

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica



Organización de la información generada por una red inalámbrica de sensores:

Load Point CRTECMOTE

**Informe de Proyecto de Graduación para optar por el título de Ingeniería
Electrónica con el grado académico de Licenciatura**

Dennis A. Rodríguez R.

Cartago, enero de 2010

INSTITUTO TECNOLÓGICO DE COSTA RICA
ESCUELA DE INGENIERÍA ELECTRÓNICA
PROYECTO DE GRADUACIÓN
TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



Ing. Johan Carvajal Godínez

Profesor asesor



Ing. Anibal Coto Cortés

Profesor lector



Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 27 Enero 2010

Declaratoria de autenticidad

Declaro que el presente informe ha sido realizado por mi persona, utilizando y aplicando literatura referente al tema, así como la información que haya suministrado la empresa para la que se realizará el proyecto, y aplicando e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad por el contenido de este proyecto.

San José, 27 de enero de 2010



Dennis A. Rodríguez Rodríguez

Cédula: 1-1220-0751

Resumen

Este documento describe los aspectos más relevantes para centralizar la información generada por una red inalámbrica de sensores de mediana escala construida por la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica. Para lograr el objetivo principal se diseñó el modelo de estructuración de datos y las rutinas en software para demostrar el funcionamiento del modelo en un punto de carga (Load Point por su nombre en inglés). Además se muestra el detalle de las rutinas en software para el control de un dispositivo GSM para el acceso inalámbrico de un nodo de carga.

Load Point CRTECMOTE es un sistema que funciona a modo de nodo central en la red inalámbrica de sensores; organiza y almacena la información proveniente de la red siguiendo un modelo de estructura de datos que se ejecuta sobre un sistema gestor de bases de datos MySQL. Para obtener los datos se conecta al medio inalámbrico mediante un dispositivo o módem GSM. El presente sistema da la posibilidad de presentar a una persona la información recolectada del ambiente donde se encuentren en funcionamiento los nodos distribuidos.

Palabras clave: Red Inalámbrica de Sensores (WSN), nodo central, Sistema Global para Móviles (GSM), modem GSM, Servicio de Mensajes Cortos (SMS), MySQL, procedimientos almacenados, Lenguaje C#, .NET Framework.

Abstract

This document describes the most important aspects to centralize the information generated by a wireless sensor network of medium-built by the Electronic Engineering School at the Costa Rica Institute of Technology. To achieve the main objective a data structure and software routines were designed to demonstrate the functionality of the model in a load point. Also shows the details of software routines to control a GSM device in order to give wireless access to a load point.

Load Point CRTECMOTE is a system that functions as a central node in the wireless sensor network, organizes and store information from the network using a data structure model that runs on a MySQL database management system. To obtain the data it is connected to the wireless medium using a GSM device or modem. This system makes it possible to present a user the information collected from the environment where the distributed nodes are running.

Keywords: Wireless Sensor Network (WSN), sink node, Global System for Mobile (GSM), GSM modem, Short Message Service (SMS), MySQL, stored procedures, C# Language, .NET Framework.

INDICE GENERAL

1	Capítulo 1: Introducción.....	1
1.1	Problema existente e importancia de su solución	1
1.1.1	Definición del problema	1
1.1.2	Síntesis del problema	2
1.1.3	Importancia de la solución	2
1.2	Solución seleccionada	3
1.2.1	Requerimientos y restricciones del sistema	3
1.2.2	Síntesis de la solución seleccionada	4
2	Capítulo 2: Meta y Objetivos.....	6
2.1	Meta	6
2.2	Objetivo general	6
2.3	Objetivos específicos	6
3	Capítulo 3: Marco teórico	7
3.1	Sistema global para las comunicaciones móviles (GSM).....	7
3.1.1	Red de radio celular GSM	8
3.1.2	Radio frecuencias en GSM.....	8
3.1.3	Arquitectura de la red GSM	10
3.1.3.1	Sistema de conmutación	11
3.1.3.2	Sistema base-estación	12
3.1.3.3	Sistema de Operación y Soporte	12
3.1.3.4	Elemento funcional para mensajes cortos	13
3.1.4	Servicio de mensajes cortos (SMS).....	14
3.2	Arquitectura del dispositivo inalámbrico para GSM.....	15
3.2.1	Diagrama de bloques de un modem GSM.....	15
3.2.2	Comandos de Hayes.....	17
3.2.2.1	Registros de configuración (Setting-Registers).....	18
3.2.2.2	Comandos de Hayes para enviar y recibir mensajes cortos.....	18
3.3	Bases de Datos digitales	20
3.3.1	Tipos de bases de datos	21
3.3.2	Modelos de bases de datos.....	21
3.3.3	Sistema de gestión de bases de datos	22
3.3.3.1	Características de un sistema de gestión de bases de datos.....	22
3.3.3.2	Sistema de gestión MySQL	23
3.3.3.3	Motor de almacenamiento InnoDB	23
3.3.3.4	Procedimientos almacenados.....	24
3.4	Antecedentes bibliográficos.....	25
4	Capítulo 4: Procedimiento Metodológico.....	26
4.1	Métodos y actividades	26
4.1.1	Seleccionar el hardware de acceso inalámbrico	26
4.1.2	Determinación del lenguaje de programación a utilizar.....	26
4.1.3	Verificación del funcionamiento de una estación móvil como módem GSM mediante los comandos de Hayes	27
4.1.4	Desarrollo de un modelo de base de datos	27
4.1.5	Desarrollo de un protocolo de comunicaciones	27
4.1.6	Desarrollo de una rutina para envío de mensajes.....	27

4.1.7	Desarrollo de una rutina para recepción de mensajes cortos.....	28
4.1.8	Desarrollo de las rutinas para crear las tablas en el SGBD relacional	28
4.1.9	Desarrollo de las rutinas para crear los procedimientos almacenados en el SGBD relacional	28
4.1.10	Desarrollo de los programas que implementan cada una de las rutinas con el lenguaje de programación	29
5	Capítulo 5: Descripción detallada de la solución.....	30
5.1	Descripción general.....	30
5.2	Diseño de un modelo para la organización de la información	31
5.2.1	Modelo de base de datos relacional	31
5.2.2	Definición de las tuplas en el diseño del modelo relacional	32
5.3	Puesta en marcha de una base de datos relacional MySQL.....	34
5.3.1	Requerimientos para el funcionamiento del SGBD relacional MySQL	34
5.3.2	Instalación del SGBD MySQL.....	34
5.3.3	Creación del esquema de la base de datos (<i>schema</i>) en el SGBD relacional MySQL	36
5.3.4	Creación de una tabla en el SGBD.....	37
5.3.5	Diseño de los procedimientos almacenados en el sistema de gestión de bases de datos	38
5.3.5.1	Procedimiento almacenado para mostrar una tabla específica	38
5.3.5.2	Procedimiento almacenado para insertar en una tabla específica	39
5.3.5.3	Procedimiento almacenado para actualizar la fila en una tabla.....	40
5.3.5.4	Procedimiento almacenado para borrar la fila en una tabla	40
5.4	Diseño del protocolo de comunicaciones.....	42
5.4.1	Definición de la estructura de la información	42
5.4.2	Definición de la integridad de los datos	43
5.4.3	Diagrama de flujo del protocolo de comunicaciones	44
5.5	Diseño del software para el control del hardware inalámbrico GSM	45
5.5.1	Verificación de funcionamiento del hardware inalámbrico GSM.....	45
5.5.1.1	Selección del dispositivo inalámbrico GSM.....	45
5.5.1.2	Verificación de las operaciones de los comandos AT sobre el dispositivo GSM seleccionado	47
5.5.2	Rutina de inicialización del dispositivo GSM.....	47
5.5.3	Rutina de envío de información	49
5.5.4	Rutina de recepción de información	50
5.5.5	Selección del lenguaje de programación de alto nivel	51
5.6	Diseño de pruebas de funcionamiento del sistema de organización de información	52
5.6.1	Pruebas de estructura del modelo de base de datos relacional	52
5.6.2	Prueba de bombardeo de mensajes.....	55
5.6.3	Prueba de integridad de la información	57
5.6.4	Prueba para medir el retraso en la entrega de la información.....	57
6	Capítulo 6: Análisis de resultados.....	59
6.1	Resultados	60
6.1.1	Verificación de la creación del esquema (<i>schema</i>) de una base de datos en el SGBD MySQL	60
6.1.2	Resultados de la creación de tablas en el SGBD de MySQL	60
6.1.3	Resultados de la creación de los procedimientos almacenados en un SGBD de MySQL	63
6.1.4	Resultados de la comprobación de la estructura del modelo relacional diseñado	66
6.1.5	Resultados de la selección del dispositivo inalámbrico GSM	68
6.1.6	Resultados de la implementación de las rutinas de inicialización, envío y recepción de información	70
6.1.7	Resultados del funcionamiento del sistema	71
6.2	Análisis	82
7	Capítulo 7: Conclusiones y recomendaciones	89

7.1	Conclusiones.....	89
7.2	Recomendaciones.....	90
8	Bibliografía y referencias.....	91
9	Apéndices	94
A.1	Abreviaturas	94

INDICE DE FIGURAS

Figura 1.1	Representación del principio de diseño jerárquico para una WSN.....	2
Figura 1.2	Esquema general de la solución seleccionada.....	5
Figura 3.1	Esquema del modelo de División del Tiempo/ Múltiple Acceso.....	10
Figura 3.2	Arquitectura de un sistema global para móviles.....	10
Figura 3.3	Centro de mensajes cortos (SMSC) en la arquitectura GSM.....	13
Figura 3.4	Diagrama de bloques de un dispositivo GSM.....	16
Figura 3.5	Comunicación con un módem GSM.....	19
Figura 3.6	Procedimiento almacenado en MySQL.....	24
Figura 5.1	Modelo de la base de datos relacional para el Load Point CRTECMOTE.....	30
Figura 5.2	Modelo de la base de datos relacional para el Load Point CRTECMOTE.....	32
Figura 5.3	Pantalla tomada del instalador de MySQL para escoger el tipo de gestor.....	35
Figura 5.4	Pantalla tomada del instalador de MySQL para escoger el tipo de base de datos.....	35
Figura 5.5	Pantalla tomada del instalador de MySQL para otorgar seguridad al SGBD.....	36
Figura 5.6	Ventana de comandos del SGBD MySQL 5.0 para crear una base de datos.....	36
Figura 5.7	Diagrama de flujo para crear una tabla en el modelo relacional.....	37
Figura 5.8	Diagrama de flujo de un procedimiento almacenado para mostrar el contenido de una tabla en MySQL.....	39
Figura 5.9	Diagrama de flujo de un procedimiento almacenado para insertar una nueva entrada en una tabla en MySQL.....	39
Figura 5.10	Diagrama de flujo de un procedimiento almacenado para actualizar una fila en una tabla en MySQL.....	40
Figura 5.11	Diagrama de flujo de un procedimiento almacenado para borrar una fila en una tabla en MySQL.....	41
Figura 5.12	Diseño básico del paquete de datos en el protocolo de comunicaciones.....	42
Figura 5.13	Encapsulado del paquete de datos con el código "hash.....	43
Figura 5.14	Diagrama de flujo del protocolo de comunicaciones diseñado para el sistema de organización de información.....	44
Figura 5.15	Diagrama de flujo para seleccionar un dispositivo GSM.....	46
Figura 5.16	Diagrama de flujo para inicializar el dispositivo GSM.....	48
Figura 5.17	Diagrama de flujo para enviar un mensaje de texto con un dispositivo GSM.....	49
Figura 5.18	Diagrama de flujo para recibir y almacenar información con un dispositivo GSM.....	50
Figura 5.19	Interfaz gráfica para administrar la base de datos relacional.....	53
Figura 5.20	Diagrama de flujo para verificar la estructura del modelo de base de datos relacional.....	55
Figura 5.21	Interfaz gráfica para simular el envío de datos desde un nodo líder.....	56
Figura 5.22	Diagrama de flujo para el envío de mensajes en modo masivo.....	56
Figura 6.1	Verificación de la creación del esquema de una base de datos en el SGBD de MySQL.....	60
Figura 6.3	Fragmento de código para crear la tabla "Ubicaciones" en el SGBD.....	61
Figura 6.4	Fragmento de código para crear la tabla "Unidad" en el SGBD.....	61
Figura 6.5	Fragmento de código SQL para crear la tabla "Nodo_líder" en el SGBD.....	61
Figura 6.6	Fragmento de código en SQL para crear la tabla "Tip_variable" en el SGBD.....	62
Figura 6.7	Fragmento de código SQL para crear las tablas "Nodo" y "Datos" en el SGBD.....	62
Figura 6.8	Resultado de la creación de las tablas del modelo relacional en el SGBD de MySQL.....	63
Figura 6.9	Fragmento de código del procedimiento almacenado para mostrar el contenido de una tabla en MySQL.....	63
Figura 6.10	Resultado de llamar el procedimiento almacenado para mostrar el contenido de una tabla en MySQL.....	64
Figura 6.11	Procedimiento almacenado que inserta una nueva entrada en una tabla en MySQL.....	64
Figura 6.12	Resultado de llamar el procedimiento almacenado para insertar una fila de una tabla en MySQL.....	65
Figura 6.13	Fragmento de código del procedimiento almacenado para actualizar una fila en una tabla en MySQL.....	65
Figura 6.14	Resultado de llamar el procedimiento almacenado para actualizar una fila de una tabla en MySQL.....	65
Figura 6.15	Fragmento de código del procedimiento almacenado para borrar una fila en una tabla en MySQL.....	66
Figura 6.16	Resultado de llamar el procedimiento almacenado para borrar una fila de una tabla en MySQL.....	66

Figura 6.17	Resultado de intentar borrar una fila en una tabla padre de la que depende una tabla hijo.....	67
Figura 6.18	Resultado de intentar insertar o actualizar una fila en una tabla padre de la que depende una tabla hijo	67
Figura 6.19	Características de los dispositivos Motorola L7 y Nokia 5310	68
Figura 6.20	Comandos AT de verificación y configuración del funcionamiento de un Motorola L7.....	69
Figura 6.21	Comandos AT de verificación y configuración del funcionamiento de un Nokia 5310.....	69
Figura 6.22	Fragmento de código para la inicialización del dispositivo GSM	70
Figura 6.23	Fragmento de código para enviar información con el dispositivo GSM	70
Figura 6.24	Fragmento de código para recibir información con el dispositivo GSM	71

ÍNDICE DE TABLAS

Tabla 3.1 Resumen de radio frecuencias en GSM.....	9
Tabla 3.2 Algunos registros de configuración de un dispositivo GSM.....	18
Tabla 3.3 Comandos AT extendidos para mensajes cortos	19
Tabla 6.1 Tiempo de retardo y resultado de la comparación del hash de 50 paquetes de información enviados el 4 de Enero del 2010 a las 6:30 am.....	73
Tabla 6.2 Valores para el cálculo de la desviación estándar del tiempo de retardo para la prueba del 04 de Enero del 2010 a las 6:30 am.	74
Tabla 6.3 Resumen de los resultados del análisis de los datos recibidos en el nodo central el 04 de Enero del 2010 a las 6:30 am	75
Tabla 6.4 Tiempo de retardo y resultado de la comparación del hash de 50 paquetes de información enviados el 04 de Enero del 2010 a las 12:10 pm.....	76
Tabla 6.5 Valores para el cálculo de la desviación estándar del tiempo de retardo para la prueba del 04 de Enero del 2010 a las 12:10 pm.	77
Tabla 6.6 Resumen de los resultados del análisis de los datos recibidos en el nodo central el 04 de Enero del 2010 a las 12:10 am.....	78
Tabla 6.7 Tiempo de retardo y resultado de la comparación del hash de 50 paquetes de información enviados el 4 de Enero del 2010 a las 05:00 pm.....	79
Tabla 6.8 Valores para el cálculo de la desviación estándar del tiempo de retardo para la prueba del 04 de Enero del 2010 a las 05:00 pm.	80
Tabla 6.9 Resumen de los resultados del análisis de los datos recibidos en el nodo central el 04 de Enero del 2010 a las 05:00 pm.....	81
Tabla 6.10 Resumen del promedio general del tiempo de retardo y desviación general de un total de 150 mensajes recibidos el 04 de Enero del 2010.....	81

1 Capítulo 1: Introducción

Se introduce en el presente capítulo el problema existente y la importancia de resolver dicho problema. Además se describen los requerimientos y las restricciones del proyecto.

1.1 Problema existente e importancia de su solución

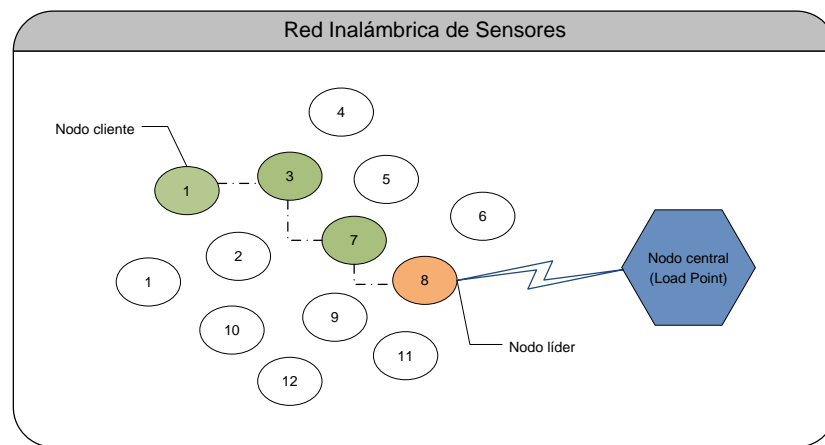
1.1.1 Definición del problema

La Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica (ITCR) es un centro académico a nivel superior que se encarga de la docencia e investigación en el campo de la ingeniería electrónica en sus diferentes campos de aplicación: sistemas digitales, control automático, comunicaciones eléctricas, electrónica de potencia, entre otras [2].

Dicha escuela tiene entre sus planes la creación de una red inalámbrica de sensores para aplicaciones principalmente ambientales. La idea es que la arquitectura de dicha red sea abierta y pueda ser usada libremente por terceras personas [1].

La composición de una red inalámbrica de sensores (WSN) es de elementos llamados nodos [3]. Según el principio de diseño para interconectar los nodos se utilizan diferentes nombres y esto categoriza éste tipo de redes. El principio de diseño para la red de sensores del ITCR es el “jerárquico” donde los nodos se dividen según su jerarquía en: nodo cliente, el cual obtiene la información, nodo líder o nodo de salida que toma los datos y los envía a un nodo central o punto de carga que finalmente centraliza la información obtenida como se muestra en la Fig. 1.1. Los nodos tipo cliente y líder están siendo construidos por otros colaboradores del proyecto [1].

El presente proyecto se inscribe en la problemática de carecer en la infraestructura de un nodo central (Load Point). No se ha definido el hardware que permita el acceso al medio inalámbrico, no se posee las rutinas en software para controlar dicho hardware; mucho menos existe una estructura para organizar y almacenar la información proveniente de un nodo líder [1]. No hay una rutina que simule el envío de información hacia el punto de carga en la red inalámbrica del ITCR. De modo general, no se cuenta con las rutinas necesarias para el funcionamiento básico del sistema.



Microsoft Office Visio 2007

Figura 1.1 Representación del principio de diseño jerárquico para una WSN

1.1.2 Síntesis del problema

Carencia en la infraestructura de la Red Inalámbrica de Sensores de la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica de un nodo central (Load Point).

1.1.3 Importancia de la solución

La importancia de resolver estas carencias radica en la posibilidad de completar una parte importante de la red inalámbrica de sensores para la Escuela de Ingeniería Electrónica del ITCR, así como una pronta puesta en marcha para el inicio de la pruebas de validación de los nodos sensores. Entre otras cosas, el desarrollo

de éste proyecto permite explorar la utilidad de enviar información de forma inalámbrica desde un nodo líder al punto de carga para organizar y almacenar los datos capturados por los nodos de la WSN. De ésta forma se obtiene como rédito la habilidad de manipular estos datos para su posterior análisis.

- ✓ Al cumplir con el objetivo general, los futuros diseñadores de la red podrán utilizar un sistema básico para centralizar, organizar y almacenar los datos de la red inalámbrica de sensores.
- ✓ Los diseñadores de la etapa de la red inalámbrica pueden validar el trasiego de información utilizando esta primera etapa del sistema de organización.
- ✓ Los futuros diseñadores podrán utilizar las rutinas de software para el control del hardware inalámbrico utilizado en éste sistema.

1.2 Solución seleccionada

1.2.1 Requerimientos y restricciones del sistema

Los principales requerimientos para la organización de la información generada por una red inalámbrica de sensores son:

- ✓ Seleccionar el hardware necesario para proporcionar acceso al medio inalámbrico en el nodo central.
- ✓ Desarrollar los programas o bibliotecas necesarias para controlar el hardware de acceso al medio inalámbrico seleccionado.
- ✓ Desarrollar un modelo de almacenamiento y estructuración de datos.
- ✓ Desarrollar los programas o bibliotecas necesarias para implementar los modelos de almacenamiento y estructuración de datos.

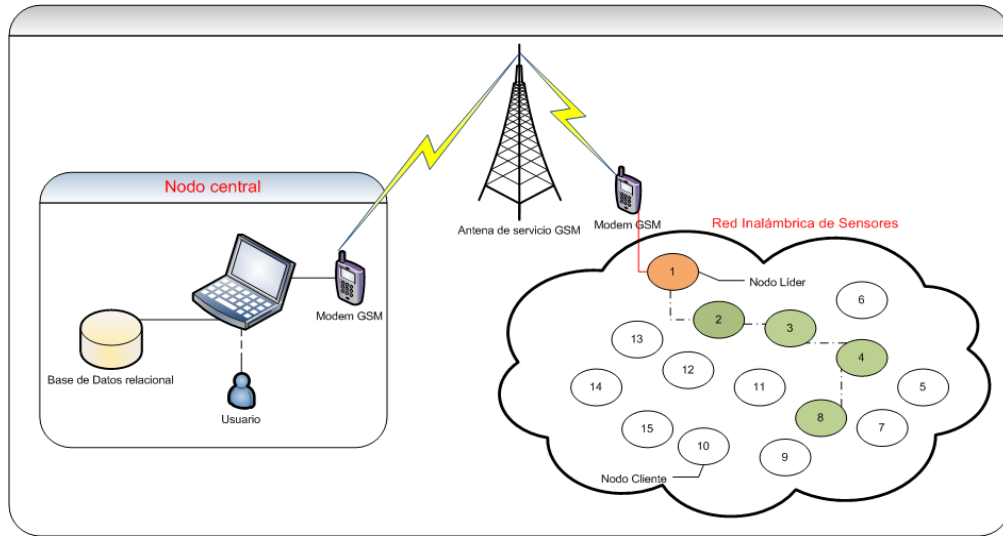
Las restricciones que se establecieron para el presente proyecto son las siguientes:

- ✓ El sistema Load Point CRTECMOTE debe de ser capaz de recibir la información de forma inalámbrica desde un nodo líder.
- ✓ El sistema debe de simular la rutina de envío de información desde un nodo líder a un nodo central.
- ✓ El sistema debe ser escalable.
- ✓ Se debe de establecer un protocolo o estructura de datos para tipificar el dato transmitido.

1.2.2 Síntesis de la solución seleccionada

La Fig. 1.2 detalla el esquema general del sistema que se desarrolló para la solución a la carencia en la infraestructura de un nodo central. Con el fin de cumplir con el objetivo general se definió el hardware para el nodo central como un sistema computacional con un sistema operativo capaz de soportar el sistema gestor de base de datos MySQL. Sobre el sistema gestor se implementó el diseño de un modelo relacional de almacenamiento y estructuración de información.

El hardware de control de acceso al medio inalámbrico seleccionado es un teléfono celular funcionando como módem con capacidad de conexión a la popular red inalámbrica conocida como Sistema Global para Móviles (GSM). El uso del módem se basó en el Servicio de Mensajes Cortos (SMS) para enviar información de un punto a otro sobre el sistema global para comunicaciones móviles. El módem se conectó a un puerto del bus serial universal en el sistema computacional.



Microsoft Office Visio 2007

Figura 1.2 Esquema general de la solución seleccionada.

