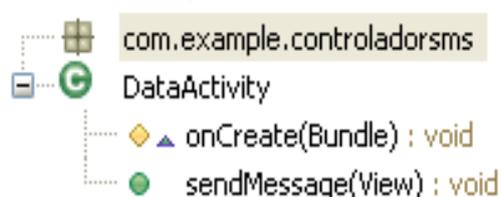


Figura 2.27: Estructura clase DataActivity.

*onCreate()*: Habilita la pantalla secundaria que sirve para dar de alta o baja a un usuario. Este método está ubicado dentro de ésta clase por estrategia.

*sendMessage()*: Permanece a la espera de que arribe un mensaje al dispositivo.

## Clase MainActivity

La clase *MainActivity* se muestra en la figura 2.28 y sus métodos son los siguientes.

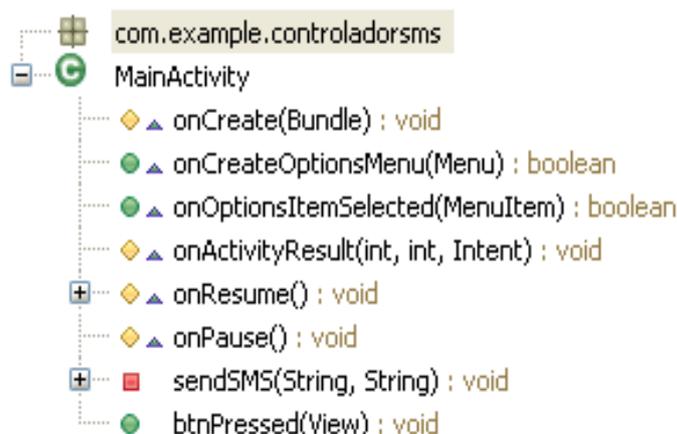


Figura 2.28: Estructura clase MainActivity.

*OnCreate()*: Despliega la pantalla *main\_layout* que está contenida en los recursos, dentro de la carpeta *layout*.

*OnCreateOptionsMenu()*: Añade elementos a la barra de menú, si está activa.

*onOptionsItemSelected()*: Compara la acción seleccionada con las opciones en el menú, para mandar el mensaje correspondiente a cada acción.

*onActivityResult()*: En el caso de que sea seleccionada la acción alta usuario o baja usuario, se tiene que mostrar otra pantalla para que el usuario pueda ingresar el número de teléfono. El método *onActiviytResult* se encarga de controlar esta pantalla y la estructura del mensaje.

*onResume()*: Este método hace visible a la actividad y se encarga de procesar la respuesta del sistema electrónico, para saber la acción correcta que se debe mostrar en pantalla.

*onPause()*: Es llamada cuando la aplicación va a dejar de ser visible por que otra actividad tomará el control.

*sendSMS()*: En estos métodos se muestra el código correspondiente para mandar mensajes desde la aplicación, el método *sendSMS* gestiona el envío del mensaje y el método *onReceive* muestra si este fue o no entregado.

*btnPressend()*: Con este método se controla el zoom del indicador analógico.

## Clase myView

La clase *myView* se muestra en la figura 2.29 y sus métodos son los siguientes [Mind The Robot, 2010].

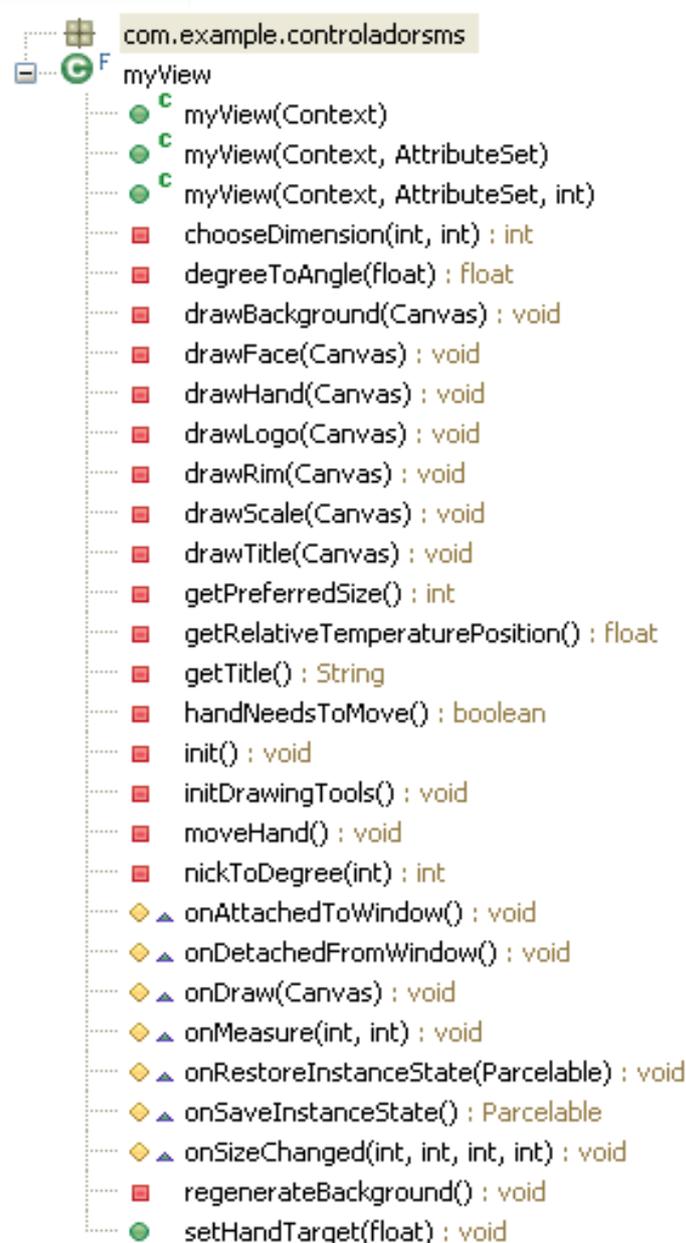


Figura 2.29: Estructura de la clase myView.

Las primeras 3 clases *myView()*, se utilizan por defecto, cuando se modifica una vista previamente echa.

*chooseDimension()*: Define las dimensiones de la pantalla, en las que se mostrará la aplicación.

*degradeToAngle()*: Este método se encarga de degradar el ángulo que existe entre la posición inicial y final de la aguja, hasta que éste sea cero. Se puede decir que este método es el encargado de la precisión.

*drawBackground()*: Método utilizado para reiniciar.

*drawFace()*: Método para darle una textura de fondo.

*drawhand()*: Dibuja la aguja del indicador analógico.

*drawLogo()*: Dibuja el logo de Android.

*drawRim()*: Dibuja un círculo de aspecto metálico.

*drawScale()*: Dibuja la escala y los valores que se mostrarán en el indicador analógico.

*drawTitle()*: Dibuja la cadena de caracteres contenida en *getTitle()*.

*getPreferredsize()*: En el caso de que las dimensiones de la pantalla no sean especificadas, este método proporciona los valores predeterminados.

*getRelativeTemperaturePsition()*: Proporciona la posición actual de la aguja (si es menor o mayor a 600).

*getTitle()*: Almacena una cadena de caracteres.

*hanNeedsToMove()*: Retorna un valor *true* o *false* que indica si la diferencia entre el valor actual del indicador analógico con respecto del valor final es mayor a un valor previamente establecido.

*int()*: Habilita a *initDrawingTools*.

