

**UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA DE INGENIERÍA ELÉCTRICA**



**“Diseño y construcción de un sistema de  
monitoreo remoto para equipo médico”**

PRESENTADO POR:

**RENÉ RICARDO JAIME GÓMEZ  
JULIO ALBERTO RAMOS REGALADO  
MARVIN RAFAEL VALENCIA PÉREZ**

PARA OPTAR AL TÍTULO DE:  
**INGENIERO ELECTRICISTA**

CIUDAD UNIVERSITARIA, JUNIO DE 2007

**UNIVERSIDAD DE EL SALVADOR**

RECTORA :

**DRA. MARÍA ISABEL RODRÍGUEZ**

SECRETARIA GENERAL :

**LICDA. ALICIA MARGARITA RIVAS DE RECINOS**

**FACULTAD DE INGENIERÍA Y ARQUITECTURA**

DECANO :

**ING. MARIO ROBERTO NIETO LOVO**

SECRETARIO :

**ING. OSCAR EDUARDO MARROQUÍN HERNÁNDEZ**

**ESCUELA DE INGENIERÍA ELÉCTRICA**

DIRECTOR :

**ING. LUIS ROBERTO CHÉVEZ PAZ**

**UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA DE INGENIERÍA ELÉCTRICA**

Trabajo de Graduación previo a la opción al Grado de:

**INGENIERO ELECTRICISTA**

Título :

**“Diseño y construcción de un sistema de  
monitoreo remoto para equipo médico”**

Presentado por :

**RENÉ RICARDO JAIME GÓMEZ  
JULIO ALBERTO RAMOS REGALADO  
MARVIN RAFAEL VALENCIA PÉREZ**

Trabajo de Graduación aprobado por :

Docente Director :

Ing. Ricardo Ernesto Cortez

San Salvador, Junio de 2007

Trabajo de Graduación Aprobado por:

Docente Director :

**Ing. Ricardo Ernesto Cortez**



A Dios todopoderoso, que nos ha permitido culminar con éxito el presente trabajo de graduación, sea la gloria, el honor y la majestad por los siglos de los siglos.

“Mira que te mando que seas valiente, no temas ni desmayes, por que yo, Jehová tu Dios, estaré contigo en cualquier lugar que tú fueres”.

Josué 1: 7 - 9.

Dios no nos trajo hasta aquí para volver atrás...

René Ricardo Jaime Gómez

Con mucho cariño, principalmente a mis padres que me dieron la vida y han estado conmigo en todo momento. Gracias por todo papá y mamá por darme una carrera para mi futuro y por creer en mí, aunque hemos pasado momentos difíciles siempre han estado apoyándome y brindándome todo su amor, por todo esto les agradezco de todo corazón el que estén conmigo a mi lado.

A mis hermanos, compañeros y amigos, gracias por estar conmigo y apoyarme siempre, los quiero mucho.

Esto fue posible primero que nadie con la ayuda de Dios, gracias por otorgarme la sabiduría y la salud para lograrlo.

Julio Alberto Ramos Regalado

A Dios todo poderoso y misericordioso, este es un paso de mi humilde intento para honrarte, adonde me permitas llegar, Padre, ahí quiero darte la gloria; y a mi querida familia, ahora podré agradecerles todo su amor y apoyo incondicional, moral y material a un nivel económico, lo que considero un privilegio y me llena de alegría.

Marvin Rafael Valencia Pérez



## AGRADECIMIENTOS

Agradezco por su apoyo incondicional y constante a Dios Todopoderoso que me ha brindado siempre las herramientas y medios para salir adelante. A mis padres, René Ricardo Jaime y Ana María Gómez de Jaime. A mi hermano, Saúl Ernesto Jaime, por su compañía y apoyo a lo largo del desarrollo de nuestro estudio. A mi abuelita Ángela Gómez, quien con un enorme sacrificio siempre ha estado en la disposición de dar todo, absolutamente todo por sus nietos y quien ha orado por mí; cuando ni siquiera yo mismo lo he hecho.

Especialmente, con mención honorífica, agradezco a mis compañeros: Julio Alberto Ramos Regalado, por su participación en el presente trabajo de graduación, manteniéndose firme en las horas de flaqueza y con templanza en los momentos críticos; asimismo a mi compañero y amigo, Marvin Rafael Valencia Pérez, con quien la amistad es más fuerte que cualquier mal entendido, discusión o altercado; con quien la amistad tiene sentido y con quien la amistad toma su único y verdadero valor.

Agradezco de igual forma a Karla Guadalupe Ruano, su apoyo incondicional, su cariño y su amor para fortalecerme y seguir adelante, - sos una bendición en mi vida -.

Agradezco a mis amigos: Héctor Daniel Barrera, Francis René Amaya, el “Joven” Eduardo Zamora; Jaime Arévalo (por las desveladas de Kikito, entre otras cosas), Edgar Ramos, Ana María Figueroa. A mis maestros: Ing. Ricardo Ernesto Cortez, Ing. Hugo Colato, Ing. Carlos Arturo Ruano, Ing. Werner Meléndez, Ing. Wilber Calderón, Ing. Eugenio Martínez, Ing. Armando Martínez Calderón, Ing. Luis Chévez entre muchos otros.

Un agradecimiento especial para personas que aparte de ser amigos han sido personas que me han ayudado en momentos críticos y de quienes he aprendido, aprendo y seguiré aprendiendo siempre; Berenice Foligno, quien a pesar de todo y todos mantuvo siempre la firme disposición de anteponer sus ideas y confianza en mí; Don Felipe, que fue el enviado de Dios en mi niñez; Ovidio Pérez Jaime, por sus consejos y su ejemplo de superación; Prof. Martín Samayoa, quien fue la primera persona que me llamó Ingeniero; Don Manuel Avilés (Q.D.D.G.), por ser un ejemplo de bondad; Eduardo Guevara (El Burro), por ser ejemplo de voluntad, laboriosidad y trabajo; Julio César (el Ácido), por su ayuda en el momento justo; A mis jefes, Sergio Pérez, Ramón Monterrosa, Luis Pérez; Mauricio Castillo y Luis Guevara, por su enseñanza y modelaje profesional; A Marni Hernández, Nelson Cabrera, Yaneth López, Rafael Huevo, el Big Brother en conjunto; por ser excelentes compañeros y amigos. A la Sra. Elisa Sosa de Chavarría por su ayuda y comprensión.

A la presidencia de la República y FANTEL a través de su sistema de becas presidenciales a la excelencia académica, gracias a lo cual tuve los medios necesarios para culminar mi estudio.

Finalmente a PDS Health, a través de sus representantes para América Latina, Jorge Luis de León y Joel Westermarck; por brindarnos las herramientas materiales para desarrollar y llevar a cabo el presente trabajo de graduación; asimismo agradezco a todas aquellas personas que de una u otra forma han influido en mi vida para ser lo que ahora soy y ayudarme para ser lo que el futuro ha de demandar de mí.

René Ricardo Jaime Gómez

Principalmente a Dios, por permitirme alcanzar una meta más en mi vida, lo que representa el inicio del resto del camino que me queda por recorrer, esperando siempre su iluminación en las decisiones venideras que me permita tomar siempre el camino correcto.

A mis padres por que gracias a su cariño, guía y apoyo incondicional, he llegado a realizar uno de los más grandes anhelos de la vida, fruto del inmenso apoyo, amor y confianza que en mi se depositó y con los cuales he logrado terminar mis estudios profesionales que constituyen el legado mas grande que pudiera recibir y por lo cual les viviré eternamente agradecido

A mi familia y amigos por su ejemplo de superación incansable, por su comprensión y confianza, por su amor y amistad incondicional, por que sin su apoyo no hubiera sido posible la culminación de mi carrera profesional.

A mis maestros por llevarme de la mano y compartirme generosamente sus conocimientos fruto de lo cual veo hoy realizado mi objetivo de ser un profesional capaz de aportar soluciones y útil a la sociedad.

Julio Alberto Ramos Regalado

A Dios todo poderoso, que sin su intervención amorosa, ayuda y misericordia, ninguno de mis objetivos jamás habría sido alcanzado.

A mis queridos compañeros de tesis: Julio Alberto Ramos Regalado y René Ricardo Jaime Gómez, gracias por todo el arduo trabajo y esmero que depositaron en la tesis, por todas las noches de desvelo que compartimos trabajando juntos, por escuchar mis preocupaciones y brindarme su apoyo. Gracias, Julio, por soportar mi carácter, por las quemadas a causa del cautín y por el don de su paciencia. Gracias René por acompañarme a lo largo de la carrera, por levantarme el ánimo y creer en mí, por no dejarme vencer en los momentos más duros, por enseñarme y aprender conmigo, por la aceptación mutua que hemos conquistado, por tu presencia y por nuestra eterna amistad, mil gracias amigo, que Dios te bendiga y te lleve muy alto.

A mis amados padres: Carlos Roberto Valencia y Cecilia Isabel de Valencia, por forjar buenos principios en mí, por criarme con amor y responsabilidad, por enseñarme a compartir con mis hermanos, por mostrarme que el esfuerzo es el camino a conseguir nuestros anhelos, por apoyarme en mi formación académica y porque han sido para mí, recto y digno ejemplo de vida. Los amo mucho.

A mis tías: Rina del Carmen y Maritza Dolores, gracias por todo su cariño y apoyo, por ayudarme a conseguir el machete con el que abrí camino a través de mi carrera (mi computadora) y por la ayuda que le han brindado a mis padres para mantener a flote el hogar donde me formé. Me siento eternamente agradecido con Uds.

A Karla Guadalupe Ruano Guadrón, gracias por tu ayuda integral durante el desarrollo de la tesis, brindarme sustento, apaciguar mis ánimos y confortarme con tu compañía. Sin vos, el desarrollo de esta tesis no hubiera alcanzado los objetivos planteados en el debido tiempo. Muchas gracias.

A los representantes de PDS Health en Miami: Jorge de León y Joel Westermarck, gracias por el completo respaldo económico durante la etapa de desarrollo del dispositivo electrónico, y por solucionar nuestros problemas económicos de cambio de domicilio en uno de los momentos más críticos de la tesis.

Al coordinador del trabajo de graduación: Ing. Ricardo Ernesto Cortez, gracias por su enorme paciencia, y a los docentes de la carrera de Ingeniería Eléctrica, mis maestros, de quienes he aprendido mucho: Ing. Hugo Miguel Colato, Ing. Marvin Hernández, Ing. Carlos Martínez Calderón, Ing. José Roberto Ramos, Ing. Wilber Calderón, Ing. Werner Meléndez, Ing. Carlos Arturo Ruano, Ing. Mauricio Eguizábal, Ing. Edmundo Santa María y profesores en general.

A mis amados hermanos: Néstor, Jacqueline, Fátima y Josué, muchas gracias por compartir tantos gratos momentos de nuestra infancia y adolescencia, tristezas y alegrías, dificultades y bonanza, peleas, juegos y consejos, por crecer a mi lado y extinguir la soledad. Dios los bendiga y los proteja.

A mis queridos amigos: Daniel Barrera, Saúl Jaime, Francis Amaya, Jaime Arévalo, Arturo Juárez, David y Carlos Romero. Gracias por compartir las angustias inherentes de nuestra lucha por conseguir el tan anhelado cartón, y tomarnos nuestro tiempo para relajarnos después de las duras semanas del ciclo o festejar los cumpleaños, gracias porque con Uds. me siento parte de una nueva familia.

A mis compañeros de carrera: Ana María Figueroa Iglesias, Manuel Fernández, Pedro Boanerges Paz, Marco Antonio Munguía, Julio Cesar (el ácido) y Edgar Antonio Ramos, muchas gracias por el apoyo mutuo y las maratones de estudio. Les deseo muchos éxitos.

A los padres de mi mejor amigo: Sr. René Ricardo Jaime y Sra. Ana María de Jaime, por su preocupación e interés en nuestra formación académica, por su apoyo y consejos. Dios los bendiga.

A la Sra. Francisca Elisa Sosa de Chavarría, Srita. Berenice Foligno, Licda. Margarita de Castillo y a todas las personas que de una u otra forma aportaron su valiosa colaboración para nuestra formación académica y el desarrollo del presente trabajo de graduación, muchísimas gracias.

Marvin Rafael Valencia Pérez.

## PREFACIO

El tratamiento de la diabetes requiere de un riguroso control de los niveles de azúcar presentes en la sangre del paciente; esto implica la toma regular de muestras de sangre; sin mencionar, la enorme responsabilidad y compromiso del paciente sobre su condición de salud y la toma a tiempo de dichas muestras. Este proceso puede tornarse incómodo, dado que se ve la necesidad de asistir frecuentemente a laboratorios y/o clínicas médicas, para efectos de evaluación del resultado que han tenido en el organismo del paciente los medicamentos y dietas recetadas por el médico; dicha evaluación está basada en el historial de los niveles de azúcar en la sangre del paciente, que puede estar registrado en una libreta proporcionada por el médico. El presente trabajo de graduación ofrece un sistema electrónico - informático que reduce la frecuencia de asistencia del paciente a la clínica de su médico, esto beneficia a salvadoreños que viven en zonas rurales o de difícil acceso, donde no hay doctores especialistas cerca, o que pueden ser atendidos en centros de salud locales que cuenten con dicho sistema y sea supervisado remotamente por un especialista; además registra automáticamente el historial de glucemia del paciente, eliminando el riesgo de que éste pueda olvidar anotar su lectura de glucemia, ó deja en evidencia que el tratante no ha efectuado sus lecturas de glucemia en su debido tiempo. El objetivo del presente trabajo de graduación es modernizar y facilitar el control de la salud de personas que han sido diagnosticadas con diabetes.

Los alcances planteados al inicio de la tesis fueron: la construcción de un dispositivo electrónico que extrajera las lecturas de glucemia de la memoria del glucómetro GlucoLab de Infopía (que posee el paciente), y transmitiera la información vía línea telefónica y/o WiFi hasta un ordenador remoto en la clínica del médico; además debía desarrollarse el servicio de atención a las llamadas generadas por el dispositivo electrónico, la base de datos que almacenaría el historial de glucemia de los pacientes y las aplicaciones web de consulta del médico y de los pacientes; todo este software debe ser capaz de ejecutarse en plataformas Windows y Linux. Los alcances desarrollados al final de la tesis fueron la construcción del dispositivo electrónico funcional como lo plantean los alcances al inicio de la tesis, con la diferencia que dicho dispositivo no sólo trabaja con el glucómetro GlucoLab de Infopía sino también con el OneTouch Ultra y OneTouch Ultra 2 de Jhonson y Jhonson; además se investigó sobre los protocolos de comunicación utilizados en la familia de glucómetros FreeStyle de Abbot (FreeStyle, FreeStyle Freedom y FreeStyle Flash); el dispositivo electrónico es capaz de transmitir la información únicamente usando un MODEM a través de una línea telefónica. Se desarrolló el software que se ejecuta en la computadora remota del médico (servicio de atención de llamadas, base de datos y aplicaciones web de consulta) tal como lo plantean los alcances al inicio de la tesis.

## RESUMEN DEL TRABAJO

El presente trabajo de graduación se divide en tres partes principales, la primera contiene aspectos básicos sobre la diabetes, como controlarla, y los dispositivos tecnológicos disponibles para su control, como lo son los glucómetros; además, entra en detalle sobre los protocolos de comunicación que el GlucoLab de Infopía; el OneTouch Ultra y el OneTouch Ultra2 de Jhonson y Jhonson, el FreeStyle, FreeStyle Freedom y FreeStyle Flash de Abbot; utilizan al comunicarse con el ordenador, utilizando cables especiales y software propietario desarrollado por las compañías fabricantes de los correspondientes. Esta investigación se realizó monitoreando la comunicación entre los diferentes glucómetros y el software del fabricante, con algún programa vigilante del puerto serial, como lo es el software Serial Monitor de HDD; además la investigación fue necesaria para seleccionar el microcontrolador apropiado, que es la pieza central del hardware de la parte electrónica del sistema y configurar su módulo de comunicaciones.

La segunda parte del documento expone aspectos de diseño y composición del hardware del dispositivo electrónico, diagramas esquemáticos de los circuitos, generalidades sobre el microcontrolador 18F4550 de la gama alta mejorada de Microchip, y características de éste; que facilitaron en gran medida la elaboración del código fuente que gobierna el funcionamiento del dispositivo electrónico. Se detalla también la lógica de las operaciones que se llevan a cabo en la consola electrónica. Las pistas del dispositivo electrónico fueron hechas empleando un programa de diseño asistido por computadora, como lo es el EAGLE, el código fuente se compiló utilizando el software MPLab de Microchip, el programa se transmitió a la memoria del microcontrolador utilizando el hardware de programación especial ICD2 de Microchip. Las pistas de la tarjeta electrónica y el código fuente del microcontrolador pueden encontrarse en el disco compacto adjunto al presente documento.

La tercera parte del documento detalla las herramientas utilizadas y el proceso de desarrollo del software que se ejecuta en el ordenador remoto del médico; el cual ,comprende el servicio que atiende las llamadas generadas por el dispositivo electrónico, recibe los datos de lecturas de glucemia y los almacena en la base de datos, la base de datos que contiene los diferentes usuarios del sistema (médicos, pacientes y administradores); y las aplicaciones de consulta web para los diferentes usuarios.

Esta tercera parte incluye a su vez toda la bibliografía necesaria para comprender los conceptos generales relacionados a base de datos, presentación Web y el establecimiento de un servidor de aplicaciones para consulta de base de datos utilizando tecnología Servlet.

La base de datos se creó con la ayuda del servidor de base de datos de MySQL AB, más conocido popularmente como MySQL; el cual es un software de administración de base de datos gratuito para proyectos de desarrollo e investigación. La interface web y el servicio de consulta a la base de datos con tecnología Servlet, se desarrollo haciendo uso del software servidor de aplicaciones llamado Apache Tomcat. Todo el software desarrollado en la tercera parte del presente documento se encuentra contenido en el disco compacto adjunto al presente documento.

Los resultados obtenidos al final del presente trabajo de graduación, es un sistema electrónico – informático que proporciona de manera remota el historial de glucemia del paciente al médico, de esta forma el médico puede tener información de sus pacientes en el tiempo oportuno, y llevar un mejor control sobre la salud de sus pacientes, recetar cambios de medicamentos, dietas o actividades físicas, sin tener que concertar una cita médica tan frecuentemente.

## TABLA DE CONTENIDOS

Capítulo	Página
1 GENERALIDADES DE LA DIABETES, LOS GLUCÓMETROS Y EL PROTOCOLO RS232 .....	1
1.1 Glucosa, Glucemia y Diabetes.....	1
1.1.1. Glucosa.....	1
1.1.2. Glucemia .....	2
1.1.3. Diabetes.....	2
1.1.4. Control de la Diabetes .....	3
1.1.5. Monitoreo de Glucosa.....	3
1.2 Generalidades de los Glucómetros.....	5
1.3 El GlucoLab de Infopía, Especificaciones y Tramas.....	10
1.4 El OneTouch Ultra2 de Jhonson & Jhonson, Especificaciones y Tramas .....	14
1.5 El OneTouch Ultra de Jhonson & Jhonson, Especificaciones y Tramas .....	20
1.6 Otros Glucómetros.....	26
1.6.1. Glucómetros de la familia FreeStyle .....	26
1.7 Protocolo RS-232 .....	30
1.8 CONCLUSIONES DEL CAP. I.....	33
1.9 REFERENCIAS BIBLIOGRÁFICAS.....	35
2 HARDWARE Y SOFTWARE DEL DISPOSITIVO ELECTRÓNICO.....	36
2.1 Generalidades sobre microcontroladores PIC y el PIC18F4550.....	36
2.2 Generalidades del dispositivo electrónico.....	39
2.3 Hardware del dispositivo electrónico.....	44
2.4 Generalidades sobre MODEMS y los Comandos AT .....	52
2.5 Software del dispositivo electrónico .....	55
2.6 CONCLUSIONES DEL CAP. II.....	60
2.7 REFERENCIAS BIBLIOGRÁFICAS.....	61
3 DESARROLLO DEL SERVICIO DE ATENCIÓN DE LLAMADAS E INTERFACE WEB PARA ADMINISTRACIÓN Y CONSULTA DE DATOS DEL MÉDICO Y EL USUARIO .....	62
3.1 Lenguaje Java .....	63
3.1.1. El compilador de Java.....	64
3.1.2. La Java Virtual Machine.....	64
3.1.3. Programación Orientada a Objetos.....	64
3.1.4. Variables en Java .....	66

3.1.5. Eventos.....	66
3.1.6. Ejecución multihilo ó Threads.....	67
3.1.6.1. Estados de un Thread.....	68
3.2 Acceso a Puertos.....	68
3.3 JDBC.....	70
3.3.1. Acceso JDBC a Bases de Datos.....	71
3.3.1.1. Modelo de tres capas.....	71
3.4 Servicio Web.....	73
3.4.1. HTML.....	73
3.4.2. Método GET y POST.....	74
3.4.3. Soluciones Web de Servidor.....	75
3.4.3.1. Servlets.....	75
3.5 Servidor Tomcat.....	75
3.6 Aplicaciones para el servicio telefónico y web.....	76
3.6.1. Programa de atención de llamadas.....	76
3.6.1.1. Validación de los datos recibidos.....	78
3.6.2. Programa de aplicación web.....	79
3.6.2.1. Acceso como usuario.....	83
3.6.2.2. Acceso como médico.....	83
3.6.2.3. Acceso como administrador.....	83
3.6.2.4. Programas que componen la aplicación web.....	84
3.7 Base de datos y el servidor MySQL.....	85
3.7.1. Historia de MySQL.....	86
3.7.2. Consultas.....	87
3.7.3. Comandos.....	88
3.7.3.1. Show.....	88
3.7.3.2. Create database.....	88
3.7.3.3. Use.....	89
3.7.3.4. Create table.....	89
3.7.3.5. Describe.....	90
3.7.3.6. Truncate y drop.....	91
3.7.3.7. Load data e insert.....	91
3.7.3.8. Select.....	92
3.7.4. Tablas usadas en la base de datos.....	93
3.7.4.1. Tabla Usuario_tabla.....	95
3.7.4.2. Tabla Medico_tabla.....	96
3.7.4.3. Tabla Adm_tabla.....	96
3.7.4.4. Tabla Tablalecturas.....	97
3.8 CONCLUSIONES DEL CAP. III.....	100
3.9 REFERENCIAS BIBLIOGRÁFICAS.....	102
4 CONCLUSIONES GENERALES Y RECOMENDACIONES.....	103
5 ANEXO A – Hojas de datos del PIC18F4550.....	105
6 ANEXO B – Hojas de datos del MAX232.....	117
7 ANEXO C – Inventario del hardware del dispositivo electrónico.....	119



## LISTA DE TABLAS

Tabla	Página
1. 1 Especificaciones técnicas del GlucoLab .....	11
1. 2 Trama de respuesta al comando 'S' del GlucoLab.....	13
1. 3 Especificaciones técnicas del OneTouch Ultra2 .....	15
1. 4 Comando de solicitud de información para el OneTouch Ultra2 .....	16
1. 5 Trama de respuestas ante el comando de solicitud de tramas del OneTouch Ultra2.....	17
1. 6 Especificaciones técnicas del OneTouch Ultra2 .....	21
1. 7 Comando de solicitud de información para el OneTouch Ultra.....	22
1. 8 Trama de respuestas ante el comando de solicitud de tramas del OneTouch Ultra .....	23
1. 9 Especificaciones técnicas generales de la familia FreeStyle.....	27
1. 10 Comando de solicitud de información para los glucómetros de la familia FreeStyle .....	27
1. 11 Trama de respuestas ante el comando de solicitud de tramas de los glucómetros de la familia FreeStyle .....	28
1. 12 Correspondencia de pines y señales del puerto serial .....	31
2. 1 Trama que envía el dispositivo electrónico por cada lectura de glucemia .....	41
2. 2 Pasos generales en una transacción de información exitosa .....	42
2. 3 Registros S básicos .....	54
2. 4 Códigos resultado básicos.....	54
2. 5 Lista de mensajes que pueden presentarse en el display LCD .....	58
3. 1 Principales clases e interfaces que componen el paquete java.sql.....	72

## LISTA DE FIGURAS

<b>Figura</b>	<b>Página</b>
1. 1 Imagen del GlucoLab.....	10
1. 2 Señales en la espiga estéreo que se conecta al GlucoLab .....	11
1. 3 Imagen del OneTouch Ultra2 .....	15
1. 4 Señales en la espiga estéreo que se conecta al OneTouch Ultra2.....	16
1. 5 Imagen del OneTouch Ultra.....	21
1. 6 Señales en la espiga estéreo que se conecta al OneTouch Ultra .....	22
1. 7 a) FreeStyle. b) FreeStyle Freedom. c) FreeStyle Flash. ....	26
1. 8 Señales en la espiga estéreo que se conecta a los glucómetros de la familia FreeStyle. ....	27
1. 9 Configuración física para el puerto RS-232 DB25 y DB9 .....	31
2. 1 Esquema general del funcionamiento del sistema electrónico - informático.....	40
2. 2 Fotografía del dispositivo electrónico.....	41
2. 3 Esquemático del núcleo del dispositivo electrónico .....	46
2. 4 Señales en la espiga estéreo que se conecta al núcleo del dispositivo electrónico.....	47
2. 5 Conexión resultante con salida baja en el pin RA3 .....	48
2. 6 Conexión resultante con salida alta en el pin RA3 .....	49
2. 7 Fotografía del MODEM MB-R 5V RS-232 de Radicom .....	50
2. 8 Imágenes de la cara frontal y trasera del display LK162-12 de Matrix Orbital .....	51
2. 9 Flujograma de configuración del software del dispositivo electrónico .....	56
2. 10 Flujograma de ejecución general del software del dispositivo electrónico.....	57
3. 1 Diagrama de capas del API javax.comm.....	69
3. 2 Distribución de pines de un Conector DB-9.....	70
3. 3 Configuración del acceso a una BD en tres capas.....	72

3. 4 Diagrama secuencial de la aplicación de recepción de llamadas .....	80
3. 5 Esquema de funciones de la aplicación Web .....	81
3. 6 Página principal de la aplicación de consulta web.....	82
3. 7 Presentación esquemática de la tabla usuario_tabla.....	96
3. 8 Presentación esquemática de la tabla medico_tabla .....	97
3. 9 Presentación esquemática de la tabla adm_tabla .....	97
3. 10 Presentación esquemática de la tabla tablalecturas .....	98

# CAPÍTULO I

## GENERALIDADES DE LA DIABETES, LOS GLUCÓMETROS Y EL PROTOCOLO RS232

### Introducción

En el presente capítulo se incluyen generalidades de la diabetes como enfermedad, con lo que se pretende conocer un poco más de todas las complicaciones que puede generar este padecimiento. Se menciona además una descripción general de lo que es un glucómetro digital aterrizando en los modelos GlucoLab de Infopía y OneTouch Ultra2 y OneTouch Ultra de Jhonson & Jhonson, de los cuales se tratará más adelante, describiendo sus características físicas y electrónicas, hasta presentar sus comandos de petición de información y sus respectivas respuestas a dichos comandos; además se presenta el microcontrolador PIC18F4550 de la gama mejorada de Microchip como el instrumento con el cual se ha de interrogar a los glucómetros.

El Capítulo I tiene como objetivos generales: dar una aproximación a la diabetes como padecimiento y conocer las diversas formas de controlarla, así también como los dispositivos utilizados para dicho propósito y presentar los glucómetros que se utilizaron en la tesis, sus características técnicas y formas de comunicación serial.

### 1.1 Glucosa, Glucemia y Diabetes

#### 1.1.1. Glucosa

La Glucosa es el compuesto orgánico más abundante de la naturaleza; es la fuente principal de energía de las células y el componente principal de los polímeros de almacenamiento energético como el almidón. Representa ni más ni menos que la energía del sol, y es sólo a través de la glucosa que dicha energía puede llegar hasta nuestras células.

La Glucosa es una sustancia que el organismo absorbe o genera a partir del metabolismo de los alimentos que éste consume; el cuerpo humano al ingerir

alimentos como carbohidratos, lípidos, grasas, etc.; los transforma en glucosa, la cual es asimilada gracias a una hormona que produce el páncreas denominada insulina que hace que la glucosa en la sangre pueda entrar en las células. La insulina regula los niveles de azúcar (glucosa) en el organismo, de manera que ésta no alcance niveles extremos.

### **1.1.2. Glucemia**

Glucemia es una medida de la concentración de glucosa en el plasma. En ayunas, los niveles normales de glucosa son entre 70 mg/dL y 120 mg/dL.

Cuando falta glucosa (Hipoglucemia o Hipoglicemia, glucemia menor que 70 mg/dL), proteínas esenciales para el organismo se pierden para convertirse en glucosa, y así evitar daños irreversibles; ya que debido a falta de combustible, no se podrían realizar funciones importantes (sobre todo en el cerebro) y comenzarían a morir neuronas, de la misma forma como si nos faltara oxígeno. Estudios han demostrado que el 70% de la población padece de reiteradas Hipoglucemias en el día y la noche; y debido a éstas se les han diagnosticado el Síndrome de Estrés, Depresión y Adicciones (SEDA).

Cuando la glucosa se eleva por encima de los niveles normales (Hiperglucemia o Hiperglicemia, glucemia mayor a 120 mg/dL en ayunas), el organismo, por medio del páncreas segrega la insulina que permitirá absorber dicha glucosa.

### **1.1.3. Diabetes**

La concentración de glucosa o glucemia permite detectar la diabetes, la cual es una enfermedad caracterizada por una alta concentración de glucosa o Hiperglucemia. Esta enfermedad puede causar con el tiempo problemas serios en la salud. La diabetes es una condición crónica que puede darse generalmente por dos razones: el páncreas no produce o produce poca insulina (Diabetes tipo I) ó las células del cuerpo no responden a la insulina que se produce (Diabetes tipo II). Puede ser originada por diferentes causas, entre ellas la herencia, la obesidad, los malos hábitos alimentarios, la falta de ejercicio físico entre otras.

De acuerdo a estudios realizados por la Federación Internacional de la Diabetes y las Naciones Unidas, la diabetes se está convirtiendo en una pandemia. Más de 200 millones de personas sufren de diabetes, lo cual representa un 3% de la población mundial. La diabetes es responsable de más de un millón de amputaciones al año y es una de las causas principales de ceguera a nivel mundial; es la condición que produce la mayoría de los casos de insuficiencia renal y los altos costos de diálisis a nivel mundial. Millones de personas mueren

cada año innecesariamente. La diabetes causa el mismo número de muertes que el SIDA, consecuencias sociales y económicas por falta de productividad y costos elevados de salud.

No existe una cura para la diabetes; por lo tanto el método de cuidar la salud es controlarla, manteniendo los niveles de glucosa en la sangre lo más cerca posible de los normales. Un buen control de los niveles de azúcar es posible mediante las siguientes medidas básicas: una dieta planificada, actividad física, toma correcta de medicamentos y los chequeos frecuentes del nivel de azúcar. Gracias a los avances tecnológicos de los últimos años, es más fácil mejorar el control del nivel de glucosa en la sangre.

#### **1.1.4. Control de la Diabetes**

Para las personas que padecen de Diabetes tipo I, estas pueden mejorar su control mediante terapias intensivas de insulina, mediante múltiples inyecciones diarias o mediante bombas de insulina.

Para aquellas personas que padecen de Diabetes tipo II, el control se lleva a cabo ingiriendo píldoras de metformina (Glucóphago), que aumentan la sensibilidad del cuerpo a la insulina. El glucóphago generalmente causa disminución del peso, a diferencia de otras píldoras similares que lo aumentan. Otro tipo de píldoras funciona bloqueando la absorción de fécula, con lo cual se reduce la oleada de glucosa que se produce inmediatamente después de comer.

El control de la diabetes requiere de un monitoreo adecuado de los niveles de glucemia del paciente; esto implica el medir dichos niveles con regularidad dependiendo de diversos factores como: tipo de diabetes del paciente, promedio de los resultados anteriores, historial de niveles de glucemia obtenidos, entre otros.

#### **1.1.5. Monitoreo de Glucosa**

Las mediciones de glucemia se pueden realizar durante el día, incluso durante la noche, sin embargo, no todos los pacientes requieren del mismo número de determinaciones, ya que cada paciente es distinto y lleva un tratamiento específico; dependiendo de esto será el número de veces que necesite hacer sus mediciones para tener un buen monitoreo de sus cifras de glucosa en la sangre.

En términos generales, se recomienda hacer por lo menos una determinación al día, en diferentes horarios cada día, de la siguiente manera:

Día 1 En ayunas (al despertar)  
Día 2 2 horas después de desayunar  
Día 3 Antes del almuerzo  
Día 4 2 horas después de comer  
Día 5 Antes de cenar  
Día 6 Antes de dormir  
Día 7 En la madrugada (3 AM)

Continuar así con la misma secuencia.

Sin embargo, cuando el paciente contrae otra enfermedad como gripe, diarrea, fiebre, etc.; debe aumentarse el número de determinaciones para evaluar si debe variarse su tratamiento de diabetes.

En los últimos años ha explotado el progreso en la confección de aparatos con los que los pacientes pueden medir su nivel de glucosa en la sangre, al instante y en su propio hogar. Cada vez más pequeños y más baratos que los modelos antiguos y que además funcionan con una muestra de sangre más pequeña.

Existen en el mercado diferentes marcas que comercializan dispositivos de éste tipo, tal es el caso del glucómetro de Infopía modelo GlucoLab; el cual permite realizar lecturas de glucemia al instante; su tamaño reducido lo hace muy cómodo de portar, puede almacenar hasta 250 lecturas de glucemia y presentarlas en display de cuarzo en unidades de mg/dL o mmol<sup>1</sup>/L, su operación y control por microprocesador le da mucha versatilidad al aparato, permitiéndole contar con características como: establecimiento de la fecha y hora en el reloj interno del microprocesador, distinción del paciente por identificador numérico almacenable junto con datos de glucemia, configuración de control de actividades realizadas por el paciente al momento de realizar la prueba, almacenamiento de fecha y hora de realización de prueba de glucosa, almacenamiento de lectura de temperatura ambiente en el momento de la prueba, comunicación por protocolo serial RS232 a niveles TTL a través del puerto incorporado, entre otras.

Son éstas características las que vuelven a éste glucómetro atractivo y práctico para el paciente; y sobre todo su característica de soporte de protocolo RS232 es la que brinda la oportunidad de realizar nuevas aplicaciones para mejorar el manejo de la información paciente-médico, lo cual es muy importante para el control de las enfermedades relacionadas con la glucosa y defectos de insulina que tanto afectan a la población mundial.

---

<sup>1</sup> Milésima parte de un mol, que es la unidad básica del Sistema Internacional de Unidades, que mide la cantidad de sustancia.

## 1.2 Generalidades de los Glucómetros

Un glucómetro digital es una unidad electrónica portátil, que tiene por objeto medir el nivel de glucosa en una sustancia específica. Estos aparatos constituyen el método de medición moderno de glucemia, sea este de tipo electroquímico<sup>2</sup> o de reflectancia<sup>3</sup>. En general, estos aparatos tienen una capacidad de medición de hasta 600 mg/dL, nivel prácticamente imposible de lograrse dentro del flujo sanguíneo de una persona. No se necesita limpiar ni enjuagar, como sucedía con los medios antiguos. Cada equipo completo contiene un mecanismo de punción semiautomático; no obstante, algunos otros siguen recurriendo a las agujas o lancetas de aplicación manual para obtener las muestras de sangre capilar, tomando 10 micro litros como volumen máximo para la prueba. Lo mejor de estos modernos aparatos se basa en el tiempo de trabajo, ya que arrojan la lectura de la prueba en un tiempo máximo de 45 segundos, algunos inclusive logran tiempos de 5 segundos. Algunos de estos glucómetros digitales tienen memoria de pruebas, y muchos otros accesorios o características muy especiales que sólo poseen instrumentos electrónicos de última generación, como indicador de batería baja, indicador de temperatura de operación en un rango bastante amplio, etc.

La mayoría de estos instrumentos utilizan como transductor de lectura y conversión a parámetros eléctricos tiras reactivas que se colocan en el dispositivo, las cuales toman o absorben la cantidad de sangre necesaria para la prueba; cuando toman esta muestra, la tira reacciona con la sangre, generando una diferencia de potencial, en el nivel de los micro voltios ( $\mu\text{V}$ ). Esta diferencia de potencial le sirve al microcontrolador del glucómetro digital para procesar la lectura, y posteriormente presentarla a través del LCD que posee.