Para cumplir a cabalidad con el desarrollo de la solución del sistema, se diseñó las rutinas en software para la aplicación del modelo de almacenamiento y estructuración de datos en el sistema gestor de bases de datos MySQL. Además, se diseñó las rutinas en software para controlar el dispositivo GSM seleccionado. Para implementar dichas rutinas, se utilizó un ambiente de desarrollo de software en el lenguaje de alto nivel C#.

Finalmente, para demostrar el funcionamiento total del sistema de organización de información se desarrolló las interfaces de usuario para administrar la base de datos y para simular el envío de datos desde un nodo líder.

2 Capítulo 2: Meta y Objetivos

2.1 Meta

Lograr un sistema que organice los datos generados por una red inalámbrica de sensores de mediana escala.

2.2 Objetivo general

Centralizar la información generada por una red inalámbrica de sensores de mediana escala construida por la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica.

2.3 Objetivos específicos

- ✓ Diseñar un modelo de organización y estructuración de datos para el almacenamiento de la información obtenida de una red inalámbrica de sensores.
- ✓ Diseñar un protocolo de comunicaciones que permita a un nodo líder intercambiar información con un nodo central.
- ✓ Sintetizar las bibliotecas de software requeridas para conectar la tecnología de acceso al medio y el sistema de estructuración de datos.
- ✓ Medir los parámetros críticos de operación del sistema, según la tecnología seleccionada, ante estímulos que simulen la operación de la red inalámbrica de sensores.

3 Capítulo 3: Marco teórico

El siguiente capítulo describe los conceptos más relevantes utilizados en el desarrollo de un sistema de organización de datos para una red inalámbrica de sensores.

3.1 Sistema global para las comunicaciones móviles (GSM)

El sistema global para comunicaciones móviles es un estándar que especifica las principales regulaciones para la comunicación entre estaciones móviles. El conjunto de reglas que conforman el sistema es definido por el Instituto Europeo de Estándares en Telecomunicaciones (ETSI) en su documento ETSI TS 151 010-1 llamado "*Digital cellular telecommunications system (Phase 2+): Mobile Station (MS) conformance specification*" [8]. Éste es el estándar más popular para teléfonos móviles en el mundo.

El estándar define recomendaciones para la comunicación digital entre estaciones móviles. Las especificaciones de dicho estándar establecen las funciones y los requisitos de interfaz en detalle, pero no hace definiciones en cuanto a hardware. La razón de esto es para no limitar a los diseñadores y dar así un margen para que los operadores del servicio puedan comprar equipos de diferentes proveedores.

El sistema global para móviles revolucionó la tecnología predecesora (analógica) estableciendo la transmisión de voz y datos en formato digital [6]. Éste cambio es considerado como la segunda generación o segunda fase (2G) en los sistemas de comunicación móviles, lo que hizo posible bajar el costo de las redes de operación aumentando la capacidad para cubrir la demanda del servicio, bajar los niveles de interferencia, integrar la transmisión y conmutación de la información.

El estándar ha incursionado con alternativas más económicas a las llamadas de voz, introduciendo el servicio de mensajes cortos de texto y en sus últimas

publicaciones se han agregado las capacidades en la transmisión y recepción de paquetes de datos bajo el Servicio General de Paquetes por Radio (GPRS) así como el envío de información a altas velocidades utilizando la Transmisión de Datos Mejorada para la Evolución de GSM (EDGE) como preámbulo al Sistema Universal de Telecomunicaciones Móviles (UMTS) o tercera generación (3G) [8].

3.1.1 Red de radio celular GSM

Las estaciones móviles en un sistema global para comunicaciones forman parte de una red y se conectan a dicha red buscando por celdas de radio frecuencia en la vecindad. En una red GSM existen diferentes tamaños de celdas que se definen según el área de cobertura. Los cinco tamaños de celdas son macro, micro, pico, femto y sombrilla. Una celda macro es la que provee una antena colocada en lo alto de un edificio, una celda micro se produce por una antena colocada en urbanizaciones de mediana escala, las pico-celdas son debido a antenas colocadas dentro de edificios, las femto-celdas se deben a las antenas para zonas residenciales y se usan para conectar a servicios de banda ancha en conexiones de internet; por último, las celdas sombrilla se producen debido a antenas para cubrir áreas de puntos muertos del servicio. El radio horizontal de cada celda depende de la altura de la antena, su ganancia y las condiciones de propagación. El sistema de antenas y celdas es el que permite la gran área de cobertura del sistema global para móviles. En el caso de Costa Rica el área de cobertura es casi total, con problemas de puntos muertos por lo escarpado de la geografía.

3.1.2 Radio frecuencias en GSM

La red del sistema global para móviles opera en diferentes rangos de frecuencia. Hoy en día se separan los rangos de frecuencia para GSM de segunda generación y para UMTS de tercera generación. La mayoría de las redes GSM operan en las bandas de 900 MHz o 1800 MHz. Los valores de Tabla 3.1 corresponden a las bandas de frecuencias comúnmente usadas en diferentes continentes. La banda GSM-900 es la más usada debido a los inicios de la tecnología en Europa y usa 890-915 MHz para enviar información desde una

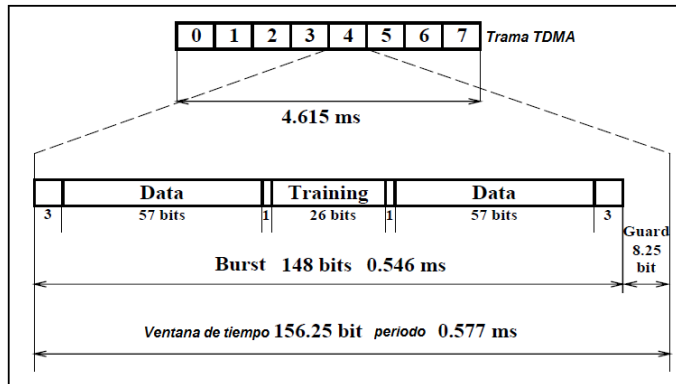
estación móvil a una base estación (enlace de carga) y 935-960 MHz de forma inversa (enlace de descarga); posee 124 canales (numerados desde 1 hasta 124) espaciados por 200 kHz cada uno para evitar solapamiento. La distancia dúplex (diferencia en frecuencia entre el enlace de carga y descarga) es de 45 MHz [5].

En algunos países la banda GSM-900 se ha extendido para cubrir un mayor rango de frecuencias. La E-GSM (banda extendida) usa 880-915 MHz de enlace de carga y 925-960 MHz de enlace de descarga, otorgando 50 canales más a GSM-900 numerados desde 975 hasta 1023 y 0 inclusive [5].

Tabla 3.1 Resumen de radio frecuencias en GSM [7]

Banda	Nombre	Canales	Enlace de carga (MHz)	Enlace de descarga (MHz)	Notas
GSM 850	GSM 85	128-251	824.0-849.0	869.0-894.0	Se usa en Norteamérica, Sudamérica y Asia.
GSM 900	P-GSM 900	1-124	890.0-915.0	935.0-960.0	Se usa en Europa y es la más extendida.
	E-GSM 900	975-1023	880.0-915.0	925.0-960.0	Extensión GSM-900
	R-GSM 900	n/a	876.0-880.0	921.0-925.0	GSM ferroviario
GSM 1800	GSM 1800	512-885	1710.0-1785.0	1805.0-1880.0	-
GSM 1900	GSM 1900	512-810	1850.0-1910.0	1930.0-1990.0	En EEUU, solapa con GSM-1800

A cada canal se le aplica el método de acceso para la multiplexación de división del tiempo (TDMA) lo que permite obtener por cada canal de radio frecuencia ocho canales de voz en ambos sentidos (full dúplex) o dieciséis en un solo sentido (half dúplex). Específicamente TDMA permite dividir un tiempo de 4.61 ms en ocho fracciones de 577 μ s (ver Fig. 3.1). De ésta forma, ocho estaciones móviles pueden compartir un mismo canal de frecuencia. En un tiempo de 4.61 ms, cada dispositivo toma el mando sobre el canal solamente 577 μ s. Ocho espacios de 577 μ s conforman una trama de tiempo en la que ocho estaciones móviles pueden enviar sus datos [5].

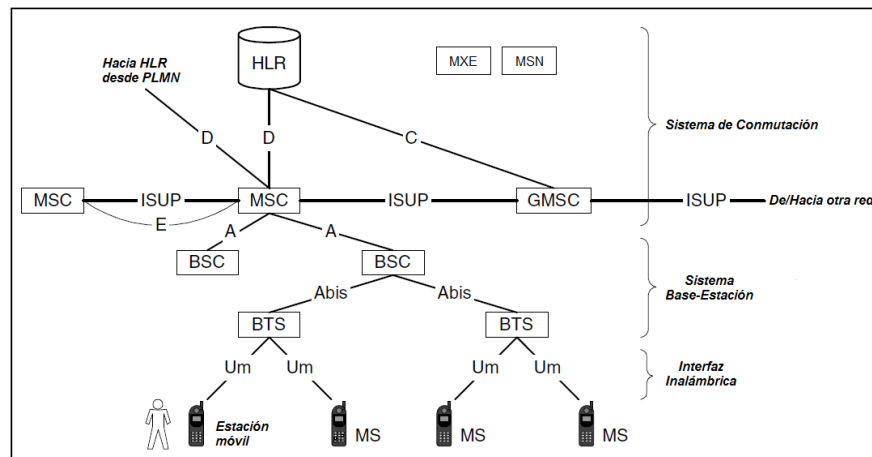


Microsoft Paint 6.0

Figura 3.1 Esquema del modelo de División del Tiempo/ Múltiple Acceso [5].

3.1.3 Arquitectura de la red GSM

La arquitectura de la red del sistema global para comunicaciones móviles se divide en tres grandes sistemas: el Sistema de Conmutación (SS), el Sistema de Base-Estación (BSS) y el Sistema de Operación-Soporte (OSS) [7]. La Fig. 3.2 muestra el diagrama de bloques de una arquitectura de la red GSM [6].



Microsoft Paint 6.0

Figura 3.2 Arquitectura de un sistema global para móviles [6].

3.1.3.1 Sistema de conmutación

El sistema de conmutación es el responsable de ejecutar el proceso de la llamada y realizar las funciones propias del operador. Sus unidades funcionales se muestran en los bloques de la Fig. 3.2.

- ✓ **Registro de localización (HLR):** es una base de datos que se usa para almacenamiento y el manejo de las suscripciones de los usuarios. Es considerada la base de datos más importante ya que almacena las suscripciones de los usuarios de forma permanente, incluyendo el perfil del servicio del suscriptor, la información de su ubicación y el estado de su actividad. Cuando un individuo compra una suscripción, la persona es registrada en el HLR.
- ✓ **Centro de servicios de conmutación para móviles (MSC):** ejecuta la función de conmutación del servicio de telefonía móvil. Controla las llamadas hacia y desde otros teléfonos o sistemas de datos.
- ✓ **Registro de localización del visitante (VLR):** es una base de datos que contiene información temporal de los suscriptores de otros operadores que funcionan en un momento dado en la red de otro operador de servicios. Normalmente el VLR está integrado en el MSC. Cuando una estación móvil usa la red de otro operador el MSC pide la información al HLR y cuando ésta estación requiere hacer otra llamada el VLR ya posee la información necesaria del suscriptor para establecer la conexión.
- ✓ **Centro de autenticación (AUC):** esta unidad provee de autenticación y encriptación de parámetros que verifica la identidad del usuario para asegurar confidencialidad en cada llamada. Protege además a los operadores de la red de los fraudes hoy en día.

- ✓ **Registro de identificación de equipo (EIR):** es una base de datos que contiene la identidad de un equipo móvil para así prevenir el robo de llamadas. La AUC y la EIR pueden ser unidades separadas o estar juntas como AUC/EIR.

3.1.3.2 Sistema base-estación

Todas las funciones relacionadas con radio frecuencia se realizan en este módulo. Se tiene básicamente dos elementos, el Control de la Base-Estación (BCS) y la Estación Base-Transductor (BTS).

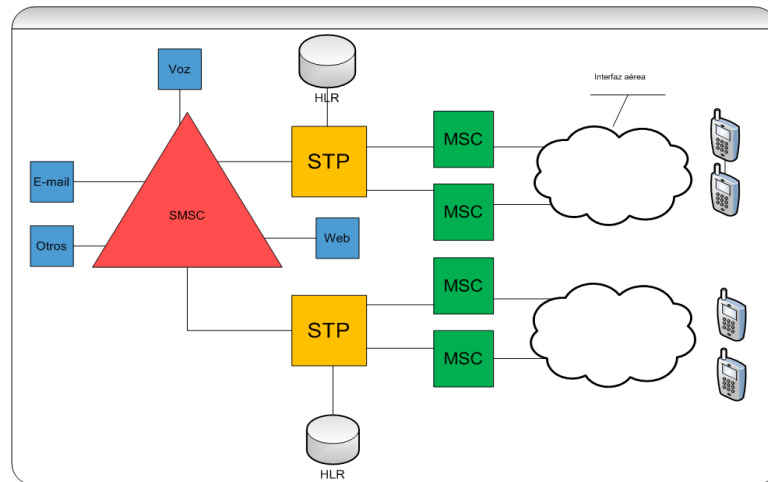
- ✓ **Controlador Base-Estación (BSC):** provee las funciones de control de los enlaces físicos entre el módulo MSC y BTS. Es un conmutador con gran capacidad de procesamiento que tiene la función de configurar la información del dispositivo móvil y controlar los niveles de radio frecuencia en las estaciones Base-Transductores.
- ✓ **Estación Base-Transductor (BTS):** maneja la interfaz de radio con la estación móvil. Las BTS es todo el equipo de radio (transductores y antenas) necesario para servir cada celular de la red. Un grupo de BTS es controlado por una BSC. El canal de comunicación entre BTS y BSC normalmente se conoce con el nombre de conexión “A-bis”.

3.1.3.3 Sistema de Operación y Soporte

Un Centro de Operaciones y Mantenimiento (OMC) se conecta a todo el equipo del Sistema de Conmutación y al Sistema Base-Estación. El proceso de implementar el OMC se conoce como el Sistema de Operación y Soporte (OSS) [7]. El OSS es la entidad funcional donde un operador de la red monitorea y controla el sistema. Permite soporte económicamente accesible y centralizado.

3.1.3.4 Elemento funcional para mensajes cortos

El elemento más utilizado es el Centro de Mensajes (MXE) ya que entre sus múltiples funciones se encuentra la de envío y recepción de mensajes de texto. Un esquema de cómo encaja el centro de mensajes en la arquitectura del sistema global para comunicaciones móviles se detalla en la Fig. 3.3.



Microsoft Office Visio 2007

Figura 3.3 Centro de mensajes cortos (SMSC) en la arquitectura GSM [7]

De manera general es un nodo que provee servicio de voz integrada, fax, correo electrónico y mensajería instantánea. Específicamente el MXE se utiliza para el correcto manejo del servicio de mensajes cortos por medio del Centro de Mensajes Cortos (SMSC). El SMSC se conecta a la base de datos HLR por medio del Punto de Transferencia de Señales (STP). Las funciones del SMSC son:

- ✓ Recibir y almacenar los mensajes cortos enviados por una estación móvil hasta que puedan ser enviados a su destino.
- ✓ Autenticar la estación móvil por medio del VLR.
- ✓ Consultar el HLR para verificar si el estado de la estación móvil destino es activo.

- ✓ Verificar periódicamente el estado de los dispositivos móviles con mensajes pendientes.

3.1.4 Servicio de mensajes cortos (SMS)

El servicio de mensajes cortos es un servicio inalámbrico de comunicaciones incluido como estándar en el sistema global para móviles [8]. Usa protocolos de comunicaciones para permitir el intercambio de mensajes cortos de texto entre estaciones móviles y otros equipos como sistemas de mensajes de voz y correo electrónico (ver Fig. 3.3). El sistema global para comunicaciones móviles ha sido diseñado especialmente para la telefonía pero la innovación de enviar mensajes cortos utilizando el sistema optimizado ha crecido increíblemente en los últimos años.

Originalmente la idea principal era utilizar el sistema optimizado de telefonía para transportar mensajes que controlaran el tráfico de información. Estos mensajes se enviaban en los momentos en que no existía tráfico alguno y así aprovechar los recursos de la red cuando no se utilizaban sin costo adicional. Para lograr esto se limitó el tamaño de los mensajes a 128 bytes (actualmente 140 bytes o 160 caracteres de 7 bits cada uno) para que se ajustaran a los formatos de los paquetes de transmisión. Este concepto se aplicó luego a cada estación móvil provocando el uso de nuevas rutinas de software, la introducción del centro de mensajes cortos para su control, la mejora de las capacidades en radio frecuencia y la optimización en infraestructura de la red GSM.

El servicio hace uso del centro de mensajes cortos para almacenar y dirigir los mensajes. La arquitectura de la red GSM proporciona los mecanismos necesarios para encontrar la estación destino y transportar los mensajes entre el centro y los dispositivos terminales. Los elementos en la arquitectura de la red están diseñados para garantizar el envío de la información al destino otorgando seguridad.

3.2 Arquitectura del dispositivo inalámbrico para GSM

La arquitectura de un dispositivo inalámbrico que funciona bajo el sistema global para comunicaciones móviles puede variar su constitución de un dispositivo a otro dependiendo del fabricante. Los elementos más importantes que componen una estación móvil inalámbrica son la unidad de transducción de radio frecuencia, la unidad de procesamiento central, la memoria y la unidad de periféricos. Hoy en día los teléfonos celulares son módems inalámbricos más complejos pero con los mismos módulos para su funcionamiento básico. Un módem es controlado por medio de un lenguaje de comandos conocido como los comandos de Hayes [9].

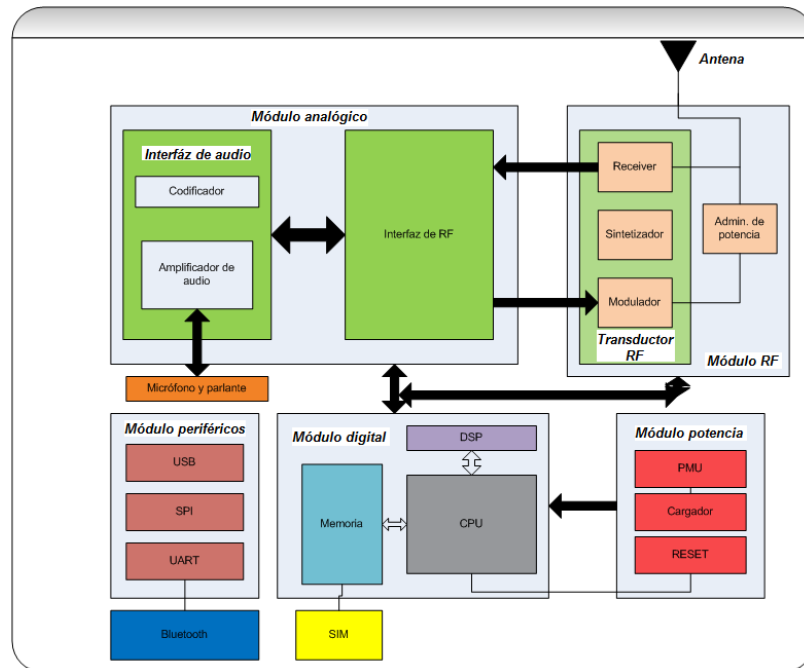
3.2.1 Diagrama de bloques de un modem GSM

Los elementos básicos de un módem inalámbrico que funciona en el sistema global de comunicaciones móviles son: el módulo de radio frecuencia, el módulo digital y el módulo de periféricos. En un sistema de estación móvil o teléfono celular se pueden encontrar más módulos que otorgan otras capacidades complejas (sistema empotrado) como el módulo para la administración de potencia, el módulo analógico y el de “*Bluetooth*” entre otros. La Fig. 3.4 detalla el diagrama de bloques de un módem con capacidades de estación móvil sobre GSM [16].

El módulo de radio frecuencia se compone de un transductor de radio frecuencia para recibir, sintetizar y modular las señales de voz. También tiene un bloque para ajustar la tensión y la potencia de las señales que entran o salen del dispositivo.

El módulo digital se compone de una unidad de procesamiento central y memoria donde se procesan y almacenan las instrucciones básicas para el control de todos los componentes del dispositivo. En éste módulo se encuentran también los registros de configuración del dispositivo (*setting registers*). El módulo digital es necesario debido a que los datos son enviados y recibidos en formato digital por lo que es necesario procesar toda la información digital con un microprocesador. Actualmente gracias a las nuevas tecnologías de integración muchos módulos son

integrados en un solo chip en conjunto con el procesador lo que reduce el tamaño, el consumo de potencia y aumenta la capacidad de procesamiento de los sistemas.



Microsoft Office Visio 2007

Figura 3.4 Diagrama de bloques de un dispositivo GSM

El módulo analógico se utiliza para convertir las señales digitales en analógicas y viceversa. Es la interfaz de los datos digitales con la información analógica como el audio y la voz que comprende el ser humano. La interfaz de radio frecuencia en éste módulo realiza la función de conversión ya que actualmente los módulos de radio frecuencia se han digitalizado.

El módulo de periféricos se utiliza para proporcionar conectividad al dispositivo con otros equipos. Éste módulo permite la administración del dispositivo como un periférico para una computadora y entre sus protocolos de conexión básicos se encuentra el Bus Serial Universal (USB), la Interfaz Serial de Periféricos (SPI) y el Transmisor-Receptor Asíncrono Universal (UART). Los fabricantes han incorporado un módulo de tecnología inalámbrica “*Bluetooth*” como alternativa a la comunicación serial por cable de datos.

El Módulo de Identificación de Suscriptor (SIM) normalmente recibe a una tarjeta que contiene una llave del suscriptor que identifica al usuario de un dispositivo móvil. El módem GSM y el teléfono celular tienen ambos en su arquitectura el módulo SIM para poder ser identificados por la red y poder aprovechar los servicios de GSM.

Finalmente el módulo para la administración de potencia se utiliza para el control de la potencia de un dispositivo móvil. Posee una Unidad de Control de Potencia (PMU) que regula las funciones de una batería y la carga de dicha batería en dispositivos móviles que las usen para obtener corriente. Normalmente un módem GSM simple toma la corriente que le proporciona el puerto USB para su funcionamiento.

3.2.2 Comandos de Hayes

En 1981 Dennis Hayes desarrolló un modem inteligente llamado “Hayes Smartmodem” en su compañía *Hayes Microcomputer Products, Inc* [9]. Este módem era único en su tipo ya que era un dispositivo con el cual se podía interactuar enviando comandos para configurarlo, ejecutar ciertas operaciones como marcar un número, apagar el parlante, colgar y leer el estado actual de la conexión con el modem. Hayes desarrolló y publicó un conjunto de comandos para controlar un modem a través de una conexión serial. El conjunto de comandos tomó tanta popularidad por los usuarios y fabricantes de módems y/o teléfonos digitales, que hoy en día son un estándar para controlar módems comerciales y profesionales. Los módems que soportan los comandos de Hayes se les llaman “compatibles con Hayes”. Los módems basados en GSM son Hayes compatibles.

Los comandos se estandarizaron, pero como normalmente ocurre, existen muchos estándares. Existen extensiones de los comandos específicos que varían de un fabricante a otro debido a su arquitectura y como resultado el conjunto de comandos de los módems modernos no son completamente compatibles entre sí.

Sin embargo, los comandos de Hayes siguen funcionando y son el núcleo para el control de los módems y teléfonos celulares comerciales.

3.2.2.1 Registros de configuración (Setting-Registers)

En la arquitectura de un módem GSM o de un teléfono celular los comandos de Hayes se utilizan para configurar los registros de configuración (S-Registers) ubicados en el módulo digital. Estos registros varían según cada fabricante pero existen registros estándar que todos los dispositivos poseen. Algunos registros de configuración estándar y su función se detallan en la Tabla 3.2.

Tabla 3.2 Algunos registros de configuración de un dispositivo GSM.

Registro	Función	Rango/Unidades
S0	Auto-contestar al sonar	0-255/timbres
S1	Contador de timbres	0-255/timbres
S2	Código del caracter de escape	0-127/ASCII
S3	Caracter retorno de carro	0-127/ASCII
S4	Caracter de línea	0-127/ASCII
S5	Caracter de retorno	0-32, 127/ASCII

3.2.2.2 Comandos de Hayes para enviar y recibir mensajes cortos

Los comandos de Hayes son también conocidos como “comandos AT” ya que una de las primeras palabras de configuración enviadas al dispositivo móvil es “AT” de la palabra imperativa *atención* [11]. El estándar comúnmente utilizado por los fabricantes para regular las especificaciones de los comandos AT es publicado también por el Instituto Europeo de Estándares en Telecomunicaciones (ETSI) en su documento ETSI TS 100 916 llamado “*Digital cellular telecommunications system (Phase 2+): AT command set for GSM Mobile Equipment (ME)*” [10].