*initDrawingTools()*: Con este método se proporcionan los parámetros de todos los métodos encargados de dibujar las vistas de la aplicación. Como son color, estilo, grosor, escala etc.

*moveHand()*: Controla la velocidad con la que se realizarán los eventos que se encargan de pintar el logo de Android y la aguja del medidor analógico.

*nickToDegree()*: Proporciona los números que se dibujaran en la escala del indicador analógico.

*onAttachedToWindow()*: Se llama cuando la vista está unida a su ventana.

*onDetavhedFromWindow()*: Se llama cuando la vista se separa de su ventana.

*onDraw()*: Controla a los métodos *moveHand()*, *drawLogo()*, *drawBackground()* y *drawhand()*.

*onMeasure()*: Define las dimensiones de la pantalla, en las que se mostrará la aplicación.

*onRestoreInstanceState()*: Restaura el estado de una actividad.

*onSaveInstanceState()*: Graba el estado de una actividad.

*onSizeChange()*: Proporciona el alto y ancho de la pantalla.

*regenerateBackground()*: Como se tienen elementos en el medidor analógico que no se van a mover, como es el borde, la cara, la escala y el título, ésta función ayuda a juntar estos elementos en un mapa de bits, para que no se vean afectados con el cambio de color del logo o el movimiento de la aguja.

*setHandTarget()*: Modifica los valores utilizados para posicionar la aguja y definir el color del logo de Android que se encuentra en el medidor analógico, con los valores que recibe del sistema electrónico.

### Clase SmsReceiver

La clase *SmsReceiver* se muestra en la figura 2.30 y su único método se utiliza para recibir un mensaje SMS y enviarlo a la actividad principal para su procesamiento, validación y despliegue en la vista.



Figura 2.30: Estructura de la clase SmsReceiver.

### Layout activity\_data

La pantalla *activyti\_data*, es un “linear layout” el cual permite ordenar componentes uno tras otro, ya sea en orientación vertical u horizontal. Dentro de este *layout* se coloca primero un cuadro de texto para ingresar el número telefónico y seguido de éste, en orientación horizontal, se coloca el botón que ejecutara la operación y regresará a la ventana principal, como se muestra en la figura 2.31.



Figura 2.31: Layout activity\_data.

### Layout main\_layout

La pantalla *main\_layout* se muestra en la figura 2.32.

Para crear la vista principal se utiliza la función *RelativeLayout*, la cual permite ubicar elementos, que dependerán de la ubicación de otros elementos. Dentro de este *layout*, se crea una ventana transparente, de apenas 10 pixeles de grosor llamada “anchor”, que estará centrada verticalmente en la pantalla. Esta ventana servirá de ayuda para ubicar los diferentes elementos, ya sea en la parte izquierda o derecha de la pantalla.

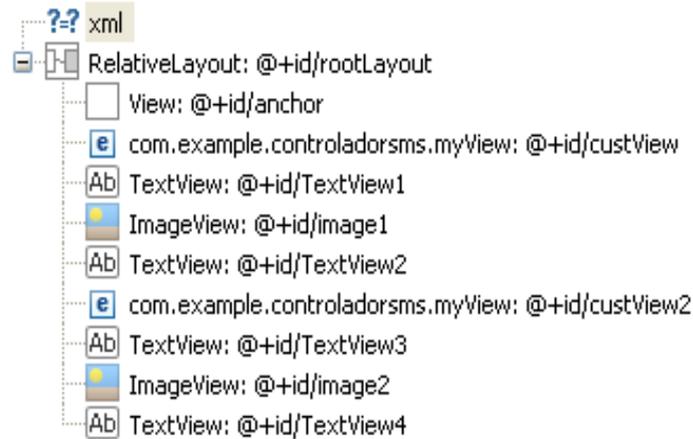


Figura 2.32: Layout main\_layout.

En el siguiente bloque se coloca un *custom view* o vista personalizada del lado izquierdo de anchor (éste *custom view* es el indicador analógico que dibuja la clase *myView*) y en la parte superior del *layout*, posteriormente, se ubican uno de bajo del otro los siguientes elementos: el texto analógico 1, la imagen que simula un led para las respuestas digitales y el texto digital 1. Para terminar, se ingresa lo mismo de lado derecho de la ventana *anchor*.

### Menú main

El menú de opciones se muestra en la figura 2.33.

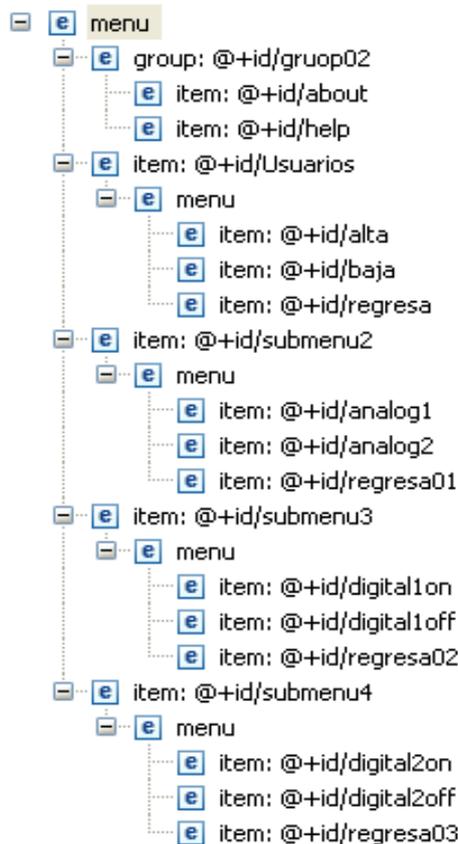


Figura 2.33: Menú main.

Para crear este menú se utilizaron los elementos *menu*, *group* e *item* de la siguiente forma.

<menú> Este debe de ser el nodo raíz que contiene a los atributos *item* y *group*.

<group> Define un grupo

<item> Este es el encargado de generar un elemento o botón que se mostrara en el menú. Puede contener al elemento *menu* o *group*, en el caso de un submenú, <ítem> siempre debe de ser hijo de los atributos *menu* o *group*.

Como se aprecia en la figura 2.33 el menú consta de 6 botones principales, donde cuatro de ellos proporcionan acceso a más opciones.

### Manifiesto AndroidManifest

La estructura del archivo de manifiesto *AndroidManifest* se muestra en la figura 2.34.

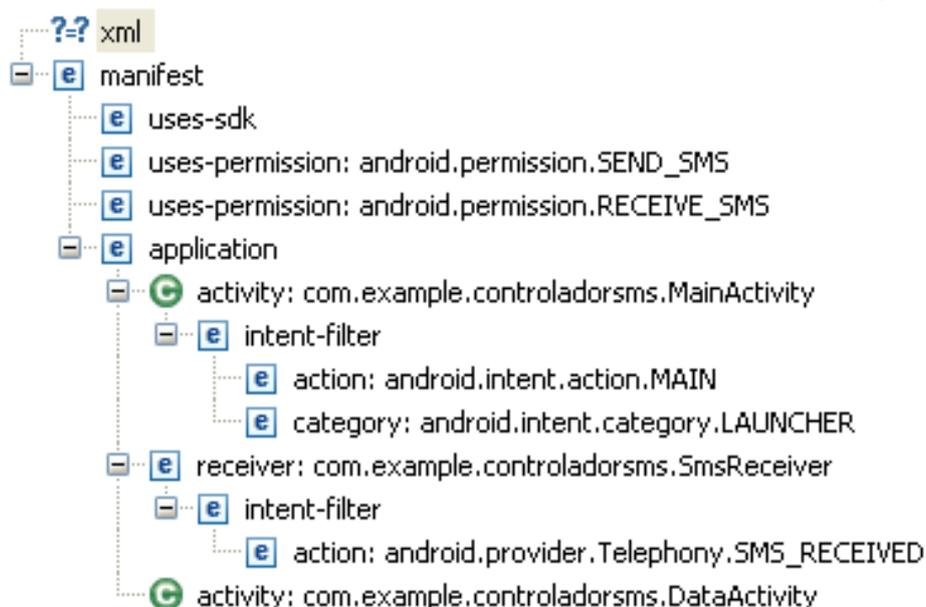


Figura 2.34: Archivo de manifiesto AndroidManifest.