Existen varios tipos de tiras reactivas; cada vez que se cambie el tipo de tiras, es necesario especificarle al glucómetro el código de las tiras reactivas que se va a utilizar, esto se hace a través de un menú en el software del glucómetro. Estas unidades también vienen provistas de un líquido especial de control que se debe aplicar, en lugar de la sangre, en una tira reactiva, la prueba debe arrojar un resultado comprendido dentro de un rango especificado por el fabricante del glucómetro y de la solución. Con el objeto de corroborar que el glucómetro está trabajando correctamente, dicha prueba debe realizarse una vez cada tres o cuatro meses; este líquido se conoce comúnmente como Solución de Control.

Lo interesante de éstos aparatos resalta cuando una persona con diabetes, tratando de preservar su salud, debe mantener un nivel estable de glucosa en su sangre, tratando de contribuir a evitar o retrasar el apareamiento de complicaciones propias de la enfermedad; es necesario para poder llevar a cabo

---

<sup>2</sup> Que produce electricidad por reacciones químicas

<sup>3</sup> Que devuelve la luz en múltiples direcciones



ese cometido tener un método de monitoreo versátil y fácil de utilizar; sin que esto, reste confiabilidad al control. Cuando se detecta en una persona el apareamiento de la enfermedad, la persona y todo un equipo de profesionales dedicados al trato de la enfermedad, necesitan llevar un control y un registro de la enfermedad para generar un plan, con el propósito de hacerle frente a dicho mal, sea con planes de equilibrio alimenticio, deportivo, dosis de medicamentos o aplicación directa de insulina.

La medición de glucemia es el único medio directo y eficiente para medir, controlar y monitorear el desarrollo de la diabetes, es por eso que es necesario llevar un estricto control acerca de tal parámetro en un registro especial; el cual deberá revisar, después de un tiempo determinado, el profesional encargado del paciente para decidir, en base al historial de glucemia, cuales serán las medidas o cambios a implementar en el estilo y costumbres de vida del paciente, en su hábito alimenticio y en su medicamento. Este historial de glucemia refleja las medidas que tuvieron un efecto positivo en la salud del paciente.

Es primordialmente necesario que todo diabético lleve un registro del nivel de glucosa presente en su torrente sanguíneo; pero debe ser obligatorio en toda aquella persona que está bajo un estricto tratamiento de insulina, si toma pastillas para la diabetes, si está en período de embarazo o si registra un nivel demasiado bajo de glucosa en la sangre.

Anteriormente, los registros y las pruebas de glucosa, se hacían con muestras de orina; incluso ahora aun se hacen pruebas con ese método, aunque no es tan confiable como cuando se utiliza una muestra de sangre. Se recomienda una prueba de glucosa con orina, única y exclusivamente cuando es imposible el control de glucosa con sangre, sea por que la enfermedad está fuera de control en el paciente o por que el paciente adolece de otra enfermedad; inclusive en estos casos se vuelve importante dicha prueba, ya que permite conocer también el nivel de cetonas<sup>4</sup> en el organismo; una persona presenta un nivel alto de cetonas cuando el organismo quema solamente grasa y muy pocas cantidades de glucosa; esto es debido a la poca presencia de insulina en el organismo.

Las pruebas con el glucómetro se hacen a través de una muestra de sangre; la cual se obtiene pinchándose con una lanceta en alguna parte del cuerpo, normalmente se hace en los dedos de la mano, en su parte lateral; sin embargo, se puede utilizar el antebrazo, la pierna o la parte carnosa de la mano. Esta muestra se coloca en la tira reactiva.

Una interrogante de importancia se genera cuando el paciente tiene que escoger uno entre muchos tipos de glucómetros digitales, que varían en sus

---

<sup>4</sup> Compuesto orgánico cuyo elemento activo es el doble enlace entre un átomo de carbono con uno de oxígeno

características, precio, formas, tamaños, etc.; hay glucómetros para personas con problemas de visión, otros que son sencillos de utilizar, unos con memoria, etc. En esta decisión influyen: el médico del paciente, si el glucómetro es de fácil uso, que tan sencillo es darle mantenimiento y calibración y en algunos casos, hasta si el glucómetro es aceptado por la compañía de seguro médico del paciente.

En la actualidad existen 2 tipos de glucómetros, los cuales se dividen en base a la técnica empleada para la medición, estos pueden ser del tipo: Reflectómetros o Biosensores. A continuación se define la técnica empleada por cada uno de ellos.

**Reflectómetros:** Miden la luz reflejada desde el reactivo, después de que ha experimentado una reacción química (oxidación enzimática de la glucosa). En la reacción se produce un producto cromático. La intensidad del color es proporcional a la cantidad de glucosa presente.

**Biosensores:** Corresponden a la nueva tecnología, mide la corriente eléctrica (esta corriente se genera por la oxidación de la glucosa) producida por la sangre presente en el reactivo, este paso de corriente genera una diferencia de potencial que sirve para que el microcontrolador de estos dispositivos pueda arrojar una lectura, puesto que esta diferencia de potencial es directamente proporcional al nivel de glucosa presente en la sangre del paciente.

Cuando un paciente desea adquirir un glucómetro debe tomar en cuenta muchos factores tales como: el precio del dispositivo y de sus respectivas tiras reactivas, la técnica de medición, la precisión del glucómetro, apariencia, tamaño de la muestra de sangre, velocidad de análisis, etc. Uno de los factores más incidentes en la decisión del usuario es el precio, pues hay unos que son muy baratos pero las tiras reactivas son muy caras; otros, por el contrario, son muy caros y sus respectivas tiras reactivas son baratas, y algunos otros son equilibrados en estos factores. Algunos glucómetros son desechables, es decir que trabajan en base a baterías que no pueden ser reemplazadas después de su agotamiento. En el tiempo actual, todo se hace de la forma más rápida posible, por lo que los glucómetros han sido diseñados para tener tiempos de análisis muy rápidos, que van desde 45 segundos hasta 5 segundos en el caso de los de última generación.

Los glucómetros de tipo Biosensores son los más difundidos en el mercado actual, por lo que la cantidad de sangre necesaria para desarrollar la prueba toma una importancia relevante; hoy en día las pruebas se realizan con una pequeña muestra de sangre. Es importante que aquellos medidores que necesitan solamente una gota de sangre; se les aplique la cantidad requerida, pues una disminución en la cantidad de sangre puede suministrar un resultado falso. Los últimos medidores funcionan con tiras reactivas que absorben la

sangre por aspirado capilar, mientras que otros requieren aplicar la gota de sangre en la parte superior de la tira reactiva.

Uno de los factores de mayor importancia para el paciente y el médico es la exactitud que poseen los glucómetros digitales, que cada vez salen al mercado en mayor número y variedad. Puesto que la exactitud y precisión constituye uno de los aspectos más importantes al momento de elegir, es necesario que los medidores sean bastante precisos cuando se realiza el análisis de glucemia de acuerdo con las instrucciones y recomendaciones del fabricante. La principal causa de inexactitud la constituye el error humano. Con el tiempo y uso puede que el medidor pierda exactitud, es por ello necesario realizar las pruebas de calibración cada tres o cuatro meses, según sugiera el fabricante, vigilar las tiras reactivas y conservarlas en un recipiente sin o con poca humedad, la exposición a la luz y humedad son factores que deterioran la precisión. Es necesario consultar con el médico cuando los resultados no corresponden a los esperados.

Una forma para probar la exactitud de su medidor consiste en medir la glucosa tanto capilar como en sangre venosa. Para realizarla es necesario que el usuario asista al laboratorio clínico y le tomen una muestra de sangre de la vena para analizar su glucemia, conviene en ese mismo instante realizar una prueba de glucemia capilar, es decir debe obtener una muestra de sangre pinchando el pulpejo del dedo y procesarla en su medidor, hacemos la aclaración “no utilizar sangre venosa para analizarla con el glucómetro” pues los glucómetros en general vienen calibrados para usarse con sangre totalmente fresca de partes capilares (típicamente es la sangre obtenida del dedo). La sangre tomada de la vena pasa por un proceso de centrifugación, es decir queda el plasma de la sangre, dado que el contenido de glucosa es mayor en el plasma que en la sangre pura y fresca es que se explica porque el resultado en sangre venosa suele ser un 15% más alto que en muestras capilares. Actualmente los nuevos medidores están calibrados para medir glucosa en plasma y por esta razón unos tienen mayor precisión.

Lecturas erróneas pueden presentarse cuando el medidor está sucio o viejo o se han almacenado las tiras reactivas en extremos de temperatura y de humedad. También, cuando las tiras reactivas han caducado o cuando existe error humano en el correcto uso del medidor.

Por lo que es necesario que cada paciente le muestre a su médico el método que utiliza para hacerse sus pruebas, para que éste determine si el método es el adecuado.

Los glucómetros son equipos y aparatos sofisticados que han venido a normalizar el ritmo de vida de las personas diabéticas, teniendo un laboratorio para sus pruebas diarias en casa, sin tener que ir hasta el hospital o una clínica para hacerse el chequeo. Sin embargo, aún y cuando los glucómetros son

digitales y desarrollados con las últimas tecnologías, existe un pequeño inconveniente con el registro de lecturas almacenadas dentro del glucómetro; puesto que un gran número de pacientes no sabe que hacer con el registro electrónico después de hacerse sus respectivas pruebas. Hay personas que escriben en una hoja o agenda todas sus lecturas para enviárselas al médico por cualquier medio de comunicación; sea por que no tienen un glucómetro digital con salida electrónico computarizada o por que no están familiarizados con las computadoras ni con el Internet. Hay personas que por miedo a pincharse los dedos, por escasez de tiempo o simplemente por que le restan importancia no se toman las medidas de glucosa en el momento preciso, entregándole al médico un registro ficticio a la hora de su chequeo respectivo.

Para evitar todos estos inconvenientes es necesario crear un dispositivo que trabaje de manera adjunta al glucómetro para poder darle una eficiencia total a una máquina sofisticada como lo es un glucómetro digital. La mayoría de estos aparatos poseen una salida RS232 a niveles TTL para comunicación con software de carácter propietario, los cuales tratan la información y la preparan para enviársela al médico vía correo electrónico; sin embargo, no todas las personas tienen acceso a un computador, o en el peor de los casos, no todas las personas están familiarizadas con los computadores, por esta razón es necesaria la creación de un módulo que pueda dirigir toda esta información vital e importante de cada paciente hacia su médico de forma íntegra, sin importar hora, día, fecha o inclusive condición climática, utilizando medios de comunicación de fácil acceso como lo es una línea telefónica común. Sin mencionar que con un sistema de monitoreo domiciliario, es fácil para el médico determinar si el usuario ha realizado sus pruebas de glucemia a la hora y con la debida responsabilidad, ya que los glucómetros digitales almacenan fecha y hora de la prueba, información que se puede enviar con las mismas lecturas de glucemia dentro de una sola trama de comunicación. Este equipo debe ser económico, de fácil acceso y utilización para personas de todo tipo y edad.

Debido a que no todos los glucómetros son iguales y obviamente no todos utilizan los mismos protocolos de comunicación es necesario realizar un prototipo que desarrolle todas estas actividades para al menos un modelo de glucómetro, para posteriormente pensar en una estandarización para todos los modelos y tipos de glucómetros digitales.

Se persigue con este trabajo que cada paciente tenga acceso a un equipo con el cual, él o ella pueda entregar su historial de glucemia a su médico en cualquier momento utilizando la vía telefónica; asimismo que el médico pueda observar las lecturas y el estado del paciente en cualquier momento, inclusive en tiempo real al momento que el paciente se realiza su prueba de glucemia. Los glucómetros que se utilizaron en el desarrollo de este proyecto son: el modelo GlucoLab de Infopía y el OneTouch Ultra2 de Jhonson & Jhonson.

### 1.3 El GlucoLab de Infopía, Especificaciones y Tramas

El GlucoLab es un glucómetro digital del tipo de biosensores, está entre los más rápidos y eficientes del mercado, puesto que es uno de los más sencillos de utilizar arrojando resultados para cada lectura en sólo 5 segundos. Posee una memoria de 250 resultados de pruebas, pudiendo identificar hasta 100 usuarios diferentes a través de un sistema de reconocimiento de ID propio del aparato. El uso de este aparato está diseñado para uso externo únicamente, además su uso no es para diagnosticar diabetes ni mucho menos para hacer pruebas en recién nacidos; este glucómetro sólo puede trabajar con sangre fresca, es decir no sirve si se le aplica sangre de venas (con plasma).

El GlucoLab tiene muchas características especiales entre las cuales podemos mencionar:

- Indicador de Batería Baja.
- Identificación multiusuario.
- Alarma de recordatorio.
- Identificación de actividades especiales al momento de realizar la prueba, como: antes de comer, después de comer, después de actividades deportivas, después del uso de medicamentos, o prueba realizada utilizando la solución de control.
- Registra la, fecha, hora y temperatura junto con la lectura de glucemia.
- Selección de las unidades del resultado de la prueba mg/dL ó mmol/L.

En la **Figura 1.1** se muestra una imagen del GlucoLab.



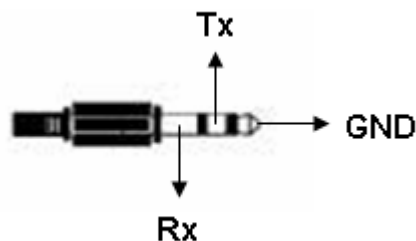
**Figura 1. 1 Imagen del GlucoLab**

La **Tabla 1.1** que se muestra a continuación enlista las especificaciones funcionales del glucómetro GlucoLab de la empresa Infopía.

**Tabla 1. 1 Especificaciones técnicas del GlucoLab**

Tipo de Muestra	Sangre Pura Capilar
Volumen de la Muestra	Mínimo 1 uL
Rango de Análisis	10 - 600 mg/dl
Tiempo de Lectura	5 segundos
Calibración	Plasma Equivalente
Humedad de Operación	10 - 90%
Capacidad de Memoria	250 Resultados
Temperatura de Operación	10 – 40 °C
Tipo de Pantalla	LCD
Medida	74x53x20 mm.
Peso	40g ± 1g = 1.41 oz.
Fuente de Energía	1 Batería de 3V Li (CR2032)
Duración de la Batería	5000 Análisis
Unidad de Glucosa	mmol/l ó mg/dl
ID de paciente	100

El GlucoLab responde a comandos de petición de datos emitidos desde su software propietario, con este software es posible comunicar el glucómetro con la PC y extraer toda la memoria de datos que tiene el aparato. La comunicación es serial usando el protocolo RS232 a 4800 baudios, 8 bits de datos, sin paridad y un bit de paro, como cualquier dispositivo serial común, con la variante que los niveles de voltaje para la comunicación son TTL. El puerto serial del GlucoLab consiste en un conector hembra para una espiga estéreo de 2.5 mm., las señales en la espiga estéreo que se conectará al GlucoLab deben ir ordenadas como se muestra en la **Figura 1.2** a continuación:



**Figura 1. 2 Señales en la espiga estéreo que se conecta al GlucoLab**

Los comandos a los que responde el GlucoLab son determinados caracteres ASCII; la respuesta del glucómetro ante algún comando es una señal de ACK que consiste en el envío del mismo carácter o caracteres ASCII que ha recibido,

seguido de la ejecución del comando que se le envió al glucómetro. Existen varios comandos a los cuales responde el GlucoLab que ejecutan variadas acciones, pero solo uno de ellos se utiliza para la aplicación que se necesita, dicho comando es el de solicitud de tramas y es interpretado por el GlucoLab por medio del caracter 'S' (mayúscula). Entre la Información contenida en la trama de respuesta ASCII del glucómetro ante el comando 'S', después del caracter de ACK, está el ID de usuario al que pertenece dicha lectura, un caracter de actividad especial relacionado a dicha lectura, la fecha y hora de la lectura, la lectura en si y la temperatura ambiental en la cual se tomó dicha lectura. Para poder determinar que el ASCII 'S' era el comando de solicitud de tramas que entiende el GlucoLab, fue necesario el uso de un software espía, como es el HHD Serial Monitor, para poder monitorear toda la información que circulaba a través del puerto serial, durante la comunicación GlucoLab – Software Propietario.

Para poder determinar y comprobar la información que envía el glucómetro ante la solicitud de tramas es bueno conocer los parámetros de configuración y lecturas almacenadas en la memoria del glucómetro antes de la petición de datos. Estos parámetros y lecturas presentes son:

Utilizando un GlucoLab nuevo, sin lecturas almacenadas en su memoria, se configuró de la siguiente manera:

- Funcionalidad multiusuario: Desactivada
- Fecha/Hora: 2006/06/23 05:46 p.m.
- Unidades del resultado: mg/dL
- Unidades de temperatura: °C
- Relaciona la lectura con una Actividad Especial: Desactivado

Posteriormente se llevaron a cabo dos lecturas o análisis de glucemia, que arrojaron los siguientes resultados:

Prueba # 1:

- Fecha: 060623 (AAMMDD)
- Hora: 1746 (hhmm)
- Resultado: 80 (mg/dL)

Prueba # 2:

- Fecha: 060623 (AAMMDD)
- Hora: 1752 (hhmm)
- Resultado: 66 (mg/dL)

En la **Tabla 1.2** se muestra la trama de respuesta del GlucoLab después de enviar el comando 'S':

**Tabla 1. 2 Trama de respuesta al comando 'S' del GlucoLab**

<b>Caracter #</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>
<b>Trama</b>	S	0	0	2	.	.	1	2	1	0	0	6	0	6	2
<b>Caracter #</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
<b>Trama</b>	3	1	7	4	6	0	8	0	2	9	8	?	~	1	2
<b>Caracter #</b>	<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>	<b>36</b>	<b>37</b>	<b>38</b>	<b>39</b>	<b>40</b>	<b>41</b>	<b>42</b>	<b>43</b>	<b>44</b>	<b>45</b>
<b>Trama</b>	1	0	0	6	0	6	2	3	1	7	5	2	0	6	6
<b>Caracter #</b>	<b>46</b>	<b>47</b>	<b>48</b>	<b>49</b>	<b>50</b>										
<b>Trama</b>	2	9	2	A	é										

De la Tabla anterior podemos observar que:

- **Caracter 1:** es el caracter de ACK del comando 'S' que devuelve el GlucoLab.
- **Caracteres del 2 al 4:** representan el número de lecturas almacenadas en la memoria del glucómetro, que para este caso son 2.
- **Caracteres 5 y 6:** son caracteres sin representación gráfica definida y es por ello que se muestran con un punto. Así el **caracter 5** es un Retorno de Carro <CR> ó 0x0D en hexadecimal; y el **caracter 6** es un avance de línea <LF> ó 0x0A en hexadecimal.
- **Caracteres del 7 al 9:** al igual que el trío de **caracteres del 29 al 31** representan el número identificador del usuario o ID para cada lectura, para este caso, ambos tríos presentan el mismo valor 121 que es el identificador especial de las lecturas realizadas cuando la característica de multiusuario está desactivada, de lo contrario los tríos podrían presentar números diferentes según el ID donde hayan sido almacenadas las lecturas; esto lo puede seleccionar el usuario en el GlucoLab después de cada lectura si éste estaba configurado con la característica multiusuario activada. Como se dijo en la **Tabla 1.1** el GlucoLab es capaz de reconocer hasta 100 IDs diferentes.
- **Caracter 10:** al igual que el **caracter 32** son caracteres de actividad especial, y vinculan a su respectiva lectura a una actividad especial de la siguiente manera, si el caracter de actividad es:
  - 0** → No se especifica ninguna actividad especial
  - 1** → Después de tomar medicamento
  - 2** → Después de actividades deportivas
  - 3** → Antes de comer



- 4 → Después de comer
- 5 → Lectura realizada con la solución de control

Para este caso ambas lecturas presentan caracteres de actividad **0** lo cual significa que no se especificó ninguna actividad especial para ambas lecturas.

- **Caracteres del 11 al 16:** al igual que el grupo de **caracteres del 33 al 38**, informan sobre la fecha en que fue tomada su respectiva lectura en el formato AAMMDD. Así, para este caso, los **caracteres del 11 al 16** dicen que dicha lectura fue tomada el 23 de junio del año 2006, mismo día al que hacen referencia los **caracteres del 33 al 38** refiriéndose a la otra lectura.
- **Caracteres del 17 al 20:** al igual que el grupo de **caracteres del 39 al 42** comunican la hora en que fue tomada su lectura respectiva en el formato hhmm. Así, para este caso, los **caracteres del 17 al 20** dicen que su respectiva lectura se tomó a las 5:46 p.m. (prueba # 1) y los **caracteres del 39 al 42** dicen que su respectiva lectura se hizo a las 5:52 p.m. (prueba # 2).
- **Caracteres del 21 al 23:** al igual que el trío de **caracteres del 43 al 45** contienen el valor de la lectura en si que fue tomada. Así, para este caso, los **caracteres del 21 al 23** informan que el valor de la lectura es 80 mg/dL, y los **caracteres del 43 al 45** detallan que el valor de la lectura tomada es 66 mg/dL.
- **Caracteres 24 y 25:** al igual que el par de **caracteres 46 y 47** reportan la temperatura ambiental en el momento en que se tomo la lectura respectiva. Así, en este caso, para ambas lecturas la temperatura ambiental fue de 29 grados centígrados.
- **Caracteres del 26 al 28:** al igual que el trío de **caracteres del 48 al 50** no contienen información útil sobre su respectiva lectura, son aleatorios y nada más sirven para delimitar el fin de una lectura y el inicio de otra dentro de la trama de respuesta del GlucoLab.

#### 1.4 El OneTouch Ultra2 de Jhonson & Jhonson, Especificaciones y Tramas

Al igual que el GlucoLab el OneTouch es un glucómetro del tipo biosensores y su tiempo de análisis es de 5 segundos. La muestra debe ser sangre capilar aunque internamente el glucómetro calibra el resultado como si la muestra hubiese sido sangre de venas, es decir un 12% mayor al resultado obtenido con la sangre capilar; dicho ajuste hace posible que el resultado obtenido con el OneTouch sea directamente comparable con el resultado obtenido a un análisis de laboratorio utilizando sangre de venas. El OneTouch puede almacenar hasta

500 resultados en su memoria y a diferencia del GlucoLab éste glucómetro es de uso personal o mono usuario. Igualmente el glucómetro no debe ser utilizado para diagnosticar diabetes ni debe ser utilizado en recién nacidos.

En la **Figura 1.3** se muestra una imagen del OneTouch Ultra2.



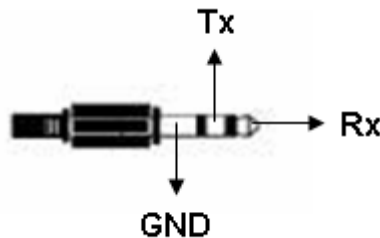
**Figura 1. 3 Imagen del OneTouch Ultra2**

La **Tabla 1.3** muestra las especificaciones técnicas del OneTouch Ultra2 de LifeScan de Jhonson & Jhonson.

**Tabla 1. 3 Especificaciones técnicas del OneTouch Ultra2**

Tipo de Muestra	Sangre Pura Capilar
Volumen de la Muestra	Mínimo 1 uL
Rango de Análisis	20 - 600 mg/dl
Tiempo de Lectura	5 segundos
Calibración	Plasma Equivalente
Humedad de Operación	10 - 90%
Capacidad de Memoria	500 Resultados
Temperatura de Operación	6.1 – 43.9 ° C
Tipo de Pantalla	LCD
Medida	3.12×2.25×0.90 pulg.
Peso (con batería)	1.5 oz.
Fuente de Energía	2 Batería de 3V Li (CR2032)
Apagado automático	2 min. de inactividad
Unidad de Glucosa	mg/dl
Característica multiusuario	No soportada

Igualmente el OneTouch posee un software propietario para descargar los datos almacenados en su memoria al ordenador. Se ejecutaron el software propietario y el programa espía del puerto serial, mientras se realizaba una descarga de datos desde el glucómetro; en dicha actividad se observaron todos los comandos y respuestas entre el software propietario y el glucómetro y se determinó que la comunicación con el glucómetro es RS232 a niveles TTL a 9600 baudios, 8 bits de datos, sin paridad y un bit de paro. El puerto serial del OneTouch Ultra2 consiste en un conector hembra para una espiga estéreo de 3.5 mm., la distribución de señales en la espiga que se conecta al OneTouch Ultra2 se muestra en la **Figura 1.4**.



**Figura 1. 4 Señales en la espiga estéreo que se conecta al OneTouch Ultra2**

Además se determinó que el comando de solicitud de información es el que se muestra en la **tabla 1.4**.

**Tabla 1. 4 Comando de solicitud de información para el OneTouch Ultra2**

Caracter #	1	2	3	4	5
Comando	.	.	D	M	P

Los **caracteres 1 y 2** de la tabla anterior no poseen representación gráfica definida, es por ello que se muestran con un punto. El **caracter 1** es el número hexadecimal 0x11 y el **caracter 2** es un Retorno de Carro <CR> o en hexadecimal el número 0x0D. Los **caracteres 3, 4 y 5** son letras mayúsculas.

Para poder determinar y comprobar la información que envía el glucómetro ante la solicitud de tramas es bueno conocer los parámetros de configuración y lecturas almacenadas en la memoria del glucómetro antes de la petición de datos. Estos parámetros y lecturas presentes son:

Utilizando un OneTouch Ultra2 nuevo, sin lecturas almacenadas en su memoria, se configuró de la siguiente manera:

- Fecha/Hora: 2006/10/13 02:51 p.m.
- Unidades del resultado: mg/dL

Posteriormente se realizó una lectura o análisis de glucemia, que arrojó los siguientes resultados:

- Fecha: 10/13/06 (MM/DD/AA)
- Hora: 14:52:17 (hh:mm:ss)
- Resultado: 117 (mg/dL)

En la **Tabla 1.5** se muestra la trama de respuesta del OneTouch Ultra2 después de enviado el comando de solicitud de tramas:

**Tabla 1. 5 Trama de respuestas ante el comando de solicitud de tramas del OneTouch Ultra2**

<b>Caracter #</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>
<b>Trama</b>	P		0	0	1	,	"	V	D	X	4	3	3	6	A
<b>Caracter #</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
<b>Trama</b>	Y	"	,	"	M	G	/	D	L		"		0	5	B
<b>Caracter #</b>	<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>	<b>36</b>	<b>37</b>	<b>38</b>	<b>39</b>	<b>40</b>	<b>41</b>	<b>42</b>	<b>43</b>	<b>44</b>	<b>45</b>
<b>Trama</b>	1	.	.	P		"	F	R	I	"	,	"	1	0	/
<b>Caracter #</b>	<b>46</b>	<b>47</b>	<b>48</b>	<b>49</b>	<b>50</b>	<b>51</b>	<b>52</b>	<b>53</b>	<b>54</b>	<b>55</b>	<b>56</b>	<b>57</b>	<b>58</b>	<b>59</b>	<b>60</b>
<b>Trama</b>	1	3	/	0	6	"	,	"	1	4	:	5	2	:	1
<b>Caracter #</b>	<b>61</b>	<b>62</b>	<b>63</b>	<b>64</b>	<b>65</b>	<b>66</b>	<b>67</b>	<b>68</b>	<b>69</b>	<b>70</b>	<b>71</b>	<b>72</b>	<b>73</b>	<b>74</b>	<b>75</b>
<b>Trama</b>	7				"	,	"			1	1	7		"	,
<b>Caracter #</b>	<b>76</b>	<b>77</b>	<b>78</b>	<b>79</b>	<b>80</b>	<b>81</b>	<b>82</b>	<b>83</b>	<b>84</b>	<b>85</b>	<b>86</b>	<b>87</b>	<b>88</b>	<b>89</b>	<b>90</b>
<b>Trama</b>	"	N	"	,	"	0	0	"	,		0	0		0	9
<b>Caracter #</b>	<b>91</b>	<b>92</b>	<b>93</b>	<b>94</b>											
<b>Trama</b>	A	8	.	.											

De la Tabla anterior podemos observar que:

- **Caracter 1:** es la letra P mayúscula, y sirve como caracter de ACK, y determina que el comando fue recibido, además es repetido como el primer caracter de cada nueva línea (después del ASCII de Avance de Línea <LF>) en la respuesta del comando de petición de tramas, tal es el caso del **caracter 34**.
- **Caracter 2:** es un caracter de separación, específicamente un Espacio en Blanco <SP> ó 0x20 en hexadecimal que va después del caracter "P" en cada línea de respuesta al comando de petición de tramas, igual al **caracter 35**.
- **Caracteres del 3 al 5:** informan sobre el número de lecturas almacenadas en la memoria del glucómetro. Para este caso, 1 lectura.
- **Caracter 6:** es un caracter de separación, específicamente es una coma (,).

- **Caracteres del 7 al 17:** detallan el número serial del glucómetro dado por el fabricante. Dicho número serial está compuesto por 9 caracteres ASCII (**caracteres del 8 al 16**) y están encerrado por comillas (“, **caracteres 7 y 17**). Para este caso se utilizó el OneTouch Ultra2 con el número serial “VDX4336AY”.
- **Caracteres del 19 al 26:** informan sobre las unidades en las que se expresa la lectura de glucemia. Las unidades son determinadas en los **caracteres del 20 al 24**; para este caso mg/dL, y le sigue un Espacio en Blanco <SP> o 0x20 en hexadecimal (**caracter 25**); todo el grupo de caracteres está encerrado por comillas (“, **caracteres 19 y 26**).
- **Caracteres del 27 al 31:** son 4 caracteres alfanuméricos (**caracteres del 28 al 31**) que no contienen información útil para la finalidad del presente trabajo de graduación, previo a éstos, se encuentra un Espacio en Blanco <SP> ó 0x20 en hexadecimal (**caracter 27**).
- **Caracteres 32 y 33:** no tienen representación gráfica definida, es por ello que se muestran como puntos. Son caracteres que indican el inicio de una nueva línea. Específicamente el **caracter 32** es un Retorno de Carro <CR> ó 0x0D en hexadecimal y el **caracter 33** es un Avance de Línea <LF> ó 0x0A en hexadecimal. Son iguales a los **caracteres 93 y 94** únicamente que ellos especifican el final de la respuesta al comando de petición de tramas. Si el glucómetro tuviera más de una lectura almacenada, los caracteres 93 y 94 representarían el inicio de una nueva línea como lo hacen los caracteres 32 y 33; así la trama de respuesta al comando de petición de tramas es de longitud variable y siempre finaliza con un Retorno de Carro <CR>, seguido de un Avance de Línea <LF>.
- **Caracteres del 36 al 40:** especifican el día en que se realizó la lectura. Específicamente los **caracteres del 37 al 39** determinan el día por sus letras en inglés de la siguiente manera:

<b>MON</b>	→	lunes
<b>TUE</b>	→	martes
<b>WED</b>	→	miércoles
<b>THU</b>	→	jueves
<b>FRI</b>	→	viernes
<b>SAT</b>	→	sábado
<b>SUN</b>	→	domingo

Para el caso, la lectura de prueba fue tomada el día viernes. Los caracteres que determinan el día están encerrados por comillas (“, **caracteres 36 y 40**).

- **Caracter 41:** es un caracter de separación, específicamente es una coma (,), igual que los **caracteres 52, 66, 75, 79 y 84.**
- **Caracteres del 42 al 51:** comunican la fecha en que se realizó la lectura. Específicamente los **caracteres del 43 al 50** determinan la fecha en el formato MM/DD/AA. Para este caso, la lectura se tomó el 13 de octubre de 2006. Los **caracteres 42 y 51** solamente encierran la fecha entre comillas (“”).
- **Caracteres del 53 al 65:** expresan la hora en que se tomo la lectura. Específicamente los **caracteres del 54 al 61** determinan la hora en el formato hh:mm:ss. Para este caso, la lectura se tomó a las 2 de la tarde con 52 minutos y 17 segundos. Los **caracteres del 62 al 64** solamente son Espacios en Blanco <SP> o 0x20 en hexadecimal, y los **caracteres 53 y 65** encierran al grupo entre comillas (“”).
- **Caracteres del 67 al 74:** se explican a continuación:
  - ✓ **Caracter 68:** determina si la prueba se realizó utilizando la solución de control o no. Si se utilizó la solución de control el caracter es una “C” (mayúscula); sino se utilizó la solución de control, el caracter es un espacio en Blanco <SP> ó 0x20 en hexadecimal como es el caso para esta prueba. El usuario puede agregar el comentario a la lectura, de que ésta fue realizada usando solución de control; eso se hace desde el menú del glucómetro, después de tomada la lectura.
  - ✓ **Caracter 69:** es un espacio en Blanco <SP> ó 0x20 en hexadecimal de separación al igual que el **caracter 73.**
  - ✓ **Caracteres del 70 al 72:** son la lectura de glucemia. Para este caso la lectura fue de 117 mg/dL.
  - ✓ **Caracteres 67 y 74** encierran a todo el grupo entre comillas (“”).
- **Caracteres del 76 al 78:** determinan si la lectura se realizó antes o después de comer. Específicamente el **caracter 77**, según su valor puede determinar lo siguiente, si el caracter 77 es:
  - N →** No especifica si la lectura se tomó antes o después de comer (del inglés None)
  - B →** Indica que la lectura se tomó antes (del inglés Before) de comer
  - A →** Indica que la lectura se tomó después (del inglés After) de comer

El usuario puede especificar o no si la lectura se realizó antes o después de comer, desde el menú del glucómetro, después de tomada la lectura. Los **caracteres 76 y 78** encierran al caracter 77 entre comillas (“”).

- **Caracteres del 80 al 83:** relacionan a la lectura con una actividad especial. Específicamente los **caracteres 81 y 82** según su valor pueden determinar lo siguiente:

<b>00</b>	→	Sin actividad o la lectura se hizo con la solución de control
<b>01</b>	→	Comió poco
<b>02</b>	→	Comió demasiado
<b>03</b>	→	Después de ejercicio moderado
<b>04</b>	→	Después de ejercicio Intenso
<b>05</b>	→	Después del medicamento
<b>06</b>	→	Siente estrés
<b>07</b>	→	Está enfermo
<b>08</b>	→	Siente hipoglucemia
<b>09</b>	→	Está en menstruación
<b>10</b>	→	Está de vacaciones
<b>11</b>	→	Otros

El usuario puede relacionar a la lectura con una de las diferentes actividades, a través del menú del glucómetro, después que se tomó la lectura. Los **caracteres 80 y 83** encierran al grupo entre comillas (“”).

- **Caracteres del 85 al 92:** son caracteres varios alfanuméricos y espacios que no contienen información pertinente a los objetivos del presente Trabajo de Graduación.
- **Caracteres 93 y 94:** no tienen representación gráfica definida, es por ello que se muestran como un Punto. Específicamente el **caracter 93** es un Retorno de Carro <CR> ó 0x0D en hexadecimal y el **caracter 94** es un Avance de Línea <LF> ó 0x0A en hexadecimal; dicho par, para este caso, representan el fin de la respuesta al comando de petición de tramas para el OneTouch Ultra2.

## 1.5 El OneTouch Ultra de Jhonson & Jhonson, Especificaciones y Tramas

Como lo es su pariente el OneTouch Ultra2, el OneTouch Ultra es un glucómetro del tipo biosensores, su tiempo de análisis es de 5 segundos, la muestra sanguínea capilar es calibrada como si fuese sangre de venas, para que el resultado obtenido sea directamente comparable con el resultado de un análisis de laboratorio utilizando sangre de venas. El OneTouch Ultra puede almacenar hasta 150 resultados en su memoria y es de uso personal o mono usuario. Igualmente el glucómetro no debe ser utilizado para diagnosticar diabetes ni debe ser utilizado en recién nacidos.

En la **Figura 1.5** se muestra una imagen del OneTouch Ultra.



**Figura 1. 5 Imagen del OneTouch Ultra**

La **Tabla 1.6** muestra las especificaciones técnicas del OneTouch Ultra de LifeScan de Jhonson & Jhonson.