Los comandos utilizados para realizar funciones más complejas se llaman “comandos AT extendidos” y son una modificación de los comandos de Hayes básicos.

Para enviar y recibir mensajes de texto utilizando la tecnología de servicios SMS se debe de tener el Módulo de Identidad de Suscriptor (SIM) de un proveedor de servicios conectado a un módem o teléfono celular GSM [11]. Utilizando alguno de los módulos para control de periféricos del modem se conecta el dispositivo GSM a un puerto de la computadora y con una aplicación se envían las instrucciones de control del módem. Las instrucciones básicas para enviar y recibir mensajes se muestran en la Tabla 3.3.

Tabla 3.3 Comandos AT extendidos para mensajes cortos

Comando AT	Significado
AT+CMGS	Enviar mensaje
AT+CMSS	Enviar mensaje desde memoria
AT+CMGW	Enviar y escribir mensaje a memoria
AT+CMGD	Borrar mensaje
AT+CMGC	Enviar comando
AT+CMMS	Más mensajes para enviar
AT+CMGF=0	Mensaje en modo PDU
AT+CMGF=1	Mensaje en modo texto
AT+CNMI	Configurar bandera de mensaje nuevo
AT+CMGR	Extraer mensaje de una localidad de memoria

Una muestra del envío de instrucciones para el control un modem en el sistema global de comunicaciones móviles está en la Fig. 3.5.

```

1.// AT
2.// OK
3.// AT+CMGF=1
4.// OK
5.// AT+CMGW="+85291234567"
6.// > Ejemplo de envio de instrucciones
7.// +CMGW: 1
8.//
9.// OK

```

Microsoft Office Paint 6.0

Figura 3.5 Comunicación con un módem GSM

La descripción del proceso de la figura anterior detalla la comunicación con el dispositivo GSM por medio de los comandos de Hayes:

1. Línea 1: Se envía el comando de “atención” al dispositivo.
2. Línea 2: El dispositivo responde con la respuesta “OK” lo que significa que los comandos AT son aceptados.
3. Línea 3: Se envía la instrucción de configuración para mensajes en modo texto.
4. Línea 4: El dispositivo responde “OK” indicando que se configuró los mensajes en modo texto.
5. Línea 5 y 6: Se envía al dispositivo la instrucción para enviar mensaje y guardar en memoria local indicando el destino por medio de su número de servicio. Se escribe en esta instrucción el mensaje a enviar finalizando con un retorno de carro.
6. Línea 7: Indica el índice de la localidad de memoria donde se guardó el mensaje.
7. Línea 9: El dispositivo responde con un “OK” si el mensaje se envió exitosamente o con “ERROR” en caso contrario.

3.3 Bases de Datos digitales

Una base de datos digital (*schema* en inglés) es un conjunto de datos pertenecientes a un mismo contexto almacenados sistemáticamente utilizando los recursos electrónicos e informáticos para su posterior uso [13]. Una base de datos digital es administrada por Sistemas Gestores de Bases de Datos (SGBD) que permiten almacenar y acceder a los datos de forma rápida.

3.3.1 Tipos de bases de datos

Las bases de datos pueden clasificarse de varias maneras según el contexto en que se desempeñen o el uso que se le estén dando [13]. La clasificación más utilizada en el campo científico es según la variabilidad de los datos almacenados:

- ✓ **Bases de datos estáticas:** estas bases de datos son normalmente de sólo lectura que primordialmente almacenan datos históricos que no cambian con el tiempo.
- ✓ **Bases de datos dinámicas:** en éstas bases de datos la información almacenada se puede modificar permitiendo operaciones de actualización, borrado, inserción de nuevos datos y la operación básica de consulta.

3.3.2 Modelos de bases de datos

Además de la clasificación de las bases de datos por tipo, una base de datos se puede separar por su modelo de administración de datos. Un modelo de base de datos es una descripción del lugar de almacenamiento de la información, los métodos para almacenamiento y extracción de los datos [14]. Un modelo de bases se implementa de manera lógica utilizando algoritmos eficientes y conceptos matemáticos. Algunos de los modelos más utilizados en la actualidad son:

- ✓ **Base de datos jerárquicos:** almacenan información en una estructura jerárquica en donde la información más importante es almacenada en un nodo raíz y la información subsecuente se almacena en sus nodos hojas.
- ✓ **Bases de datos transaccionales:** el único fin es el envío y recepción de datos a altas velocidades para análisis de calidad, datos de producción entre otros. La redundancia y duplicación de la información no es importante.

- ✓ **Bases de datos relacionales:** es el más utilizado para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de “relaciones lógicas” considerando el conjunto de datos en “tuplas” o tablas compuestas por “registros” (filas) y “campos” (columnas) [14]. El lugar y la forma en que se almacenan los datos no es relevante lo que da la ventaja de recuperar y almacenar datos mediante “consultas” que flexibilizan la administración de la información. Dichas consultas se construyen por medio del Lenguaje Estructurado de Consultas (SQL) el cual es un estándar adoptado por los principales Sistemas Gestores de Bases de Datos relacionales. Durante el diseño de una base de datos relacional se le aplica el proceso de normalización el cual consiste en aplicar una serie de reglas con el fin de evitar la redundancia de los datos, evitar problemas de actualización de los datos en las tablas y proteger la integridad de los datos [13] y [14]. En el modelo relacional en una tupla cada columna debe tener su nombre único, no puede haber filas iguales y todos los datos en una columna deben ser del mismo tipo.

3.3.3 Sistema de gestión de bases de datos

Los Sistemas de Gestión de Bases de Datos o SGBD son conocidos en inglés como *Database Management Systems (DBMS)*. Estos sistemas son aplicaciones que sirven de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan [14].

3.3.3.1 Características de un sistema de gestión de bases de datos

Un sistema de gestión de datos ahorra los detalles acerca del almacenamiento físico de los datos. No interesa si una base de datos ocupa uno o cientos de archivos ya que esto es transparente al usuario. Además, la independencia de los datos da la capacidad de modificar el esquema (físico o lógico) de una base de datos sin cambiar las aplicaciones que utilizan el sistema de gestión. Un sistema de éste tipo administra coherencia en condiciones en las que un dato no debe ser admitido como una entrada debido a restricciones previamente definidas. Como la información almacenada en una base de datos puede ser confidencial

(datos personales o claves de acceso) los SGBD garantizan la seguridad de los datos. Finalmente, proporciona velocidad en la extracción y actualización de la información en una base de datos.

3.3.3.2 Sistema de gestión MySQL

MySQL es un sistema de gestión de bases de datos relacional [15]. Tiene capacidades de multihilo y multiusuario. Es un producto desarrollado por la empresa *Sun Microsystems* bajo la distribución de licencia pública general GNU GPL. Está desarrollado en su mayor parte en lenguaje ANSI C.

MySQL es muy utilizado para aplicaciones basadas en web para internet y en plataformas como Linux y Windows[®]. MySQL es una base de datos de escritura y lectura muy rápida lo que permite el acceso a los datos de manera efectiva.

Para escribir aplicaciones que utilicen el gestor de bases de datos MySQL, es conveniente utilizar lenguajes de alto nivel como C, C++, C#, Pascal, Delphi, Java entre otros en los cuales se pueda utilizar una interfaz o conector entre el sistema de gestión y la aplicación que soporte la Conectividad Abierta de Base de Datos (ODBC).

3.3.3.3 Motor de almacenamiento InnoDB

El motor de almacenamiento InnoDB es una máquina de almacenamiento que utiliza el sistema gestor de bases de datos MySQL para otorgar a la implementación de una base de datos de coherencia y habilidades de recuperación ante un error fatal de la base de datos protegiendo la información del usuario. InnoDB almacena la información del usuario mediante índices manteniendo el orden y la integridad de los datos. Está diseñado para soportar las características principales de las bases de datos relacionales.

3.3.3.4 Procedimientos almacenados

Un procedimiento almacenado (*Stored Procedure* en inglés) es una rutina que se almacena en un sistema gestor de bases de datos [14]. Esta rutina se escribe según el lenguaje que utilice el gestor de bases de datos que esté implementado; en el caso de un gestor relacional el lenguaje para escribir dichos procedimientos almacenados es el SQL.

La ventaja de usar un procedimiento almacenado es que al ser ejecutado, ya sea por orden explícita de un usuario o por una aplicación, ocurre directamente en el SGBD que usualmente funciona en un computador del tipo servidor con acceso directo a los datos. Así el procedimiento actúa sobre los datos directamente y envía los resultados al usuario eliminando la transmisión de los datos completos del servidor al usuario remoto que sobrecarga las telecomunicaciones. También simplifica los códigos de las aplicaciones al tener complejas rutinas de consultas, actualización; entre otras, al estar estos programas trabajando directamente sobre la plataforma del SGBD.

```
DELIMITER |  
  
CREATE PROCEDURE autos(IN velocidad int,IN marca varchar(50))  
BEGIN  
IF velocidad < 120 then  
INSERT INTO familiares VALUES(velocidad,marca);  
ELSE  
INSERT INTO deportivos VALUES(velocidad,marca);  
END IF;  
END;  
|
```

Microsoft Office Paint 6.0

Figura 3.6 Procedimiento almacenado en MySQL

En MySQL los procedimientos almacenados empezaron a ser soportados recientemente en su versión MySQL 5.0. Un ejemplo de un proceso almacenado en MySQL se detalla en la Fig. 3.6.

3.4 Antecedentes bibliográficos

Haciendo una serie de búsquedas en internet con el fin de encontrar sistemas que realicen la organización de la información generada por una red inalámbrica de sensores; no se encontró referencia alguna sobre un sistema implementado de organización de los datos proveniente de dicho tipo de redes. Las referencias que se hacen son orientadas a las redes de sensores propiamente y no al tratamiento de grandes cantidades de información generadas por una red de sensores.

El “*Wireless Sensor Networks Research Group*” es un grupo de investigación sobre redes inalámbricas de sensores; se tipifica como comunidad en internet que se interesa en cubrir los aspectos del diseño eficiente en uso de energía de los nodos, protocolos de enrutamiento e implementación en hardware y software de las redes inalámbricas de sensores pero no orientan sus investigaciones a la administración de la información con un sistema de organización de datos. Expone una solución para la transmisión de datos inalámbricos usando dispositivos GSM con el sistema global para comunicaciones móviles llamado “*Sending SMS alarms with SqidBee[®]*” [12] en donde se envían mensajes de un nodo a otro a través de la red de sensores para notificar sobre un suceso importante en la red.

El presente proyecto toma como referencia el envío de mensajes cortos utilizando dispositivos inalámbricos que utilizan la plataforma GSM para proporcionar al sistema de organización de datos acceso al medio inalámbrico y transmisión de información.

4 Capítulo 4: Procedimiento Metodológico

4.1 Métodos y actividades

4.1.1 Seleccionar el hardware de acceso inalámbrico

En primera instancia la necesidad de recopilar información proveniente de una red inalámbrica de sensores fue el punto de inicio para determinar el hardware de acceso inalámbrico. Debido a que normalmente una red de sensores para el monitoreo ambiental se ubica en lugares remotos se estudió sobre el funcionamiento del sistema global para comunicaciones móviles ya que es un servicio utilizado mundialmente y con una alta cobertura en Costa Rica. Específicamente se investigó sobre el servicio de mensajes cortos.

Después de ello se buscó algunos de los dispositivos existentes que utilizan la tecnología GSM. Se encontró en internet acerca de dispositivos sencillos y muy baratos conocidos como módems GSM que aprovechan la infraestructura de la red. El módem GSM más complejo y de mejores características fue la estación móvil o teléfono celular. Con la arquitectura general de una estación móvil se estudió el funcionamiento de un teléfono celular como un módem GSM para un sistema computacional.

4.1.2 Determinación del lenguaje de programación a utilizar

Con la definición del hardware de acceso al medio inalámbrico a utilizar se procedió a escoger el lenguaje de programación de alto nivel. Se estableció las principales características que se necesitan del lenguaje para la implementación del proyecto. Según estas características se considero los lenguajes Java, Delphi y C#.NET. El lenguaje que se seleccionó es lenguaje de programación de alto nivel orientado a objetos llamado C#.NET.

4.1.3 Verificación del funcionamiento de una estación móvil como módem GSM mediante los comandos de Hayes

Inicialmente uno de los primeros retos del proyecto fue utilizar un teléfono celular como módem GSM. No se contaba con un módem GSM. Para verificar el funcionamiento de éste dispositivo como módem se estudió a fondo su arquitectura. Conociendo la arquitectura general se utilizó una aplicación de prueba para conectar dicho dispositivo por medio del puerto serial a la computadora. Se investigó sobre las instrucciones de Hayes para el control de un módem y se enviaron instrucciones de configuración básicas para inicializar el dispositivo móvil mediante la aplicación de prueba.

4.1.4 Desarrollo de un modelo de base de datos

Se revisó a fondo los modelos existentes que permiten la implementación de una base de datos comprensible. Se seleccionó el modelo de bases de datos relacional [13]. En el desarrollo del modelo se diseñó las tablas básicas que corresponden a los datos comunes que se pueden obtener de una red inalámbrica de sensores. Para el funcionamiento de dicho modelo se investigó sobre los sistemas de gestión de bases de datos relacionales.

4.1.5 Desarrollo de un protocolo de comunicaciones

Se estableció un conjunto de consideraciones que definen el contenido de un mensaje enviado de un punto a otro. Se diseñó la encapsulación de los datos en forma de un paquete que es enviado y recibido por medio de la tecnología de servicios de mensajes cortos sobre GSM. En el protocolo se describió los campos necesarios para formar un paquete de datos, su nomenclatura y el método para detección de errores.

4.1.6 Desarrollo de una rutina para envío de mensajes

Se escribió una rutina que describe la forma en que se envía la información utilizando el dispositivo de acceso inalámbrico GSM. Para dicha rutina se usó las

definiciones del protocolo de comunicaciones para encapsular la información antes de ser enviada. La rutina se vale de los comandos de Hayes para el control del dispositivo en el envío de mensajes de texto sobre el servicio SMS.

4.1.7 Desarrollo de una rutina para recepción de mensajes cortos

Se desarrolló una rutina para describir la forma en que se recibe un nuevo mensaje cuando llega a una estación móvil conectada al nodo central. Para dicha rutina se usó las instrucciones de Hayes para la configuración inicial del dispositivo inalámbrico de tal forma que se notifique acerca del arribo de un nuevo mensaje. Además se utilizó el uso de los comandos para hacer lecturas de la memoria del dispositivo para obtener el mensaje, el estado de un mensaje y la información del emisor de dicho mensaje. Finalmente, se utilizó las definiciones del protocolo de comunicaciones diseñado para des-encapsular los datos provenientes de la rutina de envío.

4.1.8 Desarrollo de las rutinas para crear las tablas en el SGBD relacional

Para implementar las tablas diseñadas en el modelo relacional planteado sobre el sistema de gestión de bases de datos relacional se investigó a fondo sobre el Lenguaje Estructurado de Consultas (SQL). Con el lenguaje comprendido se procedió a la creación de cada una de las tablas mediante pequeñas rutinas en SQL.

4.1.9 Desarrollo de las rutinas para crear los procedimientos almacenados en el SGBD relacional

Después del desarrollo de las rutinas para la creación de las tablas se procedió al desarrollo de los procedimientos almacenados. La creación de los procedimientos almacenados colaboró con la administración básica de la base de datos relacional. Los procedimientos almacenados diseñados cubren las consultas simples de los datos almacenados en cada una de las tablas así como las funciones de actualización de filas, ingreso de nuevas filas y borrado de filas específicas, entre otras funcionalidades.

4.1.10 Desarrollo de los programas que implementan cada una de las rutinas con el lenguaje de programación

Una vez que se describió las rutinas para el envío-recepción de la información, las rutinas para la creación de las tablas y los procedimientos almacenados; se procedió a la implementación de éstas en programas.

En primer lugar, se diseñó un programa general para el control del dispositivo terminal GSM. Este programa general agrupa las rutinas para el envío y recepción de mensajes utilizando los comandos de Hayes. El programa se definió como una librería para controlar un dispositivo GSM-Hayes compatible.

Luego se diseñó el programa para simular el envío de información desde un nodo líder que controla un dispositivo inalámbrico GSM. Para el programa se utilizó la librería de control del dispositivo GSM para enviar datos.

Con la librería de control del dispositivo GSM se procedió a utilizar la rutina de recepción del mensaje en un programa. En el programa se aplicó la rutina de recepción del mensaje configurando el dispositivo para que alerte sobre la llegada de un nuevo mensaje. La rutina misma se encarga de leer dicho mensaje en la memoria del dispositivo móvil GSM. Se agregó al programa de recepción de mensajes la conexión con los procedimientos almacenados para el almacenamiento de los datos.

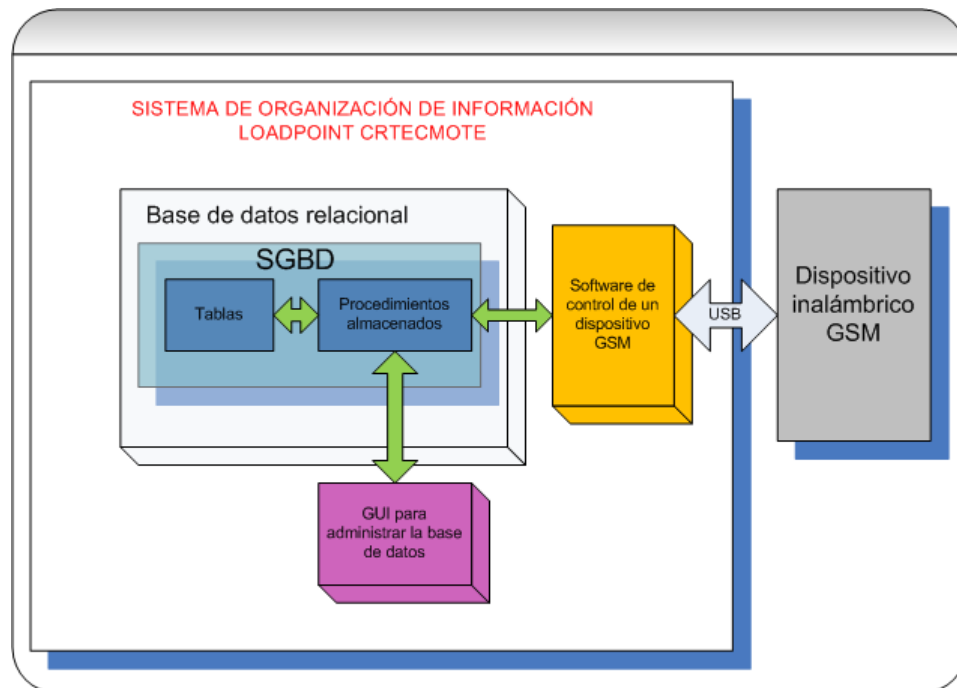
Finalmente, se diseñó una interfaz gráfica para comprobar el modelo de base de datos relacional, la correcta creación de las tablas y el correcto funcionamiento de los procedimientos almacenados.

5 Capítulo 5: Descripción detallada de la solución

A continuación se detalla la información correspondiente al proceso de solución que se abordó con el objetivo de resolver el problema planteado. Se muestran los aspectos más importantes de la solución así como los diagramas de flujo del software implementado.

5.1 Descripción general

El sistema de organización de la información proveniente de una red inalámbrica de sensores es un computador que contiene una base de datos para almacenar la información y un software de control de un dispositivo inalámbrico para la conexión al sistema global de comunicaciones móviles. Dicho software es la interfaz para obtener los datos del dispositivo GSM y almacenar en la base de datos. La Fig. 5.1 detalla el diagrama de bloques de la solución al problema del proyecto.



Microsoft Office Visio 2007

Figura 5.1 Modelo de la base de datos relacional para el Load Point CRTECMOTE

La base de datos sigue el modelo relacional para almacenar información y se compone de un sistema gestor de bases de datos (SGBD) donde se construyó las tablas para organizar la información y los procedimientos almacenados en el SGBD para realizar tareas de administración de las tablas y su contenido.

Para verificar la estructura de la base de datos relacional se realizó una interfaz gráfica de usuario (GUI) en el lenguaje de alto nivel C#. Con éste mismo lenguaje se implementó las rutinas para el programa que controla el hardware de acceso a la red GSM. El dispositivo GSM se conectó al sistema por un puerto del computador que soporta el protocolo del bus serial universal.

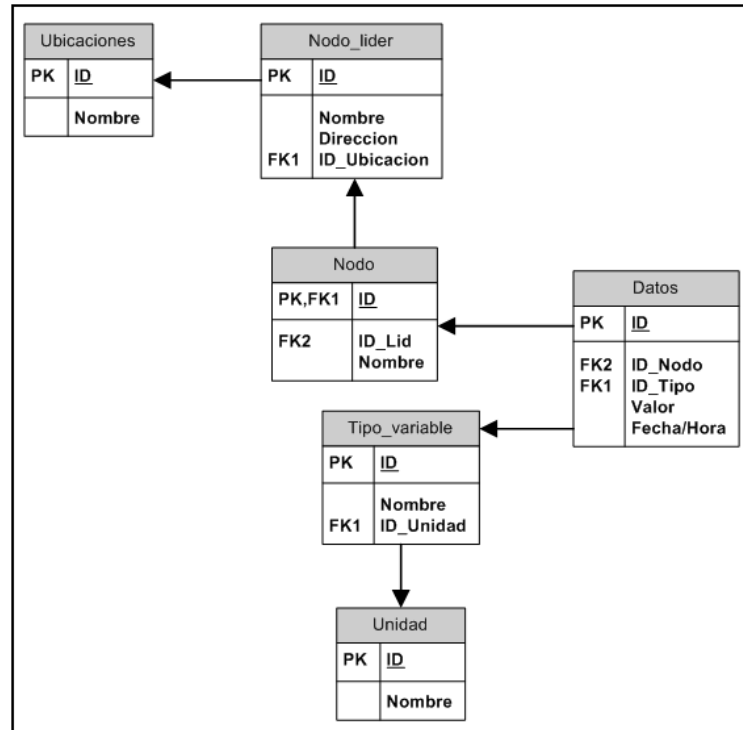
5.2 Diseño de un modelo para la organización de la información

5.2.1 Modelo de base de datos relacional

El diseño para la organización de la información proveniente de una red inalámbrica de sensores consiste en la aplicación del modelo relacional de una base de datos. Para aplicar el modelo, se procedió al establecimiento de grupos de datos de un mismo tipo o contexto mediante tablas. Las tablas son llamadas “tuplas” y están constituidas por “filas” (entradas) y columnas. Cada fila tiene una o más columnas. Los dos tipos de tuplas implementadas son basadas en el esquema de “padre” e “hijo”. Una tabla “padre” contiene un grupo de datos del mismo tipo de la cual una tabla “hijo” necesita una o más filas. Una tabla “hijo” se convierte en una tabla “padre” a su vez, cuando otra tabla hace referencia a alguna de sus filas.

Las relaciones de las tablas unas con otras se manejó bajo el esquema de uniones lógicas llamadas “llaves”. Cada tupla tiene una “llave primaria” (*primary key*) y es única. Ésta llave es la identificación de la tabla. Una tupla “padre” pura solamente tiene llave primaria. Una tabla “hijo”, además de la llave primaria, tiene llaves secundarias llamadas “llaves extranjeras” (*foreign keys*) y son utilizadas para apuntar de manera lógica a las llaves primarias de las tablas tipo padre. El diseño del modelo descrito se muestra en la Fig. 5.2.

El esquema de la figura muestra en detalle las tablas creadas para el modelo de bases de datos relacional en el sistema de organización de la información generada por una red inalámbrica de sensores.