Los bloques que muestra el diagrama, se explican a continuación:

- *uses-sdk*: Se especifica la mínima versión y la máxima versión en la que la aplicación correrá.
- *uses-permission:android.permission.SEND\_SMS* y *uses-permission:android.permission.RECEIVE\_SMS*: Se colocan los permisos para que Android permita tanto recibir mensajes como mandarlos.
- *application*: Administra los permisos que tendrá cada clase para mandar o recibir mensajes o comunicarse entre ellos.

- *intent-filter*: Son elementos usados cuando se requiere comunicar dos clases diferentes.

### Cadena strings

Por último, la figura 2.35 muestra las cadenas utilizadas.

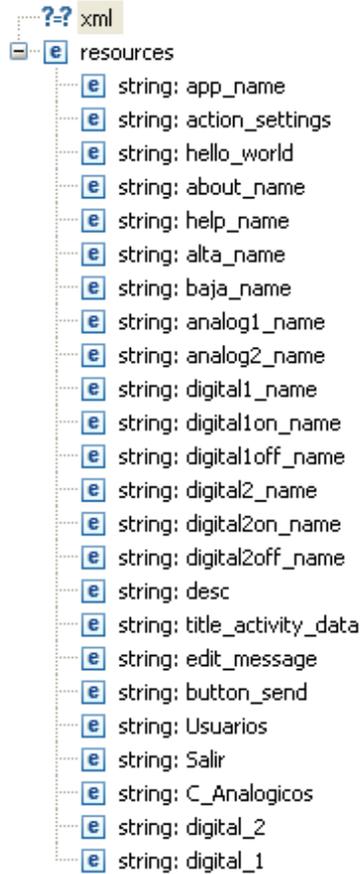


Figura 2.35: Cadena strings.

Cada cadena se conforma por un nombre, al que se hará referencia dentro del programa, que es el mostrado en la figura 2.35 y el nombre que se mostrara en pantalla.



### 3. Resultados

Los resultados obtenidos son puramente cualitativos, estos tienen como objetivo que el sistema electrónico y la aplicación para el dispositivo móvil, logren comunicarse entre ellos para actualizar los datos en ambos dispositivos.

El sistema en general depende de la saturación en la red; como Ciudad Universitaria es un lugar de muy alta densidad de usuarios, el tiempo de respuesta se ve afectado. Por este motivo la depuración del programa se limitó a los fines de semana y a vacaciones. No obstante, una vez terminado el sistema se realizaron diferentes pruebas en las cuales se buscó variar las condiciones, tanto en distancia entre los sistemas, como en la densidad de usuarios en la celda a la que estaban conectados.

Las pruebas realizadas tienen dos objetivos principales: determinar el tiempo de respuesta del sistema en general y determinar el tiempo en que se ejecuta una instrucción en el sistema electrónico.

Para determinar el tiempo de respuesta del sistema en general se tiene como referencia la hora en la que se mandaron y recibieron los mensajes, los cuales se encuentran almacenados en el historial del teléfono móvil. Con esto se obtuvo el tiempo que tarda, para un usuario del sistema, en ejecutarse un comando.

Por otra parte, para medir el tiempo que tarda en realizar una acción el sistema electrónico, se realizó lo siguiente. Se modificó el programa para permitir que los mensajes se almacenaran en la memoria de la SIM del sistema electrónico. Aparte de esto se deshabilitó el pin de seguridad USB\_RTS, para poder observar la comunicación entre el microcontrolador PIC y el chip SIM300 por medio de una computadora conectada vía USB, ver figura 3.1.

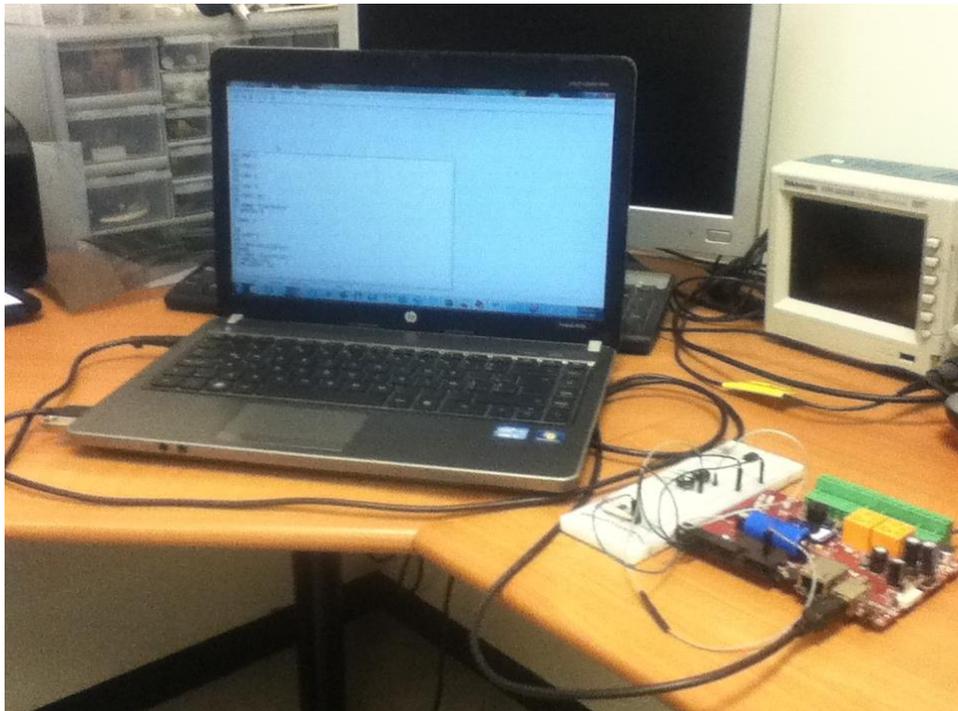


Figura 3.1: Esquema de la conexión entre el sistema electrónico y una computadora vía USB.

La figura 3.2 es un ejemplo de la comunicación entre el chip SIM300 y el microcontrolador PIC, visto desde la ventana del programa hyperterminal de Windows que sirve para monitorear la actividad en el puerto RS232 que enlaza el SIM300 con el microcontrolador PIC.

```

carlos - HyperTerminal
Archivo Edición Ver Llamar Transferir Ayuda
AT+CMGF=1
OK
+CMTI: "SM",1
AT+CMGW="5534083640"digital1
> HoraArribo
+CMGW: 2

OK
AT+CMGR=1
+CMGR: "REC UNREAD", "5534498951", , "13/11/28,21:58:27-24"
analog2;0

OK
AT+CMGD=1
OK
AT+CMGW="5534083640"digital1
> HorraEnvio
+CMGW: 1
AT+CMGS="5534498951"

> analog2;666
+CMGS: 231

OK
AT+CMGF=1
OK
AT+CMGL="ALL"
+CMGL: 1, "STO UNSENT", "5534083640", , "13/11/28,21:58:32-24"
HorraEnvio
+CMGL: 2, "STO UNSENT", "5534083640", , "13/11/28,21:58:47-24"
HoraArribo
OK

```

Figura 3.2: Comunicación entre el chip SIM300 y el microcontrolador PIC en el sistema de control.