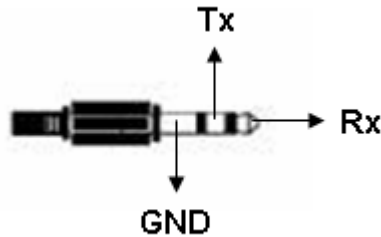
**Tabla 1. 6 Especificaciones técnicas del OneTouch Ultra2**

Tipo de Muestra	Sangre Pura Capilar
Volumen de la Muestra	Mínimo 1 uL
Rango de Análisis	20 - 600 mg/dl
Tiempo de Lectura	5 segundos
Calibración	Plasma Equivalente
Humedad de Operación	10 - 90%
Capacidad de Memoria	150 Resultados
Temperatura de Operación	6 – 44 °C
Tipo de Pantalla	LCD
Medida	3.12×2.25×0.85 pulg.
Peso (con batería)	1.5 oz.
Fuente de Energía	1 Batería de 3V Li (CR2032)
Apagado automático	2 min. de inactividad
Unidad de Glucosa	mg/dl
Característica multiusuario	No soportada

El software propietario para descargar los datos almacenados en la memoria del OneTouch Ultra al ordenador, es el mismo que utiliza el OneTouch Ultra2. Se ejecutaron el software propietario y el programa espía del puerto serial, mientras se realizaba una descarga de datos desde el glucómetro; en dicha actividad se observaron todos los comandos y respuestas entre el software propietario y el glucómetro y se determinó que la comunicación con el OneTouch Ultra, al igual que en el OneTouch Ultra2, es RS232 a niveles TTL a 9600 baudios, 8 bits de datos, sin paridad y un bit de paro. El puerto serial del OneTouch Ultra,



igualmente, consiste en un conector hembra para una espiga estéreo de 3.5 mm., la distribución de señales en la espiga que se conecta al OneTouch Ultra es la misma que la del OneTouch Ultra2, y se muestra en la **Figura 1.6**.



**Figura 1. 6 Señales en la espiga estéreo que se conecta al OneTouch Ultra**

Además se determinó que el comando de solicitud de información del OneTouch Ultra es igual que el del OneTouch Ultra2, con las diferencias que dicho comando debe ser enviado dos veces; la primera es para que el glucómetro entre en modalidad PC y la segunda es para que responda a la solicitud de las tramas. Otra diferencia es que el tiempo de espera entre el envío de un carácter y otro por el puerto serial, en contraste con el OneTouch Ultra2, que es capaz de recibir los caracteres con un tiempo de por medio de 2 milisegundos, para el OneTouch Ultra debe ser entre 14 y 28 milisegundos. El comando de solicitud de información se presenta en la **tabla 1.7**.

**Tabla 1. 7 Comando de solicitud de información para el OneTouch Ultra**

Caracter #	1	2	3	4	5
Comando	.	.	D	M	P

Los **caracteres 1 y 2** de la tabla anterior no poseen representación gráfica definida, es por ello que se muestran con un punto. El **caracter 1** es el número hexadecimal 0x11 y el **caracter 2** es un Retorno de Carro <CR> o en hexadecimal el número 0x0D. Los **caracteres 3, 4 y 5** son letras mayúsculas.

Para poder determinar y comprobar la información que envía el glucómetro ante la solicitud de tramas es bueno conocer los parámetros de configuración y lecturas almacenadas en la memoria del glucómetro antes de la petición de datos. Estos parámetros y lecturas presentes son:

Utilizando un OneTouch Ultra nuevo, sin lecturas almacenadas en su memoria, se configuró de la siguiente manera:

- Fecha/Hora: 2006/10/13 02:51 p.m.
- Unidades del resultado: mg/dL

Posteriormente se realizó una lectura o análisis de glucemia, que arrojó los siguientes resultados:

- Fecha: 10/13/06 (MM/DD/AA)
- Hora: 14:52:17 (hh:mm:ss)
- Resultado: 117 (mg/dL)

La respuesta ante el comando de solicitud de tramas del OneTouch Ultra es muy similar a la respuesta del OneTouch Ultra2. Para el OneTouch Ultra la trama es más corta, ya que el glucómetro no soporta la asignación de una actividad específica relacionada con las lecturas, es decir, el OneTouch Ultra no es capaz de especificar si la lectura se realizó antes o después de la comida. En la **Tabla 1.8** se muestra la trama de respuesta del OneTouch Ultra después de enviado el comando de solicitud de tramas:

**Tabla 1. 8 Trama de respuestas ante el comando de solicitud de tramas del OneTouch Ultra**

<b>Caracter #</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>
<b>Trama</b>	P		0	0	1	,	"	V	F	W	9	E	7	8	G
<b>Caracter #</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
<b>Trama</b>	T	"	,	"	M	G	/	D	L		"		0	5	B
<b>Caracter #</b>	<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>	<b>36</b>	<b>37</b>	<b>38</b>	<b>39</b>	<b>40</b>	<b>41</b>	<b>42</b>	<b>43</b>	<b>44</b>	<b>45</b>
<b>Trama</b>	1	.	.	P		"	F	R	I	"	,	"	1	0	/
<b>Caracter #</b>	<b>46</b>	<b>47</b>	<b>48</b>	<b>49</b>	<b>50</b>	<b>51</b>	<b>52</b>	<b>53</b>	<b>54</b>	<b>55</b>	<b>56</b>	<b>57</b>	<b>58</b>	<b>59</b>	<b>60</b>
<b>Trama</b>	1	3	/	0	6	"	,	"	1	4	:	5	2	:	1
<b>Caracter #</b>	<b>61</b>	<b>62</b>	<b>63</b>	<b>64</b>	<b>65</b>	<b>66</b>	<b>67</b>	<b>68</b>	<b>69</b>	<b>70</b>	<b>71</b>	<b>72</b>	<b>73</b>	<b>74</b>	<b>75</b>
<b>Trama</b>	7				"	,	"			1	1	7		"	,
<b>Caracter #</b>	<b>76</b>	<b>77</b>	<b>78</b>	<b>79</b>	<b>80</b>	<b>81</b>	<b>82</b>	<b>83</b>	<b>84</b>	<b>85</b>					
<b>Trama</b>		0	0		0	8	3	2	.	.					

De la Tabla anterior podemos observar que:

- **Caracter 1:** es la letra P mayúscula, y sirve como caracter de ACK, y determina que el comando fue recibido, además es repetido como el primer caracter de cada nueva línea (después del ASCII de Avance de Línea <LF>) en la respuesta del comando de petición de tramas, tal es el caso del **caracter 34**.
- **Caracter 2:** es un caracter de separación, específicamente un Espacio en Blanco <SP> ó 0x20 en hexadecimal que va después del caracter "P" en cada línea de respuesta al comando de petición de tramas, igual al **caracter 35**.

- **Caracteres del 3 al 5:** informan sobre el número de lecturas almacenadas en la memoria del glucómetro. Para este caso, 1 lectura.
- **Caracter 6:** es un caracter de separación, específicamente es una coma (,).
- **Caracteres del 7 al 17:** detallan el número serial del glucómetro dado por el fabricante. Dicho número serial está compuesto por 9 caracteres ASCII (**caracteres del 8 al 16**) y están encerrado por comillas (“, **caracteres 7 y 17**). Para este caso se utilizó el OneTouch Ultra con el número serial “VFW9E78GT”.
- **Caracteres del 19 al 26:** informan sobre las unidades en las que se expresa la lectura de glucemia. Las unidades son determinadas en los **caracteres del 20 al 24**; para este caso mg/dL, y le sigue un Espacio en Blanco <SP> o 0x20 en hexadecimal (**caracter 25**); todo el grupo de caracteres está encerrado por comillas (“, **caracteres 19 y 26**).
- **Caracteres del 27 al 31:** son 4 caracteres alfanuméricos (**caracteres del 28 al 31**) que no contienen información útil para la finalidad del presente trabajo de graduación, previo a éstos, se encuentra un Espacio en Blanco <SP> ó 0x20 en hexadecimal (**caracter 27**).
- **Caracteres 32 y 33:** no tienen representación gráfica definida, es por ello que se muestran como puntos. Son caracteres que indican el inicio de una nueva línea. Específicamente el **caracter 32** es un Retorno de Carro <CR> ó 0x0D en hexadecimal y el **caracter 33** es un Avance de Línea <LF> ó 0x0A en hexadecimal. Son iguales a los **caracteres 84 y 85** únicamente que ellos especifican el final de la respuesta al comando de petición de tramas. Si el glucómetro tuviera más de una lectura almacenada, los caracteres 93 y 94 representarían el inicio de una nueva línea como lo hacen los caracteres 32 y 33; así la trama de respuesta al comando de petición de tramas es de longitud variable y siempre finaliza con un Retorno de Carro <CR>, seguido de un Avance de Línea <LF>.
- **Caracteres del 36 al 40:** especifican el día en que se realizó la lectura. Específicamente los **caracteres del 37 al 39** determinan el día por sus letras en inglés de la siguiente manera:

<b>MON</b>	→	lunes
<b>TUE</b>	→	martes
<b>WED</b>	→	miércoles
<b>THU</b>	→	jueves
<b>FRI</b>	→	viernes
<b>SAT</b>	→	sábado
<b>SUN</b>	→	domingo

Para el caso, la lectura de prueba fue tomada el día viernes. Los caracteres que determinan el día están encerrados por comillas (“, **caracteres 36 y 40**).

- **Caracter 41:** es un caracter de separación, específicamente es una coma (,), igual que los **caracteres 52, 66, 75, 79 y 84**.
- **Caracteres del 42 al 51:** comunican la fecha en que se realizó la lectura. Específicamente los **caracteres del 43 al 50** determinan la fecha en el formato MM/DD/AA. Para este caso, la lectura se tomó el 13 de octubre de 2006. Los **caracteres 42 y 51** solamente encierran la fecha entre comillas (“).
- **Caracteres del 53 al 65:** expresan la hora en que se tomo la lectura. Específicamente los **caracteres del 54 al 61** determinan la hora en el formato hh:mm:ss. Para este caso, la lectura se tomó a las 2 de la tarde con 52 minutos y 17 segundos. Los **caracteres del 62 al 64** solamente son Espacios en Blanco <SP> o 0x20 en hexadecimal, y los **caracteres 53 y 65** encierran al grupo entre comillas (“).
- **Caracteres del 67 al 74** se explican a continuación:
  - ✓ **Caracter 68:** determina si la prueba se realizó utilizando la solución de control o no. Si se utilizó la solución de control el caracter es una “C” (mayúscula); sino se utilizó la solución de control, el caracter es un espacio en Blanco <SP> ó 0x20 en hexadecimal como es el caso para esta prueba. El usuario puede agregar el comentario a la lectura, de que ésta fue realizada usando solución de control; eso se hace desde el menú del glucómetro, después de tomada la lectura.
  - ✓ **Caracter 69:** es un espacio en Blanco <SP> ó 0x20 en hexadecimal de separación al igual que el **caracter 73**.
  - ✓ **Caracteres del 70 al 72:** son la lectura de glucemia. Para este caso la lectura fue de 117 mg/dL.
  - ✓ **Caracteres 67 y 74:** encierran a todo el grupo entre comillas (“).
- **Caracteres del 75 al 83:** son caracteres varios alfanuméricos y espacios que no contienen información pertinente a los objetivos del presente Trabajo de Graduación.
- **Caracteres 84 y 85:** no tienen representación gráfica definida, es por ello que se muestran como un Punto. Específicamente el **caracter 84** es un Retorno de Carro <CR> ó 0x0D en hexadecimal y el **caracter 85** es un Avance de Línea <LF> ó 0x0A en hexadecimal; dicho par, para este caso, representan el fin de la respuesta al comando de petición de tramas para el OneTouch Ultra.

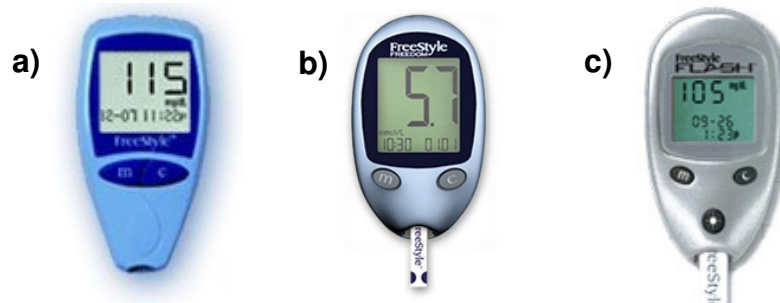
## 1.6 Otros Glucómetros

En esta sección se presentan las características y tramas de comunicación de algunos glucómetros que fueron investigados durante la tesis, pero no se incluyeron en el código fuente del dispositivo electrónico, ni se tomaron en cuenta para el diseño del hardware de éste, por limitantes del marco de tiempo.

### 1.6.1. Glucómetros de la familia FreeStyle

Los glucómetros de la familia FreeStyle son: el FreeStyle, el FreeStyle Freedom y el FreeStyle Flash; todos son del tipo biosensores, las pruebas de glucosa se realizan con 0.3 micro litros de sangre capilar pura y su resultado es calibrado a sangre capilar, el tiempo de análisis promedio es de 7 segundos.

En la **Figura 1.7** se muestran las imágenes de los glucómetros de la familia FreeStyle.



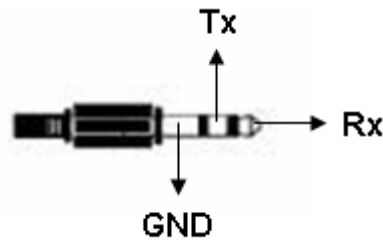
**Figura 1.7 a) FreeStyle. b) FreeStyle Freedom. c) FreeStyle Flash.**

La **Tabla 1.9** muestra las especificaciones técnicas generales de la familia de glucómetros FreeStyle.

Todos los glucómetros de la familia FreeStyle utilizan el mismo software propietario para descargar los datos almacenados en sus memorias al ordenador. Se ejecutaron el software propietario y el programa espía del puerto serial mientras se realizaban descargas de datos desde cada glucómetro de la familia FreeStyle; en dicha actividad se observaron todos los comandos y respuestas entre el software propietario y los glucómetros y se determinó que la comunicación para todos los glucómetros de la familia es RS232 a 19200 baudios, 8 bits de datos, sin paridad y un bit de paro. El puerto serial de los glucómetros de la familia FreeStyle consiste en un conector hembra para una espiga estéreo de 2.5 mm., la distribución de señales en la espiga que se conecta a los glucómetros de la familia FreeStyle se muestra en la **Figura 1.8**.

**Tabla 1. 9 Especificaciones técnicas generales de la familia FreeStyle**

Tipo de Muestra	Sangre Pura Capilar
Volumen de la Muestra	Mínimo 0.3 uL
Rango de Análisis	20 - 500 mg/dl
Tiempo de Lectura	7 segundos
Calibración	Plasma Equivalente
Humedad de Operación	5% - 90%
Capacidad de Memoria	250 Resultados
Temperatura de Operación	5 °C – 40 °C
Tipo de Pantalla	LCD
Fuente de Energía	1 ó 2 baterías de 3V Li (CR2032)
Apagado automático	2 min. de inactividad
Unidad de Glucosa	mg/dl
Característica multiusuario	No soportada



**Figura 1. 8 Señales en la espiga estereo que se conecta a los glucómetros de la familia FreeStyle.**

Además se determinó que el comando de solicitud de información para los glucómetros de la familia FreeStyle es el mismo y se presenta en la **tabla 1.10**.

**Tabla 1. 10 Comando de solicitud de información para los glucómetros de la familia FreeStyle**

<b>Caracter #</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>Comando</b>	m	e	m

Para poder determinar y comprobar la información que envían los glucómetros ante la solicitud de tramas es bueno conocer los parámetros de configuración y lecturas almacenadas en la memoria de los glucómetros antes de la petición de datos. Los parámetros y lecturas presentes en uno de los glucómetros de la familia FreeStyle fueron los siguientes:

Utilizando un FreeStyle Freedom nuevo, sin lecturas almacenadas en su memoria, se realizó una lectura o análisis de glucemia, que arrojó los siguientes resultados:

- Fecha: 02/14/07 (MM/DD/AA)
- Hora: 21:06 (hh:mm)
- Resultado: 110 (mg/dL)

La respuesta ante el comando de solicitud de tramas de los glucómetros de la familia FreeStyle es la misma. En la **Tabla 1.11** se muestra como ejemplo la trama de respuesta del FreeStyle Freedom después de enviado el comando de solicitud de tramas:

**Tabla 1. 11 Trama de respuestas ante el comando de solicitud de tramas de los glucómetros de la familia FreeStyle**

<b>Caracter #</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>
<b>Trama</b>	.	.	C	A	G	H	1	6	1	-	B	5	5	2	5
<b>Caracter #</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
<b>Trama</b>	.	.	3	.	1	0	-	P	.	.	F	e	b		
<b>Caracter #</b>	<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>	<b>36</b>	<b>37</b>	<b>38</b>	<b>39</b>	<b>40</b>	<b>41</b>	<b>42</b>	<b>43</b>	<b>44</b>	<b>45</b>
<b>Trama</b>	2	4		2	0	0	7		1	0	:	0	3	:	3
<b>Caracter #</b>	<b>46</b>	<b>47</b>	<b>48</b>	<b>49</b>	<b>50</b>	<b>51</b>	<b>52</b>	<b>53</b>	<b>54</b>	<b>55</b>	<b>56</b>	<b>57</b>	<b>58</b>	<b>59</b>	<b>60</b>
<b>Trama</b>	5	.	.	0	0	1	.	.	.	1	1	0			F
<b>Caracter #</b>	<b>61</b>	<b>62</b>	<b>63</b>	<b>64</b>	<b>65</b>	<b>66</b>	<b>67</b>	<b>68</b>	<b>69</b>	<b>70</b>	<b>71</b>	<b>72</b>	<b>73</b>	<b>74</b>	<b>75</b>
<b>Trama</b>	e	b			1	4		2	0	0	7		2	1	:
<b>Caracter #</b>	<b>76</b>	<b>77</b>	<b>78</b>	<b>79</b>	<b>80</b>	<b>81</b>	<b>82</b>	<b>83</b>	<b>84</b>	<b>85</b>	<b>86</b>	<b>87</b>	<b>88</b>	<b>89</b>	<b>90</b>
<b>Trama</b>	0	6		1	7		0	X	0	0	.	.	0	X	2
<b>Caracter #</b>	<b>91</b>	<b>92</b>	<b>93</b>	<b>94</b>	<b>95</b>	<b>96</b>	<b>97</b>	<b>98</b>							
<b>Trama</b>	3	1	C			E	N	D							

De la Tabla anterior podemos observar que:

- **Caracteres 1 y 2:** no tienen representación gráfica definida, es por ello que se muestran como puntos. Son caracteres que indican el inicio de una nueva línea. Específicamente el **caracter 1** es un Retorno de Carro <CR> ó 0x0D en hexadecimal y el **caracter 2** es un Avance de Línea <LF> ó 0x0A en hexadecimal. Son iguales a los pares de **caracteres 16 y 17, 24 y 25, 47 y 48, 52 y 53, y 86 y 87.**
- **Caracteres del 3 al 15:** muestran el número de serie asignado por el fabricante para cada glucómetro; en este caso el número de serie del glucómetro es el CAGH161-B5525.

- **Caracteres del 18 al 23:** son caracteres varios alfanuméricos y no contienen información pertinente a los objetivos del presente Trabajo de Graduación.
- **Caracteres del 26 al 46:** presentan la fecha y la hora actual que marca el glucómetro, para éste caso presenta las 10:03 A.M. (con 3 segundos) del día 24 de febrero de 2007.
- **Caracteres del 49 al 51:** especifican el número total de lecturas contenidas en la memoria del glucómetro, para éste caso es 1 lectura.
- **Caracter 54:** que precede a un Retorno de Carro <CR> (caracter 52) y a un avance de línea <LF> (caracter 53), es un avance de línea extra <LF> ó 0x0A en hexadecimal.
- **Caracteres del 55 al 57:** determinan el nivel de glucosa de la primera lectura en la trama de respuesta en unidades de mg/dL; para éste caso la lectura es de 110 mg/dL.
- **Caracteres 58, 59, 63, 64, 67 y 72:** son espacios en blanco de separación.
- **Caracteres del 60 al 62:** especifican el mes en que se realizó la lectura de la siguiente manera:

<b>Jan</b>	→	Enero
<b>Feb</b>	→	Febrero
<b>Mar</b>	→	Marzo
<b>Apr</b>	→	Abril
<b>May</b>	→	Mayo
<b>Jun</b>	→	Junio
<b>Jul</b>	→	Julio
<b>Aug</b>	→	Agosto
<b>Sep</b>	→	Septiembre
<b>Oct</b>	→	Octubre
<b>Nov</b>	→	Noviembre
<b>Dic</b>	→	Diciembre

Para el caso, la lectura de prueba fue tomada en el mes de Febrero.

- **Caracteres 65 y 66:** determinan la fecha en que se realizó la lectura; para el caso fue el 14 del mes respectivo.
- **Caracteres del 68 al 71:** informan sobre el año en que se realizó la lectura; para el caso fue en el año 2007.



- **Caracteres del 73 al 77:** muestran la hora en que se realizó la lectura; para el caso la lectura se realizó a las 9 de la noche con 6 minutos.
- **Caracteres del 78 al 85:** son caracteres varios alfanuméricos y espacios que no contienen información pertinente a los objetivos del presente Trabajo de Graduación.
- **Caracteres del 88 al 95:** son caracteres varios alfanuméricos y espacios que no contienen información pertinente a los objetivos del presente Trabajo de Graduación.
- **Caracteres del 96 al 98:** su ubicación depende del número total de lecturas almacenadas en la memoria del glucómetro (para este caso solamente 1 lectura estaba almacenada) y forman la palabra fin en inglés (**END**) que significa que la trama de respuesta al comando de solicitud de tramas del glucómetro ha finalizado.

## 1.7 Protocolo RS-232

El puerto serie RS-232C, presente en la mayoría de los ordenadores actuales, es la forma más común de realizar transacciones de datos. El RS-232C es un estándar que constituye la tercera revisión de la norma RS-232, propuesta por la EIA (Asociación de Industrias Electrónicas), realizándose posteriormente una versión internacional por el CCITT conocida como V.24. Las diferencias entre ambas son mínimas, por lo tanto a veces se habla indistintamente de V.24 y de RS-232C, refiriéndose al mismo estándar.

El terminal RS-232 consiste de un conector DB-25 de 25 pines, aunque es normal encontrar la versión en DB-9 de 9 pines, más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie del PC). Las señales con las que trabaja este puerto serie son digitales, de +12V (0 lógico) y -12V (1 lógico), para la entrada y salida de datos, y a la inversa en las señales de control. El estado de reposo en la entrada y salida de datos es -12V.

El control de la comunicación serie, y por consiguiente del puerto serie, se lleva a cabo mediante un circuito integrado específico denominado UART (Transmisor Receptor Asíncrono Universal); del cual existen los siguientes modelos:

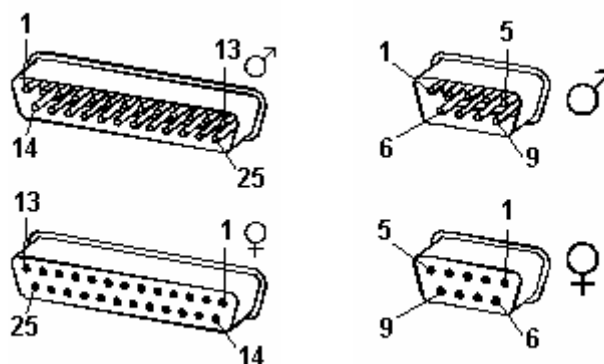
8250	Bastante antiguo, con fallos, solo llega a 9600 Baudios
16450	Versión corregida del 8250, llega hasta 115200 Baudios
16550A	Versión de alta velocidad, posee buffers de E/S.

En la **Tabla 1.12** se muestra la correspondencia de los pines del terminal DB-9 con la señal que manejan y su pin equivalente en el terminal DB-25.

**Tabla 1. 12 Correspondencia de pines y señales del puerto serial**

DESCRIPCION	SEÑAL	DB-9	DB-25	E/S
CHASIS		1	1	
TRANSMIT DATA	TXD	3	2	S
RECEIVE DATA	RXD	2	3	E
REQUEST TO SEND	RTS	7	4	S
CLEAR TO SEND	CTS	8	5	E
DATA SET READY	DSR	6	6	E
SIGNAL GROUND	SG	5	7	
(DATA) CARRIER DETECT	CD/DCD	1	8	E
TRANSMIT CLOCK	TXC*		15	S
RECEIVE CLOCK	RXC*		17	E
DATA TERMINAL READY	DTR	4	20	S
RING INDICATOR	RI	9	22	E
TRANSMIT/RECEIVE CLOCK	RTXC*		24	S

En la **Figura 1.9** se muestra el arreglo o distribución física que tienen los pines para los terminales DB25 y DB9.



**Figura 1. 9 Configuración física para el puerto RS-232 DB25 y DB9**

El RS-232 puede transmitir los datos en grupos de 5, 6, 7 u 8 bits, a unas velocidades determinadas (normalmente, 4800 ó 9600 bits por segundo). Después de la transmisión de los datos, le sigue un bit opcional de paridad que indica si el número de bits transmitidos es par o impar para detectar fallos, y después 1 o 2 bits de paro. Normalmente, el protocolo utilizado es 8N1 que significa, 8 bits de datos, sin paridad y con 1 bit de paro.

Una vez que ha comenzado la transmisión de un dato, los bits tienen que llegar uno detrás de otro a una velocidad constante y en determinados instantes de tiempo. Por eso se dice que el RS-232 es asincrónico por carácter y sincrónico por bit. Los pines que portan los datos son RXD y TXD. Las demás se encargan

de otros trabajos: DTR indica que el ordenador está encendido, DSR que el aparato conectado a dicho puerto está encendido, RTS que el ordenador puede recibir datos (porque no está ocupado), CTS que el aparato conectado puede recibir datos, y DCD detecta que existe una comunicación.

Tanto el aparato a conectar como el ordenador (o el programa terminal) tienen que usar el mismo protocolo serie para comunicarse entre sí. Puesto que el estándar RS-232 no permite indicar en que modo se está trabajando, es el usuario quien tiene que decidirlo y configurar ambas partes.

Los parámetros que hay que configurar son: cantidad de bits de palabra, paridad, cantidad de bits de paro, velocidad del puerto serie, y protocolo de control de flujo. Este último puede ser por hardware (handshaking RTS/CTS) o bien por software (XON/XOFF, el cual no es muy recomendable ya que no se pueden realizar transferencias binarias).

## 1.8 CONCLUSIONES DEL CAP. I

El GlucoLab es un glucómetro multiusuario a diferencia del OneTouch Ultra, OneTouch Ultra2 y la familia de glucómetros FreeStyle que son mono usuario.

La comunicación se da a 4800 Baudios para el GlucoLab, a 9600 baudios para el OneTouch Ultra y el OneTouch Ultra2, y a 19200 baudios para la familia de glucómetros FreeStyle; todos ellos usan palabras de 8 bits, sin bit de paridad y con un bit de paro en la comunicación serial.

La comunicación serial de los glucómetros GlucoLab, OneTouch Ultra y Ultra2 es como la de cualquier dispositivo que se pueda comunicar utilizando el estándar RS-232, con la única diferencia que los niveles de voltaje que manejan son niveles TTL, es decir, de 0 a 3 voltios; esto representa una marcada diferencia a los glucómetros de la familia FreeStyle ya que ellos utilizan niveles de voltaje RS232 en su comunicación serial, es decir, de -15 a +15 voltios.

Los puertos seriales de los glucómetros manejan las señales de Transmisión (TX), Recepción (RX) y Tierra de Señal (GND), las cuales son las señales mínimas necesarias para establecer la comunicación serial funcional.

El comando de petición de tramas para el GlucoLab es el ASCII "S", para el OneTouch Ultra y el OneTouch Ultra2 es la secuencia de 5 caracteres ASCII: 0x11, 0x0D, D, M, P; y para la familia de glucómetros FreeStyle es la cadena ASCII "mem".

Las respuestas de los glucómetros ante comandos de petición de tramas son de longitud variable dependiendo del número de lecturas almacenadas en la memoria de los glucómetros.

Cada glucómetro posee un mecanismo para indicar que la trama perteneciente a una lectura ha finalizado, y que otra trama nueva sigue a continuación, o que la respuesta ante el comando de petición de tramas ha finalizado. Dicho mecanismo para el GlucoLab consiste en tres caracteres aleatorios que no representan información útil, es así el primero un número, el segundo puede ser un número o un carácter especial como por ejemplo: \*, ~, é, ñ, etc. Y el tercero es un carácter especial como los que ya se mencionaron; además se conoce que la longitud de la trama perteneciente a una lectura es de 22 caracteres para el GlucoLab.

El mecanismo para el OneTouch Ultra y el OneTouch Ultra2 consiste en dos caracteres al final de cada trama perteneciente a una lectura, el primero es un Retorno de Carro <CR> ó 0x0D en hexadecimal, y el segundo es un Avance de

Línea <LF> ó 0x0A en hexadecimal; además se conoce que la longitud de la trama perteneciente a una lectura es de 51 caracteres para el OneTouch Ultra y de 61 caracteres para el OneTouch Ultra2.

Al igual que el OneTouch Ultra y Ultra2 de Jhonson & Jhonson, la familia de glucómetros FreeStyle separa sus tramas por medio de un Retorno de Carro <CR> ó 0x0D en hexadecimal seguido por un Avance de Línea <LF> ó 0x0A en hexadecimal; y utiliza la cadena ASCII "END" para indicar que su respuesta ante la solicitud de tramas ha finalizado.

## 1.9 REFERENCIAS BIBLIOGRÁFICAS

ESTEVEES, Jorge Valentín. La Glucosa (en línea). Argentina, (citada el 6 de junio de 2006).

<http://www.mantra.com.sr/contenido/frame-glucosa.html>

FUNDACION MANOS DEL SUR. Nuestro Proyecto: El Salvador (en línea). San Salvador, El Salvador (citada el 28 junio de 2006).

<http://www.manosdelsur.org/espanol/proyecto-elsalvador.htm>

INFOSALUD. Diabetes (en línea). México, (citada el 19 de junio de 2006).

<http://www.lilly.com.mx/infoSalud/endocrinologia/monitoreos.asp>

MCG HEALTH INC. Diabetes. Estadística (en línea). Georgia, EEUU. Enero. 2005, (citada el 7 de junio de 2006).

<http://www.mcghealth.org/Greystone/sadult/diabetes/stats.html>

PEÑA, Gonzalo Martín, HENRIKSEN, Jan Erik, BECK-NIELSEN, Henning, CAMPBELL, Lan W, DAVEY, Patrick. Niveles de Glucosa en Sangre (en línea). España, sep. 2003, (citada el 12 de junio de 2006).

<http://www.netdoctor.es/html/000437.html>

QUIRANTES HERNÁNDEZ, Alberto. Papel de la Insulina en el tratamiento de la Diabetes Mellitus tipo II (en línea). La Habana, Cuba. Enero 1997 (citada el 7 de junio de 2006).

[http://bus.sld.cu/revistas/mgi/vol13\\_5\\_97/mgi12597.htm](http://bus.sld.cu/revistas/mgi/vol13_5_97/mgi12597.htm)

VILLAMIL, Carlos F. Glicemia: ¿Que es? (en línea). Colombia, (citada el 9 de junio de 2006).

[http://www.abcmedicus.com/articulo/pacientes/id/379/pagina/1/glicemia\\_que\\_es.html](http://www.abcmedicus.com/articulo/pacientes/id/379/pagina/1/glicemia_que_es.html)

## **CAPÍTULO II**

### **HARDWARE Y SOFTWARE DEL DISPOSITIVO ELECTRÓNICO**

#### **Introducción**

En el presente capítulo se detallan los requerimientos de hardware que hacen posible la extracción y envío de la información de los glucómetros hacia un ordenador remoto, destacando el componente corazón del dispositivo, el microcontrolador. El microcontrolador seleccionado fue el PIC18F4550 de la gama mejorada de Microchip. Este capítulo también contiene los flujogramas de ejecución del software cargado en el microcontrolador, explicando sus funciones, características, formas de uso del dispositivo, limitantes, etc.

El Capítulo II tiene como objetivos generales dar a conocer el dispositivo electrónico que hace posible la transacción de información, entre el glucómetro de un usuario (que podría estar en su casa o en cualquier otro lugar donde tenga acceso a una línea telefónica común) y la base de datos del médico del usuario (que podría estar en el ordenador de su clínica), la forma de uso de dicho dispositivo y su funcionamiento en niveles de hardware y software.

#### **2.1 Generalidades sobre microcontroladores PIC y el PIC18F4550**

Se denomina microcontrolador a un dispositivo programable, capaz de realizar diferentes actividades que requieran del procesamiento de datos digitales y del control y comunicación digital de diferentes dispositivos.

Los microcontroladores poseen una memoria interna que almacena dos tipos de datos; las instrucciones que corresponden al programa que se ejecuta, y los registros, es decir, los datos que el usuario maneja, así como registros especiales para el control de las diferentes funciones del microcontrolador.

Los microcontroladores se programan en lenguaje ensamblador y cada microcontrolador varía su conjunto de instrucciones de acuerdo a su fabricante y modelo. Microchip permiten la programación de sus microcontroladores utilizando directivas de Lenguaje C y Basic, a través de paquetes especiales

incluidos en su software de desarrollo MPLab, que se encarga de traducir el código a su equivalente en lenguaje ensamblador.

De acuerdo al número de instrucciones que el microcontrolador maneja, se le denomina de Arquitectura RISC (reducido) o CISC (complejo).

Están compuestos principalmente por:

Unidad Aritmética Lógica (ALU)

Memoria de programa

Memoria de registros (RAM e EEPROM<sup>5</sup>)

Pines de Entrada/Salida.

Módulos periféricos de E/S como el USART, I2C, Convertidores A/D, Temporizadores y otros.

La Unidad Aritmética Lógica, es la encargada de procesar los datos dependiendo de las instrucciones que se ejecutan, mientras que los pines son los que se encargan de comunicar al microcontrolador con el medio externo, alimentación del microcontrolador o para controles específicos.

Diferentes tipos de microcontroladores comerciales son:

PIC de Microchip

Atmel AVR

Intel 8051

Motorola Serie HC11

Scenix

El microcontrolador PIC (Peripheral Interface Controller); es fabricado por Microchip Technology Inc., y dependiendo del número de instrucciones, este se pueden clasificar de la siguiente manera:

Gama Baja	33 instrucciones
Gama Media	35 instrucciones
Gama Alta	75 instrucciones
Gama Mejorada	83 instrucciones

El microcontrolador PIC, posee una arquitectura Harvard, que consiste de un esquema en el que el CPU está conectado a dos memorias por medio de dos buses separados. Una de las memorias contiene solamente las instrucciones de programa (Memoria de Programa), mientras que la otra memoria solo almacena los datos (Memoria de Datos). Ambos buses son totalmente independientes y pueden tener un ancho diferente cada uno.

---

<sup>5</sup> Siglas de electrically-erasable programmable read-only memory.



Dos características importantes de ésta arquitectura son:

- Como el tamaño de las instrucciones no está relacionado con el de los datos, aquel puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa; logrando así mayor velocidad y menor longitud de programa.
- El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando así una mayor velocidad de operación.

Sin embargo con la arquitectura Harvard, los microprocesadores deben poseer instrucciones especiales para el acceso a tablas de valores y constantes que pudieran ser necesarias en los programas.

Esta estructura, acompañada de un pipeline (ejecución y lectura paralelas), permite que las instrucciones se ejecuten en un sólo ciclo de reloj, salvo en el caso de saltos/rupturas de programa. Lo que se hace internamente es que mientras se ejecuta la instrucción actual se carga la siguiente instrucción en el pipeline, con lo que se alcanza una alta velocidad de ejecución.

En cuanto a los registros los PIC pueden direccionarse tanto directa como indirectamente<sup>6</sup>. Todos los registros de función especial, incluyendo el contador de programa, están accesibles en la memoria de datos y pueden ser operados empleando diferentes modos de direccionamiento (ortogonalización).

Igualmente, los diferentes módulos del microcontrolador de Microchip (temporizadores, puertos de e/s, USART, etc.) son configurados con registros de funciones especiales contenidos en la RAM del mismo.

Los microcontroladores PIC poseen módulos de comunicación que le permiten conectarse con otros dispositivos manejando el mismo protocolo; tal es el caso de los módulos: USART<sup>7</sup> para manejar protocolo RS232 e I2C para manejo de protocolo I2C.