Microsoft Office Visio 2007

Figura 5.2 Modelo de la base de datos relacional para el Load Point CRTECMOTE

5.2.2 Definición de las tuplas en el diseño del modelo relacional

Según la Fig. 5.2, las tablas de tipo padre puras son “Ubicaciones” y “Unidad”. La primera entidad tiene una única columna que se encarga de almacenar la ubicación de un nodo de la red. Cuando una persona se encargue de distribuir los nodos sensores de la red inalámbrica es imperativo que anote la ubicación de dicho nodo para que sea ingresado en esta tabla. Nótese que es una tabla padre pura debido a la ausencia de llaves extranjeras. De igual manera, la segunda entidad se encarga de guardar la información correspondiente a las unidades de los parámetros que se desean medir con los nodos sensores; sea el caso de un dato de

temperatura en la que su unidad en el sistema internacional de unidades es el “Celsius”.

La tupla “Nodo_líder” es del tipo hijo ya que apunta a su padre que es “Ubicaciones”. La función de dicha tabla es almacenar el nombre de los nodos tipo líder en la red inalámbrica y la dirección de dicho nodo para ubicar el área donde se hizo la distribución de los nodos sensores bajo el mando del nodo líder. Una llave extranjera en esta tabla apunta a las ubicaciones de cada uno de los nodos de la red.

La tabla “Nodo” es una tabla hijo de la tabla “Nodo_líder” y en su defecto de “Ubicaciones” como se observa en la Fig. 5.2. Posee una llave extranjera que apunta a su tabla padre directo pero también hereda un apuntador implícito a la tabla de “Ubicaciones”. Esta tabla se encarga de guardar los nombres de los nodos distribuidos de la red. Con la llave extranjera se hacen las relaciones de la información correspondiente a un nodo sensor en específico.

La entidad llamada “Tipo_variable” es un hijo de “Unidad” y se encarga de depositar la información referente al tipo de medición que hace un sensor. Sus columnas corresponden al nombre del tipo de medición que se está haciendo (temperatura, humedad, acidez entre muchas más) y a la llave extranjera que apunta a su unidad correspondiente en la tabla padre.

Finalmente, la tupla llamada “Datos” es una tabla “hijo” pura que almacena directamente la información proveniente de la red inalámbrica de sensores mediante sus columnas del valor medido, la fecha/hora en que se midió dicho valor, el número de identificación del nodo que hizo la medición y el número de identificación del tipo de variable. Estos últimos datos corresponden a las llaves extranjeras para relacionar el dato entrante con la información contenida en las tablas descritas anteriormente.

5.3 Puesta en marcha de una base de datos relacional MySQL

Para la implementación de una base de datos relacional se escogió un sistema gestor de bases de datos MySQL que utiliza el motor de almacenamiento InnoDB.

5.3.1 Requerimientos para el funcionamiento del SGBD relacional MySQL

Para poner en marcha el sistema gestor de bases de datos relacional MySQL fue necesario conocer el hardware y el sistema operativo en el cual se deseaba ejecutar el sistema gestor de la base de datos. En el caso del presente proyecto se utilizó una computadora personal HP Compaq nc6400 con procesador Intel Core Centrino® Duo con 2GB de memoria RAM y 2 GHz de velocidad. El sistema operativo es el ambiente Windows Vista® de 32 bits. Así el SGBD utilizado es la versión MySQL 5.0 para Windows® [15].

5.3.2 Instalación del SGBD MySQL

La instalación de MySQL 5.0 para Windows® se realiza mediante una interfaz gráfica para su configuración. MySQL presenta un archivo ejecutable para levantar dicha interfaz y empezar con la instalación del SGBD. Para lograr el funcionamiento del gestor de bases de datos, en primera instancia por medio de la interfaz, se procedió a la instalación del sistema gestor de manera manual de tal forma que se pudiera especificar las características más importantes para el funcionamiento deseado de la base de datos. En las opciones de instalación se decidió agregar además del SGBD, la ventana de comandos y la documentación de ayuda para los comandos. Con las opciones listas se procedió a la instalación.

Una vez instalado, se continuó con la configuración del SGBD con la ayuda de la interfaz gráfica. Se escogió hacer una configuración detallada con el fin de definir el tipo de gestor necesario. Para el proyecto se utilizó un gestor de base de datos para el desarrollo de código como muestra la Fig. 5.3.



Microsoft Office Paint 6.0

Figura 5.3 Pantalla tomada del instalador de MySQL para escoger el tipo de gestor

El siguiente paso fue la decisión del tipo de base de datos que se utilizó para el proyecto. Debido a que el modelo diseñado es el relacional se procedió a escoger la implementación de la base de datos “multifuncional” la cual es la instalación estándar que proporciona MySQL 5.0 para bases de datos de propósito general de prueba y en producción. Esta escogencia soporta el modelo relacional establecido. La Fig. 5.4 muestra la pantalla donde se puede hacer dicha escogencia en la interfaz del instalador.



Microsoft Office Paint 6.0

Figura 5.4 Pantalla tomada del instalador de MySQL para escoger el tipo de base de datos

Después de escoger las opciones del tipo de base de datos, se especificaron los detalles para la localidad en disco duro del motor de almacenamiento InnoDB que utiliza MySQL 5.0. Por último, se especificaron los últimos detalles para que la

base de datos funcione como un servicio en Windows® y se introdujo la clave de seguridad de administrador del SGBD (ver Fig. 5.5).

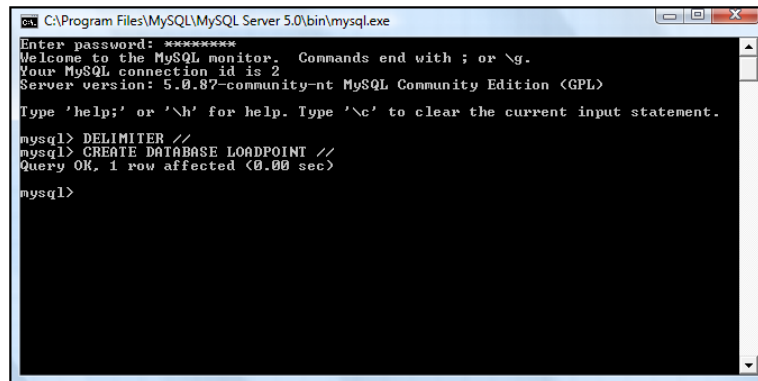


Microsoft Office Paint 6.0

Figura 5.5 Pantalla tomada del instalador de MySQL para otorgar seguridad al SGBD

5.3.3 Creación del esquema de la base de datos (*schema*) en el SGBD relacional MySQL

Para la creación de una base de datos se debe de usar el lenguaje estructurado de consultas SQL en el sistema de gestión de bases de datos MySQL. Para lograr esto se necesitó de la ventana de comandos del gestor.



Microsoft Office Paint 6.0

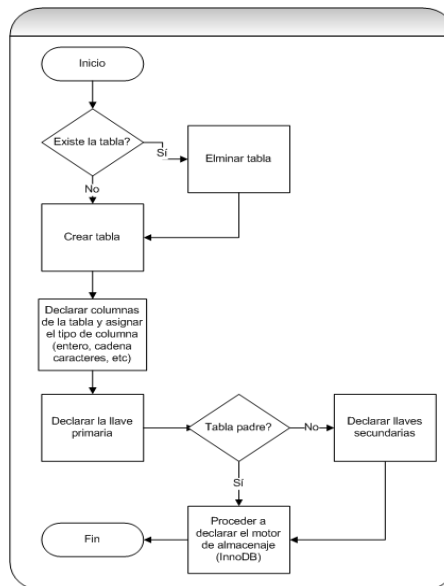
Figura 5.6 Ventana de comandos del SGBD MySQL 5.0 para crear una base de datos

En la Fig. 5.6 se muestran los comandos utilizados para crear una base de datos (*schema*) en MySQL. Se inició introduciendo la clave de acceso para

administrar el SGBD. Se estableció como delimitador el caracter “//” para señalar el fin de cada instrucción. Seguidamente se procedió a crear la base de datos mediante la instrucción “**CREATE DATABASE nombre**”, donde se escogió el nombre de la base de datos como “loadpoint” en la figura para mostrar el funcionamiento del comando (el nombre del *schema* para el proyecto es “lp”). El SGBD indica si la instrucción fue aceptada o en su defecto indica un error con un número y una explicación sencilla de dicho error.

5.3.4 Creación de una tabla en el SGBD

Para crear una tabla bajo el modelo de base de datos relacional se diseñó el algoritmo de la Fig. 5.7.



Microsoft Office Visio 2007

Figura 5.7 Diagrama de flujo para crear una tabla en el modelo relacional

Para implementar cada una de las tablas del modelo de la Fig. 5.2 se utilizó el lenguaje estructurado de consultas (SQL) en el SGBD siguiendo la secuencia de pasos descritos en el diagrama de flujo.

Como primer paso para crear una tabla se aseguró que esta sea única, si la tabla existe se elimina; mediante el comando “**CREATE TABLE nombre**” se creó la

tabla, luego se declaró las columnas de cada tupla especificando si la columna va a contener enteros, caracteres, fechas, en otros tipos de datos. Cuando se tuvo definido lo anterior se procedió a la declaración de la llave primaria mediante el comando SQL “**CONSTRAINT nombre PRIMARY KEY (columna de referencia)**”. Si la tupla es hijo se definen las llaves extranjeras; además de la primaria, mediante el comando “**REFERENCES tabla FOREIGN KEY (primary key de la tabla)**”. Finalmente se escoge el motor de almacenamiento como “InnoDB” el cual es con el que trabaja MySQL para guardar información.

5.3.5 Diseño de los procedimientos almacenados en el sistema de gestión de bases de datos

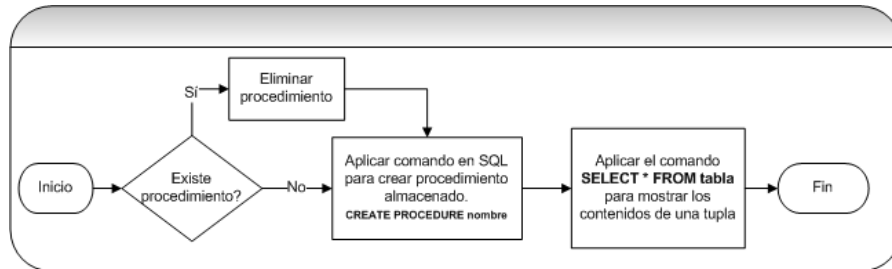
Los procedimientos almacenados son las rutinas básicas para la administración de las tablas y su contenido en el modelo de base de datos relacional diseñado. Cada procedimiento almacenado es una serie de instrucciones en lenguaje estructurado para realizar una acción específica. Se diseñó los procedimientos almacenados básicos para actualizar una fila, borrar una fila, agregar una fila en una tabla y mostrar una tabla completa. A continuación se explican los cuatro procedimientos básicos diseñados para la administración de la base de datos del sistema de organización de la información. Los diagramas de flujo presentados son muy generales ya que cada tabla tiene sus propias columnas lo que hace variar sus algoritmos de procedimientos almacenados.

5.3.5.1 Procedimiento almacenado para mostrar una tabla específica

Con el fin de mostrar el contenido de una tabla específica se diseñó un procedimiento almacenado que seleccione todo el contenido de una tupla. El algoritmo de la Fig. 5.8 muestra los pasos para crear el procedimiento que logró este objetivo.

Para crear el procedimiento en primer lugar se utilizó el comando “**CREATE PROCEDURE nombre ()**” en lenguaje SQL. El “*nombre*” es el identificador del procedimiento, los paréntesis indican que el procedimiento no necesita parámetros.

Si el procedimiento existe, éste es eliminado y creado nuevamente. Luego se aplicó el comando “**SELECT * FROM tabla;**” para seleccionar todo el contenido de la *tabla*.

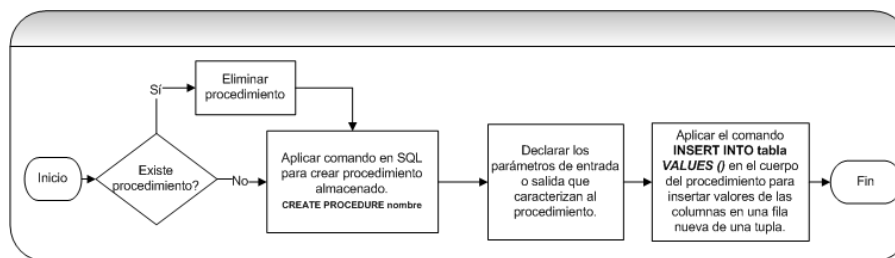


Microsoft Office Visio 2007

Figura 5.8 Diagrama de flujo de un procedimiento almacenado para mostrar el contenido de una tabla en MySQL

5.3.5.2 Procedimiento almacenado para insertar en una tabla específica

Para realizar la inserción de una nueva entrada en una tabla se diseñó un algoritmo de procedimiento almacenado como se muestra en la Fig. 5.9. En primer lugar se determinó la función del procedimiento almacenado para definir los parámetros. En el cuerpo de la rutina se finalizó con el uso del comando para insertar una nueva entrada en una tabla específica.



Microsoft Office Visio 2007

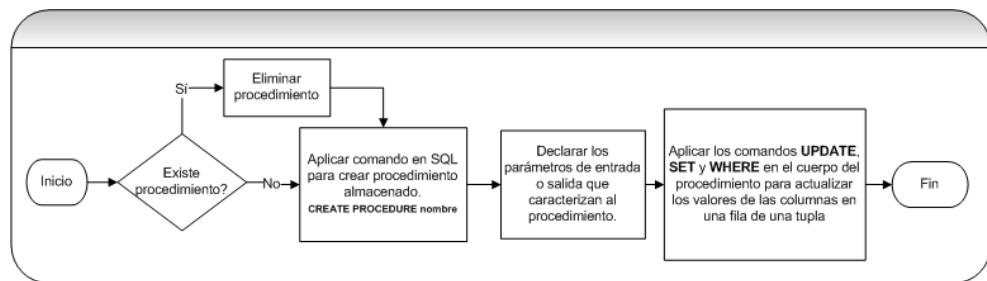
Figura 5.9 Diagrama de flujo de un procedimiento almacenado para insertar una nueva entrada en una tabla en MySQL

Para crear el procedimiento se utilizó el comando “**CREATE PROCEDURE nombre (param1, param2)**” en lenguaje SQL. Luego se aplicó el comando “**INSERT**

INTO *tabla* **VALUES** (*param1*, *param2*);” para insertar en la *tabla* los valores de los parámetros de entrada.

5.3.5.3 Procedimiento almacenado para actualizar la fila en una tabla

El procedimiento para actualizar una fila en una tabla se muestra en la Fig. 5.10. Este algoritmo se diseñó con el fin de no borrar toda la tupla y en vez de eso seleccionar una fila y actualizar sus valores.



Microsoft Office Visio 2007

Figura 5.10 Diagrama de flujo de un procedimiento almacenado para actualizar una fila en una tabla en MySQL

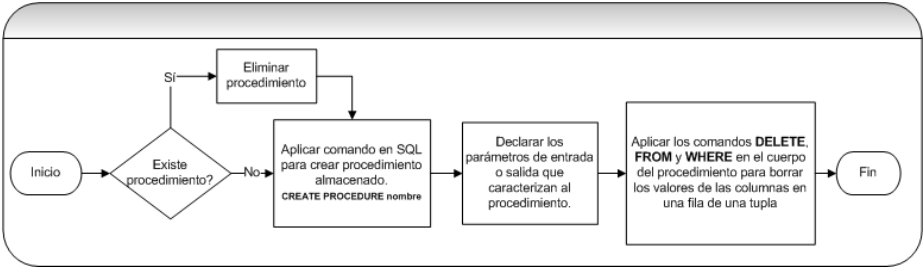
El procedimiento se escribió en lenguaje estructurado utilizando el conjunto de comandos “**UPDATE** *tabla* **SET** *columna=param2* **WHERE** *columna=param1*;”. Con este comando se indica que el contenido definido por el parámetro de entrada “*param1*” se sustituya por el contenido de “*param2*”.

5.3.5.4 Procedimiento almacenado para borrar la fila en una tabla

El procedimiento almacenado en el SGBD para borrar una fila en una tabla del diseño planteado se detalla en la Fig. 5.11. El algoritmo es similar a los anteriores ya que el lenguaje estructurado es el mismo para implementar cada uno de estos procesos.

El comando usado para borrar una fila en una tabla en MySQL es “**DELETE FROM** *tabla* **WHERE** *columna=param1*;”. Lo que se realizó en el algoritmo es buscar

la columna con el nombre contenido en el “*param1*” y borrar la fila de la tabla especificada.



Microsoft Office Visio 2007

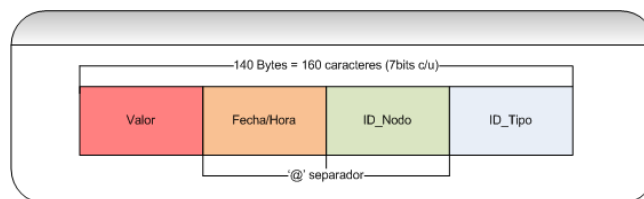
Figura 5.11 Diagrama de flujo de un procedimiento almacenado para borrar una fila en una tabla en MySQL

5.4 Diseño del protocolo de comunicaciones

El diseño del protocolo de comunicaciones se basó en la necesidad de establecer la forma en que los datos iban a ser recibidos y comprendidos por el sistema de organización de la información. Debido a que los nodos sensores en una red inalámbrica comparten información fue necesario tipificar el tipo de dato que va a recibir el sistema de organización de la información. Para tipificar el tipo de dato se estructuró la forma que debe de tener un paquete para que el nodo central pueda entenderlo y poder así darle tratamiento para que sea almacenado en la base de datos. Debido a que la información es estructurada de una manera definida, al momento de leer la estructura para su comprensión en el nodo de carga se consideró la integridad de los datos.

5.4.1 Definición de la estructura de la información

Para definir la estructura de los datos en el modelo relacional se consideró el dato medido de un nodo sensor como un valor. Dicho valor puede corresponder a múltiples tipos de variables dependiendo del ambiente en que se encuentre ubicada la red de sensores y el propósito de dicho sistema.



Microsoft Office Visio 2007

Figura 5.12 Diseño básico del paquete de datos en el protocolo de comunicaciones

Bajo el principio de funcionamiento de que la red inalámbrica de sensores se planea para el monitoreo en primera instancia ambiental [1], se definió que la estructura posible de la información proveniente de un nodo de la red puede ser un dato correspondiente a un valor, la fecha y hora en que se mide dicho valor, un identificador del nodo sensor y finalmente el identificador del tipo de dato

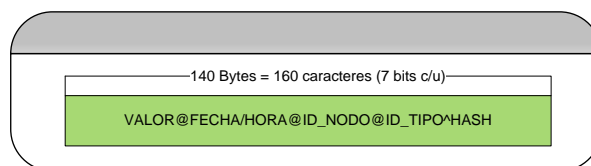
(temperatura, humedad, presión, fuerza, etc.) que se está midiendo del ambiente. De manera conjunta todos estos datos forman un paquete de datos como se muestra en la Fig. 5.12. El diseño de la estructura de la información que se muestra en la figura corresponde a la descripción dada anteriormente. Se denota que el paquete de datos o trama tiene un máximo de 140 bytes de tamaño, lo que significa que la información enviada debe de tener un máximo de 160 caracteres.

De éstos 160 caracteres se deben de restar al menos 3 caracteres que corresponden al separador (se definió el separador '@') de cada uno de los datos enviados en una trama. La cantidad de bytes enviados puede ser variable pero debe de ser menor a los 140 bytes. Los separadores se utilizan para conocer el final de un dato y el inicio de otro en la trama de información. Estos separadores son usados en una rutina para leer dicha información.

5.4.2 Definición de la integridad de los datos

Para garantizar la integridad de los datos se definió el uso de un código para detección de errores. El algoritmo que se utilizó es el cálculo de un código "Hash" mediante una librería contenida en el lenguaje de programación C#.

Dicho algoritmo se ejecuta para la cadena de texto definida en la Fig. 5.12 y el resultado es un código hash. El código se agrega al paquete de datos definido (encapsulamiento) mediante un nuevo separador escogido como el caracter "^". La Fig. 5.13 muestra el encapsulamiento del paquete de datos con el hash. El código es des-encapsulado en el nodo central y se ejecuta el algoritmo hash sobre el paquete de datos original nuevamente.

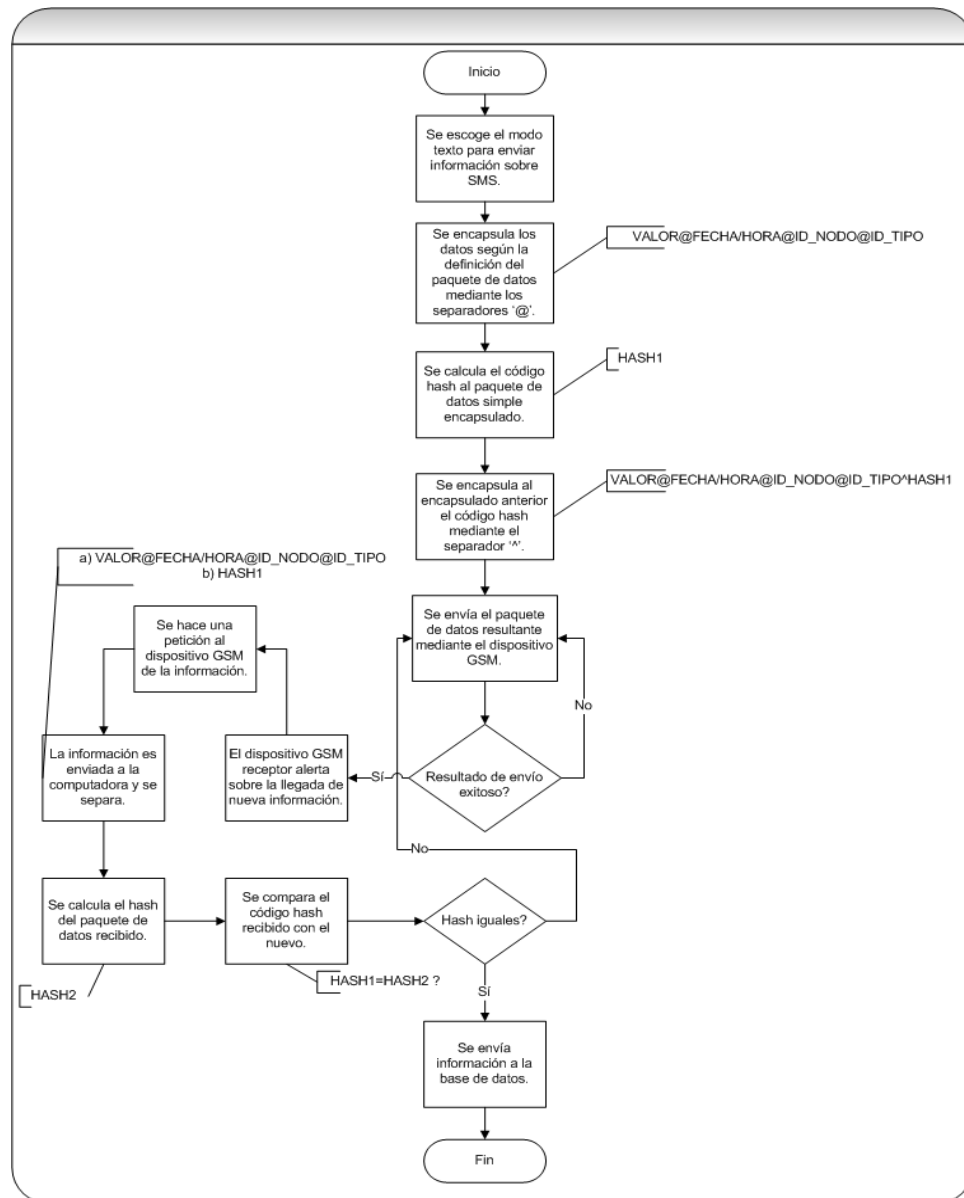


Microsoft Office Visio 2007

Figura 5.13 Encapsulado del paquete de datos con el código "hash"

5.4.3 Diagrama de flujo del protocolo de comunicaciones

El diagrama de flujo del protocolo de comunicaciones completo se muestra en la Fig. 5.14. El envío y recepción de información por medio de los dispositivos GSM ocupa del software diseñado para el control del dispositivo inalámbrico conectado al sistema de organización de la información.