Con respecto a la figura 3.2:

- La primer línea AT+CMGF=1 se asegura que la información mostrada en pantalla estará en formato texto.
- El comando +CMTI es un mensaje del SIM300 hacia el microcontrolador indicando que se ha recibido un mensaje SMS y la localidad donde se ha guardado. En este preciso momento, el PIC genera un nuevo mensaje que queda almacenado en la SIM para determinar la hora en que arribó el SMS, mediante el comando AT+CMGW. A esta hora la denominaremos *HoraArribo*.

- Posteriormente el microcontrolador lee el mensaje SMS almacenado en la localidad en la que arribó el SMS al enviar el comando AT+CMGR=1 al SIM300.
- El microcontrolador recibe la respuesta +CMGR del SIM300. Con esta información se ejecuta el comando en el sistema electrónico.
- Se elimina el SMS que arribó al sistema electrónico con el comando AT+CMGD=1.
- Se genera el contenido del mensaje y es almacenado para su posterior envío con el comando AT+CMGW. La hora contenida en este mensaje se denominara *HoraEnvio*.
- La respuesta es enviada con el comando AT+CMGS.
- Se muestran los mensajes de entrada y salida almacenados en memoria con el comando AT+CMGL=ALL para determinar la *HoraArribo* y *HoraEnvio*, con los cuales se calculará el tiempo que tardó en realizar una acción el sistema electrónico.

### 3.1. Prueba 1

La primera prueba se realizó bajo los siguientes parámetros:

- Sistema electrónico y teléfono celular en un mismo lugar o en la misma celda de la red GSM.
- Saturación de la red de telefonía móvil baja.

Para cumplir con los parámetros, se ubicó a las dos componentes del sistema en Ciudad Universitaria, el día sábado 12 de octubre de 2013. Se mandaron 3 mensajes con diferentes comandos, los resultados de esta prueba se dividen en dos partes como se describe a continuación.

En primer lugar, en la determinación del tiempo de respuesta del sistema en general, la tabla 3.1 muestra los datos obtenidos. En la primera columna se muestran los comandos enviados, en la segunda columna se indica la hora en el que se mandó el mensaje desde el teléfono celular, en la tercer columna se indica la hora en que el teléfono celular recibe la respuesta del sistema electrónico y en la cuarta columna se muestra el tiempo total de ejecución del sistema.

Comando enviado	Dispositivo móvil		Tiempo total
	Mensaje enviado	Respuesta recibida	
Digital 1 ON	13:08:54	13:09:30	36 s
ADC 2	13:15:25	13:15:58	33 s
Eliminar usuario	15:10:41	15:11:15	34 s

Tabla 3.1: tiempo de ejecución del sistema general, prueba 1.

En segundo lugar, en la determinación del tiempo que tarda en realizar una acción el sistema electrónico, la tabla 3.2 muestra los datos obtenidos. La primera columna indica el comando que se envió, la segunda columna indica el la hora en la que llegó el mensaje, la tercer columna la hora en la que se envió la respuesta y por último el tiempo total.

<b>Comando enviado</b>	<b>Sistema electrónico</b>		<b>Tiempo total</b>
	<b>HoraArribo</b>	<b>HoraEnvio</b>	
Salida digital 1 ON	13:08:59	13:09:19	20 s
Entrada analógica 2	13:15:28	13:15:47	18 s
Eliminar usuario	15:10:45	15:11:07	22 s

Tabla 3.2: Tiempo de ejecución del sistema electrónico, prueba 1.

### 3.2. Prueba 2

La segunda prueba se realizó bajo los siguientes parámetros:

- Sistema electrónico en zona de bajo tráfico.
- Software controlador en diferente zona o celda de la red GSM, con bajo tráfico.

Para cumplir con los parámetros, se ubicó al sistema electrónico en Ciudad Universitaria y el software controlador en ciudad Nezahualcóyotl, el día martes 15 de octubre de 2013. Los mensajes se mandaron durante la madrugada, para asegurar baja densidad en las dos celdas, los resultados de esta prueba se dividen en dos partes como se describe a continuación.

En primer lugar, en la determinación del tiempo de respuesta del sistema en general, la tabla 3.3 muestra los datos obtenidos. En la primera columna se muestran los comandos enviados, en la segunda columna se indica la hora en el que se mandó el mensaje desde el teléfono celular, en la tercer columna se indica la hora en que el teléfono celular recibe la respuesta del sistema electrónico y en la cuarta columna se muestra el tiempo total de ejecución del sistema.

<b>Comando enviado</b>	<b>Dispositivo móvil</b>		<b>Tiempo total</b>
	<b>Mensaje enviado</b>	<b>Respuesta recibida</b>	
Digital 1 ON	00:21:15	00:22:01	46 s
ADC 2	00:23:45	00:24:29	44 s

Tabla 3.3: tiempo de ejecución del sistema general, prueba 2.

En segundo lugar, en la determinación del tiempo que tarda en realizar una acción el sistema electrónico, la tabla 3.4 muestra los datos obtenidos. La primera columna indica el comando que se envió, la segunda columna indica el la hora en la que llegó el mensaje, la tercer columna la hora en la que se envió la respuesta y por último el tiempo total.

<b>Comando enviado</b>	<b>Sistema electrónico</b>		<b>Tiempo total</b>
	<b>HoraArribo</b>	<b>HoraEnvio</b>	
Salida digital 1 ON	00:21:25	00:21:45	20 segundos
Entrada analógica 2	00:23:57	00:24:15	18 segundos

Tabla 3.4: Tiempo de ejecución del sistema electrónico, prueba 2.

### 3.3. Prueba 3

La tercera prueba se realizó bajo los siguientes parámetros:

- Sistema electrónico y teléfono celular en un mismo lugar o en la misma celda de la red GSM.
- Saturación de la red de telefonía móvil alta.

Para cumplir con los parámetros, se ubicó a las dos componentes del sistema en Ciudad Universitaria, el día martes 19 de noviembre de 2013. Se mandó un mensaje durante el horario de clases, los resultados de esta prueba se dividen en dos partes como se describe a continuación.

En primer lugar, en la determinación del tiempo de respuesta del sistema en general, la tabla 3.5 muestra los datos obtenidos. En la primera columna se muestran los comandos enviados, en la segunda columna se indica la hora en el que se mandó el mensaje desde el teléfono celular, en la tercer columna se indica la hora en que el teléfono celular recibe la respuesta del sistema electrónico y en la cuarta columna se muestra el tiempo total de ejecución del sistema.

<b>Comando enviado</b>	<b>Dispositivo móvil</b>		<b>Tiempo total</b>
	<b>Mensaje enviado</b>	<b>Respuesta recibida</b>	
Digital 1 ON	15:30:03	22:01:25	6:31:23 horas

Tabla 3.5: tiempo de ejecución del sistema general, prueba 3.

En segundo lugar, en la determinación del tiempo que tarda en realizar una acción el sistema electrónico, la tabla 3.6 muestra los datos obtenidos. La primera columna indica el comando que se envió, la segunda columna indica el la hora en la que llegó el mensaje, la tercer columna la hora en la que se envió la respuesta y por último el tiempo total.