Puede decirse que el PIC es ampliamente aceptado debido a:

- Software de desarrollo gratis
- Repertorio de instrucciones reducido y simple
- Gran variedad de dispositivos que se ajustan a necesidades específicas
- Excelentes características de arquitectura interna

---

<sup>6</sup> Diríjase al manual PIC18F4550 Data Sheet, p. 72 (Anexo A)

<sup>7</sup> Diríjase al manual PIC18F4550 Data Sheet, p. 237 (Anexo A)

Todas estas razones han hecho del PIC un dispositivo muy utilizado en aplicaciones tales como:

- Sistemas de Control
- Automatismo Industrial
- Monitoreo y Comunicaciones
- Aplicaciones Automotrices

Entre las características más sobresalientes que se tomaron en cuenta para la selección del PIC18F4550, y que simplificaron en gran manera el código fuente de la aplicación, se pueden enlistar:

- Memoria de programa de 32 Kbytes
- Bus de instrucciones de 16 bits
- Repertorio de 75 instrucciones más 8 instrucciones del repertorio extendido
- Memoria RAM de 2048 bytes
- Bus de memoria RAM e EEPROM de 8 bits
- Memoria EEPROM de 256 bytes
- Posee 5 puertos de entrada / salida
- Protocolo I2C implementado en Firmware<sup>8</sup>
- Protocolo de comunicación serial implementado en Firmware
- Posee un temporizador / contador de 8 bits y 3 temporizadores / contadores de 16 bits
- Posee un multiplicador de frecuencia de reloj, el cual le permite trabajar a velocidades de hasta 48 MHz a partir de cristales de 4 MHz
- Multiplicación de dos bytes con una sola directiva ejecutada en un ciclo de instrucción
- Direccionamiento indirecto a toda la RAM con una sola directiva con incrementos / decrementos automáticos de registros de punteros, todo en un ciclo de instrucción
- Direccionamiento indirecto utilizando saltos relativos

El pinout del integrado PIC18F4550, así como información general sobre sus diferentes módulos y características, se encuentra en el **Anexo A**.

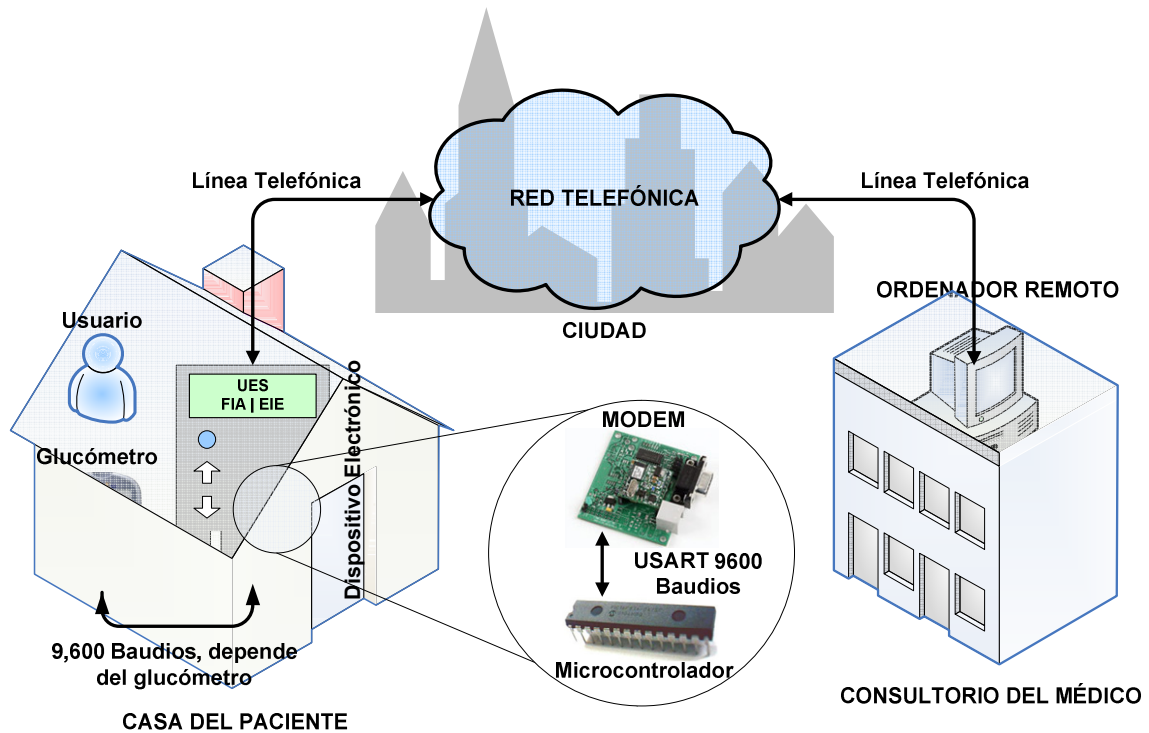
## 2.2 Generalidades del dispositivo electrónico

El dispositivo electrónico actualmente es de tipo mono usuario, será utilizado por un paciente diabético para enviar su historial de glucemia al médico que trata su condición de salud. Así dicho dispositivo se encarga de extraer la información

---

<sup>8</sup> Bloque de instrucciones de programa guardado en la memoria ROM, que establece la lógica de más bajo nivel, que controla los circuitos electrónicos de un dispositivo de cualquier tipo.

almacenada en la memoria de los glucómetros (GlucoLab, OneTouch Ultra2 ó OneTouch Ultra), y enviarla a través de una llamada telefónica, hasta la computadora del médico, la cual manipulará dicha información utilizando un servicio que atenderá la llamada e introducirá los datos recibidos a una base de datos; el proceso antes descrito está representado en la **Figura 2.1**.



**Figura 2. 1 Esquema general del funcionamiento del sistema electrónico - informático**

El dispositivo electrónico es capaz de enviar más de 100 lecturas en una sola transacción de información, pero ello incrementaría las posibilidades de corrupción de caracteres en la transmisión, debido al ruido en las líneas telefónicas, además el máximo provecho del sistema es obtenido por el usuario cuando él envía sus lecturas con regularidad, cada día o cada semana, así el médico puede llevar un control ordenado de su tratamiento; se recomienda no dejar acumular más de 30 lecturas sin ser enviadas.

El paciente podrá corroborar que el sistema informático remoto recibió sin problemas sus lecturas de glucemia, e igualmente, visualizar su prescripción médica, mediante una aplicación de consulta web que se presenta en el **Capítulo III** del presente documento. Si el ordenador remoto recibe información corrupta, debido a ruido en la línea telefónica u otra causa, éste es capaz de detectarlo y no almacenará la información en la base de datos; también enviará un código de error de vuelta al dispositivo electrónico, para que éste finalice la

llamada telefónica y presente un mensaje de error en el display LCD, permitiéndole saber al usuario que debe intentar nuevamente conectarse con el ordenador remoto.

Cabe destacar que la información proveniente de los glucómetros es recibida por el dispositivo en formatos diferentes, esto depende del fabricante del glucómetro. Como se menciona en el Capítulo I, actualmente el dispositivo electrónico puede comunicarse con los glucómetros GlucoLab de Infopía, OneTouch Ultra y OneTouch Ultra2 de LifeScan de Jhonson y Jhonson. En la **Figura 2.2** se muestra una fotografía del dispositivo electrónico utilizado por el paciente con diabetes.



**Figura 2. 2 Fotografía del dispositivo electrónico**

Luego de recibir la información almacenada en la memoria de los glucómetros y antes de transmitirla vía MODEM, el dispositivo electrónico se encarga de normalizar la información proveniente de los glucómetros a tramas simplificadas, más cortas y funcionales, para que puedan ser fácilmente manejadas por el servicio que atenderá la llamada telefónica e introducirá la información en la base de datos. El dispositivo electrónico empaqueta cada lectura de glucemia en el formato que se muestra en la **Tabla 2.1**.

De la **Tabla 2.1** se puede observar que la trama que envía el dispositivo electrónico, por cada lectura de glucemia, consta de 15 caracteres, siendo:

**Tabla 2. 1 Trama que envía el dispositivo electrónico por cada lectura de glucemia**

Caracter #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Trama		A	Y	Y	M	M	D	D	h	h	m	m	L	L	L

- **Caracter 1:** es un Espacio en Blanco <SP> ó 0x20 en hexadecimal, y sirve como separador entre tramas consecutivas.

- **Caracter 2:** es el caracter de actividad y se utiliza únicamente para especificar si la lectura se tomó antes o después de la comida, dependiendo de su valor, de la siguiente manera:
  - 0 → La lectura se tomó antes de comer ó no se especifica
  - 1 → La lectura se tomó después de comer
- **Caracteres del 3 al 8:** determinan la fecha en que se tomo la lectura. Así los **caracteres 3 y 4** especifican el año, los **caracteres 5 y 6** determinan el mes y los **caracteres 7 y 8** detallan la fecha en que se tomó la lectura.
- **Caracteres del 9 al 12:** informan sobre la hora en que se tomó dicha lectura. Específicamente los **caracteres 9 y 10** comunican la hora en un formato de 24 horas, y los **caracteres 11 y 12** especifican los minutos en que se tomó la lectura.
- **Caracteres del 13 al 15:** contienen el resultado de la lectura de glucemia en unidades de mg/dL.

La comunicación serial entre el MODEM y el microcontrolador está configurada a 9600 baudios, 8 bits de datos, 1 bit de parada y ninguna paridad.

La transacción de información entre el dispositivo electrónico ó **Cliente <C>**, del lado del usuario, y el **Servicio <S>** ejecutándose en el ordenador del médico que trata la condición de salud del usuario, en medio de una llamada telefónica posee varios pasos generales, en la **Tabla 2.2** se detallan los pasos generales en una transacción de información exitosa.

**Tabla 2. 2 Pasos generales en una transacción de información exitosa**

Paso #	Ejecutor	Acción
1	<C>	Llama al número telefónico de la línea que está conectada al MODEM del ordenador del médico.
2	<S>	Toma la llamada entrante.
3	<C>	Envía el número de serie identificador del dispositivo electrónico.
4	<S>	Envía la fecha y la hora de la última lectura registrada en la base de datos que fue recibida específicamente de ese dispositivo.
5	<C>	Envía sólo las lecturas con fechas más recientes.
6	<S>	Envía la secuencia de caracteres para indicarle al Cliente que la transacción de información ha sido exitosa.
7	<C>	Termina la llamada telefónica.

De la **Tabla 2.2** se amplía la siguiente información:

- **Paso 3:** el número serial que envía el Cliente está compuesto actualmente de 8 caracteres numéricos y se utiliza para que el Servicio identifique a cada

Cliente individualmente; así la base de datos contiene información útil referente al dueño de ese dispositivo electrónico, tales como nombre, sexo, edad, glucómetro que utiliza, medicamento, historial de glucemia, etc. Seguido del número de serie, el Cliente envía un Retorno de Carro <CR> ó 0x0D en hexadecimal, para indicarle al Servicio que el número serial ha llegado a su final.

- **Paso 4:** Si el Servicio identificó exitosamente al Cliente por su número de serie, la fecha y hora de la última lectura que envió el Cliente, en una transacción de información previa a la actual, y que está registrada en la base de datos del ordenador del médico, es transmitida por el Servicio en el formato YYMMDDhhmm, el primer par de caracteres determinan el año, el segundo par el mes, el tercer par la fecha, el cuarto par la hora y el quinto par los minutos. Dicha fecha y hora es utilizada por el Cliente para filtrar las tramas que extrajo del glucómetro y solo enviar al Servicio las tramas que tengan una fecha más reciente. Si se tratara de un Cliente nuevo que nunca ha enviado ninguna lectura a la base de datos anteriormente, el Servicio responde con diez ceros (0000000000). Seguido de la fecha y la hora, el Servicio envía un Retorno de Carro <CR> ó 0x0D en hexadecimal, para indicarle al Cliente que el envío de la fecha y la hora se ha completado. Si el número serial del Cliente no es reconocido por el servicio, el Servicio envía inmediatamente transmite un 1 seguido de un Retorno de Carro <CR> para indicarle al Cliente que hubo un error en el envío del número serial; seguido de esto el Cliente finaliza la llamada y el usuario tendrá que reiniciar el dispositivo electrónico para intentar nuevamente realizar una transacción de información exitosa.
- **Paso 5:** Si el Cliente recibió la fecha y hora proveniente del Servicio, el Cliente hará un conteo de las lecturas con fechas más recientes y enviará en formato de tres caracteres el número total de tramas que se transmitirán, seguido enviará dichas tramas (como se recomienda anteriormente, el usuario no debe dejar acumular más de 30 lecturas que no han sido enviadas anteriormente). Al finalizar el envío de las tramas el Cliente enviará un Retorno de Carro <CR> ó 0x0D en hexadecimal, para indicarle al Servicio que la transmisión de tramas se ha completado. Si hubo algún error en la recepción de la fecha y hora proveniente del Servicio, el Cliente finalizará la llamada y el usuario tendrá que reiniciar el dispositivo electrónico para intentar nuevamente realizar una transacción de información exitosa.
- **Paso 6:** Luego de recibir las tramas provenientes del Cliente, el Servicio inspecciona si todas las tramas están compuestas por un espacio seguido de 14 caracteres numéricos, y si el número de tramas recibidas concuerda con el de la cuenta de tramas especificado al inicio del envío de estas. Si la inspección es aprobada, entonces el Servicio se encarga de introducir la información de las tramas en los campos respectivos de la base de datos, y

si dicha operación finaliza satisfactoriamente, el Servicio transmite un 0 seguido de un Retorno de Carro <CR> ó 0x0D en hexadecimal para indicarle al cliente que la transacción de información se completó exitosamente. Si existe alguna corrupción en las tramas provenientes del Cliente, ó si el Servicio no pudo ingresar la información de las tramas en la base de datos exitosamente, entonces el Servicio envía un 1 seguido de un Retorno de Carro <CR> para indicarle al Cliente que hubo un error en la transacción de información, y el usuario tendrá que reiniciar el dispositivo electrónico para intentar nuevamente realizar una transacción de información exitosa.

Para ilustrar los pasos generales de una transacción de información exitosa, a continuación se plantea el escenario para la transmisión de una lectura nueva: Se sabe que el número serial del dispositivo electrónico que se comunicará con el Servicio es el 00000099. Dicho dispositivo electrónico extraerá información del glucómetro GlucoLab que contiene las lecturas que se muestran en la **Tabla 1.2** del capítulo anterior. Además se sabe que la fecha y hora de la última lectura registrada en la base de datos para el Cliente número 00000099 es el 23 de junio de 2006 a las 5:46 p.m.; entonces la transacción de información exitosa es la que se muestra a continuación:

1. <C>: Llama al número telefónico de la línea que está conectada al MODEM del ordenador del médico.
2. <S>: Toma la llamada entrante.
3. <C>: 00000099<CR>
4. <S>: 0606231746<CR>
5. <C>: 001 00606231752066<CR>
6. <S>: 0<CR>
7. <C>: Termina la llamada telefónica.

Para que la transacción de información explicada anteriormente sea exitosa, es necesario que el ordenador en la clínica del médico esté encendido y ejecutando el servicio que atiende la llamada.

Existe la posibilidad que la transacción sea iniciada por el ordenador del médico, en lugar del dispositivo electrónico; dicha modalidad tiene sus ventajas y desventajas, pero no será abordada ya que no está comprendida dentro de los límites fijados al inicio del Trabajo de Graduación.

### **2.3 Hardware del dispositivo electrónico**

El hardware del dispositivo electrónico es similar al de un computador, y comprende los siguientes componentes generales:

- El Núcleo, el cual posee una entrada para la fuente de corriente directa de 5 voltios y 1 amperio, para suministrar la energía a todo el sistema; también posee una entrada de espiga estéreo de 3.5 mm. para comunicarse con los glucómetros a través de un cable especial, según el tipo de glucómetro que se conecte; además aloja al director del sistema, el PIC18F4550.
- Los periféricos de entrada / salida para la comunicación con el usuario, como la pantalla LCD I2C<sup>9</sup> y los tres botones para la navegación en los menús que se muestran en el LCD.
- El MODEM, que se encarga de establecer la llamada telefónica con el ordenador remoto y de transmitir los datos que le indica el microcontrolador.

En la **Figura 2.3** se presenta el diagrama esquemático del núcleo del dispositivo electrónico, el cual está compuesto por los siguientes bloques de hardware interconectados:

- A. Bloque del microcontrolador, el cual está montado en un zócalo ZIF<sup>10</sup>, y la red del cristal que le proporciona la señal de reloj al microcontrolador.
- B. Bloque del MAX232 y sus capacitores de polarización.
- C. Bloque del Relay.
- D. Bloque del conector de las señales de datos para el MODEM.
- E. Bloque del conector que le proporciona la alimentación al MODEM.
- F. Bloque del conector hembra para la espiga estéreo de 3.5 mm.
- G. Bloque del conector hembra para la fuente de alimentación de 5 voltios y 1 amperio, el interruptor de encendido y apagado del sistema y el fusible de protección.
- H. Bloque de polarización de los botones que permiten la navegación en los menús que presenta el LCD y el botón interno de Reset.
- I. Bloque del conector de las señales de alimentación y datos para la pantalla LCD, y las resistencias de polarización según el protocolo de comunicación I2C.

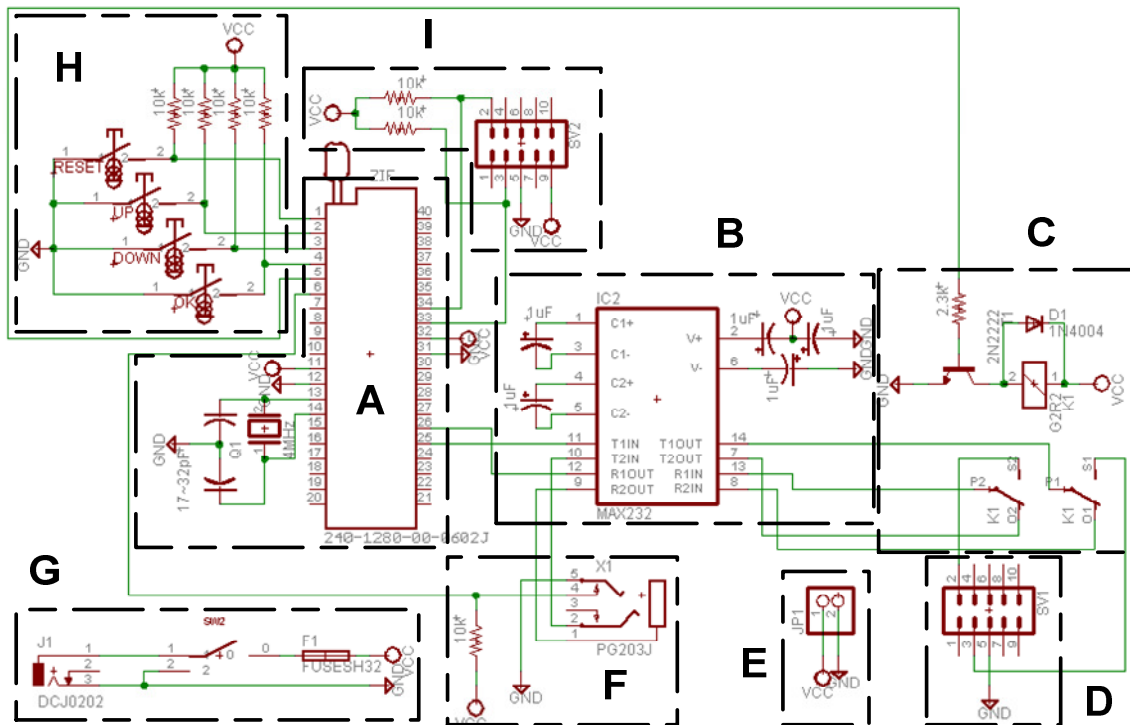
De los bloques generales antes mencionados podemos ampliar la siguiente información:

- **Bloque “A”:** contiene la red del cristal que le proporciona la señal de reloj al microcontrolador, y está compuesta por dos capacitores entre el rango de 16 a 32 pF y por un cristal de 4 MHz. En el dispositivo electrónico, el microcontrolador PIC18F4550 utiliza la frecuencia de 4 MHz para multiplicarla y trabajar a una frecuencia 8 veces mayor, es decir de 32 MHz.

<sup>9</sup> De sus siglas en inglés Inter - Integrated Circuit, es un bus de comunicaciones serie diseñado por Philips.

<sup>10</sup> De sus siglas en inglés Zero Insertion Force.



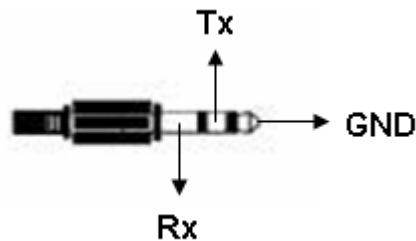


**Figura 2. 3 Esquemático del núcleo del dispositivo electrónico**

- **Bloque “B”:** contiene un circuito MAX232 y sus respectivas redes de capacitores de polarización de 1 uF. El bloque se encarga de convertir señales de voltaje de niveles TTL a señales de niveles RS232 y viceversa. Las hojas de datos y pinout de éste integrado se encuentran en el **Anexo B**.
- **Bloque “C”:** consiste en un relay de estado sólido, con su transistor de control bien polarizado y el diodo que lo protege. El relay es controlado por la señal proveniente del pin RA3 (pin 5) del microcontrolador y es activado con una señal alta. El propósito del bloque C es intercambiar la ruta de conexión entre el microcontrolador – glucómetro y microcontrolador – MODEM.
- **Bloque “D”:** es un cabezal de pines de 5x2, que mediante un cable con conectores hembra de 5x2 en las puntas, lleva las señales de transmisión y recepción del módulo USART<sup>11</sup> del microcontrolador en el bloque A, ya convertidas en niveles RS232 por el bloque B, al MODEM, dependiendo del estado del relay en el bloque C.

<sup>11</sup> De las siglas en inglés Universal Synchronous Asynchronous Receiver Transmitter.

- **Bloque “E”:** es un cabezal de pines de 1x2, que mediante un cable con conectores hembra de 1x2 en las puntas, lleva 5 voltios al MODEM para su alimentación.
- **Bloque “F”:** es el conector hembra de la espiga estéreo de 3.5 mm. donde se conectará el cable para comunicar un glucómetro específico al dispositivo electrónico. Dicho conector posee un contacto que permanece cerrado cuando la espiga no se ha insertado en el conector, y se abre cuando la espiga se inserta. Observando las conexiones en la **Figura 2.3** se aprecia que la línea que sale del bloque F es conectada al pin RA4 (pin 6) del microcontrolador, que funciona como entrada para detectar si el cable de comunicación con el glucómetro ha sido conectado (0 voltios), o no (5 voltios) al dispositivo electrónico. En la **Figura 2.4** se muestra la distribución de señales en la espiga estéreo de 3.5 mm. que se conecta al núcleo del dispositivo electrónico. Las señales de la espiga que se inserta en el conector hembra del bloque F, son conectadas a las señales de transmisión y recepción del módulo USART del microcontrolador del bloque A, a través del bloque B, dependiendo del estado del relay en el bloque C.



**Figura 2. 4 Señales en la espiga estéreo que se conecta al núcleo del dispositivo electrónico**

- **Bloque “G”:** está compuesto por el conector hembra para la fuente de voltaje que le proporcionará 5 voltios, 1 amperio al sistema. Además contiene el interruptor de encendido y apagado del dispositivo electrónico y el fusible de protección para cortocircuitos.
- **Bloque “H”:** contiene la polarización necesaria para los tres botones (Arriba, Abajo y Enter) de navegación en menús presentados en el LCD, y para el botón interno de Reset del sistema. El microcontrolador utiliza las señales activas en bajo provenientes de dichos pulsadores. El botón Arriba está conectado al pin RA0 (pin2) del microcontrolador, el botón abajo está conectado al pin RA1 (pin 3) y el botón Enter está conectado al pin RA2 (pin 4); así también el botón interno Reset está conectado al pin Master Clear (pin 0) del microcontrolador.

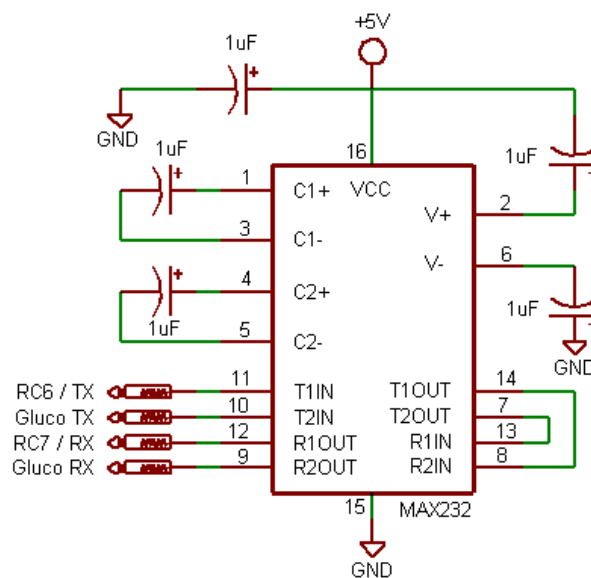
- **Bloque “I”**: contiene las resistencias de pull up para las líneas de datos SDA y de reloj SCL del protocolo I2C de comunicación con la pantalla LCD, además contiene el cabezal de pines de 5x2, que a través de un cable con una terminal hembra de 5x2 en una punta, lleva las señales de alimentación, de reloj y de datos hasta la pantalla LCD.

Algo que vale la pena destacar es que la interconexión de los **bloques A, B, D y F** depende del estado del **bloque C**, más específicamente del estado del pin RA3 del microcontrolador, que determina el comportamiento del relay en el bloque C.

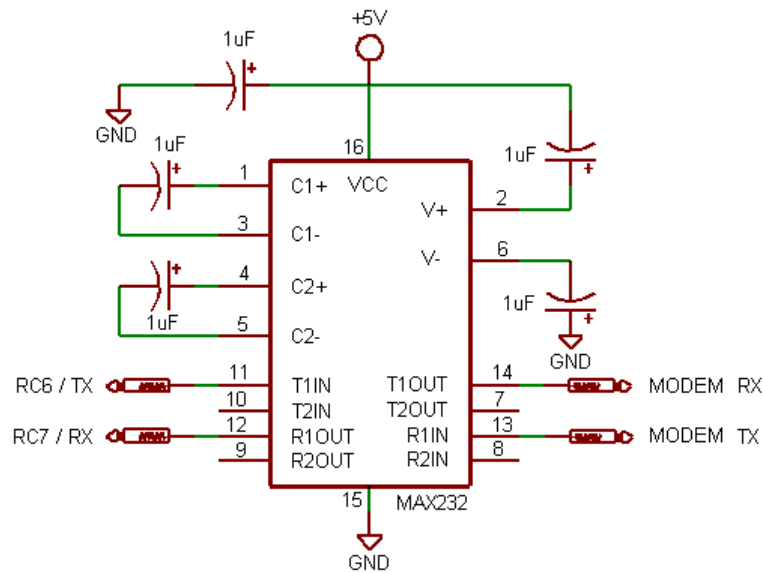
Las conexiones resultantes en el núcleo del dispositivo electrónico, cuando la señal en el pin de salida RA3 (pin 5) es baja (0 voltios) o alta (5 voltios), se muestran en la **Figura 2.5** y **Figura 2.6** respectivamente.

De las **Figuras 2.5** y **2.6** se observa que: los niveles de las señales de transmisión RC6 / TX y RC7 / RX del módulo USART del PIC18F4550 son de niveles TTL; que las señales de Transmisión y Recepción del puerto serial de los glucómetros son de niveles TTL también; y que las señales de Transmisión y Recepción utilizadas por el MODEM son de niveles RS232.

Puesto que el PIC18F4550 solo cuenta con un módulo USART el bloque C, a través de la señal de control en el pin RA3, se encarga de conmutar la dirección de las líneas del módulo USART entre el glucómetro conectado al núcleo y el MODEM que transmitirá la información al ordenador remoto.



**Figura 2. 5 Conexión resultante con salida baja en el pin RA3**



**Figura 2. 6 Conexión resultante con salida alta en el pin RA3**

Las pistas del circuito impreso del núcleo se desarrollaron utilizando el software de diseño asistido por computadora EAGLE, y se encuentran en el disco compacto adjunto al presente documento.

El MODEM utilizado en el dispositivo electrónico puede ser cualquier MODEM, como un MODEM externo de computadoras, capaz de comunicarse vía el estándar RS232 y de interpretar los comandos AT, que se explican más adelante en el apartado de Comandos AT en este capítulo. La velocidad de comunicación del microcontrolador en su puerto serial puede ser ajustada de acuerdo a las velocidades que el MODEM pueda manejar.

Para el dispositivo electrónico se seleccionó el MODEM MB-R 5V RS-232 de la compañía Radicom por su reducido tamaño, voltaje de alimentación estándar de 5V y porque el conector RJ11 para la línea telefónica ya está incluido en la placa.

La **Figura 2.7** muestra una fotografía del MODEM que se utilizó en el dispositivo electrónico, el MB-R 5V RS-232 de Radicom.

Entre las características del MODEM MB-R 5V RS-232 de Radicom destacan:

- Voltaje de operación de 5 voltios.
- Corriente de operación (bajo enlace telefónico establecido) de 101 mA.
- Temperaturas de operación de -40 °C hasta 85 °C.
- Humedad relativa de operación de 10% hasta 95%.
- Comunicación serial RS-232 por conector DB9 o por cabezal de 2x5 pines.

- Conector telefónico RJ11 en placa.
- Conjunto de instrucciones de Comandos AT.
- Velocidad de transmisión de datos de 2400 bps hasta 56 Kbps
- Soporta modalidad de bajo consumo de energía (56 mA. en Sleep Mode).
- Detecta si la línea telefónica está en uso.
- Identificador de llamadas.
- Detector de llamada en espera.
- Dimensiones: 2.165" por 2.5" en la base y 0.66" de altura.



**Figura 2. 7 Fotografía del MODEM MB-R 5V RS-232 de Radicom**

Para que el dispositivo electrónico fuera capaz de indicarle al usuario el estado de la transacción de información con el ordenador remoto, si ésta se completó satisfactoriamente o si ocurrió algún error de comunicación y es necesario volver a intentarlo; y para presentar los menús de configuración del sistema como selección del lenguaje y del glucómetro que se conectará; el dispositivo electrónico se diseñó para presentar mensajes de longitudes de 16 caracteres x 2 filas en alguna pantalla LCD que utilice el bus serial I2C de Philips para comunicarse con el microcontrolador. Dicha pantalla LCD puede ser cualquiera que esté disponible en el mercado, y que cumpla con las características antes mencionadas.

La pantalla LCD que se seleccionó para el dispositivo electrónico fue la LK162-12 del fabricante Matrix Orbital, y fue seleccionada por que, además de que cumple con las características antes mencionadas, por su disponibilidad en el mercado y por su voltaje de alimentación estándar de 5 voltios. En la **Figura 2.8** se muestran las imágenes de la cara frontal y trasera del display.

Entre las características del display LCD LK162-12 de Matrix Orbital destacan:

- Voltaje de operación de 5 voltios.
- Posee luz de fondo verde y presenta caracteres en color negro.

- Corriente de operación de 10 mA. sin luz de fondo y 104 mA. con luz de fondo.
- Temperaturas de operación de 0 °C hasta 50 °C.
- Humedad máxima de operación del 90%.
- Soporta los protocolos de comunicación RS-232, RS-232 a niveles TTL e I2C.
- Para los protocolos RS-232 soporta velocidades de comunicación de: 1200, 2400, 9600 y 19200 baudios.
- Para el Protocolo I2C utiliza la frecuencia de reloj de 100KHz.
- Es capaz de presentar hasta 32 caracteres (16 por cada fila).
- El área reservada para presentar cada caracter es de 5.55 mm por 2.95 mm.
- El área disponible para presentar un mensaje es de 64.5 mm por 15.5 mm.
- Las dimensiones del dispositivo son: 80 mm por 36 mm en la base y 27.5 mm de altura.



**Figura 2. 8 Imágenes de la cara frontal y trasera del display LK162-12 de Matrix Orbital**

Como dispositivos periféricos de entrada, el sistema cuenta con varios pulsadores. El dispositivo electrónico cuenta con un pulsador interno, colocado en el núcleo, para resetear el sistema, dicho botón se utiliza en las pruebas de depuración del programa cargado en el microcontrolador, realizadas por los diseñadores.

Además el dispositivo electrónico cuenta con un circuito impreso, donde son colocados 3 pulsadores (Arriba, Abajo y Enter) para la navegación en los menús de configuración presentados en el display. Dichos pulsadores sobresalen en la cara superior del dispositivo electrónico, como se pudo observar en la **Figura 2.2**, junto con un LED que está colocado en la misma placa; el LED se utiliza como indicador de que el sistema se encuentra energizado. Los botones montados en dicha placa están soldados con cables a los puntos en el **bloque H** que se encuentra en el núcleo, y el LED está soldado a puntos en el **bloque G**.

El circuito esquemático y las pistas del circuito impreso del núcleo del dispositivo electrónico, y de la pequeña tabla donde están montados los pulsadores de navegación y el LED, se encuentran en el disco compacto adjunto al presente documento. El **Anexo C** contiene el inventario de partes que se utilizaron para la construcción del dispositivo electrónico, no hay que perder de vista que este

dispositivo es un prototipo, y por ello el costo total de su construcción es mucho más elevado al costo del mismo dispositivo fabricado en masa.

## 2.4 Generalidades sobre MODEMS y los Comandos AT

Los MODEMS (Moduladores Demoduladores) son utilizados para originar o contestar llamadas telefónicas, con el propósito de intercambiar información entre dos lugares remotos.

El control sobre el funcionamiento de un MODEM se consigue utilizando los comandos AT. Dichos comandos son instrucciones directas para el MODEM como marcar un número telefónico, tomar una llamada, etc. Y son utilizados generalmente durante la ejecución de los programas comerciales de comunicación sin que el usuario se percate de ello.

El MODEM responderá aceptando y procesando el comando AT suministrado, en la mayoría de velocidades de comunicación serial estándar y configuraciones de bits de paridad. Por cada comando AT suministrado, el MODEM responderá con un código resultado informando sobre el estado del MODEM. El formato básico de un comando AT se muestra a continuación:

```
AT<Comando><CR>  
OK
```

Del formato anterior se amplía lo siguiente:

AT: Indica al MODEM que ponga Atención al comando que sigue.

<Comando>: Cualquier comando AT válido.

<CR>: Retorno de Carro ó 0x0D en hexadecimal.

OK: Código resultado que indica que el MODEM ha ejecutado el comando

Ejemplos sencillos de comandos AT son los siguientes:

```
ATDT22269048<CR>
```

El comando anterior hace que el MODEM realice la acción equivalente a levantar el auricular y marcar el teléfono 22269048 e intentar conectarse con el MODEM remoto del otro lado de la línea.

```
ATS0=2<CR>
```

El comando anterior configura al MODEM para contestar automáticamente una llamada entrante después de detectar dos RINGS.

Existen gran cantidad de comandos AT dependiendo del fabricante del MODEM que se esté utilizando. A continuación se presenta un listado de los comandos AT más generales y utilizados en aplicaciones básicas. La sintaxis general es **ATXn<CR>**, donde X es el comando AT y n es el parámetro del comando X, usualmente 0 ó 1.

- **A/**: Repite el comando anterior.
- **A**: Contesta la llamada entrante.
- **D**: Comando de Marcado. Los parámetros básicos se presentan a continuación:
  - 0-9**: Marca los dígitos del 0 al 9.
  - P**: Selecciona marcado por pulso, afecta a la llamada en curso y posteriores.
  - T**: Selecciona marcado por tono, afecta a la llamada en curso y posteriores.
- **En**: Eco del comando. Los parámetros son los siguientes:
  - 0**: Deshabilita el eco en los caracteres que componen a los comandos AT.
  - 1**: Habilita el eco en los caracteres que componen a los comandos AT.
- **H0**: Termina la llamada en curso.
- **Sr**: Escribe o lee el Registro S número “r”. Los parámetros básicos se detallan a continuación:
  - r**: Establece el Registro S número “r” como registro por defecto.
  - r=n**: Asigna el valor “n” al Registro S número “r”.
  - r?**: Reporta el valor asignado al Registro S número “r”.

Los valores que utilizan algunos de los comandos AT se almacenan en los “Registros S”. Existe gran cantidad de Registros S utilizados para variados propósitos pero los Registros S más básicos se presentan a continuación en la **Tabla 2.3**.

- **Vn**: Modifica el formato en que el MODEM devuelve los códigos resultado. Los parámetros se presentan a continuación:
  - 0**: Los códigos resultado se devuelven de forma numérica.
  - 1**: Los códigos resultado se devuelven en forma de cadena ASCII.

Existen gran cantidad de códigos resultado para los diferentes comandos AT. En la **Tabla 2.4** se presenta un listado de los códigos resultado básicos en forma numérica y su equivalente en forma de cadena ASCII.