Microsoft Office Visio 2007

Figura 5.14 Diagrama de flujo del protocolo de comunicaciones diseñado para el sistema de organización de información

5.5 Diseño del software para el control del hardware inalámbrico GSM

Una de las piezas claves del sistema de organización de la información generada por una red inalámbrica de sensores es la capacidad de conexión al medio inalámbrico. Se escogió el envío de información utilizando la plataforma de mensajes cortos del sistema global para comunicaciones móviles. GSM tiene la facilidad de que es la tecnología escogida por el país para telefonía móvil y tiene la particularidad de que cubre casi la totalidad de Costa Rica, lo que permite la escalabilidad de cualquier sistema que utilice el servicio. Para conectar el sistema de organización de información al servicio inalámbrico se escribió una librería en software o controlador de un dispositivo funcional sobre la red GSM.

5.5.1 Verificación de funcionamiento del hardware inalámbrico GSM

5.5.1.1 Selección del dispositivo inalámbrico GSM

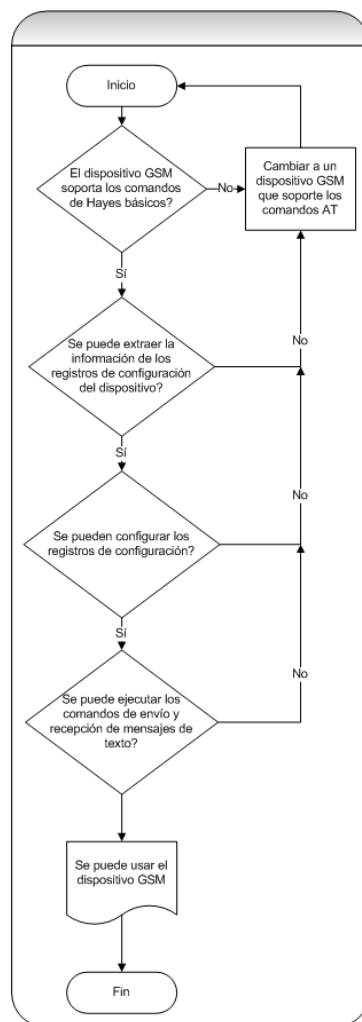
Debido a la necesidad de un modem GSM se consideró el uso de un teléfono celular como módem. Se contó con dos modelos de estaciones móviles inicialmente: Nokia 5310 y Motorola L7.

Para escoger entre los dos dispositivos se utilizó la aplicación *Hyperterminal* de Windows® y se estableció una conexión serial con el dispositivo GSM utilizando el módulo de "Bluetooth". El dispositivo seleccionado fue el teléfono Motorola L7 con el software versión R4513_G_08.B7.AER_RB.

Se estableció el diagrama de flujo de la Fig. 5.14 utilizando cuatro operaciones básicas para escoger el dispositivo que se ajustara mejor a las necesidades del proyecto:

1. **Operación de prueba:** se utilizó para revisar si cierto comando de Hayes es soportado o no por el dispositivo.

2. **Operación de configuración:** se utilizó para cambiar los registros de configuración usados por el dispositivo para realizar las funciones necesarias de envío-recepción de mensajes.
3. **Operación de lectura:** fue utilizada para extraer las configuraciones de los registros en el dispositivo.
4. **Operación de ejecución:** se utilizó para ejecutar una tarea o extraer información y/o estado de un dispositivo GSM.



Microsoft Office Visio 2007

Figura 5.15 Diagrama de flujo para seleccionar un dispositivo GSM

5.5.1.2 Verificación de las operaciones de los comandos AT sobre el dispositivo GSM seleccionado

La verificación de la operación de los comandos AT necesarios para el envío-recepción de mensajes de texto sobre el dispositivo GSM se realizó revisando el contenido los registros de configuración en el teléfono móvil utilizado para el proyecto. La lista completa de comandos y su explicación para el dispositivo Motorola L7 se encuentra en el manual de referencia de comandos AT de Motorola (*Motorola AT commands Reference Manual*) [17].

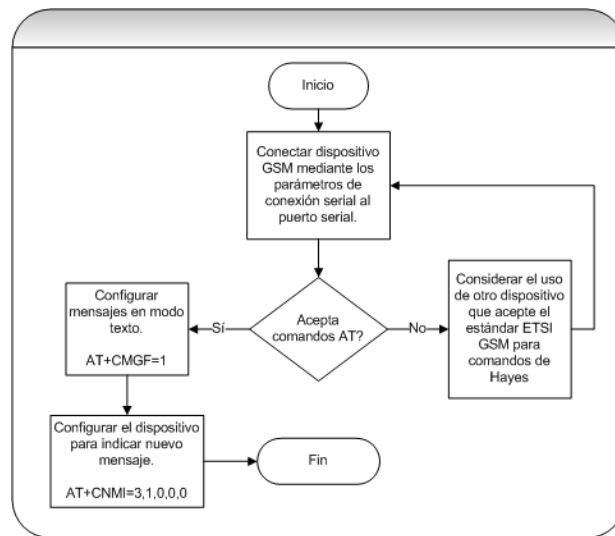
Para mostrar el formato de los mensajes soportado por el dispositivo GSM se utilizó el comando “**AT+CMGF=?**”. Según el manual de referencia de comandos AT de Motorola® existen dos modos de formato, el modo cero es para el envío-recepción de mensajes utilizando la definición de unidades de paquete de datos (PDU) y es el modo por defecto que utiliza el hardware, el modo uno es para el envío-recepción de mensajes en modo texto.

Para la recepción de mensajes, los valores posibles de configuración del registro de indicación de nuevo mensaje (CNMI) se obtuvieron enviando al dispositivo el comando “**AT+CNMI=?**”. Según el manual de referencia de comandos AT de Motorola®, el primer valor de configuración mostrado en la respuesta del comando corresponde al modo: cero para no indicar sobre un nuevo mensaje a un equipo terminal de datos (sistema computacional) y tres para indicar el arribo de un nuevo mensaje; el segundo valor corresponde al requerimiento o no de que la indicación adjunte la posición de memoria del mensaje en el dispositivo GSM, la respuesta al comando muestra otros tres valores que corresponden al aviso del cambio de celda en la red GSM, al reporte del estado del dispositivo y las indicaciones del estado del búfer de la conexión serial.

5.5.2 Rutina de inicialización del dispositivo GSM

Para inicializar el dispositivo móvil GSM se requirió de los comandos AT utilizados para verificar la operación como se muestra en el algoritmo de la Fig. 5.16.

En el diseño de la rutina de inicialización se indica que se debe de establecer la conexión con el dispositivo móvil especificando los parámetros para una conexión serial. Los parámetros básicos para una conexión serial por “Bluetooth” del Motorola L7 fueron el puerto (COM12) y la velocidad de transmisión de los datos (460800). Siguiendo con la rutina, se envió al periférico GSM el comando de atención, el comando se aceptó y se continuó con la inicialización de los mensajes en modo texto cambiando el registro CMGF (*message format command*) a uno.



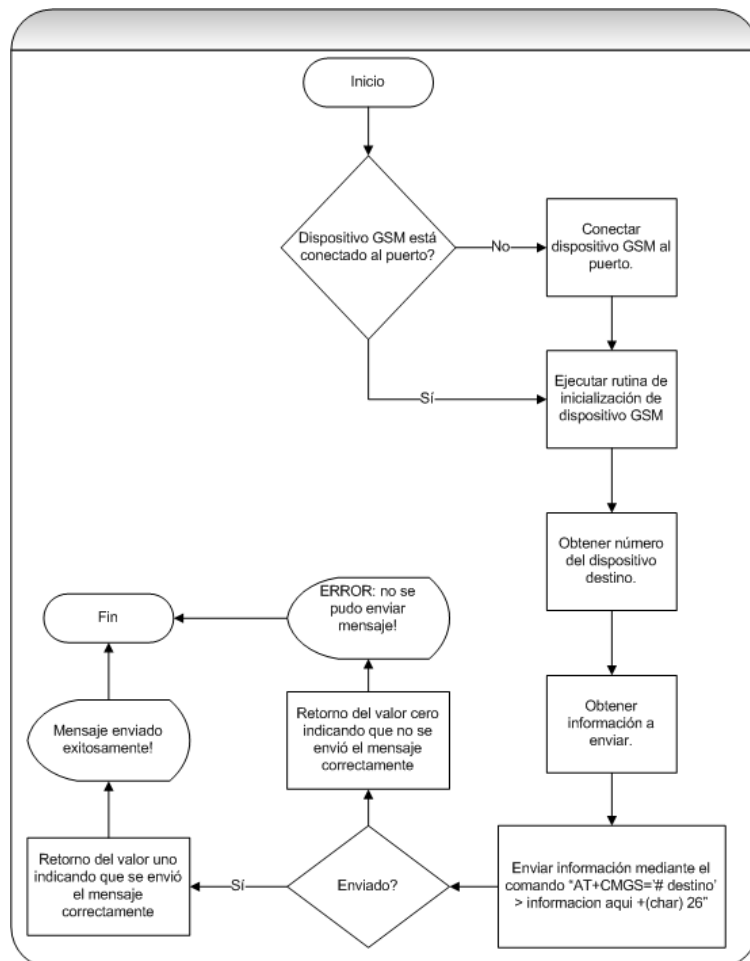
Microsoft Office Visio 2007

Figura 5.16 Diagrama de flujo para inicializar el dispositivo GSM

Una vez que el registro de modo de los mensajes se cambió se procedió a configurar en el dispositivo el registro de indicación de nuevo mensaje CNMI (*new message indication command*). Como se detalla en la Fig. 5.16, el modo se escoge en tres para que todo mensaje nuevo sea indicado por medio de una notificación al equipo terminal, luego se indica con un uno el campo para que el aviso tenga la posición en memoria del mensaje en el dispositivo GSM y el resto de los campos se ponen por defecto ya que el registro del Motorola L7 no presenta más características.

5.5.3 Rutina de envío de información

Para enviar la información necesaria por medio de un mensaje de texto se siguió la secuencia de pasos indicados de la Fig. 5.17. Para realizar la acción del envío de información utilizando el servicio de mensajes cortos se introdujo al dispositivo GSM el comando “AT+CMGS= #destino”, inmediatamente después se envió la información en forma de texto finalizando con el código en ASCII para el retorno de carro. Si el mensaje se envió con éxito la rutina devuelve un uno, de lo contrario la rutina devuelve un cero. Esta es la rutina utilizada para simular el envío de datos desde un nodo líder al nodo central.

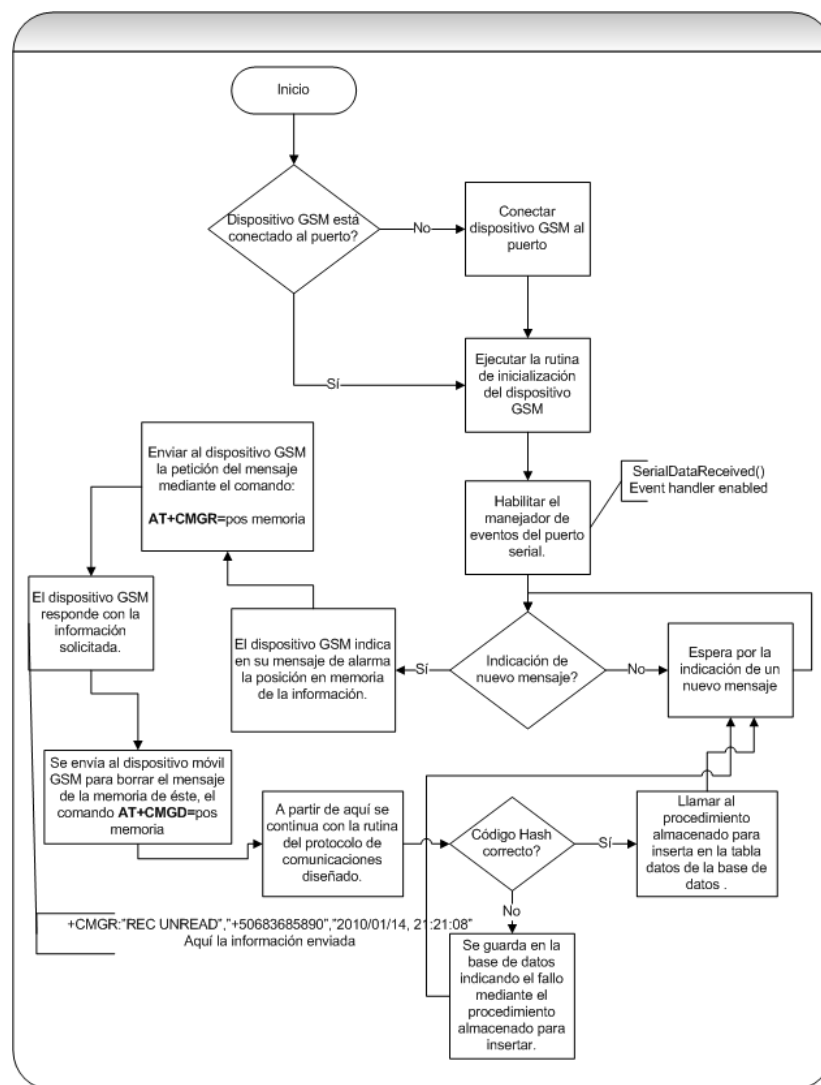


Microsoft Office Visio 2007

Figura 5.17 Diagrama de flujo para enviar un mensaje de texto con un dispositivo GSM

5.5.4 Rutina de recepción de información

Esta es la rutina más importante del sistema ya que es la que controla la recepción de información del hardware de acceso al medio GSM para el sistema de organización de información (ver Fig. 5.18). En esta rutina se concentra la comunicación con el dispositivo GSM en el nodo central para la correcta recepción de la información enviada por la rutina anterior. Se diseñó bajo el principio de funcionamiento de que el dispositivo móvil se inicializa para notificar sobre el arribo de un nuevo mensaje.



Microsoft Office Visio 2007

Figura 5.18 Diagrama de flujo para recibir y almacenar información con un dispositivo GSM

Con un manejador de eventos habilitado en el puerto serial se lee la notificación que indica la posición en memoria de la nueva información. Entonces, se envió al modem GSM el comando “**AT+CMGR=pos memoria**” para leer de memoria el nuevo mensaje. El dispositivo respondió con el mensaje y es leído por la rutina. Luego es almacenado temporalmente en una lista enlazada. Se envió al modem el comando “**AT+CMGD=pos memoria**” para borrar el mensaje almacenado en el dispositivo y así liberar el espacio de la memoria cuando se dé un funcionamiento crítico del sistema. De ésta forma se aplican las condiciones del protocolo de comunicaciones para separar e interpretar la información recibida, calculando el código hash para validar la integridad de los datos. Si los datos se validan correctamente, se procede a almacenar la información en la base de datos llamando al procedimiento almacenado de insertar del SGBD en la tupla “Datos” del modelo relacional. En su defecto se guarda indicando el fallo en la integridad de los datos.

5.5.5 Selección del lenguaje de programación de alto nivel

Para la selección del lenguaje de programación de alto nivel se consideró los siguientes aspectos:

- ✓ Necesidad de contar con una gran cantidad de bibliotecas de código que se pueda reutilizar y de un ambiente de desarrollo de software.
- ✓ Necesidad de bibliotecas de código para el control de un puerto serial.
- ✓ Facilidad para crear interfaces gráficas de usuario.
- ✓ Facilidad para crear aplicaciones para conexión a un sistema gestor de bases de datos.

De la gran gama de lenguajes de programación de alto nivel se consideró Java, Delphi y C# .NET. Se escogió C#. NET como el lenguaje de programación para el proyecto ya que cumple con todas las características anteriores.

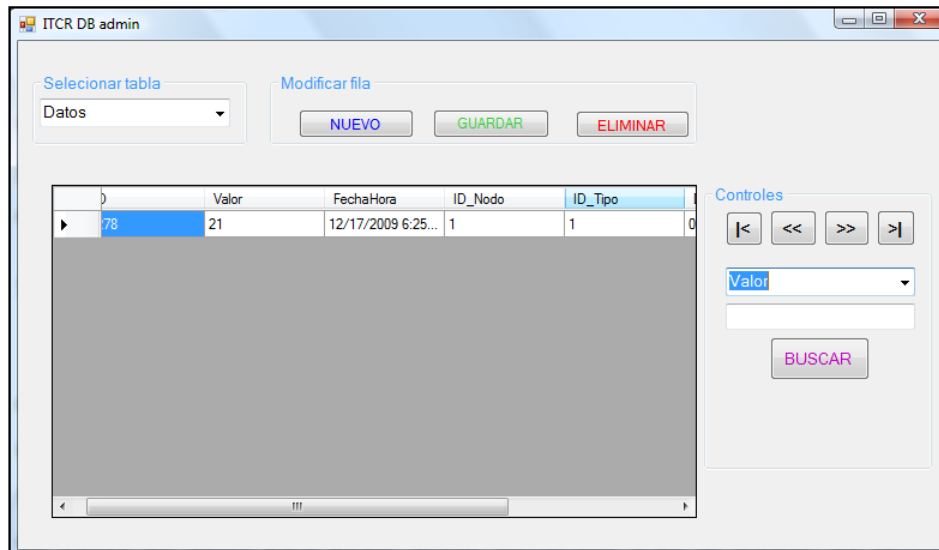
5.6 Diseño de pruebas de funcionamiento del sistema de organización de información

Para demostrar el funcionamiento del sistema de organización de la información diseñado, se realizó diversas pruebas de campo. Para lograr la realización de las pruebas de funcionamiento se debió escribir código adicional a las rutinas explicadas en el capítulo para tener la facilidad de una interfaz gráfica de usuario.

5.6.1 Pruebas de estructura del modelo de base de datos relacional

Para demostrar que la estructura del modelo de la base de datos relacional diseñada es correcta se creó la interfaz gráfica de la Fig. 5.19. La interfaz gráfica hace uso de un conector ODBC desarrollado por MySQL para poder conectar a la base de datos relacional. Este conector funciona de interfaz entre la aplicación y el sistema gestor de bases de datos para poder enviar comandos SQL al SGBD. El conector se utilizó en el presente proyecto como administrador del SGBD MySQL ya que se pueden tener varios SGBD y muchas bases de datos en cada sistema gestor. La aplicación se escribió en lenguaje de alto nivel C# utilizando las bibliotecas disponibles en .NET Framework.

Como se muestra en la figura, con esta aplicación es posible ver cada una de las tuplas diseñadas en el modelo relacional seleccionando el nombre de cada tabla en el menú “Seleccionar tabla”. En cada tabla se pueden realizar tres funciones básicas expuestas en el menú “Modificar filas” que utilizan los procedimientos almacenados en el SGBD MySQL diseñados para administrar el modelo. Se puede insertar un nuevo registro o fila, guardar o actualizar una fila o columna de una fila y eliminar una fila seleccionada. Los “Controles” son para navegar por las filas de cada una de las tablas. Finalmente, se utilizan algunos procedimientos almacenados para hacer búsquedas en cada una de las tablas.



Microsoft Paint 6.0

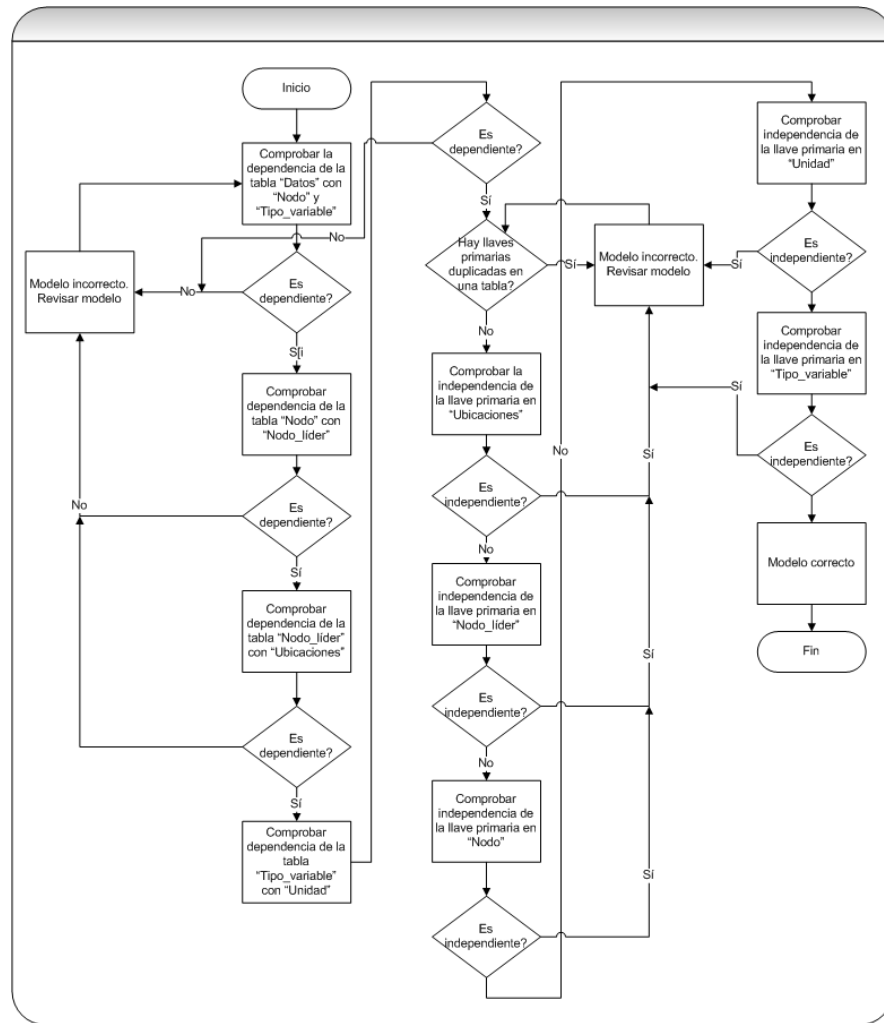
Figura 5.19 Interfaz gráfica para administrar la base de datos relacional

Se utilizó la interfaz anterior para comprobar el diseño del modelo relacional (ver Fig. 5.2) y su característica de protección de datos mediante los siguientes casos de prueba:

1. **Dependencia de la tabla “Datos” con “Nodo” y “Tipo_variable”:** insertar en la tabla “Datos” un dato con la llave ID_Nodo (llave que apunta a alguna llave primaria de la tabla “Nodo”) y la llave ID_Tipo (llave que apunta a alguna llave primaria de la tupla “Tipo_variable”) cuando no existe la llave primaria correspondiente en la tabla “Nodo” y/o la tabla “Tipo_variable”.
2. **Dependencia de la tabla “Nodo” con “Nodo_líder”:** insertar en la tabla “Nodo” el nombre de un nodo y la llave extranjera ó ID_Lid sin que exista el nodo líder correspondiente.
3. **Dependencia de la tabla “Nodo_líder” con “Ubicaciones”:** insertar en “Nodo_líder” el nombre del nodo líder y su dirección sin tener el ID_Ubic que ubica el nodo sensor al cual el nodo líder gobierna.

4. **Dependencia de la tabla “Tipo_variable” con “Unidad”:** insertar en la tabla “Tipo_variable” un tipo de dato (temperatura, presión, humedad, etc.) apuntando con el ID_Uni la unidad del tipo de dato sin haber adicionado previamente su unidad en la tabla “Unidad”.
5. **Llaves primarias duplicadas:** Insertar en cualquier tabla una llave primaria o ID duplicado.
6. **Independencia de la llave primaria en “Ubicaciones”:** borrar en la tabla “Ubicaciones” un ID o llave primaria en la cual una llave secundaria de la tabla “Nodo_lider” esté utilizando.
7. **Independencia de la llave primaria en “Nodo_líder”:** borrar en la tabla “Nodo_líder” un ID a la cual una llave secundaria de la tabla “Nodo” esté utilizando.
8. **Independencia de la llave primaria en “Nodo”:** borrar en la tabla “Nodo” un ID a la cual la llave extranjera de la tabla “Datos” esté apuntando.
9. **Independencia de la llave primaria en “Unidad”:** borrar en la tupla “Unidad” un ID a la cual la llave secundaria de la tabla “Tipo_variable” esté apuntado.
10. **Independencia de la llave primaria en “Tipo_variable”:** borrar en la tabla “Tipo_variable” una llave primaria a la cual una llave extranjera de la tabla “Datos” esté apuntando.

Para verificar la correcta creación de la estructura del modelo de base de datos se realizó el algoritmo de la Fig. 5.20 basándose en la definición de los casos mencionados anteriormente.