<b>Comando enviado</b>	<b>Sistema electrónico</b>		<b>Tiempo total</b>
	<b>HoraArribo</b>	<b>HoraEnvio</b>	
Salida digital 1 ON	22:00:30	22:00:50	20 s

Tabla 3.6: Tiempo de ejecución del sistema electrónico, prueba 3.

### 3.4. Prueba 4

La cuarta prueba se realizó bajo los siguientes parámetros:

- Sistema electrónico en zona de bajo tráfico.

- Software controlador en diferente zona o celda de la red GSM, con alto tráfico.

Para cumplir con los parámetros, se ubicó al software controlador en Ciudad Universitaria durante el horario de clases, y al sistema electrónico en ciudad Nezahualcóyotl, el día jueves 21 de noviembre de 2013. Se mandaron 2 mensajes con diferentes comandos, los resultados de esta prueba se dividen en dos partes como se describe a continuación.

En primer lugar, en la determinación del tiempo de respuesta del sistema en general, la tabla 3.7 muestra los datos obtenidos. En la primera columna se muestran los comandos enviados, en la segunda columna se indica la hora en el que se mandó el mensaje desde el teléfono celular, en la tercer columna se indica la hora en que el teléfono celular recibe la respuesta del sistema electrónico y en la cuarta columna se muestra el tiempo total de ejecución del sistema.

<b>Comando enviado</b>	<b>Dispositivo móvil</b>		<b>Tiempo total</b>
	<b>Mensaje enviado</b>	<b>Respuesta recibida</b>	
Digital 2 ON	13:20:03	13:21:01	58 s
ADC 2	13:25:54	13:26:47	53 s

Tabla 3.7: tiempo de ejecución del sistema general, prueba 4.

En segundo lugar, en la determinación del tiempo que tarda en realizar una acción el sistema electrónico, la tabla 3.8 muestra los datos obtenidos. La primera columna indica el comando que se envió, la segunda columna indica el la hora en la que llegó el mensaje, la tercer columna la hora en la que se envió la respuesta y por último el tiempo total.

<b>Comando enviado</b>	<b>Sistema electrónico</b>		<b>Tiempo total</b>
	<b>HoraArribo</b>	<b>HoraEnvio</b>	
Salida digital 1 ON	13:20:18	13:20:48	20 s
Entrada analógica 2	13:26:15	13:26:33	18 s

Tabla 3.8: Tiempo de ejecución del sistema electrónico, prueba 4.

### 3.5. Prueba 5

La quinta prueba se realizó bajo los siguientes parámetros:

- Sistema electrónico en zona de alto tráfico.
- Software controlador en diferente zona o celda de la red GSM, con bajo tráfico.

Para cumplir con los parámetros, se ubicó al sistema electrónico en Ciudad Universitaria y al software controlador en Ciudad Nezahualcóyotl, las pruebas se realizaron durante el horario de clases, el día viernes 22 noviembre de 2013. Los resultados de esta prueba se dividen en dos partes como se describe a continuación.

En primer lugar, en la determinación del tiempo de respuesta del sistema en general, la tabla 3.9 muestra los datos obtenidos. En la primera columna se muestran los comandos enviados, en la segunda columna se indica la hora en el que se mandó el mensaje desde

el teléfono celular, en la tercer columna se indica la hora en que el teléfono celular recibe la respuesta del sistema electrónico y en la cuarta columna se muestra el tiempo total de ejecución del sistema.

	Dispositivo móvil		
Comando enviado	Mensaje enviado	Respuesta recibida	Tiempo total
ADC 1	18:45:00	21:10:32	2:55:32 horas

Tabla 3.9: tiempo de ejecución del sistema general, prueba 5.

En segundo lugar, en la determinación del tiempo que tarda en realizar una acción el sistema electrónico, la tabla 3.10 muestra los datos obtenidos. La primera columna indica el comando que se envió, la segunda columna indica el la hora en la que llegó el mensaje, la tercer columna la hora en la que se envió la respuesta y por último el tiempo total.

	Sistema electrónico		
Comando enviado	HoraArribo	HoraEnvio	Tiempo total
Entrada analógica 1	21:10:01	21:10:19	18 s

Tabla 3.10: Tiempo de ejecución del sistema electrónico, prueba 5.

Las figuras 3.3 y 3.4 muestran los resultados de las pruebas.

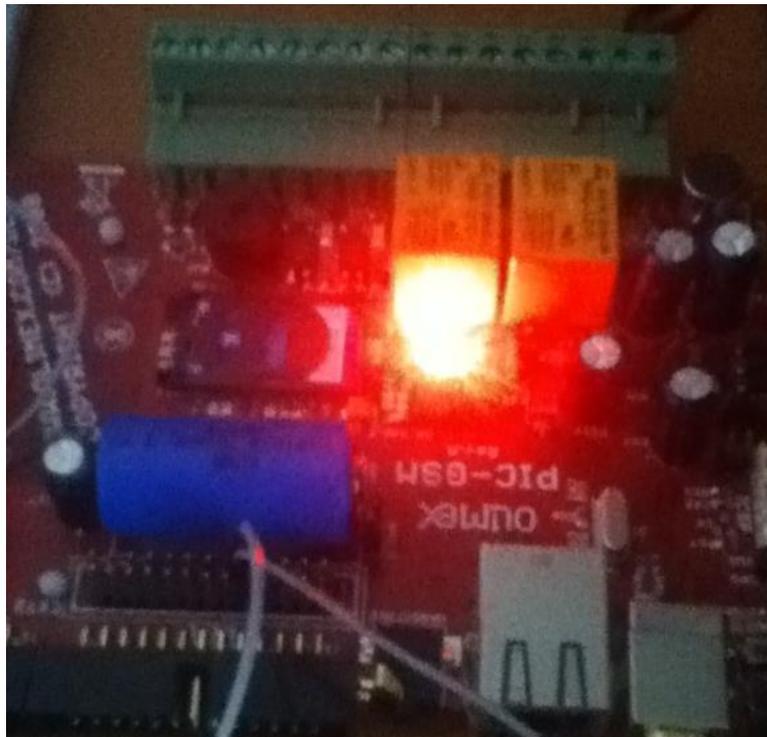


Figura 3.3: módulo PIC-GSM relay 1 activado.



Figura 3.4: Aplicación en Android funcionando.

### 3.6. Discusión

Con los resultados obtenidos en las pruebas se comprobó, que el sistema funciona de una forma fluida mientras el sistema electrónico no se encuentre en una zona de alta densidad de usuarios, como es el caso de ciudad universitaria. También se puede agregar, que no se tienen dificultades para subir un mensaje a la red GSM, en un lugar de alta o baja densidad, sino que el problema es cuando se trata de entregar el mensaje.

No se realizaron pruebas ubicados los sistemas en diferentes lugares con alta densidad de usuarios, debido a que se desconoce la saturación en las demás células de la red y se supone, que los resultados serán semejantes, como cuando se manda un mensaje estando en la misma célula, con alta densidad de usuarios.

El retraso en el servicio de mensajes cortos, no se puede tomar como un problema debido a que este es un sistema de encuesta y no se recomienda orientarlo a protección de otros equipos.

Cuando se diseña una red de telecomunicaciones se hace pensando en los servicios que se van a ofrecer, sean estos mensajes de voz, de datos, internet etc. Se les dedica un ancho de banda dependiendo de los Kbits por segundo que se requerirán para satisfacer la demanda calculada. De la misma manera se le puede dedicar un espacio a este tipo de tecnologías para que no se vean afectados por la saturación de la red.