**Tabla 2. 3 Registros S básicos**

Registro	Valores	Unidades	Por defecto	Descripción
S0	0-255	Rings	0	Determina el número de RINGS que el MODEM debe detectar antes de contestar automáticamente una llamada. Por ejemplo: ATSO=0<CR> le indica al MODEM que no debe contestar automáticamente la llamada, y ATSO=1<CR> le indica al MODEM que debe contestar la llamada automáticamente después de detentar un RING.
S1	0-255	ASCII	0	Cuenta los RINGS detectados.
S2	0-127	ASCII	43	Caracter ASCII utilizado en la secuencia de escape.
S3	0-127	ASCII	13	Caracter ASCII que finaliza los comandos AT.
S4	0-127	ASCII	10	Caracter ASCII para Avance de Línea <LF>.
S5	0-127	ASCII	8	Caracter ASCII para regresar un espacio atrás <BS>.
S7	1-255	Segundos	50	Tiempo que espera el MODEM para que el dispositivo remoto conteste la llamada.
S11	50-255	.01 seg.	85	Duración de un tono de marcado.

**Tabla 2. 4 Códigos resultado básicos**

Número	Cadena ASCII	Descripción
0	OK	El MODEM ha ejecutado el comando.
1	CONNECT	Se ha establecido conexión a 300 bps.
2	RING	El RING de una llamada entrante ha sido detectado.
3	NO CARRIER	Este código refleja un intento de desconexión o una falla al completar una conexión.
4	ERROR	El comando AT enviado al MODEM es inválido.
5	CONNECT 1200	Se ha establecido conexión a 1200 bps.
6	NO DIALTONE	El MODEM no puede detectar el tono de la línea telefónica.
7	BUSY	Se ha detectado el tono de ocupado
8	NO ANSWER	Después que transcurrió el tiempo especificado en el Registro S7, el dispositivo remoto no contestó la llamada.
10	CONNECT 2400	Se ha establecido conexión a 2400 bps.
13	CONNECT 9600	Se ha establecido conexión a 9600 bps.

- **Z0:** Resetea el MODEM a sus configuraciones de fábrica.
- **&Kn:** Control de flujo de la comunicación. Los parámetros básicos se detallan a continuación:
  - 0:** Deshabilita el control de flujo.
  - 3:** Habilita el control de flujo por líneas RTS/CTS según el estándar RS232.
  - 4:** Habilita el control de flujo por XON/XOFF.

## 5: Soporta el control de flujo transparente XON/XOFF

Para mayor información sobre el comando AT&Kn y otros comandos AT, visitar la URL: <http://www.usr.com/support/3cp3056/3cp3056-spanish-ug/3056-6.htm>

Para **desconectar una llamada en curso** existen varios métodos que se detallan a continuación:

1. Quitándole por un instante la alimentación al MODEM, hará que éste termine la llamada y vuelva a su estado inactivo.
2. El MODEM remoto puede ser el que termine la llamada.
3. Por defecto, la transición del nivel alto al nivel bajo en la línea DTR del estándar RS232, también hará que el MODEM termine la llamada en curso.
4. Los comandos ATH0<CR> ó ATZ0<CR> hacen que el MODEM desconecte la llamada en curso. Cuando el MODEM ha establecido conexión con un dispositivo remoto, se dice que el MODEM se encuentra en estado **En Línea**. Para que el MODEM pueda ejecutar comandos en medio de una llamada telefónica, y no los confunda con datos que deben ser transmitidos, es necesario que el MODEM se encuentre en el llamado estado **En Línea de Comandos**. Para lograr que el MODEM cambie del estado En Línea, al estado En Línea de Comandos, es necesario introducir una secuencia de caracteres ASCII denominada **Secuencia de Escape**, la cual consiste en tres caracteres de escape especificados por el ASCII definido en el Registro S2; por defecto el valor almacenado en el Registro S2 indica que el caracter de escape es el ASCII "+". La forma, por defecto, de introducir la secuencia de escape es: primero dejar de transmitir información con el MODEM y esperar un segundo, luego enviar los ASCII "+++", un segundo después de enviado los caracteres de escape el MODEM contestará con el código resultado **OK**; esto indica que el MODEM ya cambió al estado En Línea de Comandos. Finalmente se envía el comando **ATH0<CR>** para terminar la llamada en curso.

## 2.5 Software del dispositivo electrónico

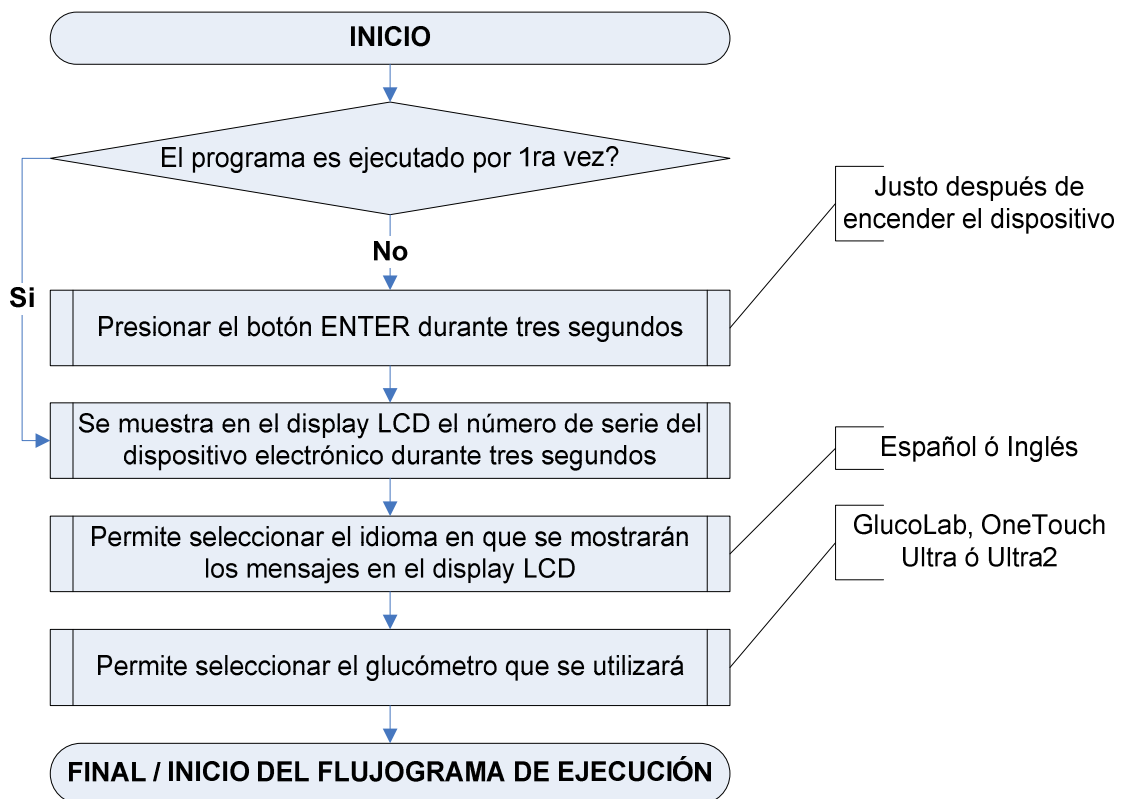
Existen algunas variables que pueden ser configuradas y afectan el funcionamiento del dispositivo electrónico. Así el software que gobierna dicho dispositivo posee dos grandes partes generales: la Configuración del dispositivo y la Ejecución del programa.

Las variables configurables del software del dispositivo son: el idioma en que se presentan los mensajes en el display LCD, teniendo dos opciones: español e

inglés; y el glucómetro a conectar al dispositivo, teniendo actualmente dos opciones funcionales: el GlucoLab y el OneTouch Ultra2.

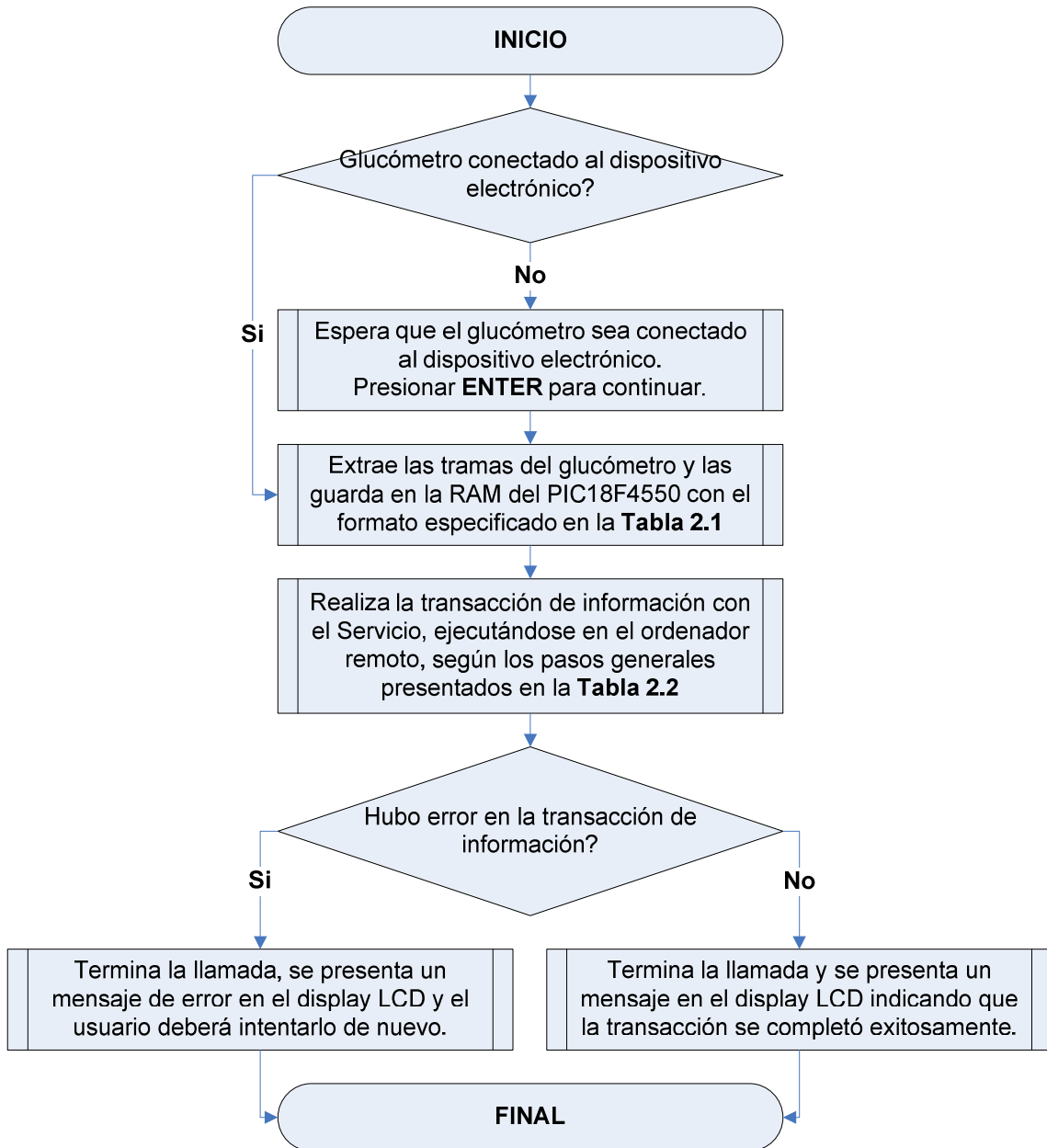
Para realizar la configuración de dichas variables se utilizan los botones de navegación en menús ubicados en la cara superior del dispositivo, los botones Arriba y Abajo son usados para visualizar las diferentes opciones dentro de un menú dado, y el botón Enter se utiliza para seleccionar una opción.

La forma para entrar al Modo Configuración y el proceso de configuración se muestran con el flujograma de la **Figura 2.9**, donde se puede observar que el programa entra en Modo Configuración automáticamente, si es la primera vez que éste se está ejecutando, o también si justo después de encender el dispositivo electrónico se mantiene presionado el botón Enter, aproximadamente durante tres segundos. Una vez que se ha entrado en Modo Configuración, el display LCD muestra el número serial específico para el dispositivo durante tres segundos y luego le permite al usuario configurar las variables del programa. Luego de configuradas las variables del programa se inicia la ejecución del mismo.



**Figura 2. 9** Flujograma de configuración del software del dispositivo electrónico

Si no es la primera vez que se ejecuta el programa, y el botón Enter no es presionado durante tres segundos justo después de encender el dispositivo, el programa esquivará el Modo Configuración e iniciará la ejecución general del software del dispositivo electrónico, tal como lo muestra el flujograma de la **Figura 2.10**.



**Figura 2. 10** Flujograma de ejecución general del software del dispositivo electrónico

De la **Figura 2.10** se aprecia que el software es capaz de detectar si el cable de comunicación dispositivo – glucómetro no ha sido conectado aún, ó no ha hecho un buen contacto. En ese caso el display muestra un mensaje, pidiendo que se conecte el glucómetro al dispositivo electrónico; una vez que el dispositivo detecta que el cable de comunicación está bien conectado, el display presenta otro mensaje pidiendo que se presione la tecla Enter para continuar con la ejecución del programa.

Como ya se ha mencionado anteriormente, el display LCD, además de presentar los menús de configuración de las variables del sistema, le comunica al usuario del dispositivo sobre los estados de ejecución del programa, y si ha ocurrido algún error o no en la comunicación con el ordenador remoto.

En la **Tabla 2.5** se presenta el listado completo de los mensajes que puede presentar el display LCD, en sus dos diferentes idiomas y en las situaciones en que dichos mensajes son desplegados.

**Tabla 2. 5 Lista de mensajes que pueden presentarse en el display LCD**

<b>Español</b>	<b>Inglés</b>	<b>Situación</b>
Español	-	Opción de selección de idioma
-	English	Opción de selección de idioma
GlucoLab	-	Opción de selección de glucómetro
OneTouch	-	Opción de selección de glucómetro
Accu Check Compact +	-	Opción de selección de glucómetro. Este glucómetro funcionará con el sistema eventualmente.
Accu Check Active	-	Opción de selección de glucómetro. Este glucómetro funcionará con el sistema eventualmente.
Por favor conecte el glucómetro	Please connect the glucometer	Mostrado antes de interrogar el glucómetro, si se detecta que el glucómetro no está conectado.
Presione OK para continuar	Press OK to continue	Mostrado a continuación del mensaje anterior, luego que se detectó que el glucómetro ha sido conectado.
Leyendo glucómetro	Reading glucometer	Durante la extracción de las lecturas del glucómetro.
Marcando...	Dialing...	Durante el marcado del número telefónico.
No hay tono / Ocupado	No dial tone / Busy	Si no hay tono de marcado en la línea ó si el ordenador remoto no atiende la llamada telefónica.
Conectado al servidor	Connected to server	Cuando se establece la conexión con el ordenador remoto.
Transmisión completada!	Transmission accomplished!	Cuando el intercambio de información entre el dispositivo y el Servicio se completó sin errores.
Error de transmisión	Transmission error	Si ocurrió algún error durante la comunicación entre el dispositivo y el Servicio en el ordenador remoto. Significa que la información no fue introducida en la base de datos y que el usuario deberá intentar reiniciar el dispositivo electrónico e intentarlo de nuevo.

Los archivos del código fuente del programa cargado en el microcontrolador PIC18F4550 se encuentran en el disco compacto adjunto al presente documento.

## 2.6 CONCLUSIONES DEL CAP. II

Las características del microcontrolador PIC18F4550 de Microchip tales como el tamaño de su memoria de programa y datos, el multiplicador interno de la frecuencia de reloj, el direccionamiento indirecto por saltos relativos, y su conjunto de poderosas instrucciones, como la multiplicación de bytes en un ciclo de instrucción, fueron factores determinantes en la elección de dicho microcontrolador y simplificaron en gran medida la tarea que desempeña el software del dispositivo electrónico. Dichas funcionalidades del PIC18F4550 sólo están disponibles en los microcontroladores de la gama mejorada dentro de los productos de Microchip.

El dispositivo electrónico funciona para un solo usuario.

Las tramas de información extraídas de los glucómetros son sustancialmente diferentes entre distintos fabricantes de glucómetros. El dispositivo electrónico se encarga de normalizar dichas tramas a un formato más corto, simple y funcional.

El control del MODEM, que establece la comunicación con el ordenador remoto, se logra utilizando comandos AT básicos, de esta forma el MODEM utilizado en el dispositivo, puede ser cualquier MODEM que funcione bajo el estándar RS232 y que interprete comandos AT. Los comandos AT son transmitidos al MODEM por el módulo de comunicación serial USART del microcontrolador a una velocidad de 9600 baudios, 8 bit de datos, 1 bit de paro y ninguna paridad.

El dispositivo electrónico podría utilizar cualquier display LCD que sea capaz de presentar mensajes de 16 caracteres x 2 filas, y que se comunique utilizando el bus serial I2C de Philips configurado para una velocidad de 100 Khz.

El dispositivo electrónico es capaz de recordar su número de serie, número telefónico a marcar y la última configuración de sus variables lenguaje y glucómetro que se le conecta, aún cuando el dispositivo es desenergizado. Esto se logra gracias a que dichas variables son almacenadas en la memoria no volátil EEPROM y memoria de programa que posee el PIC18F4550.

## 2.7 REFERENCIAS BIBLIOGRÁFICAS

3Com Corporation. Manual del Usuario del Módem (en línea). EEUU. (Citada el 30 de Noviembre de 2006).

<http://www.usr.com/support/3cp3056/3cp3056-spanish-ug/3056-6.htm>

Matrix Orbital. LK162-12 Technical Manual Revision: 3.0. Canadá. 2006.

Maxim Integrated Products. +5V-Powered, Multichannel RS-232 Drivers / Receivers Data Sheet. EEUU. 2000.

Microchip Technology Inc. PIC18F2455 / 2550 / 4455 / 4550 Data Sheet. EEUU. 2006.

Radicom Research, Inc. Designers Guide for the MB-R-5V RS232 Modem Module Family. EEUU. Enero 2006.



## **CAPÍTULO III**

# **DESARROLLO DEL SERVICIO DE ATENCIÓN DE LLAMADAS E INTERFACE WEB PARA ADMINISTRACIÓN Y CONSULTA DE DATOS DEL MÉDICO Y EL USUARIO**

### **Introducción**

En este capítulo, se trata el desarrollo del servicio remoto gestor de la comunicación y monitoreo a distancia. Se presentará inicialmente una visión bibliográfica de los métodos y herramientas que se han utilizado para hacer la persistencia y la presentación de la información o registro médico de cada paciente. Comenzando con un panorama bastante amplio del Lenguaje de Programación de Java Script, las ventajas que éste código de programación ofrece contra métodos tradicionales, la presentación de todos los procesos de gestión de comunicación; los cuales fueron desarrollados en base a Servlets y páginas HTML. En la parte final del capítulo se trata el tema del gestión de información a través de una base de datos manejada con el servidor de base de datos MySQL; se presentan los comandos más utilizados y básicos para el gestión de una base de datos, así mismo se presentan ejemplos del uso de éstos junto a su respectiva sintaxis de petición. Se presentan además las tablas que fueron creadas en la base de datos utilizada para el almacenamiento de la información, una explicación de las mismas y las funcionalidades que estas tienen; junto a un vistazo general para entender a nivel macro el funcionamiento de las distintas aplicaciones, las cuales unidas funcionan como un servicio completo de atención de llamadas telefónicas para recibir información médica, el almacenamiento seguro y confiable de dicha información y su respectiva presentación Web. Todo el código desarrollado, al cual se hace mención en este capítulo, se encuentra en el disco compacto adjunto al presente trabajo de graduación.

### 3.1 Lenguaje Java

Java es un lenguaje de programación que surgió en 1991 cuando un grupo de ingenieros de Sun Microsystems trataron de diseñar un nuevo lenguaje destinado a electrodomésticos.

Debido a la existencia de distintos tipos de CPUs y a los continuos cambios, era importante conseguir una herramienta independiente del tipo de CPU utilizada.

Se desarrolló un código “neutro” que no dependía del tipo de electrodoméstico, el cual se ejecutaba sobre una “máquina hipotética o virtual” denominada Java Virtual Machine (JVM). Como lenguaje de programación para computadores, Java se introdujo a finales de 1995. La clave fue la incorporación de un intérprete Java en la versión 2.0 del programa Netscape Navigator, produciendo una verdadera revolución en Internet.

Al programar en Java no se parte de cero. Cualquier aplicación que se desarrolle “cuelga” (o se apoya, según como se quiera ver) en un gran número de clases preexistentes. Algunas de ellas las ha podido hacer el propio usuario, otras pueden ser comerciales, pero siempre hay un número muy importante de clases que forman parte del propio lenguaje (el API o Application Programming Interface de Java).

Java incorpora en el propio lenguaje muchos aspectos que en cualquier otro lenguaje son extensiones propiedad de empresas de software o fabricantes de ordenadores (threads, ejecución remota, componentes, seguridad, acceso a bases de datos, etc.).

La compañía Sun describe el lenguaje Java como “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico”.

Java está siendo continuamente extendido para proporcionarle nuevas características y librerías que resuelven elegantemente problemas que son muy difíciles en la programación tradicional como el acceso a bases de datos, el uso de procesos multihilo, la programación de redes y la programación distribuida<sup>12</sup>.

El principal objetivo del lenguaje Java es llegar a ser el “nexo universal” que conecte a los usuarios con la información; no importando si ésta se encuentra en

---

<sup>12</sup> Programación Distribuida, es una filosofía de programación enfocada en el desarrollo de sistemas distribuidos; se caracteriza por el uso colectivo y la explotación de los recursos que brindan las redes (o los sistemas distribuidos mismos) para la realización de tareas distribuidas.

el ordenador local, en un servidor de Web, en una base de datos o en cualquier otro lugar.

### **3.1.1. El compilador de Java**

Básicamente se trata de una de las herramientas de desarrollo incluidas en el JDK (Java Development Kit), que es un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en Java. Este kit realiza un análisis de sintaxis del código escrito en los ficheros fuente de Java (con extensión \*.java). Si no encuentra errores en el código genera los ficheros compilados (con extensión \*.class) también denominados “bytecodes”. En otro caso muestra la línea o líneas erróneas. En el JDK de Sun dicho compilador se llama javac.exe.

Para llevar a cabo la compilación de un archivo fuente java basta con digitar desde la consola de comandos del sistema operativo la siguiente instrucción: “javac Nombre\_Archivo.java”; y si el código esta libre de errores se generara automáticamente el bytecode “Nombre\_Archivo.class”

### **3.1.2. La Java Virtual Machine**

Una vez compilado el programa fuente no debería ser necesaria ninguna modificación por el hecho de cambiar de procesador o de ejecutarlo en otra máquina. La clave consistió en desarrollar un código “neutro” el cual estuviera preparado para ser ejecutado sobre una “máquina hipotética o virtual”, denominada Java Virtual Machine (JVM). Es esta JVM quien interpreta este código neutro convirtiéndolo a código particular de la CPU utilizada. La JVM es el intérprete de Java. Ejecuta los “bytecodes” (ficheros compilados con extensión \*.class) creados por el compilador de Java (javac.exe). Utilizando el archivo “.class” generado con javac.exe; solo digitamos desde la consola: “java Nombre\_Archivo”; y el programa debería de ejecutarse.

### **3.1.3. Programación Orientada a Objetos**

Ya que Java es un lenguaje Orientado a Objetos, a continuación se presentan algunos conceptos importantes de este tipo de programación.

- Objeto: Es una encapsulación<sup>13</sup> genérica de datos y procedimientos para manipular esos datos. Al igual que los objetos del mundo real, los objetos de

---

<sup>13</sup> La encapsulación, consiste en ocultar la información de una clase u objeto que no es pertinente para un determinado método que utilice dicha clase para realizar una tarea.

software tienen un estado y un comportamiento. El estado de los objetos se determina a partir de los valores de una o más variables y el comportamiento de éstas con la implementación de métodos.

- Clases: El elemento básico de la programación orientada a objetos en Java es la clase. Una clase define la forma y comportamiento de un objeto. Todo en java son clases y para desarrollar una aplicación siempre es necesario derivar de una clase específica ya sea explícitamente a partir de una clase en particular o por defecto derivar de la clase `Object`.

El código de un programa ejecutable es simplemente una clase que contiene el método `main ()`, el cual utiliza otras clases ya sea de las librerías propias de java (`packages`) o creadas por el usuario, para desarrollar las acciones necesarias sobre las variables y datos de entrada.

- Interfaces: Son tipos especiales de clases, en la cual sólo se declaran los métodos pero no se definen, ya que estos se definirán en las clases “hijas” que implemente dicha interface.
- Packages: También llamadas librerías o bibliotecas estándar, son agrupaciones de clases que tienen un fin similar; la agrupación en `packages` también permite delimitar el acceso a clases, variables y métodos de las clases contenidas en el `package`.

Para definir una clase como perteneciente a un `package` es necesario que en la primera línea del archivo fuente donde se define la clase, se declare la sentencia `“package nombre_del_package;”` y cualquier clase definida en ese archivo fuente, pertenecerá al `package` indicado.

Para poder utilizar las clases contenidas en un `package`, hay que importar la o las clases específicas (pueden ser todas), cuyos métodos o atributos se desea o se necesita utilizar en el programa; esto puede hacerse de las siguientes maneras:

```
“import nombre_del_package.Nombre_Clase”
```

Para importar la clase `Nombre_Clase`

```
“import nombre_del_package.*”
```

Para importar todas las clases del `package`.

- Herencia: Una clase puede derivar de otra clase, en tal caso la clase “hija” hereda los atributos (variables y métodos) de la clase “madre”; los cuales, pueden ser redefinidos en la nueva clase y además esta clase puede agregar nuevos métodos, extendiendo de esta forma las capacidades de la clase “madre”.

### 3.1.4. Variables en Java

Las variables son localidades de memoria en las que pueden almacenarse datos. Cada una tiene un nombre, un tipo de dato y un valor. En Java se distinguen tres tipos de variables:

- Variables de Instancia: Se utilizan para definir los atributos de un objeto.
- Variables de clase: Son variables de instancia, con la particularidad de que tienen un único valor para todos los objetos creados de la misma clase.
- Variables Locales: Son las que se declaran y utilizan dentro de las definiciones de los métodos.

El tipo de dato de una variable, determina los valores que esta puede almacenar y las operaciones que pueden realizarse sobre ella. Los tipos pueden clasificarse de la siguiente manera:

- Tipos primitivos: Contiene un solo valor; ejemplos de estos pueden ser byte, short, int, long, flota, double, char y boolean.
- Tipos referencia: Son los definidos por el usuario dependiendo de los requerimientos de almacenamiento y de las capacidades deseadas para una determinada variable. Ejemplos: arreglos, clases e interfaces.

### 3.1.5. Eventos

En las aplicaciones en modo consola, el método main determina el orden en que se ejecutan las operaciones del programa. En aplicaciones con interfaces gráficas de usuario o que requieran del uso de cualquier dispositivo periférico para la entrada de datos (puertos Entrada/Salida); el orden en que se ejecutan las operaciones dependerá del usuario o de la aplicación que accede al periférico. Esto requiere de un nuevo tipo de programación no secuencial que permita a la aplicación responder adecuadamente a los requerimientos de entrada.

El modelo de eventos diseñado por Sun proporciona la base para el desarrollo de soluciones que requieran este tipo de funcionalidades. El modelo de eventos hace uso de fuentes de eventos (Source) y receptores de eventos (Listener). Una fuente de eventos es un objeto que tiene la capacidad de detectar eventos y notificar a los receptores de eventos que se han producido esos eventos. Un objeto receptor de eventos es una clase (o una subclase de una clase) que implementa un interfaz receptor específico. Para cada tipo de evento que sea

necesario manejar, se implementa el código de las acciones a tomar ante ese evento. La JVM se encarga de manejar los eventos notificados por los event sources y llamar al receptor de eventos adecuado; Aunque el programador puede establecer el entorno en que se producen esas notificaciones, siempre hay un escenario por defecto.

La forma de garantizar que los event listeners disponen de los métodos apropiados para gestionar los eventos es obligarles a implementar una determinada interface Listener. Las interfaces Listener se corresponden con los tipos de eventos que se pueden producir. De este modo, solo hemos de preocuparnos de registrar los receptores de eventos apropiados al objeto fuente de eventos, y de programar en dichos receptores de eventos que acciones ha de realizar nuestra aplicación, ante ese determinado tipo de evento.

El modelo de eventos es ampliamente utilizado para la programación de interfaces gráficas de usuario, las cuales requieren la interacción del usuario con el programa; además para el manejo óptimo de entradas y salidas de datos a través de periféricos del PC, por ejemplo manejo de eventos de Entrada/Salida de datos a puertos; si se cuenta o se dispone de las librerías adecuadas, recepción de señales de control a periféricos.

### **3.1.6. Ejecución multihilo ó Threads**

Los threads o hilos de ejecución, también llamados “procesos de peso ligero”, son segmentos de código de un programa que se ejecuta secuencialmente de modo independiente de las otras partes del programa. Esto podría interpretarse más fácilmente como: “Programas” que se ejecutan independientemente, y que pudiesen o no formar parte de una misma aplicación.

Los procesadores y sistemas operativos actuales permiten la multitarea, es decir, la funcionalidad aparente de poder realizar varias tareas al mismo tiempo.

Esto es algo como ya se mencionó, “aparente”; ya que no es posible para un microprocesador ejecutar dos tareas al mismo tiempo; lo que en realidad sucede es que el procesador distribuye el tiempo de atención de cada thread o hilo de ejecución, y realiza las tareas tan rápido que da la impresión que diferentes threads se ejecutan al mismo tiempo. Un programa de flujo único o monohilvanado (single-thread) utiliza un único flujo de control (thread) para controlar su ejecución. Muchos programas no necesitan la potencia o utilidad de múltiples flujos de control. La principal utilización de threads en un programa, es para el aislamiento y sincronización de tareas que requieran tiempos de espera para Entrada/Salida de datos como es el caso de las interfaces de entrada de datos de usuario.

### 3.1.6.1. Estados de un Thread

Un thread puede presentar cuatro estados distintos:

- Nuevo (New): El thread ha sido creado, pero no se ha ejecutado todavía el método `start ()`, por lo que todavía no se ha lanzado.
- Ejecutable (Runnable): El thread se está ejecutando, pero esto no quiere decir que se esté ejecutando en todo momento, pues ese thread tiene que repartirse el tiempo de CPU con los demás threads que sean también Runnable.
- Muerto (Dead): Cuando el thread ya acabó con su función, es decir el método `run ()` ha finalizado, o cuando el programa principal termina, u otro caso, como un error que provoque el paro total del thread.
- Bloqueado (Blocked o Not Runnable): El thread podría estar ejecutándose, pero hay algo que se lo impide, como por ejemplo una operación de E/S. Mientras un thread esté en este estado, no se le asigna tiempo de CPU. También cuando se llama a los métodos `sleep (long)` y `wait ([long])`.

## 3.2 Acceso a Puertos

Una de las formas más populares de conexión que se realiza en un PC, es a través de la puerta serie, que permite al ordenador comunicarse con todo tipo de dispositivos periféricos: módems, impresoras, scanner, lectores de código de barras, etc.

El API de Comunicaciones Java, constituido por el package `javax.comm` que proporciona JavaSoft, no forma parte del JDK, pero añade soporte a Java para dispositivos serie y paralelo al estilo Java, es decir, utilizando una semántica semejante a la que se usa con streams y eventos.

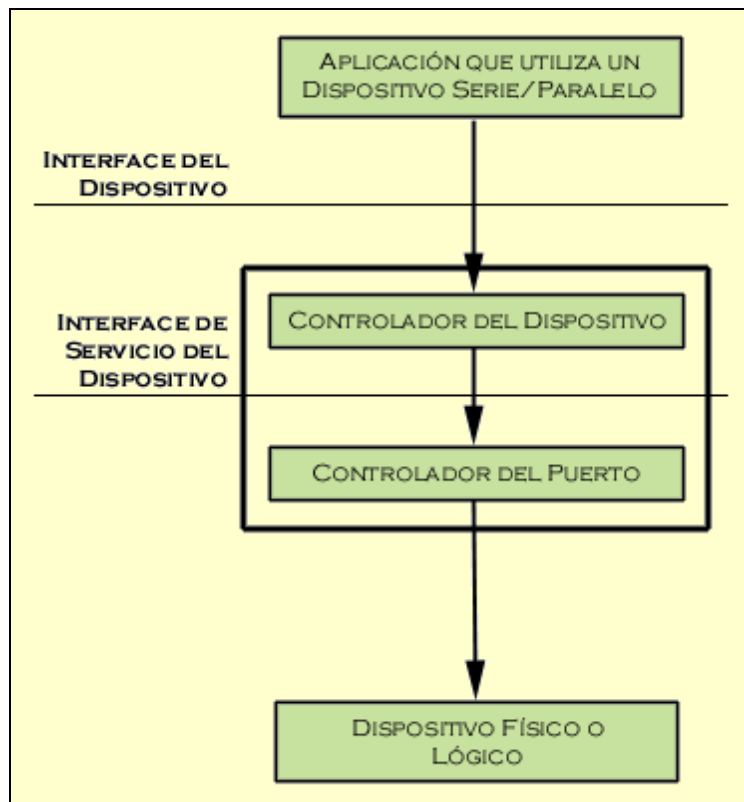
Para comunicarse con un dispositivo serie a través de uno de los puertos serie de un ordenador, desde una aplicación Java o un applet, es necesario un interfaz.

El API de Comunicaciones Java, permite transmitir y recibir datos a través de dispositivos conectados al puerto serie; proporcionando además un conjunto de opciones que permiten la configuración de todos los parámetros asociados a los puertos serie y paralelo.

Este API es una proposición para establecer un método estándar de acceso a los puertos de comunicaciones, que permita a los autores de software de

comunicaciones escribir programas Java independientes de plataforma. Así se pueden escribir programas para emulación de terminales, programas de fax, lectores de tarjetas, etc.

El diagrama de alto nivel de las capas que componen el API de comunicaciones Java es mostrado en la **Figura 3.1**.



**Figura 3. 1 Diagrama de capas del API javax.comm**

A diferencia de los drivers clásicos, que suelen venir con su propio modelo de eventos para control de las comunicaciones asíncronas, el API de Comunicaciones Java proporciona un interfaz de control de eventos basado en el modelo de eventos utilizado por el lenguaje Java, paquete `java.awt.event`.

Si se quiere saber si hay algún dato en el buffer de entrada, se puede realizar "polling"<sup>14</sup> o quedarse a la escucha. Si se usa "polling", el procesador comprueba periódicamente el buffer para ver si hay algún dato nuevo en él; en el otro método, el procesador se queda a la espera de que se produzca un evento derivado de la entrada de nuevos datos en el buffer, ya que tan pronto como

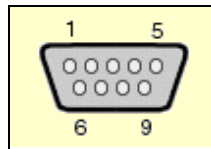
<sup>14</sup> Polling, es la técnica de control de eventos consistente en la programación de un gran bucle que se bloquee continuamente o periódicamente para comprobar si el usuario está haciendo algo interesante (pulsar una tecla, mover el ratón, etc.)



llega un nuevo dato al buffer, este envía una notificación o evento al procesador. Es este último mecanismo más acorde al manejo de aplicaciones Java el que se ha utilizado en el presente trabajo de graduación para el manejo de la comunicación serial vía MODEM.

Entre los distintos protocolos serie existentes, dos de los estándares más utilizados son el RS-232C y el RS-422, que definen los niveles eléctricos de señal y el significado de las distintas líneas de estas señales. En comunicaciones serie de baja velocidad, los datos circulan coordinados por el reloj del ordenador, que proporciona los bits de arranque (start) y parada (stop). La nomenclatura RS-232 significa Recommended Standard 232, la C se refiere a la última revisión de ese estándar.

Este estándar RS-232C especifica un conector Sub-D de 9 pines, como el mostrado en la **Figura 3.2**.



**Figura 3. 2 Distribución de pines de un Conector DB-9**

El API de comunicaciones proporciona la siguiente funcionalidad:

Una especificación completa para control y acceso a los puertos serie y paralelo. Esto facilita mucho los desarrollos, ya que disminuye en gran medida la carga de trabajo a realizar para el acceso a estos puertos.

Control total de todos los parámetros de los puertos serie: velocidad en baudios, bits de stop, paridad, bits por trama; así como control manual o automático de las líneas de control de flujo. Normalmente, en RS-232, hay dos líneas de señal y el resto son líneas de control; dependiendo del tipo de comunicación, síncrona o asíncrona, el número de líneas de control seleccionadas puede variar. El API proporciona acceso a las líneas de control fundamentales. Este API utiliza para entrada y salida los streams, además utiliza el modelo de eventos que JavaSoft ha introducido desde el lanzamiento del JDK 1.1 para proporcionar información de las líneas de señal que van cambiando y del estado del buffer.