Microsoft Office Visio 2007

Figura 5.20 Diagrama de flujo para verificar la estructura del modelo de base de datos relacional

5.6.2 Prueba de bombardeo de mensajes

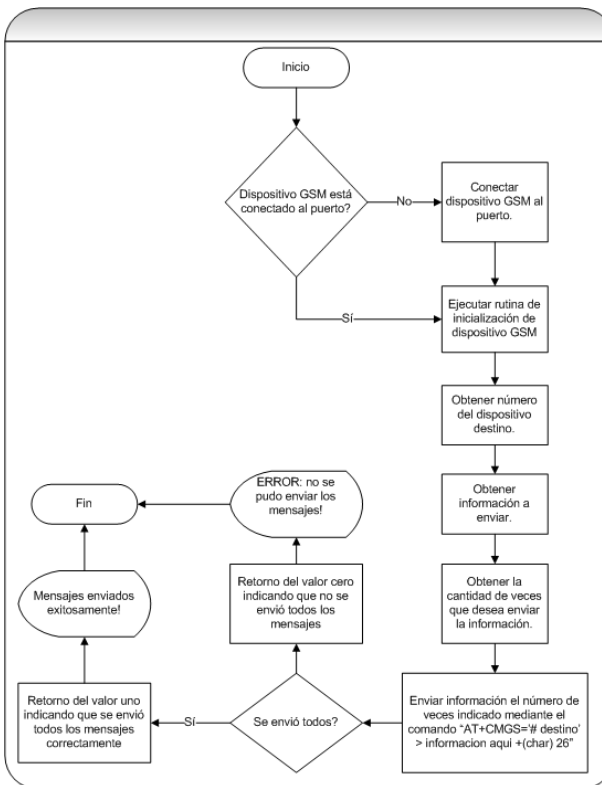
Para comprobar el funcionamiento del sistema de organización de la información ante un envío significativo de mensajes se diseñó una interfaz gráfica capaz de enviar una seguidilla de mensajes utilizando el software controlador del hardware de acceso al medio inalámbrico GSM. La Fig. 5.21 muestra la interfaz de usuario gráfica diseñada que utiliza el software controlador del dispositivo inalámbrico para enviar información. Esta interfaz se puede ejecutar en cualquier computador personal que tenga .Net Framework instalado y es el sistema que simula el envío de datos desde un nodo líder.



Microsoft Paint 6.0

Figura 5.21 Interfaz gráfica para simular el envío de datos desde un nodo líder

A la rutina de control del dispositivo GSM se le agregó una modificación para el envío de mensajes en forma masiva (“*bulk*”). El diseño de la rutina se muestra en la Fig. 5.22.



Microsoft Office Visio 2007

Figura 5.22 Diagrama de flujo para el envío de mensajes en modo masivo

De la interfaz gráfica de usuario se utilizó su funcionalidad de enviar un número de veces un mismo mensaje mediante la rutina en “*bulk*”. La prueba de bombardeo de mensajes al nodo central consistió en enviar una cantidad determinada de mensajes conteniendo un dato en el formato definido por el protocolo de comunicaciones.

Básicamente se midió la cantidad de mensajes enviados y la cantidad de mensajes recibidos para obtener el éxito del sistema ante una gran carga de transmisión.

5.6.3 Prueba de integridad de la información

Para comprobar la integridad de la información se utilizó las definiciones del protocolo de comunicaciones de la Fig. 5.14. Antes de enviar un dato se calculó el código hash de la información y se encapsuló el código en el mensaje. Cuando el mensaje se recibe en el nodo central el hash se des-encapsula y se vuelve a calcular el hash sobre el mensaje recibido. Ambos códigos se comparan para determinar si son iguales y validar la integridad de los datos.

Para esta prueba se agregó una columna a la tupla “Datos” llamada “Hash” para almacenar el resultado de la integridad de la información. Si el hash recibido y el calculado son iguales se valida el dato como “correcto” y se guarda en la base de datos esta palabra en la nueva columna de la tabla. Si es “incorrecto” se guarda el dato en la base de datos con dicha palabra en la nueva columna.

5.6.4 Prueba para medir el retraso en la entrega de la información

Se diseñó una prueba para medir la diferencia de tiempo que existe entre el momento en que se envía un dato y el momento en que se recibe el dato en el sistema. Para medir este retraso se utilizó la ecuación 5.1 donde Δt corresponde a la diferencia de tiempo entre el tiempo final t_f y el tiempo inicial t_0 .

$$\Delta t = t_f - t_0 \quad (5.1)$$

Utilizando la interfaz de envío de mensajes para la simulación del nodo líder se toma el tiempo del sistema computacional (tiempo inicial) en el momento justo antes de enviar el mensaje utilizando el controlador del dispositivo inalámbrico GSM. Este tiempo es encapsulado al mensaje total mediante una rutina que adiciona el tiempo de retraso al encapsulamiento del código hash utilizando el separador “^”. Cuando el mensaje llega al sistema de organización de información se toma el tiempo del sistema computacional como el tiempo final, se separa el tiempo inicial de la trama de datos y se ejecuta la ecuación 5.1.

Se adicionó para esta prueba una columna en la tupla de “Datos” llamada “Delta” en el modelo relacional con el fin de almacenar la duración de envío y recepción de la información.

6 Capítulo 6: Análisis de resultados

En éste capítulo se muestran los resultados obtenidos de cada una de las etapas desarrolladas para lograr un sistema de organización de la información y el análisis de dichos resultados. Entre los resultados más relevantes se encuentra la creación del “*schema*” de la base de datos, las tablas y los procedimientos almacenados en el SGBD. Se muestran también los resultados de la verificación de la estructura del modelo relacional.

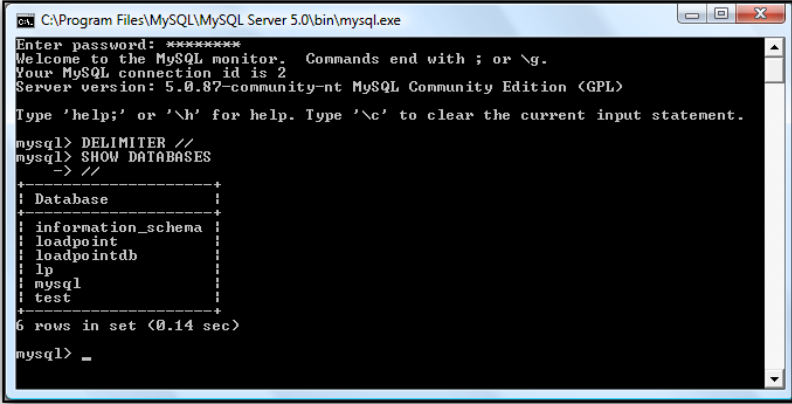
De igual forma se detallan los resultados de las operaciones para seleccionar el dispositivo GSM, los resultados de las pruebas de funcionalidad del sistema, la integridad de los datos y la medición del retraso en la entrega de información.

Para comprobar la funcionalidad del sistema de organización de datos y hacer las mediciones necesarias se realizó el envío de 50 mensajes de información, conteniendo la trama especificada en el diseño del protocolo. Estas pruebas se realizaron en diferentes días y a diferentes horas de cada día para proporcionar al grupo diseñador de la red inalámbrica de sensores de información concluyente sobre integridad, calidad y tiempo de transmisión de los datos. Se detalla aquí los resultados de la prueba para un día en tres horarios distintos.

6.1 Resultados

6.1.1 Verificación de la creación del esquema (*schema*) de una base de datos en el SGBD MySQL

Para comprobar la correcta creación de un esquema de la base de datos se procedió a introducir el comando “**SHOW DATABASES**” en la ventana de comandos de MySQL 5.0 para mostrar las actuales bases de datos existentes en el computador. La base de datos creada para el proyecto se muestra en la Fig. 6.1 bajo el nombre de “**lp**”.



```
C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql.exe
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.0.87-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> DELIMITER //
mysql> SHOW DATABASES
-> //
+-----+
| Database |
+-----+
| information_schema |
| loadpoint |
| loadpointdb |
| lp |
| mysql |
| test |
+-----+
6 rows in set (0.14 sec)

mysql> _
```

Microsoft Paint 6.0

Figura 6.1 Verificación de la creación del esquema de una base de datos en el SGBD de MySQL

6.1.2 Resultados de la creación de tablas en el SGBD de MySQL

Como resultado de la implementación del algoritmo de la Fig. 5.7 se realizó el código en lenguaje estructurado SQL para cada una de las tablas del modelo relacional diseñado según la Fig. 5.2. Para crear una tabla se utilizó el mismo procedimiento para todas las demás. Nótese que es aquí donde se especifica el motor de almacenamiento InnoDB de MySQL.

Únicamente se muestran aquí los resultados para crear la tupla “Ubicaciones”. El resultado de aplicar los pasos para crear la tabla “Ubicaciones” en un sistema gestor de base de datos MySQL se muestra en la Fig. 6.3.

```

2 #####Ubicaciones#####
3 #####Ubicaciones#####
4 #####Ubicaciones#####
5
6 DROP TABLE IF EXISTS `lp`.`ubicaciones`;
7 CREATE TABLE `lp`.`ubicaciones` (
8   `ID` int(11) NOT NULL,
9   `Nombre` varchar(255) NOT NULL,
10  PRIMARY KEY (`ID`)
11 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Microsoft Paint 6.0

Figura 6.3 Fragmento de código para crear la tabla “Ubicaciones” en el SGBD

El resultado de utilizar el algoritmo para crear la tabla “Unidad” en un sistema gestor de base de datos MySQL se muestra en la Fig. 6.4.

```

13 #####Unidad#####
14 #####Unidad#####
15 #####Unidad#####
16 DROP TABLE IF EXISTS `lp`.`unidad`;
17 CREATE TABLE `lp`.`unidad` (
18   `ID` int(11) NOT NULL,
19   `Nombre` varchar(255) NOT NULL,
20   PRIMARY KEY (`ID`)
21 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Microsoft Paint 6.0

Figura 6.4 Fragmento de código para crear la tabla “Unidad” en el SGBD

En la Fig. 6.5 se detalla el uso del lenguaje estructurado para la creación de la tupla “Nodo_Líder” y es claramente el resultado de aplicar el algoritmo de la Fig. 5.6.

```

23 #####Nodo_lider#####
24 #####Nodo_lider#####
25 #####Nodo_lider#####
26
27 DROP TABLE IF EXISTS `lp`.`nodo_lider`;
28 CREATE TABLE `lp`.`nodo_lider` (
29   `ID` int(11) NOT NULL,
30   `Nombre` varchar(255) NOT NULL,
31   `Direccion` varchar(255) NOT NULL,
32   `ID_Ubic` int(11) NOT NULL,
33   INDEX (`ID_Ubic`),
34   CONSTRAINT `fk_UbicID` FOREIGN KEY (`ID_Ubic`)
35   REFERENCES `lp`.`ubicaciones` (`ID`),
36   CONSTRAINT `pk_UbicID` PRIMARY KEY (`ID`)
37 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
38
39 #####Nodo_lider#####

```

Microsoft Paint 6.0

Figura 6.5 Fragmento de código SQL para crear la tabla “Nodo_Líder” en el SGBD

El resultado para crear la tabla de “Tipo_variable” del modelo relacional diseñado utilizando la secuencia de pasos definidas en la Fig. 5.7 se muestra en la Fig. 6.6. Nótese el código en esta tabla; así como en las demás, que se definen las columnas y las llaves justo después de darle nombre a la tabla.

```

39 #####
40 #####Tipo_Variable#####
41 #####
42
43 DROP TABLE IF EXISTS `lp`.`tipo_variable`;
44 CREATE TABLE `lp`.`tipo_variable` (
45   `ID` int(11) NOT NULL,
46   `Nombre` varchar(255) NOT NULL,
47   `ID_Uni` int(11) NOT NULL,
48   INDEX (`ID_Uni`),
49   CONSTRAINT `fk_UniID` FOREIGN KEY (`ID_Uni`)
50   REFERENCES `lp`.`unidad` (`ID`),
51   CONSTRAINT `pk_UniID` PRIMARY KEY (`ID`)
52 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Microsoft Paint 6.0

Figura 6.6 Fragmento de código en SQL para crear la tabla “Tipo_variable” en el SGBD

Finalmente, el algoritmo para la creación de las tablas de “Nodo” y “Datos” dio como resultado el código de la Fig. 6.7.

```

54 #####
55 #####Nodo#####
56 #####
57 DROP TABLE IF EXISTS `lp`.`nodo`;
58 CREATE TABLE `lp`.`nodo` (
59   `ID` int(11) NOT NULL,
60   `Nombre` varchar(255) NOT NULL,
61   `ID_Lid` int(11) NOT NULL,
62   INDEX (`ID_Lid`),
63   CONSTRAINT `fk_LidID` FOREIGN KEY (`ID_Lid`)
64   REFERENCES `lp`.`nodo_lider` (`ID`),
65   CONSTRAINT `pk_nodo` PRIMARY KEY (`ID`)
66 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
67
68 #####
69 #####Datos#####
70 #####
71 DROP TABLE IF EXISTS `lp`.`datos`;
72 CREATE TABLE `lp`.`datos` (
73   `ID` int(11) NOT NULL,
74   `Valor` double NOT NULL,
75   `FechaHora` datetime NOT NULL,
76   `ID_Nodo` int(11) NOT NULL,
77   `ID_Tipo` int(11) NOT NULL,
78   `Delta` varchar(35) NOT NULL,
79   `Hash` varchar(15) NOT NULL,
80   INDEX (`ID_Nodo`),
81   INDEX (`ID_Tipo`),
82   CONSTRAINT `fk_NodoID` FOREIGN KEY (`ID_Nodo`)
83   REFERENCES `lp`.`nodo` (`ID`),
84   CONSTRAINT `fk_TipoID` FOREIGN KEY (`ID_Tipo`)
85   REFERENCES `lp`.`Tipo_Variable` (`ID`),
86   CONSTRAINT `pk_datos` PRIMARY KEY (`ID`)
87 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

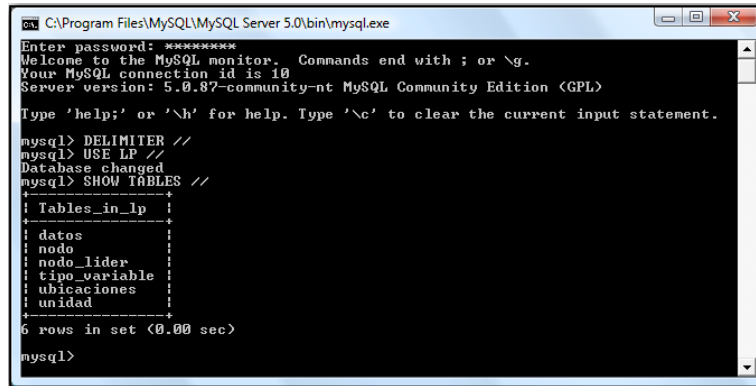
```

Microsoft Paint 6.0

Figura 6.7 Fragmento de código SQL para crear las tablas “Nodo” y “Datos” en el SGBD

Para comprobar la creación de cada una de las tablas se utilizó el comando “**SHOW TABLES**” en la ventana de comandos del sistema gestor de bases de datos como se muestra en la Fig. 6.8. Se escogió la base de datos utilizada en el proyecto

llamada “lp” mediante el comando “USE lp” para mostrar sus tablas. Para entender la relación entre las tablas es necesario observar el modelo de la Fig. 5.2.



```
C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql.exe
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.0.87-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> DELIMITER //
mysql> USE lp //
Database changed
mysql> SHOW TABLES //
+-----+
| Tables_in_lp |
+-----+
| datos        |
| nodo        |
| nodo_lider  |
| tipo_variable |
| ubicaciones |
| unidad      |
+-----+
6 rows in set (0.00 sec)

mysql>
```

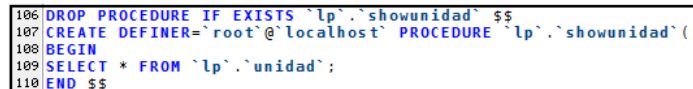
Microsoft Paint 6.0

Figura 6.8 Resultado de la creación de las tablas del modelo relacional en el SGBD de MySQL

6.1.3 Resultados de la creación de los procedimientos almacenados en un SGBD de MySQL

Para el funcionamiento básico del sistema se implementó los procedimientos almacenados: insertar, guardar, borrar y mostrar tabla. Cada uno de estos procedimientos se aplicó por cada tabla implementada del modelo de la Fig. 5.2. A continuación se muestran los procedimientos almacenados implementados solamente para la tupla “Unidad”.

La Fig. 6.9 muestra el procedimiento almacenado para mostrar la tupla “Unidad” llamado “*showunidad*”. Para mostrar una tabla se utilizó la secuencia de pasos definidos en la Fig. 5.8

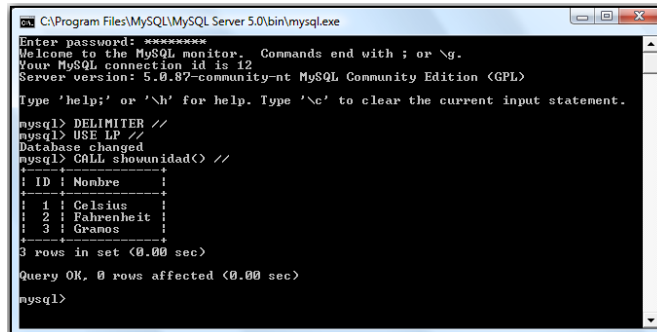


```
106 DROP PROCEDURE IF EXISTS `lp`.`showunidad` $$
107 CREATE DEFINER=`root`@`localhost` PROCEDURE `lp`.`showunidad`()
108 BEGIN
109 SELECT * FROM `lp`.`unidad`;
110 END $$
```

Microsoft Paint 6.0

Figura 6.9 Fragmento de código del procedimiento almacenado para mostrar el contenido de una tabla en MySQL

El resultado de utilizar este procedimiento en el sistema gestor de bases de datos se detalla en la Fig. 6.10.



```
CAProgram Files\MySQL\MySQL Server 5.0\bin\mysql.exe
Enter password: 
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.0.87-community-nt MySQL Community Edition (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> DELIMITER //
mysql> USE LP //
Database changed
mysql> CALL showunidad() //
+----+-----+
| ID | Nombre |
+----+-----+
| 1  | Celsius |
| 2  | Fahrenheit |
| 3  | Grados |
+----+-----+
3 rows in set (0.00 sec)

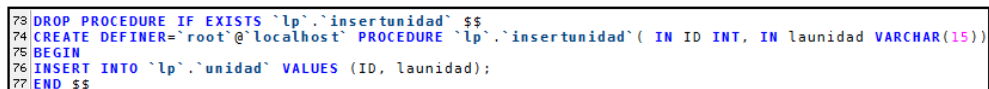
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Microsoft Paint 6.0

Figura 6.10 Resultado de llamar el procedimiento almacenado para mostrar el contenido de una tabla en MySQL

La Fig. 6.11 muestra el procedimiento almacenado para insertar una fila en la tupla “Unidad” llamado “*insertunidad*” (ver rutina del la Fig. 5.9). El procedimiento requiere de dos parámetros, el número de fila y el nombre de la unidad.



```
73 DROP PROCEDURE IF EXISTS `lp`.`insertunidad` $$
74 CREATE DEFINER=`root`@`localhost` PROCEDURE `lp`.`insertunidad`( IN ID INT, IN launidad VARCHAR(15))
75 BEGIN
76 INSERT INTO `lp`.`unidad` VALUES (ID, launidad);
77 END $$
```

Microsoft Paint 6.0

Figura 6.11 Procedimiento almacenado que inserta una nueva entrada en una tabla en MySQL

El resultado de utilizar el procedimiento en el sistema gestor de bases de datos para insertar una nueva fila en la tabla “Unidad” se detalla en la Fig. 6.12. Nótese la nueva entrada con índice cuatro que corresponde al dato de “Kilogramos”.

```

C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql.exe
mysql> CALL insertunidad(4,'Kilogramos'); //
Query OK, 1 row affected (0.12 sec)

mysql> CALL showunidad() //
+----+-----+
| ID | Nombre |
+----+-----+
| 1  | Celsius |
| 2  | Fahrenheit |
| 3  | Gramos |
| 4  | Kilogramos |
+----+-----+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>

```

Microsoft Paint 6.0

Figura 6.12 Resultado de llamar el procedimiento almacenado para insertar una fila de una tabla en MySQL

La Fig. 6.13 es un fragmento de código implementado para el sistema de organización con la función de actualizar filas en la tabla “Unidad” llamado “*updateunidad*”. Se utilizó el algoritmo de la Fig. 5.10.

```

94 DROP PROCEDURE IF EXISTS `lp`.`updateunidad` $$
95 CREATE DEFINER=`root`@`localhost` PROCEDURE `lp`.`updateunidad`( IN launidad VARCHAR (15), IN nuevaunidad VARCHAR (15))
96 BEGIN
97 UPDATE `lp`.`unidad`
98 SET unidad.Nombre = nuevaunidad
99 WHERE unidad.Nombre = launidad;
100 END $$

```

Microsoft Paint 6.0

Figura 6.13 Fragmento de código del procedimiento almacenado para actualizar una fila en una tabla en MySQL

El resultado de utilizar el procedimiento almacenado para actualizar una fila de la tabla “Unidad” en MySQL se detalla en la Fig. 6.14. Se actualizó “Kilogramos” por “Libras”.

```

C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql.exe
mysql> CALL updateunidad ('Kilogramos', 'Libras') //
Query OK, 1 row affected (0.03 sec)

mysql> CALL showunidad() //
+----+-----+
| ID | Nombre |
+----+-----+
| 1  | Celsius |
| 2  | Fahrenheit |
| 3  | Gramos |
| 4  | Libras |
+----+-----+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> _

```

Microsoft Paint 6.0

Figura 6.14 Resultado de llamar el procedimiento almacenado para actualizar una fila de una tabla en MySQL

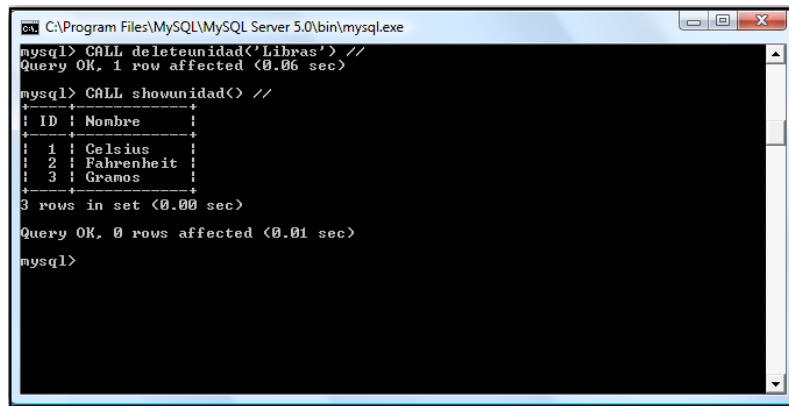
La Fig. 6.15 es el fragmento del código en SQL producto del algoritmo de la Fig. 5.11 para borrar una fila en la tabla “Unidad” del modelo relacional llamado “deleteunidad”.

```
83 DROP PROCEDURE IF EXISTS `lp`.`deleteunidad` $$
84 CREATE DEFINER=`root`@`localhost` PROCEDURE `lp`.`deleteunidad`( IN launidad VARCHAR (15))
85 BEGIN
86 DELETE FROM `lp`.`unidad`
87 WHERE unidad.Nombre = launidad;
88 END $$
```

Microsoft Paint 6.0

Figura 6.15 Fragmento de código del procedimiento almacenado para borrar una fila en una tabla en MySQL

La salida del procedimiento para borrar en la tabla “Unidad” se muestra en la Fig. 6.16. Se eliminó de dicha tupla la entrada “Libras” agregada con los anteriores procedimientos almacenados.



```
mysql> CALL deleteunidad('Libras') //
Query OK, 1 row affected (0.06 sec)

mysql> CALL showunidad() //
+----+-----+
| ID | Nombre |
+----+-----+
| 1  | Celsius |
| 2  | Fahrenheit |
| 3  | Gramos |
+----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

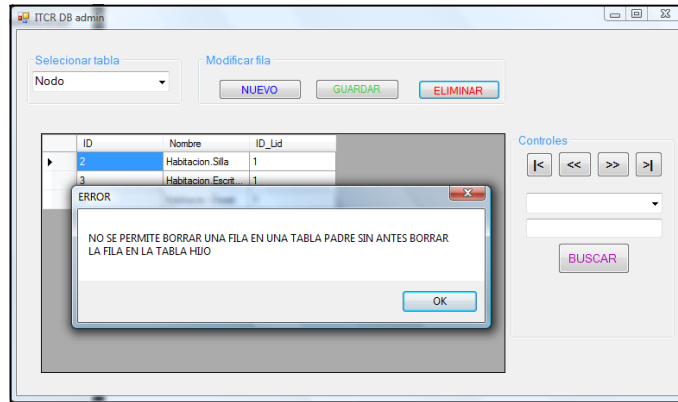
mysql>
```

Microsoft Paint 6.0

Figura 6.16 Resultado de llamar el procedimiento almacenado para borrar una fila de una tabla en MySQL

6.1.4 Resultados de la comprobación de la estructura del modelo relacional diseñado

Para demostrar el correcto funcionamiento de la estructura del modelo relacional de las tablas se utilizó el algoritmo de verificación de la Fig. 5.20. El algoritmo se implementó utilizando la interfaz gráfica de usuario diseñada para administrar y probar la base de datos (ver Fig. 5.19).