### 3.7. Trabajo a futuro

Se propone realizar el programa que controle el puerto de Ethernet, para poder realizar las mismas funciones sin necesidad de la red GSM.

Ampliar el algoritmo que controla la base de datos, para permitir la opción de capturar valores periódicos de los medidores analógicos.

El algoritmo del sistema electrónico, cuenta con una opción para realizar llamadas, solo falta realizar los cambios necesarios para configurar el modo alta voz.

Con base en lo realizado se puede desarrollar un sistema autónomo, entre una base de datos y diversos actuadores, que a su vez se puedan comunicar con un usuario a través de una aplicación en Android.



## Conclusiones

Se desarrolló un sistema electrónico y software de operación para el control de un sistema electrónico de forma remota a través de la red de telefonía móvil digital GSM.

El alcance del presente proyecto es la creación de una herramienta básica para el desarrollo de aplicaciones de instrumentación remota con base en mensajes SMS. Esta herramienta básica se puede aplicar al desarrollo de proyectos de instrumentación y control en donde las variables de interés no pongan en riesgo factores críticos. Por ejemplo, en nuestro concepto, el sistema desarrollado sería útil en sistemas de riego en donde se desean activar bombas de riego dependiendo de ciertas condiciones de humedad o PH. Otra aplicación sería la inspección de tráfico en carreteras, en donde el sistema electrónico se encargaría de administrar las estadísticas y un usuario podría acceder a ellas mediante mensajes SMS.

El presente proyecto cumple con los objetivos propuestos incluso se presentan mejoras en ciertos aspectos, por ejemplo la implementación de una pequeña base de datos externa a la tarjeta, que permite almacenar a los usuarios que tendrán acceso al el sistema electrónico.

El tiempo promedio de ejecución de una orden, es de 16 segundos para las acciones digitales y de 25 segundos de las acciones analógicas, estos tiempos de ejecución no presentan un problema debido a que no es un sistema de protección, sino un sistema de prevención, por ejemplo:

- El monitoreo del tráfico vehicular en una carretera.
- El control del riego en un sembradío.
- El monitoreo de diversas variables físicas en sistemas de difícil acceso.



## Referencias

- [1] Eugenio, R., (1999) Telecomunicaciones móviles. Segunda edición, USA, marcombo.
- [2] Parsons, J., (1989) Mobile communications system. Primera edición, USA, Blackie.
- [3] Roberto, M., (2012) Desarrollo de aplicaciones para Android. Primera edición, Madrid, Ra Ma.
- [4] García, E., (2008) Compilador ccs y simulador proteus para microcontroladores PIC, marcombo.
- [5] Sim com, (2006) “AT Commands Set SIM300\_ATC\_V1.06” *owen* [en línea], Disponible en: [http://www.owen.ru/uploads/re\\_pm01\\_list\\_command.pdf](http://www.owen.ru/uploads/re_pm01_list_command.pdf) [Acceso el día 1 de febrero de 2013].
- [6] Sim com, (2005) “Hardware Design SIM300DZ\_HD\_V2.03” *rainbow* [en línea], Disponible en: <http://rainbow.com.ua/upload/files/modemi/SIM300D.pdf> [Acceso el día 25 de marzo de 2013].
- [7] Sánchez, P., (2007) Análisis del desempeño de las redes celulares gsm gprs, *tesis de licenciatura*, México, Universidad Nacional Autónoma de México, septiembre 2005.
- [8] ETSI, (1995) “Digital cellular telecommunications system (Phase 2+) Alphabets and language-specific information (GSM 03.38)” *GSM technical specification*, GSM 3.38 [en línea]. Disponible en: [http://www.etsi.org/deliver/etsi\\_gts/03/0338/05.00.00\\_60/gsmts\\_0338v050000p.pdf](http://www.etsi.org/deliver/etsi_gts/03/0338/05.00.00_60/gsmts_0338v050000p.pdf) [Accesado el día 13 de septiembre de 2013]
- [9] ETSI, (1996) “Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS) Point-to-Point (PP) (GSM 03.40)” *GSM technical specification*, GSM 3.38 [en línea]. Disponible en: [http://www.etsi.org/deliver/etsi\\_gts/03/0340/05.03.00\\_60/gsmts\\_0340v050300p.pdf](http://www.etsi.org/deliver/etsi_gts/03/0340/05.03.00_60/gsmts_0340v050300p.pdf) [Accesado el día 13 de septiembre de 2013]
- [10] Tomas, J., (2011) “Android Programación Aplicaciones”, *Universidad Politécnica de Valencia*. [Página web] Disponible en: <https://polimedia.upv.es/catalogo/busca.asp?id=1296> [Accesado el día 18 de agosto de 2013].
- [11] Mind The Robot, (2010) “Android Custom UI: Hacer un termómetro Vintage”, *Android Custom UI: Making a Vintage Thermometer*. [Página web] Disponible en: <http://mindtherobot.com/blog/272/android-custom-ui-making-a-vintage-thermometer/> [Accesado el día 18 abril de 2013].
- [12] Google, (2009) “Develop”, *AndroidDevelopers* [Pagina web] Disponible en: <http://developer.android.com> [Accesado el día 1 de abril de 2013].
- [13] FTDI Chip, (2010) “FT232R USB UART IC” *ftdi* [en línea], Disponible en: [http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT232R.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf) [Acceso el día 3 de diciembre de 2013]
- [14] ST, (1999) “datasheetcatalog” [en línea], Disponible en: [http://pdf.datasheetcatalog.com/datasheets/134/80491\\_DS.pdf](http://pdf.datasheetcatalog.com/datasheets/134/80491_DS.pdf) [Acceso el día 3 de diciembre de 2013]
- [11] Mind The Robot, (2010) “Android Custom UI: Hacer un termómetro Vintage”, *Android Custom UI: Making a Vintage Thermometer*. [Página web] Disponible en:

<http://mindtherobot.com/blog/272/android-custom-ui-making-a-vintage-thermometer/>  
[Accesado el día 18 abril de 2013].

[15] Google, (2013) "our mobile planet" *thinkwithgoogle* [página web] Disponible en: <http://www.thinkwithgoogle.com/mobileplanet/es/graph/?contry=mx&category> [Accesado el día 23 junio de 2013].

## **Anexo A. Envío y recepción de un mensaje SMS**

Cuando se envía un mensaje el primer sistema que interviene es el VLR donde está registrado el usuario, este 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 el SC de origen. Éste envía el mensaje al SC de destino, una vez allí se convierte en SM-MT. El SC 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. Todo esto es lo que sucede cuando se envía un mensaje al centro de control (SM-MO).

Ahora, cuando el SC ha recibido el mensaje, lo almacena en su base de datos y solicita al HLR del usuario la información de localización; si el usuario destino no está disponible, se almacena el mensaje en el SC durante su periodo de vigencia. Si está disponible, el SC envía al MSC el mensaje indicando en que parte del BSS debe ser entregado. Posteriormente el MSC envía un aviso al VLR al que está conectado el usuario para indicarle que va a entregarse un mensaje; el VLR avisa al terminal del usuario y verifica si está conectado a la red, el VLR responde al MSC con el estado del usuario, si está operativo y con la información de localización el MSC envía el mensaje al usuario e informa al SC que se ha entregado el mensaje y este puede ser borrado de su base de datos.

**Anexo B. Tabla Ascii**

La siguiente tabla muestra el alfabeto de 7-bits, que se especifica en el documento GSM 03.38.