### 3.3 JDBC

JDBC (Java DataBase Connectivity) es un API de Java que permite al programador ejecutar instrucciones en lenguaje estándar de acceso a Bases de Datos, SQL (Structured Query Language, lenguaje estructurado de consultas),

que es un lenguaje de muy alto nivel que permite crear, examinar, manipular y gestionar Bases de Datos relacionales. Para que una aplicación pueda hacer operaciones en una Base de Datos, ha de tener una conexión con ella, la cual se establece a través de un driver, que convierte el lenguaje de alto nivel a sentencias de Base de Datos. Es decir, las tres acciones principales que realizará JDBC son:

- Establecer la conexión a una base de datos, ya sea remota o no.
- Enviar sentencias SQL a esa base de datos.
- Procesar los resultados obtenidos de la base de datos.

JDBC es para Java lo que ODBC es para Windows. La necesidad de JDBC, a pesar de la existencia de ODBC, viene dada porque ODBC es un interfaz escrito en lenguaje C, que al no ser un lenguaje portable, haría que las aplicaciones Java también perdiesen la portabilidad. ODBC tiene el inconveniente de que se ha de instalar manualmente en cada máquina; al contrario que los drivers JDBC, que al estar escritos en Java son automáticamente instalables, portables y seguros. Toda la conectividad de bases de datos de Java se basa en sentencias SQL, por lo que se hace imprescindible un conocimiento adecuado de SQL para realizar cualquier clase de operación de bases de datos.

### **3.3.1. Acceso JDBC a Bases de Datos**

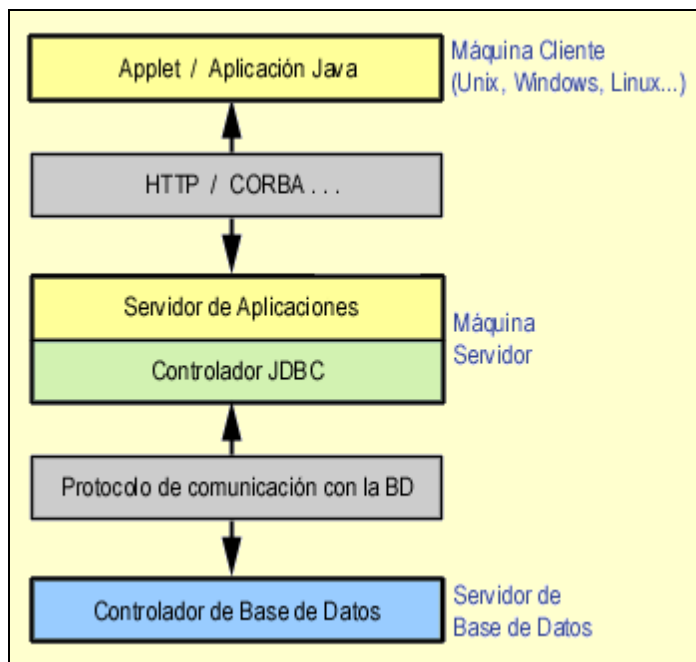
#### **3.3.1.1. Modelo de tres capas**

En este modelo de acceso a bases de datos, las instrucciones son enviadas a una capa intermedia entre Cliente y Servidor, que es la que se encarga de enviar las sentencias SQL a la base de datos y recoger el resultado desde la base de datos.

La **Figura 3.3** muestra el acceso a la base de datos en tres capas.

Este modelo presenta la ventaja de que el nivel intermedio mantiene en todo momento el control del tipo de operaciones que se realizan contra la base de datos; además, está la ventaja adicional de que los drivers JDBC no tienen que residir en la máquina cliente, lo cual libera al usuario de la instalación de cualquier tipo de driver. JDBC define ocho interfaces java para operaciones con bases de datos, de las que se derivan las clases correspondientes.

La **Tabla 3.1** muestra una lista de clases e interfaces junto con una breve descripción.



**Figura 3. 3 Configuración del acceso a una BD en tres capas**

**Tabla 3. 1 Principales clases e interfaces que componen el paquete java.sql**

<b>Clase / Interface</b>	<b>Descripción</b>
Driver	Permite conectarse a una base de datos: cada gestor de base de datos requiere un driver distinto.
DriverManager	Permite gestionar todos los drivers instalados en el sistema.
DriverPropertyInfo	Proporciona diversa información acerca de un driver.
Connection	Representa una conexión con una base de datos. Una aplicación puede tener más de una conexión a más de una base de datos.
Database Metadata	Proporciona información acerca de una Base de Datos, como las tablas que contiene, etc.
Statement	Permite ejecutar sentencias SQL sin parámetros.
PreparedStatement	Permite ejecutar sentencias SQL con parámetros de entrada.
CallableStatement	Permite ejecutar sentencias SQL con parámetros de entrada y salida, típicamente procedimientos almacenados.
ResultSet	Contiene las filas o registros obtenidos al ejecutar un SELECT.
ResultSet Metadata	Permite obtener información sobre un ResultSet, como el número de columnas, sus nombres, etc.

## **3.4 Servicio Web**

### **3.4.1. HTML**

La red Internet es hoy día la red de ordenadores más extensa del planeta. Para ser más precisos, Internet es una red que enlaza centenares de miles de redes locales heterogéneas.

El concepto de documento hipertexto no es nuevo, fue introducido por Ted Nelson en 1965 y básicamente se puede definir como texto de recorrido no secuencial. Clicando en las palabras con enlaces (links) se puede acceder al documento al que apuntan, que normalmente contiene una información más detallada sobre el concepto representado por las palabras del enlace. Pero lo realmente novedoso de la Web es la aplicación del concepto de hipertexto a la inmensa base de información accesible a través de Internet.

HTML es una herramienta fundamental de Internet. Gracias al hipertexto, desde una página Web se puede acceder a cualquier otra página Web almacenada en un servidor HTTP situado en cualquier parte del mundo. Todo este tipo de operaciones se hacen mediante un programa llamado browser o navegador, que básicamente es un programa que reconoce el lenguaje HTML, lo procesa y lo representa en pantalla con el formato más adecuado posible.

Una aplicación web es cualquier aplicación que utiliza el Protocolo de Transferencia de Hipertexto o HTTP como principal protocolo de comunicación y de intercambio de información entre un cliente y un servidor.

Cuando un navegador web envía una petición a un servidor web, éste último se encarga de buscar la página solicitada en su sistema de archivos y devolverla al navegador. Sin embargo, lo que devuelve el servidor web no tiene por qué ser siempre una página HTML estática almacenada en el servidor, ya que puede tratarse de la salida de un programa que se ejecuta en el entorno del servidor web.

En la actualidad, la mayoría de aplicaciones que se utilizan en entornos empresariales están construidas en torno a una arquitectura cliente-servidor, en la cual uno o varios computadores (generalmente de una potencia considerable) son los servidores, que proporcionan servicios a un número mucho más grande de clientes conectados a través de la red. Los clientes suelen ser PCs de propósito general, de ordinario menos potentes y más orientados al usuario final.

A veces los servidores son intermediarios entre los clientes y otros servidores más especializados, por ejemplo, los grandes servidores de bases de datos

corporativos basados en mainframes y/o sistemas Unix. En este caso se habla de aplicaciones de varias capas.

En su concepción más tradicional, los servidores HTTP se limitaban a enviar una página HTML cuando el usuario la requería directamente o clicaba sobre un enlace. La interactividad de este proceso era mínima, ya que el usuario podía pedir ficheros, pero no enviar sus datos personales de modo que fueran almacenados en el servidor u obtuviera una respuesta personalizada. Desde esa primera concepción del servidor HTTP como mero servidor de ficheros HTML el concepto ha ido evolucionando en dos direcciones complementarias:

- Añadir más inteligencia en el servidor.
- Añadir más inteligencia en el cliente.

Las formas más extendidas de añadir inteligencia a los clientes (a las páginas HTML) han sido Java script y las applets de Java. Java script es un lenguaje relativamente sencillo, interpretado, cuyo código fuente se introduce en la página HTML por medio de los tags `<SCRIPT>... </SCRIPT>`; su nombre deriva de una cierta similitud sintáctica con Java. Los applets de Java tienen mucha más capacidad de añadir inteligencia a las páginas HTML que se visualizan en el browser, ya que son verdaderas clases de Java (ficheros \*.class) que se cargan y se ejecutan en el cliente.

De cara a estos apuntes tienen mucho más interés los caminos seguidos para añadir más inteligencia en el servidor HTTP. La primera y más empleada tecnología ha sido la de programas CGI (Common Gateway Interface), unidos a los formularios HTML.

### **3.4.2. Método GET y POST**

Cada formulario lleva incluido un campo llamado "Action" con el que se asocia el nombre de una aplicación o programa en el servidor. El servidor arranca dicho programa y le pasa los datos que han llegado con el formulario. Existen dos formas principales de pasar los datos del formulario a la aplicación en el servidor:

- Método GET: Por medio de una variable de entorno del sistema operativo del servidor, de tipo String, la cual se llama QUERY\_STRING.
- Método POST: Por medio de un flujo de caracteres que llega a través de la entrada estándar (stdin o System.in), que de ordinario está asociada al teclado.

En ambos casos, la información introducida por el usuario en el formulario llega en la forma de una única cadena de caracteres en la que el nombre de cada

campo del formulario se asocia con el valor asignado por el usuario, y en la que los espacios en blanco (blancos) y ciertos caracteres especiales se han sustituido por secuencias de caracteres de acuerdo con una determinada codificación. En cualquier caso, lo primero que tiene que hacer la aplicación en el servidor, es decodificar esta información y separar los valores de los distintos campos. Después ya puede realizar su tarea específica: escribir en un fichero o en una base de datos, realizar una búsqueda de la información solicitada, realizar comprobaciones, etc.

### **3.4.3. Soluciones Web de Servidor**

#### **3.4.3.1. Servlets**

Las Servlets son la respuesta de la tecnología Java a la programación CGI. Son aplicaciones sin interfaz gráfica que se ejecutan en el servidor, y son al igual que las CGI, capaces de construir páginas web. Su ejecución como aplicación independiente es análoga a los programas desarrollados por otros lenguajes.

Construir páginas web al vuelo es útil por diferentes razones:

- La página web podría estar basada en información introducida por el usuario.
- Los datos pueden cambiar frecuentemente, sin embargo la estructura de la página sigue siendo la misma.

Con los Servlets, la Máquina Virtual de Java permanece en ejecución. Administra cada petición mediante un ligero subproceso de Java.

### **3.5 Servidor Tomcat**

Un servidor web es un programa que implementa el protocolo HTTP (Hyper Text Transfer Protocol). Este está diseñado para transferir páginas web o páginas HTML (Hyper Text Markup Language) desde el sistema de almacenamiento local hacia una aplicación cliente que lo solicite. Tales páginas pueden contener textos con enlaces a otros textos o a otras páginas web; formularios, botones y objetos incrustados como animaciones, etc.

Tomcat es la implementación de referencia para la tecnología Java Servlet 2.2. Tomcat, también llamado Yakarta Tomcat o Apache Tomcat, funciona como un contenedor de Servlets, implementa las especificaciones de los servlets y de Java Server Pages de Sun.

Tomcat es un servidor web con soporte de servlets y JSP's; el motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Es importante mencionar que Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java. Los diferentes scripts existentes dentro de la jerarquía de directorios de Tomcat, permiten definir la configuración y acciones que Tomcat tomará ante las peticiones del cliente; entre los directorios cabe destacar los siguientes:

- Bin: Contiene los archivos para arranque y cierre del servicio.
- Conf: Contiene los archivos xml y dtd correspondientes para la configuración de Tomcat.
- Webapps: Directorio donde se alojan todas las aplicaciones web que han sido creadas, tanto scripts HTML como también las servlets que complementan la aplicación.

Las tareas principales de un contenedor de servlets son:

- Proporcionar servicios de red para establecimiento de peticiones y respuestas http.
- Configurar los servlets en función de los descriptores de despliegue (web.xml).
- Gestionar el ciclo de vida de los servlets.

El contenedor web debe implementar las funciones del servidor web (según especificación Servlet).

## **3.6 Aplicaciones para el servicio telefónico y web**

### **3.6.1. Programa de atención de llamadas**

El objetivo principal de esta aplicación es el de atender las llamadas de los diferentes dispositivos y proporcionar a cada uno, una capa intermedia de acceso a la Base de Datos donde se almacenarán los resultados de las pruebas de glucosa realizadas por cada usuario.

Funcionalidades:

- Atiende las llamadas telefónicas hechas por cada dispositivo vía puerto MODEM.
- Verifica y/o valida la existencia del usuario en la BD.
- Conduce la transacción de los datos a través de comandos según el protocolo establecido para ello.
- Verifica los datos (detección de errores).
- Proporciona una capa intermedia de acceso a la BD para lectura y almacenamiento de los datos.
- Permanece a la espera de nuevas llamadas de los distintos dispositivos.

Para llevar a cabo estas tareas, el programa utiliza la implementación del API de comunicaciones de Java y la conectividad con MySQL que proporciona el package "java.sql.\*"

El programa consta de dos partes importantes:

- El segmento de código principal "main()" (**Figura 3.4** secciones 1, 3 y 4), que se encarga de arrancar la aplicación y mantener el contexto de programa para la ejecución del Thread de manejo de la comunicación.

En esta parte del programa se inicializa la aplicación realizando las siguientes tareas:

Se verifica la disponibilidad de los puertos de comunicación a través de los dispositivos registrados por el API de comunicaciones al arranque de la máquina; y si alguno de ellos es el que necesitamos (modem).

Si el MODEM está disponible se configuran los parámetros de la comunicación Baud Rate = 9600, Data Bits = 8, Stop Bits = 1, Parity = NONE.

Se crea la conexión con la Base de Datos y se deja abierta para que el Thread de comunicación realice las consultas y/o actualizaciones necesarias.

El thread principal queda a la espera de un código de salida a través de la entrada estándar (teclado) para finalizar la aplicación. Si se tecldea el código de salida, el programa cierra la conexión a la Base de Datos, cierra la conexión del MODEM, y se termina el programa.

- El segmento de comunicación, que está implementado como un thread para la atención de llamadas (**Figura 3.4** sección 2). Este thread básicamente es un receptor de eventos de puerto serial; el cual está siempre a la espera de la llamada de un dispositivo, para darle seguimiento a la información enviada por éste y llevar a cabo la transacción.



Ante la señal de llamada entrante por parte del MODEM; el thread acepta dicha llamada, respondiendo con el comando "ATA<CR>".

Después que la comunicación se ha establecido, el programa recibe el ID del dispositivo, se consulta la tabla "usuario\_tabla" de la base de datos para verificar el registro del ID de usuario y obtener la fecha y la hora de la última lectura de glucosa registrada en la base de datos en formato de 10 dígitos o en su defecto la cadena "0000000000" en caso de ser la primera vez que se envía información.

Si no se ha registrado el usuario, se devuelve "1<CR>" para cerrar la comunicación, caso contrario (el usuario ya ha sido registrado), se devuelve al dispositivo la fecha y hora de la última lectura en el formato de 10 dígitos "AAMMDDhhmm" obtenida de la consulta a la base de datos.

Luego el programa recibe los datos de lecturas de glucosa cuyas fechas sean posteriores a la fecha registrada en la BD enviada en el estado anterior.

#### **3.6.1.1. Validación de los datos recibidos**

Es importante notar en este momento que la cadena de lecturas enviadas por el dispositivo, consiste de las siguientes partes:

Un header de 3 bytes; "n" tramas (dependiendo de la cantidad de lecturas) con 15 bytes por trama, el inicial es un caracter en blanco (inicio/separador de trama), los restantes 14 bytes contienen la información médica; Un byte finalizador de cadena, el cual es un caracter <CR> (Carriage Return), que indica el fin de la información.

El proceso de validación de los datos incluye:

- Verificación del número indicado por los tres bytes de cabecera de la cadena de información.
- Verificación y conteo de los bytes de inicio de cada trama, los cuales deben corresponder a un espacio en blanco por cada trama.
- El numero contenido en los 3 bytes iniciales, debe coincidir con el conteo de espacios en blanco, caracter de inicio separador de cada trama de lectura.

Los bytes (caracteres) totales leídos del puerto serial, deben ser igual a 15 bytes por cada lectura de glucemia transmitida, más 3 bytes de la cuenta total de las lecturas enviadas, más 1 byte que indica que la transmisión de lecturas ha sido

completada (Retorno de Carro <CR> ó 0X0D en hexadecimal). Por ejemplo si el dispositivo envía 14 tramas, el total de bytes será:

14 tramas \* 15 bytes/trama + 4 bytes = 214 bytes

- Se verifica además que todos los bytes que componen cada lectura de glucemia sean numéricos.

Si existe algún error se devuelve “1<CR>”, para indicarle al dispositivo que hubo error en la transmisión de la información, así el dispositivo presenta en su display LCD un mensaje de error, permitiéndole saber al usuario que debe intentar conectarse nuevamente con el ordenador remoto.

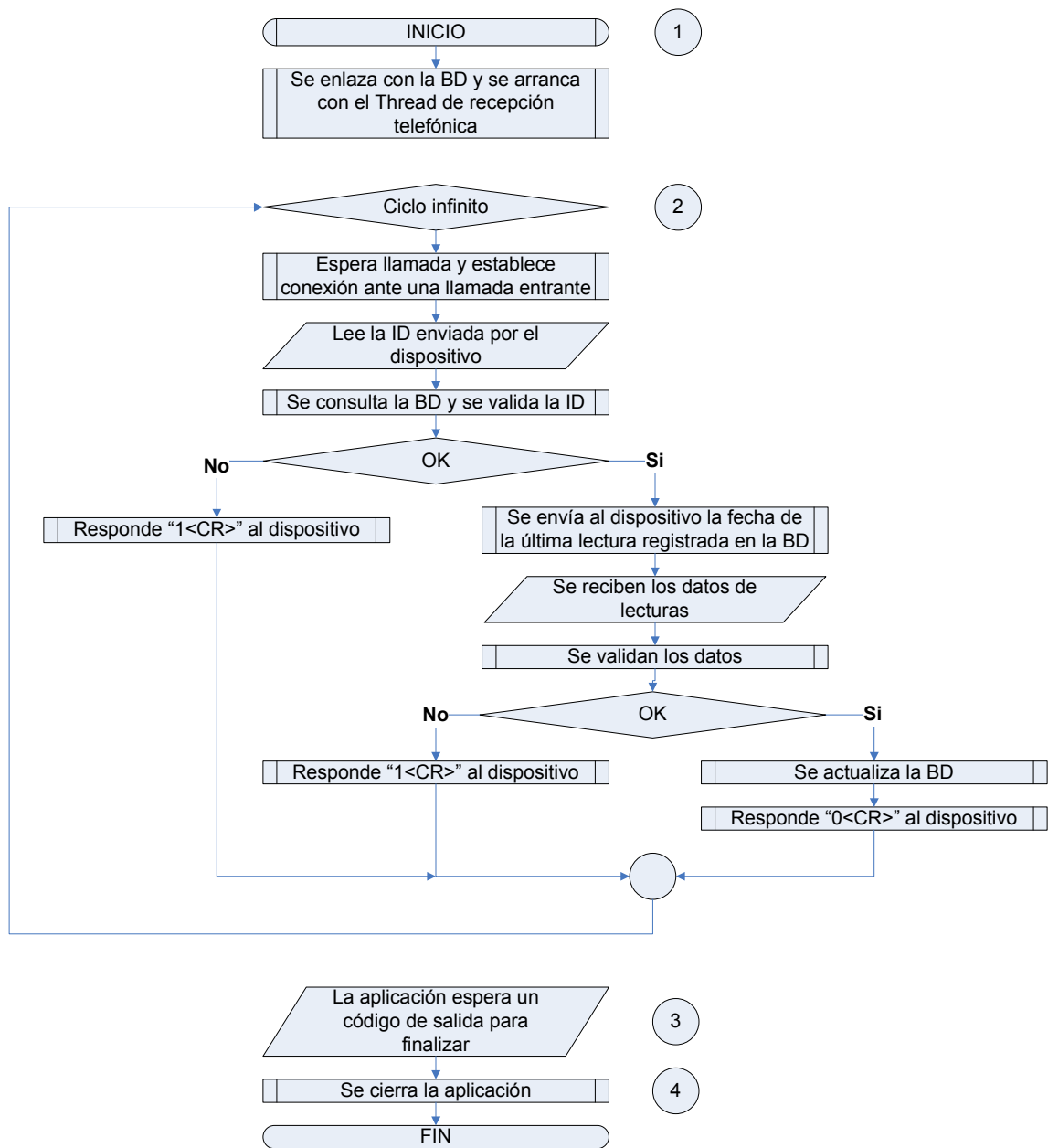
Si los datos no contienen errores, se descompone cada trama y se convierten los datos al formato adecuado a los campos de la tabla “tablalecturas”. Se actualiza la tabla “tablalecturas” de la base de datos con la información obtenida, un registro por cada lectura (trama), y se guarda en la tabla “usuario\_tabla” la fecha y hora de la lectura más reciente en el formato de 10 dígitos (AAMMDDhhmm), finalmente, se responde “0<CR>” al dispositivo para hacerle saber que los datos se actualizaron con éxito, igualmente, el dispositivo electrónico presenta un mensaje en su display LCD informándole al usuario que la transmisión de información se ha completado satisfactoriamente; el usuario puede corroborar que sus lecturas de glucemia llegaron correctamente a la Base de Datos, consultando la aplicación web que se explica en el siguiente apartado.

La **Figura 3.4** es un esquema de los pasos básicos que deben desarrollarse para alcanzar una transmisión, recepción y actualización de datos exitosa.

### **3.6.2. Programa de aplicación web**

El objetivo principal de la aplicación web es proporcionar una interface de administración y consulta de las lecturas de glucemia de los pacientes, así como sus datos personales y de su dispositivo electrónico, igualmente establecer los privilegios respectivos a los diferentes usuarios (pacientes, médicos y administradores) de la aplicación web.

El servicio consta de una página principal (principal.html) que servirá de partida para la selección de las acciones administrativas a realizar en la BD; y seis aplicaciones java (Servlets) que son las que actuarán directamente sobre la BD realizando consultas y/o actualizaciones sobre los datos.



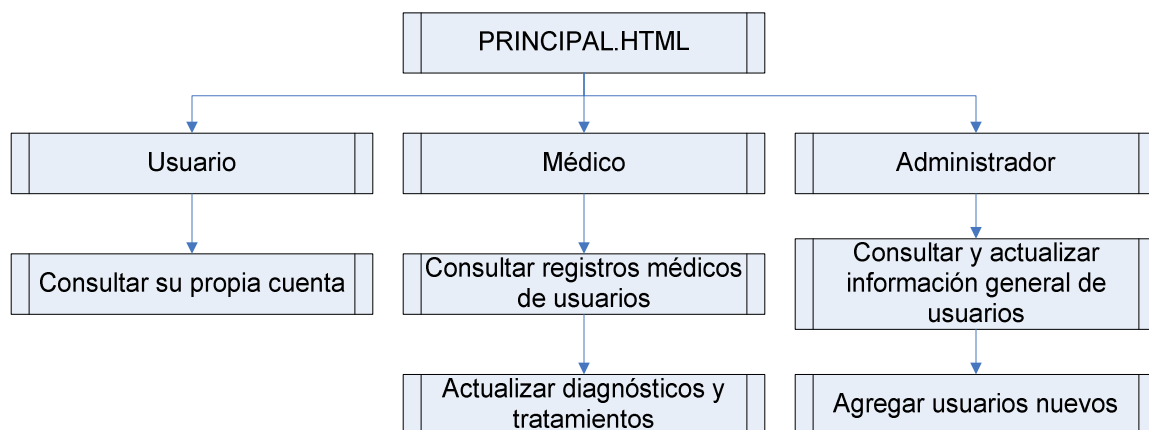
**Figura 3. 4 Diagrama secuencial de la aplicación de recepción de llamadas**

La aplicación web proporciona una interface gráfica para el acceso a la BD con los siguientes propósitos:

- Consulta de los datos personales por parte de los tipos de usuario Paciente y Administrador.
- Consulta de los datos de lectura de glucemia por parte de Médicos para los pacientes asignados.

- Actualización de Diagnósticos y Tratamientos de pacientes por parte de Médicos asignados a cada usuario.
- Consulta y actualización de datos comerciales de usuarios por parte de Administradores registrados.
- Proporciona una capa intermedia de acceso a la BD para seguridad de los datos comerciales.

La **Figura 3.5** muestra un esquema de las funciones que pueden realizar los diferentes usuarios dentro de la aplicación de consulta web. Todos los usuarios de la aplicación web: pacientes, médicos y administradores, están restringidos a realizar acciones específicas dentro de la aplicación de consulta web, dicho control de acciones se lleva a cabo por la ejecución de Servlets dependiendo de la jerarquía del usuario.

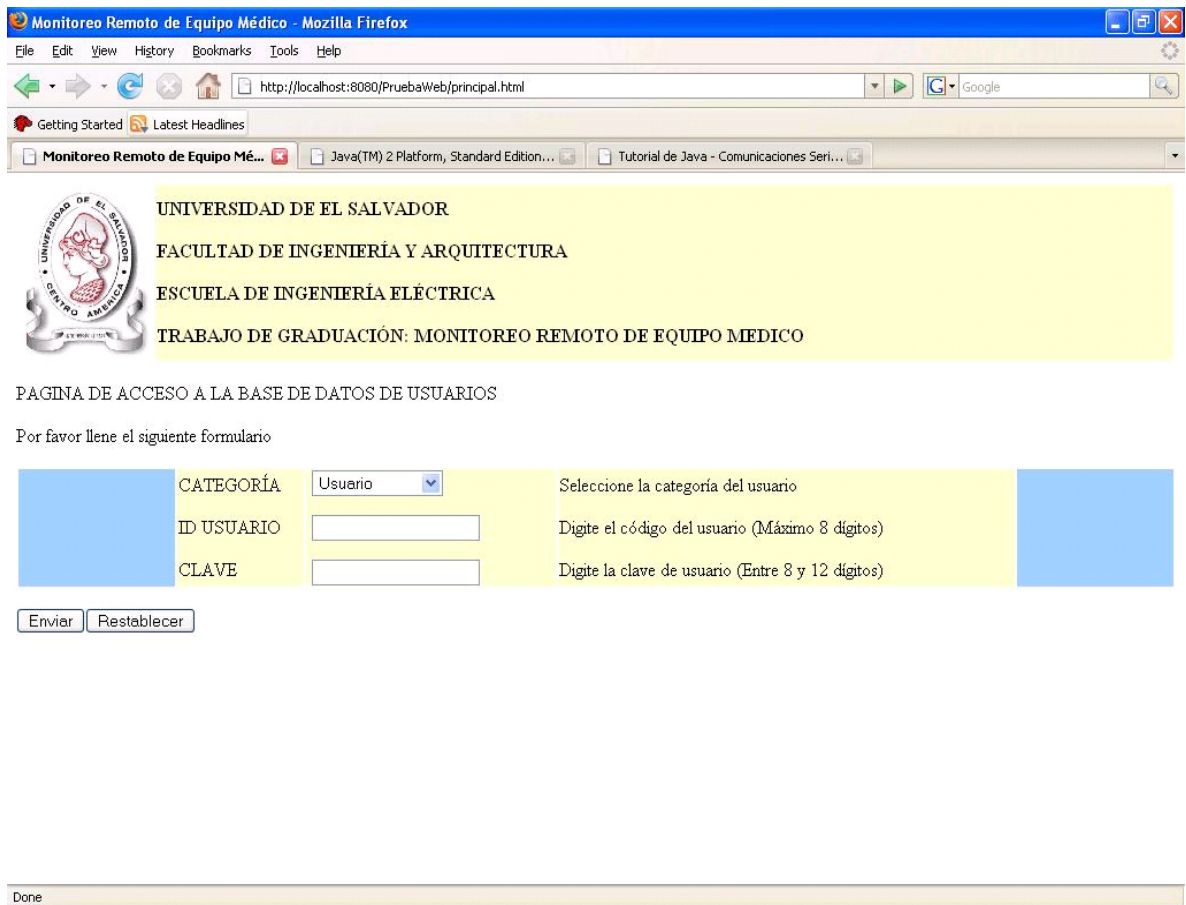


**Figura 3. 5 Esquema de funciones de la aplicación Web**

La **Figura 3.6** presenta la página principal de la aplicación de consulta web.

La dirección para acceder a la página tendrá el formato:  
<http://direccion:puerto/PruebaWeb/principal.html>

Donde **dirección**, es la dirección IP de la maquina que tiene levantado el servicio (debería ser una IP pública, pero podría ser una IP visible solo para los usuarios de una red privada), **puerto** es el número de puerto por el cual se atenderán las solicitudes de los usuarios de la aplicación para nuestro caso el puerto utilizado es el 8080.



**Figura 3. 6** Página principal de la aplicación de consulta web

El acceso se llevara a cabo definiendo en el campo **CATEGORÍA**, la categoría con la que fue registrado el usuario, la cual puede ser: Usuario, Médico o Administrador.

Se definirá además el **ID de Usuario**, que en el caso de pacientes corresponde al número de serie asignado al dispositivo; y en el caso de médicos o administradores, es un número correlativo asignado por la misma aplicación.

En el campo **CLAVE** se digitara el password registrado para el usuario.

Estos datos se compararan contra la base de datos, con la tabla respectiva según la categoría de usuario, para validar los datos ingresados a la aplicación y otorgar o denegar el acceso a la información a un determinado usuario que desee acceder a los datos almacenados.

### 3.6.2.1. Acceso como usuario

El usuario (Paciente) accede a la aplicación con ID y CLAVE.

Si el usuario esta registrado, y su ID y CLAVE son digitados adecuadamente, se despliega una página que muestra los datos personales del paciente y el historial de lecturas de glucemia, si existiesen lecturas ya almacenadas.

Si el usuario no esta registrado o si hubiese un error en el ingreso de los datos de ID y CLAVE, se desplegara el mensaje “USUARIO NO REGISTRADO”

### 3.6.2.2. Acceso como médico

El usuario Médico accede a la aplicación con ID y CLAVE.

Si el médico esta registrado, y su ID y CLAVE son digitados adecuadamente, se despliega una página que muestra los datos personales del médico y un formulario con un control de selección presentando el listado de pacientes asignados; de tal listado el medico podrá seleccionar un paciente para consultar su historial medico.

Luego se despliega una pagina con los datos médicos del paciente consultado y un formulario donde el medico puede actualizar el diagnostico y tratamiento del paciente.

Al finalizar la actualización del tratamiento, deberá aparecer una página con el mensaje “DATOS ACTUALIZADOS SATISFACTORIAMENTE”.

Al igual que en el ingreso como paciente, si el medico no esta registrado o si los datos de ID y CLAVE no son digitados correctamente, se desplegara el mensaje “MEDICO NO REGISTRADO”.

### 3.6.2.3. Acceso como administrador

El usuario Administrador accede a la aplicación con ID y CLAVE.

Si el administrador esta registrado, y su ID y CLAVE son digitados adecuadamente, se despliega una página que muestra los datos personales del administrador y un formulario con dos campos de selección:

- **ACCIÓN:** definirá la acción a realizar entre las opciones: Dar de Alta un usuario, Consultar la cuenta de un usuario o Modificar la cuenta de un usuario.

- **CATEGORÍA:** definirá el tipo de usuario (Paciente, Médico o Administrador) en que se llevara a cabo la acción antes especificada.

Los formularios desplegados posteriormente dependen del tipo de acción que se seleccionó:

- **Dar de alta un usuario:** Despliega un formulario donde se ingresan los datos personales del nuevo usuario.
- **Consultar la cuenta de un usuario:** Despliega la información personal de un usuario previamente seleccionado.
- **Modificar la cuenta de un usuario:** Despliega un formulario con los datos actuales registrados de un usuario previamente seleccionado, con la posibilidad de ser modificados y enviados para actualización de la base de datos.

#### **3.6.2.4. Programas que componen la aplicación web**

A continuación se detalla el funcionamiento de cada pieza del programa:

- **Principal.html:** Permite acceder a la aplicación con una determinada categoría de usuario o cliente, la cual puede variar entre “Usuario”, “Médico”, “Administrador”; un código numérico de usuario asignado por el sistema y una clave de usuario asignada por el administrador. Este registro a la entrada de la aplicación, permite además delimitar el nivel de acceso de cada categoría de usuario.
- **Principal.java:** En esta parte se validan los datos introducidos en el formulario de entrada y responderá al usuario dependiendo de la categoría con la que haya ingresado.
- **Categoría Usuario:** Devuelve un resumen de los datos comerciales y médicos del usuario.
- **Categoría Médico:** Devuelve un resumen de los datos personales del médico y en el caso que el médico tuviese usuarios asignados, se presenta un formulario para seleccionar y consultar los datos médicos de lectura de glucemia de un paciente en particular.
- **Categoría Administrador:** Devuelve un resumen de los datos del administrador y un formulario para la selección de las acciones a realizar sobre la BD.

- ProcFormMD.java: Es llamado a partir del formulario de Médico generado por “Principal.java”, y permite seleccionar el paciente cuyos registros médicos se desea consultar según hayan sido asignados al médico.
- ActMedico.java: Es llamado por el formulario de médico generado por “ProcFormMD.java”, y permite actualizar los datos de diagnóstico y tratamiento del paciente consultado.
- ProcFormADM.java: Este formulario se genera a partir del programa “Principal.java” y permite al administrador seleccionar una acción (consulta, entrada o actualización) a realizar sobre un usuario específico también seleccionado en este formulario.
- ProcAccion.java: Es llamado por “ProcFormADM.java” y su función es operar según la acción seleccionada en el formulario de administrador, preparar los datos introducidos en el formulario de administrador y realizar las consultas necesarias y/o actualizar la BD con los datos introducidos.
- ProcAccionT.java: Permite realizar las actualizaciones finales sobre la BD y devolver al browser una respuesta que indique si se llevó a cabo o no la transacción solicitada.

### **3.7 Base de datos y el servidor MySQL**

El software MySQL proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multi-threaded, multiusuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido. MySQL es una marca registrada de MySQL AB.

El software MySQL tiene una doble licencia. Los usuarios pueden elegir entre usar el software MySQL como un producto Open Source bajo los términos de la licencia GNU General Public License o pueden adquirir una licencia comercial estándar de MySQL AB.

MySQL AB se estableció originalmente en Suecia por David Axmark, Allan Larsson, y Michael "Monty" Widenius. MySQL, el sistema de gestión de bases de datos SQL Open Source más popular, lo desarrolla, distribuye y soporta MySQL AB. MySQL AB es una compañía comercial, fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio.



- MySQL es un sistema de gestión de bases de datos. Una base de datos es una colección estructurada de datos. Puede ser cualquier cosa, desde una simple lista de compra a una galería de pintura o las más vastas cantidades de información en una red corporativa. Para añadir, acceder, y procesar los datos almacenados en una base de datos necesita un sistema de gestión de base de datos como MySQL Server. Al ser los computadores muy buenos en tratar grandes cantidades de datos, los sistemas de gestión de bases de datos juegan un papel central en computación, como aplicaciones autónomas o como parte de otras aplicaciones.
- MySQL es un sistema de gestión de bases de datos relacionales. Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad. La parte SQL de "MySQL" se refiere a "Structured Query Language". SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL. El estándar SQL ha evolucionado desde 1,986 y existen varias versiones hasta la fecha.
- MySQL software es Open Source. Open Source significa que es posible para cualquiera usar y modificar el software. Cualquiera puede bajar el software MySQL desde internet y usarlo sin pagar nada. Si lo desea, puede estudiar el código fuente y cambiarlo para adaptarlo a sus necesidades. El software MySQL usa la licencia GPL (GNU General Public License).

### 3.7.1. Historia de MySQL

Cuando se trataba de trabajar con grandes bases de datos se empezó con la intención de usar MySQL para conectar a las diferentes tablas utilizando diversas rutinas rápidas de bajo nivel (ISAM). Sin embargo y tras algunas pruebas, los pioneros llegaron a la conclusión que MySQL no era lo suficientemente rápido o flexible para la mayoría de las necesidades. Esto provocó la creación de una nueva interfaz SQL para diversas bases de datos; pero casi con la misma interfaz API que MySQL. Esta API fue diseñada para permitir código de terceras partes que fue escrito para poder usarse con MySQL.