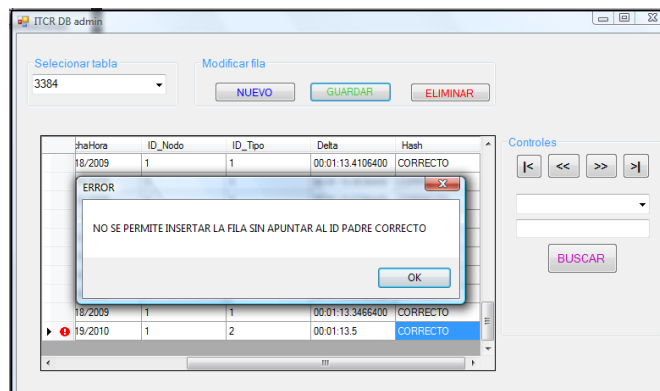


Microsoft Paint 6.0

Figura 6.17 Resultado de intentar borrar una fila en una tabla padre de la que depende una tabla hijo

Para cada caso de prueba de la estructura del algoritmo se implementó dos tipos mensajes de error generales de tal forma que un usuario no pueda ejecutar una acción que viole el modelo relacional.

Se comprobó la no independencia de una tabla con otra mediante las relaciones lógicas realizadas por medio de las llaves primarias y secundarias siguiendo el algoritmo de la Fig. 5.20; no se permite borrar una fila en una tabla padre al menos que se elimine la dependencia antes. Esta dependencia se comprobó para todas las tuplas en el diseño relacional dando como resultado un mensaje de error como el mostrado en la Fig. 6.17.



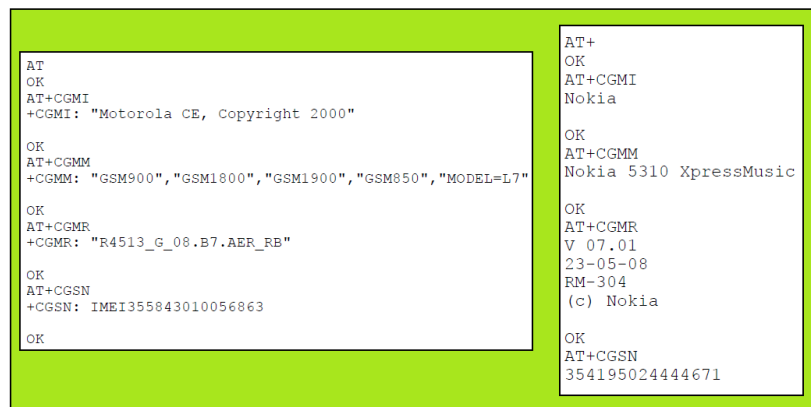
Microsoft Paint 6.0

Figura 6.18 Resultado de intentar insertar o actualizar una fila en una tabla padre de la que depende una tabla hijo

De manera análoga, se comprobó la existencia de la dependencia de un dato nuevo o de la actualización de un dato con su tabla padre. Como se muestra en la Fig. 6.18 el modelo rechaza la inserción o actualización de un dato si no se cumple con la dependencia en una tabla padre.

6.1.5 Resultados de la selección del dispositivo inalámbrico GSM

Para seleccionar el dispositivo GSM del proyecto se procedió a obtener las características de los dispositivos Nokia 5310 y Motorola L7. Como se observa en la Fig. 6.19 ambos responden a los comandos AT para obtener sus características principales.



```
AT
OK
AT+CGMI
+CGMI: "Motorola CE, Copyright 2000"
OK
AT+CGMM
+CGMM: "GSM900", "GSM1800", "GSM1900", "GSM850", "MODEL=L7"
OK
AT+CGMR
+CGMR: "R4513_G_08.B7.AER_RB"
OK
AT+CGSN
+CGSN: IMEI355843010056863
OK

AT+
OK
AT+CGMI
Nokia
OK
AT+CGMM
Nokia 5310 XpressMusic
OK
AT+CGMR
V 07.01
23-05-08
RM-304
(c) Nokia
OK
AT+CGSN
354195024444671
```

Microsoft Paint 6.0

Figura 6.19 Características de los dispositivos Motorola L7 y Nokia 5310

Cada uno presenta un software diferente. El Motorola L7 utiliza un sistema operativo básico cuya versión es la R4513_G_08.B7.AER_RB. El Nokia 5310 utiliza el sistema operativo Symbian versión 07.01 RM-304.

Utilizando el algoritmo de la Fig. 5.14 se obtuvo los resultados de la Fig. 6.20 y Fig. 6.21. La respuesta de cada uno de los comandos muestran que en el dispositivo Nokia 5310 no se acepta el comando de inicialización para la configuración de la notificación del arribo de nuevos mensajes (CNMI).

```
AT
OK
AT+CMGF=?
+CMGF: (0,1)

OK
AT+CMGF?
+CMGF: 0

OK
AT+CMGF=1
OK
AT+CNMI=?
+CNMI: (0,3),(0-2),(0),(0),(0)

OK
AT+CMGR?
OK
```

Microsoft Paint 6.0

Figura 6.20 Comandos AT de verificación y configuración del funcionamiento de un Motorola L7

Al utilizar el comando de lectura de un mensaje (CMGR) la respuesta del dispositivo Nokia también fue negativa. El Motorola L7, por su parte, sí aceptó los comandos de inicialización y de lectura. Se seleccionó éste hardware para el proyecto como el dispositivo de acceso al medio inalámbrico GSM.

```
AT
OK
AT+CMGF=?
+CMGF: (0,1)

OK
AT+CMGF?
+CMGF: 0

OK
AT+CMGF=1
OK
AT+CNMI=?
ERROR
AT+CMGR?
ERROR
```

Microsoft Paint 6.0

Figura 6.21 Comandos AT de verificación y configuración del funcionamiento de un Nokia 5310

6.1.6 Resultados de la implementación de las rutinas de inicialización, envío y recepción de información

En conjunto, la implementación de las rutinas de inicialización del dispositivo GSM, de envío y recepción de información forman el software para el control del dispositivo GSM (ver Fig. 5.1).

La implementación del algoritmo de la Fig. 5.16 dio como resultado un bloque de código dentro del programa para controlar un dispositivo GSM. El método que sigue la rutina de la figura en cuestión se llamó “**cnmi ()**” como indicativo de la utilización del comando para la configuración del registro de indicación de nuevo mensaje. La Fig. 6.22 muestra el fragmento de código del método para la inicialización del dispositivo Motorola L7.

```
93 public int cnmi()  
94 {  
95  
96     if (miserial.IsOpen)  
97     {  
98         miserial.Write("AT+CMGF=1\r"); //Se inicializa el modo texto del mensaje  
99         miserial.Write("AT+CNMI=3,1,0,0,0\r");//Para que avise sobre nuevo mensaje!
```

Microsoft Paint 6.0

Figura 6.22 Fragmento de código para la inicialización del dispositivo GSM

La implementación de la rutina de la Fig. 5.17 resultó en un método llamado “**enviar_sms (celnum, texto)**”. La rutina utiliza los parámetros del número de teléfono destino y el texto o mensaje que se quiere enviar. Un fragmento del código resultante del algoritmo diseñado se muestra en la Fig. 6.23.

```
73 public int enviar_sms(string celnum, string texto)  
74 {  
75     if (miserial.IsOpen)  
76     {  
77         miserial.Write("AT+CMGF=1\r");  
78         Thread.Sleep(3000); //Espera 3ms  
79         miserial.Write("AT+CMGS=\"" + celnum + "\"\r\n");  
80         miserial.Write(texto + (char)26);
```

Microsoft Paint 6.0

Figura 6.23 Fragmento de código para enviar información con el dispositivo GSM

La rutina de recepción de la Fig. 5.17 se implementó mediante un método llamado “**DataReceived ()**” en el código para el software controlador del dispositivo GSM. La Fig. 6.24 muestra una parte del código resultante para la lectura de información en el nodo central.

```
187 public static void DataReceived(object sender, SerialDataReceivedEventArgs e)
188 {
189     //Nueva instancia del puerto serial
190     SerialPort miptr = (SerialPort)sender;
191
192     //Descarta información en el buffer de salida para no activar el evento
193     miptr.DiscardOutBuffer();
194
195     //Lee lo que hay en el buffer de entrada
196     string resultado = miptr.ReadExisting();
197     System.Diagnostics.Debug.WriteLine(resultado); //El msj3 contiene los datos
198     //Si el resultado contiene "+CMTI:" se hacen dos splits para obtener
199     //la localidad en memoria en el modem para leer el mensaje.
200     if (resultado.Contains("+CMTI:"))
201     {
202         string[] split1 = resultado.Split(':'); //primer split
203         string[] locmem = split1[1].Split(','); //segundo split
204
205         //Ahora con locmem[1] puedo ir a leer mensaje!
206         miptr.Write("AT+CMGR=" + locmem[1] + "\r\n");

```

Microsoft Paint 6.0

Figura 6.24 Fragmento de código para recibir información con el dispositivo GSM

6.1.7 Resultados del funcionamiento del sistema

Para comprobar el funcionamiento del sistema de organización de la información se procedió a ejecutar la prueba del bombardeo de mensajes, la prueba de integridad de información y la medición del tiempo de retraso en el envío-recepción de un mensaje.

Se envió una cantidad determinada de mensajes conteniendo una trama de información tipificada por el protocolo de comunicaciones diseñado. La composición de la trama enviada se detalla en Fig. 5.13. A este paquete de información se le adiciona el tiempo inicial para medir el retardo en el envío-recepción de un mensaje.

El día 04 de Enero del 2010 se envió la cantidad de 50 mensajes en tres horarios diferentes al sistema de información de manera continua para observar su

funcionamiento. Estas pruebas de envío y recepción se realizaron en la misma celda de servicio.

La Tabla 6.1 detalla el número de la trama, el tiempo de retraso y el resultado de la comparación del hash para una prueba de envío-recepción de 50 paquetes de información el 04 de Enero del 2010 a las 06:30 am en Santa Ana en la misma celda de servicio GSM. Los valores de esta tabla se utilizaron para calcular los datos de la Tabla 6.2 con el fin de obtener la desviación estándar.

La ecuación 6.1 corresponde al tiempo de retardo promedio y la ecuación 6.2 es para el cálculo de la desviación estándar;

$$\Delta \bar{t} = \frac{\sum_{i=1}^{n=50} \Delta t_i}{n} \quad (6.1)$$

$$S_{\Delta t} = \sqrt{\frac{\sum_{i=1}^{n=50} (\Delta t_i - \Delta \bar{t})^2}{n-1}} \quad (6.2)$$

Tabla 6.1 Tiempo de retardo y resultado de la comparación del hash de 50 paquetes de información enviados el 4 de Enero del 2010 a las 6:30 am.

Trama	Tiempo de retardo (s)	Resultado del hash
1	70.5	CORRECTO
2	71.3	CORRECTO
3	71.1	CORRECTO
4	69.8	CORRECTO
5	73.1	CORRECTO
6	73.7	CORRECTO
7	72.1	CORRECTO
8	73.7	CORRECTO
9	71.1	CORRECTO
10	69.5	CORRECTO
11	72.1	CORRECTO
12	70.2	CORRECTO
13	69.7	CORRECTO
14	73.3	CORRECTO
15	69	CORRECTO
16	73.3	CORRECTO
17	71.5	CORRECTO
18	72.8	CORRECTO
19	73.8	CORRECTO
20	71.8	CORRECTO
21	73.7	CORRECTO
22	74	CORRECTO
23	73.7	CORRECTO
24	74	CORRECTO
25	73.8	CORRECTO
26	73.7	CORRECTO
27	72.2	CORRECTO
28	71.3	CORRECTO
29	71.1	CORRECTO
30	70.9	CORRECTO
31	72.6	CORRECTO
32	73.5	CORRECTO
33	73.1	CORRECTO
34	71.5	CORRECTO
35	72.9	CORRECTO
36	73.7	CORRECTO
37	74.5	CORRECTO
38	73.7	CORRECTO
39	71.6	CORRECTO
40	73.8	CORRECTO
41	73.7	CORRECTO
42	72.5	CORRECTO
43	73.7	CORRECTO
44	71.4	CORRECTO
45	69.8	CORRECTO
46	73.7	CORRECTO
47	68.9	CORRECTO
48	73.7	CORRECTO
49	69.7	CORRECTO
50	69	CORRECTO

Tabla 6.2 Valores para el cálculo de la desviación estándar del tiempo de retardo para la prueba del 04 de Enero del 2010 a las 6:30 am.

Δt_i (s)	$\Delta t_i - \bar{\Delta t}_i$	$(\Delta t_i - \bar{\Delta t}_i)^2$
70.5	-1.676	2.808976
71.3	-0.876	0.767376
71.1	-1.076	1.157776
69.8	-2.376	5.645376
73.1	0.924	0.853776
73.7	1.524	2.322576
72.1	-0.076	0.005776
73.7	1.524	2.322576
71.1	-1.076	1.157776
69.5	-2.676	7.160976
72.1	-0.076	0.005776
70.2	-1.976	3.904576
69.7	-2.476	6.130576
73.3	1.124	1.263376
69	-3.176	10.086976
73.3	1.124	1.263376
71.5	-0.676	0.456976
72.8	0.624	0.389376
73.8	1.624	2.637376
71.8	-0.376	0.141376
73.7	1.524	2.322576
74	1.824	3.326976
73.7	1.524	2.322576
74	1.824	3.326976
73.8	1.624	2.637376
73.7	1.524	2.322576
72.2	0.024	0.000576
71.3	-0.876	0.767376
71.1	-1.076	1.157776
70.9	-1.276	1.628176
72.6	0.424	0.179776
73.5	1.324	1.752976
73.1	0.924	0.853776
71.5	-0.676	0.456976
72.9	0.724	0.524176
73.7	1.524	2.322576
74.5	2.324	5.400976
73.7	1.524	2.322576
71.6	-0.576	0.331776
73.8	1.624	2.637376
73.7	1.524	2.322576
72.5	0.324	0.104976
73.7	1.524	2.322576
71.4	-0.776	0.602176
69.8	-2.376	5.645376
73.7	1.524	2.322576
68.9	-3.276	10.732176
73.7	1.524	2.322576
69.7	-2.476	6.130576
69	-3.176	10.086976
$\Sigma = 3608.8$	-1.1E-13	129.6712

Los valores de la Tabla 6.2 se calculan como muestran las ecuaciones de sus columnas. Estas ecuaciones se obtienen de la ecuación 6.2 para facilitar el cálculo de la desviación estándar mediante una tabla.

Así, aplicando la ecuación 6.1 a los datos de la Tabla 6.2 se obtiene como resultado que el tiempo de retardo promedio es de 72.176 segundos. La desviación estándar alrededor de este valor promedio, aplicando la ecuación 6.2 para el valor de la sumatoria final en la tercera columna de la Tabla 6.2, es de 1.627 segundos. Para comprender este valor se calcula el coeficiente de dispersión dado por la ecuación 6.3 que de detalla a continuación:

$$C.V. = \frac{S_{\Delta t} \cdot 100}{\Delta \bar{t}} \quad (6.3)$$

El resultado obtenido de aplicar la ecuación 6.3 es de 2.23% para el coeficiente de dispersión. El 100% de los paquetes enviados se recibieron con éxito y sin corrupción según el resultado de la comparación de hash. La Tabla 6.3 muestra un resumen de los principales resultados.

Tabla 6.3 Resumen de los resultados del análisis de los datos recibidos en el nodo central el 04 de Enero del 2010 a las 6:30 am

$\Delta \bar{t}$ (s)	$S_{\Delta t}$ (s)	C.V. (%)	Hash exitosos (%)
72.176	1.627	2.23	100

El mismo día, se realizó el procedimiento de enviar 50 paquetes de información mediante en mensajes de texto en la tarde. A las 12:10 pm del día 04 de Enero del 2010 se envió esta seguidilla de mensajes ubicado en la misma celda en Santa Ana. La Tabla 6.4 muestra los valores obtenidos del tiempo de retardo y el resultado de la comparación del código hash.

Tabla 6.4 Tiempo de retardo y resultado de la comparación del hash de 50 paquetes de información enviados el 04 de Enero del 2010 a las 12:10 pm.

Trama	Tiempo de retardo (s)	Resultado del hash
1	70.5	CORRECTO
2	71.3	CORRECTO
3	71.1	CORRECTO
4	69.1	CORRECTO
5	69.8	CORRECTO
6	71.1	CORRECTO
7	72.1	CORRECTO
8	70.9	CORRECTO
9	71.1	CORRECTO
10	69.5	CORRECTO
11	72.1	CORRECTO
12	70.2	CORRECTO
13	69.7	CORRECTO
14	69.4	CORRECTO
15	69	CORRECTO
16	72	CORRECTO
17	70.5	CORRECTO
18	72.1	CORRECTO
19	71.4	CORRECTO
20	71.8	CORRECTO
21	70.8	CORRECTO
22	71.2	CORRECTO
23	71.5	CORRECTO
24	72.1	CORRECTO
25	70.8	CORRECTO
26	70.37	CORRECTO
27	70.7	CORRECTO
28	71.1	CORRECTO
29	70.9	CORRECTO
30	70.9	CORRECTO
31	72.6	CORRECTO
32	72.2	CORRECTO
33	72.1	CORRECTO
34	71.5	CORRECTO
35	72	CORRECTO
36	71.8	CORRECTO
37	70.9	CORRECTO
38	70.8	CORRECTO
39	71.6	CORRECTO
40	69.9	CORRECTO
41	70.1	CORRECTO
42	71	CORRECTO
43	71.1	CORRECTO
44	69.8	CORRECTO
45	72.5	CORRECTO
46	72.5	CORRECTO
47	69.1	CORRECTO
48	72.1	CORRECTO
49	70	CORRECTO
50	69	CORRECTO

Tabla 6.5 Valores para el cálculo de la desviación estándar del tiempo de retardo para la prueba del 04 de Enero del 2010 a las 12:10 pm.

Δt_i (s)	$\Delta t_i - \bar{\Delta t}_i$	$(\Delta t_i - \bar{\Delta t}_i)^2$
70.5	-0.4534	0.205572
71.3	0.3466	0.120132
71.1	0.1466	0.021492
69.8	-1.8534	3.435092
73.1	-1.1534	1.330332
73.7	0.1466	0.021492
72.1	1.1466	1.314692
73.7	-0.0534	0.002852
71.1	0.1466	0.021492
69.5	-1.4534	2.112372
72.1	1.1466	1.314692
70.2	-0.7534	0.567612
69.7	-1.2534	1.571012
73.3	-1.5534	2.413052
69	-1.9534	3.815772
73.3	1.0466	1.095372
71.5	-0.4534	0.205572
72.8	1.1466	1.314692
73.8	0.4466	0.199452
71.8	0.8466	0.716732
73.7	-0.1534	0.023532
74	0.2466	0.060812
73.7	0.5466	0.298772
74	1.1466	1.314692
73.8	-0.1534	0.023532
73.7	-0.5834	0.340356
72.2	-0.2534	0.064212
71.3	0.1466	0.021492
71.1	-0.0534	0.002852
70.9	-0.0534	0.002852
72.6	1.6466	2.711292
73.5	1.2466	1.554012
73.1	1.1466	1.314692
71.5	0.5466	0.298772
72.9	1.0466	1.095372
73.7	0.8466	0.716732
74.5	-0.0534	0.002852
73.7	-0.1534	0.023532
71.6	0.6466	0.418092
73.8	-1.0534	1.109652
73.7	-0.8534	0.728292
72.5	0.0466	0.002172
73.7	0.1466	0.021492
71.4	-1.1534	1.330332
69.8	1.5466	2.391972
73.7	1.5466	2.391972
68.9	-1.8534	3.435092
73.7	1.1466	1.314692
69.7	-0.9534	0.908972
69	-1.9534	3.815772
$\Sigma = 3547.67$	-1.6E-13	49.53832

Con la ecuación 6.1 aplicada a los datos de la Tabla 6.4 se obtiene como resultado que el tiempo de retardo promedio es de 70.9534 segundos. La desviación estándar alrededor de este valor promedio, aplicando la ecuación 6.2 para el valor de la sumatoria final en la tercera columna de la Tabla 6.5, es de 1.005 segundos. El coeficiente de dispersión dado por la ecuación 6.3 es de 1.41%. El 100% de los paquetes enviados se recibieron con éxito y sin corrupción según el resultado del hash. La Tabla 6.6 muestra el resumen de los principales resultados.

Tabla 6.6 Resumen de los resultados del análisis de los datos recibidos en el nodo central el 04 de Enero del 2010 a las 12:10 am.

$\Delta \bar{t}$ (s)	$S_{\Delta t}$ (s)	C.V. (%)	Hash exitosos (%)
70.9534	1.627	1.41	100

Al final del día a las 17 horas del 04 de Enero del 2010 se realizó la misma prueba de envío de 50 mensajes cada uno con el contenido de un paquete de información. Esta prueba fue realizada en la misma celda del servicio GSM en la localidad de Santa Ana, San José. El resultado de la recepción de estos datos se muestra en detalle en la Tabla 6.7.

Tabla 6.7 Tiempo de retardo y resultado de la comparación del hash de 50 paquetes de información enviados el 4 de Enero del 2010 a las 05:00 pm.

Trama	Tiempo de retardo (s)	Resultado del hash
1	70.5	CORRECTO
2	71.3	CORRECTO
3	71.1	CORRECTO
4	69.1	CORRECTO
5	69.8	CORRECTO
6	71.1	CORRECTO
7	72.1	CORRECTO
8	70.9	CORRECTO
9	71.1	CORRECTO
10	69.5	CORRECTO
11	72.1	CORRECTO
12	70.2	CORRECTO
13	69.7	CORRECTO
14	69.4	CORRECTO
15	69	CORRECTO
16	72	CORRECTO
17	70.5	CORRECTO
18	72.1	CORRECTO
19	71.4	CORRECTO
20	71.8	CORRECTO
21	70.8	CORRECTO
22	71.2	CORRECTO
23	71.5	CORRECTO
24	72.1	CORRECTO
25	70.8	CORRECTO
26	70.37	CORRECTO
27	70.7	CORRECTO
28	71.1	CORRECTO
29	70.9	CORRECTO
30	70.9	CORRECTO
31	72.6	CORRECTO
32	72.2	CORRECTO
33	72.1	CORRECTO
34	71.5	CORRECTO
35	72	CORRECTO
36	71.8	CORRECTO
37	70.9	CORRECTO
38	70.8	CORRECTO
39	71.6	CORRECTO
40	69.9	CORRECTO
41	70.1	CORRECTO
42	71	CORRECTO
43	71.1	CORRECTO
44	69.8	CORRECTO
45	72.5	CORRECTO
46	72.5	CORRECTO
47	69.1	CORRECTO
48	72.1	CORRECTO
49	70	CORRECTO
50	69	CORRECTO

Tabla 6.8 Valores para el cálculo de la desviación estándar del tiempo de retardo para la prueba del 04 de Enero del 2010 a las 05:00 pm.