BCD	Hex	Dec	Carácter	
0000000	0x00	0	@	COMMERCIAL AT
0000001	0x01	1	£	POUND SIGN
0000010	0x02	2	\$	DOLLAR SIGN
0000011	0x03	3		YEN SIGN
0000100	0x04	4	è	LATIN SMALL LETTER E WITH GRAVE
0000101	0x05	5	é	LATIN SMALL LETTER E WITH ACUTE
0000110	0x06	6	ù	LATIN SMALL LETTER U WITH GRAVE
0000111	0x07	7	ì	LATIN SMALL LETTER I WITH GRAVE
0001000	0x08	8	ò	LATIN SMALL LETTER O WITH GRAVE
0001001	0x09	9	Ç	LATIN CAPITAL LETTER C WITH CEDILLA
0001010	0x0A	10		LINE FEED
0001011	0x0B	11	Ø	LATIN CAPITAL LETTER O WITH STROKE
			ø	LATIN SMALL LETTER O WITH STROKE
0001100	0x0C	12		
0001101	0x0D	13		CARRIAGE RETURN
0001110	0x0E	14	Ã	LATIN CAPITAL LETTER A WITH RING ABOVE
0001111	0x0F	15	ã	LATIN SMALL LETTER A WITH RING ABOVE
0010000	0x10	16	Δ	GREEK CAPITAL LETTER DELTA
0010001	0x11	17	-	LOW LINE
0010010	0x12	18	φ	GREEK CAPITAL LETTER PHI
0010011	0x13	19	Γ	GREEK CAPITAL LETTER GAMMA
0010100	0x14	20	Λ	GREEK CAPITAL LETTER LAMBDA
0010101	0x15	21	Ω	GREEK CAPITAL LETTER OMEGA
0010110	0x16	22	Π	GREEK CAPITAL LETTER PI
0010111	0x17	23	Ψ	GREEK CAPITAL LETTER PSI
0011000	0x18	24	Σ	GREEK CAPITAL LETTER SIGMA
0011001	0x19	25	Θ	GREEK CAPITAL LETTER THETA
0011010	0x1A	26	Ξ	GREEK CAPITAL LETTER XI
0011011	0x1B	27		Extensión de la tabla de caracteres
	0x1B 09	27 09	ç	LATIN SMALL CAPITAL LETTER C WITH CEDILLA
	0x1B 0A	27 10		
	0x1B 0D	27 11		

	0x1B 14	27 20	^	CIRCUMFLEX ACCENT
	0x1B 1B	27 27		
	0x1B 28	27 40	{	LEFT CURLY BRACKET
	0x1B 29	27 41	}	RIGHT CURLY BRACKET
	0x1B 2F	27 47	\	REVERSE SOLIDUS (BACKSLASH)
	0x1B 3C	27 60	[	LEFT SQUARE BRACKET
	0x1B 3D	27 61	~	TILDE
	0x1B 3E	27 62	]	RIGHT SQUARE BRACKET
	0x1B 40	27 64		VERTICAL BAR
	0x1B 41	27 65	Á	LATIN LETTER A WITH AGUDO
	0x1B 49	27 73	Í	LATIN LETTER I WITH AGUDO
	0x1B 4F	27 79	Ó	LATIN LETTER O WITH AGUDO
	0x1B 55	27 85	Ú	LATIN LETTER U WITH AGUDO
	0x1B 61	27 97	á	LATIN SMALL LETTER A WITH AGUDO
	0x1B 65	27 101	€	EURO SIGN
	0x1B 69	27 109	í	LATIN SMALL LETTER I WITH AGUDO
	0x1B 6F	27 111	ó	LATIN SMALL LETTER O WITH AGUDO
	0x1B 75	27 117	ú	LATIN SMALL LETTER U WITH AGUDO
0011100	0x1C	28	Æ	LATIN CAPITAL LETTER AE
0011101	0x1D	29	æ	LATIN SMALL LETTER AE
0011110	0x1E	30	ß	LATIN SMALL LETTER SHARP S (German)
0011111	0x1F	31	É	LATIN CAPITAL LETTER E WITH ACUTE
0100000	0x20	32	<b>Espacio</b>	SPACE
0100001	0x21	33	!	EXCLAMATION MARK
0100010	0x22	34	"	QUOTATION MARK
0100011	0x23	35	#	NUMBER SIGN
0100100	0x24	36		CURRENCY SIGN
0100101	0x25	37	%	PERCENT SIGN
0100110	0x26	38	&	AMPERSAND
0100111	0x27	39	`	APOSTROPHE
0101000	0x28	40	(	LEFT PARENTHESIS
0101001	0x29	41	)	RIGHT PARENTHESIS
0101010	0x2A	42	*	ASTERISK
0101011	0x2B	43	+	PLUS SIGN
0101100	0x2C	44	,	COMMA
0101101	0x2D	45	-	HYPHEN-MINUS
0101110	0x2E	46	.	FULL STOP
0101111	0x2F	47	/	SOLIDUS (SLASH)
0110000	0x30	48	0	DIGIT ZERO
0110001	0x31	49	1	DIGIT ONE
0110010	0x32	50	2	DIGIT TWO

0110011	0x33	51	<b>3</b>	DIGIT THREE
0110100	0x34	52	<b>4</b>	DIGIT FOUR
0110101	0x35	53	<b>5</b>	DIGIT FIVE
0110110	0x36	54	<b>6</b>	DIGIT SIX
0110111	0x37	55	<b>7</b>	DIGIT SEVEN
0111000	0x38	56	<b>8</b>	DIGIT EIGHT
0111001	0x39	57	<b>9</b>	DIGIT NINE
0111010	0x3A	58	<b>:</b>	COLON
0111011	0x3B	59	<b>;</b>	SEMICOLON
0111100	0x3C	60	<b>;</b>	LESS-THAN SIGN
0111101	0x3D	61	<b>=</b>	EQUALS SIGN
0111110	0x3E	62	<b>;</b>	GREATER-THAN SIGN
0111111	0x3F	63	<b>?</b>	QUESTION MARK
1000000	0x40	64	<b>!</b>	INVERTED EXCLAMATION MARK
1000001	0x41	65	<b>A</b>	LATIN CAPITAL LETTER A
1000010	0x42	66	<b>B</b>	LATIN CAPITAL LETTER B
1000011	0x43	67	<b>C</b>	LATIN CAPITAL LETTER C
1000100	0x44	68	<b>D</b>	LATIN CAPITAL LETTER D
1000101	0x45	69	<b>E</b>	LATIN CAPITAL LETTER E
1000110	0x46	70	<b>F</b>	LATIN CAPITAL LETTER F
1000111	0x47	71	<b>G</b>	LATIN CAPITAL LETTER G
1001000	0x48	72	<b>H</b>	LATIN CAPITAL LETTER H
1001001	0x49	73	<b>I</b>	LATIN CAPITAL LETTER I
1001010	0x4A	74	<b>J</b>	LATIN CAPITAL LETTER J
1001011	0x4B	75	<b>K</b>	LATIN CAPITAL LETTER K
1001100	0x4C	76	<b>L</b>	LATIN CAPITAL LETTER L
1001101	0x4D	77	<b>M</b>	LATIN CAPITAL LETTER M
1001110	0x4E	78	<b>N</b>	LATIN CAPITAL LETTER N
1001111	0x4F	79	<b>O</b>	LATIN CAPITAL LETTER O
1010000	0x50	80	<b>P</b>	LATIN CAPITAL LETTER P
1010001	0x51	81	<b>Q</b>	LATIN CAPITAL LETTER Q
1010010	0x52	82	<b>R</b>	LATIN CAPITAL LETTER R
1010011	0x53	83	<b>S</b>	LATIN CAPITAL LETTER S
1010100	0x54	84	<b>T</b>	LATIN CAPITAL LETTER T
1010101	0x55	85	<b>U</b>	LATIN CAPITAL LETTER U
1010110	0x56	86	<b>V</b>	LATIN CAPITAL LETTER V
1010111	0x57	87	<b>W</b>	LATIN CAPITAL LETTER W
1011000	0x58	88	<b>X</b>	LATIN CAPITAL LETTER X
1011001	0x59	89	<b>Y</b>	LATIN CAPITAL LETTER Y
1011010	0x5A	90	<b>Z</b>	LATIN CAPITAL LETTER Z
1011011	0x5B	91	<b>Ä</b>	LATIN CAPITAL LETTER A WITH DIAERESIS