La derivación del nombre MySQL no está clara. Según el directorio base del fabricante y un gran número de sus librerías, bibliotecas y herramientas han tenido el prefijo "My" por más de 10 años. Sin embargo, la hija del cofundador Monty Widenius también se llama My. Cuál de los dos dio su nombre a MySQL todavía es un misterio, inclusive para el fabricante. El nombre del delfín de MySQL (logo oficial) es "Sakila", que fue elegido por los fundadores de MySQL AB de una gran lista de nombres sugerida por los usuarios en el concurso "Name the Dolphin" (ponle nombre al delfín). El nombre ganador fue enviado por Ambrose Twebaze, un desarrollador de software Open Source de Swaziland,

África. Según Ambrose, el nombre femenino de Sakila tiene sus raíces en Siswate, el idioma local de Swaziland. Sakila también es el nombre de una ciudad en Arusha, Tanzania, cerca del país de origen de Ambrose, Uganda.

### **3.7.2. Consultas**

Para realizar una consulta, es necesario interactuar con la consola de MySQL a través de un comando; el cual, normalmente consiste en una sentencia SQL seguida de punto y coma. (Hay excepciones donde el punto y coma puede omitirse. Como el caso del comando QUIT, con el cual se abandona la sesión de MySQL).

Cuando ingresa un comando, MySQL lo envía al servidor para ser ejecutado e imprime los resultados para la consulta específica. A continuación muestra de nuevo el prompt MySQL> al usuario para informarle que está listo para recibir otro comando. Después de efectuada cada consulta, MySQL imprime los resultados de la consulta en forma tabulada (filas y columnas). La primera fila contiene etiquetas para las columnas. Las filas siguientes son los resultados de la consulta. Generalmente, el nombre de cada columna es el nombre del campo que trae desde la base de datos. Si está trayendo el valor de una expresión, en lugar del contenido de un campo o columna de una tabla MySQL etiqueta la columna usando el texto de la expresión.

MySQL informa además, posterior a cada consulta cuántas filas fueron devueltas y cuánto tiempo le tomó el ejecutar la consulta, lo cual da una idea aproximada del rendimiento del servidor. Estos valores son imprecisos porque representan tiempo de reloj corriente (no tiempo de CPU), y además, porque están afectados por factores como la carga del servidor o la latencia de red.

MySQL es un programa especialmente útil cuando se desea hacer una compilación de información de diferentes tipos; enlazadas a través de una clave primaria (Primary Key). Los tipos de información que se pueden almacenar en una base de datos, pueden ser numéricos, sean estos enteros y decimales (double), pueden ser fechas y horas; texto, etc. Para hacer una consulta es necesario interactuar con el software a través de una serie de comandos inherentes a MySQL. Los comandos que se listan a continuación, son el repertorio más básico de instrucciones en el trabajo con bases de datos; sin mencionar, que son las instrucciones utilizadas en el desarrollo del presente trabajo de graduación, y se hace referencia únicamente a los comandos utilizados en el desarrollo del mismo; para mejor referencia y conocimientos de todos los comandos refiérase al Manual de MySQL “MySQL 5.0 Reference Manual”.

### 3.7.3. Comandos

#### 3.7.3.1. Show

Mediante la sentencia `SHOW` se encuentran las bases de datos que existen actualmente en el servidor.

```
MySQL> SHOW DATABASES ;
+-----+
| Database |
+-----+
| MySQL    |
| test     |
| tmp      |
+-----+
```

La lista que se obtiene depende de las bases de datos disponibles en el computador que tiene habilitado el servicio, las que se presentan en el ejemplo son las bases que se encuentran por defecto en la configuración de MySQL. Estas bases de datos por defecto, le ayudan al software a realizar su trabajo. La base de datos MySQL es necesaria porque es la que describe los privilegios de acceso de los usuarios. La base de datos test se provee para que los usuarios hagan pruebas. Si el software no otorga el privilegio al usuario de ver las bases de datos disponibles (`SHOW DATABASES`), no podrá ver todas las bases de datos que hay en el servidor.

Asimismo, como se verá más adelante, cuando se cubra el tópico de las tablas, se podrá corroborar que el comando `show` no sólo sirve para presentar las bases de datos disponibles sino también para presentar las tablas disponibles dentro de una base de datos específica a través de la sentencia “`show tables`”.

#### 3.7.3.2. Create database

Para comenzar a utilizar una base de datos es necesario crear una base de datos nueva, es decir una base de datos en la cual se pueda almacenar toda la información que el usuario necesita tener disponible para futuras consultas. El Comando `Create` se utiliza para crear una base de datos seguida por el argumento `DATABASE`, indicando que la petición de creación que se le hace a MySQL. Posterior al comando y al argumento se agrega el nombre con el cual se quiere llamar a la base de datos por crear, de tal forma que a MySQL se le da el comando el argumento y el nombre de la tabla a crear como se muestra en el siguiente ejemplo.

```
MySQL> CREATE DATABASE menagerie;
```

Como se puede observar en el ejemplo anterior, “menagerie”, sería el nombre con el cual se desea bautizar la nueva base de datos a crear. En los sistemas Unix, los nombres asignados a las bases de datos son “case sensitive” es decir que diferencian mayúsculas de minúsculas, en cambio sobre la plataforma de Windows, los nombres de las bases de datos son “no case sensitive” es decir ven indiferente si el nombre va con mayúsculas o no.

Luego que se ha creado la tabla se puede comenzar a trabajar en ella creando los diversos campos y los tipos de caracteres o información que contendrá cada campo.

### **3.7.3.3. Use**

Posterior a crear la base de datos donde se pretende trabajar, esta base de datos recién creada se debe de activar en uso, por defecto en MySQL se debe de activar cada tabla en la que se va a trabajar. El hecho que se cree una base de datos no quiere decir que se puede comenzar a trabajar en ella inmediatamente; para activar cada base de datos se utiliza el comando USE, como se muestra a continuación.

```
MySQL> USE menagerie;  
Database changed
```

La base de datos recién creada (menagerie); con el comando USE se ha habilitado para que el usuario pueda trabajar con ella. MySQL después que se le ha hecho la petición de habilitación manda un mensaje donde avisa al usuario que se ha cambiado efectivamente la base de datos actual de trabajo.

### **3.7.3.4. Create table**

Luego que se ha activado una base de datos para trabajar, se crea una tabla donde se guarda la información; valga la pena aclarar que una base de datos puede contener “n” número de tablas, donde cada tabla puede contener “p” números de registros y cada registro “q” número de apartados de clasificación; sin embargo, cada tabla sólo posee una tan sola clave principal o “primary key”.

Todas las tablas que se creen deben pertenecer a una base de datos, es decir que MySQL no aceptará el comando “create table” si antes no se ha activado una base de datos disponible para el trabajo. La sintaxis de este comando es similar a la sintaxis para la creación de las bases de datos, se utiliza siempre el comando create seguido del argumento table y del nombre que se le dará a la

tabla por crear y entre paréntesis los nombres de los registros que se llevarán dentro de la tabla y el tipo de registro que corresponde a cada apartado como se ve a continuación.

```
MySQL> CREATE TABLE pet (name VARCHAR(20), owner
VARCHAR(20),
-> species VARCHAR(20), sex CHAR(1), birth DATE, death
DATE);
```

En la sentencia anterior, se crea una tabla de nombre “pet”, la cual contendrá 6 diferentes tipos de apartados, de los cuales tres son de tipo VARCHAR, uno de tipo CHAR y dos de tipo DATE. Posterior al nombre de cada apartado se coloca el tipo al que pertenecerá cada apartado y también el número de caracteres asignados a cada apartado.

El tipo de dato VARCHAR es una buena elección para las columnas name, owner, y species porque los datos que allí se almacenan no son de longitud uniforme. En realidad no es necesario que todas estas columnas tengan la misma longitud ni que ésta sea 20. En MySQL 5.0.3 y versiones posteriores, normalmente se puede adoptar cualquier longitud entre 1 y 65,535, según lo que se crea más razonable. El por qué se coloca el apartado birth (nacimiento en inglés) y death (muerte en inglés) como tipo DATE es más que evidente.

Ahora que ya tenemos creada la base de datos menagerie y la tabla de nombre pet, cuando nosotros solicitamos la ejecución del comando show tables se observará lo siguiente:

```
MySQL> SHOW TABLES;
+-----+
| Tables in menagerie |
+-----+
| pet                  |
+-----+
```

### 3.7.3.5. Describe

Posterior a que se ha corroborado que la tabla recientemente creada pertenece a la base de datos activa, algunas veces es necesario corroborar que toda la estructura de la tabla esté tal y cual se desea para la aplicación a desarrollar, para tal motivo se hace uso del comando DESCRIBE, de tal forma que éste se acompaña de un argumento, el cual es el nombre de la tabla existente que deseamos corroborar. A continuación un ejemplo:

```
MySQL> DESCRIBE pet;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	YES		NULL	
owner	varchar(20)	YES		NULL	
species	varchar(20)	YES		NULL	
sex	char(1)	YES		NULL	
birth	date	YES		NULL	
death	date	YES		NULL	

Como se puede observar en este ejemplo, el comando “describe” da el detalle completo de toda la estructura interna de la tabla consultada, dando los nombres de los distintos campos, tipos y valores de cada apartado de cada registro; aun mencionando el número de caracteres que posee cada campo.

### 3.7.3.6. Truncate y drop

Ambos comandos sirven para borrar una tabla e inclusive una base de datos; no obstante se debe ser cuidadoso con el uso de cada uno de ellos, puesto que la aplicación y los resultados de dicha aplicación son ampliamente diferentes.

Cuando se desean borrar todos los registros de una tabla o todas las tablas de un base de datos se utiliza el comando “truncate” seguido de un argumento, que para el caso será el nombre de la tabla o base de datos que se quiere limpiar dependiendo de cuál sea el caso. Es decir, que con truncate, se retorna una tabla o una base de datos a su estado inicial de creación, es decir sin ningún registro almacenado, para el caso de las tablas; y sin ninguna tabla creada para el caso de las bases de datos.

El comando “drop” se utiliza cuando se quiere borrar definitivamente sea una tabla o una base de datos. La sintaxis de este comando necesita que se le especifique como argumento del comando el nombre de la tabla o base de datos que se desea borrar, llámesele borrar en este caso a eliminar completamente los registros y la misma tabla en sí; o por su parte, eliminar completamente todas las tablas contenidas en una base de datos y la base de datos en sí. Valga la pena aclarar que ninguno de estos dos comandos tiene “undone” o la opción de deshacer.

### 3.7.3.7. Load data e insert

Luego de crear una tabla, necesitará completarla con datos. Para esto serán de utilidad las sentencias LOAD DATA e INSERT. Dado se que está comenzando

con una tabla vacía, una forma fácil de completarla es creando un fichero de texto que contenga una línea por cada registro, y luego insertando el contenido del fichero en la tabla mediante una sola sentencia utilizando el comando Load Data.

Para esto, debería crear un fichero de texto “.txt”, conteniendo un registro por línea, con cada valor separado por un carácter de tabulación, y dispuestos en el orden en el cual se especificaron las columnas en la sentencia CREATE TABLE. Para valores ausentes o desconocidos se puede usar valores NULL.

A continuación se presenta un ejemplo de lo anterior, en el cual se carga el fichero pet.txt dentro de tabla pet, que a su vez es parte de la base de datos menagerie, vistas en los ejemplos anteriores.

```
MySQL> LOAD DATA LOCAL INFILE '/path/pet.txt' INTO TABLE
pet;
```

Opcionalmente puede especificar en la sentencia LOAD DATA los caracteres que actuarán como separador de campo y fin de línea, pero los valores por defecto son tabulación y nueva línea. Estos son suficientes para que la sentencia lea correctamente el fichero \*.txt. Cuando lo que desea es agregar nuevos registros de a uno por vez, la sentencia INSERT resulta de utilidad. De esta sencilla manera, se suministran valores para cada columna, dispuestos en el orden en el cual se especificaron las columnas en la sentencia CREATE TABLE statement. Suponga que se desea agregar un nuevo registro en la tabla pet de la base de menagerie, la cual venimos arrastrando en los ejemplos, se podría agregar un nuevo registro, usando la sentencia INSERT de este modo:

```
MySQL> INSERT INTO pet
-> VALUES ('Puffball','Diane','hamster','f','1999-03-
30',NULL);
```

Observe que las cadenas alfanuméricas y las fechas son representadas como cadenas delimitadas por apóstrofes. También, con INSERT, se pueden insertar valores NULL directamente, para indicar un valor ausente. No se debe utilizar \N como se hace con LOAD DATA.

### 3.7.3.8. Select

La sentencia SELECT es utilizada para traer información desde una tabla. La sintaxis general de esta sentencia es:

```
SELECT seleccionar_Esto
FROM desde_tabla
```

WHERE *condiciones*;

*seleccionar\_esto* es lo que se quiere ver. Puede ser una lista de columnas, o \* para indicar “todas las columnas.” *desde\_tabla* indica la tabla donde están los datos a recuperar. La cláusula WHERE es opcional. Si está presente, *condiciones* representa las condiciones que cada registro debe cumplir para retornar como resultado. La forma más simple de SELECT recupera todo lo que hay en la tabla:

```
MySQL> SELECT * FROM pet;
```

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	NULL
Claws	Gwen	cat	m	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL
Fang	Benny	dog	m	1990-08-27	NULL
Bowser	Diane	dog	m	1979-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	NULL
Whistler	Gwen	bird	NULL	1997-12-09	NULL
Slim	Benny	snake	m	1996-04-29	NULL
Puffball	Diane	hamster	f	1999-03-30	NULL

Esta forma de SELECT es útil si se quiere revisar la tabla completa, por ejemplo, después de haberla cargado con un conjunto de datos inicial.

Cuando se actualiza una tabla, sea por que se ha insertado un registro, porque se ha seleccionado un registro con el comando “select” o por que se haya borrado un registro, y no se quiere volver a definir la tabla o la base de datos se utiliza el comando “update”, el cual actualiza inmediatamente la base de datos o la tabla que ha sido modificada. Un ejemplo del uso de este comando se presenta a continuación.

```
MySQL> UPDATE pet SET birth = '1989-08-31' WHERE name = 'Bowser';
```

Como se puede observar del ejemplo, UPDATE modifica sólo el registro en cuestión y no requiere que se vuelva a llenar la tabla.

### 3.7.4. Tablas usadas en la base de datos

Hasta ahora se han visto los comandos más generales y a la vez los que se han utilizado en la creación y el uso de las bases de datos necesarias para el correcto funcionamiento del servicio completo de atención de usuarios propuesto en el presente trabajo de graduación, a continuación se presentan las tablas que



se han creado y utilizado en el desarrollo del presente trabajo de graduación con el prompt de MySQL.

Primariamente se hacen uso de cuatros tablas, las cuales se nombran así:

- Usuario\_tabla
- Medico\_tabla
- Adm\_tabla
- Tablalecturas

Cada una de ellas desarrolla diferentes finalidades dentro del servicio de atención o sitio Web de atención, a continuación se describe cada una de ellas, su utilización, campos internos y demás. Vale la pena recordar que los nombres de las tablas o de las bases de datos son “no case sensitives” en la plataforma del Sistema Operativo Windows de Microsoft; sin embargo en los sistemas de Libre distribución (GNU) con núcleo UNIX los nombres de las tablas o de las bases de datos son “case sensitive”, es decir la plataforma de UNIX diferencia letras mayúsculas de minúsculas. Para la descripción de las tablas utilizadas y su respectiva presentación de Verbi gracia en este documento se presentan imágenes de corridas de MySQL bajo la plataforma de Windows.

La base de datos que contiene esta tabla se llama “proyecto”, por lo tal para tener acceso a cualesquiera de las tablas existentes en la base de datos proyecto es necesario habilitar dicha base de datos a través del comando USE con su respectiva sintaxis, comentada anteriormente.

A continuación se describen las tablas de las cuales está compuesta la base de datos que ha sido utilizada para el desarrollo del presente trabajo de graduación. Antes de pasar en materia con la descripción de las tablas se dará una breve explicación del tipo de clientes existentes y capaces de interactuar con la base de datos, los cuales son los siguientes:

- **Usuario:** Es el paciente que está siendo tratado o monitoreado remotamente por el sistema, es el cliente con menor jerarquía en la base de datos y es incapaz de modificar cualquier tipo de información. Para efectos prácticos, el cliente USUARIO, únicamente puede consultar sus datos a una manera de invitado en el servicio de monitoreo.
- **Médico:** Es el nivel medio en la jerarquía de administración en la base de datos, puede modificar o introducir datos en los formularios del paciente, única, estricta y exclusivamente puede manipular la información netamente médica; es decir, que el cliente médico, es el único autorizado por el sistema de monitoreo para asignar valores a los campos de diagnóstico y tratamiento médico. Este cliente es incapaz de modificar la información personal de cualquiera de los clientes existentes.

- **Administrador:** Es el rango más alto en la jerarquía administrativa de la base de datos, tiene privilegios casi totales, es el que asigna valores a todos los campos de información personal (exceptuando campos con información médica) de los usuarios e inclusive los campos con la información personal de los médicos. Es el único capaz de asignar ID de usuario o de médico o cambiar contraseñas de cualquiera de los anteriores. Sin embargo, a pesar de los altos privilegios, NINGÚN CLIENTE o USUARIO es capaz de modificar las lecturas de glucemia y/o su respectiva fecha y hora almacenada en la base de datos; a menos que se actúe directamente desde la consola del servidor de base de datos MySQL.

#### 3.7.4.1. Tabla Usuario\_tabla

Contiene todos los datos generales y específicos de cada usuario paciente que está registrado dentro del sistema de monitoreo alojado en el servidor MySQL. Para actualizar esta tabla es necesario que el médico o el Administrador del Sistema introduzcan o modifiquen los datos de un nuevo usuario o un usuario antiguo, respectivamente, pudiendo hacer cambios en cualquier campo de su información. Esta tabla contiene 20 campos de diferente tipo. Esta tabla es la más compleja, ya que los tres tipos de usuarios existentes, interactúan con ella, como se dijo, el administrador es el encargado de asignar la información en los campos de información general que presentan los formularios y el médico es el encargado de asignar la información en los campos de información médica y el usuario paciente es el que consulta la tabla para revisar su historial médico.

La tabla **Usuario\_tabla** se presenta en la **Figura 3.7**, donde se observa que el campo “idusr”, es la clave primaria de la tabla y el identificador único e inequívoco para cada usuario registrado en el sistema de monitoreo, por tal razón se declara como auto incrementable, es decir que a cada nuevo usuario que sea registrado en el sistema recibirá automáticamente por parte del sistema un identificador único, con el cual se presentará a la hora de hacer el envío remoto de información. Contiene también campos personales de información, inclusive la contraseña de ingreso del usuario (campo axsk); presenta además el diagnóstico profesional del médico encargado, el tratamiento recetado y el identificador del médico tratante que ha sido asignado al paciente.

La tabla de usuario posee un campo especial, “ultlect”, en el cual se guarda la última fecha registrada con lectura alguna, ya que esa fecha será la que se enviará hacia el dispositivo electrónico doméstico, encargado de hacer el envío de información, para que localmente, no envíe la información repetida de lecturas ya almacenadas con anterioridad en la base de datos; para más detalles leer la sección **3.6.1 Programa de atención de llamadas**.

```
mysql> describe usuario_tabla;
```

Field	Type	Null	Key	Default	Extra
idusr	int(8) unsigned zerofill	NO	PRI	NULL	auto_increment
nombres	varchar(100)	YES		NULL	
apellidos	varchar(100)	YES		NULL	
tel	varchar(10)	YES		NULL	
area	varchar(5)	YES		503	
pais	varchar(25)	YES		El Salvador	
depto	varchar(25)	YES		NULL	
munic	varchar(25)	YES		NULL	
col	varchar(25)	YES		NULL	
pol	varchar(8)	YES		NULL	
casa	varchar(8)	YES		NULL	
dgnst	varchar(60)	YES		NULL	
fnac	date	YES		NULL	
sexo	char(1)	YES		NULL	
ultlect	int(10) unsigned zerofill	YES		0000000000	
tpdev	char(1)	YES		1	
axsk	varchar(12)	YES		NULL	
tratam	varchar(250)	YES		NULL	
emailusr	varchar(60)	YES		NULL	
idmed	int(8) unsigned zerofill	YES		NULL	

20 rows in set (0.05 sec)

**Figura 3. 7 Presentación esquemática de la tabla usuario\_tabla**

### 3.7.4.2. Tabla Medico\_tabla

Esta tabla contiene toda la información y datos generales y específicos de cada médico registrado como tal dentro del sistema de monitoreo remoto alojado en el servidor MySQL. Para actualizar los datos correspondientes a los campos de información personal de esta tabla, es necesario que el Administrador del Sistema, únicamente, actualice los datos de un nuevo médico o cambie los datos de un médico ya existente.

La tabla **Medico\_tabla** contiene 16 registros y se presenta en la **Figura 3.8**, en ella se observa que para cada médico hay un registro completo con información en los campos detallados en la tabla medico\_tabla. Cada médico posee un identificador único e inequívoco dentro del sistema de monitoreo, el cual es la clave principal de la tabla, asignada para efecto de vinculación con otras tablas de la base de datos; posee además una contraseña de acceso asignada por el administrador del sistema de monitoreo.

### 3.7.4.3. Tabla Adm\_tabla

Esta tabla contiene los datos de todos aquellos usuarios registrados como administradores del Sistema, es decir los datos de todos aquellos usuarios con el poder suficiente para cambiar cualquier información. Por defecto existe solamente un administrador; no obstante éste puede otorgar privilegios de

administrador a muchos más clientes, inscribiendo su información personal en la tabla Adm\_tabla.

```
mysql> describe medico_tabla;
```

Field	Type	Null	Key	Default	Extra
idmed	int(8) unsigned zerofill	NO	PRI	NULL	auto_increment
nombres	varchar(100)	YES		NULL	
apellidos	varchar(100)	YES		NULL	
espec	varchar(60)	YES		NULL	
tel	varchar(10)	YES		NULL	
area	varchar(5)	YES		503	
pais	varchar(25)	YES		E1 Salvador	
depto	varchar(25)	YES		NULL	
munic	varchar(25)	YES		NULL	
col	varchar(25)	YES		NULL	
pol	varchar(8)	YES		NULL	
casa	varchar(8)	YES		NULL	
fnac	date	YES		NULL	
sexo	char(1)	YES		NULL	
axsk	varchar(12)	YES		NULL	
emailmed	varchar(60)	YES		NULL	

16 rows in set (0.03 sec)

**Figura 3. 8 Presentación esquemática de la tabla medico\_tabla**

La tabla **Adm\_tabla** contiene 16 registros y se presenta en la **Figura 3.9**.

```
mysql> describe adm_tabla;
```

Field	Type	Null	Key	Default	Extra
idadm	int(8) unsigned zerofill	NO	PRI	NULL	auto_increment
nombres	varchar(100)	YES		NULL	
apellidos	varchar(100)	YES		NULL	
ubic	varchar(60)	YES		NULL	
tel	varchar(10)	YES		NULL	
area	varchar(5)	YES		503	
pais	varchar(25)	YES		E1 Salvador	
depto	varchar(25)	YES		NULL	
munic	varchar(25)	YES		NULL	
col	varchar(25)	YES		NULL	
pol	varchar(8)	YES		NULL	
casa	varchar(8)	YES		NULL	
fnac	date	YES		NULL	
sexo	char(1)	YES		NULL	
axsk	varchar(12)	YES		NULL	
emailadm	varchar(60)	YES		NULL	

16 rows in set (0.02 sec)

**Figura 3. 9 Presentación esquemática de la tabla adm\_tabla**

#### 3.7.4.4. Tabla Tablalecturas

Esta tabla estará atada a través de una clave primaria (idusr) a la tabla Usuario\_tabla, ya que será en esta tabla (tablalecturas) donde se almacenarán

todas las lecturas enviadas por cada uno de los usuarios debidamente registrados dentro del sistema. La clave primaria es un artificio de vinculación en las bases de datos a través de la cual se pueden relacionar dos o más tablas dentro de una base de datos, es decir que atados a través de un mismo registro o campo existente dentro de las dos tablas, éstas se puedan vincular una a la otra para evitar confusión en la asignación de valores a los diferentes registros y usuarios inscritos dentro de la misma tabla y base de datos, evitando el asignar información a un usuario equivocado.

Vale la pena mencionar nuevamente que ningún usuario, ni siquiera el Administrador del sistema de monitoreo, puede insertar, borrar o modificar ninguno de los registros almacenados dentro de esta tabla, ya que por ser la tabla que cuenta con el historial médico de cada paciente, no se puede exponer a que por un error o una manipulación adrede algún usuario pueda tergiversar el tratamiento de un paciente. La única forma de cambiar algún registro de esta tabla es trabajando directamente con el servidor de base de datos MySQL a través de su línea de comandos, lo cual inquierere un conocimiento intermedio del tema y el acceso al servidor donde está alojada la base de datos. Dentro de los límites del presente trabajo de graduación, no se contempla una propuesta de seguridad más que las ya antes mencionadas para los registros médicos en el sistema de monitoreo. La tabla “tablalecturas” posee 5 campos; la tabla se presenta en la **Figura 3.10**.