Δt_i (s)	$\Delta t_i - \bar{\Delta t}_i$	$(\Delta t_i - \bar{\Delta t}_i)^2$
73.8	0.384	0.147456
73.8	0.384	0.147456
73.9	0.484	0.234256
72.9	-0.516	0.266256
74	0.584	0.341056
73.7	0.284	0.080656
73.8	0.384	0.147456
73.7	0.284	0.080656
73.8	0.384	0.147456
72.8	-0.616	0.379456
74	0.584	0.341056
74.1	0.684	0.467856
73.8	0.384	0.147456
72.8	-0.616	0.379456
73.5	0.084	0.007056
73.4	-0.016	0.000256
73.3	-0.116	0.013456
73.8	0.384	0.147456
73.7	0.284	0.080656
74	0.584	0.341056
72.9	-0.516	0.266256
72.9	-0.516	0.266256
73.4	-0.016	0.000256
73.4	-0.016	0.000256
73.5	0.084	0.007056
73.5	0.084	0.007056
73.7	0.284	0.080656
72.9	-0.516	0.266256
73.2	-0.216	0.046656
73.9	0.484	0.234256
73.8	0.384	0.147456
74.1	0.684	0.467856
72.8	-0.616	0.379456
73.1	-0.316	0.099856
72.7	-0.716	0.512656
72.9	-0.516	0.266256
73.1	-0.316	0.099856
73.3	-0.116	0.013456
73.4	-0.016	0.000256
73.1	-0.316	0.099856
72.8	-0.616	0.379456
72.8	-0.616	0.379456
73.5	0.084	0.007056
73.6	0.184	0.033856
73.4	-0.016	0.000256
73.5	0.084	0.007056
72.9	-0.516	0.266256
72.5	-0.916	0.839056
73.8	0.384	0.147456
73.8	0.384	0.147456
$\Sigma = 3670.8$	1.71E-13	9.3672

Con la ecuación 6.1 aplicada a los datos de la Tabla 6.7 se obtiene como resultado que el tiempo de retardo promedio es de 73.416 segundos. La desviación estándar alrededor de este valor promedio, aplicando la ecuación 6.2 para el valor de la sumatoria final en la tercera columna de la Tabla 6.8, es de 0.437 segundos. El coeficiente de dispersión dado por la ecuación 6.3 es de 0.595 %. El 100 % de los paquetes enviados se recibieron con éxito y sin corrupción según el resultado del hash en la Tabla 6.7. La Tabla 6.9 muestra el resumen de los principales resultados.

Tabla 6.9 Resumen de los resultados del análisis de los datos recibidos en el nodo central el 04 de Enero del 2010 a las 05:00 pm.

$\Delta\bar{t}$ (s)	$S_{\Delta t}$ (s)	C.V. (%)	Hash exitosos (%)
73.416	0.437	0.595	100

Finalmente, se puede obtener un promedio del tiempo de retardo general y desviación estándar de todos los valores del tiempo de retardo medida en las tres pruebas de ese día para los 150 mensajes como se muestra en la Tabla 6.10.

Tabla 6.10 Resumen del promedio general del tiempo de retardo y desviación general de un total de 150 mensajes recibidos el 04 de Enero del 2010.

$\Delta\bar{t}$ (s)	$S_{\Delta t}$ (s)	C.V. (%)	Hash exitosos (%)
72.1818	1.511	2.093	100

6.2 Análisis

Uno de los primeros pasos para el desarrollo de la solución fue la creación del esquema lógico de una base de datos en un SGBD. El nombre asignado al esquema lógico fue “lp” de las iniciales de Load Point para el proyecto. Para asegurarse de la existencia después de introducir los comandos en SQL en el sistema gestor de bases de datos se obtuvo el resultado de la Fig. 6.1. El esquema “lp” en el SGBD de la Fig. 6.1 ubica en el contexto las tablas y los procedimientos almacenados desarrollados mediante las rutinas respectivas. El uso del lenguaje estructurado se debe a la escogencia del sistema gestor de base de datos MySQL.

Los resultados de aplicar la rutina general para la creación de una tupla en el SGBD de la Fig. 5.7 se muestran en la Fig. 6.3-6.7. Se nota en estas figuras que debido al lenguaje estructurado de consultas el código para la implementación de cada una de las tablas es de fácil lectura. Es importante señalar de cada figura, que al momento de implementar cada tabla con el código indicado, si hay una tabla existente con el mismo nombre bajo el esquema de base de datos “lp”, la tabla se eliminó para dar paso a una nueva. Esto se debe a que la creación de tablas se hace una única vez al inicio del proceso de poner en funcionamiento la base de datos. Para la definición de cada tabla se indica a cual esquema lógico pertenece ya que puede haber más de una tabla con el mismo nombre pero en un esquema diferente en el mismo SGBD. De las figuras mencionadas se puede observar que a cada columna se le define el tipo de dato que va a almacenar debido a las reglas generales de un modelo relacional en el que los datos de una columna deben ser agrupados con el mismo tipo. La Fig. 6.4 y Fig. 6.5 comprueban que estas tablas son del tipo padre puras ya que solo se define una única llave primaria y sin llaves secundarias; en contraste con el resto de tuplas en las que sí se define al menos una llave extranjera. La Fig. 6.8 muestra el resultado de implementar cada línea del lenguaje SQL en el SGBD; se observa que al introducir el comando para mostrar tablas se lista cada una de las tablas creadas en el esquema de base de datos definido anteriormente.

Para el funcionamiento básico de la base de datos se escribió el código de los procesos almacenados para cada tupla en el esquema “lp”. Las Fig. 6.9 corresponde al procedimiento almacenado para mostrar el contenido de la tabla llamada “Unidad”. El algoritmo para mostrar el contenido de una tabla entera no cambia en su estructura y lo único que cambia es el nombre al que hace referencia. El uso que se le da a este procedimiento se detalla en la Fig. 6.10. Se observa en esta figura que al utilizar este procedimiento se describen las columnas que contiene cada tabla, en este caso en particular se tiene una columna para ir asignando el número único de la llave primaria y la columna para el nombre del tipo de unidad que se desea almacenar. Además, se detalla cada entrada contenida en dicha tabla. La Fig. 6.11 demuestra el uso del algoritmo para insertar una nueva entrada en una tabla; en dicho código es importante señalar el uso de los parámetros de entrada del procedimiento para conocer donde se quiere almacenar en la tabla y que contenido quiere almacenar. La Fig. 6.12 es el resultado de aplicar el procedimiento de insertar para la tabla “Unidad”, se observa en esta figura que para insertar se llama al procedimiento (mediante la instrucción **CALL**) dándole como parámetros el número de entrada y el contenido de dicha entrada. Es importante tener claro el tipo de dato que se puede almacenar en cada columna para no obtener un error al momento de ejecutar el procedimiento. Para comprobar el resultado del procedimiento se ejecutó el procedimiento almacenado de mostrar tabla para ver la nueva entrada. La Fig. 6.13 describe el detalle del código de un procedimiento almacenado de la tabla “Unidad” para actualizar una fila, se denota en la figura que la función de dos parámetros de entrada que corresponden a la unidad que se desea actualizar y el dato por el cual se desea actualizar. El efecto de ejecutar el proceso se muestra en la Fig. 6.14 donde se logró sustituir el contenido de la última entrada. Finalmente, el algoritmo para borrar la fila de una tabla muestra su resultado mediante la implementación del código de la Fig. 6.15. La Fig. 6.16 demuestra que mediante la indicación del contenido de la fila que quiere borrar se logra el efecto de dicho procedimiento.

La Fig. 6.17 y la Fig. 6.18 corresponde a la aplicación del algoritmo de la Fig. 5.20. Aquí se demuestra el funcionamiento de la estructura del modelo de la base de datos relacional. Para cada uno de los casos se mostró uno de dos tipos de error en la interfaz gráfica diseñada como se observa en las figuras. Cuando se hizo el intento de borrar una entrada de una tabla padre a la cual una entrada en una tabla hijo hacía una relación, el sistema indicó que no es permitido ejecutar el procedimiento de borrado para dicha fila según la Fig. 6.17. Esto demostró el diseño del modelo relacional en el que primero se debe de borrar la entrada que esté apuntado a la llave primaria en la tupla padre. Se comprobó lo anterior para cada una de las tablas tipo padre en la que se tenía una relación lógica; en contraposición sí fue permitido borrar una fila en una tabla padre de la cual no había dependencia de un dato en una tabla hijo. Cuando se intentó insertar una nueva entrada o actualizar una fila en una tabla hijo, si la llave extranjera apuntaba a una llave primaria inexistente se obtuvo el error de la Fig. 6.18. Para cada una de las tablas hijo se hizo dicha comprobación resultando con el mismo mensaje de salida de la aplicación lo que demostró la estructura del modelo relacional. Cuando la nueva entrada o la actualización cumplieron con las relaciones lógicas el procedimiento de inserción se ejecutó sin problemas nuevamente demostrando el éxito en la implementación del modelo.

El resultado de la Fig. 6.19 demostró que es posible utilizar un teléfono celular como un módem GSM debido a que se aceptan los comandos básicos de Hayes para obtener características de hardware y software. El resultado de la Fig. 6.20 y la Fig. 6.21 se utilizaron como criterio para la selección de uno de los dispositivos GSM con que se contaba. Se seleccionó el Motorola L7 sobre el Nokia 5310 debido a que como se detalla en éstas figuras, el Motorola L7 permite configurar los registros de configuración y leer los mensajes ubicados en la memoria del dispositivo mientras que el Nokia 5310 no permite configurar el registro para notificar sobre un nuevo mensaje y no permite el uso del comando para la lectura de un mensaje en memoria. La razón de que el Nokia no permite realizar dichas tareas se debe al sistema operativo que utiliza (Symbian), ya que es de última generación en donde el control

de la configuración de los registros y la administración de los recursos de la memoria se maneja de forma virtual.

Con el conocimiento de que las operaciones de configuración de los registros en el Motorola L7 eran posibles, se implementó las rutinas de inicialización, envío y recepción de información para el control de dicho dispositivo GSM como se muestra en los resultados de la Fig. 6.22-6.24. La Fig. 6.22 demuestra la manera en que se configura el registro del modo texto del mensaje mediante la asignación al registro el valor de uno y la configuración del registro para que se notifique sobre el arribo de un nuevo mensaje poniendo un tres en el campo del modo. El resultado de ejecutar este código permitió enviar mensajes en modo texto y alertar a la aplicación sobre la llegada de un nuevo mensaje al dispositivo. La configuración del registro CNMI en el segundo campo poniendo un valor de uno permitió el enviar, en el contenido de la alarma, la ubicación de la nueva información en el hardware. La Fig. 6.23 muestra el uso del comando para enviar un mensaje especificando el número del teléfono destino y el texto que desea enviar. Según el algoritmo de la Fig. 5.17, cuando ésta rutina es ejecutada satisfactoriamente se indica con un valor de uno el resultado exitoso y con un valor cero el fallo. Finalmente, en la Fig. 6.24 se detalla el resultado de aplicar la rutina de recepción de información en la que se hace uso del comando para la lectura de memoria. En el contenido de la indicación de un nuevo mensaje se puede extraer la localidad de memoria del mensaje en el dispositivo GSM por lo que esta rutina se vale de esta indicación para utilizar el comando de lectura de mensaje como se muestra en la figura. Mediante esta comunicación entre la aplicación y el dispositivo fue obtener el mensaje en un arreglo dinámico en la memoria del sistema en pocos milisegundos que tarda el hardware en entregar la respuesta a los comandos de Hayes. La rutina de recepción de mensajes hace uso del procedimiento almacenado de inserción específicamente de la tabla "Datos" para guardar el dato en la base de datos relacional que llega al dispositivo GSM. Con la facilidad de la implementación de las rutinas para la inicialización del dispositivo, para el envío de datos y recepción de datos se demuestra la flexibilidad del uso de un dispositivo GSM para múltiples aplicaciones.

Según los resultados de la Tablas 6.1, 6.4 y 6.7 se comprobó el éxito de las rutinas diseñadas para el envío y recepción de información por medio de un dispositivo GSM. De cincuenta mensajes de texto, conteniendo un paquete de datos especificado por el protocolo de comunicaciones, se logró recibir el 100 % de esos paquetes. Según la columna de la comprobación de la integridad de los datos fue posible obtener un conteo de 100 % de los datos sin corrupción en su contenido a los tres horarios distintos del día de la prueba. En la segunda columna de cada una de estas tablas se muestra el tiempo de retardo desde el momento en que el mensaje fue enviado al momento en que fue recibido por la aplicación en el nodo central para el almacenamiento de la información en la base de datos. Este dato dio una idea del tiempo que tarda el mensaje en la arquitectura de la red GSM del proveedor de servicios.

Los resultados de la Tabla 6.2, 6.5 y 6.8 se utilizaron para lograr un análisis estadístico del tiempo de retardo en el envío y recepción de información mediante el uso de la plataforma de mensaje corto.

En la Tabla 6.3 se tiene el resultado del análisis estadístico de los valores del tiempo de retardo de la tabla 6.1 para el día de prueba del 04 de Enero del 2010 a las 6:30 am en la localidad de Santa Ana utilizando la misma celda de radio frecuencia. Se observa que el valor promedio del tiempo de retardo para la prueba en la mañana fue de 72.176 segundos lo que indica que para ese día cada mensaje tardaba en promedio 1 minuto y 12 segundos en la red del operador de servicios. Mediante la desviación estándar se notó que los valores del tiempo de retardo son muy similares unos de otros desviándose del tiempo promedio por 1.6 segundos por lo que la muestra de la prueba es confiable. El coeficiente de dispersión de 2.23 % enfatiza en que se tiene un porcentaje muy bajo de dispersión entre dato y dato por lo que el tiempo promedio calculado es un valor aceptable. El 100 % de los datos llegaron sin corrupción lo que demuestra la calidad del uso del servicio de mensajes

costos y la fiabilidad en la implementación de la estructura del paquete de datos del protocolo utilizado.

La Tabla 6.6 muestra los resultados del análisis de los valores de la prueba en el horario de la tarde de la Tabla 6.4. La prueba del día 04 de Enero del 2010 a las 12:10 pm en la localidad de Santa Ana en la cobertura de la misma celda arroja que el tiempo de retardo promedio para el arribo del mensaje fue de 70.9534 segundos, lo que equivale a que el mensaje estuvo un tiempo promedio de 1 minuto y 10 segundos en la red del proveedor. La desviación estándar de dichos valores dicta que la diferencia promedio entre un valor y otro ronda los 1.627 segundos, lo que es muy similar a la desviación de la prueba del horario de la mañana y se puede decir entonces que es una prueba fiable. Esto se comprueba mediante el indicador del coeficiente de variación que presenta un 1.41 % de dispersión lo que permite concluir que en promedio el tiempo que tarda un mensaje entre el envío y la recepción para ese momento es muy similar uno de otro. La comprobación de la integridad de los datos arrojó el éxito del 100 % de los datos sin corrupción demostrando la fiabilidad del uso de los mensajes de texto como plataforma de envío de información.

La Tabla 6.9 resume los resultados estadísticos de los valores del tiempo promedio medidos para la prueba del 04 de Enero del 2010 a las 5 pm en la misma localidad y celda anterior. El tiempo de retraso promedio medido para ese momento fue de 73.416 segundos que comparado con las mediciones promedio anteriores es unos segundos mayor. Esto indica que el uso de la red GSM a esta hora de éste día fue mayor que el uso en los otros horarios. De todas formas, la desviación estándar muestra que la diferencia entre datos no es mayor a 0.437 segundos lo que indica que esta muestra es más confiable que las anteriores. El coeficiente de variación calculado es de 0.595 % indicando la poca dispersión entre cada tiempo de retardo lo que hace concluir que para esa hora del día los mensajes tardaban en la red un aproximado de 1 minuto y 13 segundos. El resultado de la comparación del código

hash para la comprobación de la integridad del dato recibido fue nuevamente satisfactorio en un 100 %.

Finalmente, la Tabla 6.10 resume el cálculo de todos los valores del tiempo de retardo del día en la misma locación y celda. Se observa que el tiempo de retardo general promedio fue de 72.1818 segundos y que la desviación estándar de los 150 datos medidos en las tres pruebas durante el día corresponde a 1.511 segundos con un coeficiente de dispersión de 2.093 %; estos datos lo que indican es que a pesar de que los valores de medición del tiempo de retardo para un día se hicieron en grupos separados la desviación entre ellos es cercana al 2%.

7 Capítulo 7: Conclusiones y recomendaciones

7.1 Conclusiones

1. El diseño de un sistema de organización de la información depende de la estructura de los datos, del medio para obtener los datos y de la aplicación que se busca del sistema.
2. Se comprobó que el funcionamiento de la estructura de un modelo de base de datos depende de la correcta implementación de una rutina para crear cada una de sus tablas.
3. La implementación de un modelo para organizar la información es simplificado mediante el uso de un sistema gestor de bases de datos y de los procedimientos almacenados para administrar el contenido de su estructura.
4. El porcentaje de éxito en la recepción de mensajes en el sistema de organización de la información fue de un 100 % para las tres pruebas realizadas en un día.
5. El funcionamiento del sistema de organización de la información permitió obtener que el tiempo promedio que tarda un mensaje desde el momento en que se envía al momento en que se recibe es de 1 minuto y 12 segundos en una misma celda de la red GSM.
6. El tiempo de retardo promedio general medido presenta una desviación estándar de 1.511 segundos lo que significa una dispersión de 2.093 % entre un dato y otro.
7. El porcentaje de integridad de los datos recibidos en el sistema de organización de la información fue de un 100 %

7.2 Recomendaciones

1. Se recomienda el uso de un módem GSM de fábrica en vez del uso de un teléfono móvil GSM como módem ya que los primeros están diseñados para aceptar por defecto los comandos básicos de Hayes.
2. Se recomienda realizar pruebas de campo exhaustivas del envío masivo de información mediante los mensajes cortos utilizando diferentes locaciones para cambiar las celdas de servicio y medir nuevos tiempos de retraso.
3. Se debe de considerar la realización de pruebas exhaustivas del envío masivo de información mediante la variación de los datos para aumentar la probabilidad de errores en la transmisión de información y así detectar fallas en el sistema.
4. Los futuros diseñadores deberán trabajar en un manejador de procesos en el momento en que se escale al funcionamiento de varios sistemas de información en un mismo nodo central.
5. Se recomienda optimizar el acceso inalámbrico del sistema de organización de la información de GSM 2.0 a UMTS de tercera generación debido a las velocidades que se pueden alcanzar con las nuevas plataformas de la red en Costa Rica.
6. Utilizar el modo de paquete de unidades de datos (PDU) para enviar información y así evitar que una combinación de caracteres en el modo texto genere problemas de interpretación de un dato como una instrucción en el sistema.
7. Los futuros diseñadores del sistema de organización deben de considerar la adición de nuevos procedimientos almacenados o en su defecto la modificación de éstos ante un cambio en el modelo de la base de datos.

8 Bibliografía y referencias

- [1] Carvajal Godínez, Johan. (2009). Presentación sobre desarrollo de la red inalámbrica en el ITCR, “**Diseño de un nodo con arquitectura abierta para redes inalámbricas de sensores: CRTECMOTE**”. Instituto Tecnológico de Costa Rica.
- [2] Escuela de Ingeniería en Electrónica. **La Electrónica**. [En Línea]. Última modificación: 18 Agosto 2008. Última visita: 06 Enero 2010. URL: <http://www.ie.itcr.ac.cr/>
- [3] Tarrío Alonso, Paula. (2008). Presentación introductoria de las redes inalámbricas de sensores, tipos de nodos. “**Redes de sensores, fundamentos y aplicaciones**”. Universidad Politécnica de Madrid.
- [4] Lewis, F.L. (2004). Ambientes inteligentes: tecnologías, aplicaciones y protocolos. “**Wireless Sensor Networks**”. Automation and Robotics Research Institute, Universidad de Texas.
- [5] Turletti, T. (1996) Historia del Sistema Global para Móviles. “**A brief Overview of the GSM Radio Interface**”. Laboratorio de Ciencias en la Computación, Instituto Tecnológico de Massachusetts.
- [6] Sicard, E., **Introduction to GSM**. [En Línea]. Última modificación: 27 Febrero 2009. Última visita: 05 Enero 2010. URL: <http://www.commsdesign.com>
- [7] International Engineering Consortium. (2007). Organización de ingenieros en la que publican artículos de ingeniería. “**Global System for Mobile Communication (GSM)**”. [En Línea]. Última modificación: 27 Febrero 2009. Última visita: 05 Enero 2010. URL: <http://www.iec.org/online/tutorials/>

[8] European Telecommunications Standards Institute (ETSI). (2009) Instituto para la publicación de estándares en las tecnologías de comunicación e información. “**ETSI TS 151 010-1 v8.1.0, Digital cellular telecommunications system: Mobile Station (MS) conformance specification**”. [En Línea]. Última modificación: 08 Enero 2010. Última visita: 08 Enero 2010. URL: <http://www.etsi.org/WebSite/Standards/Standard.aspx>

[9] Wikipedia. **Serial programming / modems and AT commands**. [En Línea]. Última modificación 18 diciembre 2009. Última visita: 09 Enero 2009. URL: http://en.wikibooks.org/wiki/Serial_Programming/Modems_and_AT_Commands.

[10] European Telecommunications Standards Institute (ETSI). (2009). Instituto para la publicación de estándares en las tecnologías de comunicación e información. “**ETSI TS 100 916 v7.4.0, Digital cellular telecommunications system: AT command set for GSM Mobile Equipment (ME)**”. [En Línea]. Última modificación: 08 Enero 2010. Última visita: 08 Enero 2010. URL: <http://www.etsi.org/WebSite/Standards/Standard.aspx>

[11] HarmoniousTech Limited. **Introduction to SMS messaging**. [En Línea]. Última modificación: 09 Enero 2010. Última visita: 09 Enero 2010. URL: <http://www.developershome.com/sms/smsIntro.asp>

[12] Wireless Sensor Networks Research Group. **Sending SMS with SqidBee®**. [En Línea]. Última modificación: 07 Enero 2010. Última visita: 08 Enero 2010. URL: <http://www.sensor-networks.org/index.php?page=0824024033>

[13] Wikipedia. **Bases de datos**. [En Línea]. Última modificación 04 Enero 2010. Última visita: 10 Enero 2009. URL: http://es.wikipedia.org/wiki/Bases_de_datos

[14] Peter Rob, Elie Semaan. (2004). Fundamentos del diseño de bases de datos. "***Databases: design, development & deployment***". Segunda edición. McGraw-Hill. California.

[15] Sun Microsystems. (2010). Sistema de gestión de bases de datos de código abierto. "***Software para bases de datos de código abierto***". [En Línea]. Última modificación: 11 Enero 2010. Última visita: 11 Enero 2010. URL: <http://www.mysql.com/about/>

[16] Harte, Lawrence; Levine, Richard; Levingston, Geoff. (1999). "***GSM superphones***". Primera edición. Mc-Graw Hill Professional. pp 105-119.

[17] Motorola, Inc. (2006). Technical information. "***Motorola G24 developers guide: AT commands reference manual***". USA.

9 Apéndices

A.1 Abreviaturas

AUC	Authentication Center (Centro de Autenticación)
bps	bits por segundo
BSC	Base Station Controller (Controlador Base-Estación)
BSS	Base Station System (Sistema Base-Estación)
BTS	Base Transceiver Station (Estación Base-Transductor)
CGI	Cell Global Identity (Identidad Global de la Celda)
DCS	Digital Cellular System (Sistema Digital Celular)
EIR	Equipment Identity Register (Registro de Identidad del Equipo)
GIWU	GSM Internetworking Unit (Unidad de Red GSM)
GMSC	Gateway Mobile Services Switching Center (Centro de Servicios de Conmutación Móvil de Salida)
GSM	Global System for Mobile Communications (Sistema Global para Comunicaciones Móviles)
HLR	Home Location Register (Registro de Localización)
Hz	Hertz
ITCR	Instituto Tecnológico de Costa Rica
k	kilo
kbps	kilobits por segundo
MHz	Mega Hertz
MSC	Mobile Services Switching Center (Centro de Servicios de Conmutación Móvil)
MSN	Mobile Service Node (Nodo de Servicio Móvil)
MXE	Message Center (Centro de Mensajes)
OMC	Operations and Maintenance Center (Centro de Operaciones y Mantenimiento)
OSS	Operation and Support System (Sistema de Operaciones y Soporte)
PLMN	Public Land Mobile Network (Public Land Mobile Network)
SMS	Short Message Service (Servicio de Mensajes Cortos)
SS	Switching System (Sistema de Conmutación)

TDMA Time Division Multiple Access (División del Tiempo/ Múltiple Acceso)

VLR Visitor Local Register (Registro de Visitas Local)

WSN Wireless Sensor Network (Red Inalámbrica de Sensores)