1011100	0x5C	92	Ö	LATIN CAPITAL LETTER O WITH DIAERESIS
1011101	0x5D	93	Ñ	LATIN CAPITAL LETTER N WITH TILDE
1011110	0x5E	94	Ü	LATIN CAPITAL LETTER U WITH DIAERESIS
1011111	0x5F	95	§	SECTION SIGN
1100000	0x60	96	¿	INVERTED QUESTION MARK
1100001	0x61	97	A	LATIN SMALL LETTER A
1100010	0x62	98	B	LATIN SMALL LETTER B
1100011	0x63	99	C	LATIN SMALL LETTER C
1100100	0x64	100	D	LATIN SMALL LETTER D
1100101	0x65	101	E	LATIN SMALL LETTER E
1100110	0x66	102	F	LATIN SMALL LETTER F
1100111	0x67	103	G	LATIN SMALL LETTER G
1101000	0x68	104	H	LATIN SMALL LETTER H
1101001	0x69	105	I	LATIN SMALL LETTER I
1101010	0x6A	106	J	LATIN SMALL LETTER J
1101011	0x6B	107	K	LATIN SMALL LETTER K
1101100	0x6C	108	L	LATIN SMALL LETTER L
1101101	0x6D	109	M	LATIN SMALL LETTER M
1101110	0x6E	110	N	LATIN SMALL LETTER N
1101111	0x6F	111	O	LATIN SMALL LETTER O
1110000	0x70	112	P	LATIN SMALL LETTER P
1110001	0x71	113	Q	LATIN SMALL LETTER Q
1110010	0x72	114	R	LATIN SMALL LETTER R
1110011	0x73	115	S	LATIN SMALL LETTER S
1110100	0x74	116	T	LATIN SMALL LETTER T
1110101	0x75	117	U	LATIN SMALL LETTER U
1110110	0x76	118	V	LATIN SMALL LETTER V
1110111	0x77	119	W	LATIN SMALL LETTER W
1111000	0x78	120	X	LATIN SMALL LETTER X
1111001	0x79	121	Y	LATIN SMALL LETTER Y
1111010	0x7A	122	Z	LATIN SMALL LETTER Z
1111011	0x7B	123	Ä	LATIN SMALL LETTER A WITH DIAERESIS
1111100	0x7C	124	Ö	LATIN SMALL LETTER O WITH DIAERESIS
1111101	0x7D	125	Ñ	LATIN SMALL LETTER N WITH TILDE
1111110	0x7E	126	ü	LATIN SMALL LETTER U WITH DIAERESIS
1111111	0x7F	127	à	LATIN SMALL LETTER A WITH GRAVE

## Anexo C. Descripción de los pines del SIM300

Power Supply			
PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS
VBAT		Eight BAT pins of the board-to-board connector are dedicated to connect the supply voltage. The power supply of SIM300 has to be a single voltage source of VBAT= 3.4V..4.5V. It must be able to provide sufficient current in a transmit burst which typically rises to 2A. mostly, these 8 pins are voltage input	Vmax= 4.5V Vmin=3.4V Vnorm=4.0V
VRTC	I/O	Current input for RTC when the battery is not supplied for the system. Current output for backup battery when the main battery is present and the backup battery in low voltage state.	Vmax=2.0V Vmin=1.2V Vnorm=1.8V Inorm= 20uA
VDD_EXT	O	Supply 3.0V voltage for external circuit. By measure this pin, user can judge the system is on or off. When the voltage is low, the system is off. Otherwise, the system is on.	Vmax=3.15V Vmin=2.85V Vnorm=3.0V Imax=60mA
GND		Digital ground	
Power on or power off			
PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS
PWRKEY	I	Voltage input for power on key. PWRKEY get a low level Voltage for user to power on or power off the system. The user should keep pressing the key for a moment when power on or power off the system. Because the system need margin time assert the software.	VILmax=0.3*VBAT VIHmin=0.7*VBAT VImax=VBAT

Audio interfaces			
PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS
MIC1P MIC1N	I	Positive and negative voice-band input	Audio DC Characteristics refer to chapter 3.9.4
MIC2P MIC2N	I	Auxiliary positive and negative voice-band input	
SPK1P SPK1N	O	Positive and negative voice-band output	
SPK2P SPK2N	O	Auxiliary positive and negative voice-band output	
Buzzer	O	Buzzer Output	
AGND		Analog ground	
General purpose input/output			
PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS
KBC0~KBC4	O	The GPO can be configured by AT command for outputting high or low level voltage. All of the GPOs are initial low without any setting from AT command.	VILmin=0V
KBR0~KBR4	I		VILmax=0.3 *VDD_EXT
SPI_DATA	I/O		VIHmin=0.7*VDD_EXT
SPI_CLK	O		VIHmax= VDD_EXT+0.3
SPI_CS	O		VOLmin=GND
SPI_D/C	O		VOLmax=0.2V
SPI_RST	O		VOHmin= VDD_EXT-0.2
Network LED	O		VOHmax= VDD_EXT
GPIO8	I/O		Normal Input/Output Port
Serial 1 interface			
PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS
DTR	I	Data Terminal Ready	VILmin=0V
RXD	I	Receive Data	VILmax=0.3*VDD_EXT
TXD	O	Transmit Data	VIHmin=0.7*VDD_EXT
RTS	I	Request to Send	VIHmax= VDD_EXT+0.3
CTS	O	Clear to Send	VOLmin=GND
RI	O	Ring Indicator	VOLmax=0.2V
DCD	O	Data Carrier detection	VOHmin= VDD_EXT-0.2
			VOHmax= VDD_EXT
Serial 2 interface			
DBGTX	O	Serial interface for debugging and communication	
DBGRX	I		

SIM interface			
PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS
SIM_VDD	O	Voltage Supply for SIM card	The voltage can be select by software either 1.8v or 2.8V
SIM_I/O	I/O	SIM Data Output	$V_{ILmin}=0V$
SIM_CLK	O	SIM Clock	$V_{ILmax}=0.3 \cdot SIM\_VDD$
SIM_PRESENCE	I	SIM Card Detection	$V_{IHmin}=0.7 \cdot SIM\_VDD$
SIM_RST	O	SIM Reset	$V_{IHmax}= SIM\_VDD+0.3$ $V_{OLmin}=GND$ $V_{OLmax}=0.2V$ $V_{OHmin}= SIM\_VDD-0.2$ $V_{OHmax}= SIM\_VDD$
AUXADC			
PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS
ADC0	I	General purpose analog to digital converter.	Input voltage value scope 0V to 2.4V