```
mysql> describe tablalecturas;
```

Field	Type	Null	Key	Default	Extra
idusr	int(8) unsigned zerofill	YES		NULL	
actv	char(1)	YES		NULL	
flect	date	YES		NULL	
hlect	time	YES		NULL	
lect	char(3)	YES		NULL	

5 rows in set (0.03 sec)

**Figura 3. 10 Presentación esquemática de la tabla tablalecturas**

Como se puede observar la tabla tablalecturas, posee el campo idusr, el cual es el mismo que el de la tabla Usuario\_tabla, este campo está vinculado con la otra tabla ya mencionada, para que ambos se modifiquen simultáneamente y pueda así evitarse la asignación errónea de información o historial médico al paciente equivocado. Contiene también un campo especial (actv), en el cual con un número se presentará si la lectura de glucemia respectiva fue tomada antes o después de comer. La fecha, hora y valor de cada lectura de glucemia enviada por el usuario a través de la comunicación dispositivo electrónico – servidor.

Vale la pena aclarar que toda la información alojada en los campos de la tabla tablalecturas es enviada directamente desde el dispositivo electrónico en el lugar remoto donde éste se encuentre.

El código fuente de todos los programas mencionados en este capítulo, se encuentran disponibles en el disco compacto adjunto al presente trabajo de graduación.

### 3.8 CONCLUSIONES DEL CAP. III

Se creó una base de datos capaz de almacenar toda la información médica y general de cada paciente inscrito en el sistema de monitoreo remoto, logrando esta función a través de cuatro tablas creadas para propósitos específicos. Se creó una tabla para almacenar las lecturas de glucemia de los pacientes, una tabla para la información médica y personal del paciente tratado, una tabla con la información personal de todos los médicos tratantes y una tabla con la información personal de todos aquellos usuarios con privilegios de administrador. Al no entrar dentro de las limitaciones del trabajo de graduación, la creación de una aplicación de seguridad para la página Web, se optó por que únicamente aquellos usuarios con privilegios de administrador pudiesen cambiar la información personal de los pacientes e inclusive los médicos y por su parte que sólo los médicos pudiesen cambiar la información médica de sus respectivos pacientes; de tal suerte que el paciente o usuario tratado únicamente puede tener acceso a la página de presentación Web en una modalidad de espectador invitado.

Se creó un programa en java aplicando los conceptos de manejo de eventos e hilos de ejecución con el propósito único y especial de gestionar el intercambio de información MODEM – MODEM, es decir, se utilizó una aplicación en java (prueba.java) cuyo propósito es gestionar el envío de las tramas de información desde el dispositivo de monitoreo remoto hasta el servidor que contenga la base de datos; siguiendo un algoritmo estricto mediante el cual, únicamente se enviarán las lecturas que estén registradas como realizadas en fecha posterior a la fecha última almacenada en la base de datos del servidor; haciendo con ello que en la comunicación y el intercambio de información no exista duplicidad con las lecturas ya almacenadas.

Se desarrollo una interfaz de presentación Web, a través de la cual se es capaz de maniobrar la base de datos y de modificarla a conveniencia según directrices diversas de administración, de acuerdo a las necesidades que se posean. Así mismo, el desarrollo del servicio de presentación trabaja en tres modalidades: usuario, médico y administrador, dependiendo de lo cual otorgará mas o menos privilegios dentro de la navegación de la interfaz de monitoreo remoto.

Se utilizó una base de datos creada en el servidor de bases de datos MySQL, debido a su fácil y común manejo; sin mencionar que la licencia es de carácter libre (GNU). Esta base de datos está regida por una clave principal, la cual es un identificador numérico que el mismo servicio de monitoreo asigna a cada nuevo usuario registrado por el administrador que también ser asignado manualmente por el administrador sólo para el caso de los usuarios. Esta clave primaria o principal, es la vinculación entre las diversas tablas de persistencia de datos,

para evitar la confusión de información o a la asignación errada de la misma entre diferentes pacientes.

Se estableció un modelo de administración de la base de datos en tres capas; el cual, se considera óptimo gracias a que permite la gestión de la base de datos a través de las aplicaciones Servlets contenidas en el servidor; también nos evita el tener que instalar de lado del cliente el driver de acceso a la base de datos.



### 3.9 REFERENCIAS BIBLIOGRÁFICAS

BODAS, Sagi Diego J. Introducción a la Programación Distribuida (en línea). Argentina, (citada 20 de febrero de 2007).

[http://www.puntoedu.edu.ar/comunidades/ing/informatica/+info/programacion\\_distribuida\\_en\\_java.pdf](http://www.puntoedu.edu.ar/comunidades/ing/informatica/+info/programacion_distribuida_en_java.pdf)

GARCIA. De Jalón Javier, Rodríguez José Ignacio, Mingo Iñigo. Aprenda Java Como si Estuviera en Primero (en línea). España, (citada 18 de junio de 2006).

<http://www.tecnun.es/asignaturas/Informat1/ayudainf/aprendainf/Java/Java2.pdf>

GARCIA. De Jalón Javier, Rodríguez José Ignacio. Aprenda Servlets de Java Como si Estuviera en Primero (en línea). España, (citada 20 de diciembre de 2006).

<http://www.tecnun.es/asignaturas/Informat1/ayudainf/aprendainf/JavaServlets/servlets.pdf>

MySQL Reference Manual. Versión 5.3.

Sun Microsystems. Java 2 Platform Standard Edition 5.0 API Specification. (en línea). EEUU. (Citada el 10 de Agosto de 2006).

<http://java.sun.com/j2se/1.5.0/docs/api/>

COWARD. Danny, Yoshida Yutaka. Java Servlet Specification Version 2.4. (en línea) EEUU. (Citada el 20 de Febrero de 2007)

[http://memex.dsic.upv.es/pbs/servlet-2\\_4-fr-spec.pdf](http://memex.dsic.upv.es/pbs/servlet-2_4-fr-spec.pdf)

## **CONCLUSIONES GENERALES Y RECOMENDACIONES**

El dispositivo electrónico es funcional con los glucómetros GlucoLab de Infopía, OneTouch Ultra y OneTouch Ultra2 de Jhonson y Jhonson.

Se recomienda a la hora de descifrar los protocolos y tramas de comunicación de los glucómetros, o cualquier dispositivo tecnológico capaz de conectarse con un ordenador, utilizar algún software capaz de observar el tráfico de información en el puerto donde ocurre la comunicación. Para el caso de este trabajo de graduación, los dispositivos (glucómetros) que se investigaron, se comunican a través del puerto serial del ordenador, es por ello que se usó el software Serial Monitor de HDD que registra las transacciones de información por el puerto serial de la computadora.

El dispositivo electrónico transfiere el registro de glucemia del paciente, hasta el ordenador remoto en la clínica del médico responsable, estableciendo una llamada telefónica utilizando un MODEM.

El servicio de atención de llamadas generadas por el dispositivo electrónico, reconoce dicho dispositivo por un código identificador único que éste proporciona durante la transacción de información y es verificado en la base de datos del sistema; de esta forma el servicio, asigna las lecturas de glucemia que recibe durante dicha transacción al paciente propietario del dispositivo registrado en la base de datos. Esta característica hace al aparato electrónico de uso exclusivo y personal.

El servicio de atención de llamadas, desarrollado en el presente trabajo de graduación, es capaz de tomar sólo una llamada a la vez.

Es necesario recordar que el servicio de atención de llamadas funcionará adecuadamente siempre y cuando el período de actualización de lecturas de glucemia a la base de datos no exceda de una semana a lo sumo; puesto que la idea de generar un servicio de atención y monitoreo remoto es para actuar oportunamente y mantener constantemente informado al médico tratante de la condición de un paciente, especialmente cuando éste tenga un tratamiento diferenciado o su condición sea crítica.

Para asegurar la persistencia de la información médica de cada paciente, expresada en las lecturas de glucemia respectivas, es imposible borrar el registro de lecturas de cualquier paciente desde el servicio web; es decir, que ni el Administrador del sistema de monitoreo remoto será capaz de borrar ningún

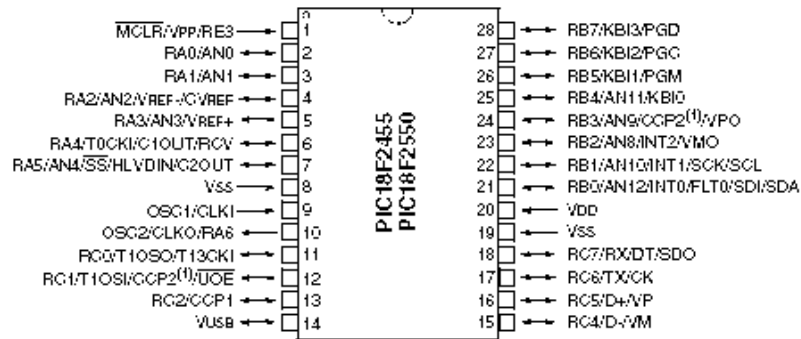
registro de glucemia a través de la página de presentación y administración Web.

La única manera de borrar los registros de las lecturas de glucemia de un paciente, es accediendo directamente a la base de datos del servidor de base de datos, MySQL; para desde ahí, borrar los registros manualmente. Se pensó en este diseño para dar una respuesta a la seguridad de la persistencia de la base de datos; ya que para acceder a la base de datos en sí, es necesario ingresar contraseñas o passwords que únicamente conocerá la persona encargada de mantener saludable el estado de la base de datos, en cuanto al número o cantidad de paciente activos se refiere.

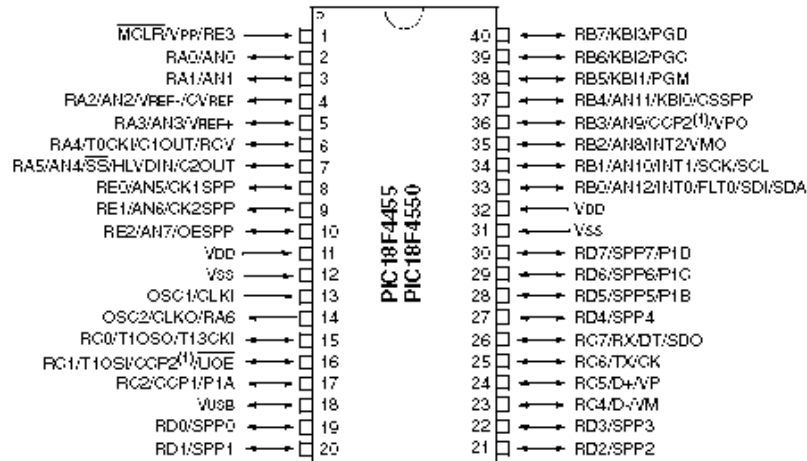
# ANEXO A – Hojas de datos del PIC18F4550

## Pin Diagrams

### 28-Pin PDIP, SOIC



### 40-Pin PDIP



Note 1: RB3 is the alternate pin for CCP2 multiplexing.

### Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

### Power-Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8  $\mu$ A typical
- Sleep mode currents down to 0.1  $\mu$ A typical
- Timer1 Oscillator: 1.1  $\mu$ A typical, 32 kHz, 2V
- Watchdog Timer: 2.1  $\mu$ A typical
- Two-Speed Oscillator Start-up

### Flexible Oscillator Structure:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
  - 8 user-selectable frequencies, from 31 kHz to 8 MHz
  - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if any clock stops

### Peripheral Highlights:

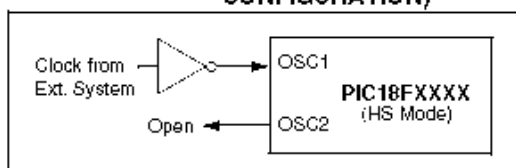
- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution 5.2 ns ( $T_{CV}/16$ )
  - Compare is 16-bit, max. resolution 83.3 ns ( $T_{CV}$ )
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead time
  - Auto-shutdown and auto-restart
- Enhanced USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I<sup>2</sup>C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

### Special Microcontroller Features:

- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICD/ICSP port (44-pin devices only)
- Wide Operating Voltage Range (2.0V to 5.5V)

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EA/USART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I <sup>2</sup> C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

**FIGURE 2-3: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)**

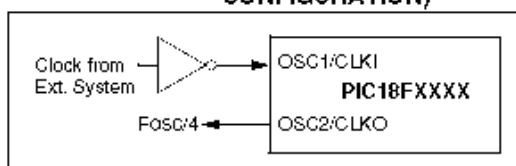


### 2.2.3 EXTERNAL CLOCK INPUT

The EC, ECIO, ECPLL and ECPIO Oscillator modes require an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset or after an exit from Sleep mode.

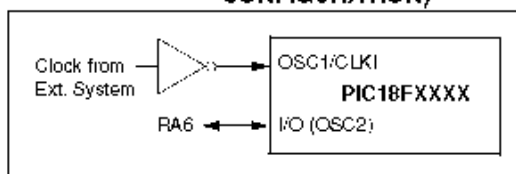
In the EC and ECPLL Oscillator modes, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-4 shows the pin connections for the EC Oscillator mode.

**FIGURE 2-4: EXTERNAL CLOCK INPUT OPERATION (EC AND ECPLL CONFIGURATION)**



The ECIO and ECPIO Oscillator modes function like the EC and ECPLL modes, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 2-5 shows the pin connections for the ECIO Oscillator mode.

**FIGURE 2-5: EXTERNAL CLOCK INPUT OPERATION (ECIO AND ECPIO CONFIGURATION)**



The internal postscaler for reducing clock frequency in XT and HS modes is also available in EC and ECIO modes.

### 2.2.4 PLL FREQUENCY MULTIPLIER

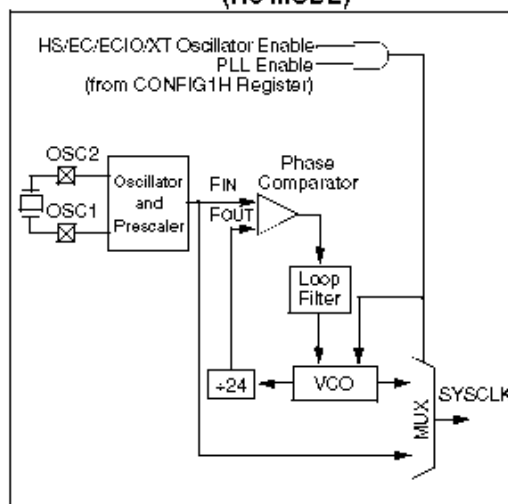
PIC18F2455/2550/4255/4550 devices include a Phase Locked Loop (PLL) circuit. This is provided specifically for USB applications with lower speed oscillators and can also be used as a microcontroller clock source.

The PLL is enabled in HSPLL, XTPLL, ECPLL and ECPIO Oscillator modes. It is designed to produce a fixed 96 MHz reference clock from a fixed 4 MHz input. The output can then be divided and used for both the USB and the microcontroller core clock. Because the PLL has a fixed frequency input and output, there are eight prescaling options to match the oscillator input frequency to the PLL.

There is also a separate postscaler option for deriving the microcontroller clock from the PLL. This allows the USB peripheral and microcontroller to use the same oscillator input and still operate at different clock speeds. In contrast to the postscaler for XT, HS and EC modes, the available options are 1/2, 1/3, 1/4 and 1/6 of the PLL output.

The HSPLL, ECPLL and ECPIO modes make use of the HS mode oscillator for frequencies up to 48 MHz. The prescaler divides the oscillator input by up to 12 to produce the 4 MHz drive for the PLL. The XTPLL mode can only use an input frequency of 4 MHz which drives the PLL directly.

**FIGURE 2-6: PLL BLOCK DIAGRAM (HS MODE)**



## 5.0 MEMORY ORGANIZATION

There are three types of memory in PIC18 enhanced microcontroller devices:

- Program Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate busses: this allows for concurrent access of the two memory spaces. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

Additional detailed information on the operation of the Flash program memory is provided in Section 6.0 “Flash Program Memory”. Data EEPROM is discussed separately in Section 7.0 “Data EEPROM Memory”.

## 5.1 Program Memory Organization

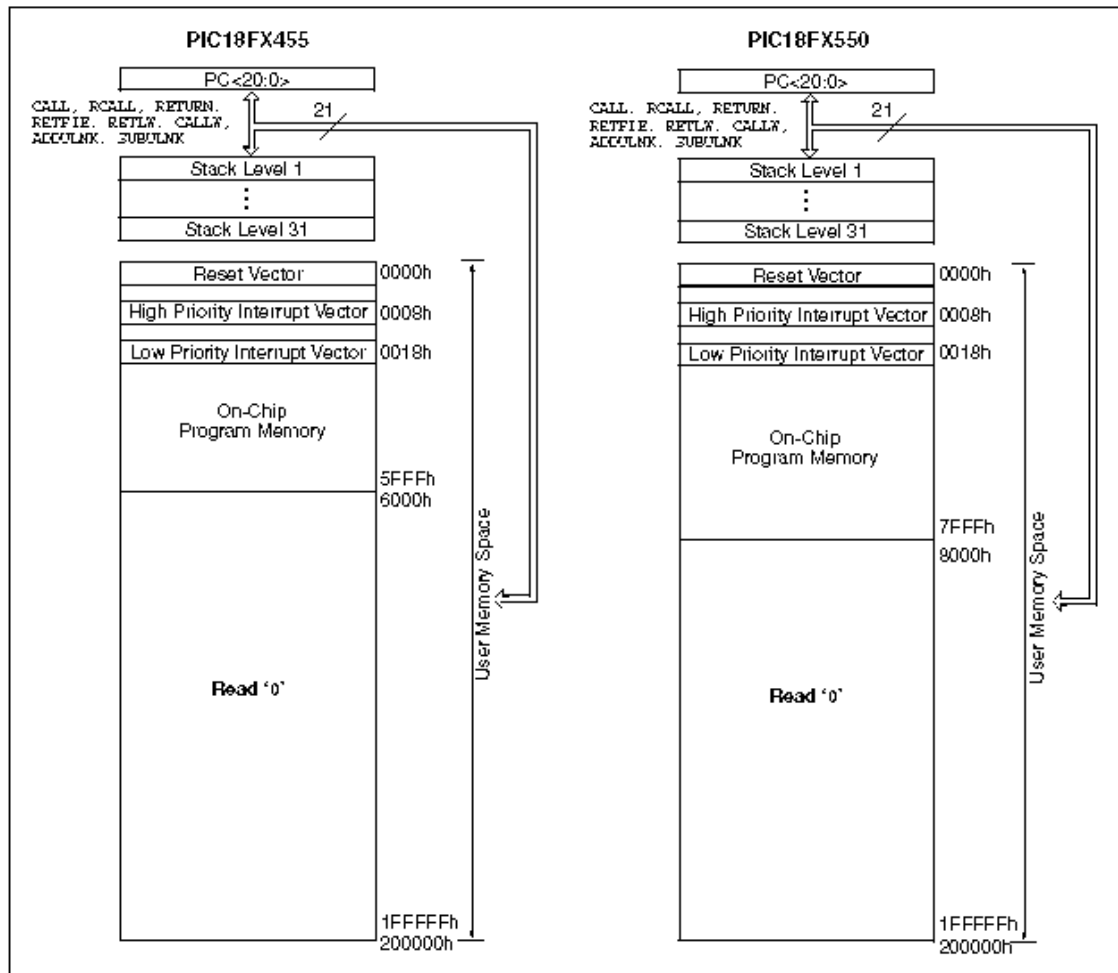
PIC18 microcontrollers implement a 21-bit program counter which is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all ‘0’s (a NOP instruction).

The PIC18F2455 and PIC18F4455 each have 24 Kbytes of Flash memory and can store up to 12,288 single-word instructions. The PIC18F2550 and PIC18F4550 each have 32 Kbytes of Flash memory and can store up to 16,384 single-word instructions.

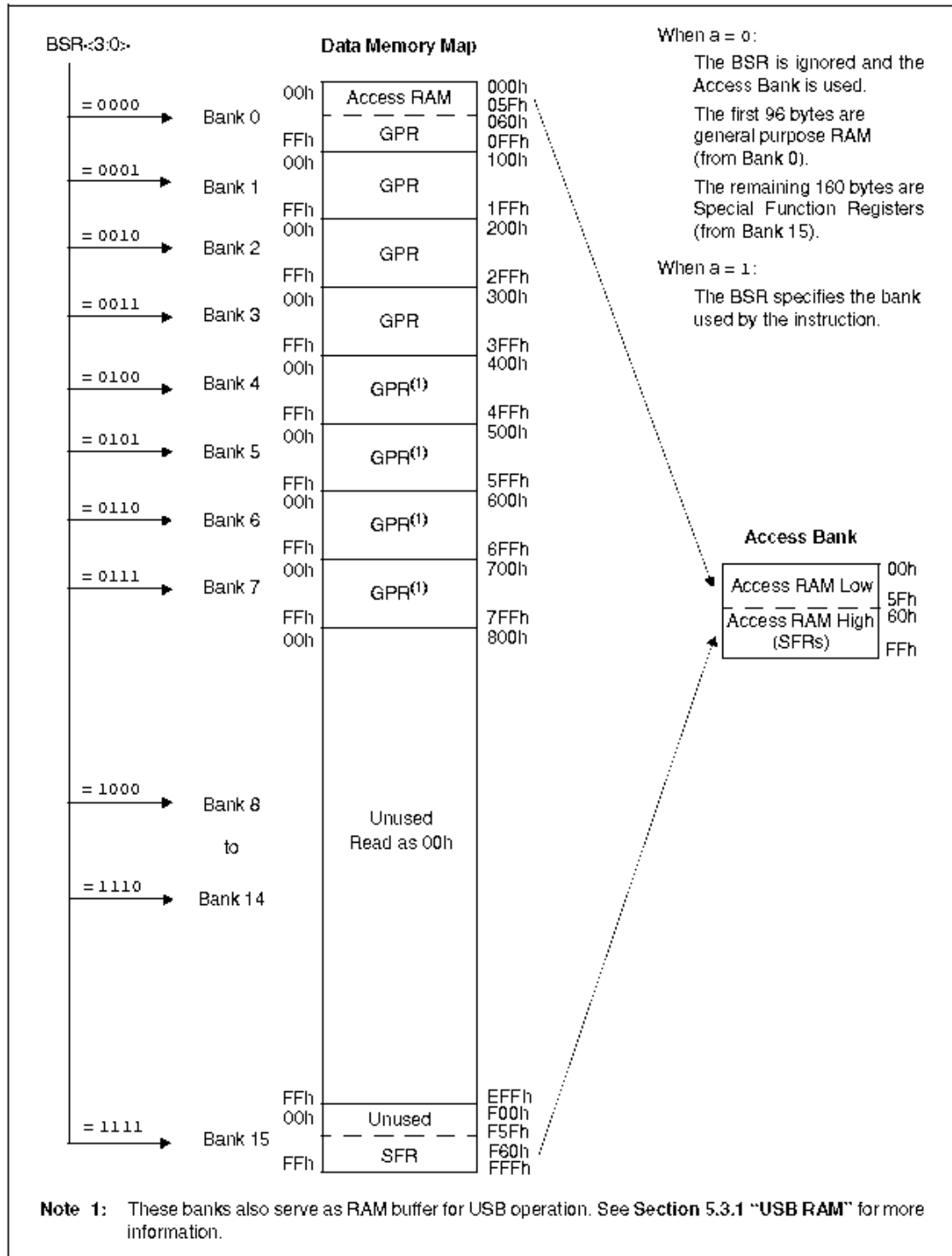
PIC18 devices have two interrupt vectors. The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

The program memory maps for PIC18FX455 and PIC18FX550 devices are shown in Figure 5-1.

**FIGURE 5-1: PROGRAM MEMORY MAP AND STACK FOR PIC18F2455/2550/4455/4550 DEVICES**



**FIGURE 5-5: DATA MEMORY MAP FOR PIC18F2455/2550/4455/4550 DEVICES**





## 5.4 Data Addressing Modes

**Note:** The execution of some instructions in the core PIC18 instruction set are changed when the PIC18 extended instruction set is enabled. See **Section 5.6 “Data Memory and the Extended Instruction Set”** for more information.

While the program memory can be addressed in only one way – through the program counter – information in the data memory space can be addressed in several ways. For most instructions, the addressing mode is fixed. Other instructions may use up to three modes, depending on which operands are used and whether or not the extended instruction set is enabled.

The addressing modes are:

- Inherent
- Literal
- Direct
- Indirect

An additional addressing mode, Indexed Literal Offset, is available when the extended instruction set is enabled (XINST Configuration bit = 1). Its operation is discussed in greater detail in **Section 5.6.1 “Indexed Addressing with Literal Offset”**.

### 5.4.1 INHERENT AND LITERAL ADDRESSING

Many PIC18 control instructions do not need any argument at all; they either perform an operation that globally affects the device or they operate implicitly on one register. This addressing mode is known as Inherent Addressing. Examples include SLEEP, RESET and DA $\bar{W}$ .

Other instructions work in a similar way but require an additional explicit argument in the opcode. This is known as Literal Addressing mode because they require some literal value as an argument. Examples include ADDLW and MOVLW, which respectively, add or move a literal value to the W register. Other examples include CALL and GOTO, which include a 20-bit program memory address.

### 5.4.2 DIRECT ADDRESSING

Direct Addressing mode specifies all or part of the source and/or destination address of the operation within the opcode itself. The options are specified by the arguments accompanying the instruction.

In the core PIC18 instruction set, bit-oriented and byte-oriented instructions use some version of Direct Addressing by default. All of these instructions include some 8-bit literal address as their Least Significant Byte. This address specifies either a register address in one of the banks of data RAM (**Section 5.3.4 “General**

**Purpose Register File”**) or a location in the Access Bank (**Section 5.3.3 “Access Bank”**) as the data source for the instruction.

The Access RAM bit ‘a’ determines how the address is interpreted. When ‘a’ is ‘1’, the contents of the BSR (**Section 5.3.2 “Bank Select Register (BSR)”**) are used with the address to determine the complete 12-bit address of the register. When ‘a’ is ‘0’, the address is interpreted as being a register in the Access Bank. Addressing that uses the Access RAM is sometimes also known as Direct Forced Addressing mode.

A few instructions, such as MOVWF, include the entire 12-bit address (either source or destination) in their opcodes. In these cases, the BSR is ignored entirely.

The destination of the operation’s results is determined by the destination bit ‘d’. When ‘d’ is ‘1’, the results are stored back in the source register, overwriting its original contents. When ‘d’ is ‘0’, the results are stored in the W register. Instructions without the ‘d’ argument have a destination that is implicit in the instruction; their destination is either the target register being operated on or the W register.

### 5.4.3 INDIRECT ADDRESSING

Indirect Addressing allows the user to access a location in data memory without giving a fixed address in the instruction. This is done by using File Select Registers (FSRs) as pointers to the locations to be read or written to. Since the FSRs are themselves located in RAM as Special Function Registers, they can also be directly manipulated under program control. This makes FSRs very useful in implementing data structures, such as tables and arrays in data memory.

The registers for Indirect Addressing are also implemented with Indirect File Operands (INDFs) that permit automatic manipulation of the pointer value with auto-incrementing, auto-decrementing or offsetting with another value. This allows for efficient code, using loops, such as the example of clearing an entire RAM bank in Example 5-5.

#### EXAMPLE 5-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

```
LFSR    FSR0, 100h ;
NEXT    CLRF    POSTINC0 ; Clear INDF
                                ; register then
                                ; inc pointer
                                BTFSS    FSR0H, 1 ; All done with
                                ; Bank1?
                                BRA     NEXT ; NO, clear next
CONTINUE                                ; YES, continue
```

### 5.4.3.1 FSR Registers and the INDF Operand

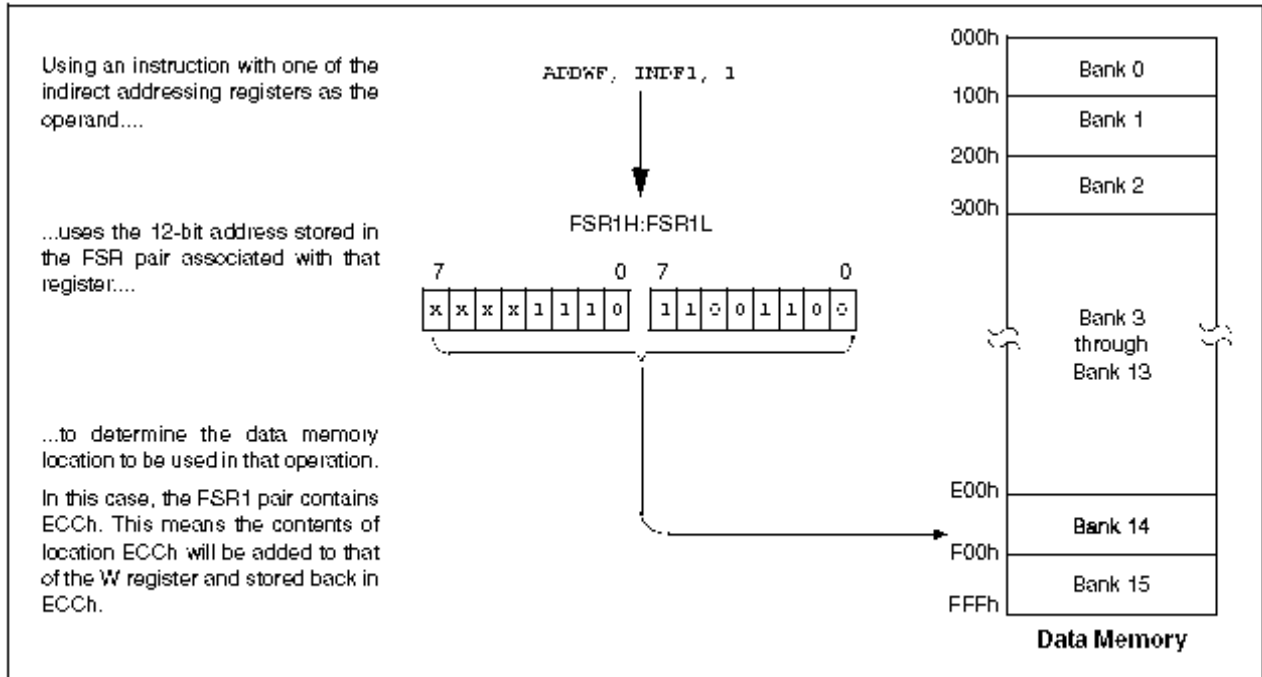
At the core of Indirect Addressing are three sets of registers: FSR0, FSR1 and FSR2. Each represents a pair of 8-bit registers: FSRnH and FSRnL. The four upper bits of the FSRnH register are not used, so each FSR pair holds a 12-bit value. This represents a value that can address the entire range of the data memory in a linear fashion. The FSR register pairs, then, serve as pointers to data memory locations.

Indirect Addressing is accomplished with a set of Indirect File Operands, INDF0 through INDF2. These can be thought of as "virtual" registers; they are

mapped in the SFR space but are not physically implemented. Reading or writing to a particular INDF register actually accesses its corresponding FSR register pair. A read from INDF1, for example, reads the data at the address indicated by FSR1H:FSR1L. Instructions that use the INDF registers as operands actually use the contents of their corresponding FSR as a pointer to the instruction's target. The INDF operand is just a convenient way of using the pointer.

Because Indirect Addressing uses a full 12-bit address, data RAM banking is not necessary. Thus, the current contents of the BSR and the Access RAM bit have no effect on determining the target address.

**FIGURE 5-7: INDIRECT ADDRESSING**



#### 5.4.3.2 FSR Registers and POSTINC, POSTDEC, PREINC and PLUSW

In addition to the INDF operand, each FSR register pair also has four additional indirect operands. Like INDF, these are "virtual" registers that cannot be indirectly read or written to. Accessing these registers actually accesses the associated FSR register pair, but also performs a specific action on its stored value. They are:

- POSTDEC: accesses the FSR value, then automatically decrements it by '1' afterwards
- POSTINC: accesses the FSR value, then automatically increments it by '1' afterwards
- PREINC: increments the FSR value by '1', then uses it in the operation
- PLUSW: adds the signed value of the W register (range of -127 to 128) to that of the FSR and uses the new value in the operation.

In this context, accessing an INDF register uses the value in the FSR registers without changing them. Similarly, accessing a PLUSW register gives the FSR value offset by that in the W register; neither value is actually changed in the operation. Accessing the other virtual registers changes the value of the FSR registers.

Operations on the FSRs with POSTDEC, POSTINC and PREINC affect the entire register pair; that is, rollovers of the FSRnL register, from FFh to 00h, carry over to the FSRnH register. On the other hand, results of these operations do not change the value of any flags in the STATUS register (e.g., Z, N, OV, etc.).

The PLUSW register can be used to implement a form of Indexed Addressing in the data memory space. By manipulating the value in the W register, users can reach addresses that are fixed offsets from pointer addresses. In some applications, this can be used to implement some powerful program control structure, such as software stacks, inside of data memory.

#### 5.4.3.3 Operations by FSRs on FSRs

Indirect Addressing operations that target other FSRs or virtual registers represent special cases. For example, using an FSR to point to one of the virtual registers will not result in successful operations. As a specific case, assume that FSR0H:FSR0L contains FE7h, the address of INDF1. Attempts to read the value of INDF1, using INDF0 as an operand, will return 00h. Attempts to write to INDF1, using INDF0 as the operand, will result in a **NOE**.

On the other hand, using the virtual registers to write to an FSR pair may not occur as planned. In these cases, the value will be written to the FSR pair but without any incrementing or decrementing. Thus, writing to INDF2 or POSTDEC2 will write the same value to the FSR2H:FSR2L.

Since the FSRs are physical registers mapped in the SFR space, they can be manipulated through all direct operations. Users should proceed cautiously when working on these registers, particularly if their code uses Indirect Addressing.

Similarly, operations by Indirect Addressing are generally permitted on all other SFRs. Users should exercise the appropriate caution that they do not inadvertently change settings that might affect the operation of the device.

## 8.0 8 x 8 HARDWARE MULTIPLIER

### 8.1 Introduction

All PIC18 devices include an 8 x 8 hardware multiplier as part of the ALU. The multiplier performs an unsigned operation and yields a 16-bit result that is stored in the product register pair, PRODH:PRODL. The multiplier's operation does not affect any flags in the STATUS register.

Making multiplication a hardware operation allows it to be completed in a single instruction cycle. This has the advantages of higher computational throughput and reduced code size for multiplication algorithms and allows the PIC18 devices to be used in many applications previously reserved for digital signal processors. A comparison of various hardware and software multiply operations, along with the savings in memory and execution time, is shown in Table 8-1.

### 8.2 Operation

Example 8-1 shows the instruction sequence for an 8 x 8 unsigned multiplication. Only one instruction is required when one of the arguments is already loaded in the WREG register.

Example 8-2 shows the sequence to do an 8 x 8 signed multiplication. To account for the sign bits of the arguments, each argument's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

#### EXAMPLE 8-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

```
MOVF ARG1, W      ;
MULWF ARG2        ; ARG1 * ARG2 ->
                  ; PRODH:PRODL
```

#### EXAMPLE 8-2: 8 x 8 SIGNED MULTIPLY ROUTINE

```
MOVF ARG1, W      ;
MULWF ARG2        ; ARG1 * ARG2 ->
                  ; PRODH:PRODL
BTFSK ARG2, SB    ; Test Sign Bit
SUBWF PRODH, F    ; PRODH = PRODH
                  ; - ARG1
MOVF ARG2, W      ;
BTFSK ARG1, SB    ; Test Sign Bit
SUBWF PRODH, F    ; PRODH = PRODH
                  ; - ARG2
```

TABLE 8-1: PERFORMANCE COMPARISON FOR VARIOUS MULTIPLY OPERATIONS

Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time		
				@ 40 MHz	@ 10 MHz	@ 4 MHz
8 x 8 unsigned	Without hardware multiply	13	69	6.9 μs	27.6 μs	69 μs
	Hardware multiply	1	1	100 ns	400 ns	1 μs
8 x 8 signed	Without hardware multiply	33	91	9.1 μs	36.4 μs	91 μs
	Hardware multiply	6	6	600 ns	2.4 μs	6 μs
16 x 16 unsigned	Without hardware multiply	21	242	24.2 μs	96.8 μs	242 μs
	Hardware multiply	28	28	2.8 μs	11.2 μs	28 μs
16 x 16 signed	Without hardware multiply	52	254	25.4 μs	102.6 μs	254 μs
	Hardware multiply	35	40	4.0 μs	16.0 μs	40 μs

## 19.4 I<sup>2</sup>C Mode

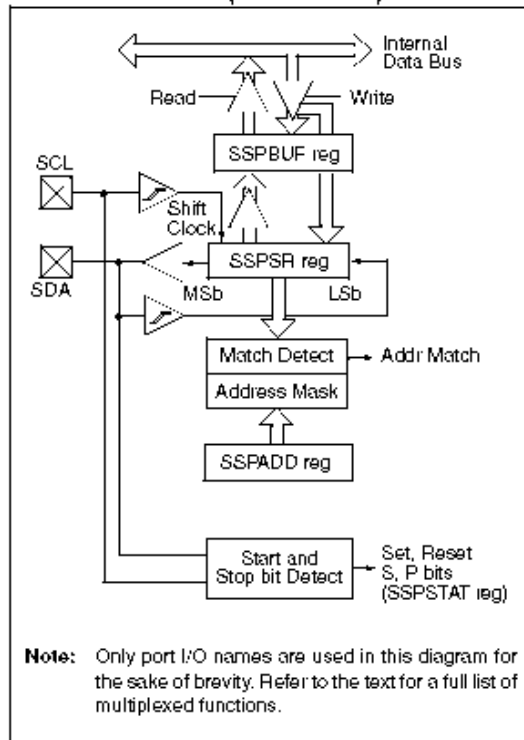
The MSSP module in I<sup>2</sup>C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer:

- Serial clock (SCL) – RB1/AN10/INT1/SCK/SCL
- Serial data (SDA) – RB0/AN12/INT0/FLT0/SDI/SDA

The user must configure these pins as inputs by setting the associated TRIS bits.

**FIGURE 19-7: MSSP BLOCK DIAGRAM (I<sup>2</sup>C™ MODE)**



### 19.4.1 REGISTERS

The MSSP module has six registers for I<sup>2</sup>C operation. These are:

- MSSP Control Register 1 (SSPCON1)
- MSSP Control Register 2 (SSPCON2)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible
- MSSP Address Register (SSPADD)

SSPCON1, SSPCON2 and SSPSTAT are the control and status registers in I<sup>2</sup>C mode operation. The SSPCON1 and SSPCON2 registers are readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

SSPADD register holds the slave device address when the MSSP is configured in I<sup>2</sup>C Slave mode. When the MSSP is configured in Master mode, the lower seven bits of SSPADD act as the Baud Rate Generator reload value.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

## 20.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of the two serial I/O modules. (Generically, the USART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN bus) systems.

The EUSART can be configured in the following modes:

- Asynchronous (full-duplex) with:
  - Auto-wake-up on Break signal
  - Auto-baud calibration
  - 12-bit Break character transmission
- Synchronous – Master (half-duplex) with selectable clock polarity
- Synchronous – Slave (half-duplex) with selectable clock polarity

The pins of the Enhanced USART are multiplexed with PORTC. In order to configure RC6/TX/CK and RC7/RX/DT/SDO as an EUSART:

- bit SPEN (RCSTA<7>) must be set (= 1)
- bit TRISC<7> must be set (= 1)
- bit TRISC<6> must be set (= 1)

<b>Note:</b> The EUSART control will automatically reconfigure the pin from input to output as needed.
--

The operation of the Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

These are detailed on the following pages in Register 20-1, Register 20-2 and Register 20-3, respectively.

**MOVFF                    Move f to f**

Syntax:                    MOVFF  $f_s, f_d$

Operands:                 $0 \leq f_s \leq 4095$   
 $0 \leq f_d \leq 4095$

Operation:                 $(f_s) \rightarrow f_d$

Status Affected:        None

Encoding:

1st word (source)	1100	ffff	ffff	ffff <sub>s</sub>
2nd word (destin.)	1111	ffff	ffff	ffff <sub>d</sub>

Description:

The contents of source register 'f<sub>s</sub>' are moved to destination register 'f<sub>d</sub>'. Location of source 'f<sub>s</sub>' can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination 'f<sub>d</sub>' can also be anywhere from 000h to FFFh. Either source or destination can be W (a useful special situation). **MOVFF** is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port). The **MOVFF** instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

Words:                    2

Cycles:                    2

Q Cycle Activity:

	Q1	Q2	Q3	Q4
	Decode	Read register 'f' (src)	Process Data	No operation
	Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example:                **MOVFF** REG1, REG2

Before Instruction  
REG1                    = 33h  
REG2                    = 11h

After Instruction  
REG1                    = 33h  
REG2                    = 33h

**MOVLB                    Move Literal to Low Nibble in BSR**

Syntax:                    MOVLW k

Operands:                 $0 \leq k \leq 255$

Operation:                 $k \rightarrow \text{BSR}$

Status Affected:        None

Encoding:

0000	0001	kkkk	kkkk
------	------	------	------

Description:

The eight-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '0' regardless of the value of k<sub>7:k<sub>4</sub></sub>.

Words:                    1

Cycles:                    1

Q Cycle Activity:

	Q1	Q2	Q3	Q4
	Decode	Read literal 'k'	Process Data	Write literal 'k' to BSR

Example:                **MOVLB**                5

Before Instruction  
BSR Register = 02h

After Instruction  
BSR Register = 05h

# ANEXO B – Hojas de datos del MAX232

MAX220-MAX249

## General Description

The MAX220-MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where  $\pm 12V$  is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5 $\mu$ W. The MAX225, MAX235, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

## Applications

- Portable Computers
- Low-Power Modems
- Interface Translation
- Battery-Powered RS-232 Systems
- Multidrop RS-232 Networks

## Features

### Superior to Bipolar

- ◆ Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)
- ◆ Low-Power Receive Mode in Shutdown (MAX223/MAX242)
- ◆ Meet All EIA/TIA-232E and V.28 Specifications
- ◆ Multiple Drivers and Receivers
- ◆ 3-State Driver and Receiver Outputs
- ◆ Open-Line Detection (MAX243)

## Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.  
\*Contact factory for dice specifications.

## Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value ( $\mu$ F)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.1	Nc	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	Nc	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	Nc	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	Nc	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	Nc	—	120	No external caps
MAX233A	+5	2/2	0	—	Nc	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	Nc	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	Nc	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	Nc	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	Nc	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	Nc	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	Nc	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/6	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package

**MAXIM**

Maxim Integrated Products 1



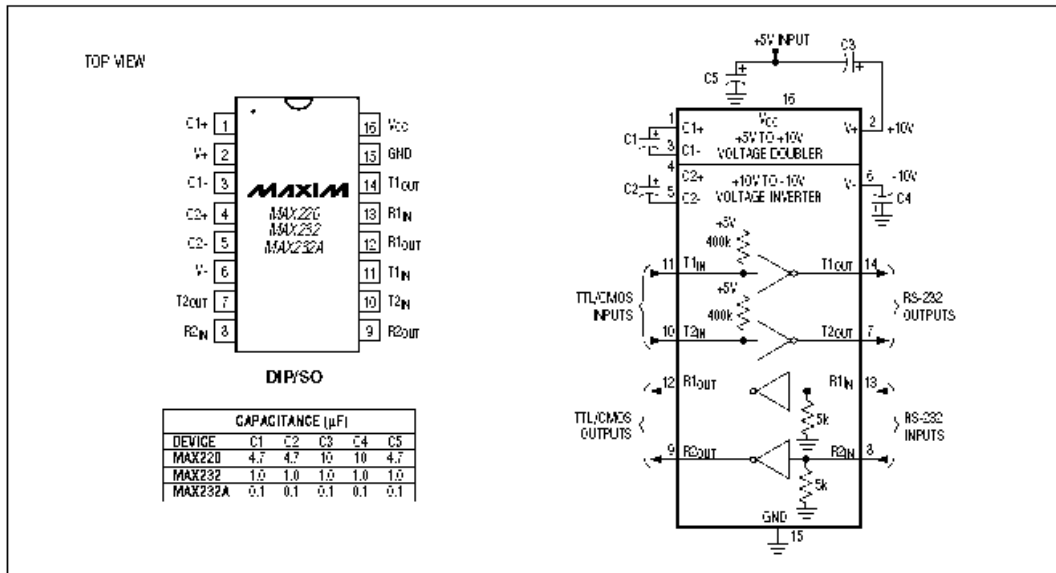


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

## ANEXO C – Inventario del hardware del dispositivo electrónico

INVENTARIO DE PARTES DEL PROTOTIPO DEL DISPOSITIVO ELECTRÓNICO				
Cantidad	Componente	Valor	Precio Unitario	Precio
2	Yardas de estaño p/soldar	-	\$ 0.2294	\$ 0.4588
1	R	330	\$ 0.2034	\$ 0.2034
1	R	2.2 K	\$ 0.2034	\$ 0.2034
7	R	10 K	\$ 0.2034	\$ 1.4238
2	C	17 ~ 32 pF	\$ 0.2260	\$ 0.4520
5	C	1 uF	\$ 0.1900	\$ 0.9500
1	D	1N4004	\$ 0.2034	\$ 0.2034
1	Q	2N2222	\$ 0.5900	\$ 0.5900
1	LED	Azul	\$ 0.3314	\$ 0.3314
1	Base para montar LED	-	\$ 0.2286	\$ 0.2286
1	XTAL	4 MHz	\$ 1.0848	\$ 1.0848
1	Fusible	1 Amp	\$ 0.1714	\$ 0.1714
1	Par de bases para fusible	-	\$ 0.0800	\$ 0.0800
1	Toggle Switch	-	\$ 0.3200	\$ 0.3200
4	Push Button	2 Patitas	\$ 0.4500	\$ 1.8000
1	Plug DC (macho)	-	\$ 0.5700	\$ 0.5700
1	Jack DC (hembra)	-	\$ 0.3500	\$ 0.3500
1	Fuente de voltaje	5V DC	\$ 5.4100	\$ 5.4100
1	Jack estéreo (hembra)	3.5 mm con switch	\$ 1.0170	\$ 1.0170
1	Cable estéreo 3.5 mm	-	\$ 0.5000	\$ 0.5000
1	Plug estéreo de 3.5 mm (macho)	-	\$ 0.2514	\$ 0.2514
1	Plug estéreo de 2.5 mm (macho)	-	\$ 0.4513	\$ 0.4513
1	Relay Tiambo	TRA3 L-05VDC-S-2Z	\$ 1.8532	\$ 1.8532
1	Pin Head	1X02	Chatarra	\$ -
1	Cable 2 líneas c/conectores para Pin Head 1X02	-	Chatarra	\$ -
2	MA05-2 (Pin Head)	MA05-2 (5X02)	Chatarra	\$ -
2	Ribbon cable	5X02	\$ 2.5000	\$ 5.0000
1	Cable/conector de Datos/Poder p/Floppy	-	Chatarra	\$ -
1	Metro de cable p/botones y switch	-	Chatarra	\$ -
1	Base para integrado	16 Pines	\$ 0.2900	\$ 0.2900
1	ZIF Socket	40 Patas	\$ 10.0000	\$ 10.0000
2	Teclados para teléfono	-	\$ 1.1700	\$ 2.3400
1	Circuito Impreso p/núcleo	-	\$ 28.0000	\$ 28.0000
1	Circuito Impreso p/botones	-	\$ 2.5000	\$ 2.5000
1	Caja p/dispositivo electrónico	-	\$ 75.0000	\$ 75.0000
-	Tornillos p/dispositivo electrónico	-	Chatarra	\$ -
1	IC Convertidor RS232 - TTL	MAX232 DIL16	\$ 3.4500	\$ 3.4500
1	Microcontrolador de Microchip	PIC18F4550-I/P	\$ 6.7200	\$ 6.7200
1	Display LCD Matrix Orbital	LK162-12	\$ 49.9500	\$ 49.9500
1	Embeded MODEM Radicom	MB-R-5V	\$ 99.9500	\$ 100.0000
			<b>TOTAL =</b>	<b>\$ 302.1539</